

Package ‘BDgraph’

October 24, 2012

Type Package

Title Gaussian Graphical Model determination based on birth-death MCMC methodology

Version 1.2

Date 2012-10-08

Author Abdolreza Mohammadi and Ernst Wit

Depends R (>= 2.14.0), network

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

Description The R package BDgraph is a statistical tool for Bayesian model determination in undirected Gaussian graphical models. The Bayesian methodology is based on birth-death MCMC algorithm. The main function is ‘bdmcmc’ which is a birth-death MCMC algorithm for Bayesian model determination in undirected Gaussian graphical models.

License GPL (>= 3)

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

Repository CRAN

Date/Publication 2012-10-24 14:34:14

R topics documented:

BDgraph-package	2
bdmcmc	3
bdmcmc.high	5
bdmcmc.low	7
I.g	9
phat	10
plotConvergency	11
plotLinks	12
prob.allg	13

prob.g	14
sample.gwishart	16
select.g	17

Index	19
--------------	-----------

BDgraph-package	<i>Bayesian model selection in Gaussian graphical models based on BDMCMC algorithm</i>
-----------------	--

Description

The R package BDgraph is a statistical tool for Bayesian model determination in undirected Gaussian graphical models based on birth-death MCMC methodology.

Details

This package provides an implementation of the procedures described in Mohammadi and Wit (2012). The main function is 'bdmcmc' which is birth-death MCMC algorithm for Bayesian model selection in Gaussian graphical models.

Functions:

bdmcmc	BDMCMC algorithm for undirected Gaussain graphical model determination
bdmcmc.high	BDMCMC algorithm for high-dimensional graphs
bdmcmc.low	BDMCMC algorithm for low-dimensional graphs
I.g	Computing normalizing constant of G-Wishart distribution
phat	Posterior edge inclusion probabilities
plotConvergency	Cumulative occupancy fractions for checking the convergency
plotLinks	Plot of posterior distribution for graphs according to number of their links
prob.allg	Posterior probability of all possible graphs
prob.g	Posterior probability for one special graph
sample.gwishart	Sampling from G-Wishart distribution
select.g	Selecting the best graphical models based on BDMCMC algorithm

Author(s)

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

References

- Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>
- 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.

bdmcmc	<i>BDMCMC algorithm for undirected Gaussain graphical model determination</i>
--------	---

Description

This function is the BDMCMC algorithm for model selection in undirected Gaussian graphical models based on birth-death MCMC methodology. It works for both low and high dimensional graphs.

Usage

```
bdmcmc(data, n = NULL, meanzero = FALSE, iter = 5000, burn = floor(iter/2),
        skip = 1, gamma.b = 1, prior.g = "Uniform", b = 3, D = NULL,
        A = NULL, MCiter = 10, print = FALSE, sumery = FALSE)
```

Arguments

data	it could be a matrix of the data ($p \times n$ matrix) or it could be a covariance matrix as $S = x'x$ which x is a matrix of the data.
n	the number of observations.
meanzero	logical: if TRUE that means the data have zero mean; if FALSE (default) the mean of the data should not be zero.
iter	the number of iterations for the BDMCMC algorithm.
burn	the number of burn-in iterations for the BDMCMC algorithm.
skip	it is the option for regularly saving part of the iterations. Default is 1 (for saving all iterations).
gamma.b	the birth rates for the birth-death process.
prior.g	a character for selecting a prior distribution for the graph. It can be either "Uniform" (default) or "Poisson". "Uniform" means discrete uniform distribution for prior distribution of the graph ($G \sim DU(\mathcal{G})$ in which \mathcal{G} is all possible graphical models). Also, "Poisson" means prior distribution for number of edges in the graph G has a truncated Poisson distribution ($degree(G) \sim TP(\gamma_b)$ in which for our model γ_b equal with a 'gamma.b' value).
b	the degree of freedom for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix.
D	the positive definite matrix for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix.
A	upper triangular matrix in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. This matrix shows the starting graphal model for BDMCMC algorithm.
MCiter	the number of iterations for Monte Carlo approximation of normalizing constant in G-Wishart distribution. We need it only when number of nodes in the graph is less than 8.

print	logical: if TRUE you will see the print of iteration, time, and number of edges in each iteration of BDMCMC algorithm; if FALSE (default), there is no any print order.
sumery	logical: if TRUE you will see the sumery result of BDMCMC algorithm; if FALSE (default), the output will be a list as below.

Value

Returns an output of the BDMCMC algorithm which is list-like and contains the following:

As	a list which includes the adjacency matrix for all iterations after burn-in iterations.
Ks	a list which includes the precision matrix for all iterations after burn-in iterations.
lambda	a vector which includes the waiting times for all iterations after burn-in iterations.
allA	a list which includes all adjacency matrix for all iterations (includes burn-in iterations). We need 'allA' list for checking the convergency of the BDMCMC algorithm.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc.low](#) and [bdmcmc.high](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))

bdmcmc(data, meanzero=T, iter=500, sumery=T)
```

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

bdmcmc.high

BDMCMC algorithm for high-dimensional graphs

Description

This function is a BDMCMC algorithm for model selection in undirected Gaussian graphical models based on birth-death MCMC methodology. This algorithm works well especially for high-dimensional graphs (graphical models with more than 8 nodes).

Usage

```
bdmcmc.high(data, n = NULL, meanzero = FALSE, iter = 5000, burn = floor(iter/2),
            skip = 1, gamma.b = 1, prior.g = "Uniform", b = 3, D = NULL,
            A = NULL, print = FALSE, sumery = FALSE)
```

Arguments

data	it could be a matrix of the data ($p \times n$ matrix) or it could be a covariance matrix as $S = x'x$ which x is a matrix of the data.
n	the number of observations.
meanzero	logical: if TRUE that means the data have zero mean; if FALSE (default) the mean of the data should not be zero.
iter	the number of iterations for the BDMCMC algorithm.
burn	the number of burn-in iterations for the BDMCMC algorithm.
skip	it is the option for regularly saving part of the iterations. Default is 1 (for saving all iterations).
gamma.b	the birth rates for the birth-death process.
prior.g	a character for selecting a prior distribution for the graph. It can be either "Uniform" (default) or "Poisson". "Uniform" means discrete uniform distribution for prior distribution of the graph ($G \sim DU(\mathcal{G})$ in which \mathcal{G} is all possible graphical models). Also, "Poisson" means prior distribution for number of edges in the graph G has a truncated Poisson distribution ($degree(G) \sim TP(\gamma_b)$ in which for our model γ_b equal with a 'gamma.b' value).
b	the degree of freedom for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix.
D	the positive definite matrix for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix.
A	upper triangular matrix in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. This matrix shows the starting graphal model for BDMCMC algorithm.

print	logical: if TRUE you will see the print of iteration, time, and number of edges in each iteration of BDMCMC algorithm; if FALSE (default), there is no any print order.
sumery	logical: if TRUE you will see the sumery result of BDMCMC algorithm; if FALSE (default), the output will be a list as below.

Value

Returns an output of the BDMCMC algorithm which is list-like and contains the following:

As	a list which includes the adjacency matrix for all iterations after burn-in iterations.
Ks	a list which includes the precision matrix for all iterations after burn-in iterations.
lambda	a vector which includes the waiting times for all iterations after burn-in iterations.
allA	a list which includes all adjacency matrix for all iterations (includes burn-in iterations). We need list 'allA' for checking the convergency of the BDMCMC algorithm.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc](#) and [bdmcmc.low](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))

bdmcmc.high(data, meanzero=T, iter=500, sumery=T)
```

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

bdmcmc.low

BDMCMC algorithm for low-dimensional graphs

Description

This function is the BDMCMC algorithm for model selection in undirected Gaussian graphical models based on birth-death MCMC methodology. The algorithm is not fast, so it is suitable only for low-dimensional graphs (roughly graphical models with less than 8 nodes).

Usage

```
bdmcmc.low(data, n = NULL, meanzero = FALSE, iter = 5000, burn = floor(iter/2),
            skip = 1, gamma.b = 1, prior.g = "Uniform", b = 3, D = NULL,
            A = NULL, MCiter = 10, print = FALSE, sumery = FALSE)
```

Arguments

data	it could be a matrix of the data ($p \times n$ matrix) or it could be a covariance matrix as $S = x'x$ which x is a matrix of the data.
n	the number of observations.
meanzero	logical: if TRUE that means the data have zero mean; if FALSE (default) the mean of the data should not be zero.
iter	the number of iterations for the BDMCMC algorithm.
burn	the number of burn-in iterations for the BDMCMC algorithm.
skip	it is the option for regularly saving part of the iterations. Default is 1 (for saving all iterations).
gamma.b	the birth rates for the birth-death process.
prior.g	a character for selecting a prior distribution for the graph. It can be either "Uniform" (default) or "Poisson". "Uniform" means discrete uniform distribution for prior distribution of the graph ($G \sim DU(\mathcal{G})$ in which \mathcal{G} is all possible graphical models). Also, "Poisson" means prior distribution for number of edges in the graph G has a truncated Poisson distribution ($degree(G) \sim TP(\gamma_b)$ in which for our model γ_b equal with a 'gamma.b' value).
b	the degree of freedom for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix.
D	the positive definite matrix for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix.
A	upper triangular matrix in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. This matrix shows the starting graphal model for BDMCMC algorithm.

MCiter	the number of iterations for Monte Carlo approximation of normalizing constant in G-Wishart distribution. We need it only when number of nodes in the graph is less than 8.
print	logical: if TRUE you will see the print of iteration, time, and number of edges in each iteration of BDMCMC algorithm; if FALSE (default), there is no any print order.
sumery	logical: if TRUE you will see the sumery result of BDMCMC algorithm; if FALSE (default), the output will be a list as below.

Value

Returns an output of the BDMCMC algorithm which is list-like and contains the following:

As	a list which includes the adjacency matrix for all iterations after burn-in iterations.
Ks	a list which includes the precision matrix for all iterations after burn-in iterations.
lambda	a vector which includes the waiting times for all iterations after burn-in iterations.
allA	a list which includes all adjacency matrix for all iterations (includes burn-in iterations). We need list 'allA' for checking the convergency of the BDMCMC algorithm.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc](#) and [bdmcmc.high](#)

Examples

```
## Not run:
require(MASS)

p <- 5 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (150 observations) from multivariate normal
```



```
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(150, c(rep(0,p)), solve(truK))

bdmcmc.low(data, meanzero=T, iter=500, sumery=T)

## End(Not run)
```

I.g

*Computing 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(A, b, D, MCiter = 500)
```

Arguments

A	upper triangular matrix in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$.
b	the degree of freedom for G-Wishart distribution, $W_G(b, D)$.
D	the positive definite matrix for G-Wishart distribution, $W_G(b, D)$.
MCiter	the number of iterations for the Monte Carlo approximation.

Value

the normalizing constant of G-Wishart distribution.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

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

```
A <- matrix(c(0,1,1,
              0,0,1,
              0,0,0), 3, 3, byrow = TRUE)
# matrix A shows full graph with 3 nodes and 3 edges
I.g(A, b=3, D=diag(3), MCiter=50)
```

phat

Posterior edge inclusion probabilities

Description

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

Usage

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

Arguments

output	a list which is the result of BDMCMC algorithm from the 'bdmcmc', 'bdmcmc.low', or 'bdmcmc.high' functions.
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 E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4
```

```
# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))

output <- bdmcmc(data, meanzero=T, iter=500, print=T)
phat(output, round=2)

## End(Not run)
```

plotConvergency

Cumulative occupancy fractions for checking the convergency

Description

Plot the cumulative occupancy fractions for all possible edges. It is a tool for checking the convergency of the BDMCMC algorithm.

Usage

```
plotConvergency(output, skip = 1)
```

Arguments

output	a list which is the result of BDMCMC algorithm from the 'bdmcmc', 'bdmcmc.low', or 'bdmcmc.high' functions.
skip	it is a option for getting fast result for a cumulative plot according to part of the iterations.

Details

Note that a spending time for this function dependes on the graph. It should be slow for the high-dimensional graphs. To make it faster you can choose bigger value for 'skip', as you can see in following example.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc.low](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))
output <- bdmcmc(data, meanzero=T, iter=2000, print=T)

# we run it for skip=5. For skip=1, it takes more time.
plotConvergency(output, skip=5)

## End(Not run)
```

plotLinks

Plot of posterior distribution for graphs according to number of their links

Description

This function is for plotting the posterior distribution of all possible graphs according to number of their links.

Usage

```
plotLinks(output, xlim = c(0, (nrow(output$As[[1]])) * (nrow(output$As[[1]] - 1)/2))
```

Arguments

output	a list which is the result of BDMCMC algorithm from the 'bdmcmc', 'bdmcmc.low', or 'bdmcmc.high' functions.
xlim	the x limits (min, max) of the plot.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

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

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))

output <- bdmcmc(data, meanzero=T, iter=500, print=T)
plotLinks(output)

## End(Not run)
```

prob.allg

Posterior probability of all possible graphs

Description

According to the output of the BDMCMC algorithm, this function gives us the posterior probability of all possible graphical models. Aslo, it give us all graphs that the BDMCMC algorithm visits them.

Usage

```
prob.allg(output)
```

Arguments

output a list which is the result of BDMCMC algorithm from the 'bdmcmc', 'bdm-cmc.low', or 'bdmcmc.high' functions.

Value

list.A a list which includes all the grpahs that the BDMCMC algorithm visits them.
prob.A a vector which includes posterior probabilities of all graphs in 'list.A'.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))

# First, we run the 'bdmcmc' function for small number of iterations
output <- bdmcmc(data, iter=40, burn=30, meanzero=T, print=T)
prob.allg(output)

## End(Not run)
```

prob.g

Posterior probability for a specific graph

Description

According to the output of the BDMCMC algorithm, this function gives us the posterior probability for a specific graph.

Usage

```
prob.g(A, output)
```

Arguments

A	upper triangular matrix in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$.
output	a list which is the result of BDMCMC algorithm from the 'bdmcmc', 'bdmcmc.low', or 'bdmcmc.high' functions.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
data <- mvrnorm(200, c(rep(0,p)), solve(truK))
output <- bdmcmc(data, meanzero=T, iter=1000, print=T)

A <- 0*truK
for (i in 1:(p-1)){
  for (j in (i+1):p){
    if (truK[i,j]!=0) A[i,j] <- 1
  }
}
A[1,p] <- 1
# matrix A shows circle graph with 8 links and 8 nodes which is the true graphical model
prob.g(A, output)

## End(Not run)
```

sample.gwishart

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

Tools for sampling from G-Wishart distribution according to Choleski decomposition of a Wishart variate with the identity as its scale parameter. The function uses the sampling method of Atay-Kayis and Massam (2005).

Usage

```
sample.gwishart(A, b, D, round = 3)
```

Arguments

A	upper triangular matrix in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$.
b	the degree of freedom for G-Wishart distribution, $W_G(b, D)$.
D	the positive definite matrix for G-Wishart distribution, $W_G(b, D)$.
round	a number for rounding all values to the specified number of decimal places (default 3).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

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

```
p <- 5
A <- matrix(0, p, p)
for (i in 1:(p-1)){
  for (j in (i+1):p){
    if (j==i+1) A[i,j] <- 1
  }
}
A[1,p] <- 1
# matrix A shows circle graph with 5 nodes and 5 edges
sample.gwishart(A, b=3, D=diag(p), round=1)
```

`select.g`*Selecting the best graphical models based on BDMCMC algorithm*

Description

According to output of the BDMCMC algorithm, this function gives us the graphical models with the highest posterior probabilities.

Usage

```
select.g(output, g = 2, K = FALSE)
```

Arguments

<code>output</code>	a list which is the result of the BDMCMC algorithm from the 'bdmcmc', 'bdmcmc.low', or 'bdmcmc.high' functions.
<code>g</code>	you can select the number of graphical models with highest probabilities (default 2).
<code>K</code>	logical: if TRUE you will see the estimation of precision matrix for the best graph; default is FALSE.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and E. Wit (2012). Efficient birth-death MCMC inference for Gaussian graphical models, arXiv:1210.5371. <http://arxiv.org/abs/1210.5371>

See Also

[bdmcmc](#) and [prob.allg](#)

Examples

```
## Not run:
require(MASS)

p <- 8 # number of nodes
# "truK" is the precision matrix of true graph
truK <- diag(p)
for (i in 1:(p-1)) truK[i,i+1] = truK[i+1,i] = 0.5
truK[1,p] <- 0.4
truK[p,1] <- 0.4

# generate the data (200 observations) from multivariate normal
# distribution with mean zero and percision matrix "truK"
```

```
data <- mvrnorm(200, c(rep(0,p)), solve(truK))  
  
output <- bdmcmc(data, meanzero=T, iter=500, print=T)  
select.g(output, K=TRUE)  
  
## End(Not run)
```

Index

*Topic **package**

BDgraph-package, [2](#)

BDgraph (BDgraph-package), [2](#)

BDgraph-package, [2](#)

bdmcmc, [3](#), [6](#), [8](#), [10](#), [13–15](#), [17](#)

bdmcmc.high, [4](#), [5](#), [8](#)

bdmcmc.low, [4](#), [6](#), [7](#), [11](#)

I.g, [9](#)

phat, [10](#)

plotConvergency, [11](#)

plotLinks, [12](#)

prob.allg, [13](#), [17](#)

prob.g, [14](#)

sample.gwishart, [16](#)

select.g, [17](#)