# Set the working directory

setwd("C:/Users/pc/Desktop/world")

library(ape)

library(betapart)

library(picante)

library(cluster)

library(dendextend)

library(kmed)

library(recluster)

library(raster)

library(phytools)

library(rgdal)

library(spatialEco)

library(phyloregion)

library(rgeos)

library(jpfxns)

library(dplyr)

library(rangeBuilder)

library(rangemap)

library(speciesgeocodeR)

library(doParallel)

library(parallel)

library(pbapply)

library(colorspace)

# Load the shapefile

tuscany <- readOGR(choose.files(), "world\_Mollweide")

crs(tuscany)<-CRS("+proj=moll")

plot(tuscany)

e <- extent(tuscany)

# coerce to a SpatialPolygons object

p <- as(e, 'SpatialPolygons')

crs(p)<-CRS("+proj=moll")

mask<- gDifference(p, tuscany)

load("C:\\Users\\pc\\Desktop\\world\\TRAQUEOS\_Corrected\_AngelinoPOWO\_v1.r")

w2<- p\_clean\_2

#rm(p\_clean\_2)

w2<-subset(w2, Correct\_POW\_dist=="TRUE")

w2<-subset(w2, Correct\_geographic=="TRUE")

tree<-read.tree("GBOTB\_extended\_TS.tre")

w3 <- data.frame(w2[w2$Resolved\_ACCEPTED %in% as.vector(tree$tip.label), ])

w3<- w3[,c(17, 8,7)]

names(w3)<-c("species", "decimallongitude", "decimallatitude")

w3$species<-as.character(w3$species)

w3$decimallongitude <- w3$decimallongitude +rnorm(n = length(w3$decimallongitude), mean = 0.0001, sd = 0.0001)

w3$decimallatitude <- w3$decimallatitude +rnorm(n = length(w3$decimallatitude), mean = 0.0001, sd = 0.0001)

w3$decimallongitude <- round(w3$decimallongitude, 4)

w3$decimallatitude <- round(w3$decimallatitude, 4)

w3<-subset(w3, decimallongitude >= -180)

w3<-subset(w3, decimallongitude <= 180)

w3<-subset(w3, decimallatitude >= -90)

w3<-subset(w3, decimallatitude <= 90)

w3<-unique(w3)

w3.orig<-w3

coordinates(w3.orig) <- cbind(w3.orig$decimallongitude, w3.orig$decimallatitude)

crs(w3.orig)<-CRS("+init=epsg:4326")

w3.orig<-spTransform(w3.orig, CRS("+proj=moll"))

w3.orig$decimallongitude<-coordinates(w3.orig)[,1]

w3.orig$decimallatitude<-coordinates(w3.orig)[,2]

#######

con<-data.frame(table(w3$species))

names(con)<-c("species", "n")

con<- con[order(con$species),]

#con<-subset(con, n>=3)

data<-split(w3, w3$species)

data<- data[as.vector(con$species)]

makeRange <- function(pts) {

pts <- pts[,c('decimallongitude', 'decimallatitude')]

pts <- pts[complete.cases(pts),]

pts <- pts[!duplicated(pts),]

if (nrow(pts) > 0) {

pts <- filterByProximity(pts, dist=0.5)

if (class(pts) == 'numeric') {

pts <- as.data.frame(matrix(pts, nrow=1, ncol=2))

}

pts <- data.frame(filterByProximity(pts, dist=0.5)) #0.5 km

}

if (nrow(pts) > 5) {

#alpha hull

res <- getDynamicAlphaHull(pts, fraction=0.95, partCount = 10, buff=7584, clipToCoast = FALSE)

}

if (nrow(pts) <= 5 & nrow(pts) >= 3) {

#convex hull

res <- list(gBuffer(gConvexHull(SpatialPoints(pts)), width=0.08), method='MCH')

}

if (nrow(pts) > 0 & nrow(pts) < 3) {

#buffered points

res <- list(gBuffer(SpatialPoints(pts), width=0.08), method='pointBuff1deg')

}

if (nrow(pts) == 0) {

res <- 'no points'

}

return(res)

}

makeRange2 <- function(pts) {

pts <- pts[,c('decimallongitude', 'decimallatitude')]

pts <- pts[complete.cases(pts),]

pts <- pts[!duplicated(pts),]

if (nrow(pts) > 0) {

pts <- filterByProximity(pts, dist=0.5)

if (class(pts) == 'numeric') {

pts <- as.data.frame(matrix(pts, nrow=1, ncol=2))

}

pts <- data.frame(filterByProximity(pts, dist=0.5)) #0.5 km

}

if (nrow(pts) > 0) {

#buffered points

res <- list(gBuffer(SpatialPoints(pts), width=0.08), method='pointBuff1deg')

}

if (nrow(pts) == 0) {

res <- 'no points'

}

return(res)

}

source("function\_ah2sp.txt")

cl<-makeCluster(6)

clusterEvalQ(cl, { library(rangeBuilder) })

clusterEvalQ(cl, { library(rgeos) })

clusterExport(cl, varlist=c("data"))

clusterExport(cl, varlist=c("makeRange"))

clusterExport(cl, varlist=c("ah2sp"))

clusterEvalQ(cl, { library(alphahull) })

clusterEvalQ(cl, { library(GeoRange) })

results<- parallel::parLapply(cl, data, function(i) {

tryCatch( {

makeRange(i)

}, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})

})

stopCluster(cl)

results3<-results

uniqueSp <- con$species

uniqueSp <- uniqueSp[complete.cases(uniqueSp)]

names(results3)<-uniqueSp

rr<- lapply(results3, function(x) x[[1]])

results.cleaned<-rr[!sapply(rr, is.null)]

x<- results.cleaned

uniqueSpN<- names(results.cleaned)

zz <- lapply(1:length(x), function(i) SpatialPolygonsDataFrame(x[[i]], data.frame(id= uniqueSpN[[i]]), match.ID = FALSE))

zzz<- list()

for (i in 1:length(zz)){

tryCatch({

out<- crop(zz[[i]], extent(-180, 180, -90, 90))

zzz[[i]] <- out

}, error=function(e){})

}

zzz.c<-zzz[!sapply(zzz, is.null)]

for (i in 1:length(zzz.c)){

tryCatch({

crs(zzz.c[[i]]) <- CRS("+init=epsg:4326")

}, error=function(e){})

}

ranges1 <- do.call("rbind", zzz.c)

ranges1$species<-ranges1$id

ranges1$id<-NULL

####

uniqueSp2 <- unique(ranges1$species)

w4 <- w3[w3$species %in% as.vector(tree$tip.label), ]

w4$species<-as.character(w4$species)

w4<- w4[!w4$species %in% uniqueSp2, ]

data2<-split(w4, w4$species)

uniqueSp <- sort(unique(w4$species))

data2<- data2[as.vector(uniqueSp)]

results4<- list()

for (i in 1:length(data2)){

tryCatch({

results4[[i]] <- makeRange2(data[[i]])

}, error=function(e){})

}

names(results4)<-uniqueSp

rr<- lapply(results4, function(x) x[[1]])

results.cleaned<-rr[!sapply(rr, is.null)]

x<- results.cleaned

uniqueSpN<- names(results.cleaned)

zz <- lapply(1:length(x), function(i) SpatialPolygonsDataFrame(x[[i]], data.frame(id= uniqueSpN[[i]]), match.ID = FALSE))

zzz.c<-zz[!sapply(zz, is.null)]

for (i in 1:length(zzz.c)){

tryCatch({

crs(zzz.c[[i]]) <- CRS("+init=epsg:4326")

}, error=function(e){})

}

ranges2 <- do.call("rbind", zzz.c)

ranges2$species<-ranges2$id

ranges2$id<-NULL

#####

####

ranges<-rbind(ranges1, ranges2)

rang<-spTransform(ranges, CRS("+proj=moll"))

pt1 <- polys2comm(rang, res = 200000.0, species = "species", trace=0)

plot\_swatch(pt1$poly\_shp, values=pt1$poly\_shp$richness, k=20, leg=10, border=NA)

pt<-pt1

sparse\_comm<-pt$comm\_dat

long<-sparse2long(sparse\_comm)

coord<-pt$poly\_shp@data

centroids <- coordinates(pt$poly\_shp)

cc<-cbind(coord, centroids)

names(cc)[4]<- "X"

names(cc)[5]<- "Y"

ccc<-merge(long, cc, by="grids")

w3 <- ccc[ccc$species %in% as.vector(tree$tip.label), ]

coordinates(w3) <- cbind(w3$lon, w3$lat)

crs(w3)<-CRS("+proj=moll")

#w3<-spTransform(w3, CRS("+init=epsg:4326"))

w2<-as.data.frame(w3)

w2<-w2[,c(2, 3,4)]

names(w2)<-c("species", "decimallongitude", "decimallatitude")

w2<-subset(w2, decimallatitude >-6312252)

w23<-rbind(w2, w3.orig)

pt2<-points2comm(dat=w23, mask=tuscany, res= 200000.0, lon= "decimallongitude", lat = "decimallatitude", species="species")

plot\_swatch(pt2$poly\_shp, values=pt2$poly\_shp$richness, k=20, leg=10, border=NA)