

# Modeling northeastern Brazil forest enclaves

Rilquer Mascarenhas

5/5/2020

Loading occurrence data:

```
library(raster)
occs.xy <- read.table('data/Input_5_mapa.txt')
extent.calib <- extent(-45.872163, -33.379374, -18.717883, -2.318956)
extent.nsf <- extent(-41.87475, -33.42819, -10.474108, -2.475827)
extent.ssf <- extent(-42.60923, -36.1213, -16.00256, -9.392212)
colnames(occs.xy) <- c('longitude', 'latitude')
sp::coordinates(occs.xy) <- ~ longitude + latitude
occs.xy <- crop(occs.xy, extent.calib)
```

Reading environmental data, BR shapefiles and setting other geographical boundaries:

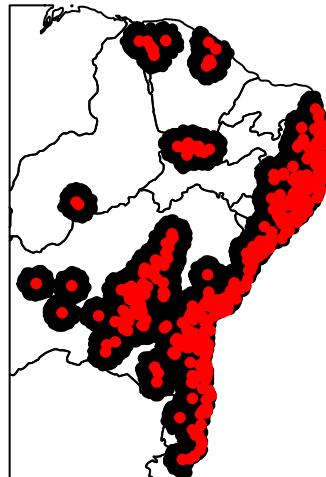
```
library(rgdal)
br <- crop(readOGR(dsn = 'data/layers/br/', layer = 'br'), extent.calib)
# Getting environmental data
env.calib <- crop(stack(list.files(path = 'data/bioclim/01_cur_chelsa_V1_2B_r2_5m/', pattern = ".tif$",
env.nsf <- crop(stack(list.files(path = 'data/bioclim/01_cur_chelsa_V1_2B_r2_5m/', pattern = ".tif$",
env.ssf <- crop(stack(list.files(path = 'data/bioclim/01_cur_chelsa_V1_2B_r2_5m/', pattern = ".tif$",
#
```

Getting background points through buffer:

```
# Creating empty lists
bgExt <- rgeos::gBuffer(occs.xy, width = 0.5)

# mask the environmental rasters by the background extent shape
envsBgMsk <- raster::mask(env.calib, bgExt)
bg.xy <- dismo::randomPoints(envsBgMsk, 5000)
# convert matrix output to data frame
bg.xy <- as.data.frame(bg.xy)
plot(br)
points(bg.xy, col = 'black', pch = 20)
points(occs.xy, col = 'red', pch = 20)
title('Enclaves - Presence and Background')
```

## Enclaves – Presence and Background



```
dev.copy(tiff,filename=paste0('output/Enclaves - Presence and Background.tif'),  
        width = 6, height = 5, units = "in", res = 500)  
dev.off()
```

First, use the following commands to avoid Java errors:

```
library(rJava)  
options(java.parameters = "-Xmx1g")
```

Then load *dismo* and other necessary packages:

```
library(rgdal)  
library(spocc)  
library(dismo)  
library(ENMeval)  
library(dplyr)  
library(maps)
```

Running ENMeval to evaluate different Maxent models:

```
# Creating optimize function  
optimize <- function(res) {  
  ##### Remove any candidate model which has an AUC less than 0.51= models with no discrimination  
  opt.auc <- res[res[,4] >= 0.5,]
```

```

####Remove any candidates which have no parameters
no.param <- opt.auc[opt.auc[,13] > 1,]
####Remove any candidates where the AIC score was NA (too many parameters)
noAICNA<- no.param[which(!is.na(no.param$AICc)),]
####Remove any models which have an OR of zero
noORO <- noAICNA[noAICNA[,9] != 0,]
####Order the remaining list by lowest OR then highest AUC, sequentially
ordered<-noORO[with(noORO, order(avg.test.or10pct, -avg.test.AUC)), ]
####Grab the settings of that first model (the optimal model)
ordered[1,]

}

ev<-ENMevaluate(occ = occs.xy, env = env.calib, bg.coords = bg.xy,
                  RMvalues = seq(1,5,0.5), fc = c("L","LQ","H","LQH"),
                  method = "checkerboard2", n.bg = 5000, rasterPreds = T, parallel = T,
                  numCores = 8, algorithm = 'maxent.jar')
best<-optimize(ev@results)

```

Running Maxent models:

```

##Calibrating model
dir.create('output/maxent_models/')
mod <- maxent(
  x=env.calib, # bio stack
  p=occs.xy, # locality csv
  factors = NULL,
  path = ('output/maxent_models/enclaves'),
  args=c('betamultiplier=1.5',
         'linear=true',
         'quadratic=true',
         'hinge=true',
         'product=false',
         'threshold=true',
         'threads=8',
         'responsecurves=true',
         'jackknife=true',
         'askoverwrite=false',
         'autofeature=false')
)

```

Now projecting into different extents in the present

```

dir.create('output/projections/')
proj.nsf <- predict(
  object = mod,
  x = env.nsf,
  filename = 'output/projections/enclaves_nsf.asc',
  na.rm = T,
  format = 'ascii',#or GTiff
  overwrite = T,
  args = "logistic"
)
proj.ssf <- predict(

```

```

object = mod,
x = env.ssf,
filename = 'output/projections/enclaves_ssf.asc',
na.rm = T,
format = 'ascii',#or GTiff
overwrite = F,
args = "logistic"
)

```

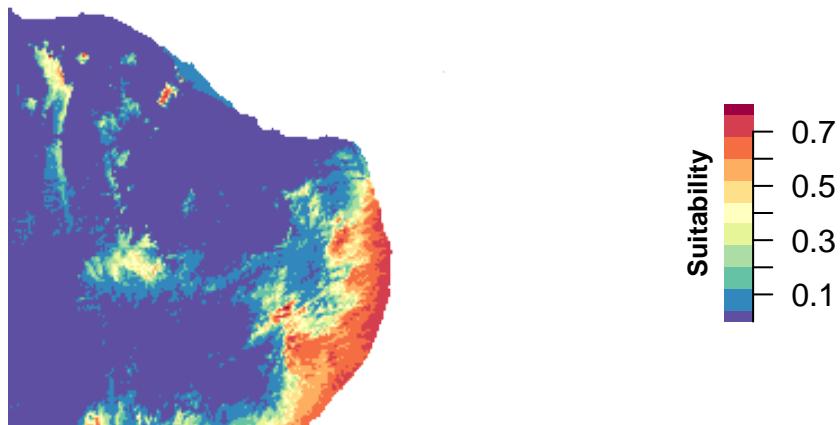
Reading output asc files and plotting results:

```

mxt.nsf <- raster('output/projections/enclaves_nsf.asc')
mxt.ssf <- raster('output/projections/enclaves_ssf.asc')
dir.create('output/maps/')
library(RColorBrewer)
par(bty='n', oma = c(0, 0, 2, 0))
r.range <- c(min(na.omit(values(mxt.nsf))),max(na.omit(values(mxt.nsf))))
plot(mxt.nsf, col = rev(brewer.pal(11, 'Spectral')), axes=FALSE,legend.width=1,
      axis.args=list(at=round(seq(r.range[1], r.range[2], 0.1),2),
                     labels=round(seq(r.range[1], r.range[2], 0.1),2)),
      legend.args=list(text='Suitability', side=4, font=2, line=-2, cex=0.8))
title('Enclaves - North of São Francisco river', outer = TRUE, line = 0.2)

```

## Enclaves – North of São Francisco river



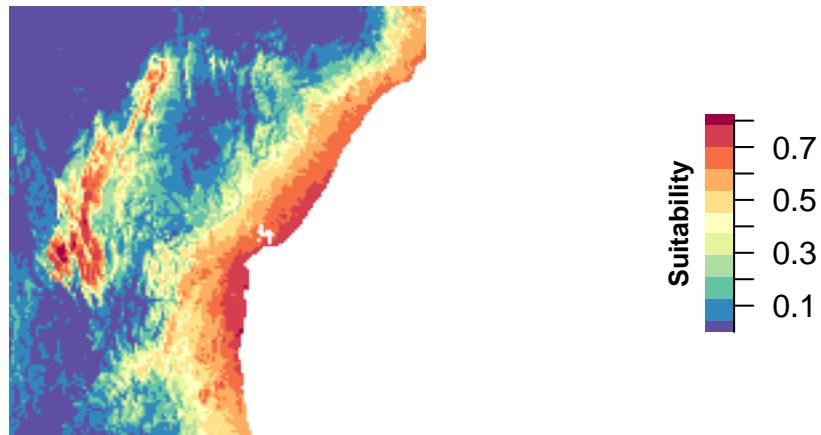
```

dev.copy(tiff,filename='output/maps/enclaves_nsf.tif',
         width = 10, height = 6, units = "in", res = 500)
dev.off()

library(RColorBrewer)
par(bty='n', oma = c(0, 0, 2, 0))
r.range <- c(min(na.omit(values(mxt.ssf))),max(na.omit(values(mxt.ssf))))
plot(mxt.ssf, col = rev(brewer.pal(11, 'Spectral')), axes=FALSE,legend.width=1,
      axis.args=list(at=round(seq(r.range[1], r.range[2], 0.1),2),
                     labels=round(seq(r.range[1], r.range[2], 0.1),2)),
      legend.args=list(text='Suitability', side=4, font=2, line=-2, cex=0.8))
title('Enclaves - South of São Francisco river', outer = TRUE, line = 0.2)

```

## Enclaves – South of São Francisco river



```

dev.copy(tiff,filename='output/maps/enclaves_ssf.tif',
         width = 10, height = 6, units = "in", res = 500)
dev.off()

```

Transforming into binary maps:

```

ts <- as.numeric(mod@results[which(row.names(mod@results)=="X10.percentile.training.presence.Cloglog.th")]
mxt.nsf.bin <- mxt.nsf
mxt.nsf.bin[which(mxt.nsf.bin[] < ts)] <- 0
mxt.nsf.bin[which(mxt.nsf.bin[] > ts)] <- 1
mxt.ssf.bin <- mxt.ssf

```

```
mxt.ssf.bin[which(mxt.ssf.bin[] < ts)] <- 0
mxt.ssf.bin[which(mxt.ssf.bin[] > ts)] <- 1
```

Plotting binary maps:

```
par(bty='n', oma = c(0, 0, 2, 0))
plot(mxt.nsf.bin, col = c('grey','black'), axes=FALSE, legend.width=1,
      axis.args=list(at=c(0,1)),
      legend.args=list(text='Absence    Presence', side=4, font=3, line=-2, cex=0.8))
title('Enclaves - North of São Francisco river (binary map)', outer = TRUE, line = 0.2)
```

## Enclaves – North of São Francisco river (binary map)



```
dev.copy(tiff,filename='output/maps/enclaves_nsf_bin.tif',
         width = 10, height = 6, units = "in", res = 500)
dev.off()

library(RColorBrewer)
par(bty='n', oma = c(0, 0, 2, 0))
plot(mxt.ssf, col = c('grey','black'), axes=FALSE, legend.width=1,
      axis.args=list(at=c(0,1)),
      legend.args=list(text='Absence    Presence', side=4, font=3, line=-2, cex=0.8))
title('Enclaves - South of São Francisco river (binary map)', outer = TRUE, line = 0.2)
```

## Enclaves – South of São Francisco river (binary map)



```
dev.copy(tiff,filename='output/maps/enclaves_ssf_bin.tif',
         width = 10, height = 6, units = "in", res = 500)
dev.off()
```