

# Package ‘NetworkExtinction’

November 22, 2022

**Type** Package

**Title** Extinction Simulation in Ecological Networks

**Version** 1.0.1

**Maintainer** Derek Corcoran <derek.corcoran.barrios@gmail.com>

**Description** Simulates the extinction of species in ecological networks and it analyzes its cascading effects, described in Dunne et al. (2002) <[doi:10.1073/pnas.192407699](https://doi.org/10.1073/pnas.192407699)>.

**URL** <https://derek-corcoran-barrios.github.io/NetworkExtinction/>

**Depends** R (>= 3.5.1)

**Imports** broom, doParallel, dplyr, foreach, ggplot2, igraph, magrittr, network, scales, sna, stats, tidyr, MASS, parallel, purrr, methods

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**Suggests** knitr, rmarkdown, pkgdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Derek Corcoran [aut, cre] (<<https://orcid.org/0000-0002-0248-6230>>),  
M. Isidora Ávila-Thieme [aut] (<<https://orcid.org/0000-0003-0772-4717>>),  
Fernanda S. Valdovinos [aut],  
Sergio A. Navarrete [aut],  
Pablo A. Marquet [aut] (<<https://orcid.org/0000-0001-6369-9339>>),  
Erik Kusch [aut] (<<https://orcid.org/0000-0002-4984-7646>>)

**Repository** CRAN

**Date/Publication** 2022-11-22 21:50:02 UTC

**R topics documented:**

chilean_intertidal . . . . .	2
chilean_potential . . . . .	3
chilean_weighted . . . . .	3
CompareExtinctions . . . . .	4
DegreeDistribution . . . . .	5
dist . . . . .	6
ExtinctionOrder . . . . .	6
ExtinctionPlot . . . . .	8
Less_Connected . . . . .	9
More_Connected . . . . .	9
net . . . . .	10
RandomExtinctions . . . . .	10
SimulateExtinctions . . . . .	12

<b>Index</b>	<b>16</b>
--------------	-----------

---

chilean_intertidal	<i>The binaryfoodweb of the intertidal zone in central chile</i>
--------------------	--

---

**Description**

A trophic network with 107 species present in the intertidal zone of central Chile. The food web was reconstructed from the Kefi et al. 2015

**Usage**

```
chilean_intertidal
```

**Format**

```
a network
```

**References**

Kefi, Sonia, Eric L. Berlow, Evie A. Wieters, Lucas N. Joppa, Spencer A. Wood, Ulrich Brose, and Sergio A. Navarrete. "Network structure beyond food webs: mapping non trophic and trophic interactions on Chilean rocky shores." *Ecology* 96, no. 1 (2015).

---

chilean\_potential      *The potential foodweb of the intertidal zone in central chile*

---

**Description**

A trophic network with 107 species present in the intertidal zone of central Chile. The food web was reconstructed from the Kefi et al. 2015

**Usage**

chilean\_potential

**Format**

a network

---

chilean\_weighted      *The weighted foodweb of the intertidal zone in central chile*

---

**Description**

A trophic network with 107 species present in the intertidal zone of central Chile. The food web was reconstructed from the Kefi et al. 2015

**Usage**

chilean\_weighted

**Format**

a network

**References**

Kefi, Sonia, Eric L. Berlow, Evie A. Wieters, Lucas N. Joppa, Spencer A. Wood, Ulrich Brose, and Sergio A. Navarrete. "Network structure beyond food webs: mapping non trophic and trophic interactions on Chilean rocky shores." *Ecology* 96, no. 1 (2015).

---

CompareExtinctions      *Comparison of Null hypothesis with other extinction histories*

---

### Description

It compares an object generated either by the Mostconnected or ExtinctionOrder functions with a null hypothesis generated by the RandomExtinctions function it is important that RandomExtinctions is in plot = T.

### Usage

```
CompareExtinctions(Nullmodel, Hypothesis)
```

### Arguments

Nullmodel      an object generated by the RandomExtinctions  
Hypothesis      Extinction history generated by the Mostconnected or ExtinctionOrder function

### Value

a plot comparing the expected value of secondary extinctions originated at random with the observed extinction history.

### Author(s)

Derek Corcoran <derek.corcoran.barrios@gmail.com>

M. Isidora Ávila-Thieme <msavila@uc.cl>

### Examples

```
## Not run:  
data("Less_Connected")  
History <- SimulateExtinctions(Network = Less_Connected, Method = "Mostconnected")  
NullHyp <- RandomExtinctions(Network = Less_Connected, nsim = 100)  
CompareExtinctions(Nullmodel = NullHyp, Hypothesis = History)  
  
## End(Not run)
```

---

DegreeDistribution      *Degree distribution of the network*

---

### Description

This function calculates the degree distribution of the network. First it fits exponential, power law and truncated power law distribution models, and calculates the AIC values to select the best fit, and finally it plots the degree distribution in a log log scale showing the three fitted models mentioned above against the observed distribution.

### Usage

```
DegreeDistribution(Network, scale = "arithmetic")
```

### Arguments

Network	a trophic network of class network
scale	a character stating if the graph is on a log-log scale ("LogLog") or arithmetic scale ("arithmetic"), defaults to arithmetic

### Value

exports three principal results: 1. A list with network degree distribution values and with the value of each fit model 2. A list with each model results and AIC of the distribution models 3. A Ghraph of the degree distribution with the models adjust In DDvalues, k represent the degree of the network and cumulative the probability that each specie could be have this degree (pk). Observation: In the graph, the zero values are not represented but this result are incorporate in the DF result

### Author(s)

Derek Corcoran <derek.corcoran.barrios@gmail.com>

M.Isidora Avila Thieme <msavila@uc.cl>

### Examples

```
library(NetworkExtinction)
data("chilean_intertidal")
DegreeDistribution(chilean_intertidal)
```

---

dist	<i>A toymodel distance matrix</i>
------	-----------------------------------

---

**Description**

A distance matrix used for demonstration of rewiring capabilities

**Usage**

```
dist
```

**Format**

a distance matrix

---

ExtinctionOrder	<i>Extinctions analysis from custom order</i>
-----------------	---

---

**Description**

This function takes a network and eliminates nodes using a custom order. Subsequently, secondary extinctions are tallied up. Secondary extinction severity can be targeted by manipulating the node-dependency on network edges (IS) and node-rewiring potential upon loss of links (Rewiring).

**Usage**

```
ExtinctionOrder(
  Network,
  Order,
  NetworkType = "Trophic",
  clust.method = "cluster_infomap",
  IS = 0,
  Rewiring = FALSE,
  RewiringDist,
  RewiringProb = 0.5,
  verbose = TRUE,
  RecalcConnect = FALSE
)
```

**Arguments**

Network	a network representation as a an adjacency matrix, edgelist, or a network object
Order	a numeric vector indexing order of primary extinctions. For Method = Mostconnected Order must be NULL. If Order is not NULL, Method is internally forced to be Ordered.

NetworkType	a character with the options Trophic and Mutualistic - is used to calculate secondary extinctions.
clust.method	a character with the options cluster_edge_betweenness, cluster_spinglass, cluster_label_prop or cluster_infomap, defaults to cluster_infomap
IS	either numeric or a named vector of numerics. Identifies the threshold of relative interaction strength which species require to not be considered secondarily extinct (i.e. IS = 0.3 leads to removal of all nodes which lose 70percent of their interaction strength in the Network argument). If a named vector, names must correspond to vertex names in Network argument.
Rewiring	either a function or a named vector of functions. Signifies how rewiring probabilities are calculated from the RewiringDist argument. If FALSE, no rewiring is carried out.
RewiringDist	a numeric matrix of NxN dimension (N... number of nodes in Network). Contains, for example, phylogenetic or functional trait distances between nodes in Network which are used by the Rewiring argument to calculate rewiring probabilities. If Rewiring == function(x)x, this matrix is expected to contain probabilities of a connection being present between species-pairs.
RewiringProb	a numeric which identifies the threshold at which to assume rewiring potential is met.
verbose	Logical. Whether to report on function progress or not.
RecalcConnect	Logical. Whether to recalculate connectedness of each node following each round of extinction simulation and subsequently update extinction order with newly mostconnected nodes.

### Details

When NetworkType = Trophic, secondary extinctions only occur for any predator, but not producers. If NetworkType = Mutualistic, secondary extinctions occur for all species in the network.

When clust.method = cluster\_edge\_betweenness computes the network modularity using cluster\_edge\_betweenness methods from igraph to detect communities When clust.method = cluster\_spinglass computes the network modularity using cluster\_spinglass methods from igraph to detect communities, here the number of spins are equal to the network size When clust.method = cluster\_label\_prop computes the network modularity using cluster\_label\_prop methods from igraph to detect communities When clust.method = cluster\_infomap computes the network modularity using cluster\_infomap methods from igraph to detect communities, here the number of nb.trials are equal to the network size

### Value

exports list containing a data frame with the characteristics of the network after every extinction and a network object containing the final network. The resulting data frame contains 11 columns that incorporate the topological index, the secondary extinctions, predation release, and total extinctions of the network in each primary extinction.

### Author(s)

Derek Corcoran <derek.corcoran.barrios@gmail.com>

M. Isidora Ávila-Thieme <msavila@uc.cl>

Erik Kusch <erik.kusch@bio.au.dk>

---

ExtinctionPlot

*Plots the extinctions history of a network*

---

### Description

It takes a NetworkTopology class object and plots the network index after every extinction

### Usage

```
ExtinctionPlot(History, Variable = "AccSecExt")
```

### Arguments

History            a NetworkTopology object obtained from the Mostconnected function or the ExtinctionOrder function

Variable           the variable of the NetworkTopology object that you want as a y variable

### Value

A plot of number of extinctions in the x axis vs the choosen variable in the Y axis

### Author(s)

Derek Corcoran <derek.corcoran.barrios@gmail.com>

M. Isidora Ávila-Thieme <msavila@uc.cl>

### See Also

[NetworkExtintion::ExtinctionOrder()]

### Examples

```
# If you don't specify the y variable it will plot the secondary extinctions
# by default
data("net")
history <- SimulateExtinctions(Network = net, Method = "Mostconnected")
ExtinctionPlot(History = history$sims)
# You can also specify the variable to be plotted in the y axis
ExtinctionPlot(History = history$sims, Variable = "Link_density")
```



---

Less_Connected	<i>A sparsely connected foodweb</i>
----------------	-------------------------------------

---

**Description**

A network with 30 species and 47 interactions. This network has a connectance of 0.03

**Usage**

Less\_Connected

**Format**

a network

**See Also**

[More\\_Connected](#)

---

More_Connected	<i>A densely connected foodweb</i>
----------------	------------------------------------

---

**Description**

A trophic network with 30 species and 222 trophic interactions. This foodweb has a connectance of 0.3

**Usage**

More\_Connected

**Format**

a network

**See Also**

[Less\\_Connected](#)

---

net	<i>A toymodel trophic network</i>
-----	-----------------------------------

---

**Description**

A trophic network with 10 species where the first four species are primery producters

**Usage**

net

**Format**

a network

---

RandomExtinctions	<i>Random extinction</i>
-------------------	--------------------------

---

**Description**

Generates a null model by generating random extinction histories and calculating the mean and standard deviation of the accumulated secondary extinctions developed by making n random extinction histories.

**Usage**

```
RandomExtinctions(
  Network,
  nsim = 10,
  Record = FALSE,
  plot = FALSE,
  SimNum = NULL,
  NetworkType = "Trophic",
  clust.method = "cluster_infomap",
  parallel = FALSE,
  ncores,
  IS = 0,
  Rewiring = FALSE,
  RewiringDist = NULL,
  RewiringProb = 0.5,
  verbose = TRUE
)
```

**Arguments**

Network	a network representation as a an adjacency matrix, edgelist, or a network object
n <sub>sim</sub>	numeric, number of simulations
Record	logical, if TRUE, records every simulation and you can read the raw results in the object FullSims
plot	logical if TRUE, will add a graph to the results
SimNum	numeric, how many nodes to register for primary extinction. By default sets all of them.
NetworkType	a character with the options Trophic and Mutualistic - is used to calculate secondary extinctions.
clust.method	a character with the options cluster_edge_betweenness, cluster_spinglass, cluster_label_prop or cluster_infomap, defaults to cluster_infomap
parallel	if TRUE, it will use parallel procesing, if FALSE (default) it will run sequentially
ncores	numeric, number of cores to use if using parallel procesing
IS	either numeric or a named vector of numerics. Identifies the threshold of relative interaction strength which species require to not be considered secondarily extinct (i.e. IS = 0.3 leads to removal of all nodes which lose 70 percent of their interaction strength in the Network argument). If a named vector, names must correspond to vertex names in Network argument.
Rewiring	either a function or a named vector of functions. Signifies how rewiring probabilities are calculated from the RewiringDist argument. If FALSE, no rewiring is carried out.
RewiringDist	a numeric matrix of NxN dimension (N... number of nodes in Network). Contains, for example, phylogenetic or functional trait distances between nodes in Network which are used by the Rewiring argument to calculate rewiring probabilities. If Rewiring == function(x)x, this matrix is expected to contain probabilities of a connection being present between species-pairs.
RewiringProb	a numeric which identifies the threshold at which to assume rewiring potential is met.
verbose	Logical. Whether to report on function progress or not.

**Details**

When NetworkType = Trophic, secondary extinctions only occur for any predator, but not producers. If NetworkType = Mutualistic, secondary extinctions occur for all species in the network.

When clust.method = cluster\_edge\_betweenness computes the network modularity using cluster\_edge\_betweenness methods from igraph to detect communities When clust.method = cluster\_spinglass computes the network modularity using cluster\_spinglass methods from igraph to detect communities, here the number of spins are equal to the network size When clust.method = cluster\_label\_prop computes the network modularity using cluster\_label\_prop methods from igraph to detect communities When clust.method = cluster\_infomap computes the network modularity using cluster\_infomap methods from igraph to detect communities, here the number of nb.trials are equal to the network size

**Value**

exports list containing a data frame with the characteristics of the network after every extinction, a network object containing the final network, and a graph with the mean and 95percent interval. The resulting data frame contains 11 columns that incorporate the topological index, the secondary extinctions, predation release, and total extinctions of the network in each primary extinction.

**Author(s)**

Derek Corcoran <derek.corcoran.barrios@gmail.com>

M. Isidora Ávila-Thieme <msavila@uc.cl>

Erik Kusch <erik.kusch@bio.au.dk>

**Examples**

```
#first example
## Not run:
data("More_Connected")
RandomExtinctions(Network = More_Connected, nsim = 20)

# Using parallel procesing
## Detect your number of cores divide by 2

cores <- ceiling(parallel::detectCores()/2)

RandomExtinctions(Network = More_Connected, nsim = 20, parallel = TRUE, ncores = cores)

## End(Not run)
```

---

SimulateExtinctions     *Extinctions analysis for ecological networks*

---

**Description**

The SimulateExtinctions function, can be used to test how the order of species extinctions, species-dependency on existing interaction strength, and rewiring potential might affect the stability of the network by comparing The extinction history and checking for secondary extinctions.

**Usage**

```
SimulateExtinctions(
  Network,
  Method,
  Order = NULL,
  NetworkType = "Trophic",
  clust.method = "cluster_infomap",
  IS = 0,
```

```

    Rewiring = FALSE,
    RewiringDist,
    RewiringProb = 0.5,
    verbose = TRUE
)

```

### Arguments

Network	a network representation as a an adjacency matrix, edgelist, or a network object
Method	a character with the options Mostconnected and Ordered
Order	a numeric vector indexing order of primary extinctions. For Method = Mostconnected Order must be NULL. If Order is not NULL, Method is internally forced to be Ordered.
NetworkType	a character with the options Trophic and Mutualistic - is used to calculate secondary extinctions.
clust.method	a character with the options cluster_edge_betweenness, cluster_spinglass, cluster_label_prop or cluster_infomap, defaults to cluster_infomap
IS	either numeric or a named vector of numerics. Identifies the threshold of relative interaction strength which species require to not be considered secondarily extinct (i.e. IS = 0.3 leads to removal of all nodes which lose 70 percent of their interaction strength in the Network argument). If a named vector, names must correspond to vertex names in Network argument.
Rewiring	either a function or a named vector of functions. Signifies how rewiring probabilities are calculated from the RewiringDist argument. If FALSE, no rewiring is carried out.
RewiringDist	a numeric matrix of NxN dimension (N... number of nodes in Network). Contains, for example, phylogenetic or functional trait distances between nodes in Network which are used by the Rewiring argument to calculate rewiring probabilities. If Rewiring == function(x)x, this matrix is expected to contain probabilities of a connection being present between species-pairs.
RewiringProb	a numeric which identifies the threshold at which to assume rewiring potential is met.
verbose	Logical. Whether to report on function progress or not.

### Details

When method is Mostconnected, the function takes the network and calculates which node is the most connected of the network, using total degree. Then remove the most connected node, and calculates the the topological indexes of the network and the number of secondary extinctions. This process is repeated until the entire network has gone extinct.

When method is Ordered, it takes a network, and extinguishes nodes using a custom order, then it calculates the secondary extinctions and plots the accumulated secondary extinctions.

When NetworkType = Trophic, secondary extinctions only occur for any predator, but not producers. If NetworkType = Mutualistic, secondary extinctions occur for all species in the network.

When clust.method = cluster\_edge\_betweenness computes the network modularity using cluster\_edge\_betweenness methods from igraph to detect communities When clust.method = cluster\_spinglass computes the

network modularity using cluster\_spinglass methods from igraph to detect communities, here the number of spins are equal to the network size When clust.method = cluster\_label\_prop computes the network modularity using cluster\_label\_prop methods from igraph to detect communities When clust.method = cluster\_infomap computes the network modularity using cluster\_infomap methods from igraph to detect communities, here the number of nb.trials are equal to the network size

### Value

exports list containing a data frame with the characteristics of the network after every extinction and a network object containing the final network. The resulting data frame contains 11 columns that incorporate the topological index, the secondary extinctions, predation release, and total extinctions of the network in each primary extinction.

### Author(s)

Derek Corcoran <derek.corcoran.barrios@gmail.com>

M. Isidora Ávila-Thieme <msavila@uc.cl>

Erik Kusch <erik.kusch@bio.au.dk>

### Examples

```
# Mostconnected example
data("net")
SimulateExtinctions(Network = net, Method = "Mostconnected",
  clust.method = "cluster_infomap")

#first Ordered example
data("net")
SimulateExtinctions(Network = net, Order = c(1,2,3,4,5,6,7,8,9,10),
  Method = "Ordered" , clust.method = "cluster_infomap")

#Second Ordered example
data("net")
SimulateExtinctions(Network = net, Order = c(2,8,9),
  Method = "Ordered", clust.method = "cluster_infomap")

#Network-Dependency Example
data("net")
SimulateExtinctions(Network = net, Order = c(2,8), IS = 0.3,
  Method = "Ordered", clust.method = "cluster_infomap")

#Rewiring
data("net")
data(dist)
SimulateExtinctions(Network = net, Order = c(2,8), IS = 0.3,
  # assuming an exponential decline in rewiring potential
  # as values in RewiringDist increase
  Rewiring = function(x){1-pexp(x, rate = 1/0.5)},
  RewiringDist = dist, # distance matrix
  RewiringProb = 0.2, # low threshold for rewiring potential
  Method = "Ordered", clust.method = "cluster_infomap")
```

```
#Rewiring, assuming dist contains probabilities
#' data("net")
data(dist)
SimulateExtinctions(Network = net, Order = c(2,8), IS = 0.3,
Rewiring = function(x){x}, # no changes to the RewiringDist object means
RewiringDist = dist, RewiringProb = 0.2,
Method = "Ordered", clust.method = "cluster_infomap")
```

# Index

## \* datasets

- chilean\_intertidal, [2](#)
- chilean\_potential, [3](#)
- chilean\_weighted, [3](#)
- dist, [6](#)
- Less\_Connected, [9](#)
- More\_Connected, [9](#)
- net, [10](#)

- chilean\_intertidal, [2](#)
- chilean\_potential, [3](#)
- chilean\_weighted, [3](#)
- CompareExtinctions, [4](#)

- DegreeDistribution, [5](#)
- dist, [6](#)

- ExtinctionOrder, [6](#)
- ExtinctionPlot, [8](#)

- Less\_Connected, [9](#), [9](#)

- More\_Connected, [9](#), [9](#)

- net, [10](#)

- RandomExtinctions, [10](#)

- SimulateExtinctions, [12](#)