Abundance and Disease Simulation - Preliminary Attempts

Reed Scott

2024-11-26

# What I’m asking of **you**

In this report I have outlined a lot. In order to wrap this up in a neat bow, I have some specific asks of each of you.

**Brittany:** I could use help thinking about how to fine tune the construction of the meta-community abundance dataframe in step one. Right now, I use a combination of binomial presence absence, rank abundance, and a way of nesting the models. However, each of these could be implented in a miriad of way and I need help considering what would be most biologically correct. Here are some specific examples of things I need your help considering

1. Does doing presence / absence then rank abundance make sense, OR could we potentially use a 0 inflated poisson distribution?
2. Does it make the most sense to use a fixed “equilibrium abundance” for each rank abundance or should we consider using some sort of distribution?
3. Currently, I have the nested model set so that if a species doesn’t occur, species further in the rank abundance order cannot occur (ex: If species 4 does not occur, then probability of species 5 and 6 occuring = 0). We could consider changing this so that “later” species could occur, but at a lower probability.
4. Is this something we could post to the “HMEcology” listserv and get thoughts / feedback on? I’ve been thinking about it and while not strictly hierarchical, this could be of interest to some in that group.

**Mark:** I could most use your help with the epidemiological model and making sure I have it implemented correctly. I also need help establishing realistic ranges for each of the parameters of the epidemiological model. So, my two asks of you are

1. Consider parameters established in Step Two: The Epidiomological Model. Specifically in “Initializing” and “Species Characteristics” as well as the “Meta-community Characteristics”. I have taken these parameters from eq. 1 of your 2020 ecology letter paper like you suggest. Right now, these parameters are essentially dummy variables and the goal will be to later add in realistic values for each parameter. I am planning to conduct a literature review to establish these values, and would appreciate your insight once that is done.
2. I also need some help understanding how to set up the connectivity matrix. I can easily make a P\*P matrix. However, I need help thinking about the rules for connectivity between patches. What rules should be in place to determine connectivity? For example, I assume that a patch will have high connectivity with itself. Are there other ecological assumptions we need to consider? What values should this parameter take and what distribution should it use?
3. Consider the rate equations used in the for loop for simulation (Step Two, running the simulation over time). Do these seem accurate to what your paper? Given that I’m aware that
   1. I will still need to switch the equation to be a frequency dependent model (just haven’t gotten around to that yet).
   2. For now, I have removed the area effect (Aj/Ap). This is just because I haven’t figured out how to implement that part in R. Yet.

# Introduction

The goal of this projection is to understand the relationship between landscape R0 and diversity. Our specific hypothesis is that as increases, landscape R0 should decrease. This is assumes that a dilution effect will occur. To test this, we will be simulating various possible meta-communities, calculating diversity for each meta-community, and comparing landscape R0. How will we create and test these meta-communities? Below, I outline what I believe are the first steps to simulating the data and testing them using an epidemiological model.

# Step One: Simulating Abundance

## A quick explanation

To test our hypothesis that landscape R0 ~ we have to simulate different ecological metacommunities. For this part of the project we make a few key assumptions:

1. We assume that communities are nested. Less diverse communities are *nested* within more diverse communities.
2. We also assume species differ in their probability of occurrence.
3. Finally, we assume that the dataset uses a rank abundance model. This essentially means that a few species are highly abundant, while most species are rarer.

So, with these assumptions in mind, let’s see how we can simulate a metacommunity!

## Parameters for simulating abundance

The first thing we need to do, as with any good simulation, is set the seed. I have chosen to set it at 1234. Then, there are 2 parameters we have to identify to determine the “dimensions” of our meta-community. Those are the number of patches in the metacommunity, and the number of species. In this example I have chose to work with 5 patches and 6 species.

*Note: 5 and 6 are just examples. We will be able to adjust these dimensions as we please*

set.seed(1234)  
num\_patches <- 5 #number of patches in metacommunity  
num\_spp <- 6 #number of POSSIBLE spp in metacommunity

Once we have the number of species and number of patches specified, we can create a p X s matrix and transform it in to a data frame. This is the dataframe we will eventually populate with our abundance data.

meta\_comm1 <- data.frame(matrix(NA, nrow = num\_patches, ncol = num\_spp))

Lastly in our setup, we have to establish the probability that a species occurs at a given site. We assume that some species are fairly ubiquitous (high probability of occurrence) while some species are rarer (low probability of occurrence). Additionally, we assign a parameter K. K is the rank abundance, or the abundance we expect at equilibrium IF the species occurs.

*Note: These are currently all made up numbers just to see if this structure works. We can adjust these* *to be more biologically relevant. Additionally, I have tried to use a rank abundance system, but we* *may find a better system, or find a better way to implement rank abundance. I am open to suggestions*

S <- c(0.95,0.9,0.75,0.50,0.20,0.1) #an array of probability values for the occurence of each spp  
K <- c(60,50,40,30,20,10) #this is just an example, but k is the abundance at each rank (i think?)

*Note: right now I have fixed K. but perhaps it could make more sense to have it vary along some distribution?*

With these parameters set, we can prepare to populate the dataframe. Woohoo!

## Populating the dataframe

So, we have our probabilities and rank abundances set. We have an empty dataframe that needs to be filled. Let’s get to it!

There may be better ways of doing this, but what I’ve found works for now is creating a for{} loop, where I:

1. Determine whether or not each species occurs at each patch
2. Assign the abundance of each species at a given patch.

For the most abundant species, this is pretty straight forward.

meta\_comm1[i,1] <- rbinom(n = 1, size = 1, prob = S[1])  
 meta\_comm1[i,1] <- ifelse(meta\_comm1[i,1] == 1, K[1],0)

This tells us that for species 1, at each patch, we will randomly assign occurrence/absence with a probability S[1] (recall that S is a vector of probabilities of occurrence for each species). If the species occurs, we then assign it a value of K[1] (recall K is a vector of rank abundance for each species).

These probabilities are where we build in the assumption of nestedness, so for every species following the most abundant, the calculation is a little more complicated. We create a dependency, where the probability of occupancy of each species *depends* on whether or not the next most abundant species is present.

meta\_comm1[i,2] <- ifelse(meta\_comm1[i,1] > 0, rbinom(n = 1, size = 1, prob = S[2]),0)  
 meta\_comm1[i,2] <- ifelse(meta\_comm1[i,2] == 1, K[2],0)

In the above example, first we must determine if the prior species occurs: meta\_comm1[i,1] > 0. If the prior species in the order *does* occur, then the current species may occur with a probability S[i]. If the prior species *does not* occur than the probability of the current species occurring becomes 0.

*Note: There are a couple of things I think could be adjusted about this. Most glaring to me is the assumption* *that if a species does not occur, it makes the probability of other species further in the rank abundance* *chain occurring 0. If we wanted to, we could adjust this so that it is a lower (but not necessarily 0)* *probability.*

### The for loop.

All of this can be wrapped in to a large for loop which looks like this:

for(i in 1:nrow(meta\_comm1)){  
 for(j in 1:ncol(meta\_comm1)){  
 #determine occurrence and abundance of spp 1  
 meta\_comm1[i,1] <- rbinom(n = 1, size = 1, prob = S[1])  
 meta\_comm1[i,1] <- ifelse(meta\_comm1[i,1] == 1, K[1],0)  
   
 #determine occurrence and abundance of spp2  
 meta\_comm1[i,2] <- ifelse(meta\_comm1[i,1] > 0, rbinom(n = 1, size = 1, prob = S[2]),0)  
 meta\_comm1[i,2] <- ifelse(meta\_comm1[i,2] == 1, K[2],0)  
   
 #determine occurrence and abundance of spp 3  
 meta\_comm1[i,3] <- ifelse(meta\_comm1[i,2] > 0, rbinom(n = 1, size = 1, prob = S[3]),0)  
 meta\_comm1[i,3] <- ifelse(meta\_comm1[i,3] == 1, K[3],0)  
   
 #determine occurrence and abundance of spp 4  
 meta\_comm1[i,4] <- ifelse(meta\_comm1[i,3] > 0, rbinom(n = 1, size = 1, prob = S[4]),0)  
 meta\_comm1[i,4] <- ifelse(meta\_comm1[i,4] == 1, K[4],0)  
   
 #determine occurrence and abundance of spp 5  
 meta\_comm1[i,5] <- ifelse(meta\_comm1[i,4] > 0, rbinom(n = 1, size = 1, prob = S[5]),0)  
 meta\_comm1[i,5] <- ifelse(meta\_comm1[i,5] == 1, K[5],0)  
   
 #determine occurrence and abundance of spp 6  
 meta\_comm1[i,6] <- ifelse(meta\_comm1[i,5] > 0, rbinom(n = 1, size = 1, prob = S[6]),0)  
 meta\_comm1[i,6] <- ifelse(meta\_comm1[i,6] == 1, K[6],0)  
 }  
}

## House Cleaning

There is a bit more code I added just to make things easier to read. I added a column to the dataframe called Patch, just so each patch has a unique ID. I then changed the column names to Spp1 - Spp6 so you know that each column represents a different species.

meta\_comm1$Patch <- c('Patch1','Patch2','Patch3','Patch4','Patch5')  
colnames(meta\_comm1) <- c('Spp1','Spp2','Spp3','spp4','spp5','Spp6','PatchID')

## Beta diversity

A major purpose of this project is to assess how beta diversity impacts R0 . I have found that an easy way to do that is to use the betadiver() function from vegan.

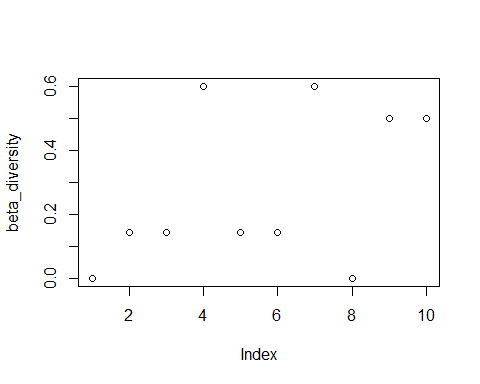
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-8

beta\_diversity <- betadiver(meta\_comm1[,1:6], method = 'w')  
plot(beta\_diversity)



And that’s it! We now have working code to simulate a meta-community. Next, we need to figure out how to use this in an epidemiological model.

# Step two: The epidemiological model

## Initializing

The first step is to determine initial conditions. We need to determine the number of susceptible individuals and infected individuals at the start of this simulation. For now, I have assumed that the most abundant species (spp1) is the most susceptible (highest probability of getting disease) while the least abundant spp (spp6) is the most resistant. Because this uses an SIS model, the number of infectious individuals, I, can be determined by the total population - S. Total population is also just S+I. Finally, we have to determine a starting disease level. In this case, we are using the amphibian chytrid fungus as an example, so I have chose what I *think* is a reasonable starting number of 10000 zoospores.

S <- ceiling(meta\_comm1[,1:6]\*c(0.8,0.85,0.9,0.93,0.95,.99)) #starting value of susceptibles  
I <- meta\_comm1[,1:6] - S #starting value of infecteds  
N <- S+I  
Z <- 10000 #starting amount of zoospores

## Species characteristics

After choosing the initial values, we must assign species specific traits to determine how their abundance will impact disease risk. These include a birth rate (b), death rate (d), transmission coefficient(beta), recovery rate (v), dispersal rate (phi), and BD load / shedding rate (lambda). These are shown in the table below.

## Species birth death trans recovery dispersal shedding  
## 1 Spp1 0.6 0.06 0.00013 0.4 0.09 3.0  
## 2 Spp2 0.5 0.05 0.00012 0.5 0.08 2.5  
## 3 Spp3 0.4 0.04 0.00011 0.6 0.07 2.0  
## 4 Spp4 0.3 0.03 0.00010 0.7 0.06 1.5  
## 5 spp5 0.2 0.02 0.00009 0.8 0.05 1.0  
## 6 spp6 0.1 0.01 0.00008 0.9 0.04 0.5

*Note: right now, these characteristics are relatively arbitrary. I came up with values that I think would be* *in the range of plausible values. However, these are essentially made up. A key next step is to find more* *realistic values. This may include asking Mark what values they observed in their study and conducting a literature review*

## Meta-community characteristics

Lastly, we must set some more characteristics about the meta-community within which we are operating. We must establish , or the decay rate of zoospores of Bd. We have to establish a matrix, c, of the connectivty values between patches. And we must establish the area of each patch. We also have to establish the length of time over which the simulation will occur. In this case, I have set time = 90, to represent a potential 90 day amphibian breeding season.

# meta-community characteristics  
gamma <- 0.001 #zoospore decay rate  
  
#Connectivity of patches  
c <- matrix(data = rnorm(n = num\_patches^2, mean = 0.5, sd = 0.1),  
 nrow = num\_patches,   
 ncol = num\_patches)  
  
A <- rnorm(n = num\_patches, mean = 0.5, sd = 0.1) #area ratios  
time <- 90 #how many "days" do I want in the season

*Note: right now, c is a matrix of random values with mean 0.5 and sd = 0.1. This will likely need to be* *adjusted considerably in my opinion. We may need to set distance values between each pair of patches, and* *then consider how distance affects connectivity. Additionally, we have to consider that within this matrix,* *along the diagonal is the probability that an individual stays in the same patch. There will likely be a* *different formulation for this probability.*

## Running the simulation over time

Alright, we’re finally at the main event! This is a relatively short bit of code, but it does a lot.

First, we’ll want to establish an empty list, whose length is equal to the variable time. We will use this list to record the number of susceptible individuals of each species, infectious individuals of each species, total number of individuals of each species, and the time (i.e. “day”). With that set we can start the for loop. The for loop runs for time t. There are 3 key steps to this loop.

1. Calculate delta\_s, delta\_I, and delta\_Z. This is the number of new susceptible and infectious individuals for each species, as well as the number of zoospores being added to the pool.
2. Calculate the new S, I, Z, and N values. This just simply adds the delta values to their respective initial values to calculate the “new” values for S,I,Z, and N.
3. Package it all in a list. We will do this so that we get a nice output we can later use for analysis.

pop\_list <- vector("list", length = time)  
for (t in 1:time) {  
 delta\_s <- b\*N - d\*S - beta\*S\*Z + v\*I + phi\*sum(-c\*S + c\*S) #for now have excluded area of patches  
 #may want to add that back in though  
 delta\_I <- beta\*S\*Z-(v+d)\*I+phi\*sum(-c\*I + c\*I)#for now have excluded area of patches  
 #may want to add that back in though  
 delta\_Z <- sum(lambda\*I - gamma\*Z)  
 S <- S+delta\_s  
 I <- I+delta\_I  
 z <- Z+delta\_Z  
 N <- S + I  
 t = t  
 pop <- list(Susceptible = S, Infectious = I, Zoospores = Z, Total = N, Time = t)  
 pop\_list[[t]] <- pop  
}

I am not going to display pop\_list because it’s huge: it’s 90 lists, each of which contain 5 dataframes. Additionally, right now the output is unrealistic. However, I think that as we fine tune the parameters, we will find this to be a good functional model.

*Note: in an earlier draft of this document, Brittany commented that death rate should be state dependent. I’m inclined to agree, and think* *this would be easy to do. we just have to include a disease specific mortality rate () in the rate equation for infection*

# Conclusion

I believe what I have now is solid progress and a really good start to the simulation! However, I have no delusions. There is still a lot of work to be done one this. Most important, I believe, is that I need to tune the parameters to be more biologically relevant / realistic. I also still need to consider how I will be analyzing the output of this SIS model. Additionally, the goal is to simulate multiple different meta-communities and compare landscape R0 values, which I have not yet done. So, yes, there is still a lot of work to do. I am nonetheless excited at the progress I’ve made.