

Analyzing occupancy data using betapart

Emily Beasley

November 15, 2018

Quantifying changes in species composition through space or time is an essential component of community ecology. The following code creates plots for a visual assessment of turnover of small mammal communities over two years of sampling. The package betapart is then used to quantify the temporal turnover.

The data used here are the result of sampling small mammals in 16 glades in southwest Missouri in summer 2016 and 2017. Glades were clustered into four sampling units. The raw occupancy data was used as the input for a multi-species occupancy model, which accounts for detection error inherent in the sampling process. The posterior output is loaded here as occmats.

Create plots of occupancy probabilities to visualize changes between sampling periods.

```
#Load data
occ mats <- readRDS(file = "Zs.RDS")

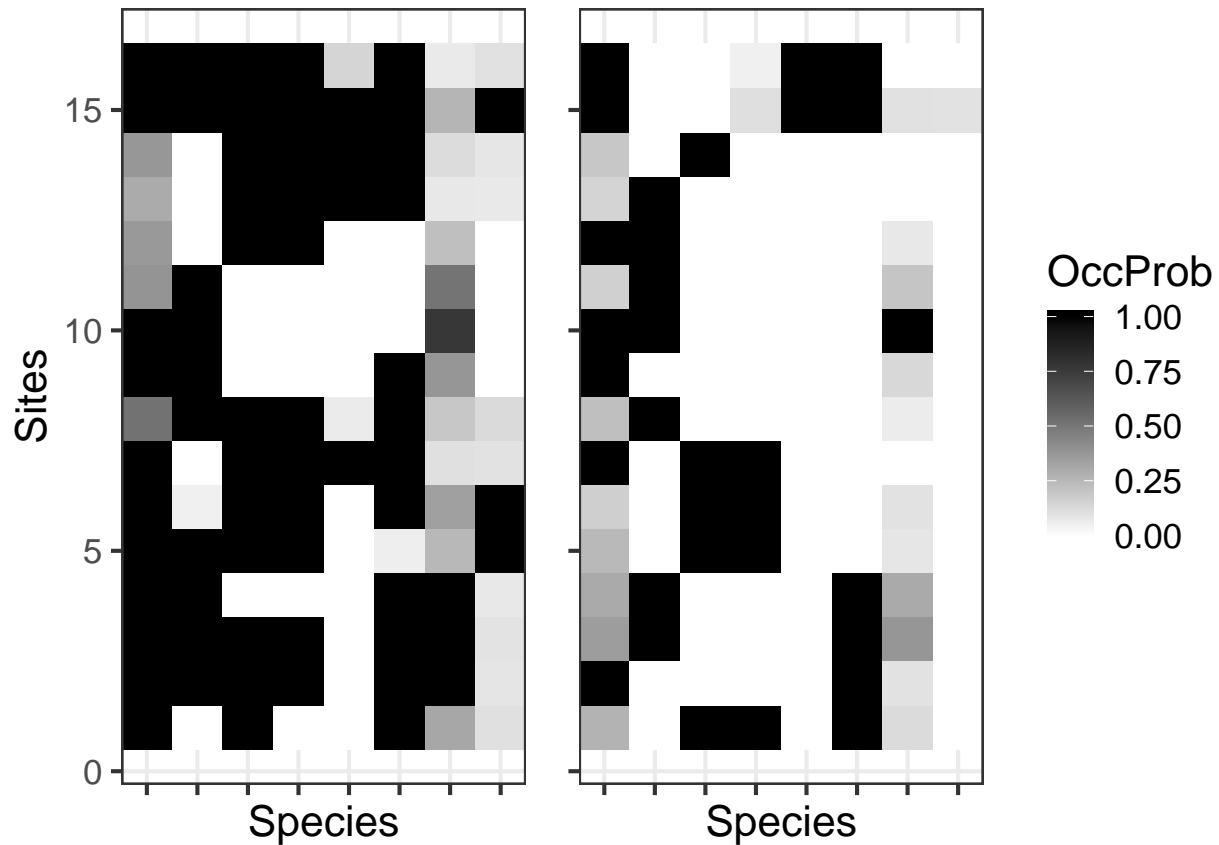
#Convert to long format for plotting
long.dat <- function(x){
  x <- as.data.frame(x)
  x$Sites <- c(1:16)
  x %>%
    gather(Spec1:Spec8, key = "Species", value = "OccProb")
}

#Run above function
long2016 <- long.dat(x = occ mats[[1]])
long2017 <- long.dat(x = occ mats[[2]])

#Plot occupancy data as a raster/tile geom
occplot1 <- ggplot(data = long2016, aes(x = Species, y = Sites, fill = OccProb))+
  geom_tile()+
  scale_fill_gradient(low = "white", high = "black", guide = F)+
  scale_x_discrete(expand = c(0, 0.5))+
  theme_bw(base_size = 16)+
  theme(axis.text.x = element_blank())

occplot2 <- ggplot(data = long2017, aes(x = Species, y = Sites, fill = OccProb))+
  geom_tile()+
  scale_fill_gradient(low = "white", high = "black")+
  scale_x_discrete(expand = c(0, 0.5))+
  theme_bw(base_size = 16)+
  theme(axis.title.y = element_blank(), axis.text.y = element_blank(),
        axis.text.x = element_blank())

#Put plots together for side-by-side comparison
plots <- occplot1|occplot2
print(plots)
```



Visually, it appears that many species went locally extinct at several sampling sites. The package `betapart` can be used to quantify that change.

```
#betapart can only use matrices with values of 0 and 1
occmats <- lapply(occmats, function(x) ifelse(x < 1, 0, x))

#Calculate Jaccard index using function beta.temp
temporal <- beta.temp(x = occmats[[1]], y = occmats[[2]], index.family = "jaccard")

#Output for beta.temp is a data frame with 3 columns
#beta.jtu indicates 1-for-1 species substitutions
#beta.jne indicates species gain or loss without substitution
#and beta.jac is the full jaccard index
#Values closer to 1 indicate greater dissimilarity
print(temporal)
```

```
##      beta.jtu  beta.jne  beta.jac
## 1         0.5 0.000000 0.500000
## 2         0.0 0.666667 0.666667
## 3         0.0 0.666667 0.666667
## 4         0.0 0.500000 0.500000
## 5         0.0 0.600000 0.600000
## 6         0.0 0.600000 0.600000
## 7         0.0 0.400000 0.400000
## 8         0.0 0.750000 0.750000
## 9         0.0 0.666667 0.666667
## 10        0.0 0.333333 0.333333
```

```
## 11      0.0 0.0000000 0.0000000
## 12      1.0 0.0000000 1.0000000
## 13      1.0 0.0000000 1.0000000
## 14      0.0 0.7500000 0.7500000
## 15      0.0 0.5714286 0.5714286
## 16      0.5 0.1666667 0.6666667
```

It looks like there was quite a bit of variation in temporal turnover, most of it due to gain or loss without substitution. Based on the occupancy plots, it was probably species loss.

The package also does pairwise dissimilarity indices across sites using function `beta.pair()`.

```
dist2016 <- beta.pair(x = occmats[[1]], index.family = "jaccard")
jac2016 <- dist2016$beta.jac
```

```
print(jac2016)
```

```
##           1           2           3           4           5           6           7
## 2  0.5000000
## 3  0.5000000 0.0000000
## 4  0.6000000 0.3333333 0.3333333
## 5  0.6666667 0.4285714 0.4285714 0.7142857
## 6  0.4000000 0.4285714 0.4285714 0.7142857 0.3333333
## 7  0.4000000 0.4285714 0.4285714 0.7142857 0.5714286 0.3333333
## 8  0.6000000 0.3333333 0.3333333 0.6666667 0.5000000 0.5000000 0.5000000
## 9  0.5000000 0.5000000 0.5000000 0.2500000 0.6666667 0.6666667 0.6666667
## 10 0.7500000 0.6666667 0.6666667 0.5000000 0.6000000 0.8333333 0.8333333
## 11 1.0000000 0.8333333 0.8333333 0.7500000 0.8000000 1.0000000 1.0000000
## 12 0.7500000 0.6666667 0.6666667 1.0000000 0.6000000 0.6000000 0.6000000
## 13 0.6000000 0.5714286 0.5714286 0.8571429 0.7142857 0.5000000 0.2000000
## 14 0.6000000 0.5714286 0.5714286 0.8571429 0.7142857 0.5000000 0.2000000
## 15 0.5714286 0.3750000 0.3750000 0.6250000 0.2857143 0.2857143 0.2857143
## 16 0.4000000 0.1666667 0.1666667 0.5000000 0.3333333 0.3333333 0.3333333
##           8           9          10          11          12          13          14
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9  0.6000000
## 10 0.8000000 0.3333333
## 11 0.7500000 0.6666667 0.5000000
## 12 0.5000000 1.0000000 1.0000000 1.0000000
## 13 0.4000000 0.8333333 1.0000000 1.0000000 0.5000000
## 14 0.4000000 0.8333333 1.0000000 1.0000000 0.5000000 0.0000000
## 15 0.4285714 0.5714286 0.7142857 0.8571429 0.7142857 0.4285714 0.4285714
## 16 0.2000000 0.4000000 0.6000000 0.8000000 0.6000000 0.5000000 0.5000000
##           15
## 2
## 3
## 4
## 5
## 6
## 7
```

```
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16 0.2857143
```