# Package 'SBICgraph'

May 7, 2026

**Title** Structural Bayesian Information Criterion for Graphical Models

**Version** 1.0.0

**Description** This is the implementation of the novel structural Bayesian information criterion by Zhou, 2020 (under review).
In this method, the prior structure is modeled and incorporated into the Bayesian information criterion framework.
Additionally, we also provide the implementation of a two-step algorithm to generate the candidate model pool.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** glmnet, MASS, network

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Quang Nguyen [cre, aut] (ORCID:
<https://orcid.org/0000-0002-2072-3279>),
Jie Zhou [aut],
Anne Hoen [aut],
Jiang Gui [aut]

**Maintainer** Quang Nguyen <Quang.P.Nguyen.GR@dartmouth.edu>

**Repository** CRAN

**Date/Publication** 2021-03-02 19:10:09 UTC

## Contents

---

`addition` *Enrichment step for constructing the model pool*

---

## Description

This is the esnrichment step in the two-step algorithm to construct the model pool (internal use only)

## Usage

```
addition(data, lambda, P)
```

## Arguments

| | |
|---|---|
| `data` | An `n` by `p` matrix of observations |
| `lambda` | Vector of tuning parameter |
| `P` | Prior adjacency matrix |

## Value

A list of model objects

## Author(s)

Jie Zhou

---

`comparison` *Comparing the real and estimated adjacency matrix*

---

## Description

Comparing the two adjacency matrices for false discovery rate and positive selection rate. Used for model validation

## Usage

```
comparison(real, estimate)
```

### Arguments

| | |
|---|---|
| real | The real matrix p by p adjacency matrix likely from simulated data |
| estimate | The estimated matrix p by p adjacency matrix likely estimated using the SBIC procedure |

### Value

A list of the following evaluation metrics

| | |
|---|---|
| PSR | Positive Selection Rate |
| FDR | False Discovery rate |

### Author(s)

Jie Zhou

---

## deletion *Pruning step for constructing the model pool*

---

### Description

This is the pruning step in the two-step algorithm to construct the model pool (internal use only)

### Usage

```
deletion(data, lambda, P)
```

### Arguments

| | |
|---|---|
| data | An n by p matrix of observations |
| lambda | Vector of tuning parameter |
| P | Prior adjacency matrix |

### Value

A list of model objects

### Author(s)

Jie Zhou

`mle` *Estimate the precision matrix for multivariate normal distribution with given adjacency matrix using maximum likelihood*

## Description

This function find the maximum likelihood estimate of the precision matrix with given adjacency matrix for multivariate normal distribution.

## Usage

```
mle(data, priori)
```

## Arguments

| | |
|---|---|
| `data` | An `n` by `p` dataframe representing the observations |
| `priori` | A `p` by `p` matrix representing the given adjacency matrix |

## Details

The methods are based on the relationship between precision matrix of the multivariate normal distribution and regression coefficients.

## Value

Returns a `p` by `p` matrix estimate of the precision matrix

## Author(s)

Jie Zhou

## Examples

```
set.seed(1)
d=simulate(n=100,p=200, m1=100, m2=30)
data=d$data
priori=d$realnetwork
precision=mle(data=data,priori=priori)
```

modelset *Construct model pool using the two-step algorithm*

## Description

For a given prior graph, the two-step algorithm, including edge enrichment and pruning, is used to construct the model pool

## Usage

```
modelset(data, lambda, P)
```

## Arguments

| | |
|---|---|
| data | A n by p data frame of observations |
| lambda | Tuning parameter vector |
| P | Prior adjacency matrix |

## Value

A list including all the candidate models in the model pool. Each model is represented by a p by p adjacency matrix

## Author(s)

Jie Zhou

## Examples

```
set.seed(1)
d=simulate(n=100, p=100, m1 = 100, m2 = 30)
data=d$data
P=d$priornetwork
lambda=exp(seq(-5,5,length=100))
candidates=modelset(data=data,lambda=lambda, P=P)
```

`sbic` *Structural Bayesian information criterion for multivariate normal data with a given graph structure*

## Description

This function estimates the novel structural Bayesian information criterion given the data and a given graph structure

## Usage

```
sbic(data, theta, prob, P)
```

## Arguments

| | |
|---|---|
| `data` | A `n` by `p` dataframe representing observations |
| `theta` | The `p` by `p` matrix representing the given graph structure |
| `prob` | The expected error rate |
| `P` | The prior adjacency matrix |

## Value

The value of sbic with given temperature parameter and prior adjacency matrix

## Author(s)

Jie Zhou

## Examples

```
set.seed(1)
d=simulate(n=100, p=100, m1 = 100, m2 = 30)
data=d$data
P=d$priornetwork
theta=d$realnetwork
prob=0.15
index=sbic(data=data, theta=theta, prob=prob, P=P)
```

sggm *Model selection of Gaussian graphical model based on SBIC*

## Description

Select the model based on the SBIC criterion and the two-step algorithm

## Usage

```
sggm(data, lambda, M, prob)
```

## Arguments

| | |
|---|---|
| data | An n by p dataframe representing the observations |
| lambda | A vector of tuning parameters used to build the model pool |
| M | The prior adjacency matrix |
| prob | The mean error rate |

## Value

A list of objects containing:

| | |
|---|---|
| networkhat | The final selected adjacency matrix |
| candidates | The model pool |

## Author(s)

Jie Zhou

## Examples

```
set.seed(1)
m1 = 100
m2 = 30
p = 100
n = 100
d=simulate(n=n,p=p, m1 = m1, m2 = m2) # simulate fake data
lambda=exp(seq(-5,5,length=100)) # tuning parameter
data=d$data # data from the simulation
M=d$priornetwork # prior network from simulation
# calculating the error rate
r1=m2/m1
r2=m2/(p*(p-1)/2-m1)
r=(r1+r2)/2
# apply sggm
result=sggm(data=data, lambda=lambda, M=M, prob=r)
# compare the final network and the true network
result$networkhat
```

```
d$realnetwork
```

---

`simulate` *Randomly generate a adjacency matrix based on which to simulate data*

---

## Description

According to a given edge density, first generate the adjacency matrix P of a graph. Based on P, the simulated multivariate normal data is generated with mean zero and a specified given precision matrix

## Usage

```
simulate(n, p, m1, m2)
```

## Arguments

| | |
|---|---|
| `n` | Sample size |
| `p` | The number of vertices in graph or the number of variables |
| `m1` | The number of edges in the true graph |
| `m2` | The number of elements in adjacency matrix that stay in different states, i.e., 0 or 1, in true and prior graphs |

## Value

A list including the simulated data, real adjacency matrix and a prior adjacency matrix

| | |
|---|---|
| `data` | simulated data |
| `realnetwork` | real adjacency matrix |
| `priornetowrk` | prior adjacency matrix |

## Author(s)

Jie Zhou

## Examples

```
set.seed(1)
d=simulate(n=100,p=200, m1=100, m2=30)
d$data
d$realnetwork
d$priornetwork
```

# Index