

Sergio Bolívar



Biologo formado na Colômbia, estudante de mestrado do Museu de Zoologia da USP (MZUSP), tenho interesse em questões Sistemáticas e biogeográficas da região Neotropical.

Atualmente, realizo uma revisão taxonômica do arapaçu do Cerrado, *Lepidocolaptes angustirostris* (Dendrocolaptinae) usando informação da plumagem e morfologia.

<http://lattes.cnpq.br/5243462035267816>

[exec](#)

email contact: bolivarmz@usp.br ou bolivaruis@gmail.com

Proposta Trabalho Final AJUSTADA!

Proposta 1

Objetivo principal da função: Estabelecer correlações entre grupos e/ou indivíduos usando informação geográfica e biológica (características dos indivíduos).

A ideia é testar se um indivíduo(s) é parte de um grupo X ou Y, primeiro baseado em informação geográfica, e depois comparando com os resultados dos análises estatísticas usados para identificar diferenças entre grupos.

A ideia pode ser útil para taxonomistas que usam informação morfológica e estudam espécies híbridas.

Input:

O input 1: Uma tabela com as informações geográficas (coordenadas) e biológicas (medições morfométricas, etc) de cada indivíduo. O input 2: Uma lista com as coordenadas das localidades padrão.

Paso 1: Distâncias geográficas e diferença de médias.

a. Atribua a cada indivíduo em nosso input 1 um "localidade padrão" específica, com base na distância geográfica entre o indivíduo e cada ponto padrão (localidade padrão, 'maptools', 'rgdal').

b. Criar em R os grupos baseados no análise de distância geográfica (comandos 'data.frame', 'list', 'cbind').

c. Calcular a média e o desvio padrão para os indivíduos/grupos para cada caractere ou medições (dado, com comandos 'mean' e 'sd').

d. Calcular para cada indivíduo as diferenças de cada um de seus dados com as médias dos grupos geográficos mais próximos a ele (= média do grupo - valor individual).

e. Criar em R os novos grupos baseados no analisis de diferencias de medias (comandos 'data.frame', 'list', 'cbind').

Paso 2: Resultados, plotear cada um dos grupos ('ggplot2', 'biplot', 'maptools').

f. Plotear los grupos producidos pelo analisis de distancia geografica.

g. Plotear los grupos producidos nos resultados de diferencias nas medias.

h. Num plot só, comparar a distribuição e a estrutura de cada grupo como resultado dos analisis.

Proposta 2

Objetivo principal da função: Calcular a riqueza e abundancia de um set de dados usando metodos GIS em R (pacotes que podem interpretar mapas e informação GIS).

Ideia util para realizar analisis GIS em R, quando R esta instalado em Sistemas Operativos (O.S.) que não tenham muitos programas GIS (caso das distribuições de linux).

Input:

Input 1: Uma tabela com as informações geograficas (coordenadas) e biologicas (medições morfometricas, etc) de cada individuo/grupo.

a. Mapear (plotear) os individuos usando suas distribuições geograficas com o 'maptools'.

b. Usando pacotes de gestão espacial R ('rgdal', 'readOGR'), dividir a area de estudo em quadrantes de diferentes escalas geograficas (similar aos analisis feitos no programa Diva-Gis).

b. Calcular a abundância e riqueza das areas usando a presença/ausência de indivíduos e dos caracteres por quadrante (usando os indices de Shannon e de Riqueza Especifica = numero de especies 'i'/numero total de espececies num quadrante especifico).

c. O resultado é um mapa com as areas com maior abundancia e riqueza ('maptools').

Woow... As propostas ficaram muito claras! Fique à vontade para seguir com qualquer uma delas. — [Leonardo](#)

Codigo e arquivo HELP da função geocalInfor (Proposta A), postado 18 Maio 2014

TRABALHO FINAL

Função geocalInfor: Uma função simples que realiza dois tipos análises principais:

1. Identifica unidade geográfica padrão mais próxima às coordenadas estudadas, calculando as distâncias em Kms entre a base de dados de estudo e as unidades padrão.
2. Usando os valores amostrados de um caracter biológico (uma medição de algum rasgo dos individuos estudados) e as médias para esse caracter (mean) para cada *grupo geográfico* da analise (1); são calculadas as diferencias das medias desse caracter (media de cada grupo identificado - o valor de cada individuo para esse caracter), e identifica o 'grupo geográfico' que tenha a minima diferencia na média, criando um novo set de grupos.

Finalmente, a função realiza um plot do mapa da area estudada (o shapefile deve ser derfinido pelo usuario) com os pontos de cada grupo identificado nos analises (1) e (2) para sua comparação.

A formula para calcular as distancias geograficas corresponde a **Spherical law of cosines** (Lei Esférica dos Cossenos).

Função geocalInfor

```
geocalInfor = function(x,y,rmNA=TRUE,map=TRUE,save.map=FALSE) # inicio da
função
{
  if(!exists("loc_pattern")) # pergunta se existe o input de nome:
loc_pattern
{
  stop("The input loc_pattern is absent") # se não existe o "loc_pattern",
  produz erro na função
}
  if(rmNA==TRUE) # se existem NA's no dataset
{
  data=(na.omit(x)) # omitir os NA's do analise
}
else # se não tem NA's, usar todos ops dados.
{
  data=x # nossos dados X re-nomeados a "data"
}
  rownames(data)=data[,1] # os nomes das filas dos dados são os nomes de
cada individuo do dataset
  data=data[2:4] # selecciona todos os datos menos a coluna com os nomes de
cada individuo do dataset
  R = 6371.009 # radio da terra.
  data$longR=data$Longitude*pi/180 # calcula a Longitude geografica do cada
individuo em radianos.
  data$latR=data$Latitude*pi/180 # calcula a Latitude geografica do cada
individuo em radianos
  loc_pattern$lon.R=loc_pattern$Longitude*pi/180 # calcula a Longitude
geografica do cada ponto padrão em radianos.
```

```
loc_pattern$lat.R=loc_pattern$Latitude*pi/180 # calcula a Longitude
geografica do cada ponto padrão em radianos.
dist=data.frame(
  acos(sapply(sin(loc_pattern$lat.R),"*",sin(data$latR))+
  sapply(cos(loc_pattern$lat.R),"*",cos(data$latR))*
  cos(sapply(loc_pattern$lon.R,"-",data$longR)*-1))*R) # calcula a distancia
geográfica para cada indivíduo com cada um dos pontos padrão.
ID=loc_pattern[,1] # cria um objeto com os nomes dos pontos padrão.
colnames(dist)=ID # troca o nome das colunas do objeto "dist" com os nomes
dos pontos padrão.
geo.groups=colnames(dist)[apply(dist,1,which.min)] # identifica o ponto
padrão (e a coluna do ponto padrão) com a minima distancia ao ponto de cada
indivíduo.
data=subset(data,select=-c(longR,latR)) # selecciona todos os datos do
objeto "data" menos a coluna com a Longitude e Latitude em radianos.
results1=data.frame(data,geo.groups) # cria um dataframe com o objeto
"data" e o resultado do analise (os grupos formados pelo analise de
distancias).
write.csv2(results1, file="geo_groups.csv") # exporta um arquivo *csv como
uma tabela com los resultados da primeira parte do analise.
data1=subset(results1,select=-c(Latitude,Longitude,geo.groups)) #
selecciona todos os datos do resultado da primeira parte, menos a coluna com
a Longitude, Latitude, e geo.groups.
r1=matrix(NA,length(unique(results1$geo.groups)),ncol(data1)) # cria uma
matriz de NA's.
colnames(r1)="media" # troca os nomes das colunas da matriz
rownames(r1)=LETTERS[(1:length(unique(results1$geo.groups)))]
r1=as.data.frame(r1) # converte a matriz num dataframe.
for(j in 1:ncol(r1))
  r1[j]=tapply(data1[,j],geo.groups,FUN=mean) # função "loop" para
calcular a media de cada caracter estudado por cada grupo proposto no
resultado da primeira parte.
matriz.diff=data.frame(sapply(r1$media,"-",data1[1])) # calcula as
diferencias entre as medições de cada indivíduo com a media do cada grupo
proposto na primeira parte do analise.
colnames(matriz.diff)=rownames(r1)
bio.groups=colnames(matriz.diff)[apply(matriz.diff,1,function(x)
which.min(abs(x-0)))] # identifica o ponto padrão (e a coluna do ponto
padrão) com o valor mais proximo ao CERO "0" de cada indivíduo.
results2=data.frame(results1,bio.groups) # cria um dataframe com o objeto
"data1" e o resultado do analise (os grupos formados pelo analise de
diferencias de medias).
write.csv2(results2, file="geo_bio_groups.csv") # exporta um arquivo *csv
como uma tabela com los resultados da segunda parte do analise.
data3=subset(results2,select=c(Latitude,Longitude,geo.groups,bio.groups))
# selecciona aLongitude, Latitude, geo.groups e bio.groups do results2.
list_geo <- split(data3, data3$geo.groups) # cria uma lista de data.frames
divididos pelos grupos da analise geografico (geo.groups)
list_bio <- split(data3, data3$bio.groups) # cria uma lista de data.frames
divididos pelos grupos das diferencia de medias (bio.groups)
```

```
### in the MAP section, replace into command points()
### the groups (X) that you want plot as follow:
###      "list_geo$ X $Longitude" where X is a group to test
###      "list_geo$ X $Latitude" where X is a group to test
if(map=="TRUE") # se a opção do mapa é marcada, plotar o mapa
{
  require(maptools) # carrega o pacote 'maptools'
  xlim<-c(min(results2$Longitude),max(results2$Longitude)) # delimita os
limites da longitude no mapa
  ylim<-c(min(results2$Latitude),max(results2$Latitude)) # delimita os
limites da latitude no mapa
  readShapeLines("americas_adm0.shp") -> shape # chama ao R o shalefile do
mapa usado.
  par(mfrow=c(1,2)) # divide o grafico em duas regiões.
  plot(shape, xlim=xlim, ylim=ylim, axes=TRUE, col='light grey') # realiza
o plot do grafico.
  title(main="geo.groups") # dá um nome ao plot do mapa o nome pode ser
trocado segundo o usuario.
  box() # cria um box ao redor do plot
  par(new=TRUE) # permite que o proximo plot não elimine o anterior
  points(list_geo$A$Longitude,list_geo$A$Latitude, pch=21,
col="black",bg="red",cex=0.9) # faz o plot dos pontos do grupo criado pela
analise de distancias
  plot(shape, xlim=xlim, ylim=ylim, axes=TRUE, col='light grey') # realiza
o segundo plot do grafico.
  par(new=TRUE) # permite que o proximo plot não elimine o anterior
  title(main="bio.groups") #dá um nome ao plot do mapa o nome pode ser
trocado segundo o usuario.
  box() # cria um box ao redor do plot
  points(list_bio$A$Longitude,list_bio$A$Latitude, pch=21,
col="black",bg="red",cex=0.9) # faz o plot dos pontos do grupo criado pela
analise de diferenca de médias
}
else # se o mapa não é chamado, não realizar outro função.
{}
if(save.map=="TRUE") # se a opção de salvar mapa é marcada, salvar o mapa
{
  jpeg(filename = "mapa.jpg", width = 480, height = 480, pointsize = 12,
quality = 100,bg = "white", res = NA) # formato da imagem para salvar o
plot.
  require(maptools) # carrega o pacote 'maptools'
  xlim<-c(min(results2$Longitude),max(results2$Longitude)) # delimita os
limites da longitude no mapa
  ylim<-c(min(results2$Latitude),max(results2$Latitude)) # delimita os
limites da latitude no mapa
  readShapeLines("americas_adm0.shp") -> shape # chama ao R o shalefile do
mapa usado.
  par(mfrow=c(1,2)) # divide o grafico em duas regiões.
  plot(shape, xlim=xlim, ylim=ylim, axes=TRUE, col='light grey') # realiza
o plot do grafico.
  title(main="geo.groups") #dá um nome ao plot do mapa o nome pode ser
```

```
trocado segundo o usuario.  
  box() # cria um box ao redor do plot  
  par(new=TRUE) # permite que o proximo plot não elimine o anterior  
  points(list_geo$A$Longitude,list_geo$A$Latitude, pch=21,  
col="black",bg="red",cex=0.9) # faz o plot dos pontos do grupo criado pela  
analise de distancias  
  plot(shape, xlim=xlim, ylim=ylim, axes=TRUE, col='light grey') # realiza  
o segundo plot do grafico.  
  title(main="geo.groups") #dá um nome ao plot do mapa o nome pode ser  
trocado segundo o usuario  
  box() # cria um box ao redor do plot  
  points(list_bio$A$Longitude,list_bio$A$Latitude, pch=21,  
col="black",bg="red",cex=0.9) # faz o plot dos pontos do grupo criado pela  
analise de diferenca de médias  
  dev.off() # fecha e salva o objeto mapa com o formato da imagem.  
  }  
  else{} # se o mapa não é chamado, não realizar outro função.  
  return(results2) # retorna o resultado final, uma tabela com o numero  
(identificação), latitude, longitude grupo geografico e grupo relacionado ao  
caracter estudado.  
} # FIM da função!!!
```

HELP geocalInfor

geocalInfor package:nenhum R Documentation

Geographic distances calculator

Description:

Uma função simples que:

(1). identifica unidade geográfica padrão mais próxima aos de pontos geográficos estudados, calculando as distâncias em Kms entre a base de dados de estudo e as unidades padrão.

(2). Usando o valor de um caracter biologico amostrado (uma medição de algum rasgo dos individuos estudados) e as médias para esse caracter ("mean") para cada 'grupo geografico' da analise (1), são calculadas as diferencias das medias desse caracter (media de cada grupo identificado - o valor de cada individuo para esse caracter), e identifica o 'grupo geográfico' que tenha a minima diferenca na média, criando um novo set de grupos.

(3). realiza um plot do mapa da area estudada (o shapefile deve ser derfinido pelo usurario) com os pontos de cada grupo identificado nos analises (1) e (2) para sua comparação.

Usage:

```
geocalInfor(x,loc_pattern)
```

```
geocalInfor(x,loc_pattern,map=TRUE,save.map=FALSE)
```

Arguments:

x: objeto contendo uma planilha de três colunas no formato "*csv" ou "*txt".

loc_pattern: o arquivo com os pontos geográficos pradrão (o nome do

objeto deve ser exato!).

map: além da matriz de distâncias gerada na função, plota os pontos (grupos de pontos) num mapa.

save.map: salva o mapa em uma figura *"jpeg".

Details:

0 argumento x (o arquivo 1) deve ser uma planilha com coordenadas em grau decimal. A planilha deve ter as distribuições geográficas estudadas, com 4 colunas com conteúdo obrigatório e ordenado:

Coluna 1: Nome ou identificação do indivíduo ou ponto. Coluna 2: dados de latitude ("Latitude").

Coluna 3: dados da Longitude ("Longitude").

Coluna 4: Os valor do caracter biologico continuo (medição em metros/cms/mm, etc).

Um arquivo 2, de nome "loc_pattern", tem as distribuições geográficas padrão (pré-definido pelo usuário), com três colunas de conteúdo obrigatório e ordenado:

Coluna 1: Nome dos pontos geográficos predefinidos (letras!).

Coluna 2: dados da Latitude ("Latitude").

Coluna 3: dados da Longitude ("Longitude").

Para utilização do argumento "map" é necessário instalar o pacote maptools: <http://cran.r-project.org/web/packages/maptools/index.html>.

Também é necessário que o usuário salve no diretório o shapelite (*"shp") que é usado na construção do mapa, nomeando-o dentro do código da função com o mesmo nome que está no diretório.

Para a elaboração dos mapas, o usuario precisa reemplazar o nome da are segundo a área que quera comparar! see message:

```
### in the MAP section, replace into command points() the groups thast you want plot! ###. as follow:
```

```
## "list_geo$ X $Longitude" where X is a group to test ##
```

```
## "list_geo$ X $Latitude" where X is a group to test ##
```

Value:

A função retorna uma tabela em formato texto com colunas com os nomes de cada ponto/individuo estudado e os grupos aos quais perteneses segundo os analises:

"geo.groups" para os grupos da analise (1). "bio.groups" para a analise (2).

A tabela é automaticamente salva no formato "*.csv" no diretório que está sendo utilizado.

Se o comando map=FALSE não é marcado, um mapa com os resultados será ploteado.

Se save.map=TRUE é marcado, a função também salva uma figura no formato "jpeg".

Warning:

Caso o objeto de entrada 2 não exista, uma mensagem de erro será dada pela função e esta será interrompida.

Note:

As distancias geográficas são calculadas usando a formula da Lei esférica dos cossenos ('Spherical law of cosines')

http://en.wikipedia.org/wiki/Spherical_law_of_cosines.

Os shapes para plot dos mapas podem ser baixados aqui:

<http://www.diva-gis.org/Data>

Author(s):

Sergio David Bolívar

References:

<http://cran.r-project.org/web/packages/maptools/index.html>

<http://www.diva-gis.org/Data>

http://en.wikipedia.org/wiki/Spherical_law_of_cosines

<http://www.r-bloggers.com/great-circle-distance-calculations-in-r/>

Examples:

```
x<read.csv("loc_pontos.csv",header=T,sep="\t",dec=",")
```

```
geocalInfor(x,map=FALSE) # Usando distancias geograficas e diferencias de médias; identifica os grupos mais proximos (geograficamente e quantitativamente) aos dados estudados e cria duas tabelas resultados desses analises.
```

```
geocalInfor(x) #Cria as tabelas e o mapa(s) com os pontos.
```

```
geocalInfor(x,map=TRUE,save.map=TRUE) #Não cria o mapa(s), mas salva no diretório.
```

Arquivos anexos:

[function_geocalinfor.r](#)

[tutorial_geocalinfor.txt](#)

[loc_pattern.csv](#)

[loc_pontos.csv](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2014:alunos:trabalho_final:bolivarmz:start 

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