Package 'enaR'

October 28, 2013

Type Package

Title Tools for ecological network analysis (ena) in R

Version 2.5

Date 2013-10-25

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

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

Description Functions for the analysis of ecological networks

Depends R (>= 2.10), sna, network, MASS, stringr, xlsx

Suggests codetools, igraph

License GPL-2

R topics documented:

---

enaR-package *Tools for ecological network analysis (ena) in R.*

Description

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

Details

Package:	enaR
Type:	Package
Version:	2.5
Date:	2013-10-25
License:	GPL-2

Author(s)

Authors: Stuart R. Borrett (UNCW) and Matthew K. Lau (NAU) Maintainer: Matthew K. Lau <enaR.maintainer@gmail.com>

References

Fath BD, Borrett SR. 2006. A Matlab Function for Network Environ Analysis. Environmental Modelling and Software 21, 375-405.

See Also

`network`

`as.extended` — *Create an extended format matrix.*

Description

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

Usage

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

Arguments

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

Details

Used in the balance function.

Value

Returns an extended format matrix.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 *Subfunction for balancing by either inputs or outputs.*

Description

Dependency for the `balance` function.

Usage

```
bal(T.star = "extended, unbalanced 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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

`x`	A network object.
`method`	Methods for model balancing, see Allesina and Bondavalli (2003).
`tol`	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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 (borretts@uncw.edu)

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 *Bigeochemical Cycling Models*

Description

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

Usage

```
data(bgcModels)
```

References

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

eigenCentrality *Calculates the eigen centrality of a network.*

Description

Calculates the eigen centrality of a network.

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 (borretts@uncw.edu) Matthew K. Lau (mkl48@nau.edu)

References

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

enaAll *Conduct all major ENA.*

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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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.

Value

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)
ASC	Returns the ascendnecy 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.

Note

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

Author(s)

David E. Hines (deh9951@uncw.edu) Matthew K. Lau (mkl48@nau.edu)

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 Serries. Springer-Verlag, Berlin.

Patrico, 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., 1997. Ecology, The Ascendent Perspective. Columbia University Press, New York.

Ulanowicz, R.E., 2004. Quantitative methods for ecological network analysis. Comput. Biol. Chem. 28, 321-339

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 of 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.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

References

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

See Also

`enaStorage`

Examples

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

enaEnviron *Ecological network environs.*

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 (borretts@uncw.edu) Matthew K. Lau (mkl48@nau.edu)

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` *Flow 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 ouput oriented integral (boundary+direct+indirect) flow intensities

`NP` matrix of the input oriented integral flow intensities

`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 amplifiation, "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 eqivalent to mode2, and "mode4.F" is the dissipative equivalent ot mode0.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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.

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 statisics
```

enaMTI *Mixed Trophic Impacts (MTI) measures the impact of one species on another.*

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,eigen.check=TRUE,zero.na=TRUE, balance.override=FALSE)
```

Arguments

`x` a network object. This includes all weighte dflows 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.

`eigen.check` LOGICAL: should the dominant eigen value be checked? By default, the function will not return utility values if the eigenvalue is larger than one; however, if eigen.check is set to FALSE, then the function will be applied regardless of the mathematic concern.

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) tropic impacts of compartment i on j.

Note

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.

With regard to the eigen.check argument, like enaFlow, enaStorage and enaUtility, this analysis considers the trophic impact propigated over path lengths ranging for zero to infinity. For the analysis to work properly, the path sequence must converge. This function checks to see if the path sequence is convergent by finding the dominant eigenvalue of the direct matrix. If this eigenvalue is less than 1, the sequence is convergent and the analysis can be applied; if the dominant eigenvalue is greater than one, then the anlysis cannot be applied.

Author(s)

Stuart R. Borrett (borretts@uncw.edu) Matthew K. Lau (mkl48@nau.edu)

References

Ulanowicz, 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

`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

`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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

References

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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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]])
```

`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"),eigen.check=TRUE,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.

`eigen.check` LOGICAL: should the dominant eigenvalue be checked. Like enaFlow and enaStorage analyses, enaUtility analysis considers the utility propigated over path lengths ranging for zero to infinity. For utility analysis to work properly, the path sequence must converge. enaUtility checks to see if the utility path sequence is convergent by finding the dominant eigenvalue of the direct utility matrix. If this eigenvalue is less than 1, the sequence is convergent and the analysis can be applied; if the dominant eigenvalue is greater than one, then the anlysis cannot be applied. By default, the function will not return utility values if the eigenvalue is larger than one; however, if eigen.check is set to FALSE, then the function will be applied regardless of the mathematic validity.

`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 eleminates 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. (fij-fji)/Ti 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 (lambda_1D), flow based network synergism (synergism.F), and flow based network mutualism (mutualism.F).

`DS` Direct storage utility intensity matrix. (fij-fji)/xi 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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`

Examples

```
data(troModels)
U <- enaUtility(troModels[[6]],type="flow",eigen.check=FALSE)
attributes(U)
US <-enaUtility(troModels[[6]],type="storage",eigen.check=FALSE)
```

environCentrality — *Calculates the environ centrality of the nodes in 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 centraility
`AEC`	average environ centrality (average of input and output)

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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)
pl10 <- findPathLength(troModels[[6]], plot.sw=TRUE,maxPath=10)
names(pl10)
pl10$thresholds
```

`force.balance` *Repeated, sequential 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu) David E. Hines (deh9951@uncw.edu)

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 *Get the current global matrix orientation setting.*

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).

Usage

```
get.orient()
```

Details

This function is intended to provide increase flexibility for users of both the Patten and Ulanowicz schools of ENA.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

mExp *Calculates the exponent of a matrix.*

Description

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

Usage

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

Arguments

x	A matrix.
n	Desired exponent (i.e. the path length).

Value

Returns an exponentiated flow matrix.

Author(s)

Alberto Monteiro (https://stat.ethz.ch/pipermail/r-help/2007-May/131330.html) Matthew K. Lau (mkl48@nau.edu)

References

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.

See Also

findPathLength

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.

Usage

```
data(oyster)
```

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,output=NA,export=NA,respiration=NA,storage=NA,living=NA)
```

Arguments

flow	The flow matrix.
input	The inputs into the system.
output	The outputs from the system.
export	The exports from the system.
respiration	The quantities respired 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

See Also

`unpack`

read.enam — *Reads in ENA data from an excel file, formatted as Mdloti.xls*

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 x 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 (borretts@uncw.edu)

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 NEA formatted network data files.*

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 (borretts@uncw.edu)

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 *Read SCOR formatted data files.*

Description

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

Usage

```
read.scor(file,type=c("network","nea","list","edge.list"),from.file=TRUE,warn = FALSE)
```

Arguments

`file`	File path or plain text.
`type`	Specifies the format of the object created.
`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 'xxxyyy' 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 *Read WAND format files.*

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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

References

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

`scc` *Finds the strongly connected component (SCC) in a graph.*

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 ("pscc")
`membership`	numeric vector giving the cluseter 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 *Standardizes 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 (mkl48@nau.edu)

See Also

read.scor

set.orient *Globally set the output matrix orientation.*

Description

Changes the orientation of output matrices.

Usage

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

Arguments

x	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.

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 orignially 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 fuction.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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)
```

ssCheck *Checks the balance of inputs and outputs from a network.*

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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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` *Calculates the structural statistics (network statistics) of an ecological network.*

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`	Precent of nodes participating in a strongly connected component.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 *Calculate 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu) David E. Hines (deh9951@uncw.edu)

References

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

See Also

`enaStorage`,`enaEnviron`

Examples

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

TET — *Calculates the total environ throughflow for a ecosystem network model.*

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 ouptut 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 (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

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 *Trophic Models*

Description

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

Usage

```
data(troModels)
```

References

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

unpack *"Unpacks" the components of a network object into separate objects.*

Description

"Unpacks" the components of a network object into separate objects. 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.

Value

`F`	matrix of flows from each node to each node oreinted 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

Note

Flows are oriented from row to column.

Author(s)

Matthew K. Lau (mkl48@nau.edu) Stuart R. Borrett (borretts@uncw.edu)

See Also

`pack`,`read.scor`

Examples

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

`write.nea` *Write a network object to file using the NEA data format.*

Description

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

Usage

```
write.nea(x, file.name='neaData.csv',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 boject to a NEA formatted file and returns the output composite matrix.

Author(s)

Stuart R. Borrett (borretts@uncw.edu)

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