Package ‘easybgm’

December 8, 2025

Type Package

Title Extracting and Visualizing Bayesian Graphical Models

Version 0.3.1

Maintainer Karoline Huth <k.huth@uva.nl>

Description

Fit and visualize the results of a Bayesian analysis of networks commonly found in psychology.
The package supports fitting cross-
sectional network models fitted using the packages 'BDgraph', 'bgms' and 'BGGM',
as well as network comparison fitted using the 'bgms' and 'BBGM'.
The package provides the parameter estimates, posterior inclusion probabilities, inclusion Bayes factor, and the
posterior density of the parameters. In addition, for 'BDgraph' and 'bgms' it allows to assess the posterior
structure space. Furthermore, the package comes with an extensive suite for visualizing results.

License GPL (>= 2)

URL https://github.com/KarolineHuth/easybgm

BugReports https://github.com/KarolineHuth/easybgm/issues

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports BDgraph, BGGM, bgms (>= 0.1.4), dplyr, ggplot2, HDInterval,
igraph, qgraph, coda

Suggests testthat (>= 3.2.3), vdiffr

Config/testthat/edition 3

NeedsCompilation no

Author Karoline Huth [aut, cre] (ORCID:
<https://orcid.org/0000-0002-0662-1591>),
Sara Keetelaar [ctb],
Nikola Sekulovski [ctb],
Gali Geller [ctb]

Repository CRAN

Date/Publication 2025-12-08 12:10:07 UTC

Contents

bgm_extract — *Extract the results of a Bayesian analysis of networks*

Description

Extract the results of a Bayesian analysis of networks

Usage

```
bgm_extract(fit, ...)
```

Arguments

`fit`	Fit object with a particular class that will dispatch to the respective package functions
`...`	Additional arguments to be passed onto the respective fitting functions

bgm_fit *Fit a Bayesian analysis of networks*

Description

Fit a Bayesian analysis of networks

Usage

```
bgm_fit(fit, ...)
```

Arguments

`fit`	Object with a particular class that will dispatch to the respective package functions
`...`	Additional arguments to be passed onto the respective fitting functions

centrality *Plot strength centralities and 95% highest density interval*

Description

Visualize the strength centralities and their uncertainties. The centrality estimate can be obtained for each sample of the posterior distribution of the association parameters to obtain an estimate of the uncertainty of the strength centrality estimate.

Usage

```
plot_centrality(output, group_names = NULL, ...)
```

Arguments

`output`	One output object or a list of several output objects from the easybgm function. Supports also objects from the bgm function of the `bgms` package.
`group_names`	Specifying the group names, when providing a list of output objects. Needs to be a vector in the same length as the provided number of objects.
`...`	Additional arguments passed onto `ggplot2`

Value

Returns a plot

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "ordinal",
                iter = 100,  # for demonstration only (> 5e4 recommended)
                edge_selection = TRUE, save = TRUE,
                centrality = TRUE)

plot_centrality(fit)
```

---

clusterBayesfactor *Calculate Clustering Bayes Factors for when using the Stochastic Block Model as an edge prior*

Description

This function calculates Bayes factors to evaluate evidence in favor of clustering for models fitted with the `bgms` package (i.e., with arguments `package = "bgms"` and `edge_prior = "Stochastic-Block"` within the `easybgm` function). The function supports two types of Bayes factors: Bayes factors between two point hypothesized number of clusters (`b1` and `b2`), and Bayes factor of the hypothesis of clustering (i.e., the complement hypothesis) against the hypothesis of no clustering (i.e., the null, which simply means that the network exibits one global cluster).

Usage

```
clusterBayesfactor(fit, type = "complement", b1 = NULL, b2 = NULL)
```

Arguments

`fit`	A fitted object of class `easybgm` or `bgms` containing the clustering results.
`type`	A character string specifying the type of Bayes factor to calculate. Options are `"point"` or `"complement"`. Defaults to `"complement"`.
`b1`	Indicates the number of clusters according to the first point hypothesis, required for `type = "point"`.
`b2`	Indicates the number of clusters according to the second point hypothesis, required for `type = "point"`.

Value

A numeric value representing the Bayes factor. When type is `"point"`, the Bayes factor represents evidence in favor of `b1` clusters against `b2` clusters. When type is `"complement"`, the Bayes factor represents evidence in favor of clustering (i.e., more than one cluster) against no clustering.

complexity_probs *Plot posterior complexity probabilities*

Description

Plots the posterior complexity probabilities of all visited structures, where complexity comprises the network density.

Usage

```
plot_complexity_probabilities(output, ...)
```

Arguments

`output`	Output object from the easybgm function. Supports also objects from the bgm function of the `bgms` package.
`...`	Additional arguments passed onto `ggplot2`

Value

Returns a plot

Examples

```

library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "ordinal", save = TRUE, edge_selection = TRUE,
                iter = 100  # for demonstration only (> 5e4 recommended)
                )

plot_complexity_probabilities(fit)
```

easybgm *Fit a Bayesian analysis of networks*

Description

Easy estimation of a Bayesian analysis of networks to obtain conditional (in)dependence relations between variables in a network.

Usage

```
easybgm(
  data,
  type,
  package = NULL,
  not_cont = NULL,
  iter = 10000,
  save = FALSE,
  centrality = FALSE,
  progress = TRUE,
  posterior_method = "model-averaged",
  ...
)
```

Arguments

`data` An n x p matrix or dataframe containing the variables for n independent observations on p variables.

`type` What is the data type? Options: continuous, mixed, ordinal, binary, or blume-capel.

`package` The R-package that should be used for fitting the network model; supports BGGM, BDgraph, and bgms. Optional argument; default values are specified depending on the datatype.

`not_cont` If data-type is mixed, a vector of length p, specifying the not-continuous variables (1 = not continuous, 0 = continuous).

`iter` number of iterations for the sampler.

`save` Logical. Should the posterior samples be obtained (default = FALSE)?

`centrality` Logical. Should the centrality measures be extracted (default = FALSE)? Note, that it will significantly increase the computation time.

`progress` Logical. Should a progress bar be shown (default = TRUE)?

`posterior_method`

Determines how the posterior samples of the edge weight parameters are obtained for models fit with BDgraph. The argument can be either MAP for the maximum-a-posteriori or model-averaged. If MAP, samples are obtained for the edge weights only for the most likely structure. If model-averaged, samples are obtained for all plausible structures weighted by their posterior probability. Default is model-averaged.

`...` Additional arguments that are handed to the fitting functions of the packages, e.g., informed prior specifications.

Details

Users may oftentimes wish to deviate from the default, usually uninformative, prior specifications of the packages to informed priors. This can be done by simply adding additional arguments to the `easybgm` function. Depending on the package that is running the underlying network estimation, researcher can specify different prior arguments. We give an overview of the prior arguments per package below.

bgms:

- `interaction_scale` the scale of the Cauchy distribution that is used as a prior for the pairwise interaction parameters. The default is 2.5.
- `edge_prior` prior on the graph structure, which can be either "Bernoulli", "Beta-Bernoulli" or "Stochastic Block". The default is "Bernoulli".
- `inclusion_probability` prior edge inclusion probability for the "Bernoulli" distribution. The default is 0.5.
- `beta_bernoulli_alpha` and `beta_bernoulli_beta` the parameters of the "Beta-Bernoulli" or "Stochastic Block" priors. The default is 1 for both.
- `beta_bernoulli_alpha_between` and `beta_bernoulli_beta_between` the parameters of the "Stochastic Block" prior for edges between blocks. This is currently only available in a developer version of bgms and will be available in version 0.1.6.2 or higher.
- `dirichlet_alpha` The shape of the Dirichlet prior on the node-to-block allocation parameters for the Stochastic Block prior on the graph structure.
- `threshold_alpha` and `threshold_beta` the parameters of the beta-prime distribution for the threshold parameters. The defaults are both set to 1.
- `variable_type` What kind of variables are there in `x`? Can be a single character string specifying the variable type of all `p` variables at once or a vector of character strings of length `p` specifying the type for each variable in `x` separately. Currently, bgm supports `ordinal'' and` blume-capel”. Binary variables are automatically treated as “ordinal”. Defaults to `variable_type = "ordinal"`.

BDgraph:

- `df.prior` prior on the parameters (i.e., inverse covariance matrix), degrees of freedom of the prior G-Wishart distribution. The default is set to 3.
- `g.prior` prior probability of edge inclusion. This can be either a scalar, if it is the same for all edges, or a matrix, if it should be different among the edges. The default is set to 0.5.

BGGM:

- `prior_sd` the standard deviation of the prior distribution of the interaction parameters, approximately the scale of a beta distribution. The default is 0.25.

We would always encourage researcher to conduct prior robustness checks.

Value

The returned object of `easybgm` contains several elements:

- `parameters` A p x p matrix containing partial associations.
- `inc_probs` A p x p matrix containing the posterior inclusion probabilities.
- `BF` A p x p matrix containing the posterior inclusion Bayes factors.

- structure Adjacency matrix of the median probability model (i.e., edges with a posterior probability larger 0.5).

In addition, for BDgraph and bgms, the function returns:

- structure_probabilities A vector containing the posterior probabilities of all visited structures, between 0 and 1.
- graph_weights A vector containing the number of times a particular structure was visited.
- sample_graphs A vector containing the indexes of a particular structure.

  For the bgms package, when edge_prior = "Stochastic-Block", the function will also return an object sbm which contains:

  - posterior_num_blocks A data frame with the estimated posterior probability of the possible number of clusters.
  - posterior_mean_allocations The posterior mean of the cluster assignments of the nodes.
  - posterior_mode_allocations The posterior mode of the cluster assignments of the nodes.
  - posterior_mean_coclustering_matrix A p x p matrix containing the estimated pairwise proportions of cluster occurrence of every variable. This matrix can be plotted to visually inspect the estimated number of clusters and visually inspect nodes that tend to switch clusters.

If using version 0.1.6.1 or higher of the bgms package, the function also returns the the Gelman-Rubin convergence statistic for each edge weight parameter. As well as the 95% Monte Carlo confidence interval for the inclusion Bayes factor.

For all packages, when setting save = TRUE and centrality = TRUE, the function will return the following objects respectively:

- samples_posterior A k x iter matrix containing the posterior samples for each parameter (i.e., k = (p/(p-1))/2) at each iteration (i.e., iter) of the sampler.
- centrality A p x iter matrix containing the centrality of a node at each iteration of the sampler.

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)

# Fitting the Wenchuan PTSD data

fit <- easybgm(data, type = "continuous",
                iter = 100 # for demonstration only (> 5e4 recommended)
                )

summary(fit)
```

```
# To extract the posterior parameter distribution
# and centrality measures

fit <- easybgm(data, type = "continuous",
                iter = 100,
                centrality = TRUE, save = TRUE)
```

easybgm_compare *Compare networks across groups using Bayesian inference*

Description

Easy comparison of networks using Bayesian inference to extract differences in conditional (in)dependence across groups.

Usage

```
easybgm_compare(
  data,
  type,
  package = NULL,
  not_cont = NULL,
  group_indicator = NULL,
  iter = 10000,
  save = TRUE,
  progress = TRUE,
  ...
)
```

Arguments

`data` A list with two n x p matrices or dataframes containing the variables for n independent observations on p variables for two groups. Note that the variables need to be the same in the two different dataframes. Alternatively, when "bgms" version > 0.1.6 is installed, 'data' can also be a matrix of binary and ordinal responses from all groups. If this is the case, the 'group_indicator' argument also needs to be specified.

`type` What is the data type? Options: continuous, mixed, ordinal, binary, or blume-capel.

`package` The R-package that should be used for fitting the network model; supports BGGM and bgms. Optional argument; default values are specified depending on the datatype.

`not_cont` If data-type is mixed, a vector of length p, specifying the not-continuous variables (1 = not continuous, 0 = continuous).

`group_indicator`
Optional integer vector of group memberships for the rows of the dataframe (multi-group comparison), when data is a matrix instead of a list of two dataframes.

`iter` number of iterations for the sampler. Default is 1e4.

`save` Logical. Should the posterior samples be obtained (default = TRUE)?

`progress` Logical. Should a progress bar be shown (default = TRUE)?

`...` Additional arguments that are handed to the fitting functions of the packages, e.g., informed prior specifications.

Details

Users may oftentimes wish to deviate from the default, usually uninformative, prior specifications of the packages to informed priors. This can be done by simply adding additional arguments to the `easybgm` function. Depending on the package that is running the underlying network estimation, researcher can specify different prior arguments. Please consult the original packages "bgms" and "BGGM" for the specific informed prior options.

We always encourage researcher to conduct prior robustness checks.

Value

The returned object of `easybgm` contains several elements:

- `parameters` A p x p matrix containing difference across partial associations.
- `inc_probs` A p x p matrix containing the posterior inclusion probabilities of subgroup differences.
- `inc_BF` A p x p matrix containing the posterior inclusion Bayes factors of subgroup differences.
- `structure` Adjacency matrix of the median probability model (i.e., edges with a posterior probability larger 0.5).

In addition, for `bgms`, the function returns:

- `structure_probabilities` A vector containing the posterior probabilities of all visited structures, between 0 and 1.
- `graph_weights` A vector containing the number of times a particular structure was visited.
- `sample_graph` A vector containing the indexes of a particular structure.
- `convergence_parameter` A vector containing the R-hat (Gelman–Rubin) statistic for the difference parameter measuring how well MCMC chains have converged to the same target distribution.

For both packages, when setting `save = TRUE`, the function will also return the following object:

- `samples_posterior` A k x iter matrix containing the posterior samples of parameter differences (i.e., k = (p/(p-1))/2) at each iteration (i.e., iter) of the sampler.

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(ADHD)

group1 <- data[1:10, 1:3]
group2 <- data[11:20, 1:3]

# Fitting the Wenchuan PTSD data

fit <- easybgm_compare(list(group1, group2),
                type = "binary", save = TRUE,
                iter = 50 # for demonstration only (> 5e4 recommended)
                )

summary(fit)

# For multigroup estimation
fit_multi <- easybgm_compare(data[1:200, 1:5],
                group_indicator = rep(c(1, 2, 3, 4), each = 50),
                type = "binary", save = TRUE,
                iter = 100 # for demonstration only (> 5e4 recommended)
                )

summary(fit_multi)
```

---

edgeevidence *Edge evidence plot*

Description

The edge evidence plot colors edges according to their hypothesis testing results: blue for included, light blue for weakly included, gray for inconclusive, light yellow for weakly excluded, and yellow for excluded. This plot can be used to visualize the hypothesis testing results whether edge presence or absence. The edge evidence plot can aid researchers in deciding which edges provide robust inferential conclusions

Usage

```
plot_edgeevidence(
  output,
  evidence_thresh = 10,
  split = FALSE,
  show = "all",
  ...
)
```

Arguments

`output` Output object from the easybgm function. Supports also objects from the bgm function of the `bgms` package.

`evidence_thresh` Bayes Factor which will be considered sufficient evidence for in-/exclusion, default is 10. Note that this parameter defines when edges provide sufficient evidence, thus when the edge color will turn saturated blue or yellow. All edges with a BF between 3 and the evidence threshold will receive a light saturated edge color.

`split` if TRUE, plot is split in included and excluded edges. Note that by default separate plots are shown and appear after each other in the plot window. To show the plots side-by-side specify par(mfrow = c(1, 2)).

`show` specifies which edges should be shown, indicated by "all", "included", "inconclusive", "excluded".

`...` Additional arguments passed onto `qgraph`.

Value

Returns a plot

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "continuous",
                iter = 100  # for demonstration only (> 5e4 recommended)
                )

plot_edgeevidence(fit)

oldpar <- par(mfrow = c(1,1))

par(mfrow = c(1, 2))
plot_edgeevidence(fit, split = TRUE)

#' par(mfrow = c(1, 3))
plot_edgeevidence(fit, show = "included")
plot_edgeevidence(fit, show = "inconclusive")
plot_edgeevidence(fit, show = "excluded")

par(oldpar)
```

HDI *Plot of interaction parameters and their 95% highest density intervals*

Description

Plots the 95% highest density interval of the posterior distribution of the parameter estimates. The plot can be used to visualize the uncertainty of the partial association estimates. The x-axis indicates the strength of the partial association. The y-axis indicates the edge between nodes $i$ and $j$. The farther the posterior estimates (i.e., the points in the plot) are from zero, the stronger the partial association of the edge. The wider the highest density intervals (i.e., the error bar around the point), the less certain we are about the strength of the association.

Usage

```
plot_parameterHDI(output, ...)
```

Arguments

`output`	Output object from the easybgm function. Supports also objects from the bgm function of the `bgms` package.
`...`	Additional arguments passed onto `ggplot2`

Value

Returns a plot

Examples

```

library(easybgm)
library(bgms)


data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "ordinal",
              iter = 100,  # for demonstration only (> 5e4 recommended)
              edge_selection = TRUE, save = TRUE)
plot_parameterHDI(fit)
```

`network` *Network plot*

Description

The network plot visualizes the strength of interactions between two nodes, the partial associations. Solely edges with a posterior inclusion probability larger than the `exc_prob` argument (default = 0.5) are shown. Edge thickness and saturation represent the strength of the association; the thicker the edge, the stronger the association. Red edges indicate negative relations and blue edges indicate positive associations.

Usage

```
plot_network(output, exc_prob = 0.5, evidence_thresh = 10, dashed = FALSE, ...)
```

Arguments

`output` Output object from the easybgm function. Supports also objects from the bgm function of the `bgms` package.

`exc_prob` The threshold for excluding edges. All edges with a lower inclusion probability will not be shown. The default is set to 0.5 in line with the median probability plot.

`evidence_thresh` If dashed = TRUE, users can specify the threshold for sufficient evidence for inclusion. All edges with evidence lower than `evidence_tresh` are dashed.

`dashed` A binary parameter indicating whether edges with inconclusive evidence should be dashed. Default is FALSE

`...` Additional arguments passed onto `qgraph`.

Value

Returns a plot

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "continuous",
                iter = 100  # for demonstration only (> 5e4 recommended)
                )

plot_network(fit)

# Shows all edges with an inclusion probability larger than 0.1
```

```
plot_network(fit, exc_prob = 0.1)

# Indicate which edges have insufficient evidence for inclusion through a dashed line
plot_network(fit, dashed = TRUE, evidence_thresh = 10)
```

---

print.easybgm *Print method for* easybgm *objects*

Description

Used to print easybgm results. The nicest overview is created by first feeding it to `summary()`

Usage

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

Arguments

x	easybgm object
...	unused argument

Value

Prints the output of a Bayesian cross-sectional network model fitted with 'easybgm'

---

print.easybgm_compare *Print method for* easybgm_compare *objects*

Description

Used to print easybgm results. The nicest overview is created by first feeding it to `summary()`

Usage

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

Arguments

x	easybgm_compare object
...	unused argument

Value

Prints the output of a Bayesian cross-sectional network comparison fitted with 'easybgm'

prior_sensitivity *Plot sensitivity to edge inclusion prior setting*

Description

For a given list of easybgm outputs with different prior edge inclusion probabilities, the function plots the percentage of edges that are included, excluded, and inconclusive.

Usage

```
plot_prior_sensitivity(output, ...)
```

Arguments

`output`	A list of easybgm outputs with different prior edge inclusion probabilities
`...`	Additional arguments passed onto ggplot2.

Details

Prior sensitivity plot

Value

Returns a plot

Examples

```

library(easybgm)
library(bgms)

#data <- na.omit(Wenchuan)
#fit1 <- easybgm(data, type = "ordinal",
#              iter = 100  # for demonstration only (> 5e4 recommended),
#               inclusion_probability = .1
#              )
#fit2 <- easybgm(data, type = "ordinal",
#                 iter = 100,
#                 inclusion_probability = .5
#            )
#fit3 <- easybgm(data, type = "ordinal",
#               iter = 100, inclusion_probability = .9)

#plot_prior_sensitivity(list(fit1, fit2, fit3))
```

structure *Structure plot*

Description

The plot shows the resulting graph structure, i.e. all edges with some evidence of inclusion (i.e., inclusion Bayes factor greater than 1).

Usage

```
plot_structure(output, ...)
```

Arguments

`output` Output object from the easybgm function. Supports also objects from the bgm function of the bgms package.

`...` Additional arguments passed onto `qgraph`

Value

Returns a plot

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "ordinal",
                iter = 100  # for demonstration only (> 5e4 recommended)
               )

plot_structure(fit)
```

structure_probs *Plot Posterior Structure Probabilities*

Description

Plots the posterior structure probabilities of all visited structures, sorted from the most to the least probable.

Usage

```
plot_structure_probabilities(output, as_BF = FALSE, ...)
```

Arguments

`output`	Output object from the easybgm function. Supports also objects from the bgm function of the `bgms` package.
`as_BF`	If TRUE plots the y-axis as Bayes factors instead of posterior structure probability. Default is FALSE.
`...`	Additional arguments passed onto `ggplot2`

Value

Returns a plot

Examples

```

library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "ordinal", save = TRUE, edge_selection = TRUE,
                iter = 100  # for demonstration only (> 5e4 recommended)
                )

plot_structure_probabilities(fit)
```

`summary.easybgm` *Summary method for* `easybgm` *objects*

Description

Used to create a object of easybgm results and in turn print it

Usage

```
## S3 method for class 'easybgm'
summary(object, evidence_thresh = 10, BF_uncertainty = FALSE, ...)
```

Arguments

`object`	easybgm object
`evidence_thresh`	Bayes Factor which will be considered sufficient evidence for in-/exclusion, default is 10.
`BF_uncertainty`	Whether the MC uncertainty estimates for the Bayes factors should be included in the output (only for bgms package)
`...`	unused argument

Value

Creates and prints the output of a Bayesian cross-sectional network analysis. The summary output has four parts. The first part lists the package used, the number of variables, and the data type. The second part is a matrix of edge-specific information. Each edge is listed in a row. This row contains the posterior parameter estimate, the posterior inclusion probability, the inclusion Bayes factor, and the categorization of the edge. The category encodes whether an edge is included, excluded, or inconclusive based on the inclusion Bayes factor. Users can set the threshold for the Bayes factor classification with the evidence threshold. By default, the threshold is set to 10. The third part of the summary provides aggregated edge information. It lists the number of included, excluded, and inconclusive edges in the network, as well as the number of possible edges. This gives the user a quick overview of the robustness and density of the network. The higher the number of conclusive edges (i.e., classified as either included or excluded), the more robust the network. Conversely, if the network has a high percentage of inconclusive edges, the network is not robust. Researchers should refrain from making strong inferential conclusions. The final output section is a description of the structure uncertainty. It shows the number of structures visited, the number of possible structures, and the highest posterior structure probability. This last section can only be obtained for networks fitted with 'BDgraph' and 'bgms'.

`summary.easybgm_compare`

Summary method for `easybgm_compare` *objects*

Description

Used to create a object of easybgm results and in turn print it

Usage

```
## S3 method for class 'easybgm_compare'
summary(object, evidence_thresh = 10, ...)
```

Arguments

`object`	easybgm_compare object
`evidence_thresh`	Bayes Factor which will be considered sufficient evidence for in-/exclusion, default is 10.
`...`	unused argument

Value

Creates and prints the output of a Bayesian cross-sectional network analysis. The summary output has four parts. The first part lists the package used, the number of variables, and the data type. The second part is a matrix of edge-specific information. Each edge is listed in a row. This row contains the posterior parameter estimate, the posterior inclusion probability, the inclusion Bayes factor, and the categorization of the edge. The category encodes whether an edge is included, excluded, or inconclusive based on the inclusion Bayes factor. Users can set the threshold for the Bayes factor classification with the evidence threshold. By default, the threshold is set to 10. The third part of the summary provides aggregated edge information. It lists the number of included, excluded, and inconclusive edges in the network, as well as the number of possible edges. This gives the user a quick overview of the robustness and density of the network. The higher the number of conclusive edges (i.e., classified as either included or excluded), the more robust the network. Conversely, if the network has a high percentage of inconclusive edges, the network is not robust. Researchers should refrain from making strong inferential conclusions. The final output section is a description of the structure uncertainty. It shows the number of structures visited, the number of possible structures, and the highest posterior structure probability. This last section can only be obtained for networks fitted with 'BDgraph' and 'bgms'.

Index