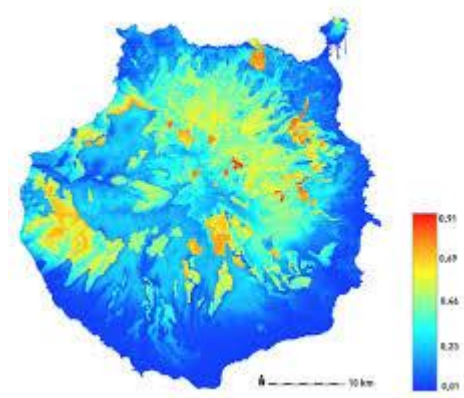


Modelando el nicho Ecológico



Biol. Maribel Arenas Navarro
Dr. Oswaldo Téllez Valdés

¿Qué vamos a modelar?

Distribución potencial

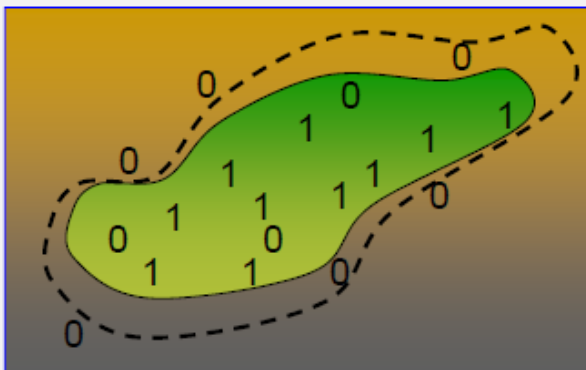
Distribución Realizada

Potential Distribution



Equilibrium with the
environmental conditions

Background absences



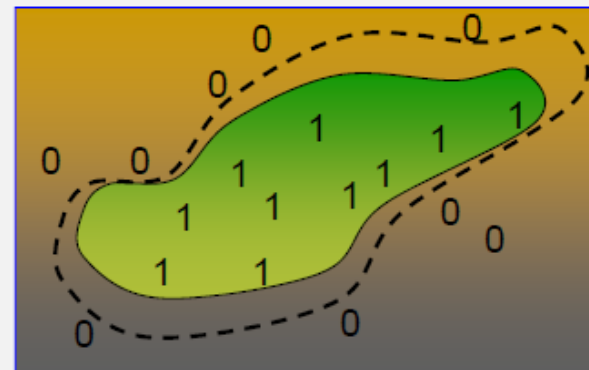
llll

Realized Distribution



Role of dispersal limitations,
biotic interactions and other
contingent factors

Absences selected outside
environmentally favourable regions.



near to PD

Jorge M.
Lobo

¿Qué necesitamos?

- Base de Datos

¿Donde?

CONABIO, GBIF, TROPICOS, etc. **CAMPO!!!**

Artículos de Datos

Minería de datos (Exploración y análisis de los datos).

Depuración de los datos (sinonimia, puntos NA, consultar a un especialista, bibliografía, etc.

<http://www.gbif.es/Recursos.php>



	A	B	C
1	Especie	decimalLat	decimalLon
2	Achatocarpus gracilis	24.9958333	-107.804167
3	Achatocarpus gracilis	22.0253611	-105.613778

.CSV

Especie				
	A	B	C	D
1	Especie	decimalLat	decimalLon	
2	Achatocarpus gracilis	24.9958333	-107.804167	
3	Achatocarpus gracilis	22.0253611	-105.613778	
4	Achatocarpus gracilis	19.9783333	-105.463333	
5	Achatocarpus gracilis	19.93385	-105.254642	
6	Achatocarpus gracilis	19.94	-105.246667	
7	Achatocarpus gracilis	19.53725	-105.076833	
8	Achatocarpus gracilis	19.543	-105.07225	
9	Achatocarpus gracilis	19.5083333	-105.058333	
10	Achatocarpus gracilis	19.505	-105.050278	
11	Achatocarpus gracilis	19.5	-105.05	

- Datos Ambientales

Clima

WORLDCLIM

CLIMOND

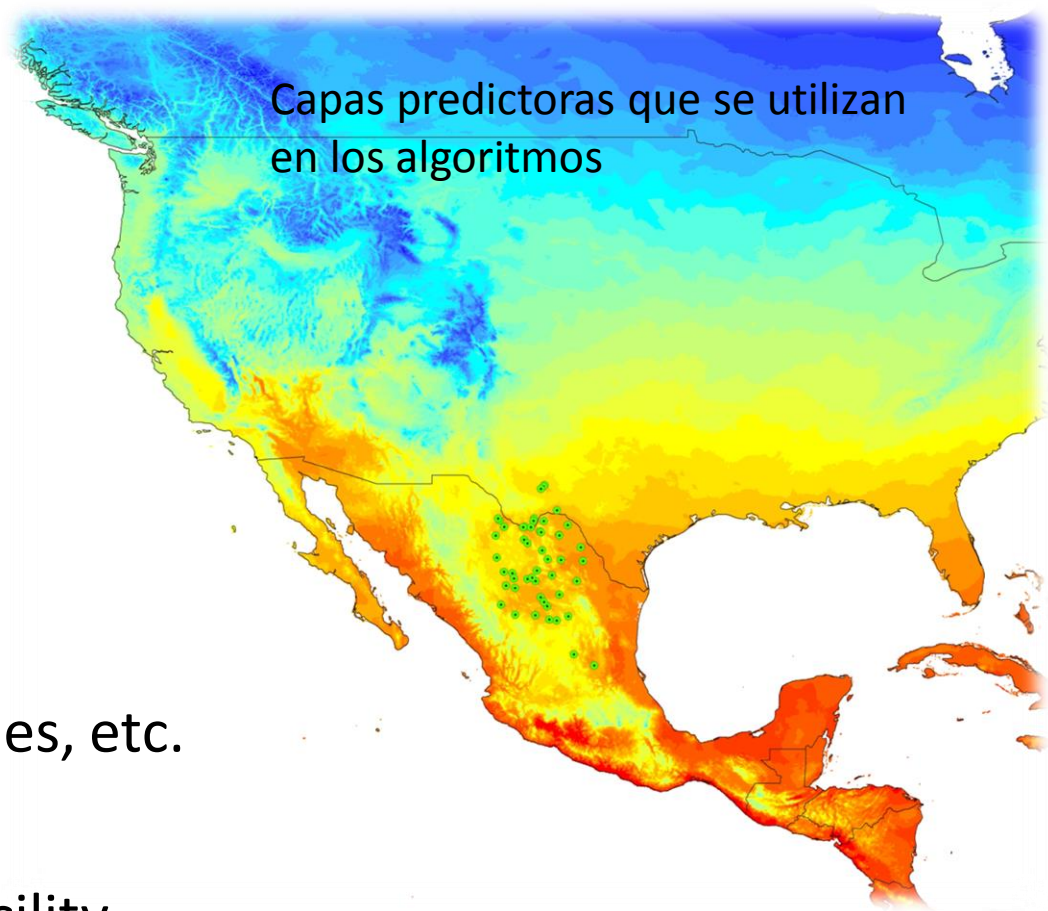
CRU

Bioclimas Neotropicales, etc.

Edáficas

Global Land-Cover Facility

IGBP-DIS

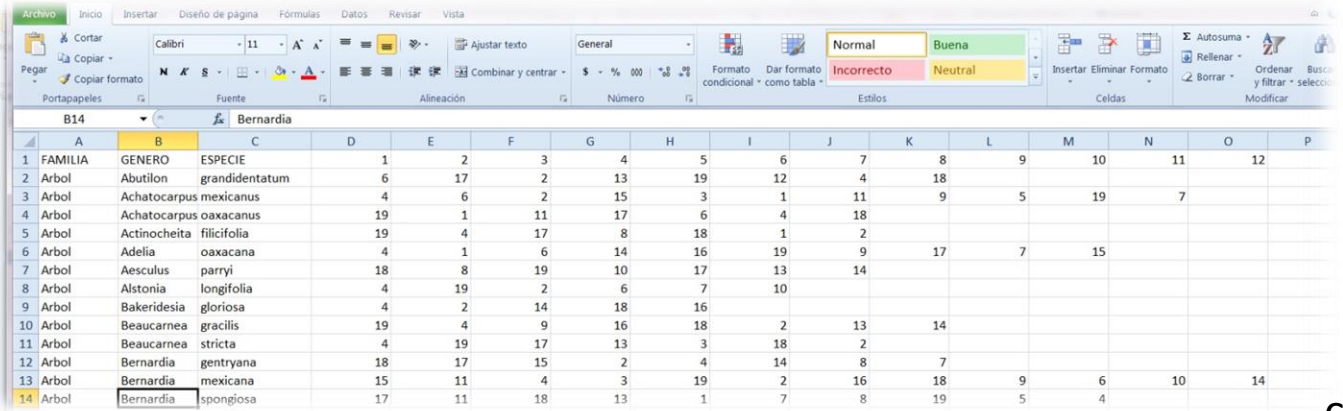


¿Qué variables usamos para nuestra especie?

- Todas?
- Cuántas?
- Cuáles?
- Por qué?

Algunas herramientas

- PCA
- Jackknife
- seleccionar las variables crudas mas importantes biológicamente desechando las mas correlacionadas (Jiménez-Valverde et al. 2009).



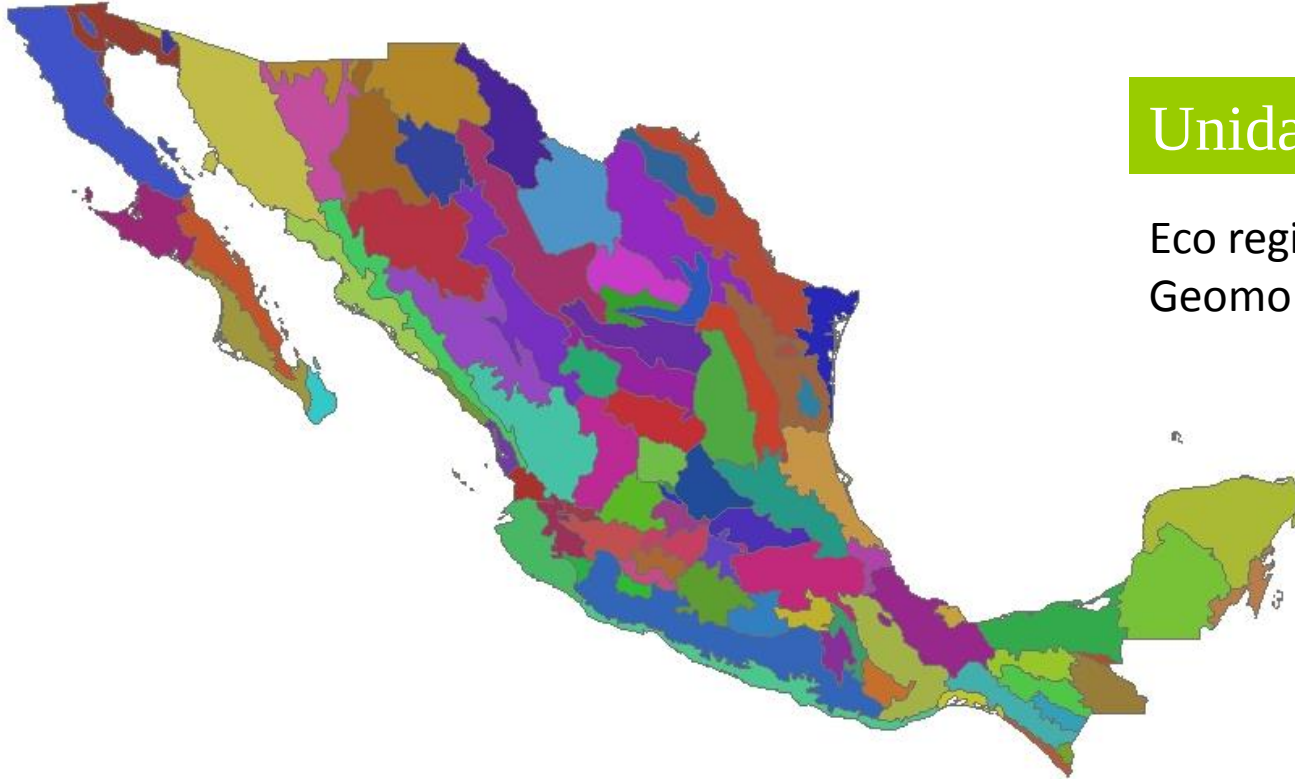
The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	FAMILIA	GENERO	ESPECIE	1	2	3	4	5	6	7	8	9	10	11	12	
2	Arbol	Abutilon	grandidentatum	6	17	2	13	19	12	4	18					
3	Arbol	Achatocarpus	mexicanus	4	6	2	15	3	1	11	9	5	19	7		
4	Arbol	Achatocarpus	oaxacanus	19	1	11	17	6	4	18						
5	Arbol	Actinocheita	filicifolia	19	4	17	8	18	1	2						
6	Arbol	Adelia	oaxacana	4	1	6	14	16	19	9	17	7	15			
7	Arbol	Aesculus	parryi	18	8	19	10	17	13	14						
8	Arbol	Alstonia	longifolia	4	19	2	6	7	10							
9	Arbol	Bakeridesia	gloriosa	4	2	14	18	16								
10	Arbol	Beaucarnea	gracilis	19	4	9	16	18	2	13	14					
11	Arbol	Beaucarnea	stricta	4	19	17	13	3	18	2						
12	Arbol	Bernardia	gentryana	18	17	15	2	4	14	8	7					
13	Arbol	Bernardia	mexicana	15	11	4	3	19	2	16	18	9	6	10	14	
14	Arbol	Bernardia	spongiosa	17	11	18	13	1	7	8	19	5	4			

M

Unidades naturales

Eco regiones
Geomorfológica, etc.



Provincias fisiográficas de CONABIO

Cervantes-Zamora, Y., Cornejo-Olgín, S. L., Lucero-Márquez, R., Espinoza-Rodríguez, J. M., Miranda-Viquez, E. y Pineda-Velázquez, A, (1990). 'Provincias Fisiográficas de México'. Extraído de Clasificación de Regiones Naturales de México II, IV.10.2. Atlas Nacional de México. Vol. II. Escala 1:4000000. Instituto de Geografía, UNAM. México.

Chapter 11

Machine learning methods

11.1 Maxent

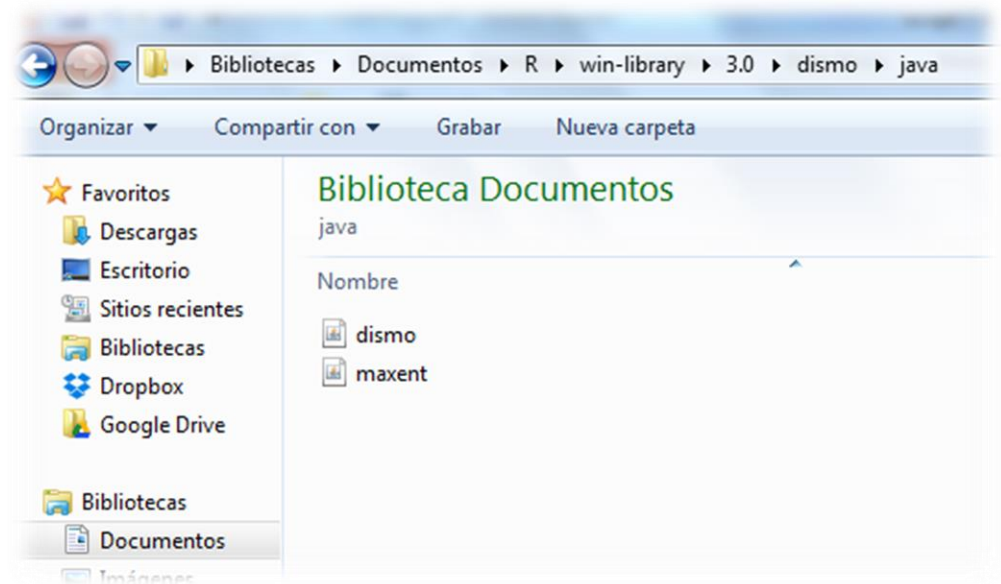
Species distribution modeling with R

Robert J. Hijmans and Jane Elith

July 8, 2013

```
install.packages("dismo")
```

```
library("dismo")
```



A modelar!!!!!!!



- `#Modelando`
- `library("raster")`
- `library("sp")`
- `library("rgdal")`
- `library("dismo")`
- `options(java.parameters = "-Xmx1g" )`
- `Sys.setenv(NOAWT=TRUE)`
- `library("rJava")`
- `Sys.setenv(NOAWT=FALSE)`
- `# Donde estan los archivos`
- `# ruta que se debe cambiar, dependiendo de cada maquina`
- `rootPath <- "/modelos_conabio/"`

modelando arboles

```
gridPath <- paste(rootPath, "asciamb/", sep = "") #Donde estan las capas ambientales
shapePath <- paste(rootPath, "fisiografico/", sep = "") #Donde esta la regionalización que se usa
para construir M
```

```
outGridPath <- paste(rootPath, "salida_ar/grids/", sep = "")
dir.create(outGridPath, recursive = TRUE) # Si ya existe el directorio manda la alerta y sigue sin
problema
```

```
outPerfilesPath <- paste(rootPath, "salida_ar/perfiles/", sep = "")
dir.create(outPerfilesPath, recursive = TRUE)
especiesPath <- rootPath
especiesFile <- "arboles2.csv" # archivo de coordenadas(especie, lat, long )
shapeFile <- "rfisio4mg.shp" #cambiar según la regionalización
layerShape <- "rfisio4mg" # cambiar según la regionalización
variablesEspecies <- "jack_arp2.csv" # variables con las que se modelara la especie
```

```
pathModelos <- paste(rootPath, "salida_ar/modelos", sep = "")
```

#####

Windows 7 (C:) ▶ modelos_conabio ▶

en biblioteca ▼ Compartir con ▼ Grabar Nueva carpeta

Nombre	Fecha de modifica...	Tipo	Tamaño
asciamb	19/11/2014 12:05 a...	Carpeta de archivos	
fisiografico	19/11/2014 12:05 a...	Carpeta de archivos	
salida_ar	19/11/2014 12:06 a...	Carpeta de archivos	
arboles_BD	01/11/2013 11:12 a...	Microsoft Excel C...	320 KB
modelando	16/02/2014 12:08 a...	Archivo R	18 KB
rfisio4mg.shp	29/10/2011 10:41 a...	Archivo SHP	315 KB
script	15/02/2014 10:43 ...	Documento de tex...	18 KB
variables_arboles	01/11/2013 12:25 ...	Microsoft Excel C...	4 KB

BD.csv script

gridPath

shapePath

outGridPath

especiesFile

Shapefile (regionalización)

Variables por especie.csv

Windows 7 (C:) ▶ modelos_conabio ▶ salida_ar ▶

en biblioteca ▼ Compartir con ▼ Grabar Nueva carpeta

Nombre	Fecha de modifica...	Tipo	Tamaño
grids	19/11/2014 12:19 a...	Carpeta de archivos	
modelos	19/11/2014 12:20 a...	Carpeta de archivos	
perfiles	19/11/2014 12:20 a...	Carpeta de archivos	

```
#####  
#variables de control, dependiendo si se quiere  
salvar o no  
salvarGrids <- 1 # <= 0 = no, > 0 si  
visualizarGrids <- 1 # <= 0 = no, > 0 si  
salvarPerfil <- 1 # <= 0 = no, > 0 si  
ejecutarMaxEnt <- 1 # <= 0 = no, > 0 si  
salvarNAs <- 1 # <= 0 = no, > 0 si  
salvarRegiones <- 1 # <= 0 = no, > 0 si  
#####
```

Cargando las variables climáticas

#Variables climáticas por especie

```
variablesClimaticas <- read.table(paste(especiesPath, variablesEspecies, sep = ""),  
                                header=TRUE , sep=",",fileEncoding = "latin1")
```

#Hago que especie sea el binomio no el epiteto específico como viene en el archivo

```
variablesClimaticas["ESPECIE"] <- paste(variablesClimaticas[, "GENERO"],  
variablesClimaticas[, "ESPECIE"], sep = " ")
```

#de acuerdo al nombre de las capas (en este caso en letra) y el archivo de las variables climáticas por especie esta en número hacemos una matriz :

```
variablesLetra <- matrix(  
  c(1, "uno", 2, "dos", 3, "tres", 4, "cuatro", 5, "cinco", 6, "seis", 7, "siete",  
    8, "ocho", 9, "nueve", 10, "diez", 11, "once", 12, "doce", 13, "trece", 14, "catorce",  
    15, "quince", 16, "dseis", 17, "dsiete", 18, "docho", 19, "dnueve"),  
  nrow = 19, ncol = 2, byrow = TRUE)
```

```
rasterFiles <- list.files(path=gridPath, pattern = '.*\\.asc$')
```

```
climVars <- gsub("(^.*)\\.asc$", "\\1", rasterFiles, ignore.case = TRUE)
```

```
grids <- sapply(rasterFiles, function(x) {  
  fileVarN <- paste(gridPath, x, sep = "  
  rasterstoimport <- toString(paste('raster(', fileVarN, ')', sep = ""))  
  rasters <- eval(parse(text = rasterstoimport))  
})
```

```
names(grids) <- climVars  
varClimaticaGrids <- stack(grids)  
rm(grids) #Para ahorrar memoria
```

#cargamos el raster

#creamos una base de raster que sirve para rasterizar los poligonos seleccionados

```
r <- raster(ncol=varClimaticaGrids@ncols, nrow=varClimaticaGrids@nrows)  
extent(r) <- extent(varClimaticaGrids)
```

Regionalización y especies

#cargamos el shapefile de la regionalización a utilizar

```
dsn = paste(shapePath, shapeFile, sep = "")
```

```
regionalizacion <- readOGR(dsn = dsn, layer = layerShape)
```

#-----(deben aparecer las características del shapefile)

#cargamos las especies

```
especies <- read.table(paste(especiesPath, especiesFile, sep = ""),  
                      header=TRUE , sep="," ,fileEncoding = "latin1")
```

#que los datos de especies sean de tipo espacial en R

```
coordinates(especies) <- ~decimalLon+decimalLat
```

#La proyección geográfica, la copiamos de las regiones

```
proj4string(especies) <- proj4string(regionalizacion)
```

Loop!!! A modelar (se debe correr de corrido)

#Aquí sucede Todo

```
procesar <- function(x) {
```

```
  print(paste("Procesando:", x, sep = " "))
```

```
  #Quitar los espacios en blanco al nombre de la especie
```

```
  xSinBlancos <- gsub("\\s", "", x)
```

```
  sp = especies[especies$Especie == x,]
```

```
  ###Creando directorios por especie
```

```
  pathModeloEspecie <- paste(pathModelos, xSinBlancos, sep = "")
```

```
  dir.create(pathModeloEspecie)
```

```
  ####
```

#obtener los datos de los polígonos que tienen sitios de colecta y los NA (que caen fuera de la regionalización)

```
valorPoligonoReg <- over(sp, as(regionalizacion, "SpatialPolygons"))
tienesNAs <- length(valorPoligonoReg[is.na(valorPoligonoReg)])
print(paste(" Tengo",tienesNAs,"NAs"))
```

#Salvar en archivo delimitado por comas los registros NA que caen fuera de la regionalización

```
if (salvarNAs > 0 && tienesNAs > 0) {
  NAs <- cbind(sp@coords[c(which(is.na(valorPoligonoReg))),], x)
  filenameNA = paste(pathModelos, xSinBlancos, "/", xSinBlancos,
"_NAs.csv", sep = "")
  write.csv(NAs, filenameNA, na="")
}
else {
  print(" No se salvan los registros con NAs")
}
```

```
valorPoligonoReg <- unique(valorPoligonoReg)
regionRecortada <- regionalizacion[valorPoligonoReg[!is.na(valorPoligonoReg)],]
```

#Regiones donde los puntos de colecta caen y crea un .csv

```
if(salvarRegiones > 0) {
  filenameRegiones = paste(pathModelos, xSinBlancos, "/", xSinBlancos,
"_Regiones.csv", sep = "")
  write.csv(regionRecortada@data, filenameRegiones, na="",)
}
else {
  print(" No se salvan los registros con NAs")
}
```

#rasterizamos los polígonos con sitios de colecta, asignandole valor de 1

```
regionRecortadaGrid <- rasterize(regionRecortada, r, 1)
```

#Recortamos las capas, solo de las variables que son importantes para la especie, quitamos los campos de FAMILIA, GENERO y ESPECIE

```
varClimEspecie <- variablesClimaticas[variablesClimaticas["ESPECIE"] == x,
!names(variablesClimaticas) %in% c("FAMILIA","GENERO",
"ESPECIE")]
```

#Le quito los NA's

```
varClimEspecie <- varClimEspecie[!is.na(varClimEspecie)]
```

#vector de variables climáticas en letra

```
varClimEspecieLetra <- variablesLetra[varClimEspecie,][,2]
```

varClimaticaEspecieGrids será el raster stack de las variables que importan para la especie

```
varClimaticaEspecieGrids <- sapply(varClimEspecieLetra, function(x) {  
  rasters <- raster(varClimaticaGrids, layer = x)  
})
```

```
varClimaticaEspecieGrids <- stack(varClimaticaEspecieGrids)
```

```
recorteGrid <- regionRecortadaGrid * varClimaticaEspecieGrids
```

```
names(recorteGrid) <- names(varClimaticaEspecieGrids)
```

```
recorteGrid <- crop(recorteGrid, extent(regionRecortada))
```

```

if (salvarGrids > 0) {
  #salvar los rasters recortados, con nombre grid_numeroenletra_especie.asc
  print(" Salvando los grids recortados (M)")
  climVarsSpecies <- names(varClimaticaEspecieGrids)
  sapply(climVarsSpecies, function(climVar) {
    outRasterFile = paste("grid_", climVar, "_", xSinBlancos, sep = "")
    gridtmp <- paste("recorteGrid", "$", climVar, sep = "")
    dir.create(paste(outGridPath, xSinBlancos, sep = ""), recursive = TRUE)
    print(paste(" Salvando: ", outRasterFile))
    writeRaster(eval(parse(text = gridtmp)), paste(outGridPath, xSinBlancos, "/",
outRasterFile, sep = ""),
      format = "ascii", overwrite=TRUE)
    if (visualizarGrids > 0) {
      #Para ver los resultados
      plot(eval(parse(text = gridtmp)), main = paste(x, climVar, sep = ": "))
    }
  })
}
else {
  print(" No se salvan los grids recortados")
}

```

```
if (salvarPerfil > 0) {  
  #Salvar perfil climático de la especie  
  print(" Salvando los perfiles ")  
  perfilEspecie <- extract(recorteGrid, sp, method='bilinear')  
  perfilEspecie <- cbind(sp@coords, sp@data, perfilEspecie)  
  filename = paste(outPerfilesPath, "perfil_", xSinBlancos, ".csv",  
sep = "")  
  write.csv(perfilEspecie, filename, na="")  
}  
else {  
  print(" No se salvan los perfiles")  
}
```

```
if (ejecutarMaxEnt > 0) {  
  #INICIA maxent  
  pathModeloEspecie <- paste(pathModelos, xSinBlancos, sep = "")  
  print(" Ejecutando MaxEnt")  
  print(paste(" Resultados del modelo en: ", pathModeloEspecie, sep = ""))  
  dir.create(pathModeloEspecie)  
  datos <- sp@coords  
  fold <- kfold(datos, k=5)  
  datos.test <- datos[fold == 1, ]  
  datos.train <- datos[fold != 1, ]  
  me <- maxent(recorteGrid, datos.train, removeDuplicates = TRUE, path =  
pathModeloEspecie, args=c("-J", "-P"))  
  #response(me)
```

###MaxEnt Output Format acepta raw, cumulative, logistic

```
  r <- predict(me, recorteGrid, args=c("outputformat=logistic"), progress='text',  
format = "ascii",  
          filename = paste(pathModeloEspecie, "/maxent_predict_", xSinBlancos,  
".asc", sep = ""))
```

#####testing ROC en archivo pdf

```
bg <- randomPoints(recorteGrid, 1000)
#simplest way to use evaluate
e1 <- evaluate(me, p = datos.test, a = bg, x = recorteGrid)
pdf(paste(pathModeloEspecie, "/ROC_", xSinBlancos, ".pdf", sep = ""))
plot(e1, 'ROC', main = x)
dev.off()
```

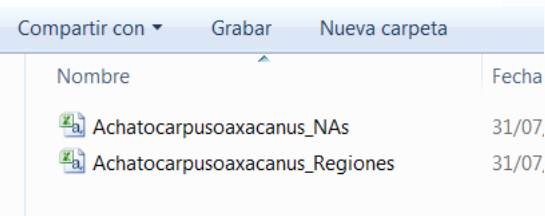
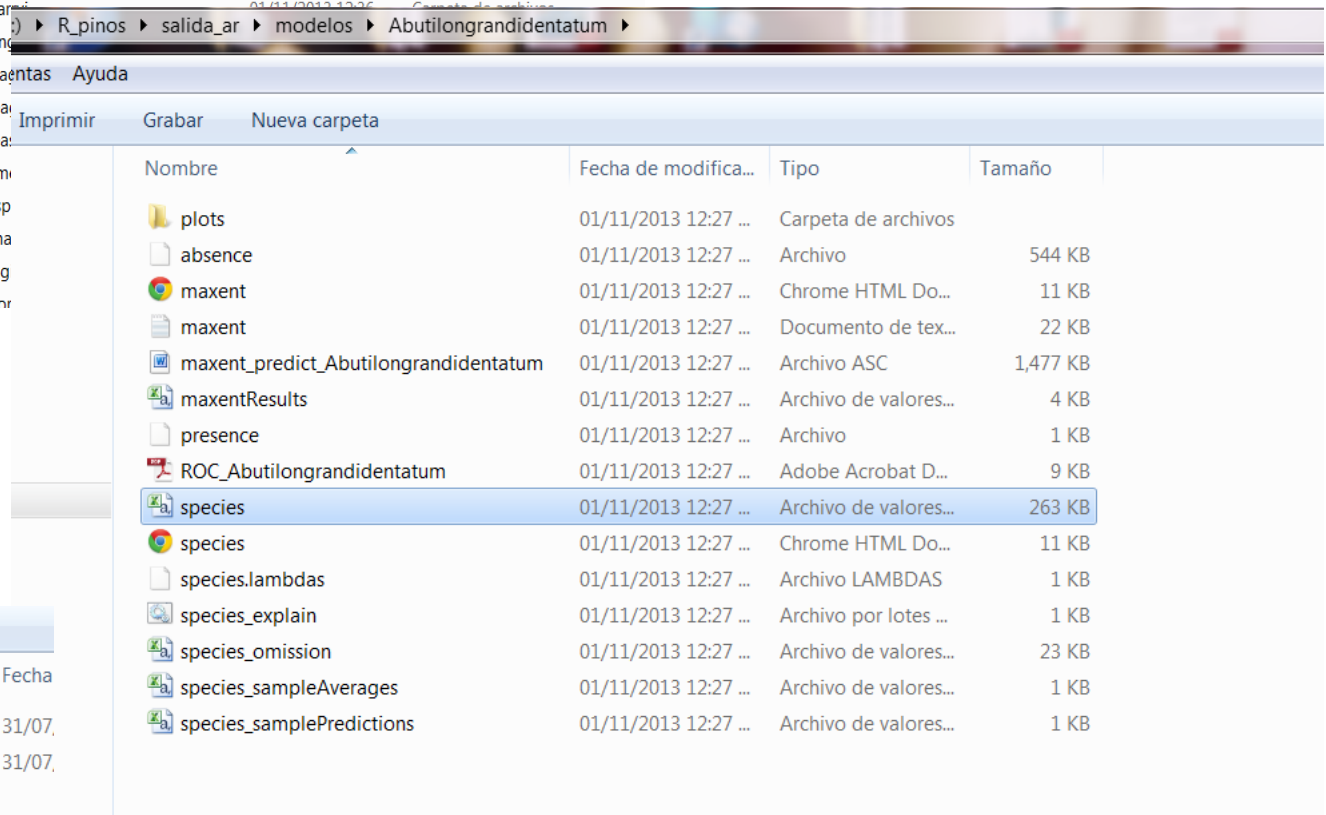
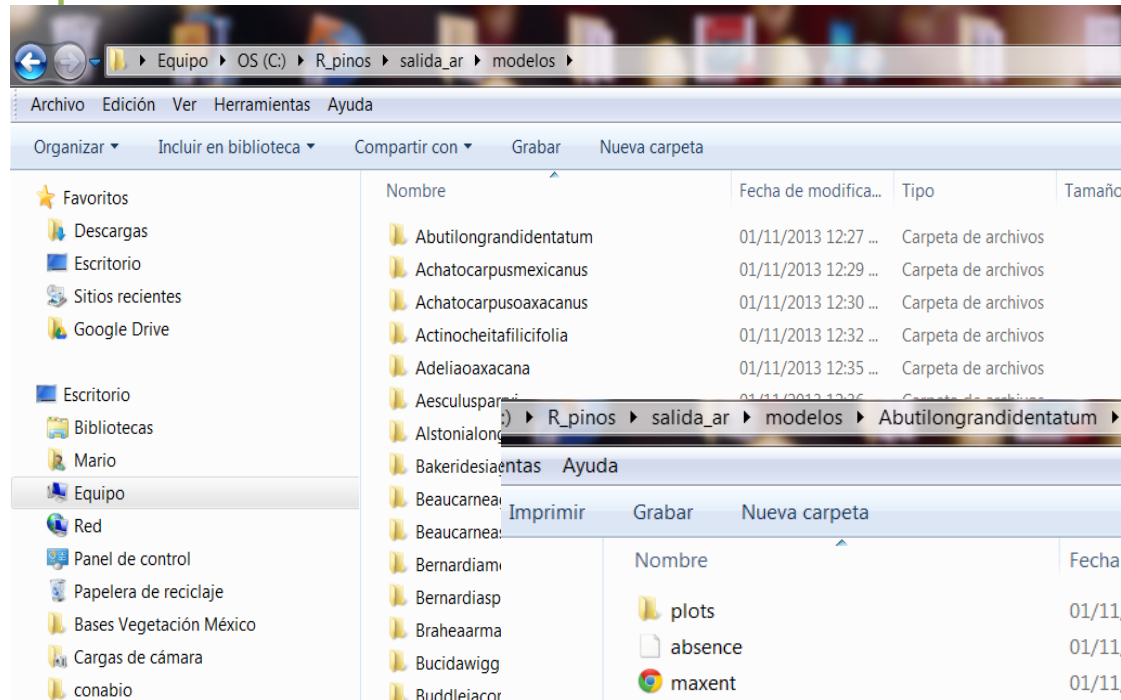
#Fin de maxent

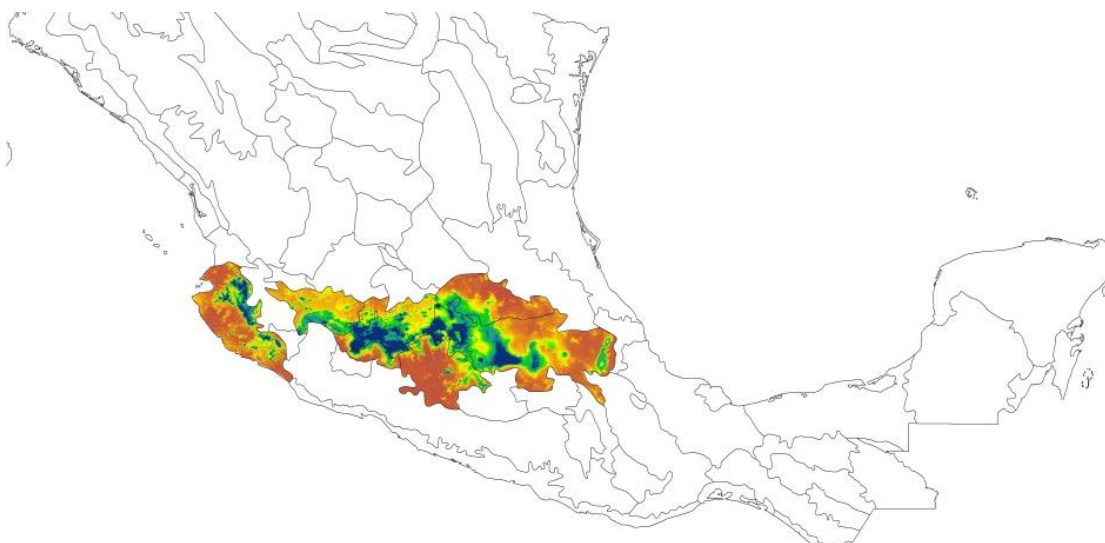
```
}
else {
  print(" No se ejecuta MaxEnt")
}
}
```

#Separamos las especies

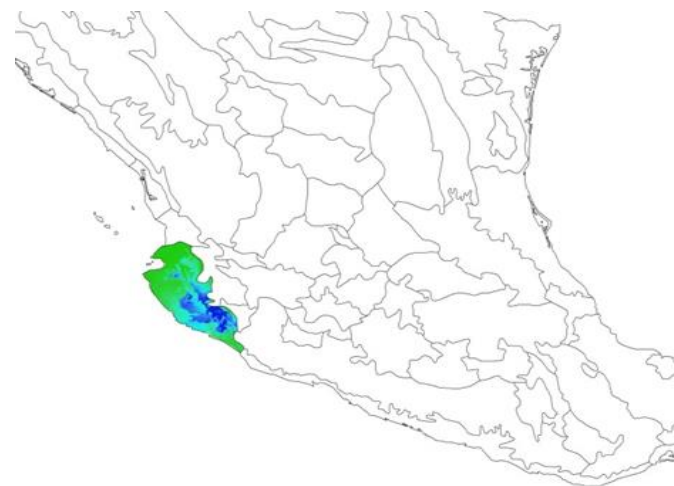
```
listaEspecies = as.character(unique(especies$Especie))
#Mandamos todo a la función procesar, x lleva el nombre de la especie, una a una
sapply(listaEspecies, function(x) procesar(x))
```

Obtenemos...

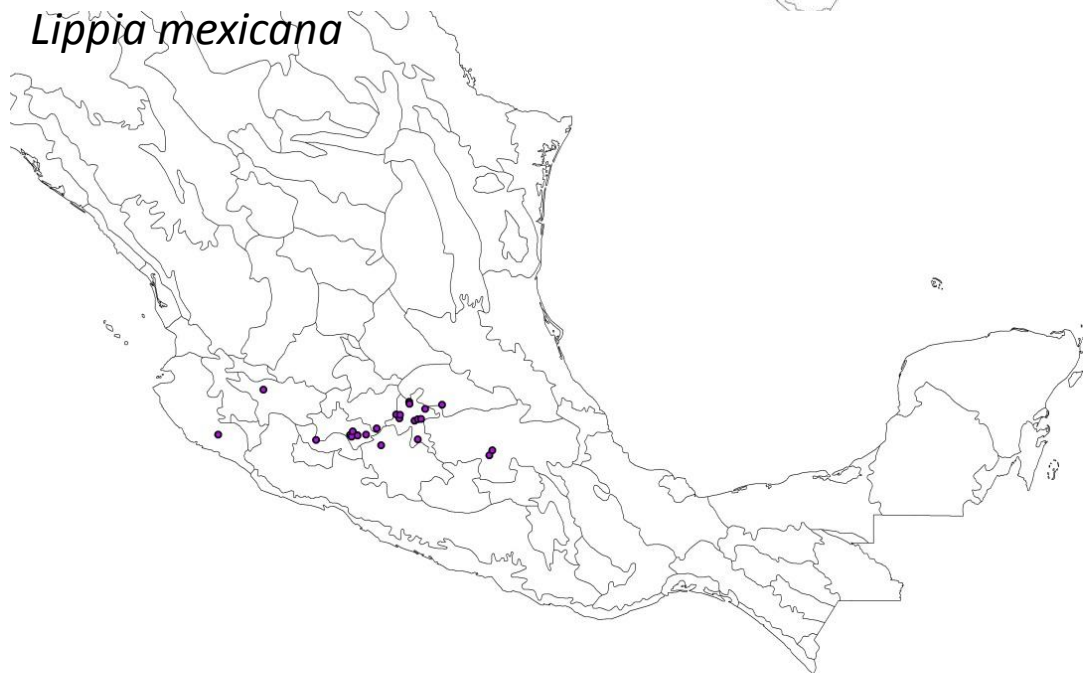




Lippia mexicana



39 registros



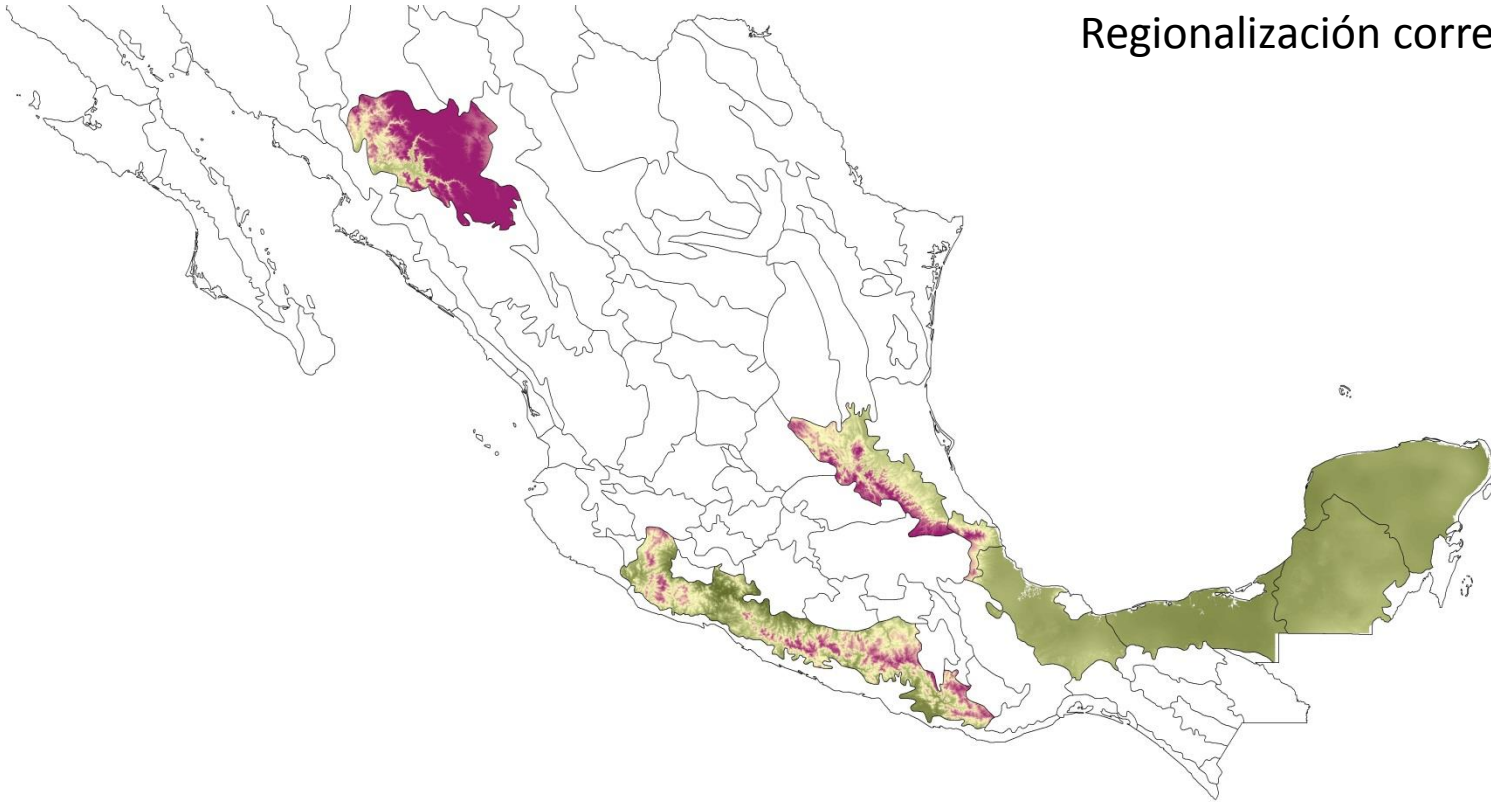
23 registros

Problemas??

Distribución natural? Disyunta?

Esfuerzo de colecta suficiente?

Regionalización correcta?



Muy importante



- Conocer la distribución de la especie que se va a modelar
- Que se va a modelar? Potencial o Realizada
- Seleccionar una M adecuada a lo que se va modelar
- Limpiar y dar mantenimiento a las bases de datos
- Compartir los datos “artículos de datos”
- Seguir leyendo, compartiendo y explorando



Gracias!!!