

# JMPH\_basic\_model\_proofs

Prepare the workspace.

```
knitr::opts_chunk$set(warning=FALSE, message=FALSE, error = FALSE) # rmd options
rm(list = ls()); invisible(gc()) # cleaning
```

Control Block

```
# f, f, 0.25 looks nice.
allowmigrants <- F # OPTIONS: T, F
allowsympatry <- F # OPTIONS: T, F
minoverperc <- 0 # remove pairs that do not have elevational overlap
costvar <- "ele"
```

Packages, Prefs, and Functions

```
options(scipen = 999) # turn off scientific notation
'%notin%' <- Negate('%in%')
require(ggplot2) # load packages
require(GGally)
require(viridis)
require(caper)
library(dplyr)
library(stringr)
library(maps)
library(ape)
library(EnvStats)
library(forecast)
require(nlme)
require(geodist)
require(letsR)
require(spdep)
require(spatialreg)
require(rnaturalearth)
require(rnaturalearthdata)
require(rgeos)
require(sf)
require(rgdal)
require(raster)
world <- ne_coastline(scale = "medium", returnclass = "sf")
```

Load main data

```
# main dataframe -----
setwd("~/Box Sync/CB_VF_Shared/Dry_Lab/Projects/JMPH/Process_Cluster_Outputs/Data")
load(file = "Pair_Barrier_Data_FEB2021.rdata")
mydata <- mypairedata; rm(mypairedata)
rownames(mydata) <- mydata$Species.1
mydatahold <- mydata
```

Impose masks & do calculations

```

mydata$cost <- mydata[, paste0(costvar, "_c25")]
mydata <- mydata[!is.na(mydata$cost),]
mydata$lcost <- log( mydata$cost + abs(min( mydata$cost , na.rm = T)) + 1)

mydata$ele_overlap <- mydata[,paste0(costvar, "_ov_perc_smrnge")]
mydata <- mydata[mydata$ele_overlap > minoverperc,]

if(allowmigrants == F){
  mydata <- mydata[which(mydata$Migration == 1.0),]
} # retain only non-migrants if asked to do so.

if(allowsympatry == F){
  mydata <- mydata[which(mydata[,paste0(costvar, "_c0")] > 0 ),]
} # retain only allopatric/parapatric species if asked to do so.

mydata$lat <- mydata[,paste0("lat_mean_pair_", costvar, "_c25")]
mydata$lon <- mydata[,paste0("lon_mean_pair_", costvar, "_c25")]
mydata$thermal_niche_breadth <- mydata$MAT_range_pair_mean # mydata[,paste0(costvar, "_range_pair_mean")]

mydata <- mydata[,c("uniquePairId", "Species.1", "Species.2", "Species.1b1", "Species.2b1",
  "thermal_niche_breadth", "lcost", "lat", "lon")]

dism <- mydata[,c("lon", "lat")]
dism <- geodist(dism, measure = "geodesic")
rownames(dism) <- rownames(mydata)
colnames(dism) <- rownames(mydata)
basecols <- ncol(mydata)

# neighbors
coords<-cbind(mydata$lon, mydata$lat); coords<-as.matrix(coords) ; row.names(coords)<-rownames(mydata)
k1 <- knn2nb(knearneigh(coords, longlat = T))
nb<- dnearneigh(coords,row.names = row.names(coords), d1=0,d2=max(unlist(nbdists(k1, coords, longlat = T)))


```

## load phylo and prune

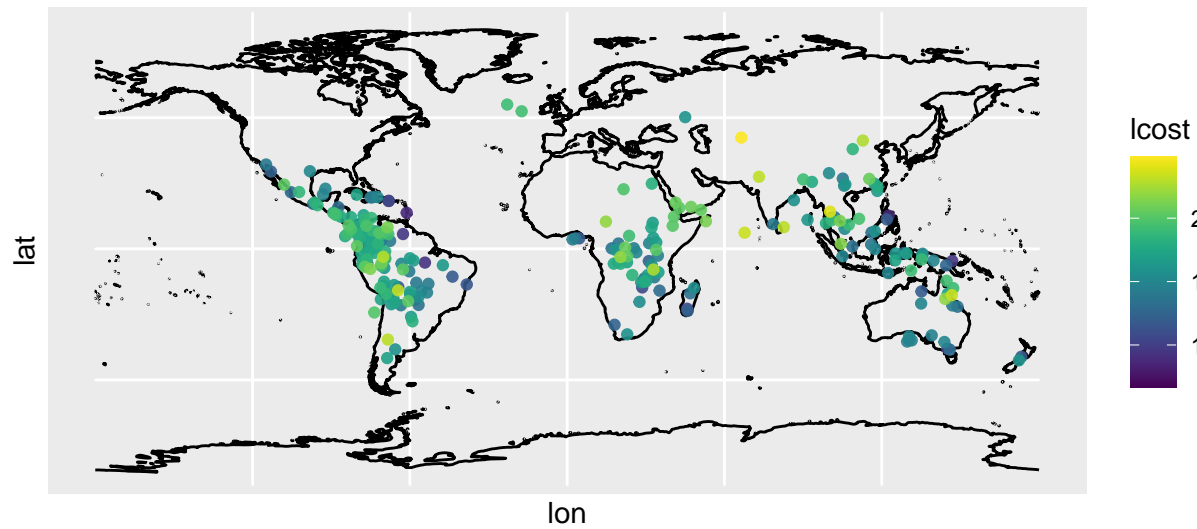
```

# phylo -----
setwd("~/Box Sync/CB_VF_Shared/Dry_Lab/Projects/JMPH/Other_Input_Data/BirdTrees")
load(file = "BirdTrees.Rdata")
tree <- trees[[1]]; rm(trees) # pick tree (VF GET TREES FROM COONEY!!! and use MCC tree.)
tree <- drop.tip(tree, tree$tip.label[which(tree$tip.label %notin% mydata$Species.1)]) # initial name m
mydata <- mydata[which(mydata$Species.1 %in% tree$tip.label),]
mydata <- mydata[match(tree$tip.label, mydata$Species.1),]
sum(mydata$Species.1 != tree$tip.label)

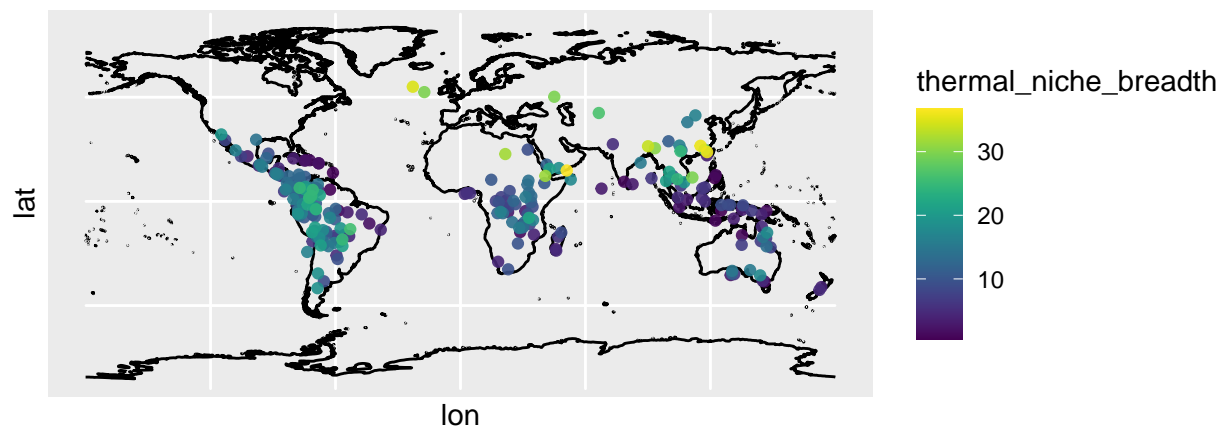
## [1] 0

ggplot(world)+
  geom_sf() +
  geom_point(data = mydata[order(mydata$lcost),], aes(y=lat, x=lon, color = lcost), alpha = 0.9) +
  scale_color_viridis()

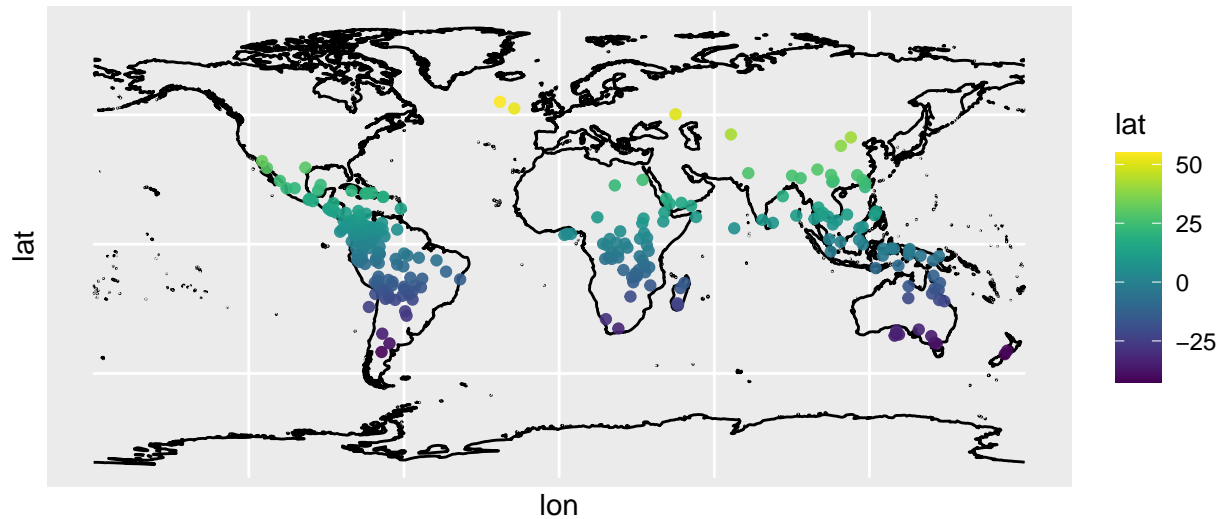
```



```
ggplot(world)+
  geom_sf() +
  geom_point(data = mydata[order(mydata$thermal_niche_breadth),], aes(y=lat, x=lon, color = thermal_niche_breadth)) +
  scale_color_viridis()
```



```
ggplot(world)+
  geom_sf() +
  geom_point(data = mydata[order(mydata$lat),], aes(y=lat, x=lon, color = lat), alpha = 0.9) +
  scale_color_viridis()
```



CB models

```
# elevational barrier size related to thermal niche breadth??? YES
m <- gls(scale(lcost) ~ scale(thermal_niche_breadth), correlation = corPagel(0.99, phy = tree, fixed = 1)

## Generalized least squares fit by REML
## Model: scale(lcost) ~ scale(thermal_niche_breadth)
## Data: mydata
##      AIC      BIC    logLik
## 706.2155 720.3962 -349.1077
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
##      lambda
## -0.02386435
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)    -0.0692864 0.00441174  -15.705012     0
## scale(thermal_niche_breadth) 0.3824313 0.05765592   6.632993     0
##
## Correlation:
##              (Intr)
## scale(thermal_niche_breadth) 0.444
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -2.76691113 -0.62545330 -0.01783909  0.72322614  2.88982479
##
## Residual standard error: 0.9228722
## Degrees of freedom: 258 total; 256 residual

print(paste("Correlation between data and prediction: ", cor(predict(m), scale(mydata$lcost))))

## [1] "Correlation between data and prediction: 0.375630278893303"

# spatial autocorr -----
matx <- as.matrix(m$residuals); rownames(matx) <- rownames(mydata)
```

```

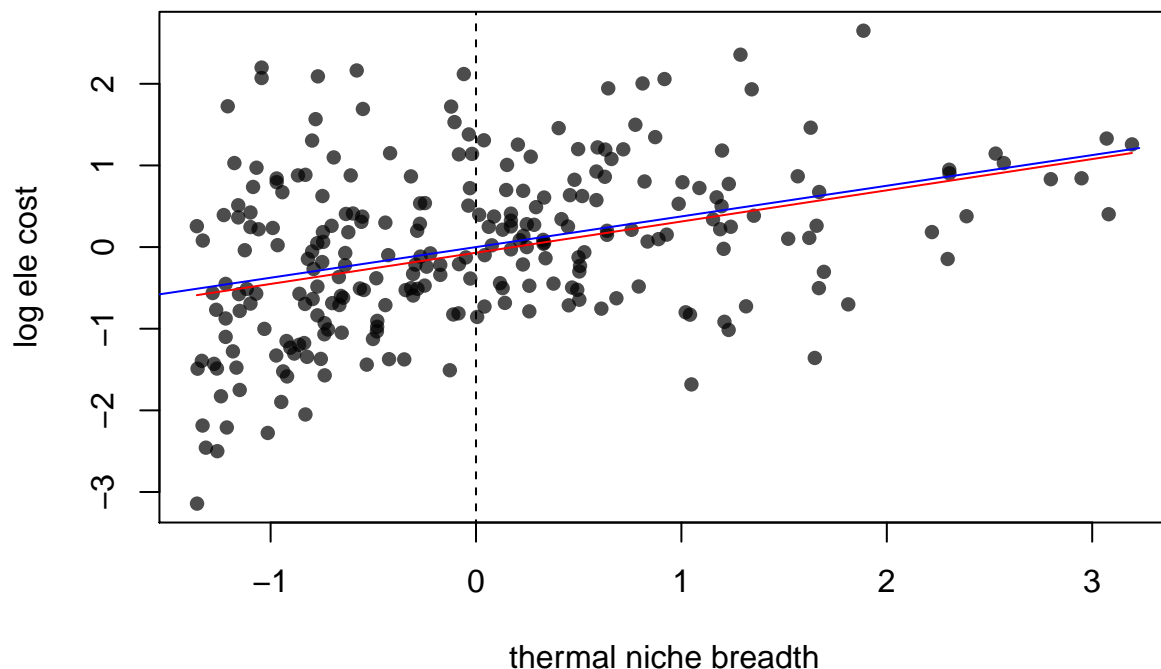
spac <- lets.correl(x=matx, y=distm, z=12, equidistant = T, plot = F)
print(paste("moran's I: ", moran.test(residuals(m), nb2listw(nb))$p.value)) # no evidence for spatial a

## [1] "moran's I: 0.605896562020738"

# plot -----
plot(scale(mydata$thermal_niche_breadth), scale(mydata$lcost), pch = 16, col = rgb(0,0,0,0.7),
      xlab = "thermal niche breadth", ylab = paste0("log ", costvar, " cost" ), main = "Global model")
myx <- seq(min(scale(mydata$thermal_niche_breadth)), max(scale(mydata$thermal_niche_breadth)), length.out=100)
mycoefs<-coef(m)
myy <- mycoefs[1] + mycoefs[2]*myx
lines(c(0,0), c(-10,100), lty=2)
lines(myx,myy,col = 'red')
m <- gls(scale(lcost) ~ scale(thermal_niche_breadth), data = mydata, method = "REML")
myx <- seq(min(scale(mydata$lat)), max(scale(mydata$lat)), length.out=100) # plot fit
mycoefs<-coef(m)
myy <- mycoefs[1] + mycoefs[2]*myx
lines(c(0,0), c(-10,100), lty=2)
lines(myx,myy,col = 'blue')

```

### Global model



```

m <- gls(scale(thermal_niche_breadth) ~ scale(lat), correlation = corPagel(0.99, phy = tree, fixed = F)

## Generalized least squares fit by REML
## Model: scale(thermal_niche_breadth) ~ scale(lat)
## Data: mydata
##      AIC      BIC    logLik
## 729.5677 743.7484 -360.7838
##
## Correlation Structure: corPagel
## Formula: ~1

```

```
## Parameter estimate(s):
##   lambda
## 0.1094065
##
## Coefficients:
##               Value Std.Error t-value p-value
## (Intercept) 0.09844786 0.15470862 0.636344 0.5251
## scale(lat) 0.25683445 0.06322032 4.062530 0.0001
##
## Correlation:
##      (Intr)
## scale(lat) 0.01
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7444582 -0.8684905 -0.2067069 0.4907781 2.8961840
##
## Residual standard error: 0.992048
## Degrees of freedom: 258 total; 256 residual

print(paste("Correlation between data and prediction: ", cor(predict(m),scale(mydata$thermal_niche_breadth))

## [1] "Correlation between data and prediction: 0.240571679990748"

# spatial autocorr -----
matx <- as.matrix(m$residuals); rownames(matx) <- rownames(mydata)
spac <- lets.correl(x=matx, y=distm, z=12, equidistant = T, plot = F)
print(paste("morans I: ", moran.test(residuals(m), nb2listw(nb))$p.value)) # no evidence for spatial autocorrelation

## [1] "morans I: 0.895449858098874"

# plot -----
plot(scale(mydata$lat), scale(mydata$thermal_niche_breadth), pch = 16, col = rgb(0,0,0,0.7),
      xlab = "latitude", ylab = "thermal niche breadth", main = "Global model")
myx <- seq(min(scale(mydata$lat)), max(scale(mydata$lat)), length.out=100) # plot fit
mycoefs<-coef(m)
myy <- mycoefs[1] + mycoefs[2]*myx
lines(c(0,0), c(-10,100), lty=2)
lines(myx,myy,col = 'red')
m <- gls(scale(thermal_niche_breadth) ~ scale(lat), data = mydata, method = "REML")
myx <- seq(min(scale(mydata$lat)), max(scale(mydata$lat)), length.out=100) # plot fit
mycoefs<-coef(m)
myy <- mycoefs[1] + mycoefs[2]*myx
lines(c(0,0), c(-10,100), lty=2)
lines(myx,myy,col = 'blue')
```

# Global model

