Package 'easybgm'

September 30, 2026

Type Package

Title Extracting and Visualizing Bayesian Graphical Models

Version 0.5.1

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 cross-
sectional network models for ordinal, binary, continuous, and mixed data,
fitted using the packages 'bgms' (default), 'BDgraph', and 'BGGM',
as well as network comparison tests fitted using the packages 'bgms' and 'BGGM'.
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

Depends R (>= 4.1.0)

Imports BDgraph, BGGM, bgms (>= 0.1.6.3), dplyr, ggplot2, HDInterval, qgraph, coda

Suggests testthat (>= 3.2.3)

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

RoxygenNote 7.3.3

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 2026-09-30 21:50:02 UTC

Contents

bgm_extract 2
bgm_fit 3
centrality 3
clusterBayesfactor 4
complexity_probs 5
easybgm 6
easybgm_compare 11
edgeevidence 15
HDI 17
network 18
print.easybgm 19
print.easybgm_compare 20
prior_sensitivity 20
structure 21
structure_probs 22
summary.easybgm 23
summary.easybgm_compare 24

Index **26**

`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

```
## Not run:
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data[1:50, 1:5], type = "ordinal",
               iter = 100,    # for demonstration only
               edge_selection = TRUE, save = TRUE,
               centrality = TRUE)

plot_centrality(fit)

## End(Not run)
```

`clusterBayesfactor` *Test whether a network splits into clusters*

Description

For a network fitted with the Stochastic Block Model (SBM) edge prior, this gives a Bayes factor for the number of clusters. With `type = "complement"` it weighs more than one cluster against exactly one; with `type = "point"` it weighs `b1` clusters against `b2`. Values above 1 favour the first of the two; take the reciprocal to read the evidence the other way round.

Usage

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

Arguments

`fit`	A fit of class `easybgm` or `bgms`, fitted with `edge_prior = bgms::sbm_prior()`. Raw draws are used when the fit carries them, otherwise the stored summary, which must be at full precision.
`type`	Either `"complement"`, the default, or `"point"`.
`b1, b2`	Whole numbers between 1 and the number of variables, required when `type = "point"`.

Details

The count is the number of clusters the model has available, which can be larger than the number that actually hold variables, since a cluster may come out empty. For the memberships themselves, see `bgms::extract_sbm()`. Evidence for clustering concerns the network's edge structure and is not by itself evidence of multidimensionality.

Value

A single unrounded Bayes factor. NA with a warning if neither point hypothesis appears in the posterior. A result of 0 or Inf means the sampler never visited one of the two, so run more iterations rather than reading it as decisive.

`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

```
## Not run:

library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)[1:50, 1:5]
fit <- easybgm(data, type = "ordinal", save = TRUE, edge_selection = TRUE,
               iter = 100   # for demonstration only
               )

plot_complexity_probabilities(fit)

## End(Not run)
```

easybgm *Bayesian analysis of networks*

Description

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

Usage

```
easybgm(
  data,
  type,
  package = NULL,
  save = FALSE,
  centrality = FALSE,
  iter = 1000,
  progress = TRUE,
  baseline_category = NULL,
  not_cont = NULL,
  ...
)
```

Arguments

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

`type` Specifies the type of data. There are two ways to use this argument:

1. A single string, applied to every variable:

- `"continuous"`: For continuous (Gaussian) data. Estimates a Gaussian Graphical Model (GGM).
- `"ordinal"`: For ordinal (Likert-type) data. Estimates an Ordinal Markov Random Field (OMRF).
- `"binary"`: For binary (0/1) data. Estimates an Ising model.
- `"blume-capel"`: For Blume-Capel ordinal data. Requires a reference category via the `baseline_category` argument.
- `"mixed"`: For data with both continuous and discrete variables. Requires the `not_cont` argument to indicate which variables are not continuous.

2. A character vector of length p (per-variable specification): Each element gives the type of the corresponding column of `data`, can be any of: `"ordinal"`, `"continuous"`, `"blume-capel"`, and `"binary"`. For example,`type = c("ordinal", "ordinal", "continuous")` specifies that the first two columns are ordinal and the third is continuous. Per-variable vectors are fitted by `bgms` and require `bgms` version 0.2.0.0 or later.

Data types and package support in the Details section below specifies which package will be used for which variable type.

`package` The R-package used for fitting the network model. Optional. Supported options:

- `"bgms"`: Fits ordinal, binary, and blume-capel data, and – from `bgms` version 0.2.0.0 onwards – continuous data and per-variable type vectors as well.
- `"BDgraph"`: Fits continuous data (a GGM), mixed and ordinal data (a GCGM), and binary data (a discrete graphical model). For continuous data, missing values are not allowed; use `na.omit()` on the data first.
- `"BGGM"`: Fits continuous, mixed, ordinal, and binary data.

If `package` is not specified, `bgms` is used for all data types from `bgms` version 0.2.0.0 onwards. With `bgms` 0.1.6.3, `bgms` is used for binary, ordinal, and blume-capel data and `BGGM` for continuous and mixed data.

`save` Logical. Should the posterior samples be obtained (default = `FALSE`)? If `TRUE`, the output includes a `samples_posterior` matrix with the posterior samples for each edge weight parameter. Setting `centrality = TRUE` automatically sets `save = TRUE`.

`centrality` Logical. Should the strength centrality measures be extracted (default = `FALSE`)? Automatically sets `save = TRUE`.

`iter` Number of iterations for the sampler. The default depends on the package:

- `bgms`: 1e3 (1,000 iterations)
- `BDgraph`: 1e4 (10,000 iterations)
- `BGGM`: 1e4 (10,000 iterations)

The recommended number of iterations depends on the data, model complexity, and desired precision. Check the convergence diagnostics in the output to determine if more iterations are needed.

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

`baseline_category`
Integer or vector, required if at least one variable is of type `"blume-capel"`. Baseline category used in Blume–Capel variables. Can be a single integer (applied to all) or a vector of length `p`.

`not_cont` A binary vector of length `p`, required whenever `type = "mixed"`. Each element indicates whether the corresponding variable is not continuous (`1` = not continuous/ordinal, `0` = continuous). This parameter is not required when specifying a variable vector with `bgms`.

`...` Additional arguments passed to the fitting functions of the underlying packages (e.g., prior specifications). See the **Prior specification** section in Details for available prior options per package and the package help files for all other potential arguments.

Details

Data types and package support

`BGGM` and `BDgraph` fit continuous, mixed, ordinal and binary data; the Blume-Capel model and per-variable `type` vectors are fitted only by `bgms`. "Default" marks the package used when `package` is left unspecified. Because `bgms` 0.1.6.3 cannot fit continuous data, the `bgms` column is split by version:

Data type	bgms >= 0.2.0.0	bgms 0.1.6.3	BDgraph	BGGM
continuous	Yes (default)	No	Yes	Yes
ordinal	Yes (default)	Yes (default)	Yes	Yes
binary	Yes (default)	Yes (default)	Yes	Yes
blume-capel	Yes (default)	Yes (default)	No	No
mixed	Yes (default)	No	Yes	Yes
per-variable vector	Yes	No	No	No

Prior specification

Users may wish to deviate from the default (uninformative) prior specifications. This can be done by passing additional arguments via `...` to the fitting function of the chosen package. We give an overview of the available prior arguments per package below.

bgms (>= 0.2.0.0) uses prior-constructor objects from the `bgms` package. Pass them through `...`:

- `interaction_prior`: A parameter prior on pairwise interactions. Use `normal_prior(scale)` (default `normal_prior(scale = 1)`), `cauchy_prior(scale)`, or `beta_prime_prior(alpha, beta)`. For example, a cauchy prior with scale 1 would be specified with adding the argument `interaction_prior = cauchy_prior(1)` to the easybgm call.
- `threshold_prior`: A parameter prior on threshold (main effect) parameters. Use `beta_prime_prior(alpha, beta)` (default `beta_prime_prior(0.5, 0.5)`), `cauchy_prior(scale)`, or `normal_prior(scale)`. For example, a cauchy prior with scale 1 would be specified with adding the argument `threshold_prior = cauchy_prior(1)` to the easybgm call.
- `means_prior`: A prior on the means of continuous variables in mixed MRF models. Default `normal_prior(scale = 1)`, specify with `means_prior = normal_prior(scale = 1)`.
- `precision_scale_prior`: A prior on the diagonal entries of the precision matrix (GGM and mixed MRF). Use `exponential_prior(rate)` (default `exponential_prior(eta = 1)`) or `gamma_prior(shape, rate)`, for example, specified as `precision_scale_prior = exponential_prior(1)`
- `precision_graph_prior`: How the graph prior is applied to the precision matrix (GGM and mixed MRF). Either `"hierarchical"` (the default), which tracks the normalizing constant, or `"joint"`.
- `edge_prior`: An indicator prior on edge inclusion. Use `bernoulli_prior(inclusion_probability)` (default `bernoulli_prior(0.5)`; `inclusion_probability` can also be a symmetric $p \times p$ matrix of edge-specific probabilities), `beta_bernoulli_prior(alpha, beta)`, or `sbm_prior(alpha, beta, alpha_between, beta_between, dirichlet_alpha, lambda)` for the Stochastic Block Model prior. For example, a bernoulli prior with prior probabilit of 0.5 would be specified with adding the argument `edge_prior = bernoulli_prior(0.5)` to the easybgm call.

For backwards compatibility of *bgms* (< 0.2.0.0), the previous prior specifications are still accepted and translated into the relevant constructs. Check the previous bgms version for its prior arguments.

BDgraph:

- `df.prior`: Degrees of freedom of the prior G-Wishart distribution on the precision matrix. Default is 3.
- `g.prior`: Prior probability of edge inclusion. Can be a scalar (same for all edges) or a matrix (edge-specific). This can also be a symmetric pxp matrix of edge-specific inclusion probabilities. Default is 0.5.

BGGM:

- `prior_sd`: Standard deviation of the prior on interaction parameters (approximately the scale of a beta distribution). Default is 0.25.

We encourage researchers to conduct prior sensitivity checks.

Value

An object of class `easybgm` containing the following elements:

Always returned:

- `parameters`: A p x p matrix of posterior mean partial association estimates. Note that the scale differs between variable types. For continuous variables, the parameter represents partial correlations, for discrete variables partial associations. Edge weights from different variable types are therefore not directly comparable.
- `inc_probs`: A p x p matrix of posterior inclusion probabilities.
- `inc_BF`: A p x p matrix of posterior inclusion Bayes factors.
- `structure`: A p x p adjacency matrix of the median probability model (edges with posterior inclusion probability > 0.5).
- `model`: A string indicating the model type (e.g., `"continuous"`, `"ordinal"`, `"mixed"`).
- `thresholds`: Threshold/intercept parameters (bgms only). The format depends on the model type: a matrix for ordinal/binary models, `NULL` for continuous models, or a list for mixed models. For Blume-Capel variables the two columns are the linear and quadratic effects rather than category thresholds; they are named accordingly, and are also reported in `blume_capel_parameters` (see below). Where Blume-Capel and ordinal variables share one matrix, the column headers cannot describe both, so the per-row meaning is recorded in the `"variable_type"` attribute of the matrix.

Returned for bgms and BDgraph:

- `structure_probabilities`: Posterior probabilities of all visited graph structures (values between 0 and 1).
- `graph_weights`: Number of times each graph structure was visited.
- `sample_graphs`: Identifiers for each visited graph structure.

Returned for bgms only:

- `packagefit`: The underlying bgms fit object, on which bgms extractor functions can be called without refitting.
- `convergence_parameter`: The Gelman-Rubin (R-hat) convergence statistic for each edge weight parameter. Values close to 1 indicate good convergence.
- `MCSE_BF`: A matrix with the 95 percent Monte Carlo confidence interval for each inclusion Bayes factor, derived from the Monte Carlo standard error of the Rao-Blackwellized inclusion probability. Entries are `NA` where the interval is not defined, which happens when the posterior inclusion probability is numerically 0 or 1 and the Bayes factor is therefore 0 or infinite. This is routine for decisive edges rather than a sign of a problem.

Returned when edge_prior = "Stochastic-Block" (bgms only):

- sbm: A list containing Stochastic Block Model results, including posterior_num_blocks (posterior probabilities for each number of clusters), posterior_mean_allocations (posterior mean cluster assignments), posterior_mode_allocations (posterior mode cluster assignments), and posterior_mean_coclustering_matrix (a p x p matrix of pairwise coclustering proportions).

Interpretable parameter scales (bgms only):

In addition to the raw pairwise interaction parameters in parameters, the following transformations are provided when the model type supports them. They are NULL otherwise.

- partial_correlations: A matrix of posterior mean partial correlations. Available for continuous (GGM) models (full p x p matrix) and for the continuous block of mixed models. NULL for ordinal models.
- precision_matrix: The posterior mean precision (inverse covariance) matrix. Same availability as partial correlations.
- log_odds: A matrix of posterior mean log adjacent-category odds ratios. Available for ordinal/binary models (full p x p matrix) and for the discrete block of mixed models. NULL for continuous models.

Returned for Blume-Capel variables (bgms >= 0.2.0.0):

- blume_capel_parameters: A data frame with two rows per Blume-Capel variable, holding the posterior mean, posterior standard deviation, 95 percent credible interval and R-hat convergence statistic of its linear and quadratic effects, together with the baseline category used. Unlike the category thresholds of an ordinal variable, these are usually of substantive interest: for a variable with baseline category b, the threshold of category x is $\mu(x) = \alpha x + \beta(x - b)^2$, where α is the linear and β the quadratic effect. A negative quadratic effect indicates responses concentrated around the baseline category, a positive one a preference for the extreme categories. Baseline categories are reported on the scale of the input data. This element is NULL when no variable is Blume-Capel, and is not available with bgms 0.1.6.3.

Returned when save = TRUE:

- samples_posterior: A k x iter matrix of posterior samples for each edge weight parameter (k = p*(p-1)/2 edges).
- samples_blume_capel: Posterior samples of the Blume-Capel linear and quadratic effects, with one column per parameter, pooled across chains. Only present when at least one variable is Blume-Capel.

Returned when centrality = TRUE:

- centrality: An iter x p matrix of centrality values for each node at each iteration.

Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)[1:50, 1:3]

# --- Continuous data (fitted by bgms >= 0.2.0.0, otherwise by BGGM) ---
```

```
fit <- easybgm(data, type = "continuous",
               iter = 100,   # for demonstration only; increase for real analyses
               # arguments from bgms to reduce the runtime of the example
               warmup = 50, chain = 2, cores = 1, precision_graph_prior = "joint"
               )
summary(fit)

## Not run:
# --- Mixed data using per-variable type vector (requires bgms >= 0.2.0.0) ---
if (utils::packageVersion("bgms") >= "0.2.0.0") {
  dat3 <- data[, 1:3]
  fit_vec <- easybgm(dat3,
                     type = c("ordinal", "ordinal", "continuous"),
                     iter = 100)
}

# --- Extract posterior samples and centrality ---
fit_full <- easybgm(data, type = "continuous",
                    iter = 100,
                    centrality = TRUE, save = TRUE)

# --- Using BDgraph for continuous data ---
fit_bd <- easybgm(data, type = "continuous",
                  package = "BDgraph",
                  iter = 100)

# --- Using BGGM for continuous data ---
fit_bggm <- easybgm(data, type = "continuous",
                    package = "BGGM",
                    iter = 100)


## End(Not run)
```

`easybgm_compare` *Compare networks across groups using Bayesian inference*

Description

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

Usage

```
easybgm_compare(
  data,
  type,
  package = NULL,
  not_cont = NULL,
  group_indicator = NULL,
```

```
  iter = 1000,
  save = TRUE,
  progress = TRUE,
  ...
)
```

Arguments

data	The data can be provided in two formats: **1. A list of two dataframes** (two-group comparison): Each list element is an n x p matrix or dataframe for one group. The variables (columns) must be the same across both dataframes. This format is supported by both bgms and BGGM. **2. A single matrix or dataframe** (multi-group comparison): An n x p matrix containing responses from all groups combined. Requires the group_indicator argument to specify which rows belong to which group. This format supports two or more groups and is only available with the bgms package.
type	Specifies the data type. Can be used in two ways: **1. A single string**: • "continuous": For continuous data. Default package: BGGM. • "ordinal": For ordinal (Likert-type) data. Default package: bgms. • "binary": For binary (0/1) data. Default package: bgms. • "blume-capel": For Blume-Capel ordinal data. Requires baseline_category. Default package: bgms. • "mixed": For mixed data, requires not_cont. Default package: BGGM. **2. A character vector of length p** (per-variable specification): Each element specifies the type of the corresponding column. Valid values are "ordinal", "blume-capel", and "binary"; note that "continuous" is *not* supported for group comparison with bgms. For example: type = c("ordinal", "ordinal", "blume-capel"). Per-variable vectors are fitted by bgms and require bgms version 0.2.0.0 or later.
package	The R-package used for fitting the comparison model, either "bgms" or "BGGM". If not specified, bgms is used for ordinal, binary, and blume-capel data, and BGGM for continuous and mixed data.
not_cont	A binary vector of length p, required when type = "mixed". Each element indicates whether the corresponding variable is not continuous (1 = not continuous/ordinal, 0 = continuous).
group_indicator	An integer vector of length n specifying group membership for each row in data. Required when data is a single matrix/dataframe (multi-group comparison). Supports two or more groups (e.g., rep(c(1, 2, 3), each = 50) for three groups of 50 observations each). Only available with the bgms package.
iter	Number of iterations for the sampler. The default is 1e4. The recommended number of iterations depends on the data and model complexity. Check convergence diagnostics in the output.

save	Logical. Should the posterior samples be obtained (default = TRUE)? If TRUE, the output includes a samples_posterior matrix with posterior samples for each difference parameter.
progress	Logical. Should a progress bar be shown (default = TRUE)?
...	Additional arguments passed to the fitting functions of the underlying packages (e.g., prior specifications). Consult the documentation of bgms and BGGM for the specific options available.

Details

Data types and package support for group comparison

Data type	bgms	BGGM
ordinal	Yes (default)	No
binary	Yes (default)	Yes
blume-capel	Yes (default)	No
continuous	No	Yes (default)
mixed	No	Yes (default)

Prior specification

Users may wish to adjust priors via the ... argument. We summarize the bgms options here and refer to bgmCompare and explore for full details.

bgms (>= 0.2.0.0)

- interaction_prior: Prior on the baseline pairwise interactions. Use normal_prior(scale) (default normal_prior(scale = 1)), cauchy_prior(scale), or beta_prime_prior(alpha, beta). For example, a cauchy prior with scale 1 would be specified with adding the argument threshold_prior = cauchy_prior(1) to the easybgm call.
- threshold_prior: Prior on threshold parameters. Default beta_prime_prior(0.5, 0.5), for example, specified by adding threshold_prior = beta_prime_prior(0.5, 0.5) to the easybgm call.
- difference_prior: Indicator prior on group differences. Use bernoulli_prior(inclusion_probability) (default bernoulli_prior(0.5)) or beta_bernoulli_prior(alpha, beta). For example, a beta bernoulli prior with alpha 1 and beta 3 can be specified by adding difference_prior = beta_bernoulli_prior(1, 3) to the easybgm call.
- difference_family: The family of the prior on the magnitude of the pairwise differences, either "Normal" (the default) or "Cauchy". Versions of bgms before 0.2.0.0 had no such argument and always used a Cauchy, so comparison Bayes factors obtained with those versions correspond to difference_family = "Cauchy".
- difference_scale: Scale of the prior on the magnitude of pairwise differences, on the family set by difference_family. Default 1, for example, specified by adding the argument difference_scale = 1 to the easybgm call.
- difference_selection: Logical, whether to perform Bayesian selection on group differences. Default TRUE.

For backwards compatibility of *bgms* (< 0.2.0.0), the previous prior specifications are still accepted and translated into the relevant constructs. Check the previous bgms version for its prior arguments.

We always encourage researchers to conduct prior sensitivity checks.

Value

An object of class `easybgm_compare` with the following elements:

Always returned:

- `parameters`: A p x p matrix of posterior mean differences in partial associations across groups. For two groups fitted with bgms, the difference is group 2 minus group 1. Not returned for a bgms comparison of more than two groups, where a single difference per edge is not defined; see `pairwise_group_differences` below.
- `inc_probs`: A p x p matrix of posterior inclusion probabilities for group differences (i.e., the probability that an edge differs between groups).
- `inc_BF`: A p x p matrix of inclusion Bayes factors for group differences.
- `structure`: A p x p adjacency matrix of the median probability model for differences (edges with posterior inclusion probability > 0.5).
- `model`: A string indicating the data type used.

Returned for bgms only:

- `structure_probabilities`: Posterior probabilities of all visited graph structures.
- `graph_weights`: Number of times each structure was visited.
- `sample_graph`: Identifiers for each visited structure.
- `convergence_parameter`: The Gelman-Rubin (R-hat) convergence statistic for each difference parameter. Values close to 1 indicate good convergence.
- `pairwise_group_differences`: For a comparison fitted with `group_indicator`, a matrix of posterior mean pairwise group differences with one row per edge and one column per pair of groups (e.g., `"group2 - group1"`).
- `contrast_coefficients`: For a comparison of 3+ groups fitted with `group_indicator`, bgms does not estimate each edge directly per group. Instead it decomposes each edge into a shared baseline across groups plus a set of contrast coefficients that describe how each group deviates from that baseline. This element is that raw bgms output, unprocessed, labelled `"edge (diff1)"`, `"edge (diff2)"`, etc. For an interpretable "group A minus group B" difference for every pair of groups, use `pairwise_group_differences` instead.
- `packagefit`: The underlying bgms fit object, on which bgms extractor functions can be called without refitting.

Returned when save = TRUE:

- `samples_posterior`: A k x iter matrix of posterior samples for each difference parameter (k = p*(p-1)/2 edges).

Examples

```
## Not run:
library(easybgm)
library(bgms)

data <- na.omit(ADHD)

# --- Two-group comparison (list input) ---
group1 <- data[1:10, 1:3]
group2 <- data[11:20, 1:3]

fit <- easybgm_compare(list(group1, group2),
                type = "binary", save = TRUE,
                iter = 100   # for demonstration only
                )
summary(fit)

# --- Multi-group comparison (single dataframe + group_indicator) ---
fit_multi <- easybgm_compare(data[1:80, 1:5],
                group_indicator = rep(c(1, 2, 3, 4), each = 20),
                type = "binary", save = TRUE,
                iter = 100   # for demonstration only
                )
summary(fit_multi)

## End(Not run)
```

---

edgeevidence *Edge evidence plot*

Description

The edge evidence plot colors edges according to their hypothesis testing results: blue for included, dashed + light blue for weakly included, dashed + gray for inconclusive, dashed + 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 = NULL,
  evidence_thresh_strong = 10,
  evidence_thresh_weak = 3,
  edge_legend = TRUE,
  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	Deprecated. Use evidence_thresh_weak and evidence_thresh_strong.
evidence_thresh_strong	Bayes Factor which will be considered sufficient for strong in-/exclusion evidence, default is 10.
evidence_thresh_weak	Bayes Factor which will be considered sufficient for weak in-/exclusion evidence, default is 3
edge_legend	binary indicator specifying whether edge legend should be plotted. Default is TRUE.
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" for included and weakly included edges, "inconclusive", and "excluded" for excluded and weakly excluded edges.
...	Additional arguments passed onto qgraph.

Value

Returns a plot

Examples

```
## Not run:
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)[1:50, 1:5]
fit <- easybgm(data, type = "continuous",
               iter = 100   # for demonstration only
               )

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)

## End(Not run)
```

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

```
## Not run:

library(easybgm)
library(bgms)


data <- na.omit(Wenchuan)
fit <- easybgm(data[1:50, 1:5], type = "ordinal",
               iter = 100,    # for demonstration only
               edge_selection = TRUE, save = TRUE)
plot_parameterHDI(fit)

## End(Not run)
```

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 = NULL,
  evidence_thresh_strong = 10,
  dashed = FALSE,
  partial_correlations = 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	Deprecated. Use evidence_thresh_weak and evidence_thresh_strong.
evidence_thresh_strong	If dashed = TRUE, users can specify the threshold for evidence for inclusion. All edges with evidence lower than evidence_thresh_strong are dashed. Default is 10.
dashed	A binary parameter indicating whether edges with inconclusive evidence should be dashed. Default is FALSE
partial_correlations	A binary parameter indicating whether edges should show partial correlations instead of partial associations for continuous models fit with bgms. Default is FALSE.
...	Additional arguments passed onto qgraph.

Value

Returns a plot

Examples

```
## Not run:
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)[1:50, 1:5]
fit <- easybgm(data, type = "continuous",
               iter = 100   # for demonstration only
               )

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_strong = 10)

## End(Not run)
```

`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,
  evidence_thresh_strong = 10,
  evidence_thresh_weak = 3,
  ...
)
```

Arguments

output	A list of easybgm outputs with different prior edge inclusion probabilities
evidence_thresh_strong	Bayes Factor which will be considered sufficient for strong in-/exclusion evidence, default is 10.
evidence_thresh_weak	Bayes Factor which will be considered sufficient for weak in-/exclusion evidence, default is 3
...	Additional arguments passed onto ggplot2.

Details

Prior sensitivity plot

Value

Returns a plot

Examples

```
## Not run:
library(easybgm)
library(bgms)

#data <- na.omit(Wenchuan)
#fit1 <- easybgm(data[1:50, 1:5], type = "ordinal",
#               iter = 100,  # for demonstration only
#               inclusion_probability = .1
#               )
#fit2 <- easybgm(data[1:50, 1:5], type = "ordinal",
#                  iter = 100,
#                  inclusion_probability = .5
#             )
#fit3 <- easybgm(data[1:50, 1:5], type = "ordinal",
#                iter = 100, inclusion_probability = .9)

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

## End(Not run)
```

---

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

```
## Not run:

library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data[1:50, 1:5], type = "ordinal",
               iter = 100   # for demonstration only
              )

plot_structure(fit)

## End(Not run)
```

`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

```
## Not run:

library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)[1:50, 1:5]
fit <- easybgm(data, type = "ordinal", save = TRUE, edge_selection = TRUE,
               iter = 100   # for demonstration only
               )

plot_structure_probabilities(fit)

## End(Not run)
```

`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 = NULL,
  evidence_thresh_weak = 3,
  evidence_thresh_strong = 10,
  BF_uncertainty = FALSE,
  ...
)
```

Arguments

`object`	easybgm object
`evidence_thresh`	Deprecated. Use `evidence_thresh_weak` and `evidence_thresh_strong`.
`evidence_thresh_weak`	Bayes Factor which will be considered sufficient for weak in-/exclusion evidence, default is 3
`evidence_thresh_strong`	Bayes Factor which will be considered sufficient for strong in-/exclusion evidence, 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 = NULL,
  evidence_thresh_weak = 3,
  evidence_thresh_strong = 10,
  ...
)
```

Arguments

`object`	easybgm_compare object
`evidence_thresh`	Deprecated. Use `evidence_thresh_weak` and `evidence_thresh_strong`.
`evidence_thresh_weak`	Bayes Factor which will be considered sufficient for weak in-/exclusion evidence, default is 3

`evidence_thresh_strong`
: Bayes Factor which will be considered sufficient for strong in-/exclusion evidence, 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

bernoulli_prior, *8*
beta_bernoulli_prior, *8*
beta_prime_prior, *8*, *13*
bgm_extract, 2
bgm_fit, 3
bgmCompare, *13*
bgms::extract_sbm(), *4*

cauchy_prior, *8*, *13*
centrality, 3
clusterBayesfactor, 4
complexity_probs, 5

easybgm, 6
easybgm_compare, 11
edgeevidence, 15
explore, *13*

HDI, 17

network, 18
normal_prior, *8*, *13*

plot_centrality *(*centrality*)*, 3
plot_complexity_probabilities
        *(*complexity_probs*)*, 5
plot_edgeevidence *(*edgeevidence*)*, 15
plot_network *(*network*)*, 18
plot_parameterHDI *(*HDI*)*, 17
plot_prior_sensitivity
        *(*prior_sensitivity*)*, 20
plot_structure *(*structure*)*, 21
plot_structure_probabilities
        *(*structure_probs*)*, 22
print.easybgm, 19
print.easybgm_compare, 20
prior_sensitivity, 20

sbm_prior, *8*
structure, 21
structure_probs, 22
summary.easybgm, 23
summary.easybgm_compare, 24