

# Package: easybgm (via r-universe)

October 18, 2024

**Type** Package

**Title** Extracting and Visualizing Bayesian Graphical Models

**Version** 0.2.1

**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'. 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** 7.3.2

**Imports** BDgraph, BGGM, bgms (>= 0.1.3), dplyr, ggplot2, HDInterval, igraph, qgraph

**Suggests** testthat (>= 3.0.0), vdiff

**Config/testthat/edition** 3

**NeedsCompilation** no

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

**Repository** CRAN

**Date/Publication** 2024-10-17 08:30:02 UTC

## Contents

bgm_extract . . . . .	2
bgm_fit . . . . .	3
centrality . . . . .	3
complexity_probs . . . . .	4
easybgm . . . . .	5
edgevidence . . . . .	8
HDI . . . . .	9
network . . . . .	10
print.easybgm . . . . .	11
prior_sensitivity . . . . .	11
sparse_or_dense . . . . .	12
structure . . . . .	14
structure_probs . . . . .	14
summary.easybgm . . . . .	15
<b>Index</b>	<b>17</b>

---

bgm_extract	<i>Extract the results of a Bayesian analysis of networks</i>
-------------	---

---

### 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	<i>Fit a Bayesian analysis of networks</i>
---------	--

---

**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	<i>Plot strength centralities and 95% highest density interval</i>
------------	--

---

**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, ...)
```

**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 = 1000, # for demonstration only (> 5e4 recommended)
               save = TRUE, edge_selection = TRUE,
               centrality = TRUE)

plot_centrality(fit)
```

---

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 = 1000 # 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
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_alpha` the parameters of the "Beta-Bernoulli" or "Stochastic Block" priors. The default is 1 for both.
- `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"`.
- `reference_category` the reference category in the Blume-Capel model. Should be an integer within the range of integer scores observed for the 'blume-capel' variable. Can be a single number specifying the reference category for all Blume-Capel variables at once, or a vector of length  $p$  where the  $i$ -th element contains the reference category for variable  $i$  if it is Blume-Capel, and `bgm` ignores its elements for other variable types. The value of the reference category is also recoded when `bgm` recodes the corresponding observations. Only required if there is at least one variable of type "blume-capel".

### **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 2.5.
- `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  $p \times p$  matrix containing partial associations.
- `inc_probs` A  $p \times p$  matrix containing the posterior inclusion probabilities.
- `BF` A  $p \times p$  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 all packages, when setting `save = TRUE` and `centrality = TRUE`, the function will return the following objects respectively:

- `samples_posterior` A  $k \times \text{iter}$  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  $p \times \text{iter}$  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 = 1000 # for demonstration only (> 5e4 recommended)
               )

summary(fit)

# To extract the posterior parameter distribution
# and centrality measures

fit <- easybgm(data, type = "continuous",
               iter = 1000, # for demonstrative purposes, generally, 1e5 iterations are recommended
               save = TRUE,
               centrality = TRUE)
```

edgeevidence

*Edge evidence plot***Description**

The edge evidence plot colors edges according to their hypothesis testing results: blue for included, red for excluded, and gray for inconclusive. 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.
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 <code>par(mfrow = c(1, 2))</code> .
show	specifies which edges should be shown, indicated by "all", "included", "inconclusive", "excluded".
...	Additional arguments passed onto <code>qgraph</code> .

**Value**

Returns a plot

**Examples**

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
fit <- easybgm(data, type = "continuous",
               iter = 1000 # for demonstration only (> 5e4 recommended)
               )
```

```

plot_edgееvidence(fit)

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

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

#' par(mfrow = c(1, 3))
plot_edgееvidence(fit, show = "included")
plot_edgееvidence(fit, show = "inconclusive")
plot_edgееvidence(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 = 1000, # for demonstration only (> 5e4 recommended)
              save = TRUE, edge_selection = 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

<code>output</code>	Output object from the <code>easybgm</code> function. Supports also objects from the <code>bgm</code> function of the <code>bgms</code> package.
<code>exc_prob</code>	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.
<code>evidence_thresh</code>	If <code>dashed = TRUE</code> , users can specify the threshold for sufficient evidence for inclusion. All edges with evidence lower than <code>evidence_tresh</code> are dashed.
<code>dashed</code>	A binary parameter indicating whether edges with inconclusive evidence should be dashed. Default is <code>FALSE</code>
<code>...</code>	Additional arguments passed onto <code>qgraph</code> .

### Value

Returns a plot

### Examples

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)
```

```

fit <- easybgm(data, type = "continuous",
              iter = 1000 # 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	<i>Print method for easybgm objects</i>
---------------	---

---

### 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'

---

prior_sensitivity	<i>Plot sensitivity to edge inclusion prior setting</i>
-------------------	---

---

### 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 = 1000 # for demonstration only (> 5e4 recommended),
#               inclusion_probability = .1
#               )
#fit2 <- easybgm(data, type = "ordinal",
#               iter = 1000,
#               inclusion_probability = .5
#               )
#fit3 <- easybgm(data, type = "ordinal",
#               iter = 1000, inclusion_probability = .9)

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

---

sparse\_or\_dense

*Test for sparse against dense topologies*

---

**Description**

The function tests if a network is sparse (i.e., few edges in between nodes) or dense (i.e., a lot of edges between nodes). It estimates the network model of a given data set under the hypothesis that it is sparse and that it is dense, and computes the Bayes factor.

**Usage**

```
sparse_or_dense(x, type, ...)
```

**Arguments**

x	An n x p matrix or dataframe containing the variables for n independent observations on p variables.
type	What is the data type? Options: currently only binary and ordinal are supported
...	additional arguments of the bgms function

**Value**

List containing results of the analysis:

- `log.BF` The log of the Bayes factor of the test of a sparse against a dense network structure. A value larger 0 indicates evidence for a sparse structure. Contrarily, a value smaller 0 indicates evidence for a dense structure.
- `BF` The Bayes factor of the test of a sparse against a dense network structure. A value larger 1 indicates evidence for a sparse structure. Contrarily, a value between 0 and 1 indicates evidence for a dense structure.
- `relative.complexity.sparse` The relative complexity under a sparse prior hypothesis, which is the proportion of estimated included edges relative to the total possible edges under the different hypotheses.
- `relative.complexity.dense` The relative complexity under a dense prior hypothesis.
- `relative.complexity.uniform` The relative complexity under a uniform prior hypothesis.
- `no.hypotheses` The number of hypotheses that are computed in the analysis. At least the sparse, dense and uniform are used, but sometimes additional bridge hypotheses have to be computed to be able to determine the Bayes factor.

**Examples**

```
library(easybgm)
library(bgms)

data <- na.omit(Wenchuan)

# Fitting the Wenchuan PTSD data

fit <- sparse_or_dense(data, type = "ordinal",
                      iter = 1000 # for demonstration only (> 5e4 recommended)
                      )
```

---

structure	<i>Structure plot</i>
-----------	-----------------------

---

**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 = 1000 # for demonstration only (> 5e4 recommended)
               )

plot_structure(fit)
```

---

structure_probs	<i>Plot Posterior Structure Probabilities</i>
-----------------	---

---

**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 = 1000 # 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, ...)
```

**Arguments**

object	easybgm 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

`bgm_extract`, 2  
`bgm_fit`, 3

`centrality`, 3  
`complexity_probs`, 4

`easybgm`, 5  
`edgeevidence`, 8

HDI, 9

`network`, 10

`plot_centrality` (`centrality`), 3  
`plot_complexity_probabilities`  
  (`complexity_probs`), 4  
`plot_edgeevidence` (`edgeevidence`), 8  
`plot_network` (`network`), 10  
`plot_parameterHDI` (HDI), 9  
`plot_prior_sensitivity`  
  (`prior_sensitivity`), 11  
`plot_structure` (`structure`), 14  
`plot_structure_probabilities`  
  (`structure_probs`), 14  
`print.easybgm`, 11  
`prior_sensitivity`, 11

`sparse_or_dense`, 12  
`structure`, 14  
`structure_probs`, 14  
`summary.easybgm`, 15