

Package ‘enaR’

September 28, 2017

Type Package

Title Tools for Ecological Network Analysis

Version 3.0.0

Date 2017-09-28

Author M.K. Lau, S.R. Borrett, D.E. Hines, P. Singh

Maintainer Matthew K. Lau <enaR.maintainer@gmail.com>

Description Provides algorithms for the analysis of ecological networks.

Depends R (>= 3.1.0)

Imports stringr, sna, network, MASS, gdata, graphics, stats, limSolve,
utils

Suggests codetools, igraph, R.rsp

VignetteBuilder R.rsp

License GPL-3

URL <https://github.com/SEELab/enaR>

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-09-28 16:06:06 UTC

R topics documented:

enaR-package	3
as.bipartite	3
as.extended	4
bal	5
balance	6
bcratio	7
bgcModels	7
cycliv	8
EcoNetWeb	9

eigenCentrality	10
enaAll	11
enaAscendency	12
enaControl	14
enaCycle	15
enaEnviron	17
enaFlow	18
enaModelInfo	20
enaModels	20
enaMTI	21
enaStorage	22
enaStructure	24
enaTroAgg	25
enaUncertainty	27
enaUtility	32
environCentrality	33
findPathLength	34
force.balance	35
get.ns	36
get.orient	37
m.list	38
meanTrophicLevel	38
mExp	39
netOrder	39
oyster	40
pack	41
read.EcoNet	42
read.enam	42
read.nea	43
read.scor	44
read.wand	45
relationalChange	46
scc	47
scifix	48
set.orient	49
ShannonDiversity	50
signs	51
ssCheck	52
structure.statistics	53
TES	54
TET	55
troModels	56
unpack	56
write.EcoNet	57
write.nea	58

enaR-package

Tools for Ecological Network Analysis (ena)

Description

This package compiles functions for the analysis of ecological networks, building on tools previously developed in the MatLab language (Borrett 2006) with multiple additions of functionality.

Details

Package: enaR
Type: Package
Version: 2.10
Date: 2017-02-06
License: GPL-3

Author(s)

Authors: Stuart R. Borrett, Matthew K. Lau, Pawandeep Singh, David E. Hines
Maintainer: Matthew K. Lau <enaR.maintainer@gmail.com>

References

Borrett SR and Lau MK 2014. enaR: An r package for Ecosystem Network Analysis. *Methods in Ecology and Evolution* 5:1206-1213.

See Also

[network](#)

as.bipartite

Converts a network object (unipartite) to a two-mode (bipartite) network representation.

Description

Bipartite network approaches are often used for analyzing the structure of interactions among species in communities. Although typically ecosystem networks are handled using a unipartite representation, analyzing them from a bipartite perspective may be informative. This function provides an easy means for converting to a bipartite representation as long as there is a natural division to categorize species into distinct modes.

Usage

```
as.bipartite(x = "network object", y = "membership vector")
```

Arguments

x A network object.
y A vector of membership values.

Value

Returns a matrix with the species of one mode arrayed in rows and the other in columns.

Author(s)

Matthew K. Lau

Examples

```
data(oyster)  
as.bipartite(oyster, gl(2, 3))
```

as.extended

Create an Extended Format Matrix

Description

Converts a network object to the extended format of Allesina and Bondavalli (2003). Used in the balance function.

Usage

```
as.extended(x, zero.na = TRUE)
```

Arguments

x A network object.
zero.na Logical: should NA's be replaced with zeros?

Value

Returns an extended format matrix.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Allesina, S., Bondavalli, C., 2003. Steady state of ecosystem flow networks: a comparison between balancing procedures. *Ecological Modelling* 165(2-3):231-239.

See Also

[balance](#)

Examples

```
data(troModels)
as.extended(troModels[[6]])
```

bal

Balancing by Either Inputs or Outputs

Description

Dependency for the balance function.

Usage

```
bal(T.star = "matrix", method = c("input", "output"))
```

Arguments

T.star	Extended, unbalanced matrix.
method	Balance by inputs or outputs.

Value

Returns an extended matrix for balancing by inputs or outputs.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Fath, B.D. and S.R. Borrett. 2006. A MATLAB function for network environ analysis. *Environmental Modelling & Software* 21:375-405.

See Also[balance](#)

balance*Balance Flow Network Models*

Description

Applies the methods of Allesina and Bondavalli (2003) for balancing flow network models.

Usage

```
balance(x, method = c("AVG2", "AVG", "IO", "OI", "I", "O"), tol = 5)
```

Arguments

<code>x</code>	A network object.
<code>method</code>	Methods for model balancing, see Allesina and Bondavalli (2003).
<code>tol</code>	Percent error tolerance used in the steady state check prior to balancing.

Value

Returns a network object with a balanced flow network model.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Allesina, S., Bondavalli, C., 2003. Steady state of ecosystem flow networks: a comparison between balancing procedures. *Ecological Modelling* 165(2-3):231-239.

See Also[bal](#)**Examples**

```
data(troModels)
balance(troModels[[6]])
```

bcratio *Calculates the Ratio of Positive to Negative Elements in a Network*

Description

Dependent function for the enaUtility function.

Usage

```
bcratio(x = "matrix")
```

Arguments

x A matrix of flow values.

Value

Returns the ratio of positive to negative elements in the flow matrix.

Author(s)

Stuart R. Borrett

References

Fath, B.D. and S.R. Borrett. 2006. A MATLAB function for network environ analysis. *Environmental Modelling & Software* 21:375-405.

See Also

[enaUtility](#)

bgcModels *Biogeochemical Cycling Models*

Description

A set of 43 biogeochemical cycling models compiled by the SEE Lab at UNCW.

References

Borrett, S. R., and M. K. Lau. In Prep. enaR: An R package for Ecological Network Analysis. *Ecological Modeling and Software*.

cycliv

*Analysis of Feeding Cycles in a Network***Description**

Performs the full cycle analysis on the living subset of the network based on the algorithm described in Ulanowicz (1983) and implemented in NETWRK 4.2b. It returns data.frames with details of the simple cycles and nexus, vectors of Cycle distributions and Normalized distribution and matrices of Residual Flows and Aggregated Cycles.

Usage

```
cycliv(x)
```

Arguments

x a network object. This includes all weighted flows into and out of each node. It must also include the "Living" vector that identifies the living (TRUE/FALSE) status of each node. Also, non-living nodes must be placed at the end of the node vector. The function netOrder can be used to reorder the network for this.

Details

This function uses the same mechanism for analysis as used in the enaCycle function but is restricted to the living nodes only.

Also, similar to the enaCycle function, if the number of cycles in a nexus is more than 50, the "Table.cycle" has a blank line after 50 cycles followed by the cycles for the next nexus.

The analysis requires all the non-living nodes to be placed at the end in the network object.

Value

Table.cycle	data.frame that presents the details of the simple cycles in the network. It contains "CYCLE" the cycle number, "NEXUS" the nexus number corresponding to the cycle, "NODES" the nodes constituting the cycle
Table.nexus	data.frame that presents the different nexuses characterized by their corresponding weak arcs. It contains "NEXUS" the nexus number, "CYCLES" the number of simple cycles present in that Nexus, "W.arc.From" the starting node of the corresponding weak arc, "W.arc.To" the ending node of the corresponding weak arc and "W.arc.Flow" the flow through that weak arc
CycleDist	vector of the Cycle Distribution that gives the flow which is cycling in loops of different sizes
NormDist	vector of the Normalized Distribution i.e. the Cycle Distribution normalized by the Total System Throughput for the network
ResidualFlows	matrix of the straight-through (acyclic) flows in the network
AggregatedCycles	matrix of the Aggregated Biogeochemical Cycles in the network

ns vector of the full cycle analysis based network statistics. These include "NCYCS" the number of simple cycles identified in the network, "NNEX" the number of the disjoint cycles or number of Nexuses detected in the network and "CI" the cycling index of the network.

Author(s)

Pawandeep Singh

References

- D.B. 1975. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.* 4:77–84
- Ulanowicz, R.E. 1983. Identifying the structure of cycling in ecosystems. *Mathematical Biosciences* 65:219–237
- Ulanowicz, R.E. and Kay, J.J. 1991. A package for the analysis of ecosystem flow networks. *Environmental Software* 6:131 – 142.

See Also

[enaTroAgg](#), [enaCycle](#), [netOrder](#)

EcoNetWeb

Access example EcoNet models from the EcoNet website.

Description

This function allows the user to access models that are presented on the website for EcoNet, the web-based interface for conducting ENA (<http://eco.engr.uga.edu/>), by Caner Kazanci at the University of Georgia.

Usage

```
EcoNetWeb(model.name = "prompt",  
          url = "http://eco.engr.uga.edu/Examples/examples.html")
```

Arguments

`model.name` The model to be accessed. If 'prompt' the user will be asked for the model they wish to use. Can also be a number for the model or the name of the model.

`url` The URL for the EcoNet examples.

Value

Returns the model formatted as a network object.

Author(s)

Matthew K. Lau

References

Kazanci, C., 2007. EcoNet: A new software for ecological modeling, simulation and network analysis, *Ecol. Model.*, Vol 208/1 pp 3-8.

See Also

[read.EcoNet](#)

eigenCentrality *the Eigen Centrality of a Network*

Description

Calculates the centrality of a network using eigen vectors.

Usage

```
eigenCentrality(x = "matrix")
```

Arguments

x A matrix defining a network graph.

Value

Returns the eigen based centrality of the network.

Author(s)

Stuart R. Borrett Matthew K. Lau

References

Bonacich, P., 1987. Power and centrality: a family of measures. *American Journal of Sociology* 92: 1170-1182.

enaAll	<i>Conduct All Major ENA</i>
--------	------------------------------

Description

Conducts all major ENA with default settings and returns the output as a named list.

Usage

```
enaAll(x = "network object")
```

Arguments

x	A network object.
---	-------------------

Value

ascendency	enaAscendency
control	enaControl
environ	enaEnviron
flow	enaFlow
mti	enaMTI
storage	enaStorage
structure	enaStructure
utility	enaUtility with eigen.check=FALSE

Author(s)

Matthew K. Lau Stuart R. Borrett

See Also

[enaAscendency](#), [enaControl](#), [enaEnviron](#), [enaFlow](#), [enaMTI](#), [enaStorage](#), [enaUtility](#)

Examples

```
data(troModels)
output = enaAll(troModels[[6]])
names(output)
```

enaAscendency *Calculates the Ascendency of an Ecological Network*

Description

Calculates the average mutual information (AMI), ascendency, overhead, and capacity of input-output networks. It also returns the ratios of ascendency and overhead to capacity. These metrics describe the organization of flow in an ecological network (Ulanowicz 1997).

Usage

```
enaAscendency(x = "network object")
```

Arguments

x A network object.

Details

This and other Ulanowicz school functions require that export and respiration components of output be separately quantified.

Value

H	Total flow diversity (MacArthur 1955). Uses the Shannon Information measure (aka Boltzmann entropy) applied to the individual flows.
AMI	Returns the Average Mutual Information (AMI) in a network. AMI provides a measure of the constraints placed on a given peice of energy matter moving through a network (Patricio et al. 2006)
Hr	Residual uncertainty that remains about the flow distribution once the ecosystem structure is specified ($H_r = H - AMI$).
ASC	Returns the ascendency of a network. Ascendency is a scaled form of AMI relative to the total system throughput (Ulanowicz 1997; 2004). Total system throughput is the sum of all activity in a network (Kay et al. 1989).
OH	Returns the overhead of a network. Overhead is the proportion of the capacity in a network that is not used as ascendency (Ulanowicz 2004).
CAP	Returns the capacity of a network. Capacity is defined as the sum of ascendency and overhead (Ulanowicz 2004).
ACS.CAP	Returns the proportion of capacity used by ascendency.
OH.CAP	Returns the proportion of capacity used by overhead.
robustness	Returns the robustness of the network.
ELD	Returns the Effective Link Density of the network(c) (Ulanowicz et al. 2014).
TD	Returns the Trophic Depth of the network(r) (Ulanowicz et al. 2014).
A. input	Returns the input ascendency of a network.

A.internal	Returns the internal ascendency of a network.
A.export	Returns the export ascendency of a network.
A.respiration	Returns the respiration ascendency of a network.
OH.input	Returns the input overhead of a network.
OH.internal	Returns the internal overhead of a network.
OH.export	Returns the export overhead of a network.
OH.respiration	Returns the respiration overhead of a network.
CAP.input	Returns the input capacity of a network.
CAP.internal	Returns the internal capacity of a network.
CAP.export	Returns the export capacity of a network.
CAP.respiration	Returns the respiration capacity of a network.

Author(s)

David E. Hines Matthew K. Lau Stuart R. Borrett

References

- Kay, J.J., Graham, L.A., Ulanowicz, R.E., 1989. A detailed guide to network analysis. p. 15-61 In: Wulff, F., Field, J.G., Man, K.H. (eds.) Network analysis in marine ecology. Coastal Estuarine Study Series. Springer-Verlag, Berlin.
- Patricio, J., Ulanowicz, R.E., Pardal, M.A., Marques J.C., 2004. Ascendency as an ecological indicator: a case study of estuarine pulse eutrophication. Estuar. Coast Shelf S. 60, 23-35.
- Ulanowicz, R.E. and Norden, J.S., 1990. Symmetrical overhead in flow networks. International Journal of Systems Science, 21(2), pp.429-437.
- Ulanowicz, R.E., 1997. Ecology, The Ascendent Perspective. Columbia University Press, New York.
- Ulanowicz, R.E., 2004. Quantitative methods for ecological network analysis. Comput. Biol. Chem. 28, 321-33
- Ulanowicz, R.E., Holt, R.D., Barfield, M., 2014. Limits on ecosystem trophic complexity: insights from ecological network analysis. Ecology Letters 17:127-136

See Also

[read.scor](#), [read.wand](#), [enaStorage](#), [enaUtility](#)

Examples

```
data(troModels)
enaAscendency(troModels[[6]])
```

enaControl

*Control Analyses of Ecological Networks***Description**

Analyses for analyzing the control amongst the nodes in ecological networks.

Usage

```
enaControl(x, zero.na = TRUE, balance.override = FALSE)
```

Arguments

x	A network object.
zero.na	Makes undefined (NA) values zero.
balance.override	Turns off balancing and checks of network balance.

Value

CN	Control matrix using flow values.
CQ	Control matrix using storage values.
CR	Schramski Control Ratio Matrix
CD	Schramski Control Difference Matrix
CA	Control Allocation Matrix
CDep	Control Dependency Matrix
sc	Schramski System Control vector
scp	Schramski system control vector as percent of total control
ns	vector of network-level summary statistics

Author(s)

Matthew K. Lau Stuart R. Borrett Pawandeep Singh

References

- Fath, B. D., Borrett, S. R. 2006. A MATLAB function for Network Environ Analysis. *Environmental Modelling & Software* 21:375-405
- Schramski, J.R., Gattie, D.K., Patten, B.C., Borrett S.R., Fath, B.D., Thomas, C.R., and Whipple, S.J. 2006. Indirect effects and distributed control in ecosystems: Distributed control in the environ networks of a seven compartment model of nitrogen flow in the Neuse River Estuary, USA Steady-state analysis. *Ecological Modelling* 194:189-201
- Schramski, J.R., Gattie, D.K., Patten, B.C., Borrett S.R., Fath, B.D., and Whipple, S.J. 2007. Indirect effects and distributed control in ecosystems: Distributed control in the environ networks of a

seven compartment model of nitrogen flow in the Neuse River Estuary, USA Time series analysis. Ecological Modelling 206:18-30

Chen, S., Fath, B.D., Chen, B. 2011. Information-based network environ analysis: a system perspective for ecological risk assessment. Ecol. Ind. 11:1664-1672.

Chen, S. and Chen, B. 2015. Urban energy consumption: Different insights from energy flow analysis, input-output analysis and ecological network analysis. Applied Energy 138:99-107.

See Also

[enaStorage](#)

Examples

```
data(troModels)
enaControl(troModels[[6]])
```

enaCycle

Full Cycle Analysis of Ecological Networks

Description

It performs the full cycle analysis on the network based on the algorithm described in Ulanowicz (1983) and implemented in NETWRK 4.2b. It returns data.frames with details of the simple cycles and nexus, vectors of Cycle distributions and Normalized distribution and matrices of Residual Flows and Aggregated Cycles.

Usage

```
enaCycle(x)
```

Arguments

x a network object. This includes all weighted flows into and out of each node.

Details

The "NODES" in "Table.cycle" are arranged such that the weak arc for the nexus is the arc between the first two nodes of the cycle. This function uses the backtracking procedure for the identification of simple cycles, which are cycles that cross a node only once except the starting node. The backtracking process is a depth-first search algorithm.

In the data.frame "Table.cycle", if the number of cycles in a nexus is more than 50, then a blank line is displayed after 50 cycles of the nexus, followed by the cycles of the next nexus.

The results of the analysis of Feeding Cycles can be obtained as a byproduct of the `enaTroAgg` function that analyzes the trophic dynamics of a network.

At every multiple of 5000 cycles in a nexus, the program prints an indication for the user to know that it is still running.

Value

<code>Table.cycle</code>	data.frame that presents the details of the simple cycles in the network. It contains "CYCLE" the cycle number, "NEXUS" the nexus number corresponding to the cycle, "NODES" the nodes constituting the cycle
<code>Table.nexus</code>	data.frame that presents the different nexuses characterized by their corresponding weak arcs. It contains "NEXUS" the nexus number, "CYCLES" the number of simple cycles present in that Nexus, "W.arc.From" the starting node of the corresponding weak arc, "W.arc.To" the ending node of the corresponding weak arc and "W.arc.Flow" the flow through that weak arc
<code>CycleDist</code>	vector of the Cycle Distribution that gives the flow cycling in loops of different sizes
<code>NormDist</code>	vector of the Normalized Distribution i.e. the Cycle Distribution normalized by the Total System Throughput of the system
<code>ResidualFlows</code>	matrix of the straight-through (acyclic) flows in the network
<code>AggregatedCycles</code>	matrix of the Aggregated Biogeochemical Cycles in the network
<code>ns</code>	vector of the full cycle analysis based network statistics. These include "NCYCS" the number of simple cycles identified in the network, "NNEX" the number of the disjoint cycles or number of Nexuses detected in the network and "CI" the cycling index of the network.

Author(s)

Pawandeep Singh

References

- D.B. 1975. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.* 4:77–84
- Ulanowicz, R.E. 1983. Identifying the structure of cycling in ecosystems. *Mathematical Biosciences* 65:219–237
- Ulanowicz, R.E. and Kay, J.J. 1991. A package for the analysis of ecosystem flow networks. *Environmental Software* 6:131 – 142.

See Also

[enaTroAgg](#)

Examples

```
data(troModels)
cyc6 <- enaCycle(troModels[[6]])
attributes(cyc6)
```

enaEnviron	<i>Ecological Network Environs</i>
------------	------------------------------------

Description

Calculates the environs for an ecological network.

Usage

```
enaEnviron(x, input = TRUE, output = TRUE, type = "unit",
  err.tol = 1e-10, balance.override = FALSE)
```

Arguments

x	A network object.
input	Should the input environ be calculated?
output	Should the output environ be calculated?
type	Specifies the type of environs ("unit" or "realized") to be calculated.
err.tol	Error threshold for numerical error fluctuations in flows. Values below err.tol will be set to zero.
balance.override	Logical specifying whether (TRUE) or not (FALSE) the model needs to be balanced prior to calculations. If TRUE and the model is not balanced, environs will not be calculated.

Value

The function returns the input, output or both environs depending upon which were requested.

Author(s)

Stuart R. Borrett Matthew K. Lau

References

Fath, B.D. and S.R. Borrett. 2006. A MATLAB function for network environ analysis. *Environmental Modelling & Software* 21:375-405.

Examples

```
data(troModels)
enaEnviron(troModels[[6]])
```

 enaFlow

Analyses of Ecological Networks

Description

Performs the primary throughflow analysis developed for input-output systems. It returns a vector of throughflows, the input and output oriented matrices for "direct flow intensities" and "integral flow intensities", and a set of flow based network statistics.

Usage

```
enaFlow(x, zero.na = TRUE, balance.override = FALSE)
```

Arguments

x	a network object. This includes all weighted flows into and out of each node.
zero.na	LOGICAL: should NA values be converted to zeros.
balance.override	Flow analysis assumes the network model is at steady-state (inputs = outputs). Setting balance.override = TRUE allows the function to be run on unbalanced models.

Value

T	vector of node throughflows total amount of energy-matter flowing into or out of each node
G	matrix of the output oriented direct flow intensities
GP	matrix of the input oriented direct flow intensities
N	matrix of the output oriented integral (boundary+direct+indirect) flow intensities
NP	matrix of the input oriented integral flow intensities
TCC	matrix of total contribution coefficients (Szyrmer & Ulanowicz 1987). The elements of TCC indicate the fraction of total output of i which reaches j
TDC	matrix of total dependency coefficients (Szyrmer & Ulanowicz 1987). The elements of TDC indicate the fraction j's total consumption which passes through i

ns vector of flow based network statistics. These include "Boundary" the total input into or output from the system, "TST" the total system throughflow, "TSTp" total system throughPUT, "APL" is the network aggradation TST/Boundary which is also called average path length, "FCI" (Finn Cycling Index) is a metric of the amount of cycling in a system, "BFI" is the boundary flow intensity Boundary/TST, "DFI" is the direct flow intensity Direct/TST, "IFI" is the indirect flow intensity Indirect/TST, "ID.F" is the realized indirect to direct flow intensity, "ID.F.I" is the input idealized indirect flow intensity, "id.F.O" is the output idealized indirect flow intensity, "HMG.I" is the input network homogenization, "HMG.O" is the output network homogenization, "AMP.I" is the strong measure of input network amplification, "AMP.O" is the strong measure of output network amplification, "mode0.F" is the boundary flow - flow that reaches a compartment from across the system boundary, "mode1.F" is internal first passage flow, "mode2.F" is cycled flow, "mode3.F" is the dissipative equivalent to mode2, and "mode4.F" is the dissipative equivalent of mode0.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

- Borrett, S. R., Freeze, M. A., 2011. Reconnecting environs to their environment. *Ecol. Model.* 222, 2393-2403.
- Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. *Environ. Model. Softw.* 21, 375-405.
- Fath, B. D., Patten, B. C., 1999. Review of the foundations of network environ analysis. *Ecosystems* 2, 167-179.
- Finn, J. T., 1976. Measures of ecosystem structure and function derived from analysis of flows. *J. Theor. Biol.* 56, 363-380.
- Patten, B.C. Higashi, M., Burns, T. P. 1990. Trophic dynamics in ecosystem networks: significance of cycles and storage. *Ecol. Model.* 51, 1-28.
- Schramski, J. R., Kazanci, C., Tollner, E. W., 2011. Network environ theory, simulation and EcoNet 2.0. *Environ. Model. Softw.* 26, 419-428.
- Szyrmer, J., Ulanowicz, R. E., 1987. "Total Flows in Ecosystems". *Ecol. Mod.* 35:123-136.
- Ulanowicz, R.E., 2004. Quantitative methods for ecological network analysis. *Comput. Biol. Chem.* 28, 321-33
- Ulanowicz, R.E., Holt, R.D., Barfield, M., 2014. Limits on ecosystem trophic complexity: insights from ecological network analysis. *Ecology Letters* 17:127-136.

See Also

[read.scor](#), [read.wand](#), [enaStorage](#), [enaUtility](#)

Examples

```
data(troModels)
F = enaFlow(troModels[[6]]) # completes the full analysis
F$ns # returns just the network statistics
```

enaModelInfo	<i>Ecosystem Model Information</i>
--------------	------------------------------------

Description

Model information for the set of ecosystem models compiled by the SEE Lab at UNCW.

References

Borrett, S. R., and M. K. Lau. In Prep. enaR: An R package for Ecological Network Analysis. Ecological Modeling and Software.

enaModels	<i>Ecosystem Models</i>
-----------	-------------------------

Description

A set of ecosystem models compiled by the SEE Lab at UNCW.

References

Borrett, S. R., and M. K. Lau. In Prep. enaR: An R package for Ecological Network Analysis. Ecological Modeling and Software.

enaMTI	<i>Mixed Trophic Impacts (MTI) Analysis</i>
--------	---

Description

Calculates the Mixed Trophic Impacts of one species on another in the given ecosystem model following the algorithm of Ulanowicz and Puccia (1990). This considers both the direct and indirect trophic impacts.

Usage

```
enaMTI(x, zero.na = TRUE, balance.override = FALSE)
```

Arguments

x	a network object. This includes all weighted flows into and out of each node. It must also include the "Living" vector that identifies the living (TRUE/FALSE) status of each node.
zero.na	A logical parameter that specifies if NAs generated in the analysis should be reset to zero. The default is TRUE.
balance.override	Mixed Trophic Impacts analysis builds on flow analysis and thus assumes the network model is at steady-state (inputs = outputs). Setting balance.override = TRUE allows the function to be run on unbalanced models, though this is unadvised.

Details

This and other Ulanowicz school functions require that export and respiration components of output be separately quantified.

This analysis is similar in concept to the ENA Utility analysis.

Value

G	output-oriented direct flow intensity matrix as in enaFlow, except oriented from row to column.
FP	input-oriented direct flow intensity matrix similar to enaFlow; however, the calculation exclude respiration losses from the throughflow in the denominator to focus on NET production. Also, if the receiver compartment is not living, the flux intensity is set to zero.
Q	direct net trophic impacts (G-t(FP)).
M	Total (direct and indirect) trophic impacts of compartment i on j.
Relations.Table	A table indicating the qualitative pairwise relationships between the nodes as determined from the net (direct) and the mixed (integral) perspectives.

Author(s)

Stuart R. Borrett Matthew K. Lau

References

R.E. and C.J. Puccia. 1990. Mixed trophic impacts in ecosystems. *Coenoses* 5, 7–16.

See Also

[enaFlow](#), [enaUtility](#)

Examples

```
data(troModels)
mti <- enaMTI(troModels[[6]])
attributes(mti)
```

enaStorage

Storage Analyses of Ecological Networks

Description

Calculates storage-based Ecological Network Analyses.

Usage

```
enaStorage(x, balance.override = FALSE)
```

Arguments

x A network object. This This includes all weighted flows into and out of each vertex as well as the amount of energy–matter stored at each vertex.

balance.override

LOGICAL: should an imbalanced model be analyzed? If FALSE, the functions checks to make sure the network model provided is at steady-state. If TRUE, then the function will run without ensuring that the model meets the steady-state assumption.

Value

X	The storage values themselves.
C	output or donor-storage normalized output-oriented direct flow intensity matrix (Jacobian community matrix)
S	dimensionalized integral output community matrix
VS	variance in expected output-oriented residence times (Barber 1979)
Q	integral output storage matrix - non-dimensional
CP	input or recipient-storage normalized oriented flow intensity matrix (Jacobian community matrix)
SP	dimensionalized integral input community matrix
VSP	variance in expected input-oriented residence times (Barber 1979)
QP	integral input storage matrix - non-dimensional
dt	selected time step to create P, PP, Q and QP - smallest whole number to make diag(C) nonnegative
ns	vector of the storage based whole system network statistics. These statistics include total system storage (TSS), storage cycling index (CIS), Boundary storage intensity (BSI), Direct storage intensity (DSI), Indirect storage intensity (ISI), realized ratio of indirect-to-direct storage (ID.S), unit input-oriented ratio of indirect-to-direct storage intensities (IDS.I), unit output ratio of indirect-to-direct storage intensities (IDS.O), input-oriented storage-based network homogenization (HMG.S.I), output-oriented storage-based network homogenization (HMG.S.O), input-oriented storage-based network amplification (AMP.S.I), output-oriented storage-based network amplification (AMP.S.O), Storage from Boundary flow (mode0.S), storage from internal first passage flow (mode1.S), storage from cycled flow (mode2.S), dissipative equivalent to mode1.S (mode3.S), dissipative equivalent to mode0.S (mode4.S).

Author(s)

Matthew K. Lau Stuart R. Borrett

References

- Barber, M. C. 1978a. A Markovian Model for Ecosystem Flow Analysis. *Ecol. Model.* 5(3):193-206.
- Barber, M. C. 1978b. A Retrospective Markovian Model for Ecosystem Resource Flow. *Ecol. Model.* 5(2): 125-35.
- Barber, M. C. 1979. A Note Concerning Time Parameterization of Markovian Models of Ecosystem Flow Analysis. *Ecol. Model.* 6(4): 323-28.
- Matis, J. H., Patten, B. C. 1981. Environ analysis of linear compartmental systems: the static, time invariant case. *Bulletin of the International Statistical Institute*, 48: 527-565.
- Fath, B. D., Patten, B. C. 1999. Review of the foundations of network enviorn analysis. *Ecosystems* 2:167-179.

Fath, B. D. Patten, B. C., Choi, J. 2001. Compementarity of ecological goal functions. *Journal of Theoretical Biology* 208: 493-506.

Fath, B. D., Borrett, S. R. 2006. A MATLAB function for Network Environ Analysis. *Environmental Modelling & Software* 21:375-405.

See Also

[read.scor](#), [read.wand](#), [enaFlow](#), [enaUtility](#)

Examples

```
data(oyster)
S <- enaStorage(oyster)
attributes(S)
```

enaStructure

Structure Analyses of Ecological Network

Description

Analysis of the structure of an ecological flow network.

Usage

```
enaStructure(x = "network object")
```

Arguments

x A network object.

Value

A
ns A vector of structure based network statistics. These include n = number of nodes, L = number of edges, C = connectivity, LD = link density, ppr = pathway proliferation rate, lam1A = dominant eigenvalue, mlam1A = multiplicity of dominant eigenvalue, rho = damping ratio, R = distance of the dominant eigen value from the eigen spectra, d = difference between dominant eigen value and link density, no.scc = number of strongly connected components, no.scc.big = number of strongly connected components with more than one node, pscc = percent of nodes in strongly connected components.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. Environ. Model. Softw. 21, 375-405.

See Also

[structure.statistics](#)

Examples

```
data(troModels)
enaStructure(troModels[[6]])
```

enaTroAgg

Trophic Aggregations (TroAgg) Analysis

Description

It returns the data quantifying the underlying trophic structure of a given model based on the interaction of the living and non-living nodes. It is based on the Trophic Aggregations suggested by Lindeman (1942) and follows the algorithm by Ulanowicz and Kemp (1979) implemented in NETWRK 4.2b. It removes the Feeding cycles in the network beforehand to provide accurate results.

Usage

```
enaTroAgg(x, balance.override = FALSE)
```

Arguments

x a network object. This includes all weighted flows into and out of each node. It should include separate respiration and export values for the Canonical Exports and Canonical Respirations results respectively. It must also include the "Living" vector that identifies the living (TRUE/FALSE) status of each node. It must contain the non-living nodes at the end of the node vector, the function [netOrder](#) can be used for the same.

balance.override

Flow analysis assumes the network model is at steady-state (inputs = outputs). Setting `balance.override = TRUE` allows the function to be run on unbalanced models.

Details

This and other Ulanowicz school functions require that export and respiration components of output be separately quantified.

This analysis involves the ENA Cycle analysis for removal of the Feeding Cycles in the network. These are cycles amongst only the living nodes and cause error in the trophic aggregations.

The analysis requires all the non-living nodes to be placed at the end in the network object.

Value

Feeding_Cycles	List that gives the details of the Feeding Cycles in the network. The output being according to the enaCycle function applied to the Living components in the network
A	matrix that distributes the species in integer Trophic Levels (Lindeman Transformation Matrix). The dimension of A is (NL X NL) where NL is the number of Living nodes.
ETL	vector of the Effective Trophic Level of each species.
M.flow	vector of the Migratory flows, if present, in the network.
CI	vector of Canonical Inputs to the integer trophic levels. Displayed if the Migratory flows are present.
CE	vector of Canonical exports or the exports from the integer trophic levels
CR	vector of the Canonical Respirations or the respiration values for integer trophic levels.
GC	vector of the input flow to a trophic level from the preceeding trophic level. It represents the Grazing Chain for the network.
RDP	vector of the Returns to Detrital Pool from each trophic level.
LS	vector of the Lindeman trophic spine. It combines the Detrital pool with the autotrophs and forms a monotonically decreasing sequence of flows from one trophic level to the next, starting with the said combination.
TE	vector of the trophic efficiencies i.e. the ratio of input to a trophic level to the amount of flow that is passed on the next level from it.
ns	vector of trophic aggregations based network statistics. These include the average Trohic Level ("ATL"), "Detritivory" the flow from the detrital pool to the second trophic level, "DetritalInput" the exogenous inputs to the detrital pool, "DetritalCirc" the circulation within the detrital pool, "NCYCS" the number of feeding cycles removed, "NNEX" the number of feeding cycle Nexuses removed and "CI" the Cycling Index for the Feeding Cycles.

Author(s)

Pawandeep Singh

References

- R.L. 1942. The trophic-dynamic aspect of ecology. *Ecology* 23:399–418.
- Ulanowicz, R.E. and Kemp, W.M. 1979. Towards canonical trophic aggregations. *The American Naturalist*. 114:871–883.
- Ulanowicz, R.E. 1995. Ecosystem trophic foundations: Lindeman exonerata. pp. 549–560. B.C. Patten and S.E. Jorgensen (eds.) *Complex Ecology: The part-whole relation in ecosystems*. Prentice Hall, New Jersey.
- Ulanowicz, R.E. and Kay, J.J. 1991. A package for the analysis of ecosystem flow networks. *Environmental Software* 6:131 – 142.

See Also

[enaCycle](#), [netOrder](#)

Examples

```
data(troModels)
tro6 <- enaTroAgg(troModels[[6]])
attributes(tro6)
```

enaUncertainty

Produce a set of plausible network models

Description

Connects enaR to linSolve to apply Linear Inverse Modelling to conduct an uncertainty analysis for Ecological Network Analysis. Users supply an initial ecosystem model (in the enaR format) and uncertainty information (several ways of specifying), and the function returns a list (length = "iter") of balanced plausible instantiations of the model. This has been used to determine the 95 and to determine the statistical significance of selected comparisons (Hines et al. 2015, 2016).

Usage

```
enaUncertainty(x = "network object", type = "percent", iter = 10000,
  p.err = NA, F.sym = NA, z.sym = NA, y.sym = NA, e.sym = NA,
  r.sym = NA, F.bot = NA, z.bot = NA, y.bot = NA, e.bot = NA,
  r.bot = NA, F.top = NA, z.top = NA, y.top = NA, e.top = NA,
  r.top = NA)
```

Arguments

x	a network object. This includes all weighted flows into and out of each node.
type	is a parameter to switch the kind of uncertainty analysis to complete: "percent", "sym", "asym". The "percent" options explores the parameter space for all parameters by a fixed percentage. The "sym" options let the user specify an amount to explore around each flow estimate (internal flows (F) and boundary flows (inputs, exports, respirations)). This option assumes that the possible deviation is symmetric around the original values. The "asym" lets the user specify upper and lower limits for each flow value.
iter	is the number of plausible models to sample (number of iterations of the sampling algorithm). The default is 10000, which is often a sufficient sample size for Monte Carlo sampling.
p.err	If the user selects the "percent" type, they must also specify the percent change with this parameter.
F.sym	If the user selects the "sym" type, then this parameter specifies the 1/2 the symmetric parameter range for each internal flow. This should be specified as a data frame in a sparse matrix format with columns identifying the starting node, the target node, and the change value (in same units as flows).
z.sym	If the user selects the "sym" type, then this parameter specifies the 1/2 the symmetric parameter range for each input flow. This is specified as a data frame in a sparse matrix format with columns identifying the node number and the change value (in same units as flows).
y.sym	If the user selects the "sym" type, then this parameter specifies the 1/2 the symmetric parameter range for each output flows. This is specified as a data frame in a sparse matrix format with columns identifying the node number and the change value (in same units as flows).
e.sym	If the user selects the "sym" type, then this parameter specifies the 1/2 the symmetric parameter range for each export flows. This is specified as a data frame in a sparse matrix format with columns identifying the node number and the change value (in same units as flows).
r.sym	If the user selects the "sym" type, then this parameter specifies the 1/2 the symmetric parameter range for each respiration flows. This is specified as a data frame in a sparse matrix format with columns identifying the node number and the change value (in same units as flows).
F.bot	If the user selects the "asym" type, then this data.frame specifies the minimum possible value for each internal flows. This should be specified as a data frame in a sparse matrix format with columns identifying the starting node, the target node, and the change value (in same units as flows).
z.bot	If the user selects the "asym" type, then this data.frame specifies the minimum value for each non-zero model input. This is specified as a data frame in a sparse matrix format with columns identifying the node number and minimum value (in same units as flows).
y.bot	If the user selects the "asym" type, then this data.frame specifies the minimum value for each non-zero model output. This is specified as a data frame in a sparse matrix format with columns identifying the node number and minimum value (in same units as flows).

e.bot	If the user selects the "asym" type, then this data.frame specifies the minimum value for each non-zero model export. This is specified as a data frame in a sparse matrix format with columns identifying the node number and minimum value (in same units as flows).
r.bot	If the user selects the "asym" type, then this data.frame specifies the minimum value for each non-zero model respiration. This is specified as a data frame in a sparse matrix format with columns identifying the node number and minimum value (in same units as flows).
F.top	If the user selects the "asym" type, then this data.frame specifies the maximum possible value for each internal flows. This should be specified as a data frame in a sparse matrix format with columns identifying the starting node, the target node, and the change value (in same units as flows).
z.top	If the user selects the "asym" type, then this data.frame specifies the maximum value for each non-zero model input. This is specified as a data frame in a sparse matrix format with columns identifying the node number and maximum value (in same units as flows).
y.top	If the user selects the "asym" type, then this data.frame specifies the maximum value for each non-zero model output. This is specified as a data frame in a sparse matrix format with columns identifying the node number and maximum value (in same units as flows).
e.top	If the user selects the "asym" type, then this data.frame specifies the maximum value for each non-zero model export. This is specified as a data frame in a sparse matrix format with columns identifying the node number and maximum value (in same units as flows).
r.top	If the user selects the "asym" type, then this data.frame specifies the maximum value for each non-zero model respiration. This is specified as a data frame in a sparse matrix format with columns identifying the node number and maximum value (in same units as flows).

Value

plausible.models

A length=iter list of the plausible models in the network data object format specified for enaR

Author(s)

David E. Hines

References

Hines, D.E., J.A. Lisa, B. Song, C.R. Tobias, S.R. Borrett. 2015. Estimating the impacts of sea level rise on the coupling of estuarine nitrogen cycling processes through comparative network analysis. *Marine Ecology Progress Series* 524: 137-154.

Hines, D.E., Singh, P., Borrett, S.R. 2016. Evaluating control of nutrient flow in an estuarine nitrogen cycle through comparative network analysis. *Ecological Engineering* 89:70-79. doi:10.1016/j.ecoleng.2016.01.009

Examples

```

rm(list = ls())
library(enaR)

# === INPUT ===

# load model for analysis
data(troModels)
m <- troModels[[6]] # cone sping model (Kay et al. 1989; from Tilly)

# Set Uncertainty Analysis parameters
no.samples = 150 # the number of plausible models to return (number of samples);
                # 10,000 would be better.
f.error = 25 # flow parameters percent error to investigate

# === ACTION ===

# perform uncertainty analysis
m.uncertainty.list <- enaUncertainty(m, # original model
                                   type = "percent", # type of uncertainty to use
                                   p.err = f.error, # define percent error
                                   iter = no.samples ) # specify the number of samples

# apply selected ENA
ns <- lapply(m.uncertainty.list, get.ns) # get ENA whole network statistics (metrics, indicators)
ns <- as.data.frame(do.call(rbind, ns))

ns.original <- as.data.frame(get.ns(m))

# === OUTPUT ===

# lets see how the uncertainty in model flows changed the model inputs and total system throughflow.
opar <- par(las = 1, mfcol = c(2,1))
hist(ns$Boundary, col = "steelblue", border = "white", main = "Total Boundary Input")
abline(v = ns.original$Boundary, col = "orange", lwd = 2)
hist(ns$TST, col = "blue2", border = "white", main = "Total System ThroughFLOW")
abline(v = ns.original$TST, col = "orange", lwd = 2)
rm(opar)

# Lets use the 95% CI to make statistitcal inferences about the
# hypothesized "dominance of indirect effects" (Higashi and Patten
# 1991, Salas and Borrett 2010, Borrett et al. 2016), and "network
# homogenization" (Fath and Patten 1999, Borrett and Salas 2010,
# Borrett et al. 2016)

# find 95% confidence intervals
id.95ci <- quantile(ns$ID.F, probs = c(0.025, 0.975))
hmg.95ci <- quantile(ns$HMG.O, probs = c(0.025, 0.975))

# barplot of the calculated values for the original model
opar <- par(las = 1)

```

```

bp <- barplot(c(ns.original$ID.F, ns.original$HMG.0),
              ylim = c(0,3),
              col = "grey",
              border = NA,
              names.arg = c("Indirect/Direct", "Homogenization"))
abline(h = 1, col = "orange", lwd = 1.5) # threshold value

# add 95CI error bars from Uncertainty Analysis
arrows(bp, c(id.95ci[1], hmg.95ci[1]),
        bp, c(id.95ci[2], hmg.95ci[2]),
        code = 3, lwd = 1.5, angle = 90, length = 0.2, col = "black")

# === OUTPUT ===

# lets see how the uncertainty in model flows changed the model inputs and total system throughflow.
opar <- par(las = 1, mfcol = c(2,1))
hist(ns$Boundary, col = "steelblue", border = "white", main = "Total Boundary Input")
abline(v = ns.original$Boundary, col = "orange", lwd = 2)
hist(ns$TST, col = "blue2", border = "white", main = "Total System ThroughFLOW")
abline(v = ns.original$TST, col = "orange", lwd = 2)
rm(opar)

# Lets use the 95% CI to make statistcal inferences about the
# hypothesized "dominance of indirect effects" (Higashi and Patten
# 1991, Salas and Borrett 2010, Borrett et al. 2016), and "network
# homogenization" (Fath and Patten 1999, Borrett and Salas 2010,
# Borrett et al. 2016)

# find 95% confidence intervals
id.95ci <- quantile(ns$ID.F, probs = c(0.025, 0.975))
hmg.95ci <- quantile(ns$HMG.0, probs = c(0.025, 0.975))

opar <- par(las = 1)
bp <- barplot(c(ns.original$ID.F, ns.original$HMG.0),
              ylim = c(0,3),
              col = "grey",
              border = NA,
              names.arg = c("Indirect/Direct", "Homogenization"))
abline(h = 1, col = "orange", lwd = 1.5) # threshold value

# add 95CI error bars from Uncertainty Analysis
arrows(bp, c(id.95ci[1], hmg.95ci[1]),
        bp, c(id.95ci[2], hmg.95ci[2]),
        code = 3, lwd = 1.5, angle = 90, length = 0.2, col = "black")

# The results show that the original value of the Indirect-to-Direct
# flows ratio is larger than one, indicating the "dominance of
# indirect effects"; however, the 95% confidence interval for this
# indicator with a 25% uniform uncertainty spans the threshold value
# of 1 (ranging from 0.9 to 1.16). Thus, we are not confident that
# this parameter exceeds the interpretation threshold given this
# level of uncertainty. In contrast, the network homogenization
# parameter exceeds the interpretation threshold of 1.0, and the 95% CI

```

```
# for our level of uncertainty suggests that we are confident that
# this interpretation is correct.

hist(ns$TST, col = "blue")
```

 enaUtility

Utility Analysis of Ecological Networks

Description

Performs the flow and storage based utility analysis developed for input-output network models of ecosystems. It returns a set of matrices for the direct and integral utilities as well as a set of utility based network statistics.

Usage

```
enaUtility(x, type = c("flow", "storage"), balance.override = FALSE,
  tol = 10)
```

Arguments

x	a network object. This includes all weighted flows into and out of each node. For the storage utility analysis this must also include the amount of energy–matter stored at each node (biomass).
type	Determines whether the flow or storage utility analysis is returned.
balance.override	LOGICAL: should model balancing be ignored. enaUtility assumes that the network model is at steady-state. The default setting will not allow the function to be applied to models not at steady-state. However, when balance.override is set to TRUE, then the function will work regardless.
tol	The integral utility matrix is rounded to the number of digits specified in tol. This approximation eliminates very small numbers introduced due to numerical error in the ginv function. It does not eliminate the small numerical error introduced in larger values, but does truncate the numbers.

Value

D	Direct flow utility intensity matrix. $(f_{ij}-f_{ji})/T_i$ for $i,j=1:n$
U	Nondimensional integral flow utility
Y	Dimensional integral flow utility
ns	If type is set to 'flow', this is a list of flow utility network statistics including: the dominant eigenvalue of D (λ_{1D}), flow based network synergism (synergism.F), and flow based network mutualism (mutualism.F).
DS	Direct storage utility intensity matrix. $(f_{ij}-f_{ji})/x_i$ for $i,j=1:n$

US	Nondimensional integral storage utility
YS	Dimensional integral storage utility
ns	If type is set to 'storage', this is a list of storage utility network statistics including: the dominant eigenvalue of DS (lambda_1DS), storage based network synergism (synergism.S), and storage based network mutualism (mutualism.S).

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Fath, B.D. and Patten, B.C. 1998. Network synergism: emergence of positive relations in ecological systems. *Ecol. Model.* 107:127–143.

Fath, B.D. and Borrett, S.R. 2006. A Matlab function for Network Environ Analysis. *Environ. Model. Soft.* 21: 375–405.

Patten, B.C. 1991. Network ecology: Indirect determination of the life-environment relationship in ecosystems. In: Higashi, M. and Burns, T. (eds). *Theoretical Studies of Ecosystems: The Network Perspective*. Cambridge University Press. New York.

See Also

[enaFlow](#), [enaStorage](#), [enaMTI](#)

environCentrality *Environ Centrality an Ecological Network*

Description

This function calculates the input, output, and average environ centrality of the nodes in the network (Fath and Borret, 2012). This is a type of weighted degree centrality that indicates the relative importance of the nodes in the flow activity in the network.

Usage

`environCentrality(x = "matrix")`

Arguments

x A square matrix. Usually the integral flow marix from enaFlow. The assumption is that the flows are oriented column to row.

Value

ECin input oriented environ centrality
 Ecout output oriented environ centrality
 AEC average environ centrality (average of input and output)

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Fann, S.L. and Borrett, S.R. 2012. Environ centrality reveals the tendency of indirect effects to homogenize the functional importance of species in ecosystems. *Journal of Theoretical Biology* 294: 74-86.

See Also

[enaFlow](#)

Examples

```
data(troModels)
F <- enaFlow(troModels[[6]])
ec <- environCentrality(F$N)
attributes(ec)
barplot(sort(ec$AEC, decreasing = TRUE), col = 4, ylab = "Average Environ Centrality",
        ylim = c(0, 0.4))
```

findPathLength

Cumulative Flow over a Range of Path Lengths

Description

Calculates the flow throughout the entire network over a given path length.

Usage

```
findPathLength(x, maxPath = 100, plot.sw = FALSE)
```

Arguments

x	Network model object.
maxPath	The maximum path length to calculate total flow.
plot.sw	LOGICAL: should a plot be generated showing flow accumulation?

Value

thresholds	thresholds indicating the development of throughflow as path length increases: the path length at which indirect flow exceeds direct flow (mID), path length at which 50%, 90%, and 95% of total system throughflow is achieved (m50, m90, and m95, respectively)
tf	total flow across paths from length 0 (Boundary inputs) to maxPath
ctf	cumulative total flow from path length 0 to maxPath

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Borrett, S.R, Patten, B.C., Whipple, S.J. 2010. Rapid development of indirect effects in ecological networks. *Oikos* 119:1136–1148.

See Also

[enaFlow](#)

Examples

```
data(troModels)
p110 <- findPathLength(troModels[[6]], plot.sw = TRUE, maxPath = 10)
names(p110)
p110$thresholds
```

force.balance

Repeated Application the Balance Function

Description

This function repeatedly balances a model, sequentially with the output being passed back to the balance function, until it is within tolerance or the maximum number of iterations is reached.

Usage

```
force.balance(x, tol = 5, max.itr = 10, method = "AVG2")
```

Arguments

x	A network object.
tol	Percent error tolerance for difference between inputs and outputs.
max.itr	Maximum number iterations.
method	The balancing method to use, see balance. DEFAULT = AVG2.

Value

Returns a balanced network model.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Allesina, S., Bondavalli, C., 2003. Steady state of ecosystem flow networks: a comparison between balancing procedures. *Ecological Modelling* 165(2-3):231-239.

See Also

[balance](#)

Examples

```
data(troModels)
ssCheck(troModels[[1]])
fb.model = force.balance(troModels[[2]]) #produces a balanced model
```

get.ns

Quick Calculation of a Range of Network Statistics.

Description

This is a high level function for calculated the main network analyses (Ascendancy, Flow, Structure, Storage and Utility) on an ecological network.

Usage

```
get.ns(x, balance.override = FALSE)
```

Arguments

x A network object.
balance.override Turns off balancing and balance checking.

Value

Returns the network statistics (ns) of all of the major ENA functions: enaStructure, enaFlow, enaAscendency, enaStorage and enaUtility (both flow and storage).

Author(s)

Matthew K. Lau Stuart R. Borrett David E. Hines

References

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. Environ. Model. Softw. 21, 375-405.

See Also

[enaStructure](#),[enaFlow](#),[enaAscendency](#),[enaUtility](#)

Examples

```
data(troModels)
get.ns(troModels[[6]])
```

get.orient

Returns the global orientation

Description

Returns the current setting for the expected orientation of all matrices, which is either 'rc' (DEFAULT) or 'school' (output orientation as expected for the school of analysis for a given function). This function is intended to provide increase flexibility for users of both the Patten and Ulanowicz schools of ENA.

Usage

```
get.orient()
```

Author(s)

M.K. Lau and S.R. Borrett

m.list

Sub-set of the Larger Ecosystem Models

Description

A set of ecosystem models compiled by the SEE Lab at UNCW.

References

Borrett, S. R., and M. K. Lau. In Prep. enaR: An R package for Ecological Network Analysis. Ecological Modeling and Software.

meanTrophicLevel

meanTrophicLevel INPUT = network data object, min trophic level choice OUTPUT = mean trophic level

Description

=====

Usage

```
meanTrophicLevel(x = "model", minTL = 2)
```

Arguments

x a network object. This includes all weighted flows into and out of each node. The model needs to be at steady-state

minTL is the minimum trophic level to be included in the calculation fo the mean trophic level. The default is 2, but users often use 3.25 and 4.

Author(s)

Stuart R. Borrett

Examples

```
data(enaModels)
meanTrophicLevel(enaModels[[12]], minTL = 3.25)
```

`mExp`*Calculate the exponent of a given matrix*

Description

Function for calculating the pathway proliferation of flows in a network model through matrix exponentiation.

Usage

```
mExp(x = "matrix", n = 2)
```

Arguments

<code>x</code>	A matrix.
<code>n</code>	Desired exponent (i.e. the path length).

Details

This function was originally designed by Alberto Monteiro in the following R help thread: <https://stat.ethz.ch/pipermail/r-help/2007-May/131330.html>.

Value

Returns an exponentiated flow matrix.

Author(s)

Alberto Monteiro (<https://stat.ethz.ch/pipermail/r-help/2007-May/131330.html>) Matthew K. Lau

See Also

[findPathLength](#)

`netOrder`*Reorder Nodes in a Network in enaR*

Description

Reorders nodes in a network either through a user defined node order vector or by default places the non-living nodes to the end of the node vector, minimizing the order change for other nodes.

Usage

```
netOrder(x = "network object", order = 0)
```

Arguments

x	A network object. This includes all weighted flows into and out of each node.
order	An integer vector of length N, where N is number of nodes in x, specifying the new order of the nodes (by default order = 0, which indicates moving non-living nodes to the end)

Details

The node order vector "order" must be of length equal to the number of nodes in x (i.e. N) and must contain all integers from 1 to N. This function can be used with default conditions (i.e. without "order" vector) to reorder the nodes of a network which does not have non-living nodes placed at the end so that the Trophic Aggregations analysis (`enaTroAgg`) can be run on the reordered model.

Value

Returns a network object with nodes ordered as per the node order vector or without the node order vector, by default moves the non-living nodes to the end of the node vector, minimizing the order change for other nodes.

Author(s)

Pawandeep Singh

See Also

[enaTroAgg](#)

Examples

```
data(troModels)
new.network <- netOrder(troModels[[6]], c(1, 3, 2, 5, 4))
# new.network is the required rearranged network with nodes in the desired order.
```

oyster

Intertidal Oyster Reef Ecosystem Model

Description

Intertidal oyster reef ecosystem model created by Dame and Patten (1981). Data were taken from Patten (1985). Model flows are in kcal m⁻² day⁻¹; storage data is kcal m⁻².

References

- Dame, R. F., and B. C. Patten. 1981. Analysis of energy flows in an intertidal oyster reef. *Marine Ecology Progress Series* 5:115-124.
- Patten, B. C. 1985. Energy cycling, length of food chains, and direct versus indirect effects in ecosystems. *Can. Bull. Fish. Aqu. Sci.* 213:119-138.

pack

Compile Network Information into a Network Class

Description

This function provides a flexible framework for importing flow network information into a network class object for analyses.

Usage

```
pack(flow, input = NA, respiration = NA, export = NA, output = NA,  
      storage = NA, living = NA)
```

Arguments

flow	The flow matrix.
input	The inputs into the system.
respiration	The quantities respired from the system.
export	The exports from the system.
output	The output (i.e. exports + respiration) from the system.
storage	The quantities stored in compartments within the system.
living	A logical vector indicating whether a node is either 'living' (= TRUE) or 'dead' (=FALSE).

Value

Returns a network object for the supplied model.

Author(s)

Matthew K. Lau Stuart R. Borrett

See Also

[unpack](#)

read.EcoNet	<i>Read an EcoNet model.</i>
-------------	------------------------------

Description

This function allows the user to access models that are formatted for EcoNet, the web-based interface for conducting ENA (<http://eco.engr.uga.edu/>), by Caner Kazanci at the University of Georgia.

Usage

```
read.EcoNet(file = "file path", verbose = FALSE, parse = FALSE)
```

Arguments

file	Path to an EcoNet formatted file
verbose	LOGICAL: should warnings be suppressed?
parse	Is the file input parsed from a webpage?

Value

Returns the model formatted as a network object.

Author(s)

Matthew K. Lau

References

Kazanci, C., 2007. EcoNet: A new software for ecological modeling, simulation and network analysis, *Ecol. Model.*, Vol 208/1 pp 3-8.

See Also

[EcoNetWeb](#)

read.enam	<i>R function to read in a matrix formatted as Mdloti (Ursula Sharler)</i>
-----------	--

Description

This function reads network data from an excel file commonly used by Ursula Sharler. The file has three header lines (name/source, number of compartments, number of living nodes) and then a $n+2 \times n+2$ matrix of flows. This is the flow matrix with an additional row for imports and biomass each and additional columns for exports and respirations.

Usage

```
read.enam(file = "file path and name")
```

Arguments

file The name and path for the data file. This function assumes the data are stored on the first sheet of an Microsoft Excel formatted. NOTE: this function depends on the read.xlsx function from the xlsx package, which requires that the entire path be specified from the root directory (i.e. the absolute path).

Value

Returns the network object.

Author(s)

Stuart R. Borrett

References

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. Environ. Model. Softw. 21, 375-405.

See Also

[read.scor](#)

read.nea

Read an NEA formatted model into a network object

Description

This function reads in and creates a network object from a NEA formatted data file (Fath and Borrett 2006).

Usage

```
read.nea(file = "file name", sep = ",", warn = TRUE)
```

Arguments

file The name and path for the data file.
sep The separation character used to delimit data values.
warn LOGICAL: should pack warnings be reported?

Value

Returns the network object.

Author(s)

Stuart R. Borrett

References

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. Environ. Model. Softw. 21, 375-405.

See Also

[write.nea](#)

read.scor

SCOR formatted file into R in multiple formats INPUT = file path

Description

Read in network model data files that are in the SCOR format (REFERENCE).

Usage

```
read.scor(file, from.file = TRUE, warn = FALSE)
```

Arguments

file	File path or plain text.
from.file	States whether the file argument input should be treated as a file path (TRUE) or plain text (FALSE).
warn	Turn on (TRUE) or off (FALSE) warnings.

Details

The SCOR file must be formatted properly. In particular, the number of nodes on the second line must have the first three characters dedicated to the total number of nodes and the next three characters should contain the number of living nodes. That is, the second line of the file should be formatted as 'xxxxyy' where x and y are the characters for the total number of nodes and the number of living nodes, respectively. Thus, if the total number of nodes is 10 and the number of living nodes is 1, then the second line should read, " 10 1."

Value

Returns the network model in one of several formats. The default format is a network object used by the statnet package (type="network"). Three other options are the network environ analysis format (type="nea") as defined by (Fath and Borrett 2006), a list format (type="list") and an edge list (type="edge.list").

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Ulanowicz, R.E. and J.J. Kay. 1991. A package for the analysis of ecosystem flow networks. *Environmental Software* 6:131-142.

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. *Environ. Model. Softw.* 21, 375-405.

read.wand

WAND formatted file into R

Description

Reads WAND formatted network models.

Usage

```
read.wand(file = "file name with path")
```

Arguments

file File path to WAND formatted data file.

Value

Returns a network object from a WAND formatted data file.

Note

IMPORTANT: this function depends on the read.xlsx function from the xlsx package, which requires that the entire path be specified from the root directory (i.e. the absolute path).

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Allesina, S., Bondavalli, C., 2004. WAND: an Ecological Network Analysis user-friendly tool. *Environmental Modelling and Software* 19(4):337-340.

relationalChange *Relational change compared between two matrices.*

Description

Identifies the signs and pairwise relationships of two matrices and compares the difference between them.

Usage

```
relationalChange(x = "Direct.U", y = "Integral.U")
```

Arguments

x x is a square matrix of real numbers. While this function is more general, the initial intention was for this to be the direct utility matrix.

y y is a square matrix of real numbers. While this function is more general, the initial intention was for this to be the integral utility matrix or the mixed trophic impacts matrix.

Value

Direct.Signs A sign matrix for matrix x.

Integral.Signs A sign matrix for matrix x.

Direct.Relations
 A matrix of the pairwise sign relationships for matrix x.

Integral.Relations
 A matrix of the pairwise signed relationships in matrix y.

Relations.Table
 A table that summarizes the relations.

Changed.Table A summary table of only the pairwise relationships that changed between x and y.

ns A vector of network statistics which currently includes one whole-network statistic - a ratio of the relationships changed between x and y.

Note

This function is called by enaUtility and enaMTI to summarize results.

Author(s)

Stuart R. Borrett

See Also

[enaUtility](#), [enaMTI](#), [signs](#)

scc *Find the strongly connected component*

Description

This function finds the strongly connected components (SCCs) of an adjacency matrix A and returns a number of derived network statistics.

Usage

```
scc(A = "adjacency")
```

Arguments

A an n x n adjacency matrix.

Value

sp a list of structural properties including: the number of SCCs ("no.scc"), the number of SCCs with more than 1 node ("no.scc.big"), and the fraction of the network nodes participating in a large SCC ("pssc")

membership numeric vector giving the cluster id to which each node belongs (as in `igraph:clusters`)

scc.id numeric vector of the numeric identity in "membership" of SCCs with more than 1 node

Note

Input matrix is assumed to be oriented from columns to rows.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Allesina, S., Bodini, A., Bondavalli, C., 2005. Ecological subsystems via graph theory: the role of strongly connected components. *Oikos* 110, 164-176.

Berman, A., Plemmons, R.J., 1979. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York.

Borrett, S.R., Fath, B.D., Patten, B.C. 2007. Functional integration of ecological networks through pathway proliferation. *Journal of Theoretical Biology* 245, 98-111.

See Also

[enaStructure](#)

Examples

```
data(troModels)
A <- enaStructure(troModels[[6]])$A
scc(A)
```

scifix

scifix Corrects missing e or E in scientific notation

Description

This is a support function that corrects the scientific notation in SCOR formatted data files.

Usage

```
scifix(x)
```

Arguments

x A numeric or character scalar.

Value

Returns a numeric scalar in appropriate scientific notation.

Author(s)

Matthew K. Lau

See Also

[read.scor](#)

set.orient	<i>Globally reorients matrices</i>
------------	------------------------------------

Description

Changes the orientation of output matrices.

Usage

```
set.orient(x = c("rc", "school"))
```

Arguments

x The school determining the orientation

Details

The enaR package as a whole, and the broader network analysis community, assumes a row to column orientation; thus, the default orientation for the package is row to column (DEFAULT = 'rc'). However, functions from the Patten school were originally developed to conduct calculations and produce output in the column to row orientation. In order to facilitate the use of these functions, we also provide the option for users to return output in the orientation of the "school" (i.e. Patten results will be column to row oriented) by setting the global orientation to "school" using this function. Orientation setting. If "rc" (DEFAULT), all matrix output will be returned in row (=input) to column (=output) orientation, regardless of school. If "school", then output matrices from functions from particular ENA schools will be oriented as expected in that school (i.e. Patten = column-row or Ulanowicz = row-column). Note, that all functions in the enaR package expect input matrices to be oriented row-column.

Author(s)

Matthew K. Lau Stuart R. Borrett

See Also

[get.orient](#)

Examples

```
original.orientation = get.orient()
original.orientation
set.orient("school")
get.orient()
set.orient("rc")
get.orient()
set.orient(original.orientation)
```

ShannonDiversity *ShannonDiversity Shannon Diversity Metrics*

Description

Calculates a number of metrics based on the Shannon information entropy measure of diversity in a vector, *x*.

Usage

```
ShannonDiversity(x)
```

Arguments

x 1 x n vector.

Value

<i>H</i>	Shannon entropy-based metric of diversity. This captures the effects of both richness (the length of the vector, <i>n</i>) and the evenness of the distribution.
<i>Hmax</i>	The maximum possible value of <i>H</i> given a vector of the length <i>n</i> provided.
<i>Hr</i>	Relative evenness $H_r = H/H_{max}$
<i>Hcentral</i>	The centralization or concentration of the values among the <i>n</i> elements
<i>n</i>	Number of elements in the vector.
<i>effective.n</i>	effective number of elements in the vector, given the distribution of the relative weights.

Note

The formulation for Shannon Diversity uses a natural logarithm. As the natural logarithm of zero is undefined, the input vector cannot contain zeros. Analytically, there are two approaches to dealing with this issue if your vector contains zeros. First, you can apply the analysis to only the non-zero elements. Second, you can add a tiny amount to all of the elements such that the zero elements are now very small numbers, relative the original vector values.

Author(s)

Stuart R. Borrett

Examples

```
data(oyster)

## throughflow diversity
T <- enaFlow(oyster)$T
ShannonDiversity(T)
```

```
## storage (biomass) biodiversity
## X <- oyster %V% "storage"
## ShannonDiversity(X)
```

 signs

Signs and summary of input matrix

Description

Identifies the signs and pairwise relationships of a given matrix. This includes also returns a summary table that provides the ecological name of each pairwise relationship, and a summary of the counts.

Usage

```
signs(x = "matrix")
```

Arguments

x a square matrix of real numbers. While this function is more general, the initial intention was for this to be a utility matrix or the mixed trophic impacts matrix.

Value

sign A sign matrix for matrix x.
relations A matrix of the pairwise signed relationships in x.
rs.tab Table summarizing the pairwise relationships and identifying their ecological label.
relationship.counts A count of the different kinds of pairwise relationships found in matrix x.

Note

This function is called by `relationalChange`, and was created to generate more informative output from `enaUtility` and `enaMTI`.

Author(s)

Stuart R. Borrett

See Also

[relationalChange](#)

ssCheck	<i>Checks if the given network is out of balance by a given tolerance threshold</i>
---------	---

Description

This function supports the balancing process by checking if the inputs and outputs of a given network model are within acceptable limits.

Usage

```
ssCheck(x, tol = 5, more = FALSE, zero.na = TRUE)
```

Arguments

x	A network object.
tol	The threshold for balance in percent difference between input and outputs.
more	LOGICAL: should more detailed results be returned?
zero.na	LOGICAL: should NA values be changed to zeros?

Value

Returns a logical value stating if the model is within acceptable limits of balance (TRUE) or if it is not (FALSE).

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Fath, B.D. and S.R. Borrett. 2006. A MATLAB function for network environ analysis. Environmental Modelling & Software 21:375-405.

See Also

[balance](#)

Examples

```
data(troModels)
ssCheck(troModels[[2]])
ssCheck(troModels[[6]])
```

structure.statistics *structural statistics*

Description

This function returns several network statistics that describe a network.

Usage

```
structure.statistics(A = "adjacency matrix")
```

Arguments

A An adjacency matrix.

Value

n Number of nodes in A.
L Number of direct connections in A.
C Connectivity of A.
LD Link density.
lam1A First dominant eigenvalue of A.
mlam1A Multiplicity of the dominant eigenvalue.
lam2A Magnitude of the second largest eigenvalue.
rho Damping ratio (see Caswell 2001).
R Distance of lam1A from the bulk of the eigen spectrum.
d Difference between the dominant eigenvalue and the link density.
no.scc Number of strongly connected components.
no.scc.big Number of strongly connected components greater than 1.
pscc Percent of nodes participating in a strongly connected component.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. Environ. Model. Softw. 21, 375-405.

See Also

[enaStructure,scc](#)

TES*Calculates the Total Environ Storage*

Description

Calculates the total storage in each n input and output environs. This function calculates the storage for both the unit input (output) and the realized input (output) environs. Realized uses the observed inputs (outputs) rather than an assumed unit input (output) to each node.

Usage

```
TES(x, balance.override = FALSE)
```

Arguments

x A network object.
balance.override LOGICAL: should balancing being ignored.

Value

realized.input input oriented, realized storage in each environ.
realized.output output oriented, realized storage in each environ.
unit.input input oriented, unit storage in each environ.
unit.output input oriented, unit storage in each environ.

Author(s)

Matthew K. Lau Stuart R. Borrett David E. Hines

References

Matis, J.H. and Patten, B.C. 1981. Environ analysis of linear compartmental systems: the static, time invariant case. Bulletin of the International Statistical Institute. 48, 527–565.

See Also

[enaStorage](#), [enaEnviron](#)

Examples

```
data(troModels)  
tes <- TES(troModels[[6]])  
tes
```

TET *Calculate the Total Environ Throughflow*

Description

Determines the total environ throughflow (TET) for each of the 2 x n environs of the selected network model. It returns both the TET calculated from a unit input (output) vector and from the observed or realized input (output) vector.

Usage

```
TET(x, balance.override = FALSE)
```

Arguments

x A network object.
balance.override Logical: should the function work if the model is not at steady-state?

Value

realized.input vector of the n realized total environ throughflows for the n input oriented environs.
realzied.output vector of the n realized total environ throughflows for the n ouput oriented environs.
unit.input vector of the n unit total environ throughflows for the n input oriented environs.
unit.output vector of the n unit total environ throughflows for the n output oriented environs.

Author(s)

Matthew K. Lau Stuart R. Borrett

References

Gattie, D.K., Schramski, J.R., Borrett, S.R., Patten, B.C., Bata, S.A., and Whipple, S.J. 2006. Indirect effects and distributed control in ecosystems: Network environ analysis of a seven-compartment model of nitrogen flow in the Neuse River Estuary, USA—Steady-state analysis. *Ecol. Model.* 194:162–177.

Whipple, S.J., Borrett, S.R., Patten, B.C., Gattie, D.K., Schramski, J.R., and Bata, S.A. 2007. Indirect effects and distributed control in ecosystems: Comparative network environ analysis of a seven-compartment model of nitrogen flow in the Neuse River Estuary, USA—Time series analysis. *Ecol. Model.* 206: 1–17.

See Also

[enaEnviron](#)

Examples

```
data(troModels)
tet <- TET(troModels[[6]])
tet
```

troModels	<i>Trophic Models</i>
-----------	-----------------------

Description

A set of 58 trophic models compiled by the SEE Lab at UNCW.

References

Borrett, S. R., and M. K. Lau. In Prep. enaR: An R package for Ecological Network Analysis. Ecological Modeling and Software.

unpack	<i>Extracts network object into a list</i>
--------	--

Description

Separates the components of a network object into separate components within a list. This includes inputs, exports, respirations, outputs (exports + respirations), storage, and internal flows.

Usage

```
unpack(x = "network object")
```

Arguments

x A network object. This includes all weighted flows into and out of each node.

Details

Flows are oriented from row to column.

Value

F	matrix of flows from each node to each node oriented row to column.
z	Node boundary inputs.
r	Node boundary loss from respiration.
e	Node boundary loss due to exportation
y	Node boundary loss; summation of r and e
X	Node storage or biomass
living	Logical vector indicating whether each node is living or not

Author(s)

Matthew K. Lau Stuart R. Borrett

See Also

[pack](#), [read.scor](#)

Examples

```
data(troModels)
unpack(troModels[[6]])
```

write.EcoNet

Write enaR models to an EcoNet formatted file.

Description

Creates an EcoNet model from an enaR network object that can be used with the online interface for EcoNet.

Usage

```
write.EcoNet(x = "model", file = "file path", mn = "ena_model",
             zero.flows = FALSE)
```

Arguments

x	Network object.
file	The file name or path. If a simple file name is given, this function uses the current working directory by default.
mn	The model name that EcoNet will use. The DEFAULT is 'ena_model'.
zero.flows	LOGICAL: should zero flow values be written?

Value

An EcoNet formatted text file is created from the model, which can be input at <http://eco.engr.uga.edu>.

Author(s)

Matthew K. Lau

References

About EcoNet (<http://eco.engr.uga.edu/DOC/econet1.html>) Kazanci, C. 2009. Handbook of Ecological Modelling and Informatics, by WIT Press.

write.nea	<i>Create a formatted file with data arranged as expected input for NEA.m</i>
-----------	---

Description

This function writes a network object to a NEA formatted data file (Fath and Borrett 2006).

Usage

```
write.nea(x, file.name, sep = ",")
```

Arguments

x	Network object.
file.name	The file name or path. If a simple file name is given, this function uses the current working directory by default.
sep	The separation character used to delimit data values.

Value

Writes a network object to a NEA formatted file and returns the output composite matrix.

Author(s)

Stuart R. Borrett

References

Fath, B. D., Borrett, S. R. 2006. A Matlab function for Network Environ Analysis. Environ. Model. Softw. 21, 375-405.

See Also

[read.nea](#)

Index

- *Topic **datasets**
 - bgcModels, 7
 - enaModelInfo, 20
 - enaModels, 20
 - m.list, 38
 - oyster, 40
 - troModels, 56
- *Topic **enaFlow**
 - enaStorage, 22
- *Topic **package**
 - enaR-package, 3
- *Topic **read.scor**
 - enaStorage, 22
- as.bipartite, 3
- as.extended, 4
- bal, 5, 6
- balance, 5, 6, 6, 36, 52
- bcratio, 7
- bgcModels, 7
- cycliv, 8
- EcoNetWeb, 9, 42
- eigenCentrality, 10
- enaAll, 11
- enaAscendency, 11, 12, 37
- enaControl, 11, 14
- enaCycle, 9, 15, 27
- enaEnviron, 11, 17, 54, 55
- enaFlow, 11, 18, 22, 24, 33–35, 37
- enaModelInfo, 20
- enaModels, 20
- enaMTI, 11, 21, 33, 46
- enaR (enaR-package), 3
- enaR-package, 3
- enaStorage, 11, 13, 15, 19, 22, 33, 54
- enaStructure, 24, 37, 47, 53
- enaTroAgg, 9, 16, 25, 40
- enaUncertainty, 27
- enaUtility, 7, 11, 13, 19, 22, 24, 32, 37, 46
- environCentrality, 33
- findPathLength, 34, 39
- force.balance, 35
- get.ns, 36
- get.orient, 37, 49
- m.list, 38
- meanTrophicLevel, 38
- mExp, 39
- netOrder, 9, 25, 27, 39
- network, 3
- oyster, 40
- pack, 41, 57
- read.EcoNet, 10, 42
- read.enam, 42
- read.nea, 43, 58
- read.scor, 13, 19, 24, 43, 44, 48, 57
- read.wand, 13, 19, 24, 45
- relationalChange, 46, 51
- scc, 47, 53
- scifix, 48
- set.orient, 49
- ShannonDiversity, 50
- signs, 46, 51
- ssCheck, 52
- structure.statistics, 25, 53
- TES, 54
- TET, 55
- troModels, 56
- unpack, 41, 56
- write.EcoNet, 57
- write.nea, 44, 58