Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

Ecosystem Network Analysis with R: A guide for using enaR

Matthew K. Lau
Harvard Forest
Harvard University

Stuart R. Borrett
Department of Biology and Marine Biology
University of North Carolina Wilmington
and
Duke Network Analysis Center
Social Science Research Institute
Duke University

Pawandeep Singh
Department of Biology and Marine Biology
University of North Carolina Wilmington

Abstract

Ecosystem Network Analysis (ENA) provides a framework for investigating the structure, function and dynamics of ecological systems, primarily ecosystem models with physically conserved units. This paper documents the *enaR* R package that collects the core ENA functions including those developed by the Ulanowicz and Patten schools. We detail how to use the primary functions for the analysis of single models as well as simultaneous, synthetic analysis of multiple ecosystem models.

Keywords: ecology, ENA, ecosystems, network ecology, food web, network environ analysis, species interactions, networks, R.

1. Introduction

Network models have provided an in-road to a variety of complex systems (Watts and Strogatz,

1998; Newman, 2001; Barabási, 2012; Newman, Barabási, and Watts, 2006; Wasserman and Faust, 1994), and although the network approach has deep roots (Newman *et al.*, 2006), its use has been expanding rapidly in a variety of disciplines including ecology (Borrett, Moody, and Edelmann, 2014; Ings, Montoya, Bascompte, Blüthgen, Brown, Dormann, Edwards, Figueroa, Jacob, Jones, Lauridsen, Ledger, Lewis, Olesen, van Veen, Warren, and Woodward, 2009); and investigators are currently building a science of networks (National Research Council, Committee on Network Science for Army Applications, 2006; Brandes, Robins, McCranie, and Wasserman, 2013). This is due in part to the flexibility of the core representation, its utility in answering relational questions, and its applicability to "Big Data" problems.

Ecosystem ecologists have developed and used network modeling and analysis for several decades (Borrett, Christian, and Ulanowicz, 2012; Ulanowicz, 1986; Fath and Patten, 1999). The network models map transfers of thermodynamically conserved energy or matter (represented by weighted, directed graph edges) between nodes that represent species, groups of species, or non-living components (e.g., dead organic matter) of the ecosystem. These analyses, collectively known as Ecosystem Network Analysis (ENA), have been used in a variety of ways including to reveal the relative importance of indirect effects in ecosystems (Patten, 1983; Higashi and Patten, 1989; Salas and Borrett, 2011) and their capacity to effectively transform the relations among organisms (Ulanowicz and Puccia, 1990; Patten, 1991; Fath and Patten, 1998; Bondavalli and Ulanowicz, 1999). From these applications a new theoretical understanding of ecosystems has emerged (Higashi and Burns, 1991; Belgrano, Scharler, Dunne, and Ulanowicz, 2005; Jørgensen, Fath, Bastianoni, Marques, Müller, Nielsen, Patten, Tiezzi, and Ulanowicz, 2007). Recently, scientists have applied these methods to understand trophic dynamics in the Sylt-Rømø Bight (Baird, Asmus, and Asmus, 2004a, 2008), biogeochemical cycling in lakes and estuaries (Christian and Thomas, 2003; Small, Sterner, and Finlay, 2014; Hines, Lisa, Song, Tobias, and Borrett, 2015), and urban sustainability (Zhang, Yang, and Fath, 2010; Chen and Chen, 2012).

Two major schools of ENA have developed (Scharler and Fath, 2009). The first is based on Dr. Robert E. Ulanowicz's work with a strong focus on trophic dynamics and a use of information theory (Ulanowicz, 1986, 1997, 2004). The second school has an environment focus and is built on the environ concept introduced by Dr. Bernard C. Patten (Patten, Bosserman, Finn, and Cale, 1976; Patten, 1978; Fath and Patten, 1999). Patten's approach has been collectively referred to separately as *Network Environ Analysis*. At the core the two approaches are very similar; however, they make some different starting assumptions and follow independent yet braided development tracks.

Disparate software packages have been created to support ENA. Initially algorithms were developed and distributed as the DOS based NETWRK4 (Ulanowicz and Kay, 1991), which is still available from `http://www.cbl.umces.edu/~ulan/ntwk/network.html`. Some of these algorithms were re-implemented in a Microsoft Excel based toolbox, WAND (Allesina and Bondavalli, 2004). The popular Ecopath with Ecosim software that assists with model construction (Christensen and Walters, 2004) also provides multiple ENA algorithms. The algorithms for flow analysis – one component of ENA – were collected into a stand-alone software tool (Latham II, 2006). Fath and Borrett (2006) published NEA.m that collects most of the Patten School ENA algorithms together in a single MATLAB© function. Similarly, the online tool EcoNet (Kazanci, 2007) has made many of the ENA algorithms available in an easy access framework. Although these packages collectively provide access to a large set of powerful analytical tools, the fragmented distribution of the key algorithms among the software

tools has inhibited the development of theory and the further implementation of important algorithms.

The *enaR* package brings together the ENA algorithms into one common software framework that is readily available and extensible. The package is written in the R language, which is free and open-source. Due largely to this, R is now one of the most widely used analytical programming languages in the biological sciences. *enaR* builds on existing R packages for network analysis. For example, it uses the *network* data structure developed by Butts (2008a) and the network analysis tools built into the *network*, *sna* (social network analysis) (Butts, 2008b), and *statnet* (Handcock, Hunter, Butts, Goodreau, and Morris, 2008) packages. While Borrett and Lau (2014) introduced the *enaR* package, here we provide a richer documentation of the software and illustrate its use.

2. Getting Started

In this section we describe the data necessary for Ecosystem Network Analysis and show how to build the central network data object in R that contains the model data for subsequent analysis. To start, the current stable version can be installed from CRAN:

```
> install.packages('enaR')
```

The beta version can be installed from GitHub:

```
> library(devtools)
> install_github('SEELab/enaR',ref='beta')
```

You can now load the package:

```
> library(enaR)
```

2.1. Ecosystem Network Model

ENA is applied to a network model of energy–matter exchanges among system components. The system is modeled as a set of n compartments or nodes that represent species, species-complexes (i.e., trophic guilds or functional groups), or non-living components of the system in which energy–matter is stored. Nodes are connected by L observed fluxes, termed directed edges or links. This analysis requires an estimate of the energy–matter flowing from node i to j over a given period, $\mathbf{F}_{n\times n} = [f_{ij}]$, $i, j = 1, 2, \ldots, n$. These fluxes can be generated by any process such as feeding (like a food web), excretion, and death. As ecosystems are thermodynamically open, there must also be energy or matter inputs into the system $\mathbf{z}_{1\times n} = [z_i]$, and output losses from the system $\mathbf{y}_{1\times n} = [y_i]$. While the Patten School treats all outputs the same, the Ulanowicz School typically partitions outputs into respiration $\mathbf{r}_{1\times n} = [r_i]$ and export $\mathbf{e}_{1\times n} = [e_i]$ to account for differences in energetic quality. Note that $y_i = r_i + e_i, \forall i$. Some analyses also require the amount of energy–matter stored in each node (e.g., biomass), $\mathbf{X}_{1\times n} = [x_i]$. The final required information is a categorization of each node as living or not, which is essential for algorithms from the Ulanowicz School. For our implementation, we have

created a logical vector $\mathbf{Living}_{1\times n}$ that indicates whether the i^{th} node is living (TRUE) or not (FALSE). This obviates the need to order the nodes in a specific way (i.e., living before non-living). Together, the model data $\mathcal{M}$ can be summarized as $\mathcal{M} = \{\mathbf{F}, \mathbf{z}, \mathbf{e}, \mathbf{r}, \mathbf{X}, \mathbf{Living}\}$.

Notice the row-to-column orientation of the flow matrix: **F**. This is consistent with the Ulanowicz School of network analysis, as well as the orientation commonly used in Social Network Analysis and used in the *statnet* packages. However, this is the opposite orientation typically used in the Patten School of analysis that conceptually builds from a system of differential equations and thus uses the column-to-row orientation common in this area of mathematics. Even though the difference is only a matrix transpose, this single difference may be the source of much confusion in the literature and frustration on the part of users. We have selected to use row-to-column orientation for our primary data structure, as it is the dominant form across network analytics as evidenced by it use in the *statnet* packages. The package algorithms also return the results in the row-to-column orientation by default; however, we have built in functionality with `get.orient` and `set.orient` that allows users to return output in the Patten School row-to-column orientation (see Section 3.13 for details).

There are multiple methods for constructing ecosystem network models and tools for assisting with this process (Fath, Scharler, Ulanowicz, and Hannon, 2007). One approach is to construct a dynamic, processes-based, mathematical model of the system typically using ordinary differential equations. For example, the EcoPath with EcoSim (Christensen and Pauly, 1992; Christensen, 1995) software assists scientists with constructing food-web focused ecosystem models using an underlying bioenergetic approach. Alternatively, Ulanowicz (1986) has called for a more phenomenologial approach to the model construction. This modeling process starts with a conceptual network model of the system and then the node and edge weights are estimated directly from observations. Its phenomenologial in the sense that it focuses on what the flows are, rather than the forms of the mechanistic processes that generate the flows. As this approach is essentially an inverse problem, some have developed inverse linear modeling methods to assist with inferring the network weights from data (Vézina and Platt, 1988; van Oevelen, Van den Meersche, Meysman, Soetaert, Middelburg, and Vézina, 2010). The *limSolve* R pacakge can assist with this modeling approach (Soetaert, Van den Meersche, and van Oevelen, 2009). Ulanowicz and Scharler (2008) also introduced two least-inference algorithms to assist with this kind of model constuction. These methods focus on constructing models to represent specific empirical systems. Algorithms also exist for constructing simulated ecosystems, including Fath's (2004) Cyber Models that use a community assembly type approach. Currently, the *enaR* software focuses on the analysis of network models and assumes that the user has a network model to be analyzed.

2.2. Network Data Class

The *enaR* package stores the model data in the **network** class defined in the *network* package (see Butts, 2008a, for details). In this software, a complete ecosystem network model description includes:

F is the flow matrix, oriented row-to-column

z a vector of inputs

r a vector of respirations

e a vector of exports

y a vector of outputs, which are respirations plus exports

X a vector ofbiomass or storage values

Living = logical vector indicating if the node is living (TRUE) or non-living (FALSE)

2.3. Building a Network Object

Users can assemble the necessary data elements and then use the `pack` function to create the network data object. Here is an example of doing this with hypothetical data.

```
> ## Generate the flow matrix
> flow.mat <- array(abs(rnorm(100,4,2))*sample(c(0,1),100,replace=TRUE),
+                   dim=c(4,4))
> ## Name the nodes
> rownames(flow.mat) <- colnames(flow.mat) <- paste('node',(1:nrow(flow.mat)),sep='')
> ## Generate the inputs
> inputs <- runif(nrow(flow.mat),0,4)
> ## Generate the exports
> exports <- inputs
> ## "Pack" the model into a network object
> fake.model <- pack(flow=flow.mat,
+                    input=inputs,
+                    export=exports,
+                    living=TRUE)

[1] "respiration" "storage"

> ## The model network object contents
> fake.model

 Network attributes:
  vertices = 4
  directed = TRUE
  hyper = FALSE
  loops = TRUE
  multiple = FALSE
  bipartite = FALSE
  balanced = FALSE
  total edges= 8
    missing edges= 0
    non-missing edges= 8

 Vertex attribute names:
    export input living output respiration storage vertex.names
```

```
 Edge attribute names:
    flow
```

The individual components can be extracted from the data object using the form specified in the *network* package. For example, we can pull out "vertex" (i.e. node) attributes as follows:

```
> fake.model%v%'output'

[1] 3.31224122 0.05082792 0.93603094 2.03415858

> fake.model%v%'input'

[1] 3.31224122 0.05082792 0.93603094 2.03415858

> fake.model%v%'living'

[1] TRUE TRUE TRUE TRUE
```

The network flows are stored as edge weights in the network object, which lets users fully manipulate the network object with the `network` functions. The flow matrix can be extracted from the object with:

```
> as.matrix(fake.model,attrname="flow")

        node1    node2    node3    node4
node1 0.00000 0.000000 1.896386 4.077463
node2 0.00000 0.000000 7.811643 2.444950
node3 0.00000 0.000000 4.981009 0.000000
node4 3.50226 4.555268 2.485130 0.000000
```

There are times that it is useful to extract all of the ecosystem model data elements from the network data object. This can be accomplished using the `unpack` function. The `unpack` output is as follows:

```
> unpack(fake.model)

$F
        node1    node2    node3    node4
node1 0.00000 0.000000 1.896386 4.077463
node2 0.00000 0.000000 7.811643 2.444950
node3 0.00000 0.000000 4.981009 0.000000
node4 3.50226 4.555268 2.485130 0.000000

$z
[1] 3.31224122 0.05082792 0.93603094 2.03415858
```

```
$r
[1] 0 0 0 0

$e
[1] 3.31224122 0.05082792 0.93603094 2.03415858

$y
[1] 3.31224122 0.05082792 0.93603094 2.03415858

$X
[1] NA NA NA NA

$Living
[1] TRUE TRUE TRUE TRUE
```

Note that we did not specify the storage values. In these instances `pack` produces `NA` values. Although the package is designed to help users navigate missing data issues, be sure to check that you are providing the appropriate input for a given function. For more information, see the help file for the function in question.

2.4. Model Library

enaR includes a library of 100 empirically-based, previously published ecosystem models that can be categorized into two general classes: trophic and biogeochemical cycling (Christian, Fores, Comin, Viaroli, Naldi, and Ferrari, 1996; Baird *et al.*, 2008; Borrett, Whipple, and Patten, 2010; Borrett, Hines, and Carter, 2015). First, 58 of the models are trophically-based models with food webs at their core (Tables 1). Second, there are 42 models focused on biogeochemical cycling in ecosystems (Table 2). In summary, these models were originally published for a number of different types of ecosystems, though predominantly aquatic, by a number of author teams. Models in the library range in size from 4 nodes to 125 nodes with connectance values ranging from 7% to 45%.

This collection of models overlaps with other extant data sets. For example, twenty-four of the models are included in the set of forty-eight models compiled and distributed by Dr. Ulanowicz (`http://www.cbl.umces.edu/~ulan/ntwk/network.html`). All 50 of the models analyzed by Borrett and Salas (2010) and Salas and Borrett (2011) and the 45 models analyzed in Borrett (2013) are included in this model library.

The trophic models are grouped as the `troModels` object and the biogeochemically-based models are available as the `bgcModels` object. Both data objects return a list of the model network objects. To use these models simply use the R *base* `data` function. This will load the models into the working memory as a named list of network objects:

```
> ## Import the model sets
> data(bgcModels)
> data(troModels)
> ## Check the first few model names
> head(names(bgcModels))
```

```
[1] "Hubbard Brook (Ca)(Waide)"     "Hardwood Forest, NH (Ca)"
[3] "Duglas Fir Forest, WA (Ca)"    "Duglas Fir Forest, WA (K)"
[5] "Puerto Rican Rain Forest (Ca)" "Puerto Rican Rain Forest (K)"

> head(names(troModels))

[1] "Marine Coprophagy (oyster)" "Lake Findley "
[3] "Mirror Lake"                "Lake Wingra"
[5] "Marion Lake"                "Cone Springs"

> ## Isolate a single model
> x <- troModels[[1]]
> x <- troModels$"Marine Coprophagy (oyster)"
> ## Check out the model
> summary(x)

Network attributes:
  vertices = 4
  directed = TRUE
  hyper = FALSE
  loops = TRUE
  multiple = FALSE
  bipartite = FALSE
  balanced = TRUE
 total edges = 4
   missing edges = 0
   non-missing edges = 4
 density = 0.25

Vertex attributes:

 export:
   logical valued attribute
   attribute summary:
   Mode    NA's
logical       4

 input:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00   62.05   94.90  157.00  255.50

 living:
   logical valued attribute
   attribute summary:
   Mode   FALSE    TRUE    NA's
```

```
logical       2       2       0

 output:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6.60   21.67   64.45   94.90  137.70  244.10

 respiration:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6.60   21.67   64.45   94.90  137.70  244.10

 storage:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       1       1       1       1       1
  vertex.names:
   character valued attribute
   4 valid vertex names

Edge attributes:

 flow:
   numeric valued attribute
   attribute summary:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.30   20.25   37.40   42.42   59.58   79.60

Network adjacency matrix:
                       SHRIMP BENTHIC ORGANISMS
SHRIMP                      0                 0
BENTHIC ORGANISMS           0                 0
SHRIMP FECES & BACTERIA     0                 1
BENTHIC FECES & BACTERIA    0                 1
                       SHRIMP FECES & BACTERIA
SHRIMP                                       1
BENTHIC ORGANISMS                            0
SHRIMP FECES & BACTERIA                      0
BENTHIC FECES & BACTERIA                     0
                       BENTHIC FECES & BACTERIA
SHRIMP                                        0
BENTHIC ORGANISMS                             1
SHRIMP FECES & BACTERIA                       0
BENTHIC FECES & BACTERIA                      0
```

2.5. Network Visualization

Network plots are a useful tool to visualize patterns in complex datasets. Here, we present one example of how to plot a network model using the plot tools in the *network* package. The figure scaling may need to be adjusted depending on computer and the graphics devices. Also, note that the graph only shows internal system flows.

```
> ## Load data
> data(oyster)
> m <- oyster
> ## Set the random seed to control plot output
> set.seed(2)
> ## Plot network data object (uses plot.network)
> plot(m)
>
```

We can use the powerful graphics capabilities of R to make a fancier plot of the same data (Fig. 1).

```
> ## Set colors to use
> my.col <- c('red','yellow',rgb(204,204,153,maxColorValue=255),'grey22')
> ## Extract flow information for later use.
> F <- as.matrix(m,attrname='flow')
> ## Get indices of positive flows
> f <- which(F!=0, arr.ind=T)
> opar <- par(las=1,bg=my.col[4],xpd=TRUE,mai=c(1.02, 0.62, 0.82, 0.42))
> ## Set the random seed to control plot output
> set.seed(2)
> plot(m,
+  ## Scale nodes with storage
+       vertex.cex=log(m%v%'storage'),
+  ## Add node labels
+       label= m%v%'vertex.names',
+       boxed.labels=FALSE,
+       label.cex=0.65,
+  ## Make rounded nodes
+       vertex.sides=45,
+  ## Scale arrows to flow magnitude
+       edge.lwd=log10(abs(F[f])),
+       edge.col=my.col[3],
+       vertex.col=my.col[1],
+       label.col='white',
+       vertex.border = my.col[3],
+       vertex.lty = 1,
+       xlim=c(-4,1),ylim=c(-2,-2))
> ## Lastly, remove changes to the plotting parameters
> rm(opar)
```

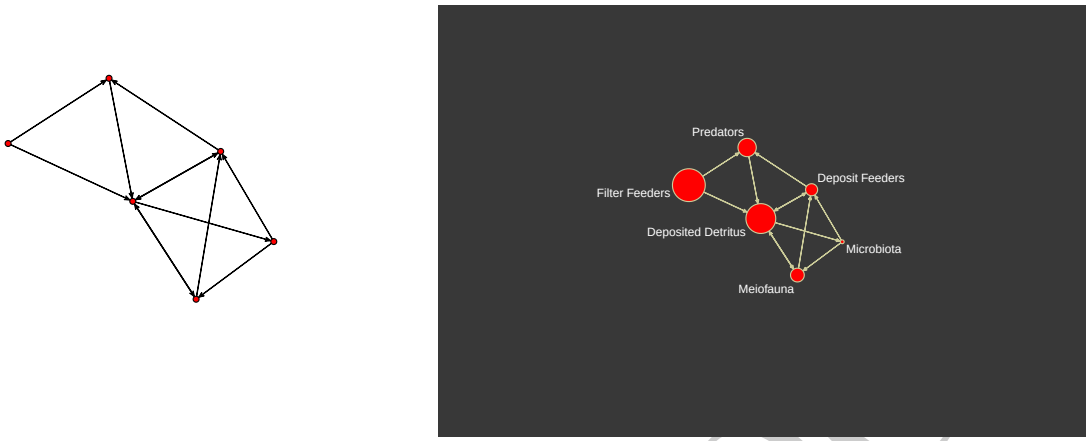

Figure 1: Two networks for the Oyster Reef model (Dame and Patten, 1981) showing a simple (left) and more elaborate (right) implementation of the network plotting function.

2.6. Data Input: Reading Common Data File Formats

Several software packages exist in the literature for running ENA. We have written functions to read in a few of the more common data formats used by them to help *enaR* users to import models formatted for these other packages. Example data files can be found in the data folder here: https://github.com/SEELab/enaR_development.

SCOR

The `read.scor` function reads in data stored in the SCOR format specified by Ulanowicz and Kay (1991) that is the input to the NETWRK4 programs. This function can be run as follows.

```
> scor.model <- readLines('../data/oyster.dat')
> m <- read.scor(scor.model,from.file=FALSE)
```

This constructs the network data object from the SCOR file that stores the ecosystem model data for an oyster reef model (Dame and Patten, 1981). The individual model elements are

```
> unpack(m)
```

```
$F
                  Filter Feeders Microbiota Meiofauna Deposit Feeders
Filter Feeders                 0     0.0000    0.0000          0.0000
Microbiota                     0     0.0000    1.2060          1.2060
Meiofauna                      0     0.0000    0.0000          0.6609
Deposit Feeders                0     0.0000    0.0000          0.0000
Predators                      0     0.0000    0.0000          0.0000
Deposited Detritus             0     8.1721    7.2745          0.6431
                  Predators Deposited Detritus
Filter Feeders       0.5135            15.7910
```

```
Microbiota            0.0000             0.0000
Meiofauna             0.0000             4.2403
Deposit Feeders       0.1721             1.9076
Predators             0.0000             0.3262
Deposited Detritus    0.0000             0.0000

$z
[1] 41.47  0.00  0.00  0.00  0.00  0.00

$r
[1] 25.1650  5.7600  3.5794  0.4303  0.3594  6.1759

$e
[1] 0 0 0 0 0 0

$y
[1] 25.1650  5.7600  3.5794  0.4303  0.3594  6.1759

$X
[1] 2000.0000    2.4121   24.1210   16.2740   69.2370 1000.0000

$Living
[1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
```

This same data is stored as a network data object that is distributed with this package, which can be accessed as:

```
> data(oyster)
> m <- oyster
```

WAND

In part to make ENA more accessible to biologists, Allesina and Bondavalli (2004) recoded some of Ulanowicz's NETWRK4 algorithms into a Microsoft Excel based tool called WAND. For this tool, the model data is stored as a separate Excel file with two worksheets. The first contains many of the node attributes and the second contains the flow matrix. The `read.wand` function will create an R network data object from a WAND model file.

```
> m <- read.wand('../data/MDmar02_WAND.xls')
```

This code creates a network data object for *enaR* from the WAND formatted Mdloti ecosystem model data (Scharler, 2012). This data is courtesy of U.M. Scharler.

NEA

For their Matlab function to perform network environ analysis (Patten School), Fath and Borrett (2006) packaged the model flows, inputs, outputs, and storage values into what they

called a system matrix $\mathrm{S} = \begin{bmatrix} \mathbf{F} & \vec{z} & \vec{X} \\ \vec{y} & 0 & 0 \end{bmatrix}_{(n+1)\times(n+2)}$. Flows in the system matrix are oriented from column to row.

The *enaR* function `read.nea` reads in data with this format stored as a comma separated value file (CSV). The function `write.nea()` will write any network model to a CSV file with this format.

While convenient, this data format does not enable inclusion of the full range of model information included in the *enaR* network data object. This format does not partition outputs into exports and respiration values, nor does it identify the node labels or their living status. This missing information will prevent the use of some *enaR* functions.

Here is an example of using these functions:

```
> data(oyster)
> ## Write oyster reef model to a CSV file
> write.nea(oyster, file.name="oyster.csv")

        [,1]  [,2]   [,3]   [,4]   [,5]   [,6]  [,7]      [,8]
[1,]  0.0000 0.000 0.0000 0.0000 0.0000 0.0000 41.47 2000.0000
[2,]  0.0000 0.000 0.0000 0.0000 0.0000 8.1721  0.00    2.4121
[3,]  0.0000 1.206 0.0000 0.0000 0.0000 7.2745  0.00   24.1210
[4,]  0.0000 1.206 0.6609 0.0000 0.0000 0.6431  0.00   16.2740
[5,]  0.5135 0.000 0.0000 0.1721 0.0000 0.0000  0.00   69.2370
[6,] 15.7910 0.000 4.2403 1.9076 0.3262 0.0000  0.00 1000.0000
[7,] 25.1650 5.760 3.5794 0.4303 0.3594 6.1759  0.00    0.0000

> ## Read in oyster reef model data from NEA.m formatted CSV file
> m <- read.nea("oyster.csv")

[1] "export" "living"

>
> ## Again, this model object does NOT contain all
> ## of the information in the "oyster" data object.
```

ENAM

Another commonly used data format stores the necessary model data in a CSV or Excel formatted file. We include an example Excel file of the Mdloti estuary stored in this form ("MDMAR02.xlsx", courtesy of U. M. Scharler). This format has not been described technically in the literature nor has it been named. We refer to it as ENAM as it is the ENA model data stored primarily as a square matrix with several preliminary rows that include meta-data, the number of nodes, and number of living nodes (similar to SCOR). The data format is generally similar in concept, if not exact form, to the data system matrix used as the input to the NEA.m function (Fath and Borrett, 2006). However, the ENAM format includes information on whether nodes are living and partitions output into respiration and exports.

Using an example data file, MDMAR02.xlsx, this data format can be read into the *enaR* package as:

```
>   m <- read.enam('../data/MDMAR02.xlsx')
```

The current read.enam function assumes the data are stored on the first worksheet of an Excel file. In the future, we expect to expand this function's capabilities to read the data from a CSV file.

3. Analyzing Ecosystem Models

ENA is often applied to investigate the structure and function of a single ecosystem model. Here, we walk through an example of applying multiple ENA algorithms to the South Carolina oyster reef model (Dame and Patten, 1981). Table 3 summarizes the main ENA algorithms encoded in *enaR*.

Again, in this package results are reported in the row-to-column orientation by default – including the algorithms from the Patten school. Please see Section 3.13 for how to change this default if needed.

3.1. Balancing for Steady-State

Many of the ENA functions assume that the network model is at steady-state (node inputs equal node outputs). Thus, this package has functions for (1) checking to see if the assumption is met and (2) automatically balancing the model so that input equal outputs.

To determine if the model is balanced and then balance it if necessary:

```
> ## Check to see if the model is balanced
> ssCheck(fake.model)

[1] FALSE

> ## To BALANCE a model if needed
> fake.model <- balance(fake.model,method="AVG2")

[1] AVG2

> ## To FORCE BALANCE a model if needed
> fake.model <- force.balance(fake.model)
```

The automated balancing routines are based on those presented in Allesina and Bondavalli (2003) and include Input, Output, AVG, and AVG2. These authors compare these alternative balancing algorithms and further discuss the implications of using automated procedures. Caution is warranted when using these techniques, as they indiscriminately alter the model flow rates. A more neuanced appraoch may be desired when the uncertainty in estimates of model fluxes are known.

3.2. Structural Network Analysis

Structural network analysis is common to many types of network analysis. The structural analyses applied here are based on those presented in NEA.m (Fath and Borrett, 2006) following the Patten School. Output of the enaStructure function is summarized in Table 4.

```
> data(oyster)
> St <- enaStructure(oyster)
> attributes(St)

$names
[1] "A"  "ns"

> St$ns

     n  L         C LD      ppr    lam1A mlam1A      rho         R
[1,] 6 12 0.3333333  2 2.147899 2.147899      1 2.147899 0.4655712
           d no.scc no.scc.big      pscc
[1,] 0.147899      2          1 0.8333333
```

The number of nodes, number of links, link density, and connectance (density) are common statistics used to describe networks like food webs (Martinez, 1992; Dunne, Williams, and Martinez, 2002; Eklöf and Ebenman, 2006; Estrada, 2007; Brandes and Erlebach, 2005). The pathway proliferation rate quantifies if and how fast the number of pathways increases with path length in the network (Borrett and Patten, 2003; Borrett *et al.*, 2007). This rate is equivalent to the dominant eigenvalue of the adjacency matrix ($\lambda_1(A)$) if the network is comprised of a single strongly connected component (Borrett *et al.*, 2007).

The structural network statistics for the oyster reef model shows that it has 6 nodes, a pathway proliferation rate of 2.14 (ppr), and that the model is comprised of two strongly connected components (no.scc) but that only one has more than one node (no.scc.big). Thus, 83% of the nodes are participating in a strongly connected component (pscc).

3.3. Flow Analysis

Flow analysis is one of the core ENA analyses for both the Ulanowicz and Patten Schools (Fath and Patten, 1999; Latham II, 2006; Fath and Borrett, 2006; Schramski, Kazanci, and Tollner, 2011). The *enaR* implementation `enaFlow` mostly follows the NEA.m function, with small updates (e.g. calculating the ratio of indirect-to-direct flows Borrett and Freeze, 2011; Borrett, Freeze, and Salas, 2011). Results returned by `enaFlow` are summarized in Table 5

To validly apply flow analysis, the network model must meet two analytical assumptions. First, the model must trace a single, thermodynamically conserved currency, such as energy, carbon, or nitrogen. Second, the model must be at steady-state for many of the analyses.

Flow analysis has been used in a variety of ways. For example, Finn (1980) used ENA flow analysis to compare the cycling of multiple nutrients through the Hubbard Brook Ecosystem, New Hampshire, USA, and van Oevelen, Duineveld, Lavaleye, Mienis, Soetaert, and Heip (2009) used the technique to show how different marine canyon conditions change the flow of carbon through the food webs in Nazaré Canyon. Gattie, Schramski, Borrett, Patten, Bata, and Whipple (2006) applied the analysis to characterize N cycling in the Neuse River Estuary (North Carolina, USA), and Zhang *et al.* (2010) used flow analysis to help assess the sustainability of the urban water metabolism of Beijing, China. Borrett (2013) showed that the throughflow vector T can be considered as a type of centrality measure that indicates the relative importance of each node to the generation of the total system throughflow or activity.

Here, we extract the flow statistics and then isolate and remove the output-oriented direct flow intensity ($\mathbf{G}$) matrix. Recall that ENA is partially derived from Input–Output analysis; the input and output orientations provide different information about the system. We also show the input-oriented integral flow matrix $\mathbf{N}'$.

```
> F <- enaFlow(oyster)
> attributes(F)

$names
[1] "T"  "G"  "GP" "N"  "NP" "ns"

> F$ns

     Boundary     TST     TSTp      APL       FCI       BFI       DFI
[1,]    41.47 83.5833 125.0533 2.015512 0.1101686 0.4961517 0.1950689
          IFI     ID.F   ID.F.I   ID.F.O    HMG.I    HMG.O AMP.I AMP.O
[1,] 0.3087794 1.582925 1.716607 1.534181 2.051826 1.891638     3     1
     mode0.F  mode1.F  mode2.F  mode3.F mode4.F
[1,]   41.47 32.90504 9.208256 32.90504   41.47

> ## Output-oriented direct flow matrix
> F$G

                  Filter Feeders Microbiota Meiofauna Deposit Feeders
Filter Feeders                 0  0.0000000 0.0000000      0.00000000
Microbiota                     0  0.0000000 0.1475753      0.14757529
Meiofauna                      0  0.0000000 0.0000000      0.07793173
Deposit Feeders                0  0.0000000 0.0000000      0.00000000
Predators                      0  0.0000000 0.0000000      0.00000000
Deposited Detritus             0  0.3670363 0.3267221      0.02888377
                   Predators Deposited Detritus
Filter Feeders    0.01238245          0.3807813
Microbiota        0.00000000          0.0000000
Meiofauna         0.00000000          0.5000059
Deposit Feeders   0.06856574          0.7600000
Predators         0.00000000          0.4757876
Deposited Detritus 0.00000000          0.0000000

> ## Input-oriented integral flow matrix
> F$NP

                  Filter Feeders Microbiota Meiofauna Deposit Feeders
Filter Feeders                 1  1.0000000 1.0000000       1.0000000
Microbiota                     0  1.1018630 0.2440716       0.6197856
Meiofauna                      0  0.2971032 1.2971032       0.5604100
Deposit Feeders                0  0.1240688 0.1240688       1.1240688
```

```
Predators                       0  0.0203426 0.0203426       0.0203426
Deposited Detritus              0  1.3885039 1.3885039       1.3885039
                   Predators Deposited Detritus
Filter Feeders     1.0000000          1.0000000
Microbiota         0.1555792          0.1018630
Meiofauna          0.1406747          0.2971032
Deposit Feeders    0.2821649          0.1240688
Predators          1.0051064          0.0203426
Deposited Detritus 0.3485436          1.3885039
```

Note that you can use the `attach` function to have access to the objects nested within an object. Since some objects may conflict in name, it's best to `detach` an object once it's not in use.

```
> attach(F)
> G

                   Filter Feeders Microbiota Meiofauna Deposit Feeders
Filter Feeders                  0  0.0000000 0.0000000      0.00000000
Microbiota                      0  0.0000000 0.1475753      0.14757529
Meiofauna                       0  0.0000000 0.0000000      0.07793173
Deposit Feeders                 0  0.0000000 0.0000000      0.00000000
Predators                       0  0.0000000 0.0000000      0.00000000
Deposited Detritus              0  0.3670363 0.3267221      0.02888377
                    Predators Deposited Detritus
Filter Feeders     0.01238245          0.3807813
Microbiota         0.00000000          0.0000000
Meiofauna          0.00000000          0.5000059
Deposit Feeders    0.06856574          0.7600000
Predators          0.00000000          0.4757876
Deposited Detritus 0.00000000          0.0000000

> detach(F)
```

3.4. Ascendency

A key contribution of the Ulanowicz School to ENA is the Ascendency concept and the development of several information based network-level statistics (Ulanowicz, 1986, 1997). This analysis is based on all of the flows in the system and does not assume the modeled system is at steady-state. The `enaAscendency` function returns several of these information based measures (Table 6). This is run as follows:

```
>   enaAscendency(oyster)

          AMI      ASC       OH      CAP   ASC.CAP    OH.CAP robustness
[1,] 1.330211 166.3473 211.0979 377.4452 0.4407191 0.5592809  0.3611021
         ELD       TD
[1,] 1.79506 2.514395
```

3.5. Storage Analysis

Storage ENA was developed in the Patten School (Barber, 1978b,a). It is similar to flow ENA, but divides the flows by storage (e.g., biomass) instead of throughflow. Several papers provide an overview of this methodology Fath and Patten (1999); Gattie *et al.* (2006); Schramski *et al.* (2011). Output of this function is summarized in Table 7. What follows is an example of applying the storage analysis to the oyster reef model.

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

$names
 [1] "X"  "C"  "P"  "S"  "Q"  "CP" "PP" "SP" "QP" "dt" "ns"

> S$ns

          TSS       CIS         BSI         DSI       ISI     ID.S
[1,] 3112.044 0.9940252 0.003331412 0.003320932 0.9933477 299.1171
      ID.S.I   ID.S.O  HMG.S.O  HMG.S.I NAS NASP mode0.S  mode1.S
[1,] 454.227 294.1527 1.115985 1.464503  20   21 10.3675 8.226261
     mode2.S  mode3.S mode4.S
[1,] 3093.45 8.226261 10.3675
```

This storage analysis of the oyster reef model indicates that the total energy stored in the system on an average day is 3112 kcal m^-2, and that 99.3% of this storage is generated by energy flowing over indirect pathways (ISI).

Whipple, Patten, and Borrett (2014) provides a detailed example of applying storage analysis to characterize the dynamic organization of an ecosystem. They investigated how the storage analysis properties changed across sixteen consecutive seasonal N cycling models of the Neuse River Estuary. They found that from this storage perspective NOx was the dominant compartment, and thus a primary controller of the system dynamics. Note that this work provides an example of applying this analysis at multiple levels of analysis (e.g., Hines and Borrett, 2014).

3.6. Utility Analysis

Utility analysis describes the relationship between node pairs in the ecosystem model when considering both direct and indirect interactions. It developed in the Patten School (Patten, 1991; Fath and Patten, 1999) and is similar to yet distinct from the Ulanowicz School mixed trophic impacts analysis (Ulanowicz and Puccia, 1990). Utility analysis can be conducted from both the flow and storage perspectives, so the "type" argument needs to be set to suit the user's needs. This is again implemented as in NEA.m. Table 8 summarizes the function output for the flow and storage versions. These analyses are executed as:

```
> UF <- enaUtility(oyster, eigen.check=TRUE,type="flow")
> US <- enaUtility(oyster, eigen.check=TRUE,type="storage")
> attributes(UF)
```

```
$names
[1] "D"  "U"  "Y"  "ns"
```

Please note the function argument "eigen.check=TRUE". For this analysis to work, the power series of the direct utility matrices must converge, which is only true if the dominant eigenvalue of the direct utility matrix is less than 1. The function default prevents the analysis from being performed if this condition is not met. Users that wish to perform the analysis anyway can set "eigen.check=FALSE". Care should be used when doing this, as the meaning of the underlying mathematics is uncertain.

3.7. Environ Analysis

Environ Analysis finds the n unit input and output environs for the model (Patten, 1978; Fath and Patten, 1999). These unit environs are returned by the *environ* function as in NEA.m. They indicate the flow activity in each subnetwork generated by pulling a unit out of a node (input environs) or pushing a unit into a node (output environ). These unit environs can be converted into "realized" environs by multiplying each by the relevant observed input or output (Borrett and Freeze, 2011; Whipple, Borrett, Patten, Gattie, Schramski, and Bata, 2007; Whipple *et al.*, 2014).

```
> E <- enaEnviron(oyster)
> attributes(E)

$names
[1] "input"  "output"

> E$output[1]

$`Filter Feeders`
                  Filter Feeders Microbiota   Meiofauna
Filter Feeders                -1  0.0000000  0.00000000
Microbiota                     0 -0.1970605  0.02908126
Meiofauna                      0  0.0000000 -0.20449723
Deposit Feeders                0  0.0000000  0.00000000
Predators                      0  0.0000000  0.00000000
Deposited Detritus             0  0.1970605  0.17541596
z                              1  0.0000000  0.00000000
                  Deposit Feeders    Predators Deposited Detritus
Filter Feeders         0.00000000  0.012382445        0.380781288
Microbiota             0.02908126  0.000000000        0.000000000
Meiofauna              0.01593682  0.000000000        0.102249819
Deposit Feeders       -0.06052568  0.004149988        0.045999518
Predators              0.00000000 -0.016532433        0.007865927
Deposited Detritus     0.01550760  0.000000000       -0.536896552
z                      0.00000000  0.000000000        0.000000000
                             y
Filter Feeders     0.606836267
```

```
Microbiota          0.138897999
Meiofauna           0.086310586
Deposit Feeders     0.010376176
Predators           0.008666506
Deposited Detritus 0.148912467
z                   0.000000000
```

The `TET` function returns vectors of the unit and realized input and output total environ throughflow. The realized total environ throughflow is an environ based partition of the total system throughflow (Whipple *et al.*, 2007).

```
> tet <- TET(oyster)
> show(tet)

$realized.input
[1] 25.165000 22.647638 14.582798  2.028052  1.053786 18.107007

$realized.output
[1] 83.5833  0.0000  0.0000  0.0000  0.0000  0.0000

$unit.input
[1] 1.000000 3.931882 4.074090 4.713111 2.932069 2.931882

$unit.output
[1] 2.015512 1.836089 2.540670 3.124836 2.234317 2.594261
```

The `TES` functions returns the both the realized and unit total environ storage for the input and output environs. Again, the realized TES is a partition of the total system storage (TSS).

```
> tes <- TES(oyster)
> show(tes)

$realized.input
    Filter Feeders          Microbiota           Meiofauna
         2000.00000             2.41209            24.12171
   Deposit Feeders           Predators  Deposited Detritus
           16.27440            69.23803          1000.03118

$realized.output
[1] 3112.044    0.000    0.000    0.000    0.000    0.000

$unit.input
    Filter Feeders          Microbiota           Meiofauna
       289.3658066            0.6561948            7.3735209
   Deposit Feeders           Predators  Deposited Detritus
        11.5308112          109.7205293          265.1036470
```

```
$unit.output
    Filter Feeders         Microbiota          Meiofauna
          75.04326           16.06273           41.03146
   Deposit Feeders          Predators Deposited Detritus
          65.81279          132.44451           66.11575
```

Realized TET and TES might be considered network centrality measures that indicate the relative importance of the environs in generating the observed flow or storage, respectively.

3.8. Control Analysis

Control analysis was implemented as in the NEA.m function, but we also include recent updates to control analysis (e.g., Schramski, Gattie, Patten, Borrett, Fath, Thomas, and Whipple, 2006; Schramski, Gattie, Patten, Borrett, Fath, and Whipple, 2007). In general, these analyses determine the pairwise control relationships between the nodes in the network. Table 9 summarizes the function output.

```
> C <- enaControl(oyster)
> attributes(C)

$names
[1] "CN" "CQ" "CR" "CD" "sc"

> C$sc

    Filter Feeders         Microbiota          Meiofauna
       0.120569086       -0.063395416       -0.042707703
   Deposit Feeders          Predators Deposited Detritus
       0.002631762       -0.069124796        0.052027067
```

The elements of the sc vector indicate the relative control exherted by each node on the system functioning.

3.9. Mixed Trophic Impacts

Mixed Trophic Impacts is a popular analysis from the Ulanowicz School of ENA (Ulanowicz and Puccia, 1990). The `enaMTI` function generates comparable results to the calculations in Ulanowicz and Puccia (1990). These are implemented as follows; Table 10 summarizes the function output.

```
> mti <- enaMTI(oyster)
> attributes(mti)

$names
[1] "G"  "FP" "Q"  "M"
```

```
> mti$M

[1] NA
```

In this case, the power series of the direct trophic impacts matrix does not converge (dominant eigenvalue is greater than one). Thus, the function returns the mti$M = NA. Like with Utility analysis, however, we can use the eigen.check argument to do the calculation despite the mathematical problem.

```
> mti <- enaMTI(oyster,eigen.check=FALSE)
> attributes(mti)

$names
[1] "G"  "FP" "Q"  "M"

> mti$M

                  Filter Feeders  Microbiota    Meiofauna
Filter Feeders      -0.0250635283  0.16956382  0.431493557
Microbiota          -0.0015848556 -0.30675078 -0.182458391
Meiofauna           -0.0001241781 -0.47413204 -0.070959618
Deposit Feeders     -0.0069255188 -0.26769125 -0.007062628
Predators           -0.0301817448  0.02000515 -0.004028911
Deposited Detritus  -0.0034657973  0.21795628  0.612654910
                  Deposit Feeders    Predators Deposited Detritus
Filter Feeders         0.26144106  0.795834137        0.516016759
Microbiota             0.20520368  0.050323410       -0.295378609
Meiofauna              0.01607831  0.003942987       -0.001592286
Deposit Feeders       -0.10329881  0.219903765        0.177109591
Predators             -0.07586335 -0.041648786       -0.019939324
Deposited Detritus     0.44874394  0.110048344       -0.251366300
```

The mixed trophic impacts analysis has been usefully applied to discover interesting and sometimes unexpected ecological relationships. For example, although alligators directly eat frogs in the Florida Everglades (USA), it appears that their net relationship when considering the whole food web is actually mutualistic (Bondavalli and Ulanowicz, 1999). This is in part because the alligators also eat other key predators of the frogs such as snakes.

3.10. Cycle Analysis

The Cycle Analysis provides the detailed account of the cycling present in the network. It follows the algorithm by the DOS-based NETWRK 4.2b software by Ulanowicz (Ulanowicz and Kay, 1991; Ulanowicz, 1983) and provides results similar to NETWRK's 'Full Cycle Analysis'. Cycles in a network are grouped together into disjoint nexuses and each nexus is characterized by a weak arc. This function gives details of the individual cycles along with the disjoint nexuses present in the network. Table 11 summarizes the function output.

```
>   cyc <- enaCycle(oyster)
> attributes(cyc)

$names
[1] "Table.cycle"      "Table.nexus"      "CycleDist"
[4] "NormDist"         "ResidualFlows"    "AggregatedCycles"
[7] "ns"

> ## The individual cycles
> names(cyc$Table.cycle)

[1] "CYCLE" "NEXUS" "NODES"

> ## The disjoint nexuses
> names(cyc$Table.nexus)

[1] "NEXUS"      "CYCLES"     "W.arc.From" "W.arc.To"   "W.arc.Flow"
```

3.11. Trophic Aggregations

The Trophic Aggregation algorithm identifies the trophic structure of the given network based on the Lindeman's trophic concepts (Lindeman, 1942). The algorithm is implemented as in NETWRK 4.2b by Ulanowicz (Ulanowicz and Kemp, 1979) and provides similar results as NETWRK's 'Lindeman Trophic Aggregations' (Ulanowicz and Kay, 1991). It apportions the nodes into integer trophic levels and estimates the corresponding inputs, exports, respirations and the grazing chain and trophic spine which represent the transfers between integer trophic levels. This analysis assumes that the ecosystem network model being analyzed represents a food web.

It is crucial for this algorithm that the cycles among the living nodes of the network (Feeding Cycles) be removed beforehand to assign trophic levels to nodes. Thus, the output for this function contains the Cycle Analysis for the Feeding Cycles in the network.

Following Ulanowicz and Kay (1991), the non-living nodes are grouped together for this analysis and referred to as the detrital pool.

Table 12 summarizes the function output except the outputs for the feeding cycles which are similar to the enaCycle outputs.

```
> trop <- enaTroAgg(oyster)
> attributes(trop)

$names
 [1] "Feeding_Cycles" "A"              "ETL"            "CE"
 [5] "CR"             "GC"             "RDP"            "LS"
 [9] "TE"             "ns"

> ## Cycle analysis output for Feeding Cycles
> trop$Feeding_Cycles
```

```
$ResidualFlows
                Filter Feeders Microbiota Meiofauna Deposit Feeders
Filter Feeders               0          0     0.000          0.0000
Microbiota                   0          0     1.206          1.2060
Meiofauna                    0          0     0.000          0.6609
Deposit Feeders              0          0     0.000          0.0000
Predators                    0          0     0.000          0.0000
                Predators
Filter Feeders     0.5135
Microbiota         0.0000
Meiofauna          0.0000
Deposit Feeders    0.1721
Predators          0.0000
```

3.12. Other Analyses

There are a number of additional tools in the package. Here selected a subset of these to highlight.

Quickly Return Multiple Analyses

There are two functions that aggregate multiple analyses and report selected results. A quick way to get a list of the global network statistics reported in Structure, Flow, Ascendency, Storage, and Utility analysis is to use the `get.ns` function.

```
> ns <- get.ns(oyster)
> ## Examine the structure of ns
> str(ns)

'data.frame':        1 obs. of  65 variables:
 $ n          : num 6
 $ L          : num 12
 $ C          : num 0.333
 $ LD         : num 2
 $ ppr        : num 2.15
 $ lam1A      : num 2.15
 $ mlam1A     : num 1
 $ rho        : num 2.15
 $ R          : num 0.466
 $ d          : num 0.148
 $ no.scc     : num 2
 $ no.scc.big : num 1
 $ pscc       : num 0.833
 $ Boundary   : num 41.5
 $ TST        : num 83.6
 $ TSTp       : num 125
 $ APL        : num 2.02
```

```
$ FCI         : num 0.11
$ BFI         : num 0.496
$ DFI         : num 0.195
$ IFI         : num 0.309
$ ID.F        : num 1.58
$ ID.F.I      : num 1.72
$ ID.F.O      : num 1.53
$ HMG.I       : num 2.05
$ HMG.O       : num 1.89
$ AMP.I       : num 3
$ AMP.O       : num 1
$ mode0.F     : num 41.5
$ mode1.F     : num 32.9
$ mode2.F     : num 9.21
$ mode3.F     : num 32.9
$ mode4.F     : num 41.5
$ AMI         : num 1.33
$ ASC         : num 166
$ OH          : num 211
$ CAP         : num 377
$ ASC.CAP     : num 0.441
$ OH.CAP      : num 0.559
$ robustness  : num 0.361
$ ELD         : num 1.8
$ TD          : num 2.51
$ TSS         : num 3112
$ CIS         : num 0.994
$ BSI         : num 0.00333
$ DSI         : num 0.00332
$ ISI         : num 0.993
$ ID.S        : num 299
$ ID.S.I      : num 454
$ ID.S.O      : num 294
$ HMG.S.O     : num 1.12
$ HMG.S.I     : num 1.46
$ NAS         : num 20
$ NASP        : num 21
$ mode0.S     : num 10.4
$ mode1.S     : num 8.23
$ mode2.S     : num 3093
$ mode3.S     : num 8.23
$ mode4.S     : num 10.4
$ lam1D       : num 0.899
$ synergism.F: num 4.92
$ mutualism.F: num 2.27
$ lam1DS      : num 0.302
$ synergism.S: num 13.1
```

```
 $ mutualism.S: num 2.6
```

It is also possible to instantly return all of the main ENA output with `enaAll`:

```
> oyster.ena <- enaAll(oyster)
> names(oyster.ena)

[1] "ascendency" "control"    "environ"    "flow"       "mti"
[6] "storage"    "structure"  "utility"
```

Centrality

Centrality analysis is a large topic in network science (Brandes and Erlebach, 2005; Wasserman and Faust, 1994). In general the goal is to describe the relative importance of parts of the networks (nodes, edges, environs). Many different types of centrality measures exist in network science (Freeman, 1979; Freeman, Borgatti, and White, 1991; Borgatti and Everett, 2006; Brandes and Erlebach, 2005). Environ centrality is unique to ENA (Fann and Borrett, 2012), but like eigenvector centrality, it is a degree-based centrality measure that considers the equilibrium effect of all pathways of all lengths in the system and as such can be classified as a global centrality measure. Both of these centralities can be calculated in *enaR* as follows:

```
> F <- enaFlow(oyster)
> ec <- environCentrality(F$N)
> show(ec)

$ECin
    Filter Feeders          Microbiota           Meiofauna
         0.1404961           0.1279889           0.1771034
   Deposit Feeders           Predators Deposited Detritus
         0.2178241           0.1557484           0.1808391

$ECout
    Filter Feeders          Microbiota           Meiofauna
        0.06970737          0.19108709          0.20595483
   Deposit Feeders           Predators Deposited Detritus
        0.12350944          0.07903903          0.33070223

$AEC
    Filter Feeders          Microbiota           Meiofauna
         0.1051017           0.1595380           0.1915291
   Deposit Feeders           Predators Deposited Detritus
         0.1706668           0.1173937           0.2557707

> eigenCentrality(F$G)
```

```
$EVCin
[1] 0.1207568 0.1093625 0.1876329 0.2518905 0.1470501 0.1833072

$EVCout
[1] 0.00000000 0.23325048 0.26566843 0.11130122 0.01286707 0.37691280

$AEVC
[1] 0.06037842 0.17130647 0.22665067 0.18159586 0.07995858 0.28011000
```

These centrality values have been normalized to sum to one. In addition, the throughflow vector from flow analysis (Borrett, 2013), the total environ throughflow, and total environ storage vectors might also be considered centrality metrics (Whipple *et al.*, 2007, 2014). Figure 2 shows one way to visualize the Average Environ and Throughflow Centralities.

```
> ## Set plotting parameters
> opar <- par(las=1,mfrow=c(1,2),mar=c(7,5,1,1),xpd=TRUE,bg="white")
> ## Find centrality order
> o <- order(ec$AEC,decreasing=TRUE)
> ## Creating a barplot
> bp <- barplot(ec$AEC[o],
+                names.arg=NA,
+                ylab="Average Environ Centrality",
+                col="black",border=NA)
> ## Adding labels
> text(bp,-0.008,
+       labels=names(ec$AEC)[o],
+       srt=35,adj=1,cex=1)
> # throughflow centrality
> T <- enaFlow(oyster)$T
> o <- order(T,decreasing=TRUE)
> bp2 <- barplot(T[o],
+                names.arg=NA,
+                ylab=expression(paste("Throughflow (kcal m"^-2, " y"^-1,")")),
+                col="black", border=NA)
> text(bp2,-1,
+       labels=names(T)[o],
+       srt=35,adj=1,cex=1)
> ## Remove the plotting parameters
>  rm(opar)
```

3.13. Output Orientation

To facilitate package use by the existing ENA community, some of which use the column-to-row orientation (e.g. the Patten School), we have created orientation functions that enable the user to set the expected output orientation for functions written in a particular "school" of analysis. Thus, functions from either school will receive network models with the standard row-to-column, but will return output with flow matrices oriented in the column-to-row

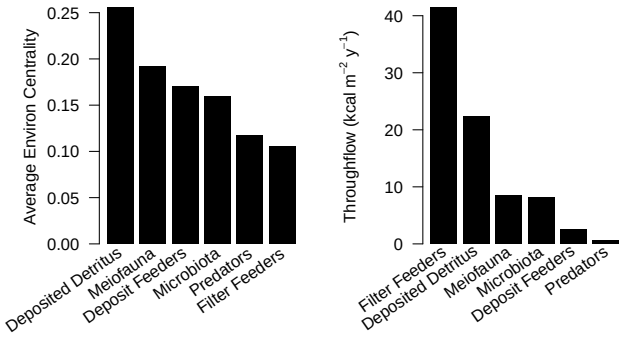

Figure 2: Bar plots of the Oyster Reef model Average Environ Centralities (left) and Throughflow Centralities (right).

orientation when appropriate (i.e. Patten school functions) and return them in that same orientation.

Here is an example of how to use the model orientation functions to re-orient the output from `enaFlow`:

```
> ## Check the current orientation
> get.orient()

[1] "rc"

> ## enaFlow output in row-column
> flow.rc <- enaFlow(oyster)$G
> ## Set the global orientation to school
> set.orient('school')
> ## Check that it worked
> get.orient()

[1] "school"

> ## enaFlow output in column-row
> flow.cr <- enaFlow(oyster)$G
> ## Check. Outputs should be transposed from each other.
> all(flow.rc == flow.cr)

[1] FALSE

> all(flow.rc == t(flow.cr))

[1] TRUE

> ## Now change back to the default orientation ('rc')
> set.orient('rc')
```

Matrix powers – raising a matrix to a power is not a native operation in R . Thus, the *enaR* package includes a function `mExp` to facilitate this matrix operation commonly used in ENA.

```
> mExp(F$G,2)
```

```
                  Filter Feeders Microbiota  Meiofauna Deposit Feeders
Filter Feeders                 0  0.1397606 0.12440966      0.01099840
Microbiota                     0  0.0000000 0.00000000      0.01150080
Meiofauna                      0  0.1835203 0.16336297      0.01444205
Deposit Feeders                0  0.2789476 0.24830879      0.02195166
Predators                      0  0.1746313 0.15545033      0.01374254
Deposited Detritus             0  0.0000000 0.05416549      0.07962750
                     Predators Deposited Detritus
Filter Feeders     0.000000000         0.005891414
Microbiota         0.010118608         0.185945731
Meiofauna          0.005343446         0.059228112
Deposit Feeders    0.000000000         0.032622730
Predators          0.000000000         0.000000000
Deposited Detritus 0.001980437         0.185314635
```

4. Multi-Model Analyses (Batch Processing)

While many investigators analyze single models, much of ENA is used to compare ecosystem models (e.g., Baird *et al.*, 1991; van Oevelen, Soetaert, Middelburg, Herman, Moodley, Hamels, Moens, and Heip, 2006; Christian and Thomas, 2003; Niquil, Chaumillon, Johnson, Bertin, Grami, David, Bacher, Asmus, Baird, and Asmus, 2012; Hines *et al.*, 2015). Investigators have also analyzed large sets of models to determine the generality of hypothesized ecosystem properties (e.g., Christensen, 1995; Borrett and Salas, 2010; Salas and Borrett, 2011). For both of these applications, investigators need to analyze multiple models. One advantage of the *enaR* R package is that it simplifies this batch processing. Here we illustrate how to batch analyze a selection of models.

Our first step is to build an R list data object with ecosystem network models to batch analyze as the elements of the list. To illustrate batch processing, we will use a subset of the trophic models distributed with *enaR*, which are already stored as a list.

```
> data(troModels)
```

Now that we have the models loaded, we can start to manipulate them. Once we have balanced the models, we can run the flow analysis on them. We are using the `lapply` function to iterate the analysis across the list of models stored in model.list. This approach is more compact and computationally efficient than a using for-loop.

```
> # balance models as necessary
> m.list <- lapply(troModels[1:10],balance)
```

```
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
[1] BALANCED
```

```
> # check that models are balanced
> unlist(lapply(m.list,ssCheck))
```

```
Marine Coprophagy (oyster)              Lake Findley
                      TRUE                      TRUE
              Mirror Lake                Lake Wingra
                      TRUE                      TRUE
              Marion Lake              Cone Springs
                      TRUE                      TRUE
           Silver Springs            English Channel
                      TRUE                      TRUE
            Oyster Reef                 Baie de Somme
                      TRUE                      TRUE
```

```
> ## If balancing fails, you can use force.balance
> ## to repeatedly apply the balancing procedure
> ## although this is not the case with our model set
>
> m.list <- lapply(m.list,force.balance)
> ## Check that all the models are balanced
> all(unlist(lapply(m.list,ssCheck)))
```

```
[1] TRUE
```

```
> ## Example Flow Analysis
> F.list <- lapply(m.list, enaFlow)
> ## The full results of the flow analysis is now stored in the elements
> ## of the F.list.  To get the results for just the first model:
> F.list[[1]]
```

```
$T
                  SHRIMP        BENTHIC ORGANISMS
                   124.1                    323.7
 SHRIMP FECES & BACTERIA BENTHIC FECES & BACTERIA
                    21.9                     79.6
```

```
$G
                         SHRIMP BENTHIC ORGANISMS
SHRIMP                        0          0.0000000
BENTHIC ORGANISMS             0          0.0000000
SHRIMP FECES & BACTERIA       0          0.6986301
BENTHIC FECES & BACTERIA      0          0.6645729
                         SHRIMP FECES & BACTERIA
SHRIMP                                  0.1764706
BENTHIC ORGANISMS                       0.0000000
SHRIMP FECES & BACTERIA                 0.0000000
BENTHIC FECES & BACTERIA                0.0000000
                         BENTHIC FECES & BACTERIA
SHRIMP                                   0.0000000
BENTHIC ORGANISMS                        0.2459067
SHRIMP FECES & BACTERIA                  0.0000000
BENTHIC FECES & BACTERIA                 0.0000000

$GP
                         SHRIMP BENTHIC ORGANISMS
SHRIMP                        0         0.00000000
BENTHIC ORGANISMS             0         0.00000000
SHRIMP FECES & BACTERIA       0         0.04726599
BENTHIC FECES & BACTERIA      0         0.16342292
                         SHRIMP FECES & BACTERIA
SHRIMP                                         1
BENTHIC ORGANISMS                              0
SHRIMP FECES & BACTERIA                        0
BENTHIC FECES & BACTERIA                       0
                         BENTHIC FECES & BACTERIA
SHRIMP                                          0
BENTHIC ORGANISMS                               1
SHRIMP FECES & BACTERIA                         0
BENTHIC FECES & BACTERIA                        0

$N
                         SHRIMP BENTHIC ORGANISMS
SHRIMP                        1          0.1473716
BENTHIC ORGANISMS             0          1.1953471
SHRIMP FECES & BACTERIA       0          0.8351055
BENTHIC FECES & BACTERIA      0          0.7943953
                         SHRIMP FECES & BACTERIA
SHRIMP                                  0.1764706
BENTHIC ORGANISMS                       0.0000000
SHRIMP FECES & BACTERIA                 1.0000000
BENTHIC FECES & BACTERIA                0.0000000
                         BENTHIC FECES & BACTERIA
```

```
SHRIMP                                0.03623966
BENTHIC ORGANISMS                     0.29394387
SHRIMP FECES & BACTERIA               0.20535805
BENTHIC FECES & BACTERIA              1.19534712

$NP
                         SHRIMP BENTHIC ORGANISMS
SHRIMP                        1         0.05649926
BENTHIC ORGANISMS             0         1.19534712
SHRIMP FECES & BACTERIA       0         0.05649926
BENTHIC FECES & BACTERIA      0         0.19534712
                         SHRIMP FECES & BACTERIA
SHRIMP                                         1
BENTHIC ORGANISMS                              0
SHRIMP FECES & BACTERIA                        1
BENTHIC FECES & BACTERIA                       0
                         BENTHIC FECES & BACTERIA
SHRIMP                                 0.05649926
BENTHIC ORGANISMS                      1.19534712
SHRIMP FECES & BACTERIA                0.05649926
BENTHIC FECES & BACTERIA               1.19534712

$ns
     Boundary   TST  TSTp     APL       FCI       BFI       DFI
[1,]    379.6 549.3 928.9 1.44705 0.1199863 0.6910614 0.1542493
           IFI     ID.F    ID.F.I    ID.F.O    HMG.I    HMG.O AMP.I
[1,] 0.1546893 1.002852 0.3603839 0.6126851 2.014161 1.891504     1
     AMP.O mode0.F  mode1.F  mode2.F  mode3.F mode4.F
[1,]     0   379.6 103.7915 65.90846 103.7915   379.6
```

We can use the same technique to extract specific information, like just the ratio of Indirect-to-Direct flow for each model.

```
> ## Example of extracting just specific information - Indirect Effects Ratio
> IDs <- unlist(lapply(m.list, function(x) enaFlow(x)$ns[9]))
> ## Look at the first few ID's
> head(IDs)

Marine Coprophagy (oyster)              Lake Findley
                  1.002852                  1.723221
               Mirror Lake                Lake Wingra
                  1.861121                  1.861719
               Marion Lake               Cone Springs
                  2.175878                  1.023016
```

We can also collect the set of output-oriented integral flow matrices.

```
> ## Here is a list containing only the
> ## output-oriented integral flow matrices
> N.list <- lapply(m.list,function(x) enaFlow(x)$N)
```

We can also apply the `get.ns` function to extract all of the network statistics for each model. We then use the `do.call` function to reshape the network statistics into a single data frame.

```
> ## Collecting and combining all network statistics
> ns.list <- lapply(m.list,get.ns) # returns as list
> ns <- do.call(rbind,ns.list)  # ns as a data.frame
> ## Let's take a quick look at some of the output
> colnames(ns)    # return network statistic names.

 [1] "n"           "L"           "C"           "LD"
 [5] "ppr"         "lam1A"       "mlam1A"      "rho"
 [9] "R"           "d"           "no.scc"      "no.scc.big"
[13] "pscc"        "Boundary"    "TST"         "TSTp"
[17] "APL"         "FCI"         "BFI"         "DFI"
[21] "IFI"         "ID.F"        "ID.F.I"      "ID.F.O"
[25] "HMG.I"       "HMG.O"       "AMP.I"       "AMP.O"
[29] "mode0.F"     "mode1.F"     "mode2.F"     "mode3.F"
[33] "mode4.F"     "AMI"         "ASC"         "OH"
[37] "CAP"         "ASC.CAP"     "OH.CAP"      "robustness"
[41] "ELD"         "TD"          "TSS"         "CIS"
[45] "BSI"         "DSI"         "ISI"         "ID.S"
[49] "ID.S.I"      "ID.S.O"      "HMG.S.O"     "HMG.S.I"
[53] "NAS"         "NASP"        "mode0.S"     "mode1.S"
[57] "mode2.S"     "mode3.S"     "mode4.S"     "lam1D"
[61] "synergism.F" "mutualism.F" "lam1DS"      "synergism.S"
[65] "mutualism.S"

> dim(ns)         # show dimensions of ns matrix

[1] 74 65

> ns[1:5,1:5]     # show selected results

                          n  L     C  LD      ppr
Marine Coprophagy (oyster) 4  4 0.250 1.0 1.000000
Lake Findley               4  6 0.375 1.5 1.004975
Mirror Lake                5  9 0.360 1.8 1.324718
Lake Wingra                5 10 0.400 2.0 2.000000
Marion Lake                5  9 0.360 1.8 1.324718
```

Given this data frame of network statistics, we can construct interesting plots for further analysis. Here we focus on results of the St. Marks Seagrass ecosystem (Baird *et al.*, 1998).

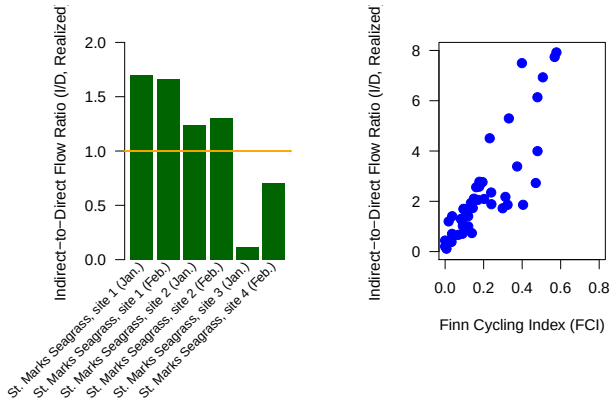

Figure 3: Ratio of Indirect-to-Direct Flow for six ecosystem models (left) and relationship between the Finn Cycling Index and the ratio of Indirect-to-Direct flow in the 74 ecosysetm models.

```
> opar <- par(las=1,mar=c(9,7,2,1),xpd=TRUE,mfrow=c(1,2),oma=c(1,1,0,0))
> ## Number of models
> x=dim(ns)[1]
> m.select <- 26:31
> bp=barplot(ns$ID.F[m.select],ylab="Indirect-to-Direct Flow Ratio (I/D, Realized)",
+          col="darkgreen",border=NA,ylim=c(0,2))
> ## Add labels
> text(bp,-0.05,
+       labels=rownames(ns)[m.select],
+         srt=45,adj=1,cex=0.85)
> opar <- par(xpd=FALSE)
> abline(h=1,col="orange",lwd=2)
> #
> plot(ns$FCI,ns$ID.F,pch=20,col="blue",cex=2,
+       ylab="Indirect-to-Direct Flow Ratio (I/D, Realized)",
+       xlab="Finn Cycling Index (FCI)",
+       xlim=c(0,0.8),ylim=c(0,8))
> ## Remove the plotting parameters
> rm(opar)
```

A strength of this software is the ease with which users can apply ENA to multiple models. We expect that this will simplify users' analytic workflows and reduce the time required to conduct the work.

5. Connecting to Other Useful Packages

Another advantage of building the *enaR* package in R is that it lets ecologists take advantage of other types of network analysis and statistical tools that already exist in R. We highlight three examples here.

5.1. network

enaR uses the network data object introduced in the *network* package (Butts, 2008a). One advantage of using this data object is that analysts can then use the tools for network construction and manipulation that are part of the *network* package. For example, *network* can import network models from Pajek project files, which is another widely used network modeling and analysis software (Batagelj and Mrvar, 2007). The package also includes functions to seamlessly add and delete nodes (edges). It also provides the capability to visualize the network shown previously.

5.2. sna: Social Network Analysis

The *sna* package for Social Network Analysis is bundled in the *statnet* package and uses the same network data object defined in *network*. Thus, the design decision to use the network data object gives users direct access to *sna* tools.

As an example, the *sna* package provides a way of calculating several common centrality measures. Thus, ecologists can now use the *sna* algorithms to determine different types of centrality for their models. This includes betweenness and closeness centrality as follows:

```
> betweenness(oyster)

[1] 0.0 0.0 0.5 3.5 0.0 9.0

> closeness(oyster)

[1] 0.625 0.000 0.000 0.000 0.000 0.000
```

The *sna* package introduced new graphical capabilities as well. For example, it will create a target diagram to visualize the centralities (Figure 4).

```
> m <- m.list[[17]]  # Okefenokee Food Web
> ## Calculate betweenness centrality
> b <- betweenness(m)
> ## Get vertex names
> nms <- m%v%'vertex.names'
> show(nms)

 [1] "Peat decomposers"
 [2] "Detritus decomposers"
 [3] "Nitrogen fixing and nitrifying bacteria"
 [4] "Autotrophic macrophytes"
 [5] "Carnivorous macrophytes"
 [6] "Phytoplankton"
 [7] "Periphyton"
 [8] "Filamentous algae"
 [9] "Herbivorous microinvertebrates"
[10] "Predaceaous microinvertebrates"
```

```
[11] "Saprotrophic microinvertebrates"
[12] "Algae-eating macroinvertebrates"
[13] "Macrophyte-eating macroinvertebrates"
[14] "Microinvertebrate-eating macroinvertebrates"
[15] "Macroinvertebrate-eating macroinvertebrates"
[16] "Vertebrate-eating macroinvertebrates"
[17] "Saprotrophic macroinvertebrates"
[18] "Algae-eating vertebrates"
[19] "Macrophyte-eating vertebrates"
[20] "Microinvertebrate-eating vertebrates"
[21] "Macroinvertebrate-eating vertebrates"
[22] "Vertebrate-eating vertebrates"
[23] "Saprotrophic vertebrates"
[24] "Superficial peat"
[25] "Non-peat detritus"
[26] "Nutrients"
```

```
> ## Exclude less central node names
> nms[b<=(0.1*max(b))] <- NA
> set.seed(2)
> opar <- par(xpd=TRUE,mfrow=c(1,1))
> ## Create target plot showing only
> ## labels of most central nodes
> gplot.target(m,b,
+               circ.lab=FALSE,
+               edge.col="grey",
+               label=nms)
> ## Remove plot settings
> rm(opar)
```

In addition to the node-level measures, *sna* includes graph-level indices.

```
> centralization(oyster, degree)

[1] 0.45

> centralization(oyster,closeness)

[1] 0.75

> centralization(oyster,betweenness)

[1] 0.41
```

5.3. iGraph

The *iGraph* package can also be useful for analyzing network data. Here are a few examples of using the package. Note that some functions in *iGraph* conflict with other functions already defined, so care is required when using *iGraph*.

```
 [1] "Peat decomposers"
 [2] "Detritus decomposers"
 [3] "Nitrogen fixing and nitrifying bacteria"
 [4] "Autotrophic macrophytes"
 [5] "Carnivorous macrophytes"
 [6] "Phytoplankton"
 [7] "Periphyton"
 [8] "Filamentous algae"
 [9] "Herbivorous microinvertebrates"
[10] "Predaceaous microinvertebrates"
[11] "Saprotrophic microinvertebrates"
[12] "Algae-eating macroinvertebrates"
[13] "Macrophyte-eating macroinvertebrates"
[14] "Microinvertebrate-eating macroinvertebrates"
[15] "Macroinvertebrate-eating macroinvertebrates"
[16] "Vertebrate-eating macroinvertebrates"
[17] "Saprotrophic macroinvertebrates"
[18] "Algae-eating vertebrates"
[19] "Macrophyte-eating vertebrates"
[20] "Microinvertebrate-eating vertebrates"
[21] "Macroinvertebrate-eating vertebrates"
[22] "Vertebrate-eating vertebrates"
[23] "Saprotrophic vertebrates"
[24] "Superficial peat"
[25] "Non-peat detritus"
[26] "Nutrients"
```

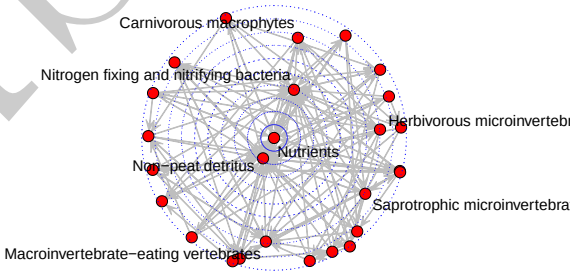

Figure 4: Target plot of node betweenness centrality for the Okefenokee Swamp trophic model.

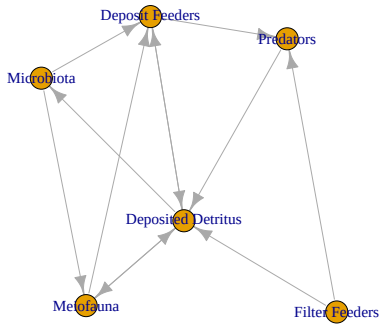

Figure 5: Plot of Oyster reef model using *iGraph*

```
> library(igraph)
> ## The adjacency matrix
> A <- St$A
> ## Creating an iGraph graph
> g <- graph.adjacency(A)
> plot(g)
```

iGraph has a different set of visualization tools and generates a different looking graph (Fig. 5).

```
> ## Betweenness centrality (calculated by iGraph and sna)
> betweenness(g)
```

```
    Filter Feeders          Microbiota          Meiofauna
               0.0                 0.0                0.5
   Deposit Feeders          Predators  Deposited Detritus
               3.5                 0.0                9.0
```

```
> ## Shortest path between any two nodes
> shortest.paths(g)
```

```
                  Filter Feeders Microbiota Meiofauna Deposit Feeders
Filter Feeders                 0          2         2               2
Microbiota                     2          0         1               1
Meiofauna                      2          1         0               1
Deposit Feeders                2          1         1               0
Predators                      1          2         2               1
Deposited Detritus             1          1         1               1
                  Predators Deposited Detritus
Filter Feeders            1                 1
Microbiota                2                 1
Meiofauna                 2                 1
```

```
Deposit Feeders            1                 1
Predators                  0                 1
Deposited Detritus         1                 0

> ## Average path length in the network (graph theory sense)
> average.path.length(g,directed=TRUE)

[1] 1.52

> ## Diameter of the graph
> diameter(g)

[1] 2

> ## Connectivity of the group and sub-components
> vertex.connectivity(g)  # connectivity of a graph (group cohesion)

[1] 0

> subcomponent(g,1,'in')  # subcomponent reachable from 1 along inputs

+ 1/6 vertex, named:
[1] Filter Feeders

> subcomponent(g,2,'in')  # subcomponent reachable from 2 along inputs

+ 6/6 vertices, named:
[1] Microbiota         Deposited Detritus Filter Feeders
[4] Meiofauna          Deposit Feeders    Predators

> subcomponent(g,1,'out') # subcomponent reachable from 1 along outputs

+ 6/6 vertices, named:
[1] Filter Feeders     Predators          Deposited Detritus
[4] Microbiota         Meiofauna          Deposit Feeders

> subcomponent(g,2,'out') # subcomponent reachable from 2 along output

+ 5/6 vertices, named:
[1] Microbiota         Meiofauna          Deposit Feeders
[4] Deposited Detritus Predators

> edge.connectivity(g)

[1] 0
```

```
> ## Detach igraph package
> detach(package:igraph)
```

5.4. EcoNet

The *EcoNet* software is an online, web-interface that provides a tool box for dynamic modeling and ENA analytics (Kazanci, 2007). We have provided a write function that enables *enaR* users to output models for easy input into the *EcoNet* interface. The *EcoNet* package and details on the model input syntax can be found at `http://eco.engr.uga.edu`. Here is an example of how to use the `write.EcoNet` function in *enaR* in your current working directory:

```
> data(oyster)
> write.EcoNet(oyster,file='oyster.txt',mn='oyster_model')
```

6. Conclusion

These examples show how to use the key features of the *enaR* package that enables scientists to perform Ecosystem Network Analysis in R. The vision for this package is that it provides access to ENA algorithms from both the Ulanowicz and Patten Schools to facilitate theoretical synthesis and broader application. In its current form it replicates, updates, and extends the functionality of the NEA.m function (Fath and Borrett, 2006) and replicates much of the main analyses in NETWRK (Ulanowicz and Kay, 1991). Through the connections that *enaR* provides to other R packages users can connect to other network analyses provided by packages, such as *sna* and *iGraph*. There are other R packages that have graph and network analysis tools, like *Bioconductor*, *WGCNA*, *tnet* and *rmangal*, that might also be useful for ecologists. Our aim is for *enaR* to serve as a nexus for the introduction of analyses from the broader field of network theory into ecology. In addition, we would like to invite users to connect, collaborate and contribute to development of ENA theory and *enaR*. Programmers that are interested can visit `https://github.com/SEELab/enaR_development` for more information on how to contribute to development of the *enaR* package.

References

Allesina S, Bondavalli C (2003). "Steady state of ecosystem flow networks: A comparison between balancing procedures." *Ecol. Model.*, **165**, 221–229.

Allesina S, Bondavalli C (2004). "WAND: An ecological network analysis user-friendly tool." *Environ. Model. Softw.*, **19**, 337–340.

Almunia J, Basterretxea G, Aistegui J, Ulanowicz RE (1999). "Benthic–pelagic switching in a coastal subtropical lagoon." *Estuar. Coast. Shelf Sci.*, **49**, 221–232.

Baird D, Asmus H, Asmus R (2004a). "Energy flow of a boreal intertidal ecosystem, the Sylt-Rømø Bight." *Mar. Ecol. Prog. Ser.*, **279**, 45–61.

Baird D, Asmus H, Asmus R (2008). "Nutrient dynamics in the Sylt-Rømø Bight ecosystem, German Wadden Sea: an ecological network analysis approach." *Estuar. Coast. Shelf Sci.*, **80**, 339–356.

Baird D, Christian RR, Peterson CH, Johnson GA (2004b). "Consequences of hypoxia on estuarine ecosystem function: Energy diversion from consumers to microbes." *Ecol. Appl.*, **14**, 805–822.

Baird D, Luczkovich J, Christian RR (1998). "Assessment of spatial and temporal variability in ecosystem attributes of the St Marks National Wildlife Refuge, Apalachee Bay, Florida." *Estuar. Coast. Shelf Sci.*, **47**, 329–349.

Baird D, McGlade JM, Ulanowicz RE (1991). "The comparative ecology of six marine ecosystems." *Philos. Trans. R. Soc. Lond. B*, **333**, 15–29.

Baird D, Milne H (1981). "Energy flow in the Ythan Estuary, Aberdeenshire, Scotland." *Estuar. Coast. Shelf Sci.*, **13**, 455–472.

Baird D, Ulanowicz RE (1989). "The seasonal dynamics of the Chesapeake Bay ecosystem." *Ecol. Monogr.*, **59**, 329–364.

Baird D, Ulanowicz RE, Boynton WR (1995). "Seasonal nitrogen dynamics in Chesapeake Bay—a network approach." *Estuar. Coast. Shelf Sci.*, **41**(2), 137–162.

Barabási AL (2012). "The network takeover." *Nat. Phys.*, **8**(1), 14–16.

Barber MC (1978a). "A Markovian model for ecosystem flow analysis." *Ecol. Model.*, **5**(3), 193–206.

Barber MC (1978b). "A retrospective Markovian model for ecosystem resource flow." *Ecol. Model.*, **5**(2), 125–135.

Batagelj V, Mrvar A (2007). "Pajek: Package for Large Network Analysis." URL http://mrvar.fdv.uni-lj.si/pajek/.

Belgrano A, Scharler UM, Dunne J, Ulanowicz RE (2005). *Aquatic Food Webs: An Ecosystem Approach*. Oxford University Press, New York, NY.

Bondavalli C, Ulanowicz RE (1999). "Unexpected effects of predators upon their prey: The case of the American alligator." *Ecosystems*, **2**, 49–63.

Borgatti SP, Everett MG (2006). "A graph-theoretic perspective on centrality." *Soc. Networks*, **28**, 466–484.

Borrett SR (2013). "Throughflow centrality is a global indicator of the functional importance of species in ecosystems." *Ecol. Indic.*, **32**, 182–196. doi:10.1016/j.ecolind.2013.03.014.

Borrett SR, Christian RR, Ulanowicz RE (2012). "Network Ecology." In AH El-Shaarawi, WW Piegorsch (eds.), *Encyclopedia of Environmetrics*, 2nd edition, pp. 1767–1772. John Wiley & Sons. doi:doi:10.1002/9780470057339.van011.pub2.

Borrett SR, Fath BD, Patten BC (2007). "Functional integration of ecological networks through pathway proliferation." *J. Theor. Biol.*, **245**, 98–111.

Borrett SR, Freeze MA (2011). "Reconnecting Environs to their Environment." *Ecol. Model.*, **222**, 2393–2403. doi:10.1016/j.ecolmodel.2010.10.015.

Borrett SR, Freeze MA, Salas A (2011). "Equivalence of the realized input and output oriented indirect effects metrics in ecological network analysis." *Ecol. Model.*, **222**, 2142–2148. doi:10.1016/j.ecolmodel.2011.04.003.

Borrett SR, Hines DE, Carter M (2015). "Six general ecosystem properties are more intense in biogeochemical cycling networks than food webs." *still seeking a home*.

Borrett SR, Lau MK (2014). "enaR: An R package for Ecological Network Analysis." *Methods. Ecol. Evol.*, **11**, 1206–1213.

Borrett SR, Moody J, Edelmann A (2014). "The rise of network ecology: maps of the topic diversity and scientific collaboration." *Ecol. Model.*, **294**, 111–127. doi:10.1016/j.ecolmodel.2014.02.019.

Borrett SR, Osidele OO (2007). "Environ indicator sensitivity to flux uncertainty in a phosphorus model of Lake Sidney Lanier, USA." *Ecol. Model.*, **200**, 371–383.

Borrett SR, Patten BC (2003). "Structure of pathways in ecological networks: relationships between length and number." *Ecol. Model.*, **170**, 173–184.

Borrett SR, Salas AK (2010). "Evidence for resource homogenization in 50 trophic ecosystem networks." *Ecol. Model.*, **221**, 1710–1716. doi:10.1016/j.ecolmodel.2010.04.004.

Borrett SR, Whipple SJ, Patten BC (2010). "Rapid Development of Indirect Effects in Ecological Networks." *Oikos*, **119**, 1136–1148.

Borrett SR, Whipple SJ, Patten BC, Christian RR (2006). "Indirect effects and distributed control in ecosystems 3. Temporal variability of indirect effects in a seven-compartment model of nitrogen flow in the Neuse River Estuary (USA)—Time series analysis." *Ecol. Model.*, **194**, 178–188.

Brandes U, Erlebach T (eds.) (2005). *Network Analysis: Methodological Foundations*. Springer-Verlag, Berlin, Heidelberg.

Brandes U, Robins G, McCranie A, Wasserman S (2013). "What is network science?" *Network Sci.*, **1**(01), 1–15.

Brylinsky M (1972). "Steady-state sensitivity analysis of energy flow in a marine ecosystem." In BC Patten (ed.), *Systems analysis and simulation in ecology*, volume 2, pp. 81–101. Academic Press.

Butts C (2008a). "network: A Package for Managing Relational Data in R." *J. Stat. Softw.*, **24**.

Butts C (2008b). "Social network analysis with sna." *J. Stat. Softw.*, **24**(6), 1–51.

Caswell H (2001). *Matrix Population Models: Construction, Analysis, and Interpretation.* 2nd edition. Sinauer Associates, Sunderland, Mass.

Chen S, Chen B (2012). "Network environ perspective for urban metabolism and carbon emissions: a case study of Vienna, Austria." *Environ. Sci. Tech.*, **46**(8), 4498–4506.

Christensen V (1995). "Ecosystem maturity—Towards quantification." *Ecol. Model.*, **77**, 3–32.

Christensen V, Pauly D (1992). "Ecopath-II—A software for balancing steady-state ecosystem models and calculating network characteristics." *Ecol. Model.*, **61**, 169–185.

Christensen V, Walters CJ (2004). "Ecopath with Ecosim: Methods, capabilities and limitations." *Ecol. Model.*, **172**, 109–139.

Christian RR, Fores E, Comin F, Viaroli P, Naldi M, Ferrari I (1996). "Nitrogen cycling networks of coastal ecosystems: influence of trophic status and primary producer form." *Ecol. Model.*, **87**, 111–129.

Christian RR, Thomas CR (2003). "Network analysis of nitrogen inputs and cycling in the Neuse River Estuary, North Carolina, USA." *Estuaries*, **26**, 815–828.

Dame RF, Patten BC (1981). "Analysis of energy flows in an intertidal oyster reef." *Mar. Ecol. Prog. Ser.*, **5**, 115–124.

Dunne JA, Williams RJ, Martinez ND (2002). "Food-web structure and network theory: The role of connectance and size." *Proc. Nat. Acad. Sci. USA*, **99**, 12917–.

Edmisten J (1970). "Preliminary studies of the nitrogen budget of a tropical rain forest." In HT Odum, RF Pigeon (eds.), *A Tropical Rain Forest*, TID-24270, pp. 211–215. USAEC Technical Information Center.

Eklöf A, Ebenman B (2006). "Species loss and secondary extinctions in simple and complex model communities." *Journal of Animal Ecology*, **75**(1), 239–246.

Estrada E (2007). "Food webs robustness to biodiversity loss: The roles of connectance, expansibility and degree distribution." *J. Theor. Biol.*, **244**(2), 296 – 307. doi:10.1016/j.jtbi.2006.08.002.

Fann SL, Borrett SR (2012). "Environ centrality reveals the tendency of indirect effects to homogenize the functional importance of species in ecosystems." *J. Theor. Biol.*, **294**, 74–86.

Farkas I, Derenyi I, Barabasi A, Vicsek T (2001). "Spectra of "real-world" graphs: Beyond the semicircle law." *Physical Review E*, **64**(2), 026704.

Fath BD (2004). "Network analysis applied to large-scale cyber-ecosystems." *Ecol. Model.*, **171**, 329–337.

Fath BD (2014). "Quantifying economic and ecological sustainability." *Ocean & Coastal Management.*

Fath BD, Borrett SR (2006). "A Matlab© function for Network Environ Analysis." *Environ. Model. Softw.*, **21**, 375–405.

Fath BD, Patten BC (1998). "Network synergism: emergence of positive relations in ecological systems." *Ecol. Model.*, **107**, 127–143.

Fath BD, Patten BC (1999). "Review of the foundations of network environ analysis." *Ecosystems*, **2**, 167–179.

Fath BD, Scharler UM, Ulanowicz RE, Hannon B (2007). "Ecological network analysis: network construction." *Ecol. Model.*, **208**, 49–55.

Finn JT (1976). "Measures of ecosystem structure and function derived from analysis of flows." *J. Theor. Biol.*, **56**, 363–380.

Finn JT (1980). "Flow analysis of models of the Hubbard Brook ecosystem." *Ecology*, **61**, 562–571.

Freeman L, Borgatti S, White D (1991). "Centrality in valued graphs: A measure of betweenness based on network flow." *Social Networks*, **13**(2), 141–154.

Freeman LC (1979). "Centrality in networks. I. Conceptual clarificaiton." *Soc. Networks*, **1**, 215–239.

Gattie DK, Schramski JR, Borrett SR, Patten BC, Bata SA, Whipple SJ (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.

Handcock M, Hunter D, Butts C, Goodreau S, Morris M (2008). "statnet: Software tools for the representation, visualization, analysis and simulation of network data." *J. Stat. Softw.*, **24**(1), 1548.

Haven DS, Morales-Alamo R (1966). "Aspects of biodeposition by oysters and other invertebrate filter feeders." *Limnol. Oceanogr.*, **11**, 487–498.

Heymans JJ, Baird D (2000). "A carbon flow model and network analysis of the northern Benguela upwelling system, Namibia." *Ecol. Model.*, **126**, 9–32.

Higashi M, Burns TP (1991). *Theoretical studies of ecosystems: The network perspective*. Cambridge University Press, Cambridge.

Higashi M, Patten BC (1989). "Dominance of indirect causality in ecosystems." *Am. Nat.*, **133**, 288–302.

Hines DE, Borrett SR (2014). "A comparison of network, neighborhood, and node levels of analyses in two models of nitrogen cycling in the Cape Fear River Estuary." *Ecological Modelling*, **293**, 210–220.

Hines DE, Lisa JA, Song B, Tobias CR, Borrett SR (2012). "A network model shows the importance of coupled processes in the microbial N cycle in the Cape Fear River estuary." *Estuar. Coast. Shelf Sci.*, **106**, 45–57. doi:10.1016/j.ecss.2012.04.018.

Hines DE, Lisa JA, Song B, Tobias CR, Borrett SR (2015). "Estimating the effects of seawater intrusion on an estuarine nitrogen cycle by comparative network analysis." *Mar. Ecol. Prog. Ser.*, **524**, 137–154.

Hinrichsen U, Wulff F (1998). "Biogeochemical and physical controls of nitrogen fluxes in a highly dynamic marine ecosystem—model and network flow analysis of the Baltic Sea." *Ecol. Model.*, **109**, 165–191.

Ings TC, Montoya JM, Bascompte J, Blüthgen N, Brown L, Dormann CF, Edwards F, Figueroa D, Jacob U, Jones JI, Lauridsen RB, Ledger ME, Lewis HM, Olesen JM, van Veen FJF, Warren PH, Woodward G (2009). "Ecological networks–beyond food webs." *The Journal of animal ecology*, **78**(1), 253–69. ISSN 1365-2656. doi:10.1111/j.1365-2656.2008.01460.x. URL http://www.ncbi.nlm.nih.gov/pubmed/19120606.

Jordan CF, Kline JR, Sasscer DS (1972). "Relative stability of mineral cycles in forest ecosystems." *Am. Nat.*, **106**, 237–253.

Jørgensen SE, Fath BD, Bastianoni S, Marques JC, Müller F, Nielsen S, Patten BC, Tiezzi E, Ulanowicz RE (2007). *A new ecology: Systems perspective*. Elsevier, Amsterdam.

Kazanci C (2007). "EcoNet: A new software for ecological modeling, simulation and network analysis." *Ecol. Model.*, **208**, 3–8.

Latham II LG (2006). "Network flow analysis algorithms." *Ecol. Model.*, **192**(3), 586–600.

Lindeman RL (1942). "The trophic-dynamic aspect of ecology." *Ecology*, **23**, 399–418.

Link J, Overholtz W, O'Reilly J, Green J, Dow D, Palka D, Legault C, Vitaliano J, Guida V, Fogarty M, Brodziak J, Methratta L, Stockhausen W, Col L, Griswold C (2008). "The Northeast US continental shelf Energy Modeling and Analysis exercise (EMAX): Ecological network model development and basic ecosystem metrics." *J. Mar. Syst.*, **74**, 453–474.

Martinez ND (1992). "Constant connectance in community food webs." *Am. Nat.*, pp. 1208–1218.

Miehls ALJ, Mason DM, Frank KA, Krause AE, Peacor SD, Taylor WW (2009a). "Invasive species impacts on ecosystem structure and function: A comparison of Oneida Lake, New York, USA, before and after zebra mussel invasion." *Ecol. Model.*, **220**(22), 3194–3209.

Miehls ALJ, Mason DM, Frank KA, Krause AE, Peacor SD, Taylor WW (2009b). "Invasive species impacts on ecosystem structure and function: A comparison of the Bay of Quinte, Canada, and Oneida Lake, USA, before and after zebra mussel invasion." *Ecol. Model.*, **220**, 3182–3193.

Monaco ME, Ulanowicz RE (1997). "Comparative ecosystem trophic structure of three US mid-Atlantic estuaries." *Mar. Ecol. Prog. Ser.*, **161**, 239–254.

National Research Council, Committee on Network Science for Army Applications (2006). *Network Science*. The National Academies Press, Washington, DC.

Newman M, Barabási AL, Watts DJ (2006). *The structure and dynamics of networks*. Princeton University Press, Princeton, NJ.

Newman MEJ (2001). "Scientific collaboration networks. I. Network construction and fundamental results." *Phys. Rev. E*, **64**(1), 016131.

Niquil N, Chaumillon E, Johnson G, Bertin X, Grami B, David V, Bacher C, Asmus H, Baird D, Asmus R (2012). "The effect of physical drivers on ecosystem indices derived from ecological network analysis: Comparison across estuarine ecosystems." *Estuar. Coast. Shelf Sci.*, **108**, 132–143. doi:{10.1016/j.ecss.2011.12.031}.

Odum HT (1957). "Trophic structure and productivity of Silver Springs, Florida." *Ecol. Monogr.*, **27**, 55–112.

Patrício J, Marques JC (2006). "Mass balanced models of the food web in three areas along a gradient of eutrophication symptoms in the south arm of the Mondego estuary (Portugal)." *Ecol. Model.*, **197**(1), 21–34.

Patten B (1983). "On the quantitative dominance of indirect effects in ecosystems." In WK Lauenroth, GV Skogerboe, M Flug (eds.), *Analysis of Ecological Systems: State-of-the-art in Ecological Modelling*, pp. 27–37. Elsevier, Amsterdam.

Patten BC (1978). "Systems approach to the concept of environment." *Ohio J. Sci.*, **78**, 206–222.

Patten BC (1991). "Network ecology: Indirect determination of the life–environment relationship in ecosystems." In M Higashi, T Burns (eds.), *Theoretical Studies of Ecosystems: The Network Perspective*, pp. 288–351. Cambridge University Press, New York.

Patten BC, Auble GT (1981). "System theory of the ecological niche." *Am. Nat.*, **117**, 893–922.

Patten BC, Bosserman RW, Finn JT, Cale WG (1976). "Propagation of cause in ecosystems." In BC Patten (ed.), *Systems Analysis and Simulation in Ecology, Vol. IV*, pp. 457–579. Academic Press, New York.

Ray S (2008). "Comparative study of virgin and reclaimed islands of Sundarban mangrove ecosystem through network analysis." *Ecol. Model.*, **215**, 207–216. doi:10.1016/j.ecolmodel.2008.02.021.

Richey JE, Wissmar RC, Devol AH, Likens GE, Eaton JS, Wetzel RG, Odum WE, Johnson NM, Loucks OL, Prentki RT, Rich PH (1978). "Carbon flow in four lake ecosystems: A structural approach." *Science*, **202**, 1183–1186.

Rybarczyk H, Elkaim B, Ochs L, Loquet N (2003). "Analysis of the trophic network of a macrotidal ecosystem: the Bay of Somme (Eastern Channel)." *Estuar. Coast. Shelf Sci.*, **58**, 405–421.

Salas AK, Borrett SR (2011). "Evidence for dominance of indirect effects in 50 trophic ecosystem networks." *Ecol. Model.*, **222**, 1192–1204. doi:DOI:10.1016/j.ecolmodel.2010.12.002.

Sandberg J, Elmgren R, Wulff F (2000). "Carbon flows in Baltic Sea food webs — a re-evaluation using a mass balance approach." *J. Mar. Syst.*, **25**, 249–260.

Scharler U (2012). "Ecosystem development during open and closed phases of temporarily open/closed estuaries on the subtropical east coast of South Africa." *Estuar. Coast. Shelf Sci.*, **108**, 119–131.

Scharler U, Fath B (2009). "Comparing network analysis methodologies for consumer–resource relations at species and ecosystems scales." *Ecol. Model.*, **220**(22), 3210–3218.

Scharler UM, Baird D (2005). "A comparison of selected ecosystem attributes of three South African estuaries with different freshwater inflow regimes, using network analysis." *J. Mar. Syst.*, **56**, 283–308.

Schramski JR, Gattie DK, Patten BC, Borrett SR, Fath BD, Thomas CR, Whipple SJ (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." *Ecol. Model.*, **194**, 189–201.

Schramski JR, Gattie DK, Patten BC, Borrett SR, Fath BD, Whipple SJ (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." *Ecol. Model.*, **206**, 18–30.

Schramski JR, Kazanci C, Tollner EW (2011). "Network environ theory, simulation and EcoNet© 2.0." *Environ. Model. Softw.*, **26**, 419–428. doi:10.1016/j.envsoft.2010.10.003.

Small GE, Sterner RW, Finlay JC (2014). "An Ecological Network Analysis of nitrogen cycling in the Laurentian Great Lakes." *Ecol. Model.*, **293**, 150–160.

Soetaert K, Van den Meersche K, van Oevelen D (2009). *limSolve: Solving Linear Inverse Models*. R package version 1.5.1.

Tilly LJ (1968). "The structure and dynamics of Cone Spring." *Ecol. Monogr.*, **38**, 169–197.

Ulanowicz RE (1983). "Identifying the structure of cycling in ecosystems." *Math. Biosci.*, **65**, 219–237.

Ulanowicz RE (1986). *Growth and Development: Ecosystems Phenomenology*. Springer–Verlag, New York.

Ulanowicz RE (1997). *Ecology, the Ascendent Perspective*. Columbia University Press, New York.

Ulanowicz RE (2004). "Quantitative methods for ecological network analysis." *Comput. Biol. Chem.*, **28**, 321–339. doi:http://dx.doi.org/10.1016/j.compbiolchem.2004.09.001.

Ulanowicz RE, Baird D (1999). "Nutrient controls on ecosystem dynamics: The Chesapeake mesohaline community." *J. Mar. Syst.*, **19**, 159–172.

Ulanowicz RE, Bondavalli C, Egnotovich MS (1997). "Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 96: The Cypress Wetland Ecosystem." *Annual Report to the United States Geological Service Biological Resources Division Ref. No. [UMCES]CBL 97-075*, Chesapeake Biological Laboratory, University of Maryland.

Ulanowicz RE, Bondavalli C, Egnotovich MS (1998). "Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 97: The Florida Bay Ecosystem." *Annual Report to the United States Geological Service Biological Resources Division Ref. No. [UMCES]CBL 98-123*, Chesapeake Biological Laboratory, University of Maryland.

Ulanowicz RE, Bondavalli C, Heymans JJ, Egnotovich MS (1999). "Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 98: The Mangrove Ecosystem." *Annual Report to the United States Geological Service Biological Resources Division Ref. No.[UMCES] CBL 99-0073; Technical Report Series No. TS-191-99*, Chesapeake Biological Laboratory, University of Maryland.

Ulanowicz RE, Bondavalli C, Heymans JJ, Egnotovich MS (2000). "Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 99: The Graminoid Ecosystem." *Annual Report to the United States Geological Service Biological Resources Division Ref. No. [UMCES] CBL 00-0176*, Chesapeake Biological Laboratory, University of Maryland.

Ulanowicz RE, Holt RD, Barfield M (2014). "Limits on ecosystem trophic complexity: insights from ecological network analysis." *Ecol. Lett.*, **17**(2), 127–136.

Ulanowicz RE, Kay J (1991). "A package for the analysis of ecosystem flow networks." *Environmental Software*, **6**, 131–142.

Ulanowicz RE, Kemp WM (1979). "Toward canonical trophic aggregations." *Am. Nat.*, pp. 871–883.

Ulanowicz RE, Puccia CJ (1990). "Mixed trophic impacts in ecosystems." *Coenoses*, **5**, 7–16.

Ulanowicz RE, Scharler UM (2008). "Least-inference methods for constructing networks of trophic flows." *Ecol. Model.*, **210**(3), 278–286.

van Oevelen D, Duineveld G, Lavaleye M, Mienis F, Soetaert K, Heip CHR (2009). "The cold-water coral community as a hot spot for carbon cycling on continental margins: A food-web analysis from Rockall Bank (northeast Atlantic)." *Limnology and Oceanography*, **54**(6), 1829.

van Oevelen D, Soetaert K, Middelburg JJ, Herman PMJ, Moodley L, Hamels I, Moens T, Heip CHR (2006). "Carbon flows through a benthic food web: Integrating biomass, isotope and tracer data." *Journal of Marine Research*, **64**(3), 453–482.

van Oevelen D, Van den Meersche K, Meysman FJR, Soetaert K, Middelburg JJ, Vézina AF (2010). "Quantifying food web flows using linear inverse models." *Ecosystems*, **13**(1), 32–45.

Vézina AF, Platt T (1988). "Food web dynamics in the ocean. 1. Best-estimates of flow networks using inverse methods." *Mar. Ecol. Prog. Ser.*, **42**(3), 269–287.

Waide JB, Krebs JE, Clarkson SP, Setzler EM (1974). "A linear system analysis of the calcium cycle in a forested watershed ecosystem." *Prog. Theor. Biol.*, **3**, 261–345.

Wasserman S, Faust K (1994). *Social network analysis: Methods and applications*. Cambridge University Press, Cambridge; New York.

Watts D, Strogatz S (1998). "Collective dynamics of 'small-world' networks." *Nature*, **393**(6684), 440–442.

Whipple SJ, Borrett SR, Patten BC, Gattie DK, Schramski JR, Bata SA (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.

Whipple SJ, Patten BC (1993). "The problem of nontrophic processes in trophic ecology: Toward a network unfolding solution." *J. Theor. Biol.*, **163**, 393–411.

Whipple SJ, Patten BC, Borrett SR (2014). "Indirect effects and distributed control in ecosystems: Comparative network environ analysis of a seven-compartment model of nitrogen storage in the Neuse River Estuary, USA: Time series analysis." *Ecol. Model.*, **293**, 161–186.

Zhang Y, Yang Z, Fath B (2010). "Ecological network analysis of an urban water metabolic system: Model development, and a case study for Beijing." *Sci. Total Env.*, **408**, 4702–4711.

Affiliation:

Matthew K. Lau
Harvard Forest
Harvard University
324 N Main St, Petersham, MA 01366, USA
E-mail: matthewklau@fas.harvard.edu
URL: http://people.fas.harvard.edu/~matthewklau

Stuart R. Borrett
Department of Biology and Marine Biology
University of North Carolina Wilmington
601 South College Road, Wilmington, NC 28403, USA
E-mail: borretts@uncw.edu
URL: http://people.uncw.edu/borretts

Pawandeep Singh
Department of Biology and Marine Biology
University of North Carolina Wilmington
601 South College Road, Wilmington, NC 28403, USA
E-mail: pawan.s@live.co.uk
URL: http://people.uncw.edu/borretts

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd

Table 1: Trophic ecosystem networks (58) included in the *enaR* model library.

Models	Units	$n^{\dagger}$	$C^{\dagger}$	$Input^{\dagger}$	$TST^{\dagger}$	$FCI^{\dagger}$	Reference
Marine Coprophagy (oyster)	kcal m^{-2} yr^{-1}	4	0.25	379	549	0.12	Haven and Morales-Alamo (1966)
Lake Findley	gC m^{-2} yr^{-1}	4	0.38	21	50	0.30	Richey *et al.* (1978)
Mirror Lake	gC m^{-2} yr^{-1}	5	0.36	72	217	0.32	Richey *et al.* (1978)
Lake Wingra	gC m^{-2} yr^{-1}	5	0.40	478	1517	0.40	Richey *et al.* (1978)
Marion Lake	gC m^{-2} yr^{-1}	5	0.36	87	242	0.31	Richey *et al.* (1978)
Cone Springs	kcal m^{-2} yr^{-1}	5	0.32	11819	30626	0.09	Tilly (1968)
Silver Springs	kcal m^{-2} yr^{-1}	5	0.28	21296	29175	0.00	Odum (1957)
English Channel	kcal m^{-2} yr^{-1}	6	0.25	1096	2280	0.00	Brylinsky (1972)
Oyster Reef	kcal m^{-2} yr^{-1}	6	0.33	41	83	0.11	Dame and Patten (1981)
Baie de Somme	mgC m^{-2} d^{-1}	9	0.30	876	2034	0.14	Rybarczyk *et al.* (2003)
Bothnian Bay	gC m^{-2} yr^{-1}	12	0.22	44	183	0.23	Sandberg *et al.* (2000)
Bothnian Sea	gC m^{-2} yr^{-1}	12	0.24	117	562	0.31	Sandberg *et al.* (2000)
Ythan Estuary	gC m^{-2} yr^{-1}	13	0.23	1258	4181	0.24	Baird and Milne (1981)
Sundarban Mangrove (virgin)	kcal m^{-2} yr^{-1}	14	0.22	111317	440931	0.19	Ray (2008)
Sundarban Mangrove (reclaimed)	kcal m^{-2} yr^{-1}	14	0.22	38484	103056	0.05	Ray (2008)
Baltic Sea	mg C m^{-2} d^{-1}	15	0.17	603	1973	0.13	Baird *et al.* (1991)
Ems Estuary	mg C m^{-2} d^{-1}	15	0.19	282	1067	0.32	Baird *et al.* (1991)
Swartkops Estuary 15	mg C m^{-2} d^{-1}	15	0.17	3544	13996	0.47	Baird *et al.* (1991)
Southern Benguela Upwelling	mg C m^{-2} d^{-1}	16	0.23	714	2545	0.31	Baird *et al.* (1991)
Peruvian Upwelling	mg C m^{-2} d^{-1}	16	0.22	14927	33491	0.04	Baird *et al.* (1991)
Crystal River (control)	mg C m^{-2} d^{-1}	21	0.19	7357	15062	0.07	Ulanowicz (1986)
Crystal River (thermal)	mg C m^{-2} d^{-1}	21	0.14	6018	12032	0.09	Ulanowicz (1986)
Charca de Maspalomas Lagoon	mg C m^{-2} d^{-1}	21	0.12	1486230	6010331	0.18	Almunia *et al.* (1999)
Northern Benguela Upwelling	mg C m^{-2} d^{-1}	24	0.21	2282	6611	0.05	Heymans and Baird (2000)
Swartkops Estuary	mg C m^{-2} d^{-1}	25	0.17	2859	8949	0.27	Scharler and Baird (2005)
Sunday Estuary	mg C m^{-2} d^{-1}	25	0.16	4440	11937	0.22	Scharler and Baird (2005)
Kromme Estuary	mg C m^{-2} d^{-1}	25	0.16	2571	11087	0.38	Scharler and Baird (2005)
Okefenokee Swamp	g dw m^{-2} y^{-1}	26	0.20	2533	12855	0.48	Whipple and Patten (1993)
Neuse Estuary (early summer 1997)	mg C m^{-2} d^{-1}	30	0.09	4385	13827	0.12	Baird *et al.* (2004b)
Neuse Estuary (late summer 1997)	mg C m^{-2} d^{-1}	30	0.11	4639	13035	0.13	Baird *et al.* (2004b)
Neuse Estuary (early summer 1998)	mg C m^{-2} d^{-1}	30	0.09	4568	14025	0.12	Baird *et al.* (2004b)
Neuse Estuary (late summer 1998)	mg C m^{-2} d^{-1}	30	0.10	5641	15031	0.11	Baird *et al.* (2004b)
Gulf of Maine	g ww m^{-2} yr^{-1}	31	0.35	5053	18381	0.15	Link *et al.* (2008)
Georges Bank	g ww m^{-2} yr^{-1}	31	0.35	4380	16889	0.18	Link *et al.* (2008)
Middle Atlantic Bight	g ww m^{-2} yr^{-1}	32	0.37	4869	17916	0.18	Link *et al.* (2008)
Narragansett Bay	mgC m^{-2} yr^{-1}	32	0.15	693845	3917246	0.51	Monaco and Ulanowicz (1997)
Southern New England Bight	g ww m^{-2} yr^{-1}	33	0.35	4717	17597	0.16	Link *et al.* (2008)
Chesapeake Bay	mg C m^{-2} yr^{-1}	36	0.09	888791	3227453	0.19	Baird and Ulanowicz (1989)
Mondego Estuary (*Zostera* sp. Meadows)	g AFDW m^{-2} yr^{-1}	43	0.19	4030	6822	0.03	Patrício and Marques (2006)
St. Marks Seagrass, site 1 (Jan.)	mg C m^{-2} d^{-1}	51	0.08	514	1315	0.13	Baird *et al.* (1998)
St. Marks Seagrass, site 1 (Feb.)	mg C m^{-2} d^{-1}	51	0.08	601	1590	0.11	Baird *et al.* (1998)
St. Marks Seagrass, site 2 (Jan.)	mg C m^{-2} d^{-1}	51	0.07	602	1383	0.09	Baird *et al.* (1998)
St. Marks Seagrass, site 2 (Feb.)	mg C m^{-2} d^{-1}	51	0.08	800	1921	0.08	Baird *et al.* (1998)
St. Marks Seagrass, site 3 (Jan.)	mg C m^{-2} d^{-1}	51	0.05	7809	12651	0.01	Baird *et al.* (1998)
St. Marks Seagrass, site 4 (Feb.)	mg C m^{-2} d^{-1}	51	0.08	1432	2865	0.04	Baird *et al.* (1998)
Sylt-Rømø Bight	mg C m^{-2} d^{-1}	59	0.08	683448	1781028	0.09	Baird *et al.* (2004a)
Graminoids (wet)	g C m^{-2} yr^{-1}	66	0.18	6272	13676	0.02	Ulanowicz *et al.* (2000)
Graminoids (dry)	g C m^{-2} yr^{-1}	66	0.18	3472	7519	0.04	Ulanowicz *et al.* (2000)
Cypress (wet)	g C m^{-2} yr^{-1}	68	0.12	1418	2571	0.04	Ulanowicz *et al.* (1997)
Cypress (dry)	g C m^{-2} yr^{-1}	68	0.12	1035	1919	0.04	Ulanowicz *et al.* (1997)
Lake Oneida (pre-ZM)	g C m^{-2} yr^{-1}	74	0.22	1034	1697	0.00	Miehls *et al.* (2009a)
Lake Oneida (post-ZM)	g C m^{-2} yr^{-1}	76	0.22	810	1462	0.00	Miehls *et al.* (2009a)
Bay of Quinte (pre-ZM)	g C m^{-2} yr^{-1}	74	0.21	984	1509	0.00	Miehls *et al.* (2009b)
Bay of Quinte (post-ZM)	g C m^{-2} yr^{-1}	80	0.21	1129	2039	0.01	Miehls *et al.* (2009b)
Mangroves (wet)	g C m^{-2} yr^{-1}	94	0.15	1531	3265	0.10	Ulanowicz *et al.* (1999)
Mangroves (dry)	g C m^{-2} yr^{-1}	94	0.15	1531	3272	0.10	Ulanowicz *et al.* (1999)
Florida Bay (wet)	mg C m^{-2} yr^{-1}	125	0.12	738	2720	0.14	Ulanowicz *et al.* (1998)
Florida Bay (dry)	mg C m^{-2} yr^{-1}	125	0.13	547	1778	0.08	Ulanowicz *et al.* (1998)

† n is the number of nodes in the network model, $C = L/n^2$ is the model connectance when L is the number of direct links or energy–matter transfers, $Input = \sum z_i$ is the total amount of energy–matter flowing into the system, $TST = \sum\sum f_{ij} + \sum z_i$ is the total system throughflow, and FCI is the Finn Cycling Index (Finn, 1980). Flow based network statistics ($Input$, TST, and FCI) were calculated after models were balanced using the AVG2 algorithm.

Table 2: Biogeochemical ecosystem networks (42) included in the *enaR* model library.

Model	Units	$n^\dagger$	$C^\dagger$	$Input^\dagger$	$TST^\dagger$	$FCI^\dagger$	Reference
Hubbard Brook (Waide)	kg Ca Ha^{-1} yr^{-1}	4	0.25	11	168	0.76	Waide *et al.* (1974)
Hardwood Forest, NH	kg Ca Ha^{-1} yr^{-1}	4	0.31	11	200	0.80	Jordan *et al.* (1972)
Douglas Fir Forest, WA	kg Ca Ha^{-1} yr^{-1}	4	0.31	4	54	0.74	Jordan *et al.* (1972)
Douglas Fir Forest, WA	kg K Ha^{-1} yr^{-1}	4	0.31	0	45	0.97	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Ca Ha^{-1} yr^{-1}	4	0.31	43	274	0.57	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg K Ha^{-1} yr^{-1}	4	0.31	20	433	0.86	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Mg Ha^{-1} yr^{-1}	4	0.31	10	70	0.58	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Cu Ha^{-1} yr^{-1}	4	0.31	0	2	0.37	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Fe Ha^{-1} yr^{-1}	4	0.31	0	7	0.95	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Mn Ha^{-1} yr^{-1}	4	0.38	0	7	0.98	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Na Ha^{-1} yr^{-1}	4	0.31	64	140	0.24	Jordan *et al.* (1972)
Puerto Rican Rain Forest	kg Sr Ha^{-1} yr^{-1}	4	0.31	0	1	0.71	Jordan *et al.* (1972)
Tropical Rain Forest	g N m^{-2} d^{-1}	5	0.24	10	71	0.48	Edmisten (1970)
Neuse River Estuary (AVG)	mmol N m^{-2} $season^{-1}$	7	0.45	795	41517	0.89	Christian and Thomas (2003)
Neuse River Estuary (Spring 1985)	mmol N m^{-2} $season^{-1}$	7	0.45	133	9120	0.91	Christian and Thomas (2003)
Neuse River Estuary (Summer 1985)	mmol N m^{-2} $season^{-1}$	7	0.45	119	20182	0.96	Christian and Thomas (2003)
Neuse River Estuary Fall 1985)	mmol N m^{-2} $season^{-1}$	7	0.45	181	8780	0.88	Christian and Thomas (2003)
Neuse River Estuary Winter 1986)	mmol N m^{-2} $season^{-1}$	7	0.43	187	6880	0.85	Christian and Thomas (2003)
Neuse River Estuary (Spring 1986)	mmol N m^{-2} $season^{-1}$	7	0.45	128	12915	0.94	Christian and Thomas (2003)
Neuse River Estuary (Summer 1986)	mmol N m^{-2} $season^{-1}$	7	0.45	165	11980	0.91	Christian and Thomas (2003)
Neuse River Estuary (Fall 1986)	mmol N m^{-2} $season^{-1}$	7	0.45	100	9863	0.94	Christian and Thomas (2003)
Neuse River Estuary (Winter 1987)	mmol N m^{-2} $season^{-1}$	7	0.45	691	7907	0.62	Christian and Thomas (2003)
Neuse River Estuary (Spring 1987)	mmol N m^{-2} $season^{-1}$	7	0.45	334	11533	0.84	Christian and Thomas (2003)
Neuse River Estuary (Summer 1987)	mmol N m^{-2} $season^{-1}$	7	0.45	90	15621	0.96	Christian and Thomas (2003)
Neuse River Estuary (Fall 1987)	mmol N m^{-2} $season^{-1}$	7	0.45	85	7325	0.93	Christian and Thomas (2003)
Neuse River Estuary (Winter 1988)	mmol N m^{-2} $season^{-1}$	7	0.45	171	8680	0.89	Christian and Thomas (2003)
Neuse River Estuary (Spring 1988)	mmol N m^{-2} $season^{-1}$	7	0.45	176	6898	0.85	Christian and Thomas (2003)
Neuse River Estuary (Summer 1988)	mmol N m^{-2} $season^{-1}$	7	0.45	132	16814	0.95	Christian and Thomas (2003)
Neuse River Estuary (Fall 1988)	mmol N m^{-2} $season^{-1}$	7	0.45	128	5732	0.87	Christian and Thomas (2003)
Neuse River Estuary (Winter 1989)	mmol N m^{-2} $season^{-1}$	7	0.45	291	5739	0.75	Christian and Thomas (2003)
Cape Fear River Estuary (Oligohaline)	nmol N cm^{-3} d^{-1}	8	0.36	3802	7088	0.20	Hines *et al.* (2012)
Cape Fear River Estuary (Polyhaline)	nmol N cm^{-3} d^{-1}	8	0.36	3068	5322	0.17	Hines *et al.* (2015)
Lake Lanier (AVG)	mg P m^{-2} day^{-1}	11	0.21	95	749	0.40	Borrett and Osidele (2007)
Baltic Sea	mg N m^{-3} day^{-1}	16	0.15	2348	44510	0.67	Hinrichsen and Wulff (1998)
Chesapeake Bay	mg N m^{-2} yr^{-1}	36	0.12	73430	484325	0.33	Baird *et al.* (1995)
Chesapeake Bay	mg P m^{-2} yr^{-1}	36	0.12	9402	101091	0.51	Ulanowicz and Baird (1999)
Chesapeake Bay (Winter)	mg P m^{-2} $season^{-1}$	36	0.08	1009	11926	0.53	Ulanowicz and Baird (1999)
Chesapeake Bay (Spring)	mg P m^{-2} $season^{-1}$	36	0.10	1932	27325	0.57	Ulanowicz and Baird (1999)
Chesapeake Bay (Summer)	mg P m^{-2} $season^{-1}$	36	0.12	4184	42935	0.46	Ulanowicz and Baird (1999)
Chesapeake Bay (Fall)	mg P m^{-2} $season^{-1}$	36	0.10	2276	18904	0.40	Ulanowicz and Baird (1999)
Sylt-Rømø Bight	mg N m^{-2} yr^{-1}	59	0.09	99613	363693	0.23	Baird *et al.* (2008)
Sylt-Rømø Bight	mg P m^{-2} yr^{-1}	59	0.09	2508	57739	0.66	Baird *et al.* (2008)

† n is the number of nodes in the network model, $C = L/n^2$ is the model connectance when L is the number of direct links or energy–matter transfers, $Input = \sum z_i$ is the total amount of energy–matter flowing into the system, $TST = \sum\sum f_{ij} + \sum z_i$ is the total system throughflow, and FCI is the Finn Cycling Index (Finn, 1980). Flow based network statistics ($Input$, TST, and FCI) were calculated after models were balanced using the AVG2 algorithm.

Table 3: Primary Ecosystem Network Analysis algorithms in *enaR*.

Analysis	**Function Name**	**School**
Structure	`enaStructure`	foundational, Patten
Flow	`enaFlow`	foundational, Patten
Ascendency	`enaAscendency`	Ulanowicz
Storage	`enaStorage`	Patten
Utility	`enaUtility`	Patten
Mixed Trophic Impacts	`enaMTI`	Ulanowicz
Control	`enaControl`	Patten
Environ	`enaEnviron`	Patten
Cycle Basis	`enaCycle`	Ulanowicz
Canonical Trophic Aggregation	`enaTroAgg`	Ulanowicz

Table 4: Resultant matrices and network statistics returned by the `enaStructure` function in *enaR*.

Label	**Description**
Matrices	
A	$n \times n$ adjacency matrix
Network statistics	
n	number of nodes
L	number of directed edges
C	connectance ($C = L/n^2$); the proportion of possible directed edges connected.
LD	Link Density (L/n)
ppr	estimated rate of pathway proliferation (Borrett and Patten, 2003)
lam1A	dominant eigenvalue of A ($lambda_1(\mathbf{A})$), which is the asymptotic rate of pathway proliferation (Borrett *et al.*, 2007)
mlam1A	multiplicity of the dominant eigenvalue (number of times repeated)
rho	damping ratio, an indicator of how quickly $[a_{ij}]^{(m)}/[a_{ij}]^{(m-1)}$ goes to $lam_1(\mathbf{A})$ (Caswell, 2001, , p. 95)
R	distance of $lam_1(\mathbf{A})$ from the bulk of the eigen spectrum (Farkas *et al.*, 2001)
d	difference between dominant eigenvalue and link density (expected value for random graph)
no.scc	number of strongly connected components (SCC)
no.scc.big	number of SCC with more than one node
pscc	fraction of network nodes included in a big SCC

Table 5: Matrices and network statistics returned by the `enaFlow` function in *enaR*.

***enaR* label**	**Description**
Matrices	
T	$n \times 1$ vector of node throughflows (M L^{-2} or $^{-3}$ T^{-1})
G	output-oriented direct throughflow intensity matrix
GP	input-oriented direct throughflow intensity matrix
N	output-oriented integral throughflow intensity matrix
NP	input-oriented integral throughflow intensity matrix
Network statistics	
Input	Total input boundary flow
TST	Total System ThroughFLOW
TSTp	Total System ThroughPUT
APL	Average Path Length (Finn, 1976)
FCI	Finn Cycling Index (Finn, 1980)
BFI	Boundary Flow Intensity, *Boundary/TST*
DFI	Direct Flow Intensity, *Direct/TST*
IFI	Indirect Flow Intensity, *Indirect/TST* (Borrett *et al.*, 2006)
ID.F	Ratio of Indirect to Direct Flow Borrett and Freeze (2011); Borrett *et al.* (2011)
ID.F.I	input oriented ratio of indirect to direct flow intensity (as in Fath and Borrett, 2006)
IF.F.O	output oriented ratio of indirect to direct flow intensity (as in Fath and Borrett, 2006)
HMG.F.I	input oriented network homogenization to direct flow intensity
HMG.F.O	output oriented network homogenization to direct flow intensity
AMP.F.I	input oriented network amplification
AMP.F.O	output oriented network amplification
mode0.F	Boundary Flow
mode1.F	Internal First Passage Flow
mode2.F	Cycled Flow
mode3.F	Dissipative Equivalent to mode1.F
mode4.F	Dissipative Equivalent to mode0.F

Table 6: Graph-level network statistics returned by the *enaR* `enaAscendency` function (see Ulanowicz, 1986, 1997, for interpretations).

Label	**Description**
AMI	average mutual information (bits)
ASC	ascendency, AMI $\times$ TSTp
OH	overhead
CAP	capacity
ASC.CAP	ascendency-to-capacity ratio (dimensionless)
OH.CAP	overhead-to-capacity ratio (dimensionless)
robustness	robustness of the network as in Fath (2014)
ELD	effective link density of the network Ulanowicz *et al.* (2014)
TD	trophic depth of the network as in Ulanowicz *et al.* (2014)

Table 7: Matrices and graph-level network statistics returned by the *enaR* `enaStorage` function.

Label	**Description**
Matrices	
X	$n \times 1$ vector of storage values [M L^{-2}]
C	$n \times n$ donor-storage normalized output-oriented direct flow intensity matrix (T^{-1})
P	$n \times n$ storage-normalized output-oriented direct flow matrix (dimensionless)
S	$n \times n$ donor-storage normalized output-oriented integral flow intensity matrix (T^{-1})
Q	$n \times n$ output-oriented integral flow intensity matrix (dimensionless)
CP	$n \times n$ recipient-storage normalized input-oriented direct flow intensity matrix (T^{-1})
PP	$n \times n$ storage-normalized input-oriented direct flow matrix (dimensionless)
SP	$n \times n$ donor-storage normalized input-oriented integral flow intensity matrix (T^{-1})
QP	$n \times n$ input-oriented integral flow intensity matrix (dimensionless)
dt	discrete time step
Network statistics	
TSS	Total System Storage
CIS	Storage Cycling Index
BSI	Boundary Storage Intensity
DSI	Direct Storage Intensity
ISI	Indirect Storage Intensity
ID.S	Ratio of Indirect-to-Direct storage (realized)
ID.S.I	storage-based input-oriented indirect-to-direct ratio (as in Fath and Borrett, 2006)
ID.S.O	storage-based input-oriented indirect-to-direct ratio (as in Fath and Borrett, 2006)
HMG.S.I	input-oriented storage network homogenization
HMG.S.O	output-oriented storage network homogenization
AMP.S.I	input-oriented storage network amplification
AMP.S.O	output-oriented storage network amplification
mode0.S	Storage from Boundary Flow
mode1.S	Storage from Internal First Passage Flow
mode2.S	Storage from Cycled Flow
mode3.S	Dissipative Equivalent to mode1.S
mode4.S	Dissipative Equivalent to mode0.S

Table 8: Matrices and graph-level network statistics returned by the *enaR* `enaUtility` function.

Label	Description
Matrices	
$D_{n\times n}$	throughflow-normalized direct utility intensity (dimensionless)
$U_{n\times n}$	integral flow utility (dimensionless)
$Y_{n\times n}$	integral flow utility scaled by original throughflow (M $L^{-2 \text{ or } -3}$ T^{-1})
$DS_{n\times n}$	storage-normalized direct utility intensity (dimensionless)
$US_{n\times n}$	integral storage utility (dimensionless)
$YS_{n\times n}$	integral storage utility scaled by original throughflow (M $L^{-2 \text{ or } -3}$ T^{-1})
Network Statistics	
lam1D	dominant eigenvalue of D
synergism.F	benefit-cost ratio or network synergism (flow)
mutualism.F	positive to negative interaction ratio or network mutualism (flow)
lam1DS	dominant eigenvalue of DS
synergism.S	benefit-cost ratio or network synergism (storage)
mutualism.S	positive to negative interaction ratio or network mutualism (storage)

Table 9: Matrices returned by the *enaR* `enaControl` function, which are based on (Dame and Patten, 1981; Patten and Auble, 1981; Schramski *et al.*, 2006, 2007).

Label	Description
Matrices	
$CN_{n\times n}$	Control matrix using flow values
$CQ_{n\times n}$	Control matrix using storage values
$CR_{n\times n}$	Schramski's Control Ratio Matrix
$CD_{n\times n}$	Schramski's Control Difference Matrix
$sc_{n\times 1}$	Schramski's System Control vector

Table 10: Matrices returned by the *enaR* `enaMTI` function, which are based on (Ulanowicz and Puccia, 1990).

Label	Description
Matrices	
$G_{n\times n}$	positive effect of prey on its predator
$F_{n\times n}$	negative impact of the predator on its prey
$Q_{n\times n}$	direct net impact of one node on another
$M_{n\times n}$	total impact of i on j (direct and indirect)

Table 11: Data frames, matrices and graph-level network statistics returned by the *enaR* `enaCycle` function, which is based on (Ulanowicz, 1983).

Label	**Description**
Data frames	
Table.cycle	Data frame of cycles in the network. Up to 50 cycles are returned per nexus.
Table.nexus	Data frame with details of the disjoint nexuses present in the network
Matrices	
$\text{CycleDist}_{n\times 1}$	Vector of flows cycling in loops of increasing length (i.e., 1, 2, ...).
$\text{NormDist}_{n\times 1}$	Vector of Cycle Distributions normalized by the total system throughput
$\text{ResidualFlows}_{n\times n}$	Matrix of straight-through flows or the underlying acyclic graph
$\text{AggregatedCycles}_{n\times n}$	Matrix of all the cycled flows or the underlying cyclic graph
Network Statistics	
NCYCS	Number of cycles detected in the network
NNEX	Number of disjoint nexuses detected in the network
CI	Cycling index of the network based on flow matrix

Table 12: Matrices and graph-level network statistics returned by the *enaR* `enaTroAgg` function, which are based on Ulanowicz and Kemp (1979).

Label	Description
Matrices	
$A_{nl \times nl}$	Lindeman transformation matrix that apportions nodes to integer trophic levels
$ETL_{n \times 1}$	Vector of the effective trophic levels of different nodes
$M.Flow_{nl \times 1}$	Migratory flows in living nodes (if present)
$CI_{n \times 1}$	Vector of canonical inputs to integer trophic levels (if migratory flows present)
$CE_{n \times 1}$	Canonical Exports. Vector of exports from Integer trophic levels
$CR_{n \times 1}$	Canonical Respirations. Vector of respiration from Integer trophic levels
$GC_{nl \times 1}$	Grazing Chain. Vector of inputs to Integer trophic levels from preceding level
$RDP_{nl \times 1}$	Vector of returns from each level to the detrital pool
$LS_{nl \times 1}$	Vector representing the Lindeman Spine
$TE_{nl \times 1}$	Vector of the trophic efficiencies for integer trophic levels
Network Statistics	
Detritivory	Flow from the detrital pool (non-living nodes) to the second trophic level
DetritalInput	Exogenous inputs to the detrital pool
DetritalCirc	internal circulation within the detrital pool
NCYCS	number of feeding cycles removed from the network
NNEX	number of disjoint nexuses detected for the feeding cycles
CI	cycling index of the living component of the network based on flow matrix