Lab #3: Transmission in networks

Bio 347 - Disease Ecology

**Objectives:**

* Build a network based on shared courses, clubs, and dorms of students in this lab section.
* Simulate transmission of a directly-transmitted parasite on a static network.
* Determine how network position influences the importance of a host to parasite spread.

**Background:**

Classic epidemiological theory is based on differential equation models from physics that treat individuals as identical particles perfectly follow the population-averaged rules of contact, transmission, recovery, etc. in an infinitely large environment. These models form the foundation of our understanding of epidemics because (1) they are relatively easy to formulate, understand, and communicate, (2) they can often be “solved” with (relatively) simple algebra and calculus. We will spend much of the rest of the semester elaborating on these classic models to understand epidemics in ecological communities.

Despite their utility, these classic models have a few obvious drawbacks. One important drawback of the classic approach is that *it ignores variation*. It turns out that real individuals vary, and this variation can change epidemic dynamics in really powerful ways. Biologists typically focus on *two types of variation*: consistent differences among individuals (sometimes called “demographic heterogeneity”) and differences in event-to-event outcomes among individuals even when we expect that they are similar (called “demographic stochasticity”).

Other types of models can deal with sources of variation. We are going to examine a *network epidemic model* that incorporates heterogeneity in contacts, some hosts contact many others and some hosts engage in few contacts. This particular network is *static*, meaning that we assume that the contact network’s structure does not change on the time scale of the epidemic. This model will also incorporate stochasticity in transmission and recovery. Transmission will depend on a probability that a contact between an infected host and an uninfected one causes a new infection. Similarly, individuals will vary in their duration of infectiousness.

**Contact networks**

Networks are graphical tools that display individual hosts (“nodes”) as circles and the contacts among hosts as lines (“edges”) that connect them. Networks can be generated randomly using simple or complex algorithms, or researchers may study a network of a specified size and structure. Below is an example of a network that I generated randomly. It has 25 hosts and the probability that any two hosts are connected is only 10%:



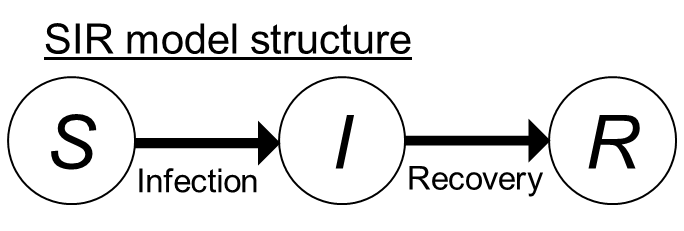
Despite the low probability of contact between any two hosts, a large cluster can still form. In this example, the largest cluster includes 19 of the 25 hosts. The potential for parasite spread on the network will be related to the contact structure. The simplest example of this is that the size of the largest cluster will determine the greatest number of infections that can result from the introduction of one infected host.

**Questions (write all necessary code in a script to submit):**

1) Generate a contact network for the students in this class. Indicate a contact for any two students if they shared a course, club/sport activity, or dorm building in this academic year.

* Create an *adjacency matrix* in Excel in which each student is represented as a row or column.
  + I have created matrix labeled across the columns and down the rows with an alphabetically order list of the names of students in this class. Obtain this from Canvas.
  + In the cells of the matrix, place a 0 if there is no contact link and a 1 if there is a contact link. Diagonal elements would represent a contact with yourself, but that is meaningless. Nonetheless, the diagonal line is filled with 1s.
  + Lastly, this matrix will be *symmetric*, the top right will be a mirror image of the bottom left if you folded the matrix along the diagonal.
  + Example: Jake and Marta have the same class. Marta plays rugby with Jen. Alex and Jen live in the same dorm building:
* Rather than code from scratch, you will use some code that I have written as a template for this assignment. Download Lab3Template.r from Canvas.
* Save the completed matrix to your working directory using the the file name SP2019Matrix.csv.
* Import the data into R using read.csv() and store it in a variable, CM.
  + We’ll need to add some arguments to the read.csv() function: header=TRUE, row.names=1, check.names=FALSE
* We need to convert our adjacency matrix into a different data format that is used by the functions we will use to simulate, called an edge list. Run the following lines to convert it:
  + CM = as.edgelist(network(CM, directed=F, matrix.type="adjacency"))
* 2) (**1.5 points)** Individuals in our network differ in how connected they are to others. In network lingo, the *degree* of a node equals the number of other nodes it is directly connected to.
* We are going to simulate epidemics on this network by randomly starting with different hosts.
* In a comment, write a hypothesis for how you think the *degree* of the first infected host will be related to the size of the epidemic (which we will calculate as the fraction of the class that became infected).
* Explain your rationale or justification for this hypothesis.
* Use the Adj.Matrix.Degrees() function provided in the template to calculate the degree of each node in your network.
* Make a histogram to illustrate the variation in degree among the individuals in the network. Explain in a comment whether you think there is lots or little variation among individuals in their degree.

3) (**1.5 points)** Use the summarize.SIR() function that I provide to obtain a final summary of an outbreak. The result that this function provides indicates the degree (but not identity) of the individual who started the outbreak (“Patient Zero”), and the proportion of hosts in the network that became infected before the outbreak ended. We will be simulating a simplified version of this model, called an *SIR* model, which indicates that we’re tracking Susceptible, Infected, and Recovered individuals. The module diagram below indicates the *transitions* in this model, infection and recovery:



* For simplicity, we will focus on variation in β, the transmission rate parameter, which determines the probability of infection. Once infected, individuals are immediately infectious until they recover. \
  + You will still need to specify the edgelist argument. There is a default value for β, the transmission rate parameter, but you can override this.
  + Run this function a few times using our edgelist, CM, and the default value of beta=0.1, then change beta a bit. You should get a different result every time, because the model is stochastic.

3) (**2 points)** Test your hypothesis relating the degree of Patient Zero to epidemic size. Use the Spread.by.Degree() function provided on Canvas to simulate at least 10,000 iterations of outbreaks on the network. You may have to try a few different values of β in order to get an informative result (If everyone is never infected or everyone is always infected, try changing β by 10-fold).

* Make a figure examines the relationship between degree and the ultimate size of the outbreak. Explain in a comment whether your hypothesis is supported and what question you would investigate next.