Sergio Marques de Souza (Bogão)

Doutorando em Zoologia pelo IB-USP Meu interesse de pesquisa reside na história evolutiva da herpetofauna neotropical e o que os sinais evolutivos podem nos ensinar sobre paleoambientes da América do Sul e sobre padrões atuais de diversidade.

Meu mestrado foi sobre o efeito de rios amazônicos sobre as populações de uma espécie de lagarto, *Leposoma osvaldoi*. marques_de_souza_et_al_2013_leposoma.pdf

Meus Exercícios

exec

Proposta de trabalho final

Plano A

Minha idéia é criar uma função que execute duas tarefas: A primeira é gerar uma matriz simétrica de distâncias geográficas entre localidades a partir de um input de uma planilha de localidades e suas respectivas coordenadas. Nessa tarefa, o usuário poderá colocar as coordenadas em três diferentes formatos (graus decimais, graus/min/seg e UTM), sendo que haverá um argumento na função para epecificar qual o formato que será utilizado.

A segunda tarefa que a função irá desempenhar é realizar um teste de Mantel de correlações entre matrizes de distâncias. Essas distâncias poderão ser de várias naturezas (por exemplo distâncias geográficas, genéticas, ecológicas ou morfológicas), e o único requisito é que haja uma correspondência exata entre os terminais (no caso, as localidades). O Teste de Mantel consiste em realizar permutações em linhas e colunas de uma das matrizes e computar o coeficiente de correlação entre as células para cada uma das permutações. Depoi é realizada uma comparaçao a distribuiçao dos valores permutados com a correlação entre as variaveis originais. Um dos argumentos da funçao será o número de permutaçoes que serão realizadas nos dados originais. Também será criado automaticamente um histograma com a frequência dos valores nulos e onde os dados originais estão situados neste gráfico.

Plano B

O plano B consiste em fazer uma função que renomeie automaticamente os terminais utilizados em análises filogenéticas moleculares. Isso possui uma grande utilidade quando se está trabalhando com conjunto de dados com centenas de indivíduos sequenciados e vários (por exemplo, 10) marcadores moleculares. Quando do momento de concatenar os dados de todos os marcadores, os nomes dos

terminais devem estar iguais, do contrário gerando erros de concatenação de dados. Normalmente este trabalho é feito manualmente, o que toma um tempo precioso e dá margem a erros de digitação. Certamente essa função faria uso do pacote “ape”, dada sua capacidade de rodar arquivos .FASTA no R.

> Sérgio, sua proposta A está bem descrita e você definiu bem o que sua função vai fazer. Fiquei com algumas dúvidas: uma das saídas é a matriz de distâncias ou ela só vai ser calculada para fazer o teste de Mantel depois? O R já tem um pacote com o teste de Mantel, acredito que você poderia incorporar na sua função, sem ter que partir do zero.
>
> A proposta B tem uma idéia legal, mas precisa de mais detalhamento (o que são os dados de entrada, qual o caminho que a função vai seguir para renomear os terminais...).
>
> —- *Sheina*

Ajuda/Help da função

```
distfilogeo                package:unknown                R Documentation

Teste de correlação de Mantel entre distâncias geográficas e distâncias
filogenéticas de amostras

Description:

    A função possui três passos principais: 1. Primeiro ela calcula as
distâncias geográficas par a par entre as localidades de n amostras a partir
de uma matriz (fornecida pelo usuário) contendo as coordenadas das amostras;
2. Calcula as distâncias filogenéticas par a par entre as mesmas amostras a
partir de um arquivo da classe "phylo" (criado a partir do pacote "ape"),
que corresponde a uma árvore filogenética; 3. Calcula o coeficiente de
correlação de Pearson entre as matrizes de distâncias geográficas e
filogenéticas, permuta n vezes a matriz de distâncias geográficas, recalcula
o coeficiente de Pearson a cada permutação, plota um histograma com os
valores dos coeficientes permutados e fornece a proporção de valores que são
menores ou maiores que o valor observado.

    Decidi usar a distância filogenética em termos de comprimento de ramos
ao invés da distância genética, amplamente usada. São duas informações
diferentes. A distância genética é uma medida da diferença genética entre
dois organismos que, por não estar situada em um contexto filogenético, pode
ser resultado de uma homoplasia (organismos próximos filogeneticamente mas
```

```
com grande distância genética ou organismos distantes filogeneticamente,
porém com diferenças genéticas mínimas geradas por convergência). Já a
distância filogenética é uma medida da distância evolutiva entre dois
organismos, de acordo com determinada hipótese (árvore) filogenética.

Usage:

     distfilogeo(y,x,dec=TRUE, nperm=1000)

Arguments:

y: Data Frame contendo as coordenadas de cada localidade.

x: Objeto da classe "phylo", proveniente do pacote "ape".

dec: se TRUE, indica que os dados de coordenadas estão em graus decimais. Se
FALSE, as coordenadas estão em graus,minutos e segundos (GMS) (ver Details).

nperm: Número de vezes que a planilha de distância geográfica será
randomizada.

Details:

    Em y, dois formatos de coordenadas são aceitos, graus decimais e
graus,minutos e segundos(GMS). No primeiro caso (graus decimais), o data
frame deve obrigatoriamente seguir a seguinte estrutura: a primeira coluna
deve conter o nome das localidades (ou amostras), a segunda coluna deve
conter os valores de latitudes e a terceira coluna os valores de longitude.
No caso do usuário entrar com as coordenadas em GMS, o data frame deve
obrigatoriamente seguir a seguinte estrutura: a primeira coluna deve conter
o nome das localidades (ou amostras), a segunda coluna os valores de graus
para latitude, a terceira os valores de minutos para latitude, a quarta os
valores de segundos para latitude, a quinta deve conter o quadrante da
latitude (N ou S), escrito em caixa alta. A sexta, sétima, oitava e nona
colunas seguem a mesma estrutura da latitude, porém com os dados de
longitude. o quadrante da longitude aceita W ou E, ambos em caixa alta.

    x corresponde a uma árvore filogenética obrigatoriamente enraizada, já
que o algoritmo não funciona com arvores desenraizadas. É importante checar
se a árvore contem terminais no mesmo número e na mesma ordem que as
amostras (ou localidades) da planilha de coordenadas. É importante também
que o arquivo que gerou o objeto contenha informações sobre os comprimentos
dos ramos da árvore. Normalmente o arquivo é do tipo .nex e foi lido no R a
partir da função "read.nexus" do pacote ape.

Value:

     É gerado um histograma contendo os valores dos coeficientes calculados
após as permutações. Uma linha vermelha indica a posição do coeficiente
observado no histograma, além de um texto indicando a proporção de valores
permutados maiores e menores do que o valor observado. Também é gerada uma
```

```
lista contendo:

  comp1 : Matriz quadrática de distâncias geográficas par a par

  comp2 : Matriz quadrática de distâncias filogenéticas par a par


Warning:

    Em y, quando entrando com dados em GMS, a função não avisará se o
usuário estiver errando nos valores dos quadrantes que são aceitos (N ou S
para latitude e W ou E para longitude, todos em caixa alta). Por exemplo,
"s" em caixa baixa será entendido como quadrante norte, gerando erro no
calculo das distâncias geográficas.
    A função também não avisará se o número e a sequência das amostras ou
localidades estiver diferente da lista de terminais da árvore filogenética.
Para checar a ordem dos terminais, use a indexação $tip.label antes de rodar
a análise.


Author(s):

    Sergio "Bogão" Marques de Souza

    sergio.bogao@gmail.com

References:

    Mantel, N. (1967). "The detection of disease clustering and a
generalized regression approach". Cancer Research 27 (2): 209–220.
    Mantel Test. http://en.wikipedia.org/wiki/Mantel_test

See Also:

    Classe de objetos "phylo" no pacote ape, assim como a função
"read.nexus". Árvores geradas a partir do software MrBayes já são arquivos
.nexus e podem ser lidas diretamente com a função "read.nexus"

Examples:

    teste<-read.nexus("arvore.con.tre")
    teste.gms<-read.csv("coordenadas_teste_m.csv", as.is=T)
    distfilogeo(teste.gms, teste, dec=FALSE, nperm=5000)
```

Código da Função

```
distfilogeo<-function(y,x,dec=TRUE, nperm=1000)
{
```

```
## Calculo da matriz quadrática de distâncias geográficas##

    if(dec==FALSE)# indica o caminho a seguir pelo algoritmo caso o usuário
tenha entrado com as coordenadas em graus, minutos e segundos
    {
      y$lat_dec<-(y[,2]+(y[,3]/60)+(y[,4]/3600)) #Cria coluna com o valor da
latitude em graus decimais
      for (i in 1:length(y$lat_dec))# Olha para cada valor da coluna de
graus decimais
      {
        if(y[i,5]=="S")# Checa se a latitude está no quadrante sul
        {
          y$lat_dec[i]<-y$lat_dec[i]*-1# Caso esteja no sul, multiplica o
valor por -1
        }
      }
      y$long_dec<-(y[,6]+(y[,7]/60)+(y[,8]/3600)) #Cria coluna com o valor
da longitude em graus decimais
      for (i in 1:length(y$lat_dec))# Olha para cada valor da coluna de
graus decimais
      {
        if(y[i,9]=="W")# Checa se a longitude está no quadrante oeste
        {
          y$long_dec[i]<-y$long_dec[i]*-1#Caso esteja no oeste, multiplica o
valor por -1
        }
      }
    }
    else# as coordenadas já estão em graus decimais
    {
      colnames(y)<-c("locality","lat_dec","long_dec")# muda o nome das
colunas do data frame
    }
    y$lat_dec<-y$lat_dec*(pi/180)# Transforma em radianos
    y$long_dec<-y$long_dec*(pi/180)# Transforma em radianos
    dist.mat<-
matrix(NA,length(y$lat_dec),length(y$lat_dec),dimnames=list((y[,1]),(y[,1]))
)# Cria a matriz quadrática com os NA's
    r<-6371 # Raio aproximado da terra, em kilometros(a medida resultante
será em Km também)
    for(a in 1:(length(y$lat_dec)-1))# "for" de nivel superior, eixo x da
matriz simétrica
    {
      for(b in (a+1):(length(y$lat_dec)))# "for" de nivel inferior, faz as
medidas uma a uma, correspondente ao eixo y da matriz simétrica
      {
        dlat<-(y$lat_dec[b]-y$lat_dec[a])# diferença entre as latitudes das
duas coordenadas
        dlong<-(y$long_dec[b]-y$long_dec[a])# diferença entre as longitudes
das duas coordenadas
z<-2*r*asin(sqrt((sin(dlat/2)^2)+cos(y$lat_dec[a])*cos(y$lat_dec[b])*(sin(dl
```

```
ong/2)^2)))# Aplicação da formula de Haversine, que calcula as dist^ncias
geográficas
        round(z)# Dá o numero de casas decimais do valor de distancia
geográfica
        dist.mat[b,a]<-z # Preenche a matriz quadráticas com os valores de
distancias calculados
      }
    }
##Calculo da matriz quadratica de distancias filogenéticas##

    vet<-c(1:length(x$tip.label))# faz um vetor do tamanho do numero de
terminais da arvore
    edge.length<-x$edge.length# Armazena o vetor que contem o tamanho dos
ramos da arvore no objeto "edge.length"
    edge<-data.frame(x$edge)# Armazena a matriz que contem os nós ancestrais
e descendentes dos ramos no objeto "edge"
    colnames(edge)<-c("anc","desc")# Muda o nome das colunas de "edge"
    root<-length(x$tip.label)+1# Cria o vetor que contém o número do nó onde
a arvore está enraizada, ou seja, o ancestral de toda a arvore
    dist.filo<-
matrix(NA,length(x$tip.label),length(x$tip.label),dimnames=list((x$tip.label
),(x$tip.label)))# Cria a matriz quadrática com os NA's onde serão
armazenadas as distâncias filogenéticas
    caminhos<-c(1:length(vet))# faz outro vetor do tamanho do numero de
terminais da árvore filogenética
    for(z in 1:(length(vet)-1))# for que irá construir a lista de caminhos,
que irá indicar o caminho por qual cada terminal da arvore chega até a raiz
    {
      a<-which(edge$desc==vet[z])# acha a linha que contem o terminal em que
se quer encontrar o caminho até a raiz
      c<-0# cria o objeto c com 0 na primeira posição
      c[1]<-a # salva o numero da linha na primeira posição de c
      for(y in 2:(2*(length(x$tip.label)-2)))# "for" que começa em 2 (já que
irá ser adicionado ao c) e irá rodar o mesmo número de vezes que o número de
nós de uma arvore filogenética enraizada (dado pela formula (2*numero de
terminais-2))
      {
        c[y]<-which(edge$desc==edge[c[y-1],]$anc)# adiciona em c a linha do
objeto edge onde o descendente da linha anterior agora é o ancestral
        if(edge[c[y],]$anc==root){break}# Indica que o "for" deve parar
quando o ancestral chegar na raiz
      }
      caminhos[z]<-list(c)# salva o caminho de cada terminal até a raiz da
arvore no vetor caminhos
    }
    caminhos[length(vet)]<-which(edge$desc==length(vet))# modifica a ultima
posição do vetor caminhos (raiz->grupo externo) para a linha do vetor edge
onde o descendente é o grupo externo
    for(i in 1:(length(x$tip.label)-1))# "for" de nivel superior, eixo x da
matriz simétrica
```

```
    {
      for(b in (i+1):(length(x$tip.label)))# "for" de nivel inferior, faz as
medidas uma a uma, correspondente ao eixo y da matriz simétrica
      {
        ib<-setdiff(caminhos[[i]],caminhos[[b]])# Compara as posições i e b
da lista "caminhos" e guarda no objeto "ib" apenas os valores únicos
contidos em "caminhos" na posição i
        bi<-setdiff(caminhos[[b]],caminhos[[i]])# mesmo que o de cima, porém
guarda no objeto "bi" apenas os valores únicos contidos em "caminhos" na
posição b
        dist<-c(ib,bi)# Guarda os valores de ib e bi no objeto dist.
Corresponde ao caminho percorrido na arvore para se ir de um terminal a
outro
        soma<-rep(NA,length(dist))# cria o vetor soma preenchido com NA's no
comprimento de dist
        for(t in 1:length(dist))# "for" que irá fazer a soma dos
comprimentos de ramos
        {
          soma[t]<-edge.length[dist[t]]# preenche o vetor soma com os
comprimentos de ramos relativos ao caminho traçado pelo vetor dist
          final<-sum(soma)# soma os comprimentos de ramos e guarda no vetor
"final"
        }
        dist.filo[b,i]<-final#Guarda o valor de "final" na posição
correspondente à distancia entre os taxons
      }
    }
## Calculo do coeficiente de correlação de Pearson entre as matrizes de
distâncias geográficas e filogenéticas, além de permutar os dados e plotar
em um histograma##

    vet.filo<-dist.filo[lower.tri(dist.filo)]# Faz um vetor com os valores
de distancia filogenética entre os terminais
    vet.geo<-dist.mat[lower.tri(dist.mat)]# Faz um vetor com os valores de
distancia geográfica entre os terminais
    med.filo<-mean(vet.filo)# Calcula a media das distancias filogenéticas
    med.geo<-mean(vet.geo)# Calcula a media das distancias geográficas
    cima<-rep(NA,length(vet.filo))# Cria um vetor que irá ser o numerador da
formula do coeficiente de correlação de Pearson
    t1den<-rep(NA,length(vet.filo))# Cria um vetor que irá ser o primeiro
termo do denominador da formula do coeficiente de correlação de Pearson
    t2den<-rep(NA,length(vet.filo))# Cria um vetor que irá ser o segundo
termo do denominador da formula do coeficiente de correlação de Pearson
    for(l in 1:length(vet.filo))#for que irá calcular o coeficiente de
correlação de Pearson
    {
      cima[l]<-(vet.filo[l]-med.filo)*(vet.geo[l]-med.geo)# Calculo do
numerador
      t1den[l]<-(vet.filo[l]-med.filo)^2# Calculo do primeiro termo do
denominador
      t2den[l]<-(vet.geo[l]-med.geo)^2# Calculo do segundo termo do
```

```
denominador
    }
    r.pearson<-sum(cima)/(sqrt(sum(t1den))*sqrt(sum(t2den)))# Calculo do
coeficiente de correlação de Pearson
    r.permut<-rep(NA,nperm)# Vetor que irá conter os dados dos coeficientes
calculados após a aleatorização dos dados
    for (q in 1:nperm)# for que irá aleatorizar uma das matrizes, recalcular
o coeficiente de pearson, e jogar o resultado no vetor r.permut
    {
      vet.geo.per<-sample(vet.geo)# aleatorizando o vetor vet.geo
      for(u in 1:length(vet.filo))# Novo calculo do coeficiente de
correlação de Pearson, mesmos calculos de acima
      {
        cima[u]<-(vet.filo[u]-med.filo)*(vet.geo.per[u]-med.geo)# Calculo do
numerador
        t1den[u]<-(vet.filo[u]-med.filo)^2# Calculo do primeiro termo do
denominador
        t2den[u]<-(vet.geo.per[u]-med.geo)^2# Calculo do segundo termo do
denominador
      }
      r.permut[q]<-sum(cima)/(sqrt(sum(t1den))*sqrt(sum(t2den)))# jogando o
resultado aleatorizado no vetor r.permut
    }
    menores<-length(which(r.permut<r.pearson))/nperm# numero de valores
aleatórios que são menores que o calculado
    maiores<-length(which(r.permut>r.pearson))/nperm# numero de valores
aleatórios que são maiores que o calculado
    hist(r.permut, axes=T, col="8", main="", xlab="",
ylab="Frequência",xaxp=c(-1,1,10), cex.axis=0.8)# Faz o histograma com os
valores de coeficientes de Pearson aleatorizados
    abline(v=r.pearson, lty=3,col="red")# Desenha uma linha tracejada
vermelha no histograma, no valor do coeficiente calculado
    if(menores>maiores)# teste se a proporção de valores menores é maior que
a de maiores (correlação negativa)
    {
      title(paste(maiores,"% valores >= que o
observado"),cex.main=0.8,col.main="red")# escreve a proporção de valores >=
do que o observado
    }
    if(maiores>menores)# teste se a proporção de valores maiores é maior que
a de menores (correlação positiva)
    {
      title(paste(menores,"% valores <= que o
observado"),cex.main=0.8,col.main="red")# escreve a proporção de valores <=
do que o observado
    }
    planilhas<-list ()# Cria uma lista chamada "planilhas"
    planilhas$Distâncias_geográficas<-dist.mat# Coloca o objeto "dist.mat"
na lista
    planilhas$Distâncias_filogenéticas<-dist.filo# Coloca o objeto
```

```
"dist.filo" na lista
    return(planilhas)# retorna a matriz de distâncias filogenéticas e
geográficas
}
```

Arquivo da função para download

distfilogeo_final.r