Package ‘BDgraph’

September 5, 2019

Version 2.61
Date 2019-09-05
Title Bayesian Structure Learning in Graphical Models using Birth-Death MCMC
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]
Maintainer Reza Mohammadi <a.mohammadi@uva.nl>
URL https://www.uva.nl/profile/a.mohammadi
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>.
Imports igraph
License GPL (>= 2)
Repository CRAN
NeedsCompilation yes
Date/Publication 2019-09-05 12:40:03 UTC

R topics documented:

  BDgraph-package .................................................. 2
  adj2link .......................................................... 4
  bdgraph .......................................................... 5
  bdgraph.mpl ....................................................... 8
  bdgraph.npn ...................................................... 10
  bdgraph.sim ...................................................... 11
  bf ................................................................. 14
  churn ............................................................. 15
  compare ........................................................ 16
  covariance ....................................................... 18
The R package **BDgraph** provides 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), 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:

BDgraph-package

Author(s)

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

References


See Also

bdgraph, bdgraph.mpl, bdgraph.sim, compare.rgwish

Examples

```r
## 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 GGMsbdgraph.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**

```r
adj2link( adj )
```

**Arguments**

- `adj` 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`.

**Value**

A matrix corresponding to the extracted links from graph structure.

**Author(s)**

Reza Mohammadi <a.mohammadi@uva.nl>

**References**


See Also

`link2adj`, `graph.sim`

Examples

```r
# 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**

```r
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, print = 1000,
        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 `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) 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 do not follow Gaussianity assumption (e.g. continuous non-Gaussian, discrete, or mixed dataset).
algorithm  A character with two options "bdmcmc" (default) and "rjmcmc". Option "bdmcmc" is based on birth-death MCMC algorithm. Option "rjmcmc" 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 BDgraph or the class "ssgraph" of R package 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.

save  Logical: if FALSE (default), the adjacency matrices are NOT saved. If TRUE, the adjacency matrices after burn-in are saved.

print  Value to see the number of iteration for the MCMC algorithm.

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:

\[
p\_links\quad \text{An upper triangular matrix which corresponds the estimated posterior probabilities of all possible links.}
\]

\[
K\_hat\quad \text{The posterior estimation of the precision matrix.}
\]

For the case "save = TRUE" is returned:

\[
sample\_graphs\quad \text{A vector of strings which includes the adjacency matrices of visited graphs after burn-in.}
\]

\[
graph\_weights\quad \text{A vector which includes the waiting times of visited graphs after burn-in.}
\]

\[
all\_graphs\quad \text{A vector which includes the identity of the adjacency matrices for all iterations after burn-in. It is needed for monitoring the convergence of the BD-MCMC algorithm.}
\]

\[
all\_weights\quad \text{A vector which includes the waiting times for all iterations after burn-in. It is needed for monitoring the convergence of the BD-MCMC algorithm.}
\]
Author(s)

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

References


See Also

bdgraph.mpl, bdgraph.sim, summary.bdgraph, compare

Examples

```r
## 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", "First 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 = "gcgm", iter = 10000 )

summary( bdgraph.obj )
```
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, print = 1000, 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 data that are discrete. Option "dgm-binary" is for discrete graphical models for the data that are binary.

transfer

For only discrete data which method = "dgm" or method = "dgm-binary".

algorithm

A character with two options "bdmcmc" (default) and "rjmcmc". Option "bdmcmc" is based on birth-death MCMC algorithm. Option "rjmcmc" is based on reversible jump MCMC algorithm. Option "hc" is based on hill-climbing algorithm; this algorithm is only for discrete 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; 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.

print Value to see the number of iteration for the MCMC algorithm.

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


**See Also**

`bdgraph`, `bdgraph.sim`, `summary.bdgraph`, `compare`

**Examples**

```r
# 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**

```r
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 "skeptic". Option "shrinkage" is for the shrunken transformation, option "truncation" is for the truncated transformation and option "skeptic" is for the non-paranormal skeptic transformation. For more details see references.

- `npn.thresh` The truncation threshold; it is only for the truncated transformation \((\text{npn} = \text{"truncation"})\). The default value is \(1/(4n^{1/4} \sqrt \pi \log(n))\).
**Value**

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

**Author(s)**

Reza Mohammadi <a.mohammadi@uva.nl>

**References**


**See Also**

bdgraph.sim, bdgraph, bdgraph.mpl

**Examples**

```r
## 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

# 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)
```

---

**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 four different types of datasets, including multivariate Gaussian, non-Gaussian, discrete, or mixed data. This function can be used also for only simulating graphs by option \( n=\emptyset \), 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, 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", "discrete",
  "mixed", and "binary". For option "Gaussian", data are generated from multivariate
  normal distribution. For option "non-Gaussian", data are transfered multivariate
  normal distribution to continuous multivariate non-Gaussian distribution. For option
  "discrete", data are transfered from multivariate normal distribution to discrete multivariate
  distribution. For option "mixed", data are transfered from multivariate normal distribution to mixture of 'count', 'ordinal',
  'non-Gaussian', 'binary' and 'Gaussian', respectively. For option "binary",
  data are generated directly from the joint distribution, in this case \( p \) must be less
  than 17.

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="discrete", it is the number of categories for simulating discrete 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.

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> and Ernst Wit

References


See Also
graph.sim, bdgraph, bdgraph.mpl

Examples
```r
## 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

\[
\text{bf( num, den, bdgraph.obj, log = TRUE )}
\]

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 \texttt{graph.sim}. It can be an object with S3 class "sim" from function \texttt{bdgraph.sim}.
- bdgraph.obj: An object of S3 class "bdgraph", from function \texttt{bdgraph}. It also can be an object of S3 class "ssgraph", from the function \texttt{ssgraph} of R package \texttt{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


See Also

\texttt{bdgraph, bdgraph.mpl, compare, bdgraph.sim}
churn

Examples

```r
## 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 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**

```r
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

Examples
data( churn )
summary( churn )

cmp <- compare( target, est, est2 = NULL, est3 = NULL, est4 = NULL, main = NULL, vis = FALSE )

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

Usage

\[
\text{compare( target, est, est2 = NULL, est3 = NULL, est4 = NULL, main = NULL, vis = FALSE )}
\]

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>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 <code>bdgraph.sim</code>. It can be an object with S3 class “graph” from function <code>graph.sim</code>.</td>
</tr>
</tbody>
</table>
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 of R package 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

See Also
bdgraph, bdgraph.mpl, bdgraph.sim, plotroc
## 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)

covariance

### covariance

<table>
<thead>
<tr>
<th>covariance</th>
<th>Estimated covariance matrix</th>
</tr>
</thead>
</table>

#### 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` of R package `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>
geneExpression

References


See Also

bdgraph, precision, plinks

Examples

```r
## 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)
```

geneExpression Human gene expression dataset

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).
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 )

gnorm

Normalization constant for G-Wishart

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


See Also

`rgwish`, `rwish`

Examples

```r
## 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)
```

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 notes $i$ 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


See Also

bdgraph.sim, bdgraph, bdgraph.mpl

Examples

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

plot( adj )

adj
**Description**

Extract links from an adjacency matrix or an object of calsses "sim" from function `bdgraph.sim` and "graph" from function `graph.sim`.

**Usage**

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

**Arguments**

- **link**: An \((2 \times p)\) matrix or a data.frame corresponding to the links from the graph structure.
- **p**: The number of nodes of the graph.

**Value**

An adjacency matrix corresponding to the 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**


**See Also**

`adj2link`, `graph.sim`
Examples

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

link <- adj2link( adj )

link2adj( link, p = 6 )
```

Description

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

Usage

```r
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


plinks


See Also

bdgraph, bdgraph.mpl

Examples

```r
## 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

```r
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` of R package `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

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, ... )
plot.bdgraph

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 bdgraph() with option save = TRUE; See the examples.
...

Author(s)

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

References


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`.
- **main**: Graphical parameter (see plot).
- **vertex.color**: The vertex color which is according to R package `igraph`.
- **vertex.label.color**: The vertex label color which is according to R package `igraph`.
- **...**: System reserved (no specific usage).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References


plot.sim

See Also

graph.sim, bdgraph.sim

Examples

# Generating a 'random' graph
adj <- graph.sim( p = 10, graph = "random" )
plot( adj )
adj

plot.sim

Plot function for S3 class "sim"

Description

Visualizes structure of the true graph.

Usage

## S3 method for class 'sim'
plot( 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


See Also

graph.sim, bdgraph.sim
Examples

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

---

**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**

```r
plotcoda( bdgraph.obj, thin = NULL, control = TRUE, 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` of R package `ssgraph`.
- `thin` An option for getting fast result for a cumulative plot according to part of the iteration.
- `control` Logical: if TRUE (default) and the number of nodes is greater than 15, then 100 links randomly is selected for visualization.
- `main` Graphical parameter (see plot).
- `...` 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**


**See Also**

`bdgraph`, `bdgraph.mpl`, `traceplot`

**Examples**

```r
## 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**

```r
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 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 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` of R package `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


See Also

bdgraph, bdgraph.mpl, compare

Examples

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

# Running 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 results
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)
```
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` of R package `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


See Also

`bdgraph`, `covariance`, `plinks`
Examples

```r
## 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

```r
## 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


print.sim

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**


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


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 notes $i$ 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


See Also
gnorm, rwish

Examples

```r
# 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**

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

**Arguments**

- `n` Number of observations.
- `mean` Mean vector, default is `rep(0, length = ncol(sigma))`.
- `sigma` Positive definite covariance matrix, default is `diag(length(mean))`. 
Value

A numeric matrix with rows equal to \( n \) and columns equal to \( \text{length}(\text{mean}) \).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

See Also

bdgraph.sim, rwish, rgwish

Examples

```r
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 )
```

rwish

Sampling from Wishart distribution

Description

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

Usage

```r
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 \(A[i, j, k]\) 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


See Also

gnorm, rgwish

Examples

```r
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

```r
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` of R package `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 &lt;a.mohammadi@uva.nl&gt; and Ernst Wit

References


See Also

`bdgraph`, `bdgraph.mpl`

Examples

```r
## 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)

---

**summary.bdgraph**  
*Summary function for S3 class "bdgraph"*

### Description

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

### Usage

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

### Arguments

- **object**: An object of S3 class "bdgraph", from function `bdgraph`.
- **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 &lt;a.mohammadi@uva.nl&gt; and Ernst Wit

### References


surveyData

See Also

bdgraph, bdgraph.mpl

Examples

```r
## 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


Examples

```r
data( surveyData )
summary( surveyData )
```

---

**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**

```r
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` of R package `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**


**See Also**

- `plotcoda`, `bdgraph`, `bdgraph.mpl`
Examples

```r
## 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 discrete data

Description

Transfers discrete data, by counting the duplicated rows.

Usage

```r
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


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 = "discrete", cut = 2 )
r_data <- data.sim $ data
r_data

# Transfer the data
transfer( r_data )
Index

*Topic Wishart
  gnorm, 20
  rgwish, 37
  rwish, 39

*Topic data preprocessing
  bdgraph.npn, 10
  transfer, 45

*Topic datagen
  bdgraph.sim, 11

*Topic datasets
  churn, 15
  geneExpression, 19
  reinis, 36
  surveyData, 43

*Topic distribution
  rgwish, 37
  rmvnorm, 38
  rwish, 39

*Topic graphs
  adj2link, 4
  graph.sim, 21
  link2adj, 23

*Topic hplot
  plot.bdgraph, 26
  plot.graph, 28
  plot.sim, 29
  plotcoda, 30
  plotroc, 31
  traceplot, 44

*Topic iteration
  bdgraph, 5
  bdgraph.mpl, 8
  gnorm, 20

*Topic package
  BDgraph-package, 2

*Topic parameter learning
  covariance, 18
  precision, 33

*Topic print
  print.bdgraph, 34
  print.sim, 35

*Topic sampling algorithms
  bdgraph, 5
  bdgraph.mpl, 8

*Topic simulation
  bdgraph.sim, 11
  graph.sim, 21

*Topic structure learning
  bdgraph, 5
  bdgraph.mpl, 8
  bf, 14
  compare, 16
  pgraph, 24
  plinks, 25
  select, 40

  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
  hill_climb_mpl (BDgraph-package), 2
  sample_ug (BDgraph-package), 2
  log_mpl_binary (BDgraph-package), 2
  log_mpl_discrete (BDgraph-package), 2
  log_mpl_binary (BDgraph-package), 2
  log_mpl_discrete (BDgraph-package), 2
adj2link, 4, 23

BDgraph, 6, 9
BDgraph (BDgraph-package), 2
bdgraph, 3, 5, 10, 11, 13, 14, 17–19, 22, 24–27, 30–35, 37, 41–44
BDgraph-package, 2
bdgraph.mpl, 3, 7, 8, 11, 13, 14, 17, 22, 25–27, 31, 32, 35, 37, 41, 43–45
bdgraph.npn, 10
bdgraph.sim, 3–5, 7, 8, 10, 11, 14, 16, 17, 22–24, 29, 31, 35–37, 39, 45
bf, 14

churn, 15
compare, 3, 7, 10, 14, 16, 32
covariance, 18, 33
geneExpression, 19
gnorm, 20, 38, 40
graph.sim, 4, 5, 13, 14, 16, 21, 23, 28, 29, 31, 36, 37
huge, 17, 31
huge.select, 17, 31
igraph, 28

link2adj, 5, 23
plinks, 19, 25, 33
plot.bdgraph, 26
plot.graph, 28
plot.sim, 29
plotcoda, 30, 44
plotroc, 17, 31
precision, 19, 33
print.bdgraph, 34
print.sim, 35

reinis, 36
rgwish, 3, 21, 37, 39, 40
rmvnorm, 38
rwish, 21, 38, 39, 39

select, 40
ssgraph, 6, 9, 14, 17, 18, 25, 30, 31, 33, 41, 44
summary.bdgraph, 7, 10, 42
surveyData, 43

traceplot, 31, 44
transfer, 45