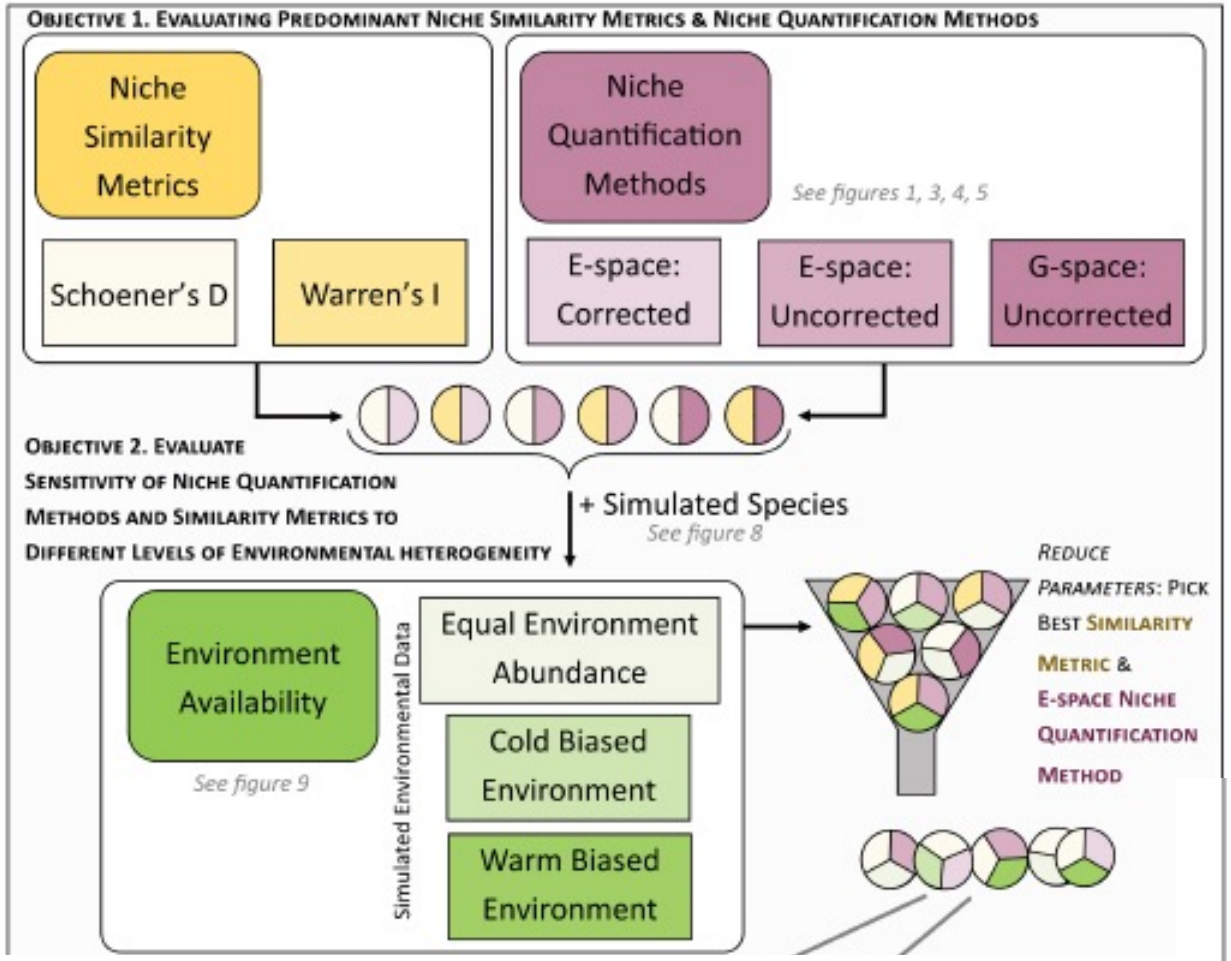
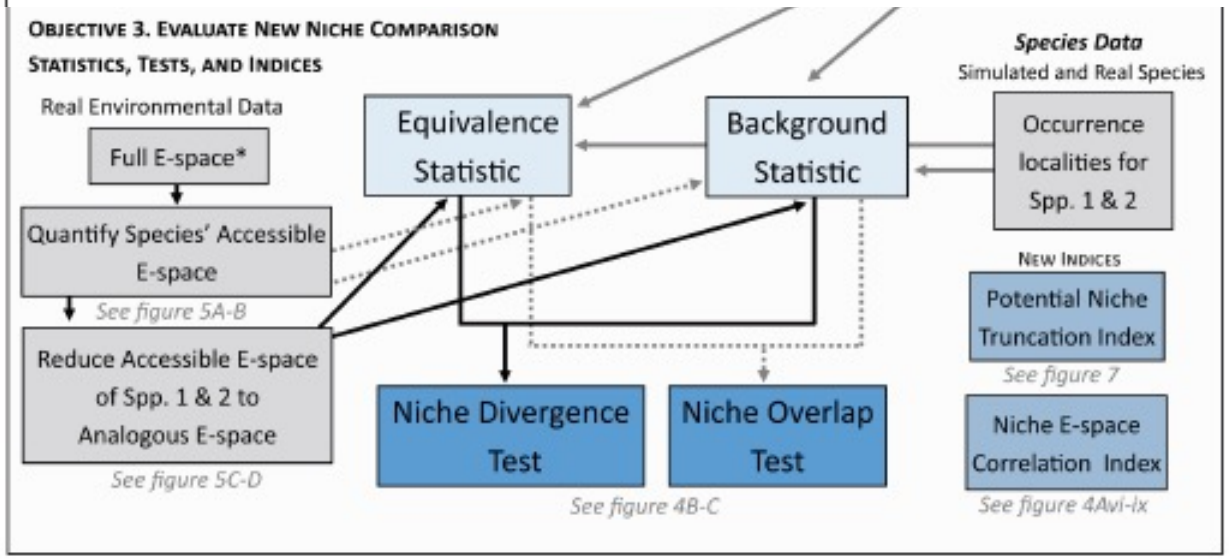
05 - Point Based

Shelly Gaynor
University of Florida

iDigBio
Integrated Digitized Biocollections

Climatic Niche





Brown and Carnaval. 2019. A tale of two niche: methods, concepts, and evolution. Frontiers of Biogeography.

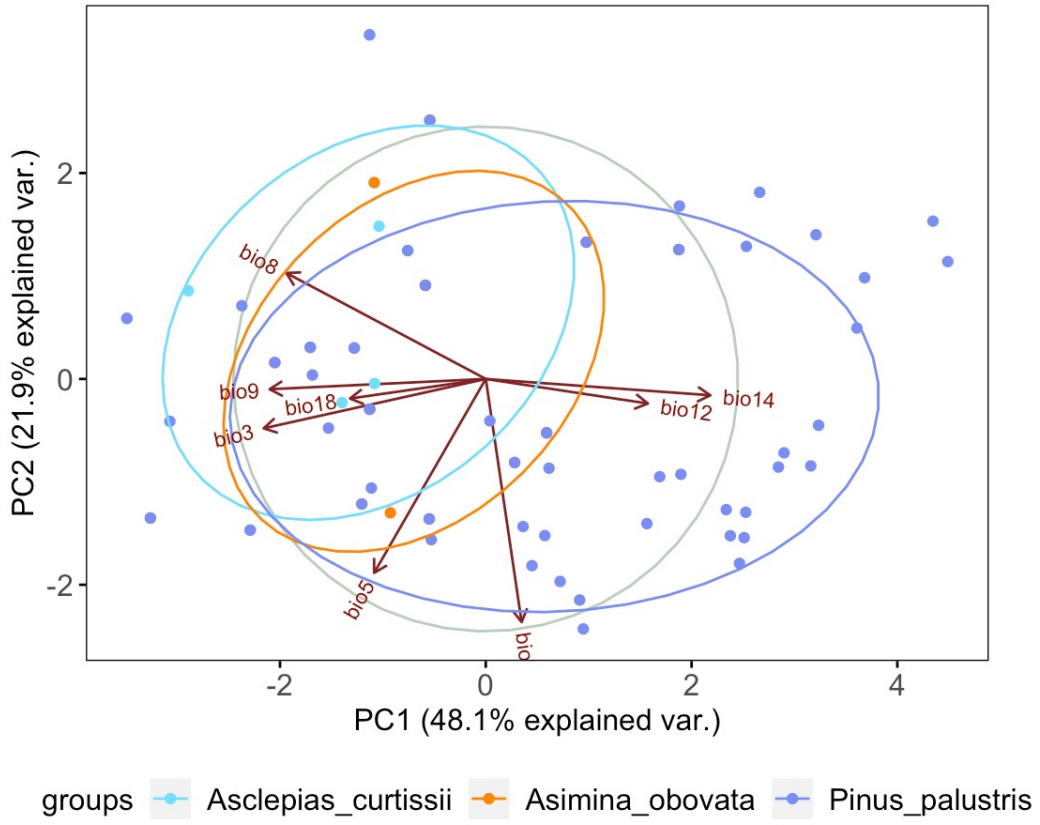
Ecological Niche

Realized Niche

- abiotic conditions that a species can occupy with the presence of biotic interactions

Fundamental Niche

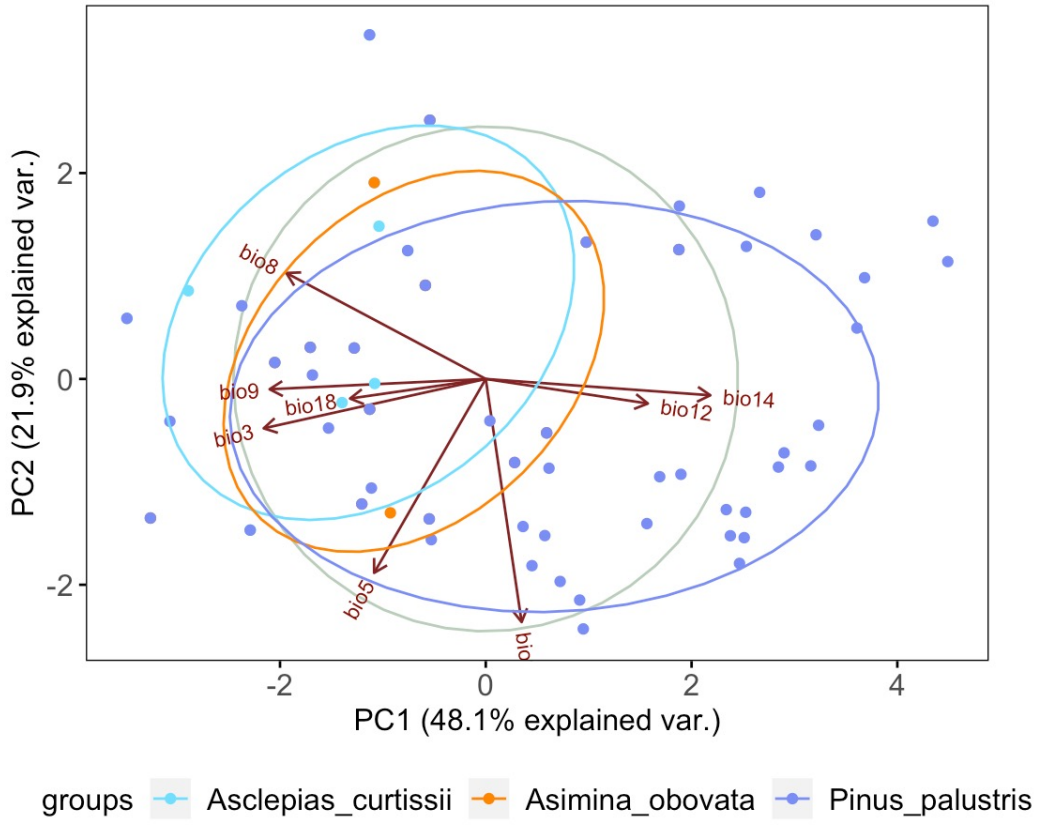
- abiotic conditions a species could potentially occupy in the absence of biotic interactions



Ecological Niche

Realized Niche

- abiotic conditions that a species can occupy with the presence of biotic interactions



Load Packages

```
library(gtools)
library(raster)
library(plyr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(multcompView)
library(gridExtra)
library(ecospat)
library(dismo)
library(ade4)
```

Load Datafiles

Load functions

This function is from vqv/ggbiplot.

```
source("functions/ggbiplot_copy.R")
```

Load data file

```
alldf <- read.csv("data/cleaning_demo/maxent_ready/diapensiaceae_maxentready_20230605.csv")
```

Load Raster Layers

```
list <- list.files("data/climate_processing/PresentLayers/all", full.names = TRUE, recursive = FALSE)
list <- mixedsort(sort(list))
envtStack <- stack(list)
```

Preparing Data

Extract value for each point

For each occurence record, extract the value for each bioclim variables using the package raster.

```
ptExtracted <- raster::extract(envtStack, alldf[2:3])
```

Convert to data frame

```
ptExtracteddf <- as.data.frame(ptExtracted)
```

Add species name

```
ptExtracteddf <- ptExtracteddf %>%
                 dplyr::mutate(name = as.character(alldf$species),
                               x = alldf$longitude,
                               y = alldf$latitude)
```

Drop any NA

```
ptExtracteddf <- ptExtracteddf %>%
                 tidyr::drop_na()
```

Principal Component Analysis

Create two dataframes.

```
data.bioclim <- ptExtracteddf[, 1:8]
data.species <- ptExtracteddf[, 9]
```

Using only the bioclim columns to run the principal components analysis.

```
data.pca <- prcomp(data.bioclim, scale. = TRUE)
```

Principal Component Analysis

```
loadings <- data.pca$rotation
summary(loadings)
```

```
loadings_relative_A <- t(t(abs(loadings))/rowSums(t(abs(loadings))))*100
loadings_relative_A
```

```
##              PC1       PC2       PC3         PC4       PC5       PC6       PC7
## bio_3   8.591746 18.290740 15.598618 26.50100048 15.693465  2.892773 10.826188
## bio_7  15.106599  3.605906 20.544960  4.56019510  8.196901  6.586901 27.743419
## bio_8  14.713687  1.268164 13.920838  0.01157387 24.676885 22.390720  4.151105
## bio_9   9.541759 24.350510  8.759713 16.03534080  7.639386 25.535124  3.884780
## bio_14 16.784509  5.136389  1.759000 16.24120193 10.151969  6.787089  7.464726
## bio_15  6.003487 25.442930 21.469601  8.28251442  9.927169  9.084507  6.348032
## bio_18 15.380721  6.468852 14.248091  7.35172686 17.912054 10.264436 18.683458
## elev   13.877493 15.436508  3.699179 21.01644654  5.802170 16.458450 20.898292
##               PC8
## bio_3   0.2683966
## bio_7  10.5755001
## bio_8   6.4386942
## bio_9   2.3740267
## bio_14 31.8174754
## bio_15 18.6736844
## bio_18 19.1865553
## elev   10.6656673
```

Principal Component Analysis

Set theme

First, I made a theme to change the background of the plot. Next, I changed the plot margins and the text size.

```
theme <- theme(panel.background = element_blank(),
               panel.border=element_rect(fill=NA),
               panel.grid.major = element_blank(),
               panel.grid.minor = element_blank(),
               strip.background=element_blank(),
               axis.ticks=element_line(colour="black"),
               plot.margin=unit(c(1,1,1,1),"line"),
               axis.text = element_text(size = 12),
               legend.text = element_text(size = 12),
               legend.title = element_text(size = 12),
               text = element_text(size = 12))
```

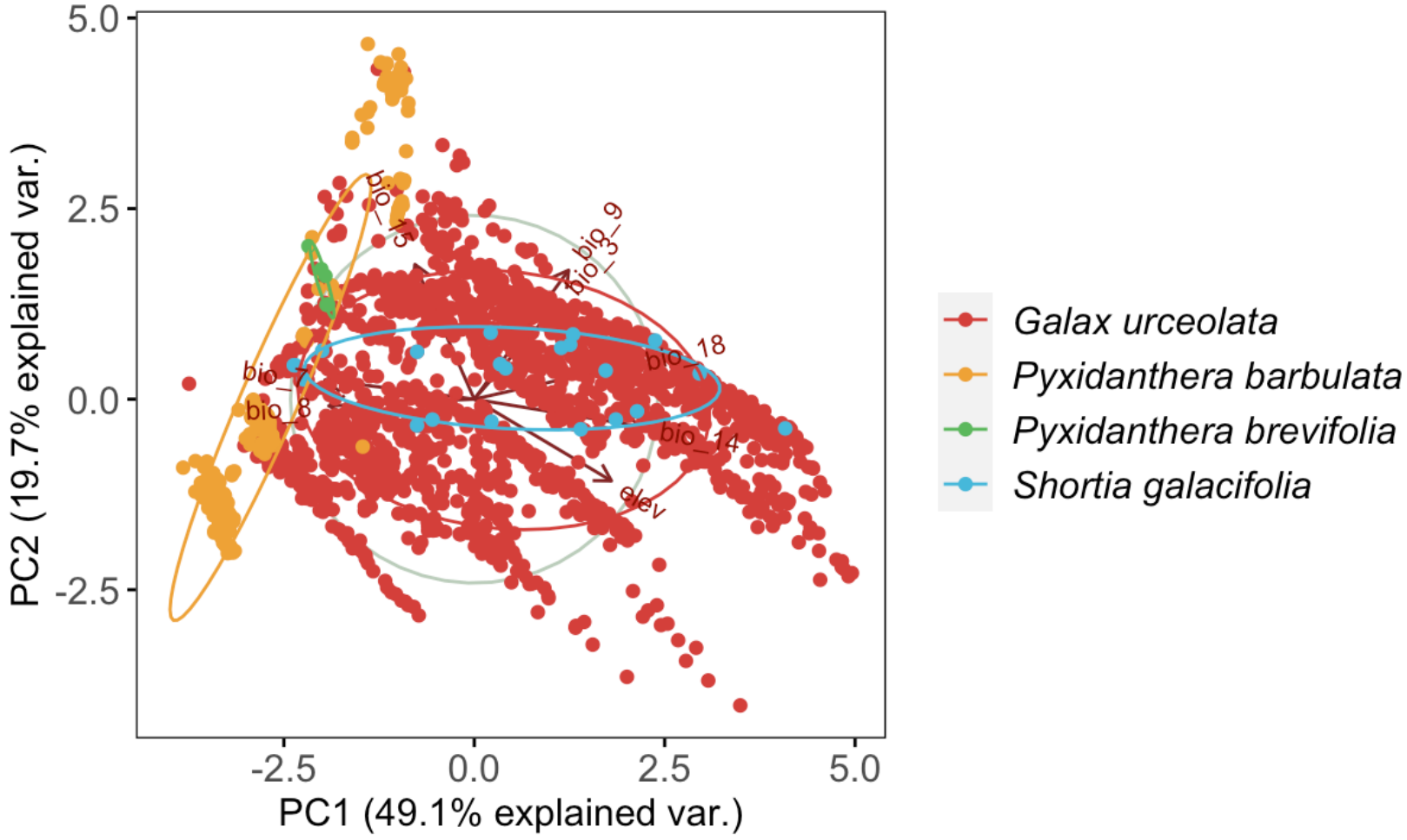
Set color palette

```
pal <- c("#D43F3AFF", "#EEA236FF", "#5CB85CFF", "#46B8DAFF")
```

Principal Component Analysis

```
g <- ggbiplot(data.pca, obs.scale = 1, var.scale = 1,
              groups = data.species, ellipse = TRUE, circle = TRUE) +
     scale_color_manual(name = '', values = pal) +
     theme(legend.direction = 'vertical', legend.position = 'bottom',
           legend.text = element_text(size = 12, face = "italic")) +
     theme

g
```

Principal Component Analysis



ANOVA

Simple function to run an ANOVA and a post-hoc Tukey-HSD test

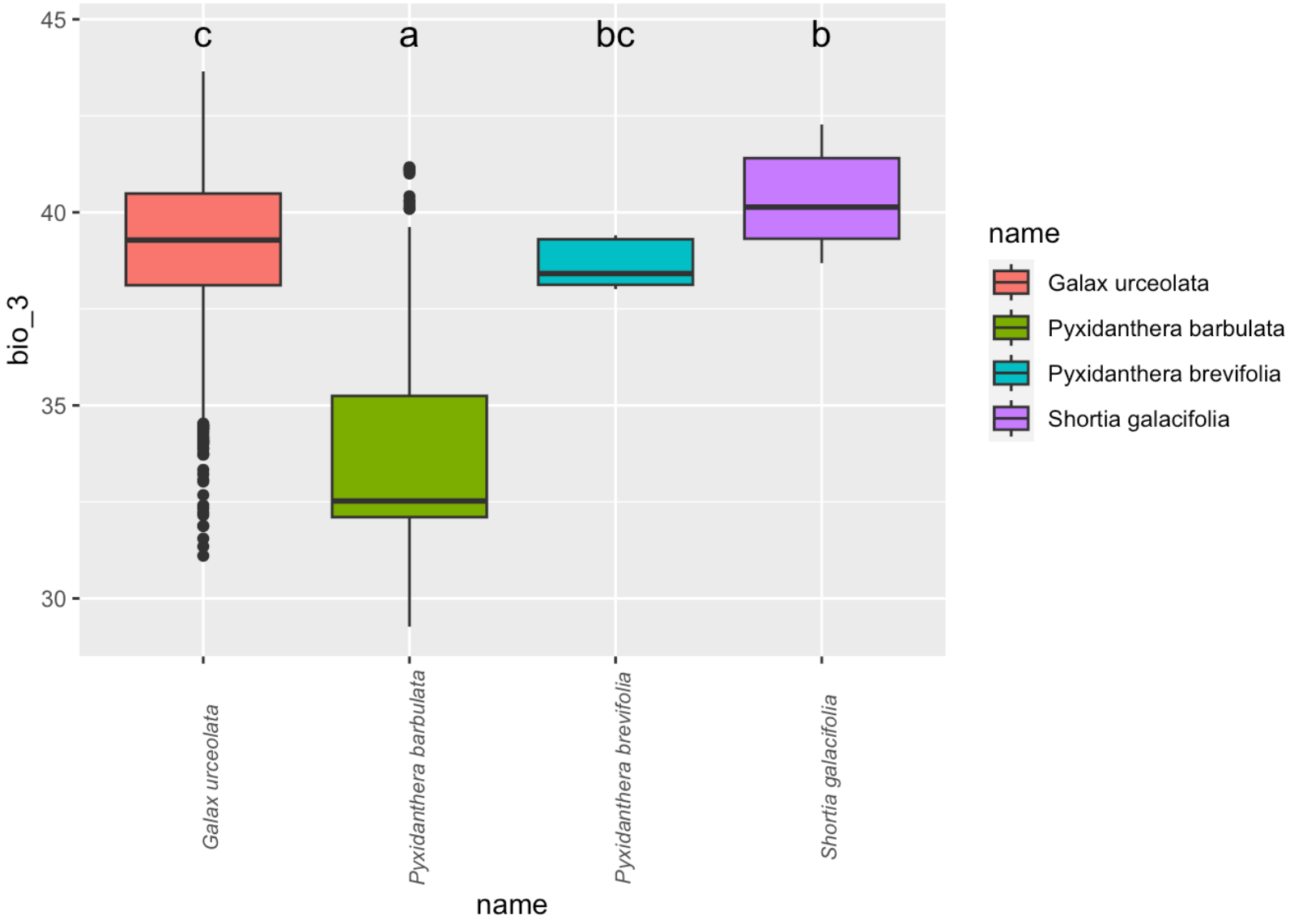
```
stat.test <- function(dataframe,  x = "name", y){
    bioaov <- aov(as.formula(paste0(y,"~",x)), data = dataframe)
    TH <- TukeyHSD(bioaov, "name")
    m <- multcompLetters(TH$name[,4])
    y.val <-  as.numeric(max(dataframe[[y]]) + 1)
    groups <- data.frame(groups = m$Letters, name = names(m$Letters), y.val = rep(y.val, 4))
    return(groups)
}
```

ANOVA

Run for bio_3 only

```
bio3aovplot <- ggplot(ptExtracteddf, aes(x = name, y = bio_3)) +
               geom_boxplot(aes(fill = name)) +
               scale_color_manual(name = '', values = pal) +
               geom_text(data = stat.test(dataframe =ptExtracteddf,  y = "bio_3"),
                         mapping = aes(x = name,
                                       y = max(ptExtracteddf["bio_3"]+1),
                                       label = groups),
                         size = 5, inherit.aes = FALSE) +
               theme(axis.text.x = element_text(angle = 90, size = 8, face = 'italic'))
bio3aovplot
```

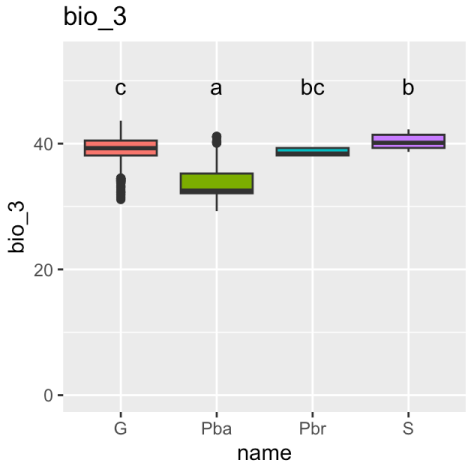
ANOVA

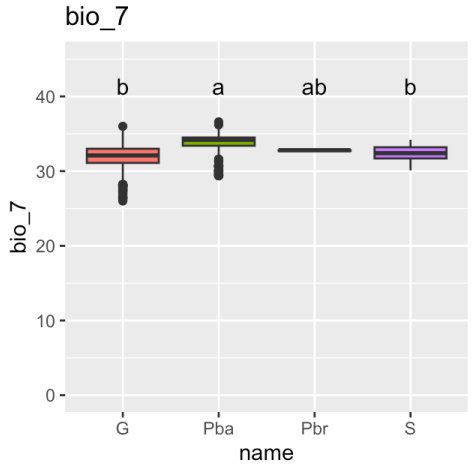

45
40
35
30
bio_3
c
a
bc
b
Galax urceolata
Pyxidanthera barbulata
Pyxidanthera brevifolia
Shortia galacifolia
name
name
Galax urceolata
Pyxidanthera barbulata
Pyxidanthera brevifolia
Shortia galacifolia

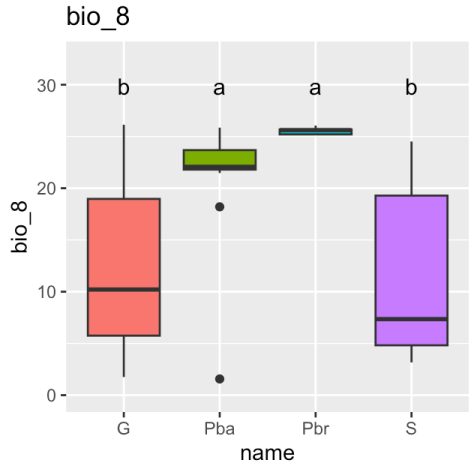
ANOVA

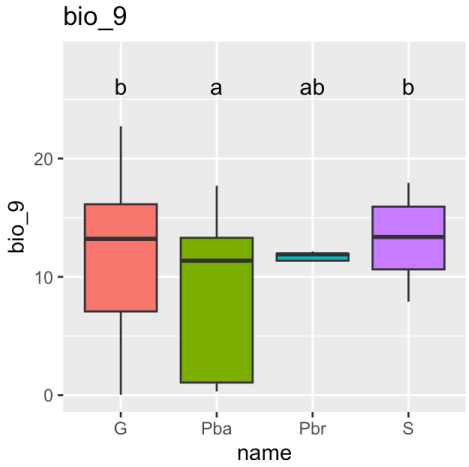
Loop through all variables

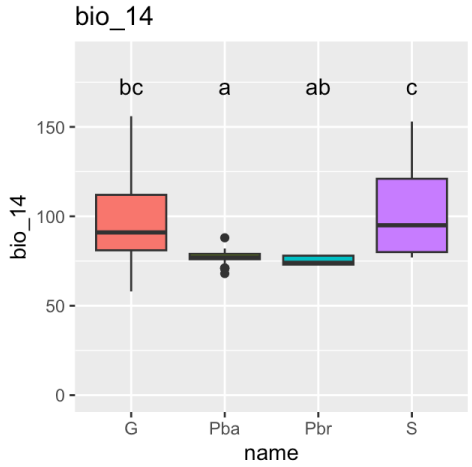
```
variablelist <- colnames(ptExtracteddf)[1:8]
plotlist <- list()
maxstats <- c()
statsout <- c()

for(i in 1:8){
  vname <- variablelist[i]
  maxstats[i] <- as.numeric(max(ptExtracteddf[[vname]]) + 1)
  statsout[[i]] <- stat.test(dataframe = ptExtracteddf, y = vname)
  plotlist[[i]] <- ggplot(ptExtracteddf, aes(x = name, y = .data[[vname]])) +
                    geom_boxplot(aes(fill = name)) +
                    scale_colour_manual(name = 'Species', values = pal) +
                    geom_text(data = statsout[[i]],
                              mapping = aes(x = name,
                                            y = (y.val*1.1),
                                            label = groups),
                              size = 4, inherit.aes = FALSE) +
                    scale_x_discrete(labels = c('G', 'Pba','Pbr', 'S')) +
                    ggtitle(label = vname) +
                    ylab(vname) +
                    theme(legend.position = "none") +
                   ylim(0, (maxstats[i]*1.2))
}

gridExtra::grid.arrange(grobs = plotlist)
```

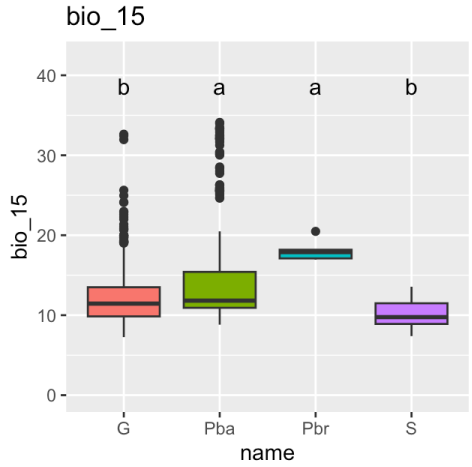

bio_3
c
a
bc
b
bio_3
40
20
0
G
Pba
Pbr
S
name

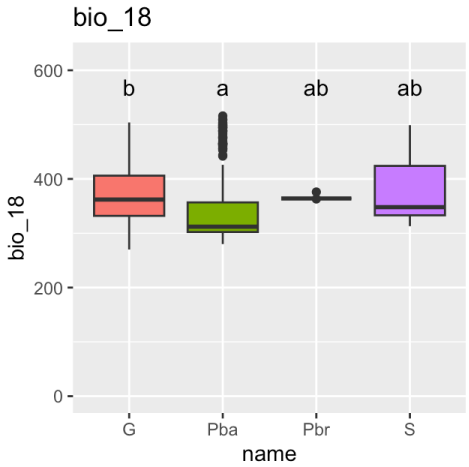

bio_7
b
a
ab
b
bio_7
40
30
20
10
0
G
Pba
Pbr
S
name

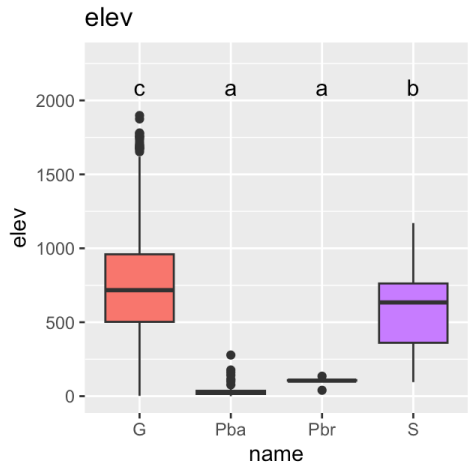

bio_8
b
a
a
b
bio_8
30
20
10
0
G
Pba
Pbr
S
name


bio_9
b
a
ab
b
bio_9
20
10
0
G
Pba
Pbr
S
name


bio_14
bc
a
ab
c
bio_14
150
100
50
0
G
Pba
Pbr
S
name


bio_15
b
a
a
b
bio_15
40
30
20
10
0
G
Pba
Pbr
S
name


bio_18
b
a
ab
ab
bio_18
600
400
200
0
G
Pba
Pbr
S
name


elev
c
a
a
b
elev
2000
1500
1000
500
0
G
Pba
Pbr
S
name

Ecospat

Set up background points

```
bg1 <- randomPoints(mask = envtStack, n = 1000, p = alldf[,3:2])
```

```
## Warning in .couldBeLonLat(x, warnings = warnings): CRS is NA. Assuming it is
## longitude/latitude
```

```
bg1.env <- raster::extract(envtStack, bg1)
bg1.env <- data.frame(bg1.env)
allpt.bioclim <- rbind(bg1.env, data.bioclim)
```

dudi.PCA to reduce variables

```
pca.env <- dudi.pca(allpt.bioclim,
                    center = TRUE, # Center by the mean
                    scannf = FALSE, # Don't plot
                    nf = 2) # Number of axis to keep
```

Ecospat

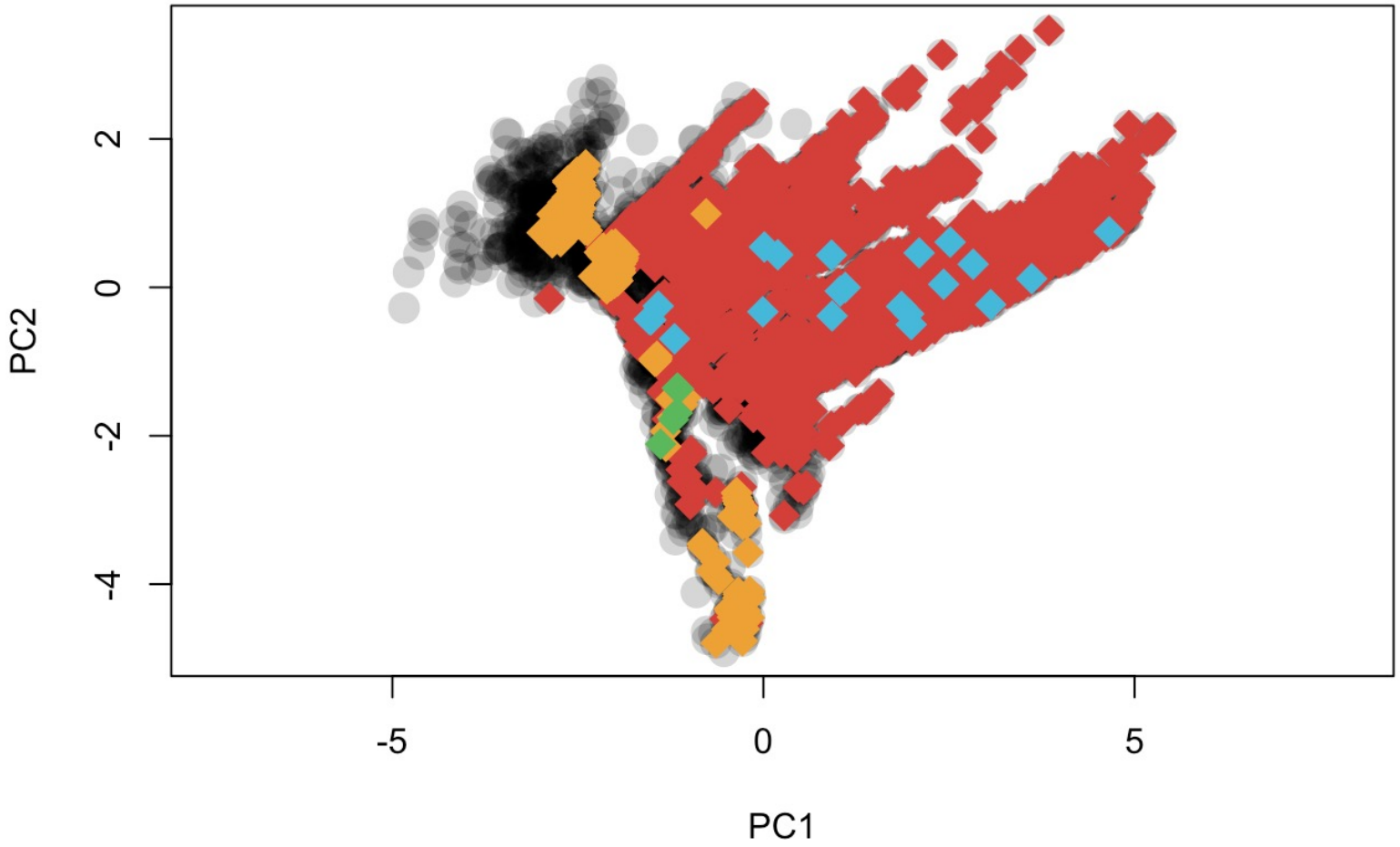
Pull out scores for each species

```
p1.score <- suprow(pca.env, dplyr::filter(ptExtracteddf, name == "Galax urceolat
a")[, 1:8])$li
p2.score <- suprow(pca.env, dplyr::filter(ptExtracteddf, name == "Pyxidanthera ba
rbulata")[, 1:8])$li
p3.score <- suprow(pca.env, dplyr::filter(ptExtracteddf, name == "Pyxidanthera br
evifolia")[, 1:8])$li
p4.score <- suprow(pca.env, dplyr::filter(ptExtracteddf, name == "Shortia galacif
olia")[, 1:8])$li
scores.clim <- pca.env$li
```

Visualize

```
plot(scores.clim, pch = 16, asp = 1,
     col = adjustcolor(1, alpha.f = 0.2), cex = 2,
     xlab = "PC1", ylab = "PC2")
points(p1.score, pch = 18, col = pal[1], cex = 2)
points(p2.score, pch = 18, col = pal[2], cex = 2)
points(p3.score, pch = 18, col = pal[3], cex = 2)
points(p4.score, pch = 18, col = pal[4], cex = 2)
```

Ecospat



Ecospat

Kernel density estimates

Create occurrence density grids based on the ordination data.

```
z1 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, p1.score, R = 100)
z2 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, p2.score, R = 100)
z3 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, p3.score, R = 100)
z4 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, p4.score, R = 100)
zlist  <- list(z1, z2, z3, z4)
```

Niche Overlap

Schoener's D ranges from 0 to 1. 0 represents no similarity between niche space. 1 represents completely identical niche space.

```
overlapD <- matrix(ncol = 2, nrow = 7)
n <- 1
for(i in 1:3){
  for(j in 2:4){
    if(i != j){
      overlapD[n, 1]<- paste0("z", i, "-", "z", j)
      overlapD[n, 2]<- ecospat.niche.overlap(zlist[[i]], zlist[[j]], cor = TRUE)$D
      n <- n + 1
    }
  }
}

overlapDdf <- data.frame(overlapD)
overlapDdf
```

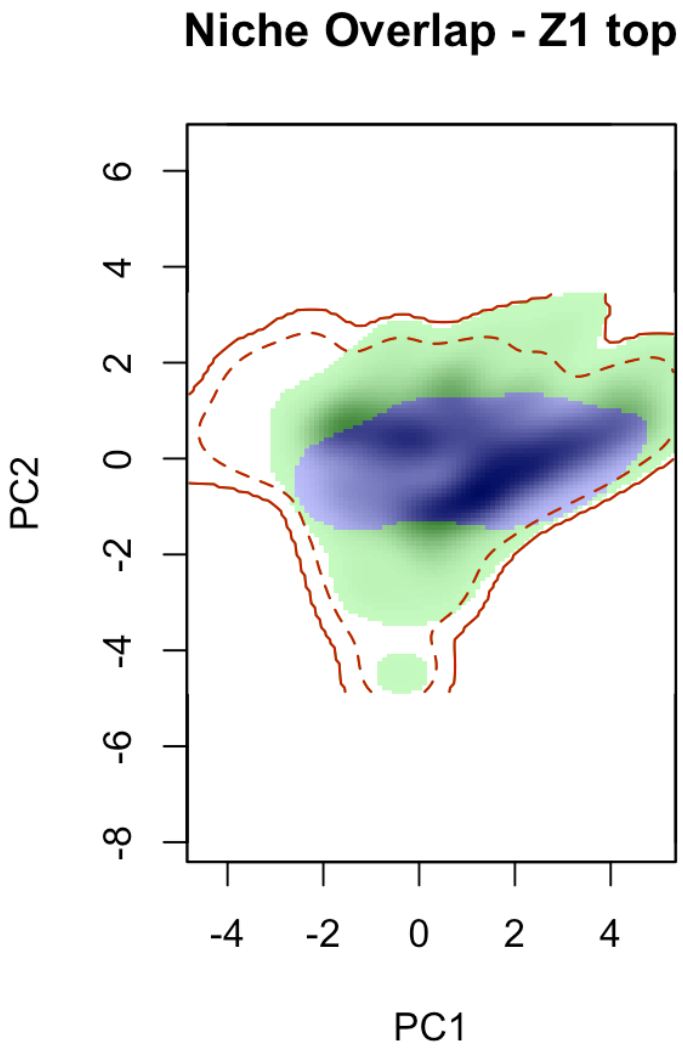
```
##      X1                  X2
## 1 z1-z2  0.0890148692326306
## 2 z1-z3 0.00297245638991617
## 3 z1-z4   0.453009449060948
## 4 z2-z3 0.00376518449381513
## 5 z2-z4  0.0739737424962652
## 6 z3-z2 0.00376518449381513
## 7 z3-z4 0.00236297216796688
```

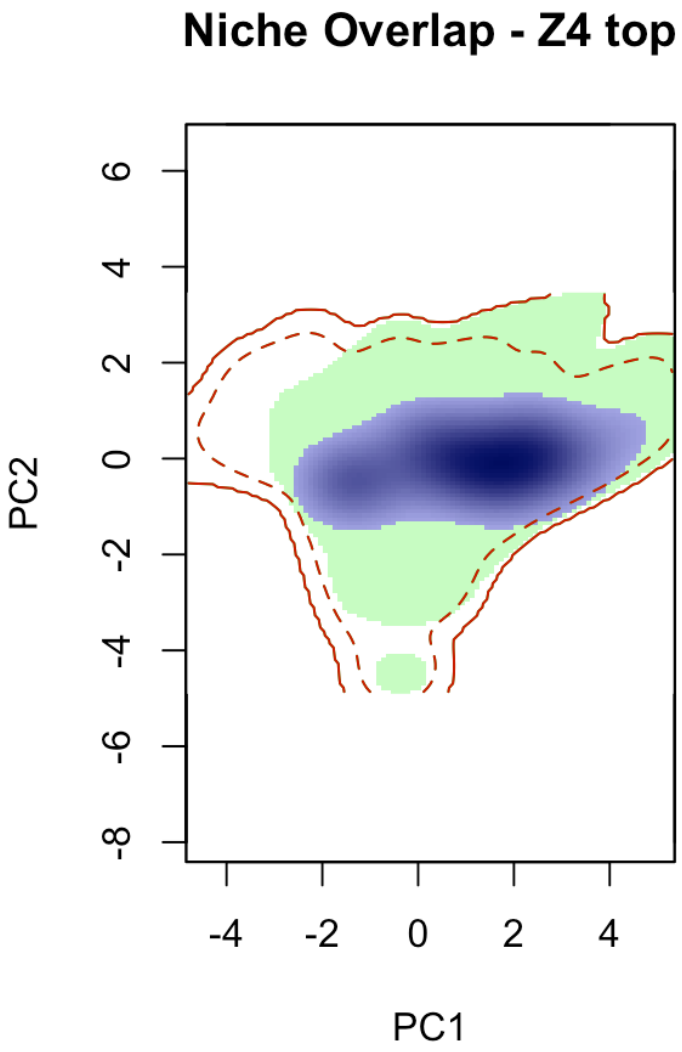
Niche Overlap

Niche Overlap Visualization

```
par(mfrow=c(2,1))
ecospat.plot.niche.dyn(z1, z4, quant=0.25, interest = 1
                       , title= "Niche Overlap - Z1 top", name.axis1="PC1", name.
axis2="PC2")
ecospat.plot.niche.dyn(z1, z4, quant=0.25, interest = 2
                       , title= "Niche Overlap - Z4 top", name.axis1="PC1", name.
axis2="PC2")
```

Niche Overlap





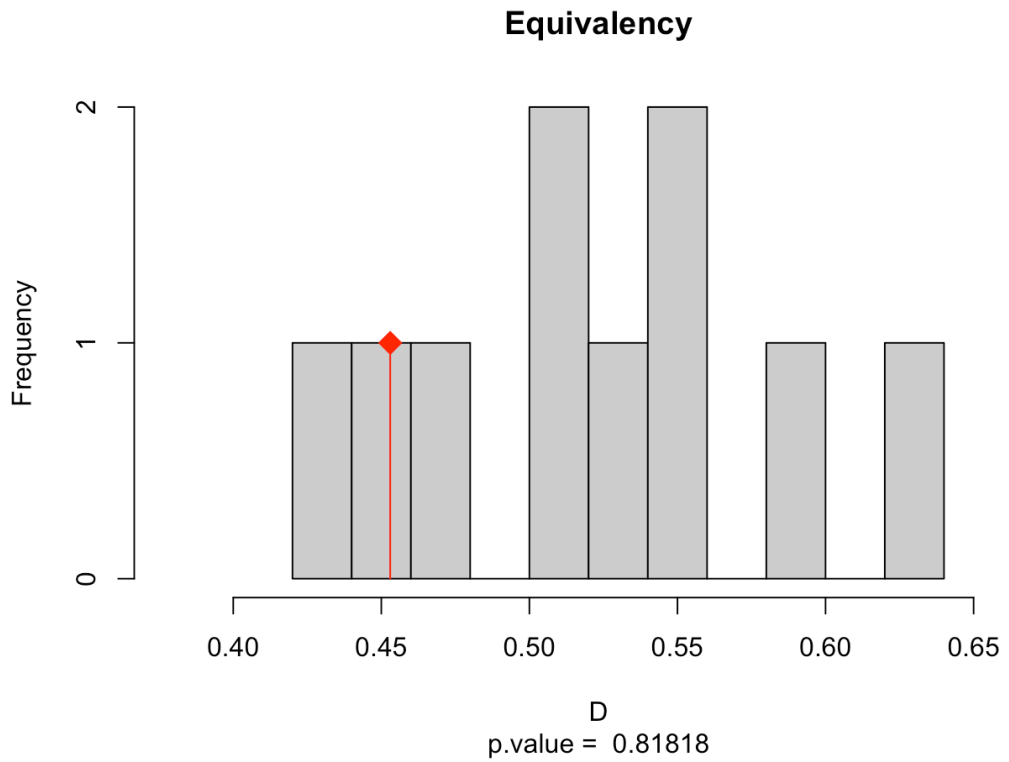
Niche Equivalency Test

Niche Equivalency Test

Based on Warren et al. 2008 - Are the two niche identical?

Hypothesis test for D, null based on randomization. H1: the niche overlap is higher than expected by chance (or when randomized).

```
eq.test <- ecospat.niche.equivalency.test(z1, z4, rep = 10)
ecospat.plot.overlap.test(eq.test, "D", "Equivalency")
```



Niche Similarity Test

Based on Warren et al. 2008 - Are the two niche similar?

Can one species' niche predict the occurrences of a second species better than expected by chance?

```
sim.test <- ecospat.niche.similarity.test(z1, z4, rep = 10, rand.type=2)
ecospat.plot.overlap.test(sim.test, "D", "Similarity")
```