

## Ecological Dynamics

### Lab 4: Food webs

#### Background

- Network science has recently emerged as a critical approach for studying complex ecosystems.
- Networks can be used to represent the feeding relationships between species.
- Understanding the structure and robustness of natural food webs is critical for basic and applied research.

#### Objectives

- Learn to use the **igraph** package to analyze and plot food web networks.
- Compare the statistical properties of empirical and model food webs.
- Use numerical simulations to determine the robustness of empirical and model food webs.

#### Instructions

- Launch RStudio, open a new script file and save it as **lab4-yourLastName.R**.
- Complete the activities below by adding the appropriate commands to your R script file.
- Embed your answers as comments in the R script file.
- At the end of your session, save your file onto a flash drive (files saved on lab computers are wiped when you log out).

#### Task 1: Analyzing empirical food webs

We will analyze nine well-resolved empirical food webs to determine their topological features and similarities.

1. These food webs are stored in plain text files as adjacency matrices. Before downloading them, write a function called **remove.disconnected** to remove any disconnected node or species from the food web. By disconnected, we mean any species that neither feeds on another species nor is being fed on by another species.

```
remove.disconnected <- function(adj) {  
  adj <- as.matrix(adj)  
  # Remove species that neither consume nor are consumed  
  empty <- which(rowSums(adj) == 0 & colSums(adj) == 0)  
  adj <- adj[-empty, -empty]  
  return(adj)  
}
```

2. Using this code, download and clean the food web data by executing the following lines of code:

```

benguela <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/benguela.txt"))
britishgrasslands <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/british-grasslands.txt"))
cantoncreek <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/canton-creek.txt"))
chesapeakebay <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/chesapeake-bay.txt"))
coachella <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/coachella.txt"))
scotchbroom <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/scotch-broom.txt"))
skipwithpond <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/skipwith-pond.txt"))
stmarks <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/st-marks.txt"))
stonystream <- remove.disconnected(read.table("http://faraway.neu.edu/data/foodwebs/stony-stream.txt"))

```

- Write a function that takes an adjacency matrix as an argument and returns a named vector containing the number of species  $S$ , the number of links  $L$ , the connectance  $C$ , the average degree `avg.degree`, the average or characteristic path length `l.G`, and the proportion of basal, intermediate and top species. The `igraph` package provides functions called `degree` and `average.path.length` to compute the average degree and characteristic path length of a graph  $g$ . To use these functions, you will need to use function `graph.adjacency` to convert your adjacency matrix into a graph.

```

library(igraph)
summ.stats <- function(adj) {
  # Create graph from adjacency matrix
  graph <- graph.adjacency(adj)
  S <- NROW(adj)
  L <- sum(adj)
  C <- L/S^2
  l.G <- average.path.length(graph)
  basal <- sum(colSums(adj) == 0)/S
  top <- sum(rowSums(adj) == 0)/S
  inter <- sum(colSums(adj) != 0 & rowSums(adj) != 0)/S
  avg.degree <- mean(degree(graph))
  result <- c(S = S, L = L, C = C, avg.degree = avg.degree, l.G = l.G,
    basal = basal, inter = inter, top = top)
  return(result)
}

```

- Now it's time to test your function. Create a 10-row  $\times$  8-column matrix called `summ.foodwebs` and use it to store the summary statistics of all 9 food webs, along with the mean of each statistic across all food webs (in the last row).

```

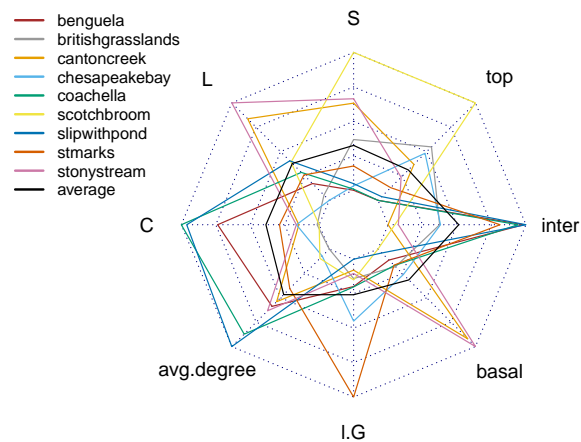
summ.foodwebs <- matrix(nrow = 10, ncol = 8)
colnames(summ.foodwebs) <- c("S", "L", "C", "avg.degree", "l.G", "basal",
  "inter", "top")
rownames(summ.foodwebs) <- c("benguela", "britishgrasslands", "cantoncreek",
  "chesapeakebay", "coachella", "scotchbroom", "skipwithpond", "stmarks",
  "stonystream", "average")
summ.foodwebs[1, ] <- summ.stats(benguela)
summ.foodwebs[2, ] <- summ.stats(britishgrasslands)
summ.foodwebs[3, ] <- summ.stats(cantoncreek)
summ.foodwebs[4, ] <- summ.stats(chesapeakebay)
summ.foodwebs[5, ] <- summ.stats(coachella)
summ.foodwebs[6, ] <- summ.stats(scotchbroom)
summ.foodwebs[7, ] <- summ.stats(skipwithpond)
summ.foodwebs[8, ] <- summ.stats(stmarks)
summ.foodwebs[9, ] <- summ.stats(stonystream)
summ.foodwebs[10, ] <- colMeans(summ.foodwebs[1:9, ])
summ.foodwebs <- as.data.frame(summ.foodwebs)

```

- Finally, it's time to visualize your results. To do so, we will use a radar plot using function

`radarchart` from package `fmsb`. Check the help file to customize your plot. Note that you will need to set `minmax=FALSE` in the `radarchart` function in order for your data to appear properly.

```
library(fmsb)
radarchart(as.data.frame(summ.foodwebs), axistype = 6, pty = 32, plty = 1,
  axislabcol = "grey", na.itp = FALSE, seg = 4, maxmin = F, pcol = c("brown",
    cb[c(2:9, 1)]))
legend(x = "topleft", legend = c("benguela", "britishgrasslands", "cantoncreek",
  "chesapeakebay", "coachella", "scotchbroom", "slipwithpond", "stmarks",
  "stonystream", "average"), col = c("brown", cb[c(2:9, 1)]), lty = 1,
  lwd = 2, bty = "n", cex = 0.8)
```



6. To quantify some of the relationships between the various metrics, use the `cor` function to compute the correlation matrix. What do the results suggest regarding the various food web metrics used to assess the structure of food webs in the literature?

```
cor(summ.foodwebs[1:9, ])
```

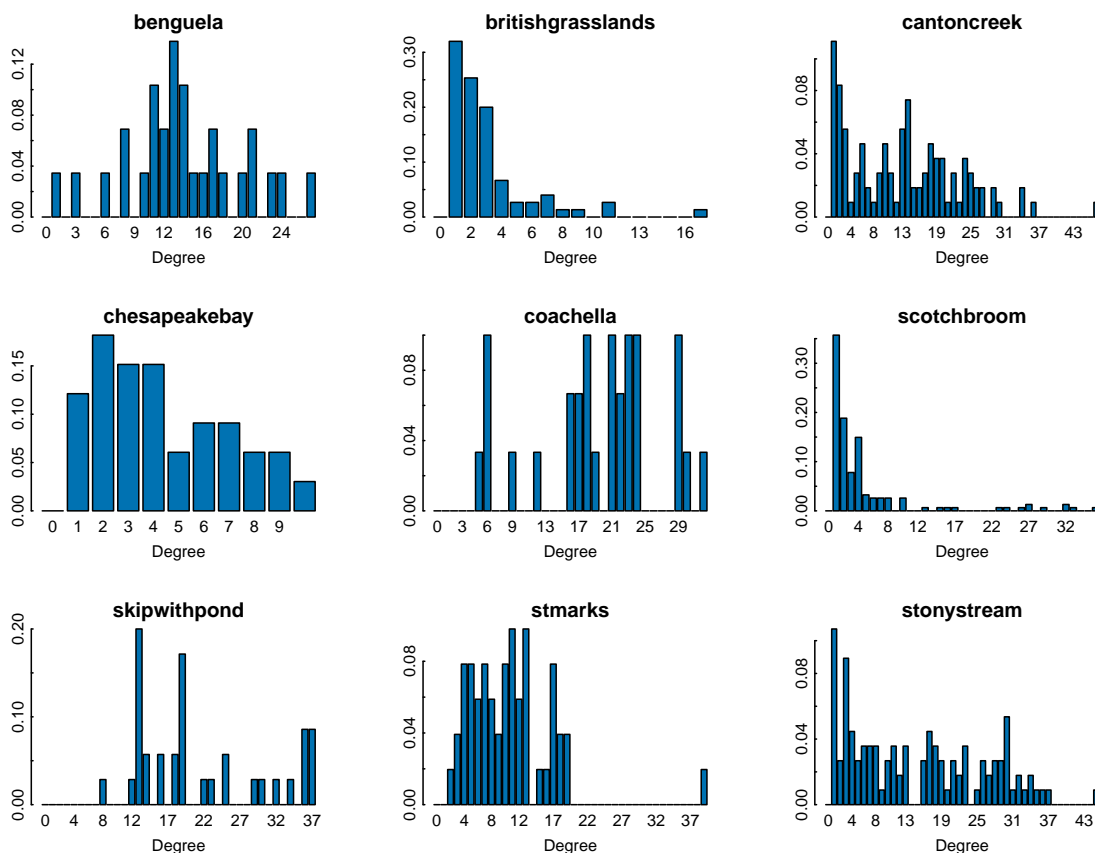
	S	L	C	avg.degree	l.G
S	1.0000000	0.57445574	-0.6982925	-0.3696355	-0.2741019
L	0.5744557	1.0000000	-0.1380066	0.4087143	-0.3533551
C	-0.6982925	-0.13800655	1.0000000	0.8375983	-0.1613722
avg.degree	-0.3696355	0.40871429	0.8375983	1.0000000	-0.2688982
l.G	-0.2741019	-0.35335511	-0.1613722	-0.2688982	1.0000000
basal	0.3925090	0.82172840	-0.3482817	0.1297883	-0.2136883
inter	-0.8547229	-0.50403401	0.8527194	0.5393763	0.2362028
top	0.7504075	-0.05125023	-0.7848760	-0.7893202	-0.1193577
	basal	inter	top		
S	0.39250903	-0.8547229	0.75040749		
L	0.82172840	-0.5040340	-0.05125023		
C	-0.34828172	0.8527194	-0.78487596		
avg.degree	0.12978826	0.5393763	-0.78932024		
l.G	-0.21368831	0.2362028	-0.11935774		
basal	1.00000000	-0.6121918	-0.06387195		
inter	-0.61219175	1.0000000	-0.74999291		
top	-0.06387195	-0.7499929	1.00000000		

**Answer:**

There are strong correlation between many of the food web metrics. For instance, connectance is strongly correlated to average degree, the proportion of top species is strongly correlated to the number of species  $S$ , and the proportion of basal species is strongly correlated to the number of links  $L$ . These correlations suggest that these metrics are non-independent.

7. Generate a  $3 \times 3$  figure with each panel containing a histogram of the degree distribution for each food web. Use function `degree.distribution` to compute the frequency of species with  $0, 1, 2, \dots, S - 1$  degrees and use function `barplot` to plot it. What do these patterns suggest?

```
foodwebs <- list(benguela, britishgrasslands, cantoncreek, chesapeakebay,
  coachella, scotchbroom, skipwithpond, stmarks, stonystream)
par(mfrow = c(3, 3))
for (i in 1:length(foodwebs)) {
  degree <- degree.distribution(graph.adjacency(foodwebs[[i]]))
  barplot(degree, names.arg = 0:(length(degree) - 1), xlab = "Degree",
    col = cb["blue"], main = rownames(summ.foodwebs)[i])
}
```



**Answer:**

The distributions are mostly right-skewed, with an overabundance of small values and an under-abundance of large values. This means that most species are linked to only a few species (e.g., specialists). Only a few species are interacting with many other species (e.g., generalists).

**Task 2: Comparing empirical and model food webs**

We are now going to compare the statistical features and robustness of empirical and model food webs.

1. Begin by downloading the code to generate food webs using the `niche`, `cascade` and `random` models:

```
# This code provides functions niche, cascade and random
source("http://faraway.neu.edu/data/foodweb.models.R")
```

2. Using the number of species  $S$  and connectance  $C$  from each empirical food web, run `nreps=1000` replicate simulations of each food web model and use the function you created in Task 1 to compute the summary statistics. Then, average these statistics across all replicates using the `aggregate` function. Compute the correlation matrix between the statistics generated from the empirical and model foodwebs. Interpret the results.

```
# Remove row for average
summ.foodwebs <- summ.foodwebs[1:9, ]
nreps <- 1000
foodweb.names <- c("benguela", "britishgrasslands", "cantoncreek", "chesapeakebay",
  "coachella", "scotchbroom", "skipwithpond", "stmarks", "stonystream")
foodweb.metrics <- c("S", "L", "C", "avg.degree", "l.G", "basal", "inter",
  "top")

# Random
rand.mean <- matrix(nrow = 9, ncol = NCOL(summ.foodwebs))
rownames(rand.mean) <- foodweb.names
colnames(rand.mean) <- foodweb.metrics

# Cascade
cascade.mean <- matrix(nrow = 9, ncol = NCOL(summ.foodwebs))
rownames(cascade.mean) <- foodweb.names
colnames(cascade.mean) <- foodweb.metrics

# Niche
niche.mean <- matrix(nrow = 9, ncol = NCOL(summ.foodwebs))
rownames(niche.mean) <- foodweb.names
colnames(niche.mean) <- foodweb.metrics
for (i in 1:length(foodweb.names)) {
  rand.mean.reps <- matrix(nrow = nreps, ncol = length(foodweb.metrics))
  cascade.mean.reps <- matrix(nrow = nreps, ncol = length(foodweb.metrics))
  niche.mean.reps <- matrix(nrow = nreps, ncol = length(foodweb.metrics))
  for (j in 1:nreps) {
    # Random
    rand.mean.reps[j, ] <- summ.stats(random(S = summ.foodwebs$S[i],
      C = summ.foodwebs$C[i]))
    # Cascade
    cascade.mean.reps[j, ] <- summ.stats(cascade(S = summ.foodwebs$S[i],
      C = summ.foodwebs$C[i]))
    # Niche
```

```

      niche.mean.reps[j, ] <- summ.stats(niche(S = summ.foodwebs$S[i],
      C = summ.foodwebs$C[i]))
    }
    # Means
    rand.mean[i, ] <- colMeans(rand.mean.reps, na.rm = TRUE)
    cascade.mean[i, ] <- colMeans(cascade.mean.reps, na.rm = TRUE)
    niche.mean[i, ] <- colMeans(niche.mean.reps, na.rm = TRUE)
  }
data.frame(rbind(diag(cor(niche.mean, summ.foodwebs)), diag(cor(cascade.mean,
summ.foodwebs)), diag(cor(rand.mean, summ.foodwebs))), row.names = c("Random",
"Cascade", "Niche"))

##           S           L           C avg.degree          1.G          basal          inter
## Random  1 0.9999836 0.9999683 0.9999616 -0.11973760 -0.1206812 0.5743326
## Cascade 1 0.9999301 0.9999691 0.9999278          NA 0.3461860 0.8418459
## Niche   1 0.9999978 0.9999976 0.9999955 -0.04671967 -0.2749010 0.3351486
##
##              top
## Random 0.7853742
## Cascade 0.7341140
## Niche 0.6429415

```

**Answer:**

The results indicate that the empirical and model food webs share some characteristics but not others. For instance, the models are generally poor predictors of the characteristic path length and degree. However, they do relatively well in predicting the proportions of top or basal species. Overall, the niche model appears to be the worst performer.

3. Determine the susceptibility of these empirical food webs to species deletions by creating a function that will compute the number of secondary extinctions that occur in response to the loss of each species. Note that secondary extinctions will occur any time a species deletion leads to another species having no prey.

```

# Compute secondary extinctions
secondary.extinct <- function(adj, ext) {
  colnames(adj) <- paste0("V", 1:NROW(adj))
  g <- graph.adjacency(adj)
  basal <- which(colSums(adj) == 0)
  basal <- basal[basal != ext]
  if (length(basal) == 0) {
    extinctions <- sum(colSums(adj) == 0)
  } else {
    g <- delete.vertices(g, ext)
    dists <- matrix(nrow = length(basal), ncol = NCOL(adj), 0)
    basal <- which(V(g)$name %in% names(basal))
    for (i in 1:length(basal)) {
      dists[i, which(sapply(suppressWarnings(get.shortest.paths(g,
        from = basal[i])$vpath), length) != 0)] <- 1
    }
    extinctions <- sum(colSums(dists) == 0)
  }
  return(extinctions)
}

# Delete species one at a time
empirical.secondary <- function(adj) {
  extinct <- matrix(nrow = NROW(adj), ncol = 2)
  for (i in 1:NROW(adj)) {
    sec <- secondary.extinct(adj, i)
  }
}

```

```

    extinct[i, ] <- c(i, sec/NROW(adj) * 100)
  }
  return(extinct)
}
# Compute statistics
benguela.extinct <- empirical.secondary(benguela)
britishgrasslands.extinct <- empirical.secondary(britishgrasslands)
cantoncreek.extinct <- empirical.secondary(cantoncreek)
chesapeakebay.extinct <- empirical.secondary(chesapeakebay)
coachella.extinct <- empirical.secondary(coachella)
scotchbroom.extinct <- empirical.secondary(scotchbroom)
skipwithpond.extinct <- empirical.secondary(skipwithpond)
stmarks.extinct <- empirical.secondary(stmarks)
stonystream.extinct <- empirical.secondary(stonystream)

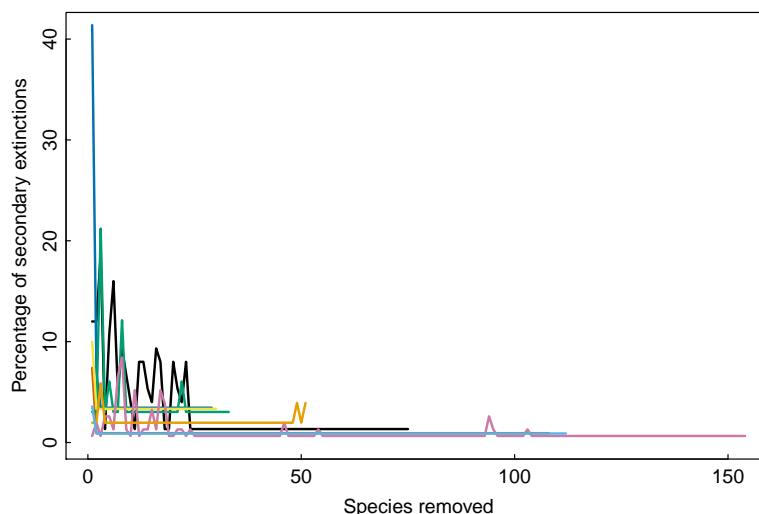
```

4. Plot the number of secondary species extinction as a function of each species being deleted for all food webs onto the same panel. Describe the patterns.

```

plot(britishgrasslands.extinct[, 1], britishgrasslands.extinct[, 2], t = "l",
     ylim = c(0, 41), xlim = range(benguela.extinct[, 1], britishgrasslands.extinct[,
     1], cantoncreek.extinct[, 1], chesapeakebay.extinct[, 1], chesapeakebay.extinct[,
     1], coachella.extinct[, 1], scotchbroom.extinct[, 1], skipwithpond.extinct[,
     1], stmarks.extinct[, 1], stonystream.extinct[, 1]), xlab = "Species removed",
     ylab = "Percentage of secondary extinctions", lwd = 2)
lines(benguela.extinct[, 1], benguela.extinct[, 2], col = cb["blue"], lwd = 2)
lines(cantoncreek.extinct[, 1], cantoncreek.extinct[, 2], col = cb["red"],
     lwd = 2)
lines(chesapeakebay.extinct[, 1], chesapeakebay.extinct[, 2], col = cb["green"],
     lwd = 2)
lines(coachella.extinct[, 1], coachella.extinct[, 2], col = cb["yellow"],
     lwd = 2)
lines(scotchbroom.extinct[, 1], scotchbroom.extinct[, 2], col = cb["pink"],
     lwd = 2)
lines(stmarks.extinct[, 1], stmarks.extinct[, 2], col = cb["orange"], lwd = 2)
lines(stonystream.extinct[, 1], stonystream.extinct[, 2], col = cb["turquoise"],
     lwd = 2)

```



**Answer:**

The results show that certain species have a much larger impact on the entire food webs. Hence, the vulnerability of food webs depends on species identity.