# Global patterns of biodiversity

*Jesús N. Pinto-Ledezma and Jeannine Cavender-Bares*

In this lab, you will learn basic tools in R for visualizing species distributions, build geographical ranges, testing drivers of gradients of biodiversity under different approaches. You will need three datasets, that will be provided for you: 1. Species occurrence data points - live.oaks.txt 2. Species geograhical ranges - Furnarii_ranges_geo.shp 3. Environmental predictors - bio1.bil and bio12.bil

## Set up your data and your working directory

Set up a working directory and put the two data files in that directory. Tell R that this is the directory you will be using, and read in your data:

```
setwd("path/for/your/directory")
```

Install and load the following packages

```
require(maptools)
require(rgdal)
require(raster)
require(sp)
require(rangeBuilder)
library(spdep)
library(ncf)
require(geiger)
require(dismo)
```

## From point occurrences to range maps

Load species occurrences data points. We will occurrences from Live oaks, that were obtained from iDigBio between 20 and 24 July 2018 by Jeannine Cavender-Bares. Notice that these occurrence data points were visually examined and any localities that were outside the known range of the species, or in unrealistic locations (e.g., water bodies) were discarded.

```
oaks <- read.table("Data/live.oaks.txt", header = TRUE)
```

```
head(oaks)

tail(oaks)
```

Plot the points (x = Logitude, y = Latitude) and a world map, for reference. We need to load a dataobject from the maptools package

```
data(wrld_simpl)

{plot(oaks[c(2:3)], col = "blue", pch = 19)
plot(wrld_simpl, add = TRUE)}
```

Lets try with a single species in this case, Quercus virginiana

```
unique(oaks$Species)

que_vir <- subset(oaks, oaks$Species == "Quercus_virginiana")
```

```
{plot(que_vir$Longitude, que_vir$Latitude, pch = 15)
plot(wrld_simpl, add = TRUE)}
```

Cool, right?

## Data checking and cleaning

Check if there are any duplicated points. We will check if exist any duplicate occurrence data point.

```
oaks_dups <- duplicated(oaks[, c(2:3)])
### NOTE: the function "duplicated" returns the results of a logical test (e.g. TRUE or FALSE)
# How many are duplicates?
length(which(oaks_dups == TRUE))
# How many are NOT duplicates?
length(which(oaks_dups == FALSE))
# Keep only those lines that are not duplicates
oaks_dups_row <- which(oaks_dups == TRUE)
# What's the size? That is, how many points are duplicates
length(oaks_dups_row)
# Create another object withoyt the duplicate records
oaks_nodups <- oaks[-oaks_dups_row,]
# What are the dimensions of the new object?
dim(oaks_nodups)
# Take a look at the first rows of data
head(oaks_nodups)
```

What was the result? Are there duplicated occurrences?

Lets plot the results!

```
{plot(oaks$Longitude, oaks$Latitude, pch = 19, col = "red", cex = 2)
points(oaks_nodups$Longitude, oaks_nodups$Latitude, pch = 16, col = "black")}
```

Better. You can mess around with tree plotting functions in plot.phylo() as much as you'd like. Try this for example

## Range maps from point data

This part can be used to create "simple" range maps based on geometry (e.g. minimum convex polygons, etc.), without considering environmental variables (no ENMs or SDMs).

### Convex hull (minimum convex polygon)

This model draws a convex hull around all 'presence' points.

```
# create a polygon around the species' records
oaks_hull <- convHull(oaks_nodups[, c(2:3)])
```

```
# Plot the created polygon
{plot(oaks_hull)
points(oaks_nodups$Longitude, oaks_nodups$Latitude, pch = 16, col = "black")}
```

Ok, now let's try with a single species,

```
que_vir <- subset(oaks_nodups, oaks_nodups$Species == "Quercus_virginiana")
```

```
que_vir_hull <- convHull(que_vir[, c(2:3)])
```

```
{plot(que_vir_hull)
points(que_vir$Longitude, que_vir$Latitude, pch = 16, col = "black")}
```

Now lets plot all ranges.

```
# Plot all live oaks
{plot(oaks_hull)
points(oaks_nodups$Longitude, oaks_nodups$Latitude, pch = 16, col = "black")

# Plot only Quecus virginiana
plot(que_vir_hull, add = TRUE)
points(que_vir$Longitude, que_vir$Latitude, pch = 16, col = "red")

# Add world maps
plot(wrld_simpl, add = TRUE)}
```

Ok, using simple convex hull seems not to be a good tool for live oaks, now lets try another approach. Now, we will use the dynamic alpha hull from the package rangeBuilder.

### Dynamic Alpha hull

```
que_vir_alphahull <- getDynamicAlphaHull(que_vir, fraction = 0.95,
                                         coordHeaders = c("Longitude", "Latitude"),
                                         clipToCoast = 'no')[[1]]
```

Now plot and see the differences.

```
{plot(que_vir_alphahull, lwd = 2, col = "red")
plot(que_vir_hull, add = TRUE, lwd = 2, lty = 2)
points(que_vir$Longitude, que_vir$Latitude, pch = 16, col = "green")
plot(wrld_simpl, add = TRUE, lwd = 2)}
```

Is there a difference? Please, explain the difference!

Until here we explored how to plot, clean and build species geographical ranges using occurrences. Now will use species geographical ranges of the largest continental endemic radiation (Furnariides) to explore the geographical gradients of species diversity.

## Diversity gradients

### Prepare data and mapping

The geographical ranges correspond to the Infraorder Furnariides (Aves). This data is available thorough BirdLife International (http://datazone.birdlife.org/species/requestdis) and you can use any other group available on IUCN or BIEN. In any case, you first need to download the polygons in shapefile format.

To load the Furnariides geographical ranges we will use the function **readShapePoly** from the package **maptools**

```r
franges <- readShapePoly("Data/Franges/Furnarii_ranges_geo.shp")
str(franges)
```

Now explore the data inside the ranges. Notice that to access to the information, we will use **@** instead of** $**.

```r
head(franges@data)
```

## Raster of species richness

First create an empty raster for the Neotropics using the extent of the furnariides ranges under a spatial resolution of 1º long-lat.

```r
neo_ras <- raster()
# Set the raster "extent"
extent(neo_ras) <- extent(franges)
res(neo_ras) <- 1

neo_ras

values(neo_ras) <- 0
```

Now using the empty raster we will **rasterize** the species identities in each cell or pixel. The resulting raster will be the species richness of Furnariides across the Neotropics.

```r
f_sr_raster <- rasterize(franges, neo_ras, field = "SCINAME",
                         fun = function(x,...){length(unique(na.omit(x)))})
```

Plot the raster.

```r
{plot(f_sr_raster)
plot(wrld_simpl, add = T)}
```

Let's try changing the colors.

```r
#change the color scale
colfuncYellows <- colorRampPalette(c("#d7191c", "#fdae61", "#ffffbf", "#abd9e9",
                                     "#2c7bb6"))
```

```r
{plot(f_sr_raster, col = rev(colfuncYellows(100)), axes = FALSE, box = FALSE,
     zlim = c(minValue(f_sr_raster), maxValue(f_sr_raster)),
     xlab = "Furnariides richness", legend.width = 2)
plot(wrld_simpl, add = T)}
```

Awesome, right?. Now, please describe the observed pattern!

Lets try to rasterize other information from the polygon data set. We will use the information in the column **RD**, this data correspond to the numbers of nodes from the tips to the root of a phylogenetic tree or just **root distance**, thus, will use the RD to calculate the MRD metric **(mean root distance)** that measures the evolutionary derivedness of species within an assemblage (Kerr & Currie, 1999) and can be used to determine whether a local fauna is constituted primarily by early-diverged or by recently originated species (Hawkins et al., 2012).

```
head(franges@data)

f_MRD_raster <- rasterize(franges, neo_ras, field = "RD", fun = mean)
```

```
{plot(f_MRD_raster)
plot(wrld_simpl, add = T)}
```

Let’s try changing the colors.

```
{plot(f_MRD_raster, col = rev(colfuncYellows(100)), axes = FALSE, box = FALSE,
     zlim = c(minValue(f_MRD_raster), maxValue(f_MRD_raster)),
     xlab = "Furnariides mean root distance", legend.width = 2)
plot(wrld_simpl, add = T)}
```

Let’s plot the two raster.

```
{par(mfrow = c(1, 2))
plot(f_sr_raster, col = rev(colfuncYellows(100)), axes = FALSE, box = FALSE,
     zlim = c(minValue(f_sr_raster), maxValue(f_sr_raster)),
     xlab = "Furnariides richness", legend.width = 2)

plot(f_MRD_raster, col = rev(colfuncYellows(100)), axes = FALSE, box = FALSE,
     zlim = c(minValue(f_MRD_raster), maxValue(f_MRD_raster)),
     xlab = "Furnariides mean root distance", legend.width = 2)}
```

Hmmm. What happened in here? So, please answer the next questions.

*From the mean root distance map, it is possible to explain the Furnariides diversity gradient? If so, please explain from an evolutionary perspective*

Let’s be more quantitative in describing that realtionship with a linear model.

## Scale dependency

Now we will explore one of the oldest problems in ecology and evolution, the **scale dependency** in the data. So to explore this scale dependence, we will rasterize the Furnariides ranges, but using different spatial resolutions from 2º to 4º degrees of long-lat.

```
# 2º degrees
neo_ras_2dg <- raster()
# Set the raster "extent"
extent(neo_ras_2dg) <- extent(franges)
res(neo_ras_2dg) <- 2
neo_ras_2dg
values(neo_ras_2dg) <- 0

# 4º degrees
neo_ras_4dg <- raster()
# Set the raster "extent"
extent(neo_ras_4dg) <- extent(franges)
res(neo_ras_4dg) <- 4
neo_ras_4dg
values(neo_ras_4dg) <- 0
```

```
# Furnariides at 2º of long-lat
f_sr_2dg_raster <- rasterize(franges, neo_ras_2dg, field = "SCINAME", fun = function(x,...){length(uniq
# Furnariides at 4º of long-lat
f_sr_4dg_raster <- rasterize(franges, neo_ras_4dg, field = "SCINAME", fun = function(x,...){length(uniq
```

Plot the three maps.

```
{par(mfrow = c (1, 3))
plot(f_sr_raster, main = "Furnariides richness 1dg")
plot(wrld_simpl, add = T)

plot(f_sr_2dg_raster, main = "Furnariides richness 2dg")
plot(wrld_simpl, add = T)

plot(f_sr_4dg_raster, main = "Furnariides richness 4dg")
plot(wrld_simpl, add = T)}
```

So, is there a scale effect?

*Explain the differences between the three maps*

## Correlative relationships

Load the environmental variables that correspond to bio1 (Annual Mean Temperature) and bio12 (Annual Precipitation). These data correspond to two variables out of 19 from WorldClim (http://www.worldclim.org/current). We will use these two variables just for educational purposes, rather to make a complete evaluation of the species-environmental relationships.

```
bio1 <- raster("Data/Envi/bio1.bil")
bio1
bio12 <- raster("Data/Envi/bio12.bil")
bio12
```

Plot the

```
{par(mfrow = c(2, 1))
plot(bio1)
plot(bio12)}
```

Ok, the bio1 and bio12 layers are at global scale, so now will will make a crop to the extent of the Neotropics.

```
bio1_neo <- crop(bio1, extent(franges))
bio12_neo <- crop(bio12, extent(franges))
```

```
{par(mfrow = c(1, 2))
plot(bio1_neo, main = "Annual Mean Temperature")
plot(bio12_neo, main = "Annual Precipitation")}
```

Much better!

Now we will obtain the coordinates from the Furnariides diversity raster. These coordinates then will be used to extract the information from the bio1 and bio12 climatic layers.

```
f_ras_coords <- xyFromCell(f_sr_raster, 1:length(values(f_sr_raster)))
```

Obtain the values from bio1 and bio12 for each cell or pixel using the coordinates.

```
f_ras_bio1 <- extract(bio1_neo, f_ras_coords)
f_ras_bio12 <- extract(bio12_neo, f_ras_coords)
```

```
f_ras_rich <- values(f_sr_raster)

fdata <- na.omit(data.frame(f_ras_coords, f_ras_rich, f_ras_bio1, f_ras_bio12))

#f_ras_rich_noNA <- ifelse(is.na(f_ras_rich), 0, f_ras_rich)
#f_ras_bio1_noNA <- ifelse(is.na(f_ras_bio1), 0, f_ras_bio1)
#f_ras_bio12_noNA <- ifelse(is.na(f_ras_bio12), 0, f_ras_bio12)

#fdata_noNA <- data.frame(f_ras_coords, f_ras_rich_noNA, f_ras_bio1_noNA,
#                          f_ras_bio12_noNA)
```

Now make a simple correlation between the Furnariides richness and bio1 and bio12.

```
cor.test(fdata$f_ras_rich, fdata$f_ras_bio1)
```

```
cor.test(fdata$f_ras_rich, fdata$f_ras_bio12)
```

*Which environmental variable is more related with Furnariides richness?*

*Please explain the relationship from an ecological perspective*

```
{par(mfrow = c(1, 2))
plot(fdata$f_ras_bio1, fdata$f_ras_rich, xlab = "Bio 1", ylab = "Richness")

plot(fdata$f_ras_bio12, fdata$f_ras_rich, xlab = "Bio 12", ylab = "Richness")}
```

## Considering spatial autocorrelation

We now that a correlation is not a causation, so to explore the relationship we need to build a model or fit a model. To explore this relationships we will first explore a simple Ordinary Least Square regression or OLS.

```
fols <- lm(f_ras_rich ~ f_ras_bio1 + f_ras_bio12, data = fdata)
summary(fols)
```

*What is telling us this OLS?*

Now, explore the spatial autocorrelation of the Furnariides richness gradient.

```
autocor_SR <- ncf::correlog(fdata$x, fdata$y, z = fdata$f_ras_rich, na.rm = T,
                         increment = 1, resamp = 1)
```

Let's use an correlogram to explore the spatial autocorrelation.

```
{plot(autocor_SR$correlation[1:50], type = "b", pch = 1, cex = 1.2, lwd = 1.5,
     ylim = c(-1, 1), xlab = "Distance class", ylab = "Moran's I", cex.lab = 1.2,
     cex.axis = 1.2)
abline(h = 0)}
```

*Is there a spatial autocorrelation in the data?*

What about the residuals? Now, we will explore the spatial autocorrelation in the residuals.

```
coords <- fdata[1:2]
coords <- as.matrix(coords)
```

Build a neighbourhood contiguity by distance

```
nb1.5 <- dnearneigh(coords, 0, 1.5)
```

Using the neighbourhood contiguity build a spatial weights for neighbours lists.

```
nb1.5.w <- nb2listw(nb1.5, glist = NULL, style = "W", zero.policy = TRUE)
```

Extract the residuals from the OLS model

```
residuals_ols <- residuals(fols)
plot(residuals_ols)
```

Calculate a univariate spatial correlogram.

```
autocor_ols_res <- correlog(fdata$x, fdata$y, z = residuals(fols),
                            increment = 1, resamp = 1)
```

plot the autocorrelagram for the residuals

```
{plot(autocor_ols_res$correlation[1:50], type = "b", pch = 1, cex = 1.2, lwd = 1.5,
     ylim = c(-0.5, 1), xlab = "distance", ylab = "Moran's I", cex.lab = 1.5,
     cex.axis = 1.2)
abline(h = 0)
title(main = "OLS residuals", cex = 1.5)}
```

Ohhh, seems that the residuals have a strong spatial autocorrelation, that is a problem because if we found autocorrelation in the residuals much of the explanation that we obtain can be biased.

Let’s inspect two two autocorrelograms.

```
{par(mfrow = c(2, 1))

plot(autocor_SR$correlation[1:50], type = "b", pch = 1, cex = 1.2, lwd = 1.5,
     ylim = c(-1, 1), xlab = "Distance class", ylab = "Moran's I", cex.lab = 1.2,
     cex.axis = 1.2)
abline(h = 0)
title(main = "OLS model", cex = 1.5)

plot(autocor_ols_res$correlation[1:50], type = "b", pch = 1, cex = 1.2, lwd = 1.5,
     ylim = c(-0.5, 1), xlab = "Distance class", ylab = "Moran's I", cex.lab = 1.5,
     cex.axis = 1.2)
abline(h = 0)
title(main = "OLS residuals", cex = 1.5)}
```

Hmmm, seems that there is a strong spatial autocorrelation, thus any conclusion using the OLS model can be biased.

To try to solve this important issue, we will use **spatial simultaneous autoregressive error model estimation (Aka SAR model)**, this model account for spatial autocorrelation by adding an extra term (autoregressive) in the form of a spatial-weight matrix that specifies the neighborhood of each cell or pixel and the relative weight of each neighbor.

Let’s fit the SAR model.

```
sar_nb1.5.w <- errorsarlm(fols, listw = nb1.5.w, data = fdata, quiet = FALSE,
                          zero.policy = TRUE)
summary(sar_nb1.5.w)
residuals_sar_nb1.5.w <- residuals(sar_nb1.5.w)
```

Now estimate the spatial autocorrelation of the SAR model.

```
autocor_sar_nb1.5.w <- correlog(fdata$x, fdata$y, z = residuals(sar_nb1.5.w),
                                na.rm = T, increment = 1, resamp = 1)
```

Plot the autocorrelogram under the SAR model.

```
{plot(autocor_sar_nb1.5.w$correlation[1:50], type = "b", pch = 4, cex = 1.2, lwd = 1.5,
     ylim = c(-0.5, 1), xlab = "distance", ylab = "Moran's I", cex.lab = 1.5,
     cex.axis = 1.2)
abline(h = 0)
title(main = "Correlogram SARerr", cex = 1.5)}
```

Ohhh, where is the autocorrelation in the residuals? Now compare the two autocorrelograms.

```
{par(mfrow = c(2, 1))
plot(autocor_ols_res$correlation[1:50], type = "b", pch = 1, cex = 1.2, lwd = 1.5,
     ylim = c(-0.5, 1), xlab = "distance", ylab = "Moran's I", cex.lab = 1.5,
     cex.axis = 1.2)
abline(h = 0)
title(main = "OLS residuals", cex = 1.5)

plot(autocor_sar_nb1.5.w$correlation[1:50], type = "b", pch = 4, cex = 1.2, lwd = 1.5,
     ylim = c(-0.5, 1), xlab = "distance", ylab = "Moran's I", cex.lab = 1.5,
     cex.axis = 1.2)
abline(h = 0)
title(main = "Correlogram SARerr", cex = 1.5)}
```

Ok, now we know that the SAR model can solve the problem in the spatial autocorrelation in the residuals, let's try to make some inferences.

```
summary(sar_nb1.5.w)
```

```
summary(fols)
```

Looking to the summary of the SAR and OLS models, explain the diferences in the coefficients between both models.

Now let's compare the prediction of both models. To calculate a R2 to the SAR model, we will use the function **SARr2()** from GitHub.

```
source("https://raw.githubusercontent.com/jesusNPL/BetaDivNA/master/SARr2.R")
```

```
SARr2(Lfull = sar_nb1.5.w$LL, Lnull = sar_nb1.5.w$logLik_lm.model, N = nrow(fdata))
```

Comparing the two models (OLS and SAR), please answer the following questions:

*1. Which model have the best explanation?*

*2. What can we conclude from these results?*

# References

1. Hawkins, B.A., McCain, Ch.M., Davies, T.J., Buckley, L.B., Anacker, B.L., Cornell, H.V., Damschen, E.I., Grytness, J.A., Harrison, S., Holt, R.D., Kraft, N.J.B. & Stephens, P.R. (2012) Different evolutionary histories underlie congruent species richness gradients on birds and mammals. Journal of Biogeography, 39, 825–841.

2. Kerr, J.T. & Currie, D.J. (1999) The relative importance of evolutionary and environmental controls on broad-scale patterns of species richness in North America. Ecoscience, 6, 329–337.