Tutorial: Phylogenetic Diversity and Endemism based on a phylogeny and Species Distribution Models

# Timings

Section times as follows, based on using coarse environmental data (10 arc minutes):

1. Package install (ca. 3 min)
2. Download and prepare environmental data (ca. 2 min)
3. Process species occurrence data (ca. 5 min)
4. Build SDMs (ca. 5 min)
5. Calculate Phylogenetic Diversity and Endemism (from the SDMs and a provided phylogeny)

# Other info

* If you are on a PC, directory paths need to be separated by back-slashes instead of forward-slashes
* Try not to resize windows/display panels while R is busy calculating/plotting - this tends to cause problems

# 1. Initial setup and package install

The following block of code will install all required R packages for this tutorial, you can copy and paste this in your R Studio and run it. All of this code works with the latest R version (4.1.0), and possibly older ones (e.g. 3.5.0), but I cannot guarantee it will work, so best to have R v.4.1.0.

# install all required R packages  
install.packages(c( 'devtools', 'sp', 'rgbif', 'readr', 'ape', 'phylobase', 'foreach', 'doParallel', 'ggplot2', 'sp', 'stringr', 'gdistance') )  
# This one installs from a github repository  
devtools::install\_github('jjvanderwal/SDMTools')

# 2. Script to prepare the environmental data

library(raster)  
library (rgdal)  
  
base.dir <- '/Users/cb76kecu/Dropbox (iDiv)/Micro-Macro\_course/2021/Tutorial/Part\_2\_PD\_PE/'  
setwd <- base.dir

Now we will create some directories where things will be saved

dir.create(paste0(base.dir))  
dir.create(paste0(base.dir,'env\_data/'))  
dir.create(paste0(base.dir,'env\_data/'))  
dir.create(paste0(base.dir,'env\_data/present\_climate'))  
dir.create(paste0(base.dir,'env\_data/future\_climate\_2070\_RCP\_8.5'))  
dir.create(paste0(base.dir,'sdm\_R/'))  
dir.create(paste0(base.dir,'sdm\_R/ensembles'))  
dir.create(paste0(base.dir,'sdm\_R/ensembles/asc'))  
dir.create(paste0(base.dir,'sdm\_R/GBIF\_data'))

download the data at 2.5 arc minute resolution globally

climate <- raster::getData('worldclim', var='bio', res=2.5)

Now we will clip the data to a specific extent (xmin,xmax,ymin,ymax)

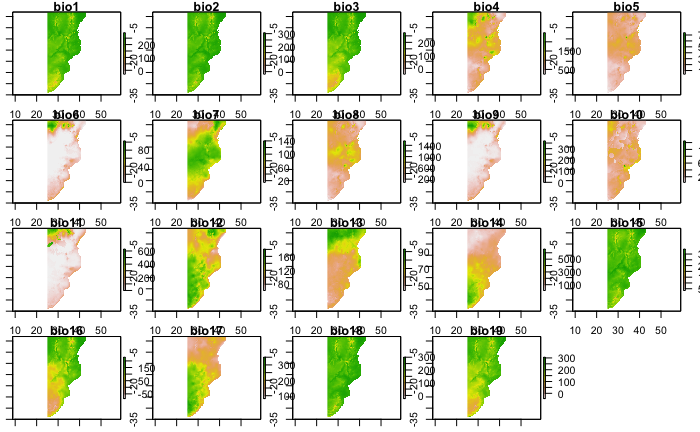
template <- extent(25, 43, -35, 2)  
climate.crop <- crop(climate, template, snap="out")

And save each of the bioclim variables as .asc grid files

for(i in 1:nlayers(climate.crop)){  
 writeRaster(climate.crop[[i]], paste0(base.dir, "env\_data/present\_climate/bioclim\_", i), "ascii", overwrite = T)  
 cat('Writing bioclim',i,   
 '... \n')  
}

Now load them from file and plot them, just to check they look ok

env\_data <- paste0(base.dir,'env\_data/present\_climate')   
lst <- list.files(path=env\_data,pattern='asc$',full.names = T)   
climate <- stack(lst)  
  
par(mar=c(1,1,1,1))  
par(mfrow=c(4,5))  
for(i in 1:nlayers(climate)){  
 plot(climate[[i]], main= paste0("bio",i))  
}



And delete temporary files

unlink('../../wc2-5/', recursive=TRUE)

# 3. Process species occurrence data

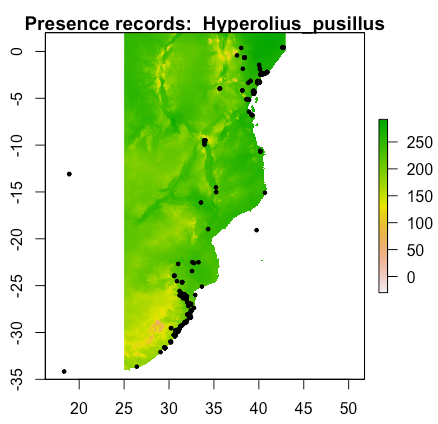
library(raster)  
library(rgbif)  
library(readr)  
library(dismo)  
library(rgbif)  
library(dplyr)  
  
base.dir <- '/Users/cb76kecu/Dropbox (iDiv)/Micro-Macro\_course/2021/Tutorial/Part\_2\_PD\_PE/'

Read in species csv files with presence data. You don’t need to do this here, but just so you know - it’s possible to provide rgbif a long list of species and save the data for each species, see here: <https://data-blog.gbif.org/post/downloading-long-species-lists-on-gbif/>

setwd(paste0(base.dir,'/sdm\_R/'))  
input.dir = "./GBIF\_data/per\_species/"  
setwd(input.dir)  
input\_files <-gsub("\\.csv$","", list.files(pattern="\\.csv$"))  
for(i in input\_files){  
 filepath <- file.path("./",paste(i,".csv",sep=""))  
 assign(i, read.csv(filepath, sep = ","))  
 cat("\n Reading",i,"csv file... done!")  
}

plot samples on a map

mar <- c(1,1,1,1)  
for(i in input\_files){  
 filepath <- file.path("./",paste(i,".csv",sep=""))  
 assign(i, read.csv(filepath, sep = ","))  
 sp <- read.csv(filepath)  
 sp <- sp[,c('decimalLongitude','decimalLatitude')]  
 sp$species <- 1  
 coordinates(sp) <- ~ decimalLongitude + decimalLatitude  
 par(mfrow=c(1,1))  
 plot(climate[[1]], main = paste("Presence records: ", i))  
 plot(sp, pch=21, cex=0.5, add=T)  
}



# 4. Build SDMs

library(sdm)  
library(raster)  
library(rgdal)  
library(sp)  
library(usdm)

# remove correlated predictor variables  
# read in all the data points  
sp <- read.csv('../../GBIF\_data/GBIF\_data.csv')  
# tell package which colums represent long and lat  
sp <- sp[,c('decimalLongitude','decimalLatitude')]  
# extract the predictor variable values from these points and put in a data frame  
spx <- extract(climate, sp)  
spx <- data.frame(spx)  
# measure variable inflation (similar to Pearson's correlation in sdm R package, and remove highly correlated variables)  
v <- vifstep(spx)  
v  
# exclude these highly correlated variables and overwrite the bio data (replacing the 19 bioclim vars with a subset of uncorrelated predictors)  
climate <- exclude(climate, v)  
climate

setwd(paste0(base.dir,'sdm\_R/ensembles/asc/'))  
  
library(sdm)  
  
wd <- getwd()  
par(mfrow=c(1,1))  
mar <- c(1,1,1,1)  
for(i in input\_files){  
 filepath <- file.path("../../GBIF\_data/per\_species/",paste(i,".csv",sep=""))  
 assign(i, read.csv(filepath, sep = ","))  
 sp <- read.csv(filepath)  
 sp <- sp[,c('decimalLongitude','decimalLatitude')]  
 sp$species <- 1  
 head(sp)  
 coordinates(sp) <- ~ decimalLongitude + decimalLatitude  
 d <- sdmData(species~., train=sp, predictors= climate, bg=list(n=1000))  
 methods <- c('maxlike','glm') # which methods required? - e.g. 'bioclim','bioclim.dismo','gam','glm','maxlike','rpart'  
 m <- sdm(species ~ . , d, methods=methods, replication='sub', test.p=30, n=1) # data partitioning/ test percentage?  
 ensemble\_model <- ensemble(m,climate, setting=list(id=1:2, method='weighted', stat='AUC', opt=2)) # how to combine models?  
 writeRaster(ensemble\_model, filename=paste0(wd,'/SDM\_',i,'.asc'), format="ascii", overwrite=TRUE)  
 plot(ensemble\_model, zlim=c(0,1),main = paste0("SDM: ", i), col=colorRampPalette(c('navy','lightyellow','orange','red'))(50))  
 lst1 <- list.files(path=getwd(),pattern='grd$',full.names = T)   
 lst2 <- list.files(path=getwd(),pattern='gri$',full.names = T)   
 file.remove(lst1)  
 file.remove(lst2)  
}

# 

# 5. Calculate SR/WE/PD/PE from these SDMs using a phylogeny

This code calculates richness and endemism from modelled suitability surfaces, it requires all the model grids to have the same extent

rm(list=ls())  
  
library(SDMTools)  
library(raster)  
library(ape)  
library(phylobase)  
library(foreach)  
library(doParallel)  
library(ggplot2)  
  
base.dir <- '/Users/cb76kecu/Dropbox (iDiv)/Micro-Macro\_course/2021/Tutorial/Part\_2\_PD\_PE/'  
phylo.dir <- (paste0(base.dir,'phylo/')) # modify to the base directory   
source(paste0(phylo.dir,'phylogenetic endemism.r'))

First define some functions

map\_raster = function(raster, output\_file, title) {  
 p <- rasterToPoints(raster)  
 p <- data.frame(p)  
 names(p) <- c("x", "y", "Model")  
 colour\_gradient <- scale\_fill\_gradientn(colours = rainbow(15), values=p$model)  
 colour\_gradient <- scale\_fill\_gradient2(low="white", mid="yellow", high="red",  
 limits=c(min(p$Model),max(p$Model)), midpoint=quantile(p$Model, 0.75), space='Lab')  
 m <- ggplot(data=p) + geom\_tile(aes(x, y, fill=Model)) + coord\_equal() + labs(x=NULL, y=NULL) + colour\_gradient  
 # delete a previous file if needed  
 if (file.exists(output\_file)) {  
 file.remove(output\_file)  
 cat("Previous", output\_file, "removed\n")  
 }  
 m <- m + ggtitle(title)  
 m <- m + theme(axis.title=element\_text(face="bold", size="18"))  
 m <- m + theme(axis.text=element\_text(face="bold", size="14"))  
 m <- m + theme(plot.title=element\_text(face="bold", size="24"))  
 m <- m + xlab("longitude") + ylab("latitude")  
 png(output\_file, width=image.width, height=image.height)  
 print(m)  
 dev.off()  
 m <- NULL  
}

Now some parameters

max.rows <- 10000000  
core\_count <- 4 # number of cores to use for parallel steps  
# size in pixels for maps  
image.width=1400  
image.height=1400

Define directories

phylo.dir <- (paste0(base.dir,'phylo/')) # modify to the base directory for your lineage modelling  
input.dir <- paste(base.dir, 'sdm\_R/ensembles/asc/', sep ='') # input lineage models  
output.dir <- paste(phylo.dir, 'output/', sep ='') # output location where diversity results and maps will be saved  
file\_pattern <- 'SDM\_' # modify this to a match the start of the name of all lineage model asc files  
  
group\_lin\_file <- paste(phylo.dir, 'All\_species\_list.csv', sep ='')

Tree details - this works for one genus at a time

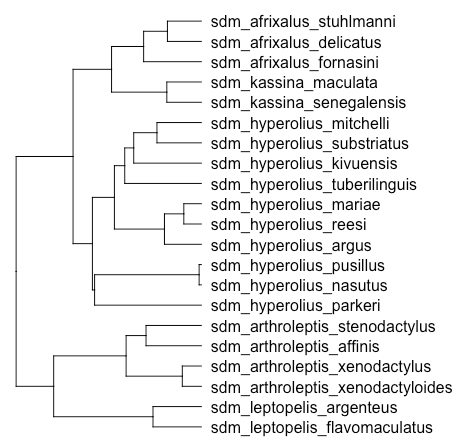
tree.file <- paste(phylo.dir, '/species\_tree.tre', sep ='')  
outgroup <- 'SDM\_scolecomorphus\_vittatus'  
preface <- ""  
genus <- ''  
output\_prefix <- 'SDMs\_'  
threshold <- 0.0000000000001 # this is not a species level threshold, but one used for each lineage model

Start Calculations

setwd(base.dir)  
files <- list.files(path = input.dir, pattern = file\_pattern, recursive = FALSE,ignore.case = TRUE, include.dirs = FALSE)  
setwd(input.dir)  
template.asc = read.asc(paste0(input.dir,'SDM\_Afrixalus\_delicatus.asc'))  
model\_rows=nrow(template.asc)  
model\_cols=ncol(template.asc)  
pos <- as.data.frame(which(is.finite(template.asc),arr.ind=TRUE)) #get all points that have data  
cat("\nLoading model rasters in parallel\n")  
cl <- makeCluster(core\_count)  
registerDoParallel(cl)  
pos\_par <- foreach (j=1:length(files), .combine=cbind, .packages='SDMTools') %dopar% {  
 tfile <- files[j]  
 pos\_temp <- pos  
 checkname = unlist(strsplit(tfile,".",fixed=T))  
 if (checkname[length(checkname)]=="asc") { # only accept filenames ending in .asc  
 cat(j)  
 tasc = read.asc(tfile) #read in the data  
 dataname=gsub(".asc",'',tfile)  
 newname <- tolower(gsub(preface, "", dataname))  
 cat("About to do pos\n")  
 pos\_temp[newname] <- tasc[cbind(pos\_temp$row, pos\_temp$col)]  
 pos\_temp[(which(pos\_temp[newname]< threshold)), newname] <- 0 # set values below the threshold to 0  
 pos\_temp[(which(is.na(pos\_temp[newname]))), newname] <- 0 # set the nulls to 0  
 cat("\n", j, newname, "loaded")  
 newcol <- data.frame(pos\_temp[, newname])  
 newcol <- round(newcol,3)  
 names(newcol) <- newname  
 newcol  
 }  
}  
  
pos <- cbind(pos,pos\_par)  
rm(pos\_par)  
setwd(base.dir)  
group\_lin\_list <-read.csv(group\_lin\_file, stringsAsFactors=F)

Read in the tree

mar=c(1,1,1,1)  
tree <- read.nexus(tree.file)  
plot(tree)  
tree <- keep.tip(tree, c('SDM\_afrixalus\_delicatus','SDM\_afrixalus\_fornasini','SDM\_afrixalus\_stuhlmanni','SDM\_arthroleptis\_affinis','SDM\_arthroleptis\_stenodactylus','SDM\_arthroleptis\_xenodactyloides','SDM\_arthroleptis\_xenodactylus','SDM\_hyperolius\_argus','SDM\_hyperolius\_kivuensis','SDM\_hyperolius\_mariae','SDM\_hyperolius\_mitchelli','SDM\_hyperolius\_nasutus','SDM\_hyperolius\_parkeri','SDM\_hyperolius\_pusillus','SDM\_hyperolius\_reesi','SDM\_hyperolius\_substriatus','SDM\_hyperolius\_tuberilinguis','SDM\_kassina\_maculata','SDM\_kassina\_senegalensis','SDM\_leptopelis\_argenteus','SDM\_leptopelis\_flavomaculatus'))  
tree <- phylo4(tree)  
labels(tree) <- tolower(labels(tree))  
plot(tree)

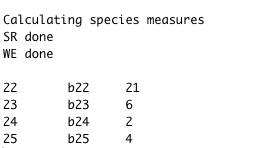


Ensure that the tree tips match the model names

model.names <- names(pos)  
model.names <- model.names[-(1:2)] # names of all columns except the 1st two which are row, col  
model.names <- tolower(gsub(preface,"",model.names))  
model.groups <- data.frame(model.groups=vector("character",nTips(tree)),stringsAsFactors=F)  
  
for (i in 1:nTips(tree)) {  
 cat(labels(tree)[i],"\n")  
 row <- group\_lin\_list[group\_lin\_list$lineage==labels(tree)[i],]  
 if (nrow(row) > 0) {  
 new\_tip\_name <- tolower(paste(row$genus, row$model\_group, row$lineage, sep="\_"))  
 labels(tree)[i] <- new\_tip\_name  
 model.groups[i,1] <- as.character(row$model\_group)  
 }  
}  
  
plot(tree)  
tree <- phylo4d(tree,tip.data=model.groups)  
tree  
tree.names <- as.character(labels(tree)[1:nTips(tree)]) ## original line  
matched.names <- intersect(model.names,tree.names)  
matched.tips <- which(labels(tree,"tip") %in% matched.names)  
cat("\nNot in tree names:",setdiff(model.names,tree.names),"\n")  
cat("\nNot in model names:",setdiff(tree.names,model.names),"\n")  
  
# a subtree containing only the tips for which there is a corresponding model  
subtree <- subset(tree,tips.include=matched.tips)  
subtree <- tree  
# limit the occurrence table to lineages which match the tree  
matching\_pos\_columns <- which(names(pos) %in% matched.names)  
matching\_pos\_columns <- unique(c(1, 2, matching\_pos\_columns)) # ensure that row and column are included  
pos <- pos[, matching\_pos\_columns]  
cat("\n\nRemoving unoccupied cells\n")  
cat("Before:",nrow(pos),"\n")  
rowsums <- apply(pos[,3:ncol(pos)],1,sum,na.rm=T)  
pos <- pos[which(rowsums>0),]  
rm(rowsums)  
cat("After:",nrow(pos),"\n")  
max.rows <- min(max.rows,nrow(pos))

Perform the calculations

result <- calc\_PE\_from\_models(subtree,pos[1:max.rows,which(names(pos) %in% matched.names)], core\_count = core\_count)  
cat("\nDiversity calculations completed. Now writing outputs to file.")  
pos\_output <- cbind(pos[1:max.rows,1:2],result)  
pos\_output <- pos\_output[,-3] # omit the site column



Plot and make output files

cellsize <- attr(template.asc,"cellsize")  
ymin <- attr(template.asc,"yll")  
xmin <- attr(template.asc,"xll")  
  
x <- ((pos\_output$row - 1) \* cellsize) + xmin  
y <- ((pos\_output$col - 1) \* cellsize) + ymin  
pos\_output <- cbind(pos\_output[,1:2],x,y,pos\_output[,-(1:2)])  
  
filenames <- c(paste(output\_prefix,"PE.asc",sep=""),paste(output\_prefix,"PD.asc",sep=""),paste(output\_prefix,"WE.asc",sep=""),paste(output\_prefix,"SR.asc",sep=""))  
dataframe2asc(pos\_output[,c(4,3,5:8)],filenames,output.dir)  
  
stopCluster(cl)  
setwd(output.dir)

Save output images

PE.ras <- raster(filenames[1])  
map\_filename <- paste(output\_prefix, "PE.png", sep="")  
map\_raster(PE.ras, map\_filename, paste(output\_prefix, "PE"))  
  
PD.ras <- raster(filenames[2])  
map\_filename <- paste(output\_prefix, "PD.png", sep="")  
map\_raster(PD.ras, map\_filename, paste(output\_prefix, "PD"))  
  
WE.ras <- raster(filenames[3])  
map\_filename <- paste(output\_prefix, "WE.png", sep="")  
map\_raster(WE.ras, map\_filename, paste(output\_prefix, "WE"))  
  
SR.ras <- raster(filenames[4])  
map\_filename <- paste(output\_prefix, "SR.png", sep="")  
map\_raster(SR.ras, map\_filename, paste(output\_prefix, "SR"))  
dev.off()  
  
par(mfrow=c(2,2))  
mar=c(1,1,1,1)  
colour\_scale <- colorRampPalette(c('lightyellow','orange','red','darkred'))(50)  
plot(SR.ras, main="Species Richness", col=colour\_scale)  
plot(WE.ras, main="Weighted Endemism", col=colour\_scale)  
plot(PD.ras, main="Phylogenetic Diversity", col=colour\_scale)  
plot(PE.ras, main="Phylogenetic Endemism", col=colour\_scale)

# 

# Bonus tutorial assignment

You can use the code above as a basis to do the following things: 1. Rerun the SDMs but do it properly - we did a quick and dirty analysis here but I’d like you to try and make some better ensembles based on what you learned in Tutorial 1 2. Redo the code above but project your models into the future to see how much your Phylogenetic Indices (Phylogenetic Diversity and Endemism) change