Package ‘econetwork’

July 8, 2019

Type Package

Title Analyzing Ecological Networks

Version 0.2

Date 2019-07-08

Author Stephane Dray [ctb], Catherine Matias [ctb], Vincent Miele [aut, cre], Marc Ohlmann [aut], Wilfried Thuiller [ctb]

Maintainer Vincent Miele <vincent.miele@univ-lyon1.fr>

Description A collection of advanced tools, methods and models specifically designed for analyzing different types of ecological networks - especially antagonistic (food webs, host-parasite), mutualistic (plant-pollinator, plant-fungus, etc) and competitive networks, as well as their variability in time and space. Statistical models are developed to describe and understand the mechanisms that determine species interactions, and to decipher the organization of these (multi-layer) ecological networks.

Imports stats, igraph, rdiversity, Matrix.utils

LinkingTo

License GPL-3

URL https://plmlab.math.cnrs.fr/econetproject/econetwork

NeedsCompilation no

Repository CRAN

Date/Publication 2019-07-08 13:10:03 UTC

R topics documented:

econetwork-package .............................................................. 2
disPairwise ................................................................. 2
divPartition .............................................................. 4
getMetaweb .............................................................. 6

Index 7
Description

A collection of advanced tools, methods and models specifically designed for analyzing different types of ecological networks.

Details

econetwork is designed for analyzing different types of ecological networks - especially antagonistic (food webs, host-parasite), mutualistic (plant-pollinator, plant-fungus, etc.) and competitive networks, as well as their variability in time and space. Statistical models are developed to describe and understand the mechanisms that determine species interactions, and to decipher the organization of these (multi-layer) ecological networks.

Author(s)

Authors: Stephane Dray, Catherine Matias, Vincent Miele, Marc Ohlmann, Wilfried Thuiller
Maintainer: Vincent Miele <vincent.miele@univ-lyon1.fr>

References


See Also

igraph bipartite vegan sna statnet enaR

disPairwise

Computation of the dissimilarity matrix (pairwise beta-diversity) for a set of networks

Description

Computation of the dissimilarity matrix for a set of networks. Each value of the matrix is the pairwise beta-diversity, computed using Hill numbers. It measures the dissimilarity in terms of groups, links, or probability of links.

Usage

disPairwise(gList, groups=NULL, eta=1,
  type=c('P','L','Pi'), abTable=NULL)
disPairwise

Arguments

- gList: A list of graph objects of class igraph. The nodes must have a name.
- groups: A named vector of class character indicating the group to which each node belongs to. The length of groups must correspond to the number of different nodes present in gList. The names names(groups) must correspond to the nodes names in gList. If NULL, the groups are the initial nodes.
- eta: A positive number that controls the weight given to abundant groups/links. Default value is 1.
- type: The type of diversity used to measure dissimilarity. It can be groups diversity ('P'), links diversity ('L') or probability of links diversity ('Pi').
- abTable: A matrix of size the number of nodes of the metanetwork times the number of networks. The rownames of this matrix must be the node names of metanetwork and the columns must be in an order corresponding to gList. The element (i,j) of this matrix is the abundance of species i in network j. Importantly, the non-nul elements in each column of abTable must correspond to the nodes present in each element of gList.

Value

Return a matrix whose elements are the pairwise dissimilarities.

Author(s)

Authors: Stephane Dray, Vincent Miele, Marc Ohlmann, Wilfried Thuiller
Maintainer: Wilfried Thuiller <wilfried.thuiller@univ-grenoble-alpes.fr>

References


Examples

# Generating a set of Erdos-Renyi graphs and give node names.
library(econetwork)
library(igraph)

nbGraph <- 10

gList <- c()
n <- 57 # number of nodes of each graph
C <- 0.1    # connectance of each graph

for(i in 1:nbGraph){
  graphLocal <- erdos.renyi.game(n, type='gnp', p.or.m=C, directed=TRUE)
  V(graphLocal)$name <- as.character(1:57)
  gList = c(gList,list(graphLocal))
}

# vector that gives the group of each node

groups <- c(rep("a",23),rep("b",34))
names(groups) <- as.character(1:57)
# generating random (non-nul) abundances data
abTable <- sapply(1:nbGraph,function(x) rpois(n,1)+1)
rownames(abTable) = unlist(unique(lapply(glist,function(g) V(g)$name)))

# Dissimilarity matrices based on links beta-diversity
# at a node level
disPairwise(glist, type = 'L')
# at a node level while taking into account node abundances
disPairwise(glist, type = 'L', abTable = abTable)
# at a group level
disPairwise(glist, groups, type = 'L')
# at a group level while taking into account node abundances
disPairwise(glist, groups, type = 'L', abTable = abTable)

---

divPartition

**Partitionning network diversity in alpha, beta and gamma diversity**

**Description**

This function computes alpha, beta and gamma diversity of a list of networks. It measures either group, links, or probability of links diversity.

**Usage**

```r
divPartition(glist, groups, eta=1, framework=c('RLC','Chao'),
              type=c('P','L','Pi'), abTable=NULL)
```

**Arguments**

- `gList`: A list of graph objects of class igraph.
- `groups`: A named vector of class character indicating the group to which each node belongs to. The length of groups must correspond to the number of different nodes present in gList. The names names(groups) must correspond to the nodes names in gList. If NULL, the groups are the initial nodes.
- `eta`: A positive number that controls the weight given to abundant groups/links. Default value is 1.
- `framework`: The framework used to partitionate diversity, either Reeve Leinster Cobbold ('RLC') or Chao ('Chao').
- `type`: The type of diversity to measure and partitionate. It can be groups diversity ('P'), link diversity ('L') or probability of link diversity ('Pi').
- `abTable`: A matrix of size the number of nodes of the metanetwork times the number of networks. The rownames of this matrix must be the node names of metanetwork and the columns must be in an order corresponding to gList. The element (i,j) of this matrix is the abundance of species i in network j. Importantly, the non-nul elements in each column of abTalbe must correspond to the nodes present in each element of gList.
**Value**

Returns a list the following components:

- **mA**<sub>lpha</sub> The mean value of alpha-diversity across all networks.
- **alph**as A vector of numeric containing the local alpha-diversities (i.e. the alpha-diversity value for each network).
- **Beta** The value of the overall beta-diversity
- **Gamma** The value of the gamma-diversity

**Author(s)**

Authors: Stephane Dray, Vincent Miele, Marc Ohlmann, Wilfried Thuiller
Maintainer: Wilfried Thuiller <wilfried.thuiller@univ-grenoble-alpes.fr>

**References**


**Examples**

```r
# Generating a set of Erdos-Renyi graphs and give node names.
library(econetwork)
library(igraph)
nbGraph <- 10
gList <- c()
n <- 57 # number of nodes of each graph
C <- 0.1 # connectance of each graph
for(i in 1:nbGraph){
  graphLocal <- erdos.renyi.game(n, type='gnp', p.or.m=C, directed=TRUE)
  V(graphLocal)$name <- as.character(1:n)
  glist = c(glist, list(graphLocal))
}

# vector that gives the group of each node
groups <- c(rep("a", 23), rep("b", 34))
names(groups) <- as.character(1:n)
# generating random (non-nul) abundances data
abTable <- sapply(1:nbGraph, function(x) rpois(n, 1)+1)
rownames(abTable) = unlist(unique(lapply(glist, function(g) V(g)$name)))

# Diversities in link abundances
# at a node level
divPartition(gList, framework='Chao', type = 'L')
# at a node level while taking into account node abundances
divPartition(gList, framework='Chao', type = 'L', abTable = abTable)
# at a group level
divPartition(gList, framework='Chao', groups, type = 'L')
# at a group level while taking into account node abundances
divPartition(gList, framework='Chao', groups, type = 'L', abTable = abTable)
```
Description

Computation of the binary metaweb from a list of graph

Usage

getMetaweb(gList)

Arguments

gList A list of graph objects of class igraph.

Details

This function computes the metaweb from a list of graph. It computes the union (in the sense of
graph theory) of the set of graphs.

Value

getMetaweb returns an object of class igraph.

Author(s)

Authors: Stephane Dray, Vincent Miele, Marc Ohlmann, Wilfried Thuiller
Maintainer: Wilfried Thuiller <wilfried.thuiller@univ-grenoble-alpes.fr>

References

Marc Ohlmann, Vincent Miele, Stephane Dray, Loic Chalmandrier, Louise O’Connor & Wilfried
Thuiller, Diversity indices for ecological networks: a unifying framework using Hill numbers. Ecol-

Examples

# Generating a set of Erdos-Renyi graphs
library(igraph)
gList <- c()
for(i in 1:4){
  graphLocal <- erdos.renyi.game(60, type = 'gnp', p.or.m=0.1, directed=TRUE)
  V(graphLocal)$name <- as.character(1:60)
  gList <- c(gList, list(graphLocal))
}
names(gList) <- c("A","B","C","D")

## building the metaweb
graphMetaweb <- getMetaweb(gList)
Index

*Topic diversity
econetwork-package, 2

*Topic ecology
econetwork-package, 2

*Topic network
econetwork-package, 2

*Topic package
econetwork-package, 2

bipartite, 2
disPairwise, 2
divPartition, 4
econetwork (econetwork-package), 2
econetwork-package, 2
enaR, 2
getMetaweb, 6
igraph, 2
sna, 2
statnet, 2
vegan, 2