

CAROLINA MONTEALEGRE TALERO

Bióloga pela Universidad de los Andes de Bogotá, Colombia e mestra em Conservação de Biodiversidade pela Universidad Javeriana de Bogotá, Colombia. Atualmente sou doutoranda em Ecologia do Instituto de Biociências da USP, orientada pelo Dr. Jean Paul Metzger. Meu interesse de pesquisa é a ecologia da paisagem, especificamente o processo de “spillover” das aves entre os remanescentes de Mata Atlântica e as matrizes de pasto e café.

Link ao currículo Lattes:

<http://lattes.cnpq.br/3419494289062418>



Exercícios de aula

[Exercícios Carolina Montealegre.](#)

PROPOSTA DO TRABALHO FINAL

PLANO A

Analysis of spatial data is crucial for decision-making in diverse contexts in conservation and planning. In the case of habitat use, researchers need to know if points (coordinates) taken in the field are particularly associated to a habitat or land cover, or if the individuals are moving randomly.

My function will analyse the spatial distribution of individuals, in the form of x and y coordinates, and compare them to randomly distributed points. Specifically, it will compare the observed distribution to a vector with random values that follow a uniform distribution, for both x and y values.

The input arguments for the function will be a data frame containing the coordinates (x,y) of the monitored individual.

Then, randomly distributed samples will be generated (1000), with the same number of points (N) as the observed sample. For each of these samples, the value of the *Mean Nearest Neighbor* (MNN) or other statistical test will be calculated, as well as for the observed sample. With these values, a probability function will be constructed.

The output will be the probability that the observed MNN value is obtained by a randomly distributed sample.

This is an interesting concept, but it feels very limited. It is just a comparison between a simple statistic in a data set and a null hypothesis, that can be interesting or not depending on the question of interest. If you added more diagnostics I think it could be more useful.

—Ogro

PLANO B

The alternative, is a function that graphs home ranges, based on the input data that will be a data frame of x and y coordinates. The output will be Minimum Convex Polygons, the most basic form of a home range.

This one feels much more interesting from a programming standpoint, and could easily be combined with the first plan.

I would suggest doing both, but focusing on the graphical part.

—Ogro

Concordo com o Diogo, faça ambos. Ou seja o teste de que a distribuição não é aleatória e o gráfico Minimum Convex Polygons. — [Alexandre Adalardo de Oliveira](#) 2016/04/29 11:13

TRABALHO FINAL

Código da Função

```
HR.MNNprob <- function (file_name, Nrep) #head of the function
{
  coord <- read.table(file_name, header=T) #creates an object called "coord"
  that contains the data, in form of x and y coordinates
  str(coord) #shows the structure of the data, lets us make sure that the
  object was properly created
  #####
  ## CALCULATING THE HOME RANGES ###
  #To call the installed packages
  library (maptools)
  library (sp)
  library (rgdal),
  library (adehabitatHR)
  library (stringr)
  library (rgeos)
```

```

library (gpclib)
class(coord) # checks data frame's class. It should be a data.frame and we
need to convert it into a SpatialPointsDataFrame
coordinates(coord) <- c("xc", "yc") #converts the data.frame into a
SpatialPointsDataFrame, by reading the x and y coordinates
class(coord) #to check the class again
summary(coord) # to check the basic statistics for this object
coord$ind <- as.factor(coord$ind) # the output from summary says that R
thinks that the animal ID column is a continuous variable. However, it is a
categorical variable and this can be fixed with the as.factor command
plot(coord, col=coord$ind) # Plot data points for the individuals
# to calculate minimum convex polygons (MCP)
coord.mcp <- mcp(coord, percent = 100) # MCP home ranges are very
susceptible to over estimate home range area - because of outliers. By
changing this percent, the user can choose the extent of the MCP he or she
wants to graph.
plot(coord.mcp, col=2:6)
plot(coord, col= coord$ind, add = TRUE)
coord.mcp #quick look at some polygon's characteristics, including area
#####
## CALCULATING MEAN NEAREST NEIGHBOR P VALUE ##
n.rows = nrow(coord) #gives the number of rows in the data set, that will
be used later
dista=matrix(NA, ncol=n.rows, nrow=n.rows) #matrix of NAs that later will
contain the distance between each individual
for(i in 1:(n.rows-1)) #calculates the observed distance between each
individual and places it inside the object "dista"
{
  for(j in (i+1):n.rows) #loop over individuals
  {
    difx2=(coord$xc[i] - coord$xc[j])^2 #calculates square distance in X
direction
    dify2=(coord$yc[i] - coord$yc[j])^2 #calculates square distance in Y
direction
    dista[i,j] <- sqrt(difx2 + dify2) #calculates distance between i and j
    dista[j,i] <- sqrt(difx2 + dify2) #calculates distance between j and i
  }
}
dista #calls the object "dista"
(nn<-apply(dista, 1, min, na.rm=TRUE)) #calculates the minimum distance
values for each individuals
(MNN<-mean(nn)) #calculates the mean nearest neighbor (MNN) value
minxc = min(coord$xc, na.rm=TRUE) #defines the minumum value in the x
position
maxxc = max(coord$xc, na.rm=TRUE) #defines the maximum value in the x
position
minyc = min(coord$yc, na.rm=TRUE) #defines the minumum value in the y
position
maxyc = max(coord$yc, na.rm=TRUE) #defines the maximum value in the y
position
## Making a simulation

```

```
simula <- rep(NA, times=Nrep) #creates a vector with any number of NAs to
later store the values of the simulation
simula[1] <- MNN #stores the calculated value of MNN in the first position
of the vector "simula"
con = 0 #starts the counter for the number of simulations that return a
MNN lower than the observed value
for(k in 2:Nrep) #creates a counting cycle that:
{
  xsim <- runif(n.rows, min= minxc, max= maxx) #creates a vector with
random values from the minimum to the maximum sampled value in the x axis
  ysim <- runif(n.rows, min= minyc, max= maxyc) #creates a vector with
random values from the minimum to the maximum sampled value in the y axis
  dista.2 <- matrix(NA, ncol=n.rows, nrow=n.rows) #creates a matrix that
stores the distance values for each simulation
  for(i in 1:(n.rows-1)) #loop over individuals of the simulated value
  {
    for(j in (i+1):n.rows) #loop over individuals of the simulated value
    {
      difx2 = (xsim[i] - xsim[j])^2 #calculates square distance in X
direction
      dify2 = (ysim[i] - ysim[j])^2 #calculates square distance in Y
direction
      dista.2[i,j]<-sqrt(difx2 + dify2) #calculates distance between i and
j
      dista.2[j,i]<-sqrt(difx2 + dify2) #calculates distance between j and
i
    }
  }
  nn2<-apply(dista.2, 1, min, na.rm=TRUE) #finds the minimum distance for
each individual in relation to the rest
  MNN2<-mean(nn2) #calculates the mean nearest neighbor values for each
simulation
  simula[k]<- MNN2 #object (vector) that contains all the MNN values from
the simulation
  if(MNN2 <= MNN) con = con+ 1 #condicional: counts the number of times
that a simulation returns a MNN value less or equal than the observed value
}
  ProbMNN = con/Nrep #calculates the probability that the value of MNN
obtained from a simulation is less or equal than the observed value
  return(ProbMNN) #returns the value of the MNN probability.
}
```

Help

ProbMNN

package:various

R Documentation

Function to calculate home ranges of individual animals and the probability that the value of the Mean Nearest Neighbor (MNN) obtained from random

samples is less or equal than the MNN of an observed sample.

Description:

First, the function computes the home range of individual animals using the Minimum Convex Polygon (mcp) estimator. Second, this function calculates the MNN_observed value for the data frame. Then, it simulates N random samples and computes the MNN values for each one (MNN_simulated). Finally, using the distribution of these MNN_simulated values, it computes the probability that the simulated MNN value is less or equal than the MNN value of the observed sample.

Usage:

```
HR.MNNprob(file_name, Nrep)
```

Arguments:

file_name	name of the input file
Nrep	number of repetitions for the simulation

Details:

- The function requires the packages "maptools", "sp", "rgdal", "adehabitatHR", "stringr", "rgeos" and "gpclib"
- The input must be a three column table, where the 1st column must be the individual ID with a header called "ind", the 2nd and 3rd columns should be named "xc" and "yc" and contain the coordinates to compute the MNN.
- If the number of repetitions for the simulations is too high (more than 10000) it will take a considerable time to run.
- This function calculates the value of the probability $P(\text{MNN_Simulated} \leq \text{MNN_Observed}) = \text{ProbMNN}$. Therefore, if the user requires $P(\text{MNN_Simulated} \geq \text{MNN_Observed})$, he or she will have to calculate $(1 - \text{ProbMNN})$.

Value:

MNN Probability value

Warning:

It should be confirmed that all the packages are compatible and loaded correctly before executing this function.

Author:

Carolina Montealegre Talero
e-mail: biocaro@gmail.com

Examples:

```
HR.ProbMNN("unicornio-hr.txt", 1000)  
HR.ProbMNN("datos-mov-aves-hr.txt", 10000)
```

Files to Execute the Function

[dados-mov-aves-hr.txt](#)

[unicornio-hr.txt](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:biocar:start 

Last update: **2020/08/12 06:04**