

Package ‘robber’

July 23, 2025

Title Using Block Model to Estimate the Robustness of Ecological Network

Version 0.2.4

Description Implementation of a variety of methods to compute the robustness of ecological interaction networks with binary interactions as described in <[doi:10.1002/env.2709](https://doi.org/10.1002/env.2709)>. In particular, using the Stochastic Block Model and its bipartite counterpart, the Latent Block Model to put a parametric model on the network, allows the comparison of the robustness of networks differing in species richness and number of interactions. It also deals with networks that are partially sampled and/or with missing values.

License GPL-3

URL <https://github.com/Chabert-Liddell/robber>

BugReports <https://github.com/Chabert-Liddell/robber/issues>

Depends R (>= 3.5)

Imports blockmodels (>= 1.1.2), ggplot2, GREMLINS, igraph, magrittr, pammtools, partitions, purrr, stats

Suggests dplyr, forcats, knitr, patchwork, pbmcapply, RColorBrewer, rmarkdown, spelling, testthat, tibble, tidyr

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Saint-Clair Chabert-Liddell [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-5604-7308>>)

Maintainer Saint-Clair Chabert-Liddell <academic@chabert-liddell.com>

Repository CRAN

Date/Publication 2024-02-07 13:50:02 UTC

Contents

auc_robustness_lbm	2
compare_robustness	3
get_lbm_param	4
hostparasite	5
plot.robber	5
pollination	6
robustness_emp	6
robustness_lbm	8
seeddispersal	9
simulate_lbm	10
var_auc_unif_lbm	11
var_fun_unif_lbm	12

Index	13
--------------	-----------

auc_robustness_lbm	<i>Fast computation of the LBM robustness statistic for uniform extinctions</i>
--------------------	---

Description

This function provides much faster computation than `robustness_lbm` for the case where `ext_seq = "uniform"`, when just the AUC statistic is needed and the robustness function is not needed. This is particularly useful if `nr` gets large.

Usage

```
auc_robustness_lbm(con, pi, rho, nr, nc)
```

Arguments

<code>con</code>	A matrix, the connectivity parameter
<code>pi</code>	A vector of length <code>nrow(con)</code> , the proportion of row blocks
<code>rho</code>	A vector of length <code>ncol(con)</code> , the proportion of column blocks
<code>nr</code>	An integer, the number of row (primary) species
<code>nc</code>	An integer, the number of column (secondary) species

Value

The AUC of the LBM Robustness function for uniform primary extinctions sequences.

Examples

```
con <- matrix(c(.5,.3,.3,.1), 2, 2)
pi <- c(.25,.75)
rho <- c(1/3, 2/3)
nr <- 50
nc <- 30
auc_robustness_lbm(con, pi, rho, nr, nc)
```

compare_robustness	<i>Compare the robustness for different LBM parameters</i>
--------------------	--

Description

This function is designed to be used with a list of parameters as the one given by the function `get_lbm_param()`. The default parameters are designed such that the return value for a uniform extinction sequence is bounded above by 0.5.

Usage

```
compare_robustness(
  list_param,
  dens = 0.0156,
  new_nr = 100,
  new_nc = 100,
  ext_seq = "uniform"
)
```

Arguments

<code>list_param</code>	A list of list of LBM parameters, each list must contain at list <code>con</code> , <code>pi</code> and <code>rho</code> entries, such as the one returned by the function <code>get_lbm_param()</code> . The size of the network is optional.
<code>dens</code>	The density (connectance) used to compare the network. The com parameters of <code>list_param</code> will be normalized so that the average probability of interaction in the network is equal to <code>dens</code> . Default to 0.0156.
<code>new_nr</code>	The number of rows of the normalized networks. Default to 100.
<code>new_nc</code>	The number of columns of the normalized networks. Default to 100.
<code>ext_seq</code>	The distribution of the primary extinction sequence used to compare the networks. One of <code>c("uniform", "decreasing", "increasing")</code> . Default to "uniform".

Value

A list of the same length as `list_param`, the robustness (AUC) after normalization for all set of parameters.

Examples

```

par1 <- list(con = matrix(c(.5,.3,.3,.1), 2, 2),
            pi = c(.25,.75),
            rho = c(1/3, 2/3))
par2 <- list(con = matrix(c(.4,.1,.1,.4), 2, 2),
            pi = c(.25,.75),
            rho = c(1/2, 1/2))
compare_robustness(list_param = list(par1, par2), ext_seq = "decreasing")
compare_robustness(list_param = list(par1, par2), ext_seq = "increasing")

```

get_lbm_param

Estimate the parameters of a LBM

Description

Estimate the parameters of a LBM

Usage

```
get_lbm_param(A, model_size = NULL, ...)
```

Arguments

A	A matrix, the incident matrix of the network
model_size	A vector of size 2, the size of the model parameters. If NULL, the default, model will be chosen as the one with the highest ICL criterion among all fitted models during the inference.
...	Additional arguments pass to the inference function of package GREMLIN if there are missing value and blockmodels if none.

Value

A list of the LBM parameters. The list is set

Examples

```

# When the size of the model is unknown:
data("seeddispersal")
param <- get_lbm_param(seeddispersal, ncores = 1L)
do.call(auc_robustness_lbm, param)

# For a fixed number of blocks (useful for comparison)
param <- get_lbm_param(seeddispersal, model_size = c(1, 3), ncores = 1L)
do.call(auc_robustness_lbm, param)

# For data with missing observations

A <- seeddispersal

```

```
A[sample(seq_len(nrow(A)), 5),
  sample(seq_len(ncol(A)), 10)] <- NA
param <- get_lbm_param(A, nbCores = 1L)
do.call(auc_robustness_lbm, param)
```

hostparasite	<i>A binarized host-parasite network</i>
--------------	--

Description

Locality: Tomsk-Tumen Reference:Hadfield JD, Krasnov BR, Poulin R, Shinichi N (2013) A tale of two phylogenies: comparative analyses of ecological interactions. *The American Naturalist* 183(2): 174-187

Usage

```
hostparasite
```

Format

A matrix of binary entries with 27 rows and 26 columns

Source

<https://www.web-of-life.es/>

<https://datadryad.org/stash/dataset/doi:10.5061/dryad.jf3tj>

plot.robber	<i>Plot function pour robber class</i>
-------------	--

Description

Plot function pour robber class

Usage

```
## S3 method for class 'robber'
plot(x, type = "function", add = FALSE, ...)
```

Arguments

x	An object of class robber
type	Type of the plot, just function
add	Whether it should be added to a previous ggplot
...	lty, col, ... to be passed to geom_step() in ggplot2

Value

A ggplot2 object

Examples

```
data("hostparasite", package = "robber")
rob <- robustness_emp(hostparasite)
plot(rob)
```

pollination	<i>A plant-pollinator network</i>
-------------	-----------------------------------

Description

Locality: Tenerife, Canary Islands Stald L (2003). Struktur og dynamik i rum og tid af et bestøvningsnetværk på Tenerife, De Kanariske Øer. Msc thesis (Univ of Aarhus, Aarhus, Denmark)

Usage

```
pollination
```

Format

A matrix of binary entries with 17 rows and 51 columns

Source

<https://www.web-of-life.es/>

robustness_emp	<i>Empirical Robustness of a Network</i>
----------------	--

Description

Compute the robustness of an ecological network by averaging over a great number of randomly generated primary extinctions sequences

Usage

```
robustness_emp(  
  A,  
  ext_seq = "uniform",  
  nb_iter = 300,  
  net_type = "bipartite",  
  method = "ordered",  
  power = 1,  
  block = NULL  
)
```

Arguments

A	A binary incident matrix
ext_seq	A string, the rule for the primary extinctions sequences, one of "uniform", the default for uniform extinctions sequences, "decreasing" and "increasing" for primary extinctions sequences by increasing and decreasing degree order sequence on the row species
nb_iter	An integer, the number of generated sequences. Default to 300.
net_type	A string, the type of network. For now, only "bipartite" is available.
method	A string used when ext_seq = "decreasing" or ext_seq = "increasing": <ul style="list-style-type: none"> • "linear" (default) set an extinction probability for each row species that is linear in its degree. It is a shortcut for method = "power" and power = 1. • "ordered" strictly follow the degree order, i.e. the most connected species will always goes last for ext_seq = "increasing" or first for ext_seq = "decreasing".
power	An integer (default to 1). Only used when method = "power", the power to which the degree are elevated to compute the primary extinctions sequences for "increasing" and "decreasing" ext_seq. For 1, it is equivalent to method = "linear", for 0, it is the same as ext_seq = "uniform". The largest the number the closest it will be to method = "ordered".
block	A vector of row species memberships for method = "block"

Value

A list (robber object) with the following fields:

- \$model, \$ext_seq, \$method, power
- \$fun is the robustness function, a vector of size (nrow(A) +1)
- \$sd Standard deviation of the secondary extinctions
- \$auc the area under the curve of the robustness function

Examples

```
A <- matrix(c(1, 0, 0,
             0, 1, 0,
             0, 0, 1,
             1, 1, 1), nrow = 4, ncol = 3, byrow = TRUE)
my_rob <- robustness_emp(A, ext_seq = "increasing")
my_rob$fun
my_rob$auc
```

robustness_lbm	<i>Robustness for LBM</i>
----------------	---------------------------

Description

Compute the robustness for a given set of Latent Block Model parameters.

Usage

```
robustness_lbm(
  con = NULL,
  pi = NULL,
  rho = NULL,
  nr = NULL,
  nc = NULL,
  ext_seq = "uniform",
  method = "exact",
  approx_threshold = 10000,
  net = NULL,
  ...
)
```

Arguments

con	A matrix, the connectivity parameter
pi	A vector of length <code>nrow(con)</code> , the proportion of row blocks
rho	A vector of length <code>ncol(con)</code> , the proportion of column blocks
nr	An integer, the number of row (primary) species
nc	An integer, the number of column (secondary) species
ext_seq	A string, the rule for the primary extinctions sequences, one of: <ul style="list-style-type: none"> • "uniform", the default for uniform extinctions sequences, • "decreasing" and "increasing" for primary extinctions sequences by increasing and decreasing row blocks connection probability, • "natural" primary extinctions sequences with the block ordering given in the function parameter. • "blocks" primary extinctions sequences for all blocks permutation.
method	A string, the method used to compute the robustness by block. One of: <ul style="list-style-type: none"> • "exact" (the default), average on all possible networks • "no_empty_block" restricts the networks space to the ones with no empty block of primary species • "expected" restricts the networks space to the ones that have approximately the same number of species per block than the expected one. Do not do anything for <code>ext_seq = "uniform"</code>.

approx_threshold	A numeric, the maximum size of the possible block partition allowed for exact robustness by block calculation. Higher threshold gives more precise results at the cost of computation times and possibly memory problem. Do not do anything for <code>ext_seq = "uniform"</code> . When the threshold is exceeded the robustness is computed by a Monte Carlo approximation with <code>approx_threshold</code> number of samples.
net	A network, if given, the function will fit a LBM to obtain the parameters of the network and then compute the robustness.
...	Option to be passed to <code>get_lbm_param</code> function.

Value

A list and a robber type object:

- `$fun` the robustness function, a vector of size `nr + 1`
- `$auc` the area under the curve of the robustness function
- `$block` a vector of size `length(pi)`, the block ordering for primary extinctions sequence by blocks. NULL if `ext_seq = "uniform"`.
- `$model`, `$method`, `$ext_seq`, `$param`.

Examples

```
con <- matrix(c(.5,.3,.3,.1), 2, 2)
pi <- c(.25,.75)
rho <- c(1/3, 2/3)
nr <- 50
nc <- 30
my_rob <- robustness_lbm(con, pi, rho, nr, nc, ext_seq = "natural")
my_rob$fun
my_rob$auc
# A easier alternative way, if you don't know the parameters of the network:
data(hostparasite)
(robustness_lbm(net = hostparasite, ncores = 1L))
```

seeddispersal

A seed-dispersal network

Description

Locality: North Negros Forest Reserve, Central Philippi Hammann, A. & Curio, B. (1999) Interactions among frugivores and fleshy fruit trees in a Philippine submontane rainforest.

Usage

```
seeddispersal
```

Format

A matrix of binary entries with 36 rows and 19 columns

Source

<https://www.web-of-life.es/>

<https://conbio.org/publications/conservation-biology>

simulate_lbm

Simulate a bipartite interaction network using block model

Description

Simulate a bipartite interaction network using block model

Usage

```
simulate_lbm(con, pi, rho, nr, nc, method = "gnp")
```

Arguments

con	A matrix, the connectivity between blocks. If method = "gnp" then each entry is the probability of interactions given 2 blocks. If method = "gnm" then each entry is the number of interactions between 2 blocks.
pi	A vector of the same length as nrow(con), the block mixture for the row species. If method = "gnp", then pi is a probability vector, if method = "gnm", then pi is the number of species per block, must sum to nr.
rho	A vector of the same length as ncol(con), the block mixture for the column species. If method = "gnm", then rho is a probability vector, if method = "gnm", then rho is the number of species per block, must sum to nc.
nr	The number of row Species
nc	The number of column Species
method	One of "gnp" (the default) where the blocks size and number of interactions is random and "gnm" where the blocks size and number of interactions are fixed.

Value

A list of 3 elements:

- A The incident matrix of size nr x nc
- Z A vector of length nr, the block assignment of the row species
- W A vector of length nc, the block assignment of the column species

Examples

```
# For a random number of interactions and blocks sizes
con <- matrix(c(.5,.3,.3,.1), 2, 2)
pi <- c(.25,.75)
rho <- c(1/3, 2/3)
nr <- 50
nc <- 30
simulate_lbm(con, pi, rho, nr, nc, method = "gnp")

# For a fixed number of interactions and blocks sizes
con <- matrix(c(50, 120, 60, 80), 2, 2)
pi <- c(10, 40)
rho <- c(10, 20)
nr <- 50
nc <- 30
simulate_lbm(con, pi, rho, nr, nc, method = "gnm")
```

var_auc_unif_lbm	<i>Compute the Conditional variance of the AUC of the LBM Robustness</i>
------------------	--

Description

Compute the Conditional variance of the AUC of the LBM Robustness

Usage

```
var_auc_unif_lbm(con, pi, rho, nr, nc)
```

Arguments

con	A matrix, the connectivity parameter
pi	A vector of length <code>nrow(con)</code> , the proportion of row blocks
rho	A vector of length <code>ncol(con)</code> , the proportion of column blocks
nr	An integer, the number of row (primary) species
nc	An integer, the number of column (secondary) species

Value

A numeric, the variance

Examples

```
con <- matrix(c(.5,.3,.3,.1), 2, 2)
pi <- c(.25,.75)
rho <- c(1/3, 2/3)
nr <- 50
nc <- 30
var_auc_unif_lbm(con, pi, rho, nr, nc)
```

var_fun_unif_lbm	<i>Compute the Conditional variance of the LBM Robustness term by term</i>
------------------	--

Description

Compute the Conditional variance of the LBM Robustness term by term

Usage

```
var_fun_unif_lbm(con, pi, rho, nr, nc)
```

Arguments

con	A matrix, the connectivity parameter
pi	A vector of length <code>nrow(con)</code> , the proportion of row blocks
rho	A vector of length <code>ncol(con)</code> , the proportion of column blocks
nr	An integer, the number of row (primary) species
nc	An integer, the number of column (secondary) species

Value

A vector, the variance after `m` extinctions

Examples

```
con <- matrix(c(.5,.3,.3,.1), 2, 2)
pi <- c(.25,.75)
rho <- c(1/3, 2/3)
nr <- 50
nc <- 30
var_fun_unif_lbm(con, pi, rho, nr, nc)
```

Index

* datasets

- hostparasite, 5
- pollination, 6
- seeddispersal, 9

auc_robustness_lbm, 2

compare_robustness, 3

get_lbm_param, 4

hostparasite, 5

plot.robber, 5

pollination, 6

robustness_emp, 6

robustness_lbm, 8

seeddispersal, 9

simulate_lbm, 10

var_auc_unif_lbm, 11

var_fun_unif_lbm, 12