Combined impacts of climate and land use change and the future restructuring of Neotropical bat biodiversity

Fernando Gonçalves, Lilian P. Sales, Mauro Galetti, Mathias M. Pires

31/07/2021

Code by Lilian P. Sale

CLIMATE

Climond Bioclim data present and furute (only year 2030 is shown)

devtools::install\_github('babaknaimi/sdm') # to install the latest version of the sdm package from github  
library(sdm)  
library(spThin )  
library(dismo)  
library(rgbif)  
library(scrubr)  
library(spocc)  
library(rvertnet)  
library(plyr)  
library(scrubr)  
library(usdm)  
library(rgdal)  
library(sp)  
library(spatialEco)  
library(plyr)  
library(letsR)  
  
setwd("D:/Leddiv/Climate/Climond\_10'/CM10\_1975H\_Bio\_ASCII\_V1.2/CM10\_1975H\_Bio\_V1.2")  
pres <- list.files(pattern = ".txt")  
pres <- pres[1:19]  
pres <- stack(pres)  
pres@crs <- CRS("+proj=longlat +datum=WGS84")  
  
# 2030 -----------------------------------------------------------------------------------------  
  
setwd("D:/Leddiv/Climate/Climond\_10'/CM10\_2030\_A1B\_CS\_Bio\_ASCII\_V1.2/CM10\_2030\_A1B\_CS\_Bio\_V1.2")  
f30a <- list.files(pattern = ".txt")  
f30a <- stack(f30a[1:19])  
f30a@crs <- CRS("+proj=longlat +datum=WGS84")  
  
setwd("D:/Leddiv/Climate/Climond\_10'/CM10\_2030\_A1B\_MR\_Bio\_ASCII\_V1.2/CM10\_2030\_A1B\_MR\_Bio\_V1.2")  
f30b <- list.files(pattern = ".txt")  
f30b <- stack(f30b[1:19])  
f30b@crs <- CRS("+proj=longlat +datum=WGS84")  
  
setwd("D:/Leddiv/Climate/Climond\_10'/CM10\_2030\_A2\_CS\_Bio\_ASCII\_V1.2/CM10\_2030\_A2\_CS\_Bio\_V1.2")  
f30c <- list.files(pattern = ".txt")  
f30c <- stack(f30c[1:19])  
f30c@crs <- CRS("+proj=longlat +datum=WGS84")  
  
setwd("D:/Leddiv/Climate/Climond\_10'/CM10\_2030\_A2\_MR\_Bio\_ASCII\_V1.2/CM10\_2030\_A2\_MR\_Bio\_V1.2")  
f30d <- list.files(pattern = ".txt")  
f30d <- stack(f30d[1:19])  
f30d@crs <- CRS("+proj=longlat +datum=WGS84")  
  
  
# Check for name consistency  
names(f30a) <- names(pres)  
names(f30b) <- names(pres)  
names(f30c) <- names(pres)  
names(f30d) <- names(pres)

### SPECIES DATA

Select South American bats and environmental info from IUCN range maps

memory.limit() #Allow greater use of memory  
Mam <- spTransform(Mam, CRS("+proj=longlat +datum=WGS84"))  
  
#Pres/Abs matrix  
presab\_poligono <- lets.presab (Mam, xmn=-131,xmx=-35,ymn=-56, ymx=66,   
 resol=0.16, count=T, crs = CRS("+proj=longlat +datum=WGS84"),  
 cover=0.5, remove.cells=T)  
  
# Plotting the raster (a species richness map)  
plot(presab\_poligono[[2]])  
setwd("D:/Leddiv/lilian/Bats")  
save.image("PresAb.RData")  
  
#load("PresAb.RData")  
  
# Getting the pres/abs matrix   
matrizPA <- as.data.frame(presab\_poligono[[1]])  
str(matrizPA)  
  
#str(matrizPA)  
#write.csv(matrizPA, "matrizPA.csv")  
# Create Presences file  
#install.packages("tidyverse", dep=T)  
library(tidyverse)  
library(reshape2)  
  
a <- data.frame(matrizPA)  
b <- melt(a, id.vars = c("Longitude.x.", "Latitude.y."))  
head(b)  
ocorr <- b %>%   
 filter(value > 0)  
ocorr <- as.data.frame(ocorr)  
ocorr <- ocorr[,c("Longitude.x.","Latitude.y.","variable")]  
colnames(ocorr) <- c("lon","lat","variable")  
head(ocorr)  
tail(ocorr)  
  
.rs.unloadPackage("tidyr") # Como este pacote contém funções que são usadas em outros pacotes, estava dando erro  
  
# Remove species with less than 30 records  
occ\_count <- table(unlist(ocorr$variable))  
ocorr <- ocorr[ocorr$variable %in% names(occ\_count)[occ\_count>30],]  
occ\_count1 <- occ\_count[occ\_count>30]  
  
write.csv(occ\_count1, "D:/Leddiv/lilian/Bats/contagem\_ocorr.csv")  
  
sp\_names <- unique(ocorr$variable)  
  
resu <- matrix(nrow = length(sp\_names), ncol = 15)  
colnames(resu) <- c("sp\_name","records","AUC","COR","Deviance","TSS", "iniDist", "30A1B", "30A2", "50A1B", "50A2", "70A1B", "70A2", "90A1B", "90A2")  
vars <- NULL

### Ecologinal niche modelling

Using ensembles and IUCN range maps

for (i in 1:length (sp\_names)){   
   
 t <- sp\_names[[i]]  
 sp <- ocorr[ocorr$variable==t,]  
   
if(nrow(sp)<100){  
 x = 1  
}  
  
if(nrow(sp)>100){  
 x= 0.5}  
  
  
if(nrow(sp)>500){  
 x= 0.25}  
  
if(nrow(sp)>1000){   
 x= 0.125}  
   
 spSample <- sample(1:nrow(sp), replace=F, size=round(x \* nrow(sp)))  
   
 sp <-sp[spSample ,c("lon", "lat")]  
 sp$species <- 1  
   
 coordinates(sp) <- ~ lon + lat  
 sp@proj4string <- CRS("+proj=longlat +datum=WGS84")  
 plot(sp, main = t)  
   
 bb <- bbox(sp)  
   
 bb.buf <- extent(bb[1]-10, bb[3]+10, bb[2]-10, bb[4]+10)  
   
 envs.backg <- crop(pres, bb.buf)  
   
 # Remove collinear variables  
   
 spx <- extract(envs.backg, sp) # extract from file  
 spx <- data.frame(spx) #convert to dataframe  
   
 v <- vifcor(spx, th=0.7) # check collinearity (variance inflation and correlation)  
 bio\_i <- exclude(envs.backg, v) # exclude collinear predictors  
   
   
 # generate sdmData  
 d <- sdm::sdmData(species~ ., train=sp, predictors= bio\_i, bg=list(n=10000,method='gRandom',remove=TRUE))  
   
 # generate sdm model  
 m <- sdm(species ~ ., d, methods=c("brt","maxlike"),   
 replication=c("sub"), test.p=25, n=100,   
 parallelSettings = list(ncore=10, method='parallel'))  
   
 # m  
   
 # Ensembling Present  
 en <- ensemble(m, bio\_i,   
 setting=list(method='weighted',stat='TSS'),  
 parallelSettings = list(ncore=10, method='parallel'))  
   
 plot(en, main="Present")  
   
 writeRaster(en, paste0('D:/Leddiv/lilian/Bats/Mapas/', gsub(" ", "\_", t),"\_pres.tif"), format = "GTiff", overwrite=TRUE)  
   
 # Evaluation  
 e <- getEvaluation(m)  
   
 d1 <- as.data.frame(d)  
   
 # Save which variables were used and evaluation results for all species  
 resu[i, "sp\_name"] <- gsub(" ", "\_", t)  
 resu[i, "records"] <- nrow(d1[d1$species>0, ])  
 resu[i, "AUC"] <- round(mean(e$AUC), 2)  
 resu[i, "COR"] <- round(mean(e$COR), 2)  
 resu[i, "Deviance"] <- round(mean(e$Deviance),2)  
 resu[i, "TSS"] <- round(mean(e$TSS),2)  
   
 vars <- c(as.character(t), names(bio\_i), vars)  
   
 # Find binarization threshold  
 df <- data.frame(as.data.frame(d),coordinates(d)) # presence points and predictors associated  
 pr <- extract(en, df[,c("lon","lat")])  
   
 ev <- evaluates(df$species, pr) # evaluate prediction (observed vs expected)   
 th <- ev@threshold\_based$threshold[[2]] # threshold that maximizes sensitiv + specificity  
   
 # Binary prediction  
 pa <- en   
 pa[] <- ifelse(pa[] >= th, 1, 0) # convert from continuous to binary  
   
 plot(pa, main = t)  
   
 # present PA  
 writeRaster(pa, format = "GTiff",  
 paste0("D:/Leddiv/lilian/Bats/Mapas/",gsub(" ", "\_", t),"\_pres\_PA.tif"),  
 overwrite = T)  
   
 iniDist <- length(pa[pa==1])  
   
 # Ensembling Future  
 # 2030 - A1B  
 # rcp8.5  
 enf1 <- ensemble(m, crop(f30a, bb.buf),  
 setting=list(method='weighted',stat='TSS'),  
 parallelSettings = list(ncore=10, method='parallel'))  
   
   
 enf2 <- ensemble(m, crop(f30b, bb.buf),  
 setting=list(method='weighted',stat='TSS'),  
 parallelSettings = list(ncore=10, method='parallel'))  
   
   
 # 2030 - A2  
 enf3 <- ensemble(m, crop(f30c, bb.buf),  
 setting=list(method='weighted',stat='TSS'),  
 parallelSettings = list(ncore=10, method='parallel'))  
   
   
 enf4 <- ensemble(m, crop(f30d, bb.buf),  
 setting=list(method='weighted',stat='TSS'),  
 parallelSettings = list(ncore=10, method='parallel'))  
   
  
 enf30.1 <- mean(enf1, enf2)  
 #plot(enf30.1, main = t)  
   
 enf30.2 <- mean(enf3, enf4)  
 #plot(enf30.2, main = t)  
   
 writeRaster(enf30.1, paste0("D:/Leddiv/lilian/Bats/Mapas/", gsub(" ", "\_", t),"\_2030\_A1B.tif"), format = "GTiff", overwrite=TRUE)  
 writeRaster(enf30.2, paste0("D:/Leddiv/lilian/Bats/Mapas/", gsub(" ", "\_", t),"\_2030\_A2.tif"), format = "GTiff", overwrite=TRUE)  
   
   
 # Binary prediction future  
 #A1B  
 paf1 <- enf30.1   
 paf1[] <- ifelse(paf1[] >= th, 1, 0) # convert from continuous to binary  
   
 #plot(paf1, main=paste(t, "A1B"))  
   
 finalDist1 <- length(paf1[paf1==1])  
   
 # Future PA  
 writeRaster(paf1, format = "GTiff",  
 paste0("D:/Leddiv/lilian/Bats/Mapas/",gsub(" ", "\_", t),"\_2030\_PA\_A1B.tif"),  
 overwrite = T)  
   
   
 resu[i, "30A1B"] <- finalDist1  
   
 # A2igation  
 paf2 <- enf30.2  
 paf2[] <- ifelse(paf2[] >= th, 1, 0) # convert from continuous to binary  
   
   
 plot(paf2, main=paste(t, "A2"))  
   
 finalDist2 <- length(paf2[paf2==1])  
   
 # Future PA  
 writeRaster(paf2, format = "GTiff",  
 paste0("D:/Leddiv/lilian/Bats/Mapas/",gsub(" ", "\_", t),"\_2030\_PA\_A2.tif"),  
 overwrite = T)  
   
 resu[i, "30A2"] <- finalDist2  
   
}  
  
  
write.csv(resu, "D:/Leddiv/lilian/Bats/Resu/resu.csv")  
write.csv(vars, 'D:/Leddiv/lilian/Bats/Resu/vars.csv')

Now, we will apply different thresholds of forest cover, depending on species vulnerability to forest loss. Three functional groups are defined: - extreme forest specialists - moderate forest specialists - generalists

For the extreme forest specialists, suitable climate regions with less than 50% forest cover are considered unable to support viable populations.

library(rgeos)  
  
# ------ Extreme forest specialists (threatened habitat specialists)  
# Species assessed as Critically Endangered (CR), Endangered (EN), or Vulnerable (VU) are referred to as "threatened" species  
# remove predicted presences from landscapes with less than 50% tree cover  
  
iu <- read.csv("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Planilhas/Lista\_sp.csv", h=T) # List with information on IUCN status  
th <- c("CR", "EN", "VU")  
th\_sp <- iu[iu$IUCN %in% th, ] # select only threatened by IUCN  
sp <- as.character(th\_sp$Species) # these species will be removed from landscapes with less than 50% tree cover  
tableS1 <- data.frame(cbind(sp, rep("Extreme forest specialist", length(sp)))) # To incorporate this information on the Table S1  
colnames(tableS1) <- c("Species", "Classification")  
sp <- gsub(" ", "\_", sp)  
  
# Landscapes with less than 50% tree cover - do not provide habitat for extreme habitat specialists  
land50 <- ref  
land50[land50 < 50] <- 0  
land50[land50 >=50 ] <- 1  
  
  
land50 <- rasterToPolygons(land50, n=16, na.rm=TRUE, dissolve=T) # create mask with only landscapes of more than 50% forest cover  
  
library(maptools)  
writeSpatialShape(land50, "C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Defor/land50.shp")  
  
land50 <- readOGR("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Defor/land50.shp")  
  
# Now remove threatened species from outside these areas  
  
setwd("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Mapas")  
dir()  
  
 for (i in 1: length(sp)) {   
  
 list1 <- list.files(pattern=paste0(sp[i], "\_PA\_"))  
   
 for (j in 1: length(list1)){   
  
 a <- raster(list1[j])  
 b <- (mask(a, land50))  
 writeRaster(b, paste0("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Mapas\_Defaun/", list1[j]), format = "GTiff", overwrite = T)  
 }  
 }

Same same procedure is applied to moderate forest specialists, but to a 30% forest cover threshold.

# ------- Moderate forest specialists (not threatened)  
# remove predicted presences from landscapes with less than 30% tree cover  
  
no\_th <- c("LC", "NT")  
no\_th\_sp <- iu[iu$IUCN %in% no\_th, ] # select only non-threatened  
  
sp\_sp <- no\_th\_sp[no\_th\_sp$Specialists==1, ] # select only specialists non-threatened  
sp\_sp <- as.character(sp\_sp$Species) # these species will be removed from landscapes with less than 50% tree cover  
tableS1.1 <- data.frame(cbind(sp\_sp, rep("Moderate forest specialist", length(sp\_sp)))) # To incorporate this information on the Table S1  
colnames(tableS1.1) <- c("Species", "Classification")  
sp\_sp <- gsub(" ", "\_", sp\_sp)  
  
# Landscapes with more than 30% tree cover - still may provide habitat for habitat specialists  
land30 <- ref  
land30[land30 < 30] <- 0  
land30[land30 >=30 ] <- 1  
plot(land30)  
  
land30 <- rasterToPolygons(land30, n=16, na.rm=TRUE, dissolve=T) # create mask with only landscapes of more than 50% forest cover  
writeSpatialShape(land30, "C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Defor/land30.shp")  
plot(land30)  
land30 <- readOGR("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Defor/land30.shp")  
  
# Now remove specialist species from outside these areas  
setwd("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Mapas")  
dir()  
  
for (i in 1: length(sp\_sp)) {   
   
 list1 <- list.files(pattern=paste0(sp\_sp[i], "\_PA\_"))  
   
 for (j in 1: length(list1)){   
   
 a <- raster(list1[j])  
 b <- (mask(a, land30))  
 writeRaster(b, paste0("C:/Users/lilia/OneDrive/Pos-doc/PNPD/Bats/UFO/Mapas\_Defaun/", list1[j]), format = "GTiff", overwrite = T)  
 }  
}

## Dispersal-restricted potential distribution of bats

In addition to climate suitability , we also simulate occupancy of potential suitable future areas under dispersal constraints posed by landscape fragmentation. The absence of trees was considered a strong barrier to extreme forest specialists; a weak barrier to moderate forest specialists; not a barrier to generalists

library(raster)  
library(rgdal)  
library(MigClim)  
library(sp)  
  
barrier <- tree # raster with future forest cover projections  
resu <- NULL # Table to store results  
  
# Extreme habitat specialists  
# Barrier type = "strong"  
  
for (i in 1: length(ext)) {   
   
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar")  
   
 a <- list.files(pattern = paste0(ext[i]))  
   
 iniDist <- raster(paste0("D:/Leddiv/lilian/Bats/Mapas\_Defaun/", ext[i], "\_pres\_PA.tif"))   
   
   
 barr <- crop(barrier, extent(iniDist)) # species-specific extent of barrier cell  
 barr <- as.data.frame(barr, xy=F)  
 barr[is.na(barr)] <- 0  
   
 iniDist <- as.data.frame(iniDist, xy=T)  
 iniDist[is.na(iniDist)] <- 0  
   
   
 # rcp45  
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar")  
 b <- stack(a)  
 b45 <- raster::subset(b, grep("rcp45", names(b), value=T))  
 b45@crs <- barrier@crs  
   
 hsMap <- as.data.frame(b45) # get coordinates  
   
 hsMap[is.na(hsMap)] <- 0 # all NAs converted to 0  
   
   
 ### Dispersal  
   
   
 MigClim.migrate(iniDist = iniDist,   
 hsMap = hsMap,   
 rcThreshold = 1, #artificial threshold, binarization was made during SDM  
 envChgSteps = ncol(hsMap), dispSteps = ncol(hsMap),   
 barrier = barr, barrierType = "strong", # Barrier is strong to extreme habitat specialists  
 replicateNb=3,   
 testMode=FALSE,   
 fullOutput=T, keepTempFiles=FALSE)   
  
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar/MigClimTest")  
  
  
 #Saving all species results in a single spreadsheet  
 results <- read.table("MigClimTest\_summary.txt", h=T, stringsAsFactors = FALSE)  
 results[1,"simulName"] <- paste0 ( ext[i],"\_rcp45")  
 resu <- rbind (resu, results)  
   
 # Reclassify raster  
 # 0 : unsuitable;   
 # 1 : climate refugia  
 # 2 =< x =<29999 : colonized   
 # x < 0 : locally extinct  
   
 distr.sp <- raster("MigClimTest\_raster.asc")  
 #plot(distr.sp)  
   
 m <- c(2, 29999, 2, #Reclassify positives  
 29999, 30000, 3)  
 rclmat <- matrix(m, ncol=3, byrow=TRUE)  
 rc <- reclassify(distr.sp, rclmat)  
   
 s <- calc(rc, fun=function(x){ #Reclassify negatives  
 x[x < 0] <- 4   
 return(x)} )  
   
 writeRaster(s, paste0("D:/Leddiv/lilian/Bats/Anthrop\_maps/", ext[i],"\_rcp45", ".tif"), format="GTiff", overwrite=T)

Note that the final map has 4 categories 0 : always unsuitable 1 : climate refugia (suitable in the present and in the future) 2 : potential colonization (became suitable and is accessible) 3 : dispersal limitation (became suitable but is unaccessible) 4 : non-analog climate (became unsuitable)

The same is applied to moderate habitat specialists, but the absence of tree is a weak barrier to them

### Moderate habitat specialists  
# Barrier type = "weak"  
  
for (i in 1: length(no\_ext)) {   
   
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar")  
   
 a <- list.files(pattern = paste0(no\_ext[i]))  
   
 iniDist <- raster(paste0("D:/Leddiv/lilian/Bats/Mapas\_Defaun/", no\_ext[i], "\_pres\_PA.tif"))   
   
 barr <- crop(barrier, extent(iniDist))  
 barr <- as.data.frame(barr, xy=F)  
 barr[is.na(barr)] <- 0  
   
 iniDist <- as.data.frame(iniDist, xy=T)  
 iniDist[is.na(iniDist)] <- 0  
   
 nrow(barr) == nrow(iniDist)  
   
 # rcp45  
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar/")  
 b <- stack(a)  
 b45 <- raster::subset(b, grep("rcp45", names(b), value=T))  
 b45@crs <- barrier@crs  
   
 hsMap <- as.data.frame(b45) # do not get coordinates  
   
 nrow(hsMap) == nrow(iniDist)  
   
 hsMap[is.na(hsMap)] <- 0 # all NAs converted to 0  
   
 nrow(hsMap) == nrow(iniDist)  
 nrow(barr) == nrow(iniDist) # TRUE Number of rows must match  
   
   
 ### Dispersal  
   
 MigClim.migrate(iniDist = iniDist,   
 hsMap = hsMap,   
 rcThreshold = 1, #artificial threshold, binarization was made during SDM  
 envChgSteps = ncol(hsMap), dispSteps = ncol(hsMap),   
 barrier = barr, barrierType = "weak", # Barrier is weak to moderate habitat specialists  
 replicateNb=3,   
 testMode=FALSE,   
 fullOutput=T, keepTempFiles=FALSE)   
  
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar/MigClimTest")  
  
  
 #Saving all species results in a single spreadsheet  
 results <- read.table("MigClimTest\_summary.txt", h=T, stringsAsFactors = FALSE)  
 results[1,"simulName"] <- paste0 ( no\_ext[i],"\_rcp45")  
 resu <- rbind (resu, results)  
   
 distr.sp <- raster("MigClimTest\_raster.asc")  
   
 m <- c(2, 29999, 2, #Reclassify positives  
 29999, 30000, 3)  
 rclmat <- matrix(m, ncol=3, byrow=TRUE)  
 rc <- reclassify(distr.sp, rclmat)  
   
 s <- calc(rc, fun=function(x){ #Reclassify negatives  
 x[x < 0] <- 4   
 return(x)} )  
   
 writeRaster(s, paste0("D:/Leddiv/lilian/Bats/Anthrop\_maps/", no\_ext[i],"\_rcp45", ".tif"), format="GTiff", overwrite=T)}

Yet for habitat generalists, the absence of trees is not considered a barrier.

### Habitat generalists  
# Barrier file = empty  
  
no.barr <- barrier  
no.barr[no.barr==1] <- 0 # Transformed all barrier into non-barrier  
  
  
for (i in 1: length(ge)) {   
   
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar/")  
   
 a <- list.files(pattern = paste0(ge[i]))  
 iniDist <- raster(paste0("D:/Leddiv/lilian/Bats/Mapas\_Defaun/", ge[i], "\_pres\_PA.tif"))   
 barr <- crop(no.barr, extent(iniDist))   
  
 iniDist <- as.data.frame(iniDist, xy=T)  
 iniDist[is.na(iniDist)] <- 0  
   
 barr <- as.data.frame(barr, xy=F)  
 barr[is.na(barr)] <- 0  
   
 # rcp45  
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar/")  
 b <- stack(a)  
 b45 <- raster::subset(b, grep("rcp45", names(b), value=T))  
   
 hsMap <- as.data.frame(b45) # do not get coordinates  
 nrow(hsMap) == nrow(iniDist)  
 hsMap[is.na(hsMap)] <- 0 # all NAs converted to 0  
   
   
 ### Dispersal  
   
 MigClim.migrate(iniDist = iniDist,   
 hsMap = hsMap,   
 rcThreshold = 1, #artificial threshold, binarization was made during SDM  
 envChgSteps = ncol(hsMap), dispSteps = ncol(hsMap),   
 barrier = barr, #Barrier file is empty, but required to run simulations  
 replicateNb=3,   
 testMode=FALSE,   
 fullOutput=T, keepTempFiles=FALSE)   
  
  
 setwd("D:/Leddiv/lilian/Bats/Disp\_bar/MigClimTest")  
  
   
 #Saving all species results in a single spreadsheet  
 results <- read.table("MigClimTest\_summary.txt", h=T, stringsAsFactors = FALSE)  
 results[1,"simulName"] <- paste0 ( ge[i],"\_rcp45")  
 resu <- rbind (resu, results)  
   
 distr.sp <- raster("MigClimTest\_raster.asc")  
 m <- c(2, 29999, 2, #Reclassify positives  
 29999, 30000, 3)  
 rclmat <- matrix(m, ncol=3, byrow=TRUE)  
 rc <- reclassify(distr.sp, rclmat)  
   
 s <- calc(rc, fun=function(x){ #Reclassify negatives  
 x[x < 0] <- 4   
 return(x)} )  
   
 writeRaster(s, paste0("D:/Leddiv/lilian/Bats/Anthrop\_maps/", ge[i],"\_rcp45", ".tif"), format="GTiff", overwrite=T)  
   
 }