

# Simulating Abundance Preliminary Attempt

Reed Scott

2024-11-25

## A quick explanation

In order to test our hypothesis that  $R_0 \sim \beta$  we have to simulate different ecological metacommunities. How do we do that? For this part of the project we make a few key assumptions:

1. We assume that communities are nested. That is to say 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!

## The parameters

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 trasform 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 probablity 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 occurrence 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 then 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  $R_0$ . I have found that an easy way to do that is to use the `betadiver()` function from `vegan`.

```
library(vegan)
```

```
## Loading required package: permute
```

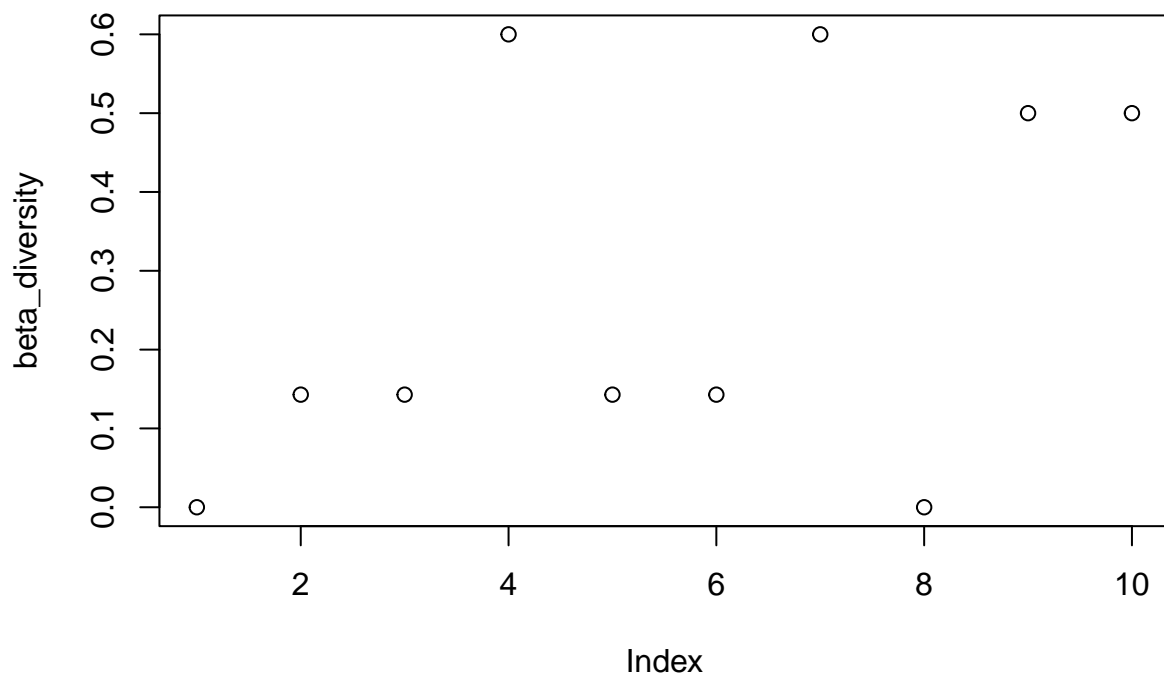
```
## Loading required package: lattice
```

```
## This is vegan 2.6-8
```

```

beta_diversity <- betadiver(meta_comm1[,1:6], method = 'w')
plot(beta_diversity)

```



## Conclusion

And that's it! We now have working code to simulate a meta-community. Check out "Epimodel.pdf" next to find out how we'll use this in an epidemiological model!