Lab: Food Webs & Network Measures (Student Version)

```{r lab_setup, include=FALSE}
knitr::opts_chunk$set(error = TRUE, warning = TRUE, message = TRUE)
suppressPackageStartupMessages({
  if (requireNamespace("tidyverse", quietly = TRUE)) library(tidyverse)
  if (requireNamespace("igraph", quietly = TRUE)) library(igraph)
  if (requireNamespace("bipartite", quietly = TRUE)) library(bipartite)
  if (requireNamespace("networks3D", quietly = TRUE)) library(bipartite)
})
net_data <- function(...) file.path("labs", "data", ...)
```

Lesson Overview

Computational Topics
- Build and visualize food webs
- Write functions to implement mathematical equations

Conservation topics
- Paleofood webs
- Species extinction

In this lab we will practice our network visualization and manipulation skills using the paleo food web data from [Yeakel et al. 2014] (https://doi.org/10.1073/pnas.1408471111).

![Paleoweb](labs/figures/paleoweb.jpg)

See the beautiful, animated version of the graphic above [here] (https://infograficos.estadao.com.br/public/cidades/extincoes-egito/)

With some interaction networks we can observe the interactions, for example plant-pollinator networks, seed-disperal networks, human social networks. In food webs sometimes feeding interactions are observed directly, through camera traps, people doing timed observations, and now molecular analysis of gut contents/scat. However, often with food webs people build probabilistic models of who interacts with who based on body size (as in the Yeakel et al. 2014), especially with paleowebs. Thus the data from Yeakel et al. is 1) an occurrence matrix (Figure 2 from the publication) and a matrix of body sizes (two columns, females then males). We will use these data to build the foodwebs for each time period. This lab is pretty challenging because it will use many of our core programming skills (for loops, writing functions, subsetting data) and our network skills.

First we will read in the data. The matrix we are reading in has no row or column names, we will have to set them.

```{r}
sp_occ <- read.table(file="data/egypt_data.txt", header = FALSE)
str(sp_occ)
sp_mass <- read.table(file="data/egypt_mass.txt", header=FALSE)
str(sp_mass)
```

![Figure 2](labs/figures/figure2.jpg)

The rows are arranged in the order of Figure 2 of the manuscript. To set the rownames we can make a vector of the names then use the function `rownames`. We also have to note which species are predators (all those in the species in the Carnivora clade in figure 2). Otherwise we will create a web where giraffes are voracious predators consuming all of the other species (I made this mistake when constructing the networks originally). I have transcribed the data from figure 2 for you:

```{r}
row_labs_sp <- c("Canis aureus", "Vulpes vulpes", "Lycaon pictus", "Hyaena hyaena",
"Crocuta crocuta", "Panthera leo (long maned)", "Panthera leo (short maned)", "Panthera
pardus", "Acinonyx jubatus", "Loxodonta africana", "Equus asinus", "Equus grevyi",
"Equus quagga", "Diceros/Ceratotherium", "Sus scrofa",  "Phacochoerus aethiopicus",
"Hippopotamus amphibius", "Giraffa camelopardalis", "Dama mesopotamica", "Camelus
```

dromedarius", "Taurotragus oryx", "Tragelaphus spekei", "Addax nasomaculatus", "Oryx dammah", "Oryx beisa", "Hippotragus equinus", "Kobus kob", "Kobus megaceros", "Alcelaphus bucelaphus", "Connochaetes taurinus", "Litocranius walleri", "Ammodorcas clarkei", "Gazella dorcas", "Gazella leptoceros", "Gazella soemmerringii", "Capra ibex", "Ammotragus lervia", "Bos primigenius", "Syncerus caffer")

```
###  Set 1 for predators, 0 for prey
carnivores <- c(rep(1, 9), rep(0, length(row_labs_sp)- 9))
names(carnivores) <- row_labs_sp
```

Lab question 1: Creating our foodwebs based on body sizes.

– 1a. Use the above vector of species names to label the row names of the species occurrence and the body size matrices. The columns of the species occurrence matrix are time points, so we can leave those as V1 etc., but we should set the column names of the mass matrix as "f", "m" (female and male). Use `head` to check each matrix to see if the names are displayed properly.

```{r}
###  your code here
rownames(sp_occ) <- row_labs_sp
rownames(sp_mass) <- row_labs_sp
colnames(sp_mass) <- c("f","m")
head(rownames(sp_occ))
head(rownames(sp_mass))
head(colnames(sp_mass))
```

Yeakel recommended an updated equation to estimate the probability a predator consumed a prey based on their relative body masses from [Rohr et al. 2010.] (https://doi.org/10.1086/653667). The probability of existence of a trophic link between a predator of body-size m_i and a prey of body-size m_j is given by:

![Probabilitic feeding equation](labs/figures/feeding_equ.png)
(P(A_{1j} = 1) is the probability predator i eats prey j).

– 1b. Write a function and call it `probEat` to implement the equation above. Round the probability to two decimal places.

Below are the values of alpha, beta, and gamma for the Serengeti. In addition, you will need a function to compute the inverse logit function because this equation is for the logit of the probability, so to calculate the 0-1 probability you will need to take the inverse logit of the other side of the equation. Also note, log^2 is equivalent to $(log($m_i$/$m_j$))^2$

```{r}
alpha <- 2.51
beta <- 0.79
gamma <- -0.37

inv_logit <- function(x) exp(x)/(1+exp(x))
inv_logit
```
```{r}
probEat <- function(mi, mj){
  r <- log(mi / mj)
  n <- alpha + beta * r + gamma * (r^2)
  p <- inv_logit(n)
  return (round(p, 2))
}
probEat
```

```{r}
# your code here
```

– 1c. Now create networks of who eats whom. We will start with adjacency matrices. We

will assume all of our species are the size of females. For this step, don`t worry about predators vs. prey yet, just calculate all of the feeding probabilities based on body sizes.

Hint: if you start with a square matrix of all zeros (one row and one column for each species), you can use a for loop to fill in that matrix with probabilities calculated from your function above.

```{r}
# your code here
eat_network <- matrix(0, nrow = length(row_labs_sp), ncol = length(row_labs_sp))
rownames(eat_network) <- row_labs_sp
colnames(eat_network) <- row_labs_sp
for (i in 1:nrow(eat_network)){
  for(j in 1:nrow(eat_network)){
    eat_network[i,j] <- probEat(sp_mass[i,1], sp_mass[j,1])

  }
}
head(eat_network)
```

- 1d. Now that you have your matrix of potential feeding interactions based on body size, use the `carnivores` vector created above to set all of the feeding interactions of herbivores (0s in that vector) to zero. In foodwebs the columns are the higher trophic level and the rows are the lower.
HINT: the function `sweep` may be useful, though there are many approaches to do the needed matrix multiplication. Print the row and column sums.

```{r}
# your code here
eat_network <- sweep(eat_network, 2, carnivores, `*` )
head(eat_network)
```

Lab question 2: Breaking the networks into time periods

- 2a. With our matrix of feeding interaction we can create a web for each time period, including only the species that were not extinct in the period. Try first just using the second time period (the second column of `sp_occ`).

Use the function `empty` from the bipartite package to empty the matrix of rows and columns with no interactions. The number of species in the second time period is 36 `sum(sp_occ[,2])`. Check to see that the number of rows in your network with probabilities > 0 is 36.

HINT: You will need to zero out the rows where a species in not present in that time period and the columns. The function `sweep` may be useful again.

```{r}
###  your code here
year_2 <- sp_occ[,2]
year_2_empty <- eat_network * year_2
year_2_empty <- sweep(year_2_empty, MARGIN = 2, year_2, `*`)
sum(rowSums(year_2_empty)>0)
```

- 2b. Now create a network for all of the time points by creating a list where each element is a network. You will need to use a for loop, or an `lapply` if you feel like experimenting with apply functions. Print the first 5 columns and rows of the 5th time period.

HINT: If choosing the for loop route, remember to create an empty list of a specific length use the function `vector`. To access a specific element of a list, use [[]], for example cool_list[[1]] accesses the first element of the list.

```{r}
# Each item in the list is an adjacency matrix
```

```r
allYear_network <- vector("list", ncol(sp_occ))
for (i in 1:ncol(sp_occ)){
  year_i <- sp_occ[,i]
  year_inew <- eat_network * year_i
  year_inew <- sweep(year_inew, MARGIN = 2, year_i, `*`)
  allYear_network[[i]] <- year_inew  #stores matrix in list
}
print(allYear_network[[5]][1:5,1:5])
```

Lab question 3: Visualize the networks
– 3a. Convert the adjacency matrices to igraph class objects using the function `graph_from_adjacency_matrix`. You can use a for loop or an lapply. Because these are food webs, set the argument mode to "directed" and the argument diag to FALSE (this means a species cannot consumer members of its own species, i.e., no canabalism/self-loops). Also remember that these interactions are weighted.
```{r}

igraph_network_list <- list()
for (i in 1:length(allYear_network)){
  igraph_network_list[[i]] <- graph_from_adjacency_matrix(allYear_network[[i]],mode =
"directed", weighted =TRUE, diag = FALSE)
        }

igraph_network_list
str(igraph_network_list)

```

– 3b. Plot three networks of your choice, using different colors for the predators and prey.

```{r}
### assign groups as carnivore or herbivore
grouped_allYear <- carnivores[V(igraph_network_list[[22]])$name]


### convert to a network 3d object
#convert to a network 3d object
food_web_net3d_22 <- igraph_to_networkD3(igraph_network_list[[22]], group =
grouped_allYear,
                                        what = "both")

### plot the network
forceNetwork(Links = food_web_net3d_22$links,
            Nodes = food_web_net3d_22$nodes,
            Source = "source", Target = "target",
            Value = "value",  NodeID = "name",
             Group = "group", arrows = TRUE,
            opacity = 0.8, zoom=TRUE)

```

```{r}
# your code here
### assign groups as carnivore or herbivore
grouped_allYear <- carnivores[V(igraph_network_list[[2]])$name]


### convert to a network 3d object
#convert to a network 3d object
food_web_net3d_2 <- igraph_to_networkD3(igraph_network_list[[2]], group =
grouped_allYear,
                                        what = "both")

### plot the network
forceNetwork(Links = food_web_net3d_2$links,
```

```
              Nodes = food_web_net3d_2$nodes,
              Source = "source", Target = "target",
              Value = "value",  NodeID = "name",
               Group = "group", arrows = TRUE,
              opacity = 0.8, zoom=TRUE)
```

```{r}
# your code here
###  assign groups as carnivore or herbivore
grouped_allYear <- carnivores[V(igraph_network_list[[15]])$name]


###  convert to a network 3d object
#convert to a network 3d object
food_web_net3d_15 <- igraph_to_networkD3(igraph_network_list[[15]], group =
grouped_allYear,
                                          what = "both")

###  plot the network
forceNetwork(Links = food_web_net3d_15$links,
              Nodes = food_web_net3d_15$nodes,
              Source = "source", Target = "target",
              Value = "value",  NodeID = "name",
               Group = "group", arrows = TRUE,
              opacity = 0.8, zoom=TRUE)

```
```