```
##########################################
# Map Current and Future Local Adaptation #
##########################################
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("qvalue")

# Load library
library(qvalue)
library(adegenet)
library(pcadapt)
library(gradientForest)
library(openxlsx)
setwd("/Volumes/Seagate_JVidal/model_adaptive/")
options(stringsAsFactors = F)
pars <- par()
source("./624functions.R")

# Loads libraries and check if load was successful
libraries <- c("openxlsx","rgdal", "raster", "gtools", "adespatial",
"ade4", "adegraphics", "spdep", "maptools", "adegenet", "qvalue",
"pcadapt", "gdm", "gradientForest", "vegan", "rangeBuilder", "rgeos",
"assigner", "ggfortify")
sapply(libraries, function(x) { suppressMessages(require(x,
character.only=T)) } )

####################
# Load input files #
####################
# Genetic
# Avicennia germinans
map <- "/Volumes/SAMSUNG/Mangue/dataset/agerm/
agerm_q30_thin200_corrected_maf_plink _familyedited.map"
ped <- "/Volumes/SAMSUNG/Mangue/dataset/agerm/
agerm_q30_thin200_corrected_maf_plink _familyedited_ordered.ped"
stru <- "/Volumes/SAMSUNG/Mangue/Analise_Mari/STRUCTURE/
agem_structure.stru"
lfmm <- "/Volumes/SAMSUNG/Mangue/Analise_Mari/
agerm_q30_thin200_corrected_maf_plink _familyedited_ordered.lfmm"
rda <- read.csv("/Volumes/Seagate_JVidal/Modelos_mangue_review/
RDA_Results/outliers_germinans.csv")

# # Avicennia schaueriana
map <- "/Volumes/SAMSUNG/Mangue/dataset/aschau/
avicennia_q30_thin200_aschau.map"
ped <- "/Volumes/SAMSUNG/Mangue/dataset/aschau/
avicennia_q30_thin200_aschau_ordered.ped"
stru <- "/Volumes/SAMSUNG/Mangue/dataset/aschau/
avicennia_q30_thin200_aschau_edited.stru"
lfmm <- "/Volumes/SAMSUNG/Mangue/dataset/aschau/
```

```
avicennia_q30_thin_lfmm.lfmm"
rda <- read.csv("/Volumes/Seagate_JVidal/Modelos_mangue_review/
RDA_Results/outliers_schaueriana.csv")

# Geographic
library(openxlsx)
geographic <- read.xlsx("/Users/jdvidal/Desktop/Desktop_2021/
Results_Frontiers/Final_figures/Supplementary_Table_1.xlsx", sheet=1)
pops <- unique(geographic[-1])
pops <- pops[-2,]

# Subset environmental dataset
p_samp_final_g <- geographic[geographic$species=="Avicennia
germinans",]
p_samp_final_s <- geographic[geographic$species=="Avicennia
schaueriana",]
table(p_samp_final_s$pop)

# Env was obtained by extracting values with a 10km buffer
env_germ <- p_samp_final_g[,6:15]
env_germ$pop <- p_samp_final_g$pop

env_schau <- p_samp_final_s[,6:15]
env_schau$pop <- p_samp_final_s$pop

# Load the sampled points
p_samp_germ <- data.frame("ID"=as.factor(p_samp_final_g$pop),

"latitude"=as.numeric(p_samp_final_g$latitude),

"longitude"=as.numeric(p_samp_final_g$longitude))
p_samp_schau <- data.frame("ID"=as.factor(p_samp_final_s$pop),

"latitude"=as.numeric(p_samp_final_s$latitude),

"longitude"=as.numeric(p_samp_final_s$longitude))

################################
# Search outliers with PCAdapt #
################################
# Read environ dataset
all.env <- env_germ[,-ncol(env_germ)]
all.env <- env_schau[,-ncol(env_schau)]
head(all.env)

# Load genetic data. Can read .vcf or .ped fileformarts.
rsqachau <- read.pcadapt(ped, type="ped")

# Locus name
loci <- read.csv(map, sep="\t", header=FALSE)
```

```
loci <- cbind(1:nrow(loci), as.character(loci[,2]))

# Performs a PCA
x <- pcadapt(rsqachau,K=5)
plot(x,option="screeplot")
plot(x) # 7.08 × 3.5
p <- score_plot(x,i=1,j=2, pop=p_samp_final_s$pop)# 7.08 × 5
library(ggplot2)
p+theme_bw()+theme(plot.title = element_blank(),panel.grid.major =
element_blank(), panel.grid.minor = element_blank())
+geom_hline(yintercept=0, linetype="dashed")+geom_vline(xintercept=0,
linetype="dashed")

# SNPs with q-values less than α (expected FDR) are considered
outliers
qval <- qvalue(x$pvalues)$qvalues
alpha <- 0.01
outliers1.ints <- which(qval<alpha)
outliers1.ints
outliers1 <- as.character(loci[,2])[outliers1.ints]
length(outliers1)
# 243 for germinans
# 224 for schaueriana

# We will keep only those identified both by PCAdapt and RDA
outliers1 <-
outliers1[gsub("_.*$","",outliers1)%in%gsub("_.*$","",rda$snp)]
length(outliers1)
# 118 for germinans
# 182 for schaueriana

# Which PCs are the most correlated with the oulier SNPs? Here are the
first 10
snp_pc <- get.pc(x,outliers1.ints)
head(snp_pc, 10)
nrow(snp_pc)
index <- snp_pc[,1]
result <- data.frame("index"=NA,
                     "locus"=NA)

for(i in index){
  result <- rbind(result,loci[i,])
}
result <- result[-1,]
head(result)

# This section follows Fitzpatrick and Keller (2014). There are two
methods that attack the problem of predicting and mapping local
adaptation in slightly different ways: gradient forests (GF) and
generalized dissimilarity modeling (GDM). GF will be used to predict
```

```
allele frequencies of “reference” and “outlier” loci across space and
time, whereas GDM will predict the genetic structure (Fst) among those
populations of frequencies.
data.file.4 <- stru

# Load the dataset
snp_file <- lfmm
snp_table <- read.table(snp_file)

############################
# Gradient Forest Modeling #
############################
#First, read in and format minor allele frequency data.
n.ind <- dim(read.table(data.file.4, skip=1)[,-c(1:2)])[1]/2
n.loc <- dim(read.table(data.file.4, skip=1)[,-c(1:2)])[2]
gi <- read.structure(data.file.4, n.ind=n.ind, n.loc=n.loc, col.lab=1,
col.pop=2,
                     onerowperind=F,col.others=0, row.marknames=1,
NA.char="-9")
gi

# Germ
gi@pop <- as.factor(env_germ$pop)

# Schau
gi@pop <- as.factor(env_schau$pop)

##  Converting data from a STRUCTURE .stru file to a genind object...
gi.p <- genind2genpop(gi)
gi.freq <- makefreq(gi.p)

# Make sure we have an even number of outliers for the next step to
work
ncol(gi.freq)
if(ncol(gi.freq)%%2!=0){gi.freq <- gi.freq[,-ncol(gi.freq)]}
if(ncol(gi.freq)%%2!=0){gi.freq <- gi.freq[,-1]}
minor <- colMeans(gi.freq[,seq(1, ncol(gi.freq), 2)], na.rm=T) >
colMeans(gi.freq[,seq(2, ncol(gi.freq), 2)], na.rm=T)
keep <- sort(c(which(minor==T)*2-1, which(minor==F)*2))
gi.freq <- data.frame(gi.freq[,keep])
print(gi.freq[1:6,1:10], digits=3)
length(outliers1)

# 118 for germinans
# 182 for schaueriana

# Make an environmental dataset.
envGF <- all.env[match(row.names(gi.freq), env_germ$pop),]
envGF <- all.env[match(row.names(gi.freq), env_schau$pop),]
```

```
# Build individual SNP datasets, with 300 randomly-selected reference
SNPs, vs. all outlier SNPs.
SNPs_ref <- gi.freq[, sample(length(gi.freq), 300)]
SNPs_out <- gi.freq[,gsub("\\.0.*", "", names(gi.freq)) %in%
outliers1]

# Account for correlations, see ?gradientForest.
maxLevel <- log2(0.368*nrow(envGF)/2)

# Fit gf models for reference and outlier SNPs.
gfRef <- gradientForest(cbind(envGF[,c(1:10)], SNPs_ref),
predictor.vars=colnames(envGF[,c(1:10)]),
                        response.vars=colnames(SNPs_ref), ntree=500,
trace=T)
gfOut <- gradientForest(cbind(envGF[,c(1:10)], SNPs_out),
predictor.vars=colnames(envGF[,c(1:10)]),
                        response.vars=colnames(SNPs_out), ntree=500,
trace=T)
# Check the results
gfRef
gfOut

# Plot the output, see ?plot.gradientForest.
plot(gfRef, plot.type="O")
plot(gfOut, plot.type="O")
gfRef$overall.imp
res_gf <- cbind("Overall importance
(reference)"=round(gfRef$overall.imp2, 2),
                "Overall importance
(outlier)"=round(gfOut$overall.imp2, 2))

# Save the result
write.csv(res_gf, "/Users/jdvidal/Desktop/Mangue_Review/
Results_December/Var_importance_schau.csv")

####################################
# Obtaining the geographic dataset #
####################################
library(raster)
# Create a data frame of raster pixel values, for all pixels in region
of interest.
files1 <- list.files(path="/Volumes/SAMSUNG/Camadas ambientais/Mangue/
CHELSA/CHELSA_Present/Uncor/", pattern='.tif$', full.names = TRUE)
files1_fut <- list.files(path="/Volumes/SAMSUNG/Camadas ambientais/
Mangue/CHELSA/CHELSA_Future/Uncor/", pattern='.tif$', full.names =
TRUE)
coordinates(geographic) <- ~longitude+latitude
poly <- buffer(geographic, 50000)

# Make a raster stack
```

```
s3 <- raster::stack(files1)

# Remove collinearity step
# To create 1000 random points
library(dismo)
sp_points <- randomPoints(s3$Present.Surface.Salinity_interpolated,
1000)

# Extract values for predictors
backgr <- extract(s3, sp_points)
colnames(backgr)
colnames(backgr) <- gsub("_.*", "",gsub("CHELSA_", "",
colnames(backgr)))
colnames(backgr)[8:9] <- c("salt", "surft")

# Calculate and visualize pearsons correlation
library(corrplot)
corr <- cor(backgr)
corrplot.mixed(corr, lower = 'square', upper = 'number')

# Extract environmental data
xy <- as.data.frame(cbind("longitude" = geographic$longitude,
"latitude" = geographic$latitude))
coordinates(xy) <- ~longitude+latitude
projection(xy) <- CRS('+proj=longlat +datum=WGS84')
projection(s3) <- CRS('+proj=longlat +datum=WGS84')
mang.env.data <- extract(s3, xy, buffer=10000, fun=mean)
mang.env.data <- cbind(geographic, mang.env.data)
write.csv(mang.env.data, "/Volumes/Seagate_JVidal/model_adaptive/
all.env_schau_final_dec_2021.csv")

# Extracts cell number to use in the prediction
all.r <- c(s3)
all.s <- stack(all.r)
mask <- all.s[[1]]/all.s[[1]] - 1
all.env.pred <- data.frame(extract(s3, poly, cellnumbers=T))
all.env.pred

# Creates the distance raster
poly_dist <- buffer(geographic, 50000, dissolve=FALSE)
rr <- rasterize(poly_dist, mask, field="dist")
#writeRaster(rr, "/Volumes/Seagate_JVidal/Modelos_mangue_review/
CHELSA_Future/dist", format="GTiff")

# Extracts values
longlat <- xyFromCell(mask, all.env.pred$cell)
all.env.pred <- data.frame(all.env.pred$cell, longlat,
all.env.pred[,-1])
head(all.env.pred)
names(all.env.pred)[1:3] <- c("cell", "longitude", "latitude")
```

```
colnames(all.env.pred)
colnames(all.env.pred) <- c("cell", "longitude", "latitude", "bio1",
"bio12", "bio15", "bio3", "bio7", "gsl", "ngd10", "dist", "salt",
"surft")
all.env.pred <- all.env.pred[,-10]
# Predict/transform allele frequencies using current environment and
gf models, see ?predict.gradientForest.
predRef.gf <- predict(gfRef, all.env.pred[,-c(1:3)])
predOut.gf <- predict(gfOut, all.env.pred[,-c(1:3)])

# Now, let's map PCAs of reference and outlier SNPs: PCA1 (red), PC2
(green), PC3 (blue).
refRGBmap <- pcaToRaster(predRef.gf, mask, all.env.pred$cell)
plotRGB(refRGBmap)
outRGBmap <- pcaToRaster(predOut.gf, mask, all.env.pred$cell)
plotRGB(outRGBmap, axes=F)

# Let's see the Procrustes difference between maps (reference vs.
outlier SNPs).
diffMap1 <- RGBdiffMap(predRef.gf, predOut.gf, rast=mask,
mapCells=all.env.pred$cell)

# Plot to inspect
library(RColorBrewer)
ramp <- brewer.pal(9,"Reds")
plot(diffMap1[[2]], xaxt='n', yaxt='n', bty="n", legend=F, box=F,
col=ramp, zlim=c(0.0001,0.0205))
plot(bound, add=T)
addRasterLegend(diffMap1[[2]], location="bottomright", ramp=ramp,
digits=4, nTicks=5)

# Write the results
writeRaster(diffMap1[[2]], "/Volumes/Seagate_JVidal/
Resultados_artigo_1/Procrustres_diff_schau", overwrite=TRUE)

##########
# Future #
##########
# Now, predict/transform allele frequencies using FUTURE environment
and gf models, see ?predict.gradientForest.
s3_f <- raster::stack(files1_fut)
names(s3_f)
s3_f <- crop(s3_f, c(extent(poly)[1]-5,
                     extent(poly)[2]+5,
                     extent(poly)[3]-5,
                     extent(poly)[4]+5))
all.r_f <- c(s3_f)
all.s_f <- stack(all.r_f)
mask <- all.s_f[[1]]/all.s_f[[1]] - 1
all.env.pred_f <- data.frame(extract(s3_f, poly, cellnumbers=T ) )
```

```
longlat_f <- xyFromCell(mask, all.env.pred_f$cell)
all.env.pred_f <- data.frame(all.env.pred_f$cell, longlat_f,
all.env.pred_f[,-1])
names(all.env.pred_f)[1:3] <- c("cell", "longitude", "latitude")
all.env.fut.pred <- all.env.pred_f[,-10]

names(all.env.fut.pred) <- names(all.env.pred)
all.env.fut.pred <- all.env.fut.pred[,-c(2:3)]
head(all.env.fut.pred)
projRef.gf <- predict(gfRef, all.env.fut.pred[,-1])
projOut.gf <- predict(gfOut, all.env.fut.pred[,-1])

# Let's see the procrustes difference (first reference, second
outliers) across time (future vs. current environment).
diffMap6 <- RGBdiffMap(projRef.gf, predRef.gf, rast=mask,
mapCells=all.env.pred$cell)
plot(diffMap6[[2]], xaxt='n', yaxt='n', bty="n", legend=F, box=F,
col=ramp)
addRasterLegend(diffMap6[[2]], location="topright", ramp=ramp,
digits=4, nTicks=5)
writeRaster(diffMap6[[2]], "/Volumes/Seagate_JVidal/
Resultados_artigo_1/Time_Ref_Procrustres_diff_aschau.grd",
overwrite=TRUE)

diffMap7 <- RGBdiffMap(projOut.gf, predOut.gf, rast=mask,
mapCells=all.env.pred$cell)
plot(diffMap7[[2]], xaxt='n', yaxt='n', bty="n", legend=F, box=F,
col=ramp)
addRasterLegend(diffMap7[[2]], location="topright", ramp=ramp,
digits=4, nTicks=5)
writeRaster(diffMap7[[2]], "/Volumes/Seagate_JVidal/
Resultados_artigo_1/Time_Out_Procrustres_diff_aschau.grd",
overwrite=TRUE)

# Lastly, what's the temporal change in local adaptation (current
reference - current outlier) - (future reference - future outlier)?
This again uses procrustes analysis.
diffMap8 <- RGBdiffMap2(predRef.gf, predOut.gf, projRef.gf,
projOut.gf, rast=mask, mapCells=all.env.pred$cell)
plot(diffMap8[[2]], xaxt='n', yaxt='n', bty="n", legend=F, box=F,
col=ramp)
addRasterLegend(diffMap8[[2]], location="topright", ramp=ramp,
digits=4, nTicks=5)
writeRaster(diffMap8[[2]], "/Volumes/Seagate_JVidal/
Resultados_artigo_1/Adapt_Procrustres_diff_aschau.grd",
overwrite=TRUE)

########################################
# Plotting differences for A. germinans #
########################################
```

```
library(raster)
rasters_results <- list.files("/Volumes/Seagate_JVidal/
Resultados_artigo_1/", pattern = ".gri", full.names = T)
rasters_results
adapt_germ <- raster(rasters_results[3])
p_samp_germ
p_samp_germ <- unique(as.data.frame(p_samp_germ))
p_samp_germ$ID <- gsub("Ag", "", p_samp_germ$ID)
p_samp_germ <- p_samp_germ[p_samp_germ$ID!="PAa",]
psplot <- as.data.frame(p_samp_germ)
p_samp_germ
class(psplot$longitude)
ggplot(psplot, aes(y=latitude, x=longitude, label=ID))+geom_point()
+geom_label()
head(as.data.frame(p_samp_germ))
coordinates(p_samp_germ) <- ~longitude+latitude
buffer_results_g <- buffer(p_samp_germ, 50000, dissolve=FALSE)
#adapt_germ <- mask(adapt_germ, buffer_results)
points_result_values_g <- rasterToPoints(adapt_germ, spatial = TRUE)
buffer_results_values_g <- intersect(points_result_values_g,
buffer_results_g)
vals_df_g <- data.frame(as.data.frame(buffer_results_values_g))
buffer_results_values_g$d <- as.factor(buffer_results_values_g$d)
levels(buffer_results_values_g$d)
names(vals_df_g) <- c("Diff", "Population", "longitude", "latitude")
buffer_results_values_g
for(i in 1:nrow(vals_df_g)){
  if(vals_df_g$latitude[i]<=-6.61){
    vals_df_g$NEESA[i] <- "South"
  }
  else{
    vals_df_g$NEESA[i] <- "North"
  }
}
vals_df_g$NEESA <- factor(vals_df_g$NEESA, levels=c("North", "South"))
vals_df_g$Population <- factor(vals_df_g$Population)
levels(vals_df_g$Population)
vals_df_backup <- vals_df_g
vals_df_g$Population <- factor(vals_df_g$Population, levels = c("MRJ",
"PAb", "ALC", "PNB", "PRC","TMD"))

library(ggplot2)
p_g <- ggplot(data=subset(vals_df_g, NEESA %in% c("North", "South")),
aes(x=Population, y=Diff, fill=Population))+
  facet_grid(~NEESA, scales = "free", space = "free")+
  geom_boxplot()+
  ggtitle("Avicennia germinans")+
  ylab("Procrustes difference\n(present - future)")+
  xlab(element_blank())+
  theme_light()+theme(plot.title = element_text(hjust = 0.5,
```

```
face="italic"), axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1),
                       axis.text.y = element_text(angle = 90, vjust =
0.5, hjust=1), legend.position = "none")
plot(p_g)
#4.5 × 3.5

############################################################
# Temporal change in local adaptation for A. schaueriana #
############################################################
rasters_results
adapt_schau <- raster(rasters_results[4])
levels(p_samp_schau$ID) <- c("ALC", "CNN", "FLN", "GPM", "LGN", "PAR",
"PPR", "PRC", "UBA", "VER")
p_samp_schau <- unique(as.data.frame(p_samp_schau))
coordinates(p_samp_schau) <- ~longitude+latitude
buffer_results <- buffer(p_samp_schau, 50000, dissolve=FALSE)
points_result_values <- rasterToPoints(adapt_schau, spatial = TRUE)
buffer_results_values <- intersect(points_result_values,
buffer_results)

vals_df_s <- data.frame(as.data.frame(buffer_results_values))
names(vals_df_s) <- c("Diff", "Population", "longitude", "latitude")
head(vals_df_s)

for(i in 1:nrow(vals_df_s)){
  if(vals_df_s$latitude[i]<=-6.61){
    vals_df_s$NEESA[i] <- "South"
  }
  else{
    vals_df_s$NEESA[i] <- "North"
  }
}
vals_df_s$Population <- factor(vals_df_s$Population,levels = c("PAR",
"ALC", "PRC", "VER", "GPM", "UBA", "CNN", "PPR", "FLN", "LGN"))
vals_df_s$NEESA <- factor(vals_df_s$NEESA,levels = c("North",
"South"))

p_s <- ggplot(data=subset(vals_df_s, NEESA %in% c("North", "South")),
aes(x=Population, y=Diff, fill=Population))+
  facet_grid(~NEESA, scales = "free", space = "free")+
  geom_boxplot()+
  ggtitle("Avicennia schaueriana")+
  ylab("Procrustes difference\n(present - future)")+
  xlab(element_blank())+
  theme_light()+theme(plot.title = element_text(hjust = 0.5,
face="italic"), axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1),
                       axis.text.y = element_text(angle = 90, vjust =
0.5, hjust=1), legend.position = "none")
```

```
plot(p_s)

########################
# Differences plotting #
########################
# Load libraries
library(raster)
library(openxlsx)
# Load data
vals_df <- read.csv("/Users/jdvidal/Desktop/Mangue_Review/
Results_December/Vals_differences_all")
vals_df_g <- vals_df[vals_df$species=="Avicennia germinans",]
vals_df_s <- vals_df[vals_df$species=="Avicennia schaueriana",]
vals_df_g$Population <- as.factor(as.character(vals_df_g$Population))
vals_df_s$Population <- as.factor(as.character(vals_df_s$Population))

# Load deforestation
def <- read.xlsx("/Users/jdvidal/Desktop/Mangue_Review/
Results_December/Deforestation_processed.xlsx")
def
def_s <- def[def$Species=="Avicennia schaueriana",]
def_s$Population <- as.factor(as.character(def_s$Population))
def_g <- def[def$Species=="Avicennia germinans",]
def_g$Population <- as.factor(as.character(def_g$Population))

# Creates a dataframe to keep the results
# Germinans
substed_def_g <- data.frame()
for(i in 1:length(levels(vals_df_g$Population))){
N <-
round(nrow(vals_df_g[vals_df_g$Population==levels(vals_df_g$Population
)[i],])*(def_g[def_g$Population==vals_df_g$Population[i],]
$`Accumulated.forest.loss.since.2000.(%)`/100))
print(N)
vals_pop_target <-
vals_df_g[vals_df_g$Population==levels(vals_df_g$Population)[i],]
vals_pop_target <- vals_pop_target[sample(nrow(vals_pop_target), N),]
vals_pop_target$treatment <- "Deforestation"
head(vals_pop_target)
substed_def_g <- rbind(substed_def_g, vals_pop_target)
}
substed_def_g <- rbind(vals_df_g, substed_def_g)
substed_def_g <-
substed_def_g[substed_def_g$treatment=="Deforestation",]

# Schaueriana
substed_def_s <- data.frame()
for(i in 1:length(levels(vals_df_s$Population))){
  N <-
round(nrow(vals_df_s[vals_df_s$Population==levels(vals_df_s$Population
```

```
)[i],])*(def_s[def_s$Population==vals_df_s$Population[i],]
$`Accumulated.forest.loss.since.2000.(%)`/100))
  print(N)
  vals_pop_target <-
vals_df_s[vals_df_s$Population==levels(vals_df_s$Population)[i],]
  vals_pop_target <- vals_pop_target[sample(nrow(vals_pop_target),
N),]
  vals_pop_target$treatment <- "Deforestation"
  head(vals_pop_target)
  substed_def_s <- rbind(substed_def_s, vals_pop_target)
}
substed_def_s <- rbind(vals_df_s, substed_def_s)
substed_def_s <-
substed_def_s[substed_def_s$treatment=="Deforestation",]

# Include deforestation
# Germ
substed_def_g$deforestation <- NA
for(i in 1:nrow(substed_def_g)){
  substed_def_g[i,]$deforestation <-
def_g[def_g$Population==substed_def_g[i,]$Population,]
$`Accumulated.forest.loss.since.2000.(%)`
}
# Schau
substed_def_s$deforestation <- NA
for(i in 1:nrow(substed_def_s)){
  substed_def_s[i,]$deforestation <-
def_s[def_s$Population==substed_def_s[i,]$Population,]
$`Accumulated.forest.loss.since.2000.(%)`
}
def_s
# Plot results
library(ggplot2)
substed_def_s$Population <- factor(substed_def_s$Population,
levels=c("PAR", "ALC", "PRC", "VER", "GPM", "UBA", "CNN", "PPR",
"FLN", "LGN"))
p_s <- ggplot(data=subset(substed_def_s, NEESA %in% c("North",
"South")), aes(x=Population, y=Diff*50, fill=Population))+
  facet_grid(~NEESA, scales = "free", space = "free")+
  geom_violin()+
  geom_boxplot(width=0.1, fill="white")+
  geom_point(data=subset(substed_def_s, NEESA %in% c("North",
"South")), aes(x=Population, y=deforestation/100), pch=23,
fill="yellow", size=2)+
  ggtitle("Avicennia schaueriana")+
  ylab("Procrustes difference\n(present - future)*50")+
  xlab(element_blank())+
  scale_y_continuous(limits = c(0,1), position = "right")+
  theme_light()+theme(plot.title = element_text(hjust = 0.5,
face="italic"), axis.text.x = element_text(angle = 90, vjust = 0.5,
```

```
hjust=1),
                       axis.text.y = element_text(angle = 90, vjust =
0.5, hjust=1), legend.position = "none")
plot(p_s)
substed_def_g$Population <- factor(substed_def_g$Population,
levels=c("MRJ", "PAb", "ALC", "PNB", "PRC","TMD"))
p_g <- ggplot(data=subset(substed_def_g, NEESA %in% c("North",
"South")), aes(x=Population, y=Diff*50, fill=Population))+
  facet_grid(~NEESA, scales = "free", space = "free")+
  geom_violin()+
  geom_boxplot(width=0.1, fill="white")+
  geom_point(data=subset(substed_def_g, NEESA %in% c("North",
"South")), aes(x=Population, y=deforestation/100), pch=23,
fill="yellow", size=2)+
  ggtitle("Avicennia germinans")+
  ylab("Procrustes difference\n(present - future)*50")+
  xlab(element_blank())+
  scale_y_continuous(limits = c(0,1), position = "right")+
  theme_light()+theme(plot.title = element_text(hjust = 0.5,
face="italic"), axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1),
                       axis.text.y = element_text(angle = 90, vjust =
0.5, hjust=1), legend.position = "none")
plot(p_g)

library(ggsci)
pg_npg <- p_g+scale_fill_npg()
ps_npg <- p_s+scale_fill_npg()
plot(pg_npg)
plot(ps_npg)

# Write an output file with the Procrustes results
ag_g <- aggregate(Diff ~ Population, substed_def_g, function(x) c(mean
= mean(x), sd = sd(x)))
ag_s <- aggregate(Diff ~ Population, substed_def_s, function(x) c(mean
= mean(x), sd = sd(x)))
ag <- rbind(cbind(species="Avicennia germinans", ag_g),
cbind(species="Avicennia schaueriana", ag_s))
write.csv(ag, "/Users/jdvidal/Desktop/Mangue_Review/Results_December/
Differences_proc_final")

##########################
# Plor individual models #
##########################
# Geographic
# Load basis for map plotting
bound <- shapefile("/Volumes/SAMSUNG/Bases_Bia/Americas.shp")
rivers <- shapefile("/Volumes/SAMSUNG/Notebook/BASES/AMB/
Data_and_Documentation/GIS_Data_Files/Rivers/riv_no_grid.shp", warnPRJ
= FALSE)
```

```
grids <- shapefile("/Volumes/SAMSUNG/Notebook/BASES/AMB/
Data_and_Documentation/GIS_Data_Files/GRID2/ne_10m_graticules_15/
ne_10m_graticules_15.shp")
base_elev <-  raster("/Volumes/SAMSUNG/Camadas ambientais/Elevation/
alt.bil")

p_samp_final_g <- read.xlsx("/Users/jdvidal/Downloads/agerm_PED_info
(1).xlsx")
p_samp_final_s <- read.csv("/Volumes/SAMSUNG/Mangue/dataset/aschau/
aschau_points.csv")

# Load the sampled points
p_samp_germ <- data.frame("ID"=as.factor(p_samp_final_g$pop),

"latitude"=as.numeric(p_samp_final_g$latitude),

"longitude"=as.numeric(p_samp_final_g$longitude))
p_samp_schau <-
data.frame("ID"=as.factor(gsub('[0-9]',"",p_samp_final_s$ID)),

"latitude"=as.numeric(p_samp_final_s$latitude),

"longitude"=as.numeric(p_samp_final_s$longitude))
p_samp_germ <- unique(p_samp_germ)
p_samp_schau <- unique(p_samp_schau)

coordinates(p_samp_germ) <- ~longitude+latitude
coordinates(p_samp_schau) <- ~longitude+latitude
buffer_germ <- buffer(p_samp_germ, 50000, dissolve=F)
buffer_schau <- buffer(p_samp_schau, 50000, dissolve=F)
map_result_germ <- mask(map_result_germ, buffer_germ)
map_result_schau <- mask(map_result_schau, buffer_schau)
buffer_germ$ID <- p_samp_germ$ID
buffer_schau$ID <- p_samp_schau$ID
library(RColorBrewer)
ramp <- rev(brewer.pal(9,"Spectral"))

# Save individual rasters for germinans
par(mfrow=c(7,1))
i <- 1
for(i in 1:length(buffer_germ$ID)){
  #pdf(file=paste0("/Volumes/Seagate_JVidal/Resultados_artigo_1/
Models_per_population/", buffer_germ$ID[i], ".pdf"), height=5,
width=5)
  ext <- extent(buffer_germ[buffer_germ$ID==buffer_germ$ID[i],])
  grd <- gridlines(buffer_germ[buffer_germ$ID==buffer_germ$ID[i],])
  ind_pop <- crop(map_result_germ, ext)
  writeRaster(ind_pop, filename=paste0("/Volumes/Seagate_JVidal/
Resultados_artigo_1/Models_per_population/Germinans",
buffer_germ$ID[i]),format="GTiff")
```

```
  plot(map_result_germ, xlim=c(ext[1], ext[2]), ylim=c(ext[3],
ext[4]), col=ramp, legend=F, main=buffer_germ$ID[i], zlim=c(0,0.015))
  plot(grd, add=TRUE, border="grey", lty="dashed", xlim=c(ext[1],
ext[2]), ylim=c(ext[3], ext[4]), lwd=0.2)
  #dev.off()
}

# Save individual rasters for schaueriana
par(mfrow=c(10,1))
i <- 1
for(i in 1:length(buffer_schau$ID)){
  #pdf(file=paste0("/Volumes/Seagate_JVidal/Resultados_artigo_1/
Models_per_population/", buffer_schau$ID[i], ".pdf"), height=5,
width=5)
  ext <- extent(buffer_schau[buffer_schau$ID==buffer_schau$ID[i],])
  grd <- gridlines(buffer_schau[buffer_schau$ID==buffer_schau$ID[i],])
  ind_pop <- crop(map_result_schau, ext)
  writeRaster(ind_pop, filename=paste0("/Volumes/Seagate_JVidal/
Resultados_artigo_1/Models_per_population/Schaueriana",
buffer_schau$ID[i]),format="GTiff")
  plot(map_result_schau, xlim=c(ext[1], ext[2]), ylim=c(ext[3],
ext[4]), col=ramp, legend=F, main=buffer_schau$ID[i], zlim=c(0,0.015))
  plot(grd, add=TRUE, border="grey", lty="dashed", xlim=c(ext[1],
ext[2]), ylim=c(ext[3], ext[4]), lwd=0.2)
  #dev.off()
}

################
# Map plotting #
################
plot(map_result_germ, xlim=c(ext[1], ext[2]), ylim=c(ext[3], ext[4]),
col=ramp, legend=T, main=buffer_germ$ID[i], zlim=c(0,0.015))
map_result_germ
plot(bound, xlim=c(-67,-37), ylim=c(-30,5), col="#e8e8e8", axes=TRUE)
points(p_samp_germ, pch=21, bg="black", cex=2,)
points(p_samp_schau, pch=24, bg="grey", cex=2)
text(p_samp_germ$latitude~p_samp_germ$longitude,
labels=p_samp_germ$ID, data=p_samp_germ, cex=0.9, font=2, pos=4)
text(p_samp_schau$latitude~p_samp_schau$longitude,
labels=p_samp_schau$ID, data=p_samp_schau, cex=0.9, font=2, pos=4)

p_samp_germ <- p_samp_germ[unique(p_samp_germ$ID),]
p_samp_schau <- p_samp_schau[unique(p_samp_schau$ID),]
# Plot to check
colors <- colorRampPalette(c("white","black"))
#colpallet <- c("#f6f5c8","#2a3171")
plot(base_elev, col=colors(1000), legend=FALSE, axes=FALSE,
xlim=c(-70,-26.5), ylim=c(-42.5,18))
plot(bound, border="black", lwd=0.6, add=TRUE, xlim=c(-70,-26.5),
ylim=c(-42.5,18))
```

```
plot(rivers, border="black", lwd=0.1, add=TRUE, xlim=c(-70,-26.5),
ylim=c(-42.5,18))
plot(grids, add=TRUE, border="grey", lty="dashed", lwd=0.2,
xlim=c(-70,-26.5))
#points(p1$Longitude, p1$Latitude, col="black",
bg=colpallet[as.factor(p1$Population)], pch=21, cex=1.3)
#points(p_samp$longitude, p_samp$latitude, col="black", pch=21,
cex=1.6)
points(p_samp_germ$longitude, p_samp_germ$latitude, pch=21,
bg="black", cex=1.2)
points(p_samp_schau$longitude, p_samp_schau$latitude, pch=21,
bg="black", cex=1.2)
scalebar(500, xy=c(-35,-38), type='line', divs=1)
#breaks <- seq(from = 1000, to = 4000, by = 500)
#op <- par(cex = 0.7)
#legend("bottomleft",leg=breaks,fill=colors(10), ncol=2, bg="white")

# Plot env per pop
mang.env.data_pres_schau <- extract(s3, p_samp_schau, buffer=10000,
fun=mean)
mang.env.data_pres_schau <- cbind(as.data.frame(p_samp_schau),
mang.env.data_pres_schau)

mang.env.data_pres_germ <- extract(s3, p_samp_germ, buffer=10000,
fun=mean)
mang.env.data_pres_germ <- cbind(as.data.frame(p_samp_germ),
mang.env.data_pres_germ)

mang.env.data_fut_schau <- extract(s3_f, p_samp_schau, buffer=10000,
fun=mean)
mang.env.data_fut_schau <- cbind(as.data.frame(p_samp_schau),
mang.env.data_fut_schau)

mang.env.data_fut_germ <- extract(s3_f, p_samp_germ, buffer=10000,
fun=mean)
mang.env.data_fut_germ <- cbind(as.data.frame(p_samp_germ),
mang.env.data_fut_germ)

mang.env.data_pres_schau <- cbind(mang.env.data_pres_schau,
species="avicennia schaueriana", time="present")
mang.env.data_pres_germ <- cbind(mang.env.data_pres_germ,
species="avicennia germinans", time="present")
mang.env.data_fut_schau <- cbind(mang.env.data_fut_schau,
species="avicennia schaueriana", time="future")
mang.env.data_fut_germ <- cbind(mang.env.data_fut_germ,
species="avicennia germinans", time="future")

colnames(mang.env.data_fut_schau) <-
colnames(mang.env.data_pres_schau)
colnames(mang.env.data_fut_germ) <- colnames(mang.env.data_pres_germ)
```

```
mang.env.data_fut_schau$pop <- as.factor(gsub("[0-9]", "",
gsub("As","",mang.env.data_fut_schau$ID)))
mang.env.data_pres_schau$pop <- as.factor(gsub("[0-9]", "",
gsub("As","",mang.env.data_pres_schau$ID)))
mang.env.data_fut_germ$pop <- as.factor(gsub("[0-9]", "",
gsub("Ag","",mang.env.data_fut_germ$ID)))
mang.env.data_pres_germ$pop <- as.factor(gsub("[0-9]", "",
gsub("Ag","",mang.env.data_pres_germ$ID)))

vals_fut_pres <- as.data.frame(rbind(mang.env.data_pres_schau,
mang.env.data_pres_germ, mang.env.data_fut_schau,
mang.env.data_fut_germ))

p_pops_p <- vals_fut_pres[ !duplicated(vals_fut_pres$pop), ]
p_pops_f <- vals_fut_pres[vals_fut_pres$time=="future",]
p_pops_f <- p_pops_f[ !duplicated(p_pops_f$pop), ]
p_pops <- rbind(p_pops_p, p_pops_f)
breaks <- round(p_pops$latitude, 2)
labels <- p_pops$pop
p_pops$time <- as.factor(p_pops$time)
summary(p_pops)
require(gridExtra)
plot1 <- ggplot(data=p_pops, aes(x=latitude, y=bio_1/10, group=time,
color=time))+scale_x_continuous(breaks=breaks, labels=labels)
+geom_line()+geom_point()+theme_bw()+coord_flip()+ylab("Annual Mean
Temperature (°C)")+xlab("Population")+theme(panel.grid.major.y =
element_line(colour="grey", linetype="dashed"), legend.position =
"none")
plot2 <- ggplot(data=p_pops, aes(x=latitude, y=bio_12, group=time,
color=time))+scale_x_continuous(breaks=breaks, position = "top")
+geom_line()+geom_point()+theme_bw()+coord_flip()+ylab("Annual
Precipitation (mm)")+xlab("Latitude")+theme(panel.grid.major.y =
element_line(colour="grey", linetype="dashed"), legend.position =
"none")
grid.arrange(plot1, plot2, ncol=2)

#########################################################################
#########################################################################
############################
########################
# Niche modelling step #
########################
# Set the working directory
setwd("/Volumes/SAMSUNG/Mangue/Mangrove_model/")

# Load libraries
library(rJava)
library(rgbif)
library(raster)
library(maptools)
```

```
library(dismo)
library(sp)
library(openxlsx)
library(CoordinateCleaner)

##############################
# List files with occurrences #
##############################
# Geographic
p_samp_final_g <- as.data.frame(read.csv("agerm_processed_final")
[,-1])
p_samp_final_s <- as.data.frame(read.csv("aschau_processed_final")
[,-1])
input_clean_valid <- read.csv('filtered_file_occurrences')

############################
# List environmental layers #
############################
files1 <- list.files(path="/Volumes/SAMSUNG/Camadas ambientais/Mangue/
CHELSA/CHELSA_Present/Uncor/", pattern='.tif$', full.names = TRUE)
files1_fut <- list.files(path="/Volumes/SAMSUNG/Camadas ambientais/
Mangue/CHELSA/CHELSA_Future/Uncor/", pattern='.tif$', full.names =
TRUE)
coordinates(input_clean_valid) <- ~longitude+latitude
poly <- buffer(input_clean_valid, 50000)

# Make a raster stack
mask <- shapefile("/Volumes/SAMSUNG/Mangue/pol.shp")
s3 <- raster::stack(files1)
s3 <- mask(s3, mask)
s3 <- stack(s3)
s3 <- dropLayer(s3, 8)

##########
# Future #
##########
# Now, predict/transform allele frequencies using FUTURE environment
and gf models, see ?predict.gradientForest.
s3_f <- raster::stack(files1_fut)
s3_f <- dropLayer(s3_f, 8)
names(s3_f) <- names(s3)
s3_f <- mask(s3_f, mask)
s3_f <- stack(s3_f)

####################
# Making the models #
####################
#install.packages("SSDM")
library(SSDM)
i <- 1
```

```
input <- input_clean_valid

for(i in 1:2){
  # Selects the species to be modelled
  active_modeling_species <- levels(input$scientificname)[i]
  active_modeling_species
  spp_coordinates <-
input[input$scientificname==levels(input$scientificname)[i],]
  summary(spp_coordinates)
  spp_coordinates <-
spp_coordinates[complete.cases(spp_coordinates$latitude),]
  spp_coordinates <- crop(spp_coordinates,
extent(s3$CHELSA_bio1_1981.2010_V.2))
  crs(spp_coordinates) <- "+proj=longlat +datum=WGS84 +no_defs
+ellps=WGS84 +towgs84=0,0,0"

  # Now we model
  ESDM <- ensemble_modelling(c('GLM', 'ANN', 'SVM', 'MARS', 'RF'),
as.data.frame(spp_coordinates),
                             s3, rep = 5, metric = 'Kappa', Xcol =
'longitude', Ycol = 'latitude',
                             ensemble.thresh = 0, verbose = TRUE,
ensemble.metric = 'Kappa')

  # Project to future scenarios
  futA1B <- project(obj=ESDM, Env=s3)
  print("Projected A1B 100%")

  # Write the ouputs
  save.esdm(ESDM, name = as.character(active_modeling_species),
            path = "/Volumes/Seagate_JVidal/Mangrove_model/models/
Present/", verbose = TRUE, GUI = FALSE)
  save.esdm(futA1B, name = as.character(active_modeling_species),
            path = "/Volumes/Seagate_JVidal/Mangrove_model/models/
Future/", verbose = TRUE, GUI = FALSE)
}

################
# Load results #
################
# Subset the popultation dataset to get the coordinates of the sampled
points
geographic <- read.xlsx("/Users/jdvidal/Desktop/Desktop_2021/
Results_Frontiers/Final_figures/Supplementary_Table_1.xlsx", sheet=1)
p_samp_final_g <- geographic[geographic$species=="Avicennia
germinans",]
p_samp_final_s <- geographic[geographic$species=="Avicennia
schaueriana",]
p_samp_germ <- data.frame("ID"=as.factor(p_samp_final_g$pop),
```

```
"latitude"=as.numeric(p_samp_final_g$latitude),

"longitude"=as.numeric(p_samp_final_g$longitude))
p_samp_schau <-
data.frame("ID"=as.factor(gsub('[0-9]',"",p_samp_final_s$ID)),

"latitude"=as.numeric(p_samp_final_s$latitude),

"longitude"=as.numeric(p_samp_final_s$longitude))
p_samp_germ <- unique(p_samp_germ)
p_samp_schau <- unique(p_samp_schau)
coordinates(p_samp_germ) <- ~longitude+latitude
coordinates(p_samp_schau) <- ~longitude+latitude

#####################################################################
# Summarize suitability for each Avicennia germinans population #
#####################################################################
buffer_germ <- buffer(p_samp_germ, 50000, dissolve=F)
buffer_germ$ID <- p_samp_germ$ID

# Load results
model_bin_pres_germ <- raster('/Volumes/Seagate_JVidal/Mangrove_model/
models/Present/Avicennia germinans/Rasters/Binary.tif')
model_bin_B1_germ <- raster('/Volumes/Seagate_JVidal/Mangrove_model/
models/Future/Avicennia germinans/Rasters/Binary.tif')

# Plot to inspect
plot(model_bin_pres_germ)
plot(model_bin_B1_germ)
plot(buffer_germ)

# Crop the model results
model_bin_pres_germ <- crop(model_bin_pres_germ, buffer_germ)
model_bin_B1_germ <- crop(model_bin_B1_germ, buffer_germ)

# Convert raster to points and intersect
# Avicennia germinans
# Present
points_result_pres_g <- rasterToPoints(model_bin_pres_germ, spatial =
TRUE)
buffer_results_values_pg <- intersect(points_result_pres_g,
buffer_germ)
df_vals_germ_pg <- as.data.frame(buffer_results_values_pg)
df_vals_germ_pg
df_vals_germ_pg <- df_vals_germ_pg[,-c(3:4)]
# Future
points_result_B1_g <- rasterToPoints(model_bin_B1_germ, spatial =
TRUE)
buffer_results_values_B1g <- intersect(points_result_B1_g,
buffer_germ)
```

```
df_vals_germ_B1g <- as.data.frame(buffer_results_values_B1g)
df_vals_germ_B1g <- df_vals_germ_B1g[,-c(3:4)]

# Count the suitability per population
i <- table(df_vals_germ_pg)[2,]
j <- table(df_vals_germ_B1g)[2,]
res_g <- rbind(i,j)
rownames(res_g) <- c("Present", "B1")

#####################################################################
# Summarize suitability for each Avicennia schaueriana population #
#####################################################################
buffer_schau <- buffer(p_samp_schau, 50000, dissolve=F)
buffer_schau$ID <- p_samp_schau$ID

# Load results
model_bin_pres_schau <- raster('/Volumes/Seagate_JVidal/
Mangrove_model/models/Present/Avicennia schaueriana/Rasters/
Binary.tif')
model_bin_B1_schau <- raster('/Volumes/Seagate_JVidal/Mangrove_model/
models/Future/Avicennia schaueriana/Rasters/Binary.tif')

# Plot to inspect
plot(model_bin_pres_schau)
plot(model_bin_B1_schau)
plot(buffer_schau)

# Crop the model results
model_bin_pres_schau <- crop(model_bin_pres_schau, buffer_schau)
model_bin_B1_schau <- crop(model_bin_B1_schau, buffer_schau)

# Convert raster to points and intersect
# Present
points_result_pres_s <- rasterToPoints(model_bin_pres_schau, spatial =
TRUE)
buffer_results_values_ps <- intersect(points_result_pres_s,
buffer_schau)
df_vals_schau_ps <- as.data.frame(buffer_results_values_ps)
df_vals_schau_ps
df_vals_schau_ps <- df_vals_schau_ps[,-c(3:4)]
# Future
points_result_B1_s <- rasterToPoints(model_bin_B1_schau, spatial =
TRUE)
buffer_results_values_B1s <- intersect(points_result_B1_s,
buffer_schau)
df_vals_schau_B1s <- as.data.frame(buffer_results_values_B1s)
df_vals_schau_B1s <- df_vals_schau_B1s[,-c(3:4)]

# Count the suitability per population
i <- table(df_vals_schau_ps)[2,]
```

```
j <- table(df_vals_schau_B1s)[2,]
res_s <- rbind(i,j)
rownames(res_s) <- c("Present", "B1")

# Divides the future by the present count of suitable cells
res_s
res_g
var_s <- round((res_s[2,]-res_s[1,])/res_s[1,],4)
var_g <- round((res_g[2,]-res_g[1,])/res_g[1,],4)
var_s
var_g

results_modelling <-
rbind.data.frame(cbind(pop=rownames(as.data.frame(var_s)),var=as.data.
frame(var_s)[,1], species="Avicennia schaueriana"),

cbind(pop=rownames(as.data.frame(var_g)),var=as.data.frame(var_g)[,1],
species="Avicennia germinans"))
results_modelling$pop <- gsub("As", "", results_modelling$pop)
write.csv(results_modelling, "/Users/jdvidal/Desktop/Mangue_Review/
Results_December/Model_variation.csv")

# Calculate the regional deforestation rates for each population
library(raster)
results <- data.frame('year'=NA,'area'=NA, species=NA, population=NA)

# GLC rasters downloaded from GoogleEarthEngine
files <- list.files("/Volumes/Seagate_JVidal/Mangrove_forest_loss/
Mangrove_10k_yearloss/", full.names=TRUE)
for(i in 1:length(files)){
  print(strsplit(strsplit(files[i],"//")[[1]][2], "_")[[1]][1])
  r1 <- raster(files[i])
  freq(r1)
  focalAreaStats <- as.data.frame(cbind(freq(r1),

substr(strsplit(strsplit(files[i],"//")[[1]][2], "_")[[1]][1], start =
1, stop = 2),

substr(strsplit(strsplit(files[i],"//")[[1]][2], "_")[[1]][1], start =
3, stop = 6)))
  names(focalAreaStats) <- c("year","area","species","population")
  results <- rbind(results, focalAreaStats)
}
results <- results[results$year!=0,]
results <- results[-1,]
results$year <- as.numeric(as.character(results$year))
results$area <- as.numeric(as.character(results$area))
results
library(ggplot2)
ggplot(results, aes(x=year, y=area, color=population))+geom_point()
```

```
+geom_smooth(method='lm', formula= y~x)

# Original forest cover
files_oc <- list.files("/Volumes/Seagate_JVidal/Mangrove_forest_loss/
Mangrove_original_cover_10km/", full.names=TRUE)
results_oc <- data.frame('area'=NA, species=NA, population=NA)
i <- 1
for(i in 1:length(files_oc)){
  print(strsplit(strsplit(files_oc[i],"//")[[1]][2], "_")[[1]][1])
  r1_oc <- raster(files_oc[i])
  r1_oc_binary <- reclassify(r1_oc, c(0,80,0,80,100,1))
  plot(r1_oc_binary)
  focalAreaStats_oc <- as.data.frame(cbind(freq(r1_oc_binary)[2,2],

substr(strsplit(strsplit(files[i],"//")[[1]][2], "_")[[1]][1], start =
1, stop = 2),

substr(strsplit(strsplit(files[i],"//")[[1]][2], "_")[[1]][1], start =
3, stop = 6)))

  names(focalAreaStats_oc) <- c("area","species","population")
  results_oc <- rbind(results_oc, focalAreaStats_oc)
}
results_oc <- results_oc[-1,]
results_oc$sppop <- paste(results_oc$species, results_oc$population)
results$sppop <- paste(results$species, results$population)

results$cumulative_prop_loss <- NA
i <- 1
for(i in 1:nrow(results)){
  original_site_cover <-
as.numeric(as.character(results_oc[(results_oc$sppop==results[i,]
$sppop),]$area))
  original_site_cover
  print("original cover:")
  print(results_oc[(results_oc$sppop==results[i,]$sppop),]$sppop)
  print(results_oc[(results_oc$sppop==results[i,]$sppop),]$area)
  results[i,]$cumulative_prop_loss <- (original_site_cover-results[i,]
$area)/
as.numeric(as.character(results_oc[(results_oc$sppop==results[i,]
$sppop),]$area))
  print("loss:")
  print(results[i,]$cumulative_prop_loss)
}

write.csv(results, file="/Volumes/Seagate_JVidal/Mangrove_forest_loss/
loss_year.csv")
write.csv(results_oc, file="/Volumes/Seagate_JVidal/
Mangrove_forest_loss/original_cover.csv")
```

```
# Read results
library(ggplot2)
library(ggsci)

dataplot <- read.xlsx('/Users/jdvidal/Desktop/Mangue_Review/
Figures_final/Supplementary_Table_2.xlsx')
dataplot$`Area.remaining.(cell.count)` <-
as.numeric(as.character(dataplot$`Area.remaining.(cell.count)`))
dataplot$`Area.remaining.(%)` <-
as.numeric(as.character(dataplot$`Area.remaining.(%)`))/100

dataplot_s <- dataplot[dataplot$Species=="Avicennia germinans",]
head(dataplot_s)
p <- ggplot(data=dataplot_s, aes(x=Year, y=`Area.remaining.(%)`,
color=Population))+geom_point(data=dataplot_s, aes(color=Population))
+geom_smooth(method="glm", method.args=list(family="binomial"),
fullrange =TRUE, se=F, lty="dashed", lwd=0.25)
p+xlim(c(2000,2100))+ylim(c(0,1))+theme_classic()+scale_color_npg()
+xlab("Year")+ylab("Remaining forest cover")+
geom_vline(xintercept=2020, linetype="dashed", color = "red")
```