

Mauro Sugawara



Estou no primeiro ano de mestrado em Ecologia na USP, sob orientação do Tiago Bosisio Quental. O nome do meu projeto é “Dinâmica da diversificação de Placentalia (Mammalia): integrando o registro fóssil com filogenias moleculares”. Uma parte da análise será realizada no site do [Paleobiology Database](#), para o restante irei utilizar o R.

Tenho alguns conhecimentos básicos de R, pois utilizei esta linguagem para o meu projeto de iniciação científica. Entretanto, nunca tive nenhuma orientação formal e com o curso pretendo conhecer melhor como esta linguagem está estruturada.

Meus Exercícios

Link para os meus [exercícios](#).

Minha Proposta

A proposta 'A' é bem útil para o meu mestrado, principalmente a parte de plotar o eixo da abcissa. Contudo, a proposta 'B' é mais desafiadora e também muito mais trabalhosa. Esta função será útil para algumas pesquisas no meu laboratório e pode ser que eu a utilize em algum ponto do meu mestrado. A princípio, pensei em dar uma olhada em como se trabalha com mapas no R e caso não seja tão complicado como imagino faria a proposta 'B'.

Plano A

As curvas de diversidade construídas a partir do registro fóssil permitem estudar diretamente a dinâmica de diversificação de um dado grupo. Há pelo menos três métodos para a construção destas curvas:

- O método *sampled in bin* apenas conta o número de táxons que aparece a cada intervalo de tempo.
- O *gap analyses* pega os dados da primeira e da última ocorrência e assume que o táxon estava presente em todos os intervalos de tempo intermediários, este método pode ser considerado como uma estimativa superior da diversidade.
- Por fim, o método *boundary-crosser* é mais conservador e apenas conta os táxons presentes em cada limite de tempo, garantindo que os táxons coexistiram.

A entrada da função pode ser uma filogenia datada (formato “ape”) ou um objeto (matriz ou data.frame) contendo pelo menos três colunas: uma com a lista de táxons e outras duas com as datas de ocorrência. O usuário pode optar por utilizar a escala geológica ou fornecer um vetor numérico com intervalos de tempo a serem utilizados na análise. Há duas escalas geológicas disponíveis no pacote “paleoPhylo” (objetos *berggren95* e *gradstein04*) que podem ser utilizadas.

A função retorna uma matriz com 3 colunas, uma para cada método, e nas linhas a diversidade em cada intervalo de tempo. Esta função pode ainda plotar as curvas de diversidade mostrando no eixo da abscissa o tempo geológico. O pacote “paleoPhylo” possui uma função que faz este plote (“timeAxis()”), porém, ela é específica para o formato “paleoPhylo”. Pretendo implementar uma função que funcione tanto para as minhas curvas de diversidade, quanto para filogenias no formato “ape”. Dessa forma, posso utilizar esta função em outros pontos do meu mestrado.

Plano B

A coordenada onde um fóssil foi encontrado quase nunca tem relação com as coordenadas onde este táxon viveu. Estas coordenadas do passado - chamadas de paleocoordenadas - podem ser recuperadas ao comparar o mapa atual com mapas do passado, levando em consideração a deriva do continentes. Alguns bancos de dados, como o [Paleobiology Database](#), já possuem os dados de paleocoordenadas disponível para download.

A função recebe uma matriz ou data.frame e constrói os mapas de ocorrência para cada táxon. Este objeto de entrada deve conter o nome dos táxons, a data e as coordenadas em que foram encontrados. Um dos argumentos da função serve para informar se as paleocoordenadas já foram fornecidas. Neste caso não é necessário fazer a conversão e apenas se constrói os mapas de ocorrência.

A saída da função pode ser uma lista com os mapas de distribuição para táxon ou um mapa de distribuição para cada época, fica a escolha do freguês.

Comentários

As duas propostas estão claras e bem apresentadas. Parece que você já tem familiaridade com as análises e os pacotes de R que já fazem coisas parecidas. Sugiro mandar bala na proposta A que está mais elaborada e parece ter mais utilidade direta a você. Sobre os mapas em R para o plano B, dê uma olhada nos task view de análise de dados espaciais no r-project.org (este também tem dicas bacanas: <http://www.molecular ecologist.com/2012/09/making-maps-with-r/>). Ainda, nas funções dos anos anteriores outros alunos já implementaram ligações com bancos de dados on line e mapas (e.g. Tauana de 2012).

Sara Mortara

Implementação - Plano A

Acabaram ocorrendo algumas modificações no desenvolver da função.

Primeiro, decidi retirar a opção de fornecer uma matriz como entrada. Em geral o que temos é uma matriz com dados numéricos (datas de ocorrência) e caracteres (nomes dos táxons), ou seja, no R isto é um data frame. Além disso, retirei a opção de escolha do método. Agora a função realiza as três análises ao mesmo tempo. Caso o usuário queira apenas o resultado de uma delas, basta selecioná-la no objeto de saída.

Na saída, retirei a opção de fazer a análise para os táxons, uma vez que é apenas uma questão de

escala. Esta análise pode ser feita utilizando um filtro para os táxons e a função 'apply'. Também modifiquei um pouco o objeto de saída, colocando os três métodos nas linhas e os intervalos de tempo nas colunas - que estão ordenadas do mais próximo do presente até o mais antigo.

Página de Ajuda / Help

divercurve package:nenhum R Documentation

Curvas de diversidade.

Description:

divercurve estima a diversidade ao longo do tempo utilizando três métodos: sampled in bin, gap analysis e boundary-crosser.

Usage:

```
divercurve(object, firstlast=FALSE, scale="berggren95", level="epoch",  
plot=TRUE, ...)
```

Arguments:

object pode ser um data frame ou uma filogenia do formato 'ape'.
firstlast controla o tipo de data frame fornecido. Caso seja 'TRUE', apenas duas colunas serão consideradas, como as datas da primeira e última ocorrências de cada táxon. Caso seja 'FALSE', todas as colunas fornecidas serão consideradas e interpretadas como um intervalo do tempo geológico.
scale controlam onde serão os "cortes" do tempo. Caso seja fornecido um único valor, este será interpretado como o tamanho desejado para os intervalos de análise; caso seja um vetor, os valores serão interpretados como os pontos de "corte"; caso o usuário queira utilizar alguma escala geológica basta fornecer o nome ("berggren95" ou "gradstein04"). O default é para a escala geológica "berggren95".
level controla qual nível da escala geológica deverá ser considerado.
plot determina se as curvas de diversidade devem ser plotadas.
... outros parâmetros que podem ser passados para a função 'plot'.

Details:

Caso seja fornecida uma filogenia esta será transformada em um data frame de primeira e última ocorrências e analisada de acordo, neste caso o 'gap analysis' é desnecessário.

No caso de ser fornecida um data frame com número de ocorrências (firstlast = FALSE), não é possível escolher quais serão as quebras no tempo. Uma vez que este data frame já possui em suas colunas uma escala temporal. Cabe ao usuário apenas indicar qual o nível da escala geológica utilizado.

Para a escala geológica "berggren95" a única opção para o level é 'epoch'.

Para a escala geológica "gradstein04" level pode ser: 'eon', 'era', 'epoch' ou 'stage'.

Value:

A função retorna uma matriz com 3 linhas, uma para cada método, e nas colunas a diversidade em cada intervalo de tempo. Os valores estimados com o método 'boundary-crosser' possuem uma entrada a menos, uma vez que só se considera os limites entre os intervalos de tempo.

Note:

Esta função exige o pacote 'paleoPhylo'. Para instalar rode o seguinte comando no R:

```
> install.packages("paleoPhylo", repos="http://R-Forge.R-project.org")
```

Esta função também utiliza uma função auxiliar 'branching.times.with.extinct' desenvolvida por Daniel Varajão de Latorre para o curso do ano passado. O código da função e o arquivo da mesma encontram-se abaixo. Para maiores informações sobre a função 'branching.times.with.extinct' consultar a página <http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:05_curso_antigo:alunos2012:alunos:trabalho_final:danielvdelatorre:start>.

Author:

Mauro Toshio Caiuby Sugawara

maurotcs@gmail.com

Daniel Varajão de Latorre - branching.times.with.extinct

Examples:

```
# Faça o download do arquivo 'Carnivora_genus.csv' e depois rode o código abaixo.
```

```
# Os dados fósseis vieram do banco de dados Paleobiology Database.
```

```
carnivora <- read.table("Carnivora_genus.csv", sep=" ", header=T,  
row.names=1)  
divercurve(carnivora, firstlast=F, scale="berggren95", level="epoch",  
plot=T)
```

```
# Simula uma filogenia aleatória com 50 táxons terminais.
```

```
library(ape)  
filogenia <- rtree(50)  
plot(filogenia)  
axisPhylo()  
divercurve(filogenia, firstlast=F, scale="berggren95", level="epoch",  
plot=T)
```

```
# Faça o download do arquivo 'Raia_et_al_2013_tree.txt' e depois rode o código abaixo.
```

```
# Análise feita para a filogenia de mamíferos de Raia et al. 2013.
```

```
raia <- read.nexus("Raia_et_al_2013_tree.txt")  
divercurve(raia, firstlast=F, scale="berggren95", level="epoch", plot=T)
```

Código da Função 'divercurve'

```
divercurve <- function(object, firstlast=FALSE, scale="berggren95",
level="epoch", plot=TRUE, ...){
  # Testa se o objeto fornecido é da classe esperada.
  if (class(object) != "phylo" & class(object) != "data.frame") {
    stop("O objeto fornecido deve ser um data frame ou uma filogenia do
formato 'ape'.")
  }
  # Transforma a filogenia em um data frame com informações de primeira e
última ocorrências de cada ramo.
  if (class(object) == "phylo"){
    require(ape)
    # retira os nomes de nós internos na filogenia, que atrapalham na
indexação.
    object$node.label<-NULL
    # calcula a distância de cada nó para o presente.
    times <- branching.times.with.extinct(object)
    edge <- object$edge
    len <- object$edge.length
    object <-
data.frame(cbind(1:nrow(edge), rep(0,nrow(edge)), rep(0,nrow(edge))))
    colnames(object) <- c("taxon", "first", "last")
    for (i in 1:nrow(edge)){
      object[i,2] <- times[match(edge[i,1], names(times))]
      object[i,3] <- times[match(edge[i,1], names(times))]-len[i]
    }
    firstlast = TRUE
  }
  nc <- ncol(object)
  nr <- nrow(object)
  if (firstlast == TRUE){
    f <- max(object[,2])
    l <- min(object[,3])
    if (class(scale) == "numeric"){
      if (length(scale) == 1){
        datas <- seq(l,f,by=scale)
      }
      if (length(scale) > 1){
        datas <- sort(scale)
      }
      namet <- c(as.character(datas), as.character(round(f,2)))
    }
    if (class(scale) == "character"){
      require(paleoPhylo)
      if (scale == "berggren95"){
        data(berggren95)
        namet <- berggren95[,match(level, names(berggren95))][berggren95$MA
<= f & berggren95$MA >= l]
```

```
      datas <- berggren95$MA[berggren95$MA <= f & berggren95$MA >= l]
    }
    if (scale == "gradstein04"){
      data(gradstein04)
      namet <- gradstein04[,match(level,
names(gradstein04))][gradstein04$MA <= f & gradstein04$MA >= l]
      datas <- gradstein04$MA[gradstein04$MA <= f & gradstein04$MA >= l]
    }
  }
  # 'sobj' recebe o data frame com informação binária.
  sobj <-
cbind(object[,1],data.frame(matrix(rep(0,nr*length(datas)),nr,length(datas))
))
  names(sobj) <- c("taxon",namet)
  for (i in 1:nr){
    for (k in length(datas):1){
      if(object[i,2] > datas[k]){
        break()
      }
    }
    for (w in 1:length(datas)){
      if(object[i,3] < datas[w]){
        break()
      }
    }
    sobj[i,((w+1):(k+1))] <- 1
  }
}
if (firstlast == FALSE){
  # 'sobj' recebe o data frame com informação binária apenas, sem dados de
numero de ocorrências em cada intervalo.
  sobj <- as.matrix(object>0)
  sobj <- apply(sobj,2,as.numeric)
  sobj <- data.frame(sobj,row.names=rownames(object))
}
# 'gobj' recebe o data frame corrigido depois do 'gap analysis'.
gobj <- sobj
nc <- ncol(sobj)
nr <- nrow(sobj)
# 'bobj' recebe o data frame correspondente a análise de boundary-crosser.
bobj <-
cbind(as.character(sobj[,1]),data.frame(matrix(data=rep(0,(nr*(nc-2))),nr,nc
-2)))
names(bobj)[1] <- names(sobj)[1]
for (i in 1:nr){
  # seleciona apenas as linhas (taxons) com mais de uma ocorrência.
  if (sum(sobj[i,2:nc]) > 1){
    f<-match(1,gobj[i,2:nc])+1 # primeira ocorrência.
    l<-nc-match(1,rev(gobj[i,2:nc])) # ultima ocorrência.
    gobj[i,f:(l+1)]<-1
  }
}
```

```

      bobj[i,f:l]<-1
    }
  }
# Sampled in bin
sib <- apply(sobj[,2:nc],MARGIN=2,FUN=sum)
# Gap analysis
gap <- apply(gobj[,2:nc],MARGIN=2,FUN=sum)
# Boundary-crosser
bcr <- c(apply(bobj[,2:(nc-1)],MARGIN=2,FUN=sum),NA)
if (plot == TRUE){
  par(mar=c(3, 5, 4, 2) + 0.1)
  plot(gap, axes=F, xlab="", ylab="", pch="", main="Curvas de
Diversidade",...)
  axis(2,at=ceiling(seq(0,max(gap),length.out=10)))
  mtext(text="Número de táxons",side=2,line=3,cex=1.3)
  axis(1,at=1:(nc-1),labels=names(sib))
  lines(sib,col="green",lwd=2)
  lines(gap,col="red",lwd=2)
  lines(x=(1:(nc-2))+0.5,y=as.numeric(na.exclude(bcr)),col="blue",lwd=2)
  legend(x=(0.8*(nc-1)),y=max(gap),legend=c("sib","gap","bcr"),col=c("green","
red","blue"),lwd=2)
}
out <- rbind(sib,gap,bcr)
out
}

```

Código da Função 'branching.times.with.extinct'

```

#####
####                                     ####
####   Daniel Varajao de Latorre       ####
####                                     ####
#####
####   BRANCHING.TIMES COMPATIVEL COM EXTINÇÃO   ####
branching.times.with.extinct<-function(phy)
{
  n <- length(phy$tip.label)
  N <- dim(phy$edge)[1]
  xx <- numeric(phy$Nnode)
  interns <- which(phy$edge[, 2] > n)
  for (i in interns)
    xx[phy$edge[i, 2] - n] <- xx[phy$edge[i,1] - n] + phy$edge.length[i]
  ##parte nova, novo algoritmo para calcular qual o tempo maximo da arvore
  ntip<-Ntip(phy)
  sum<-numeric(ntip)
  index<-numeric()
  x<-numeric()
  for(i in 1:ntip)
  {
    node<-i

```

```
while(node!=ntip+1)
{
  index<-which(phy$edge[,2]==node)
  sum[i]<-sum[i] + phy$edge.length[index]
  node<-phy$edge[index,1]
}
}
depth <- max(sum)
##volta a ser usado o codigo da funÃ§Ã£o original
xx <- depth - xx
names(xx) <- if (is.null(phy$node.label))
  (n + 1):(n + phy$Nnode)
else phy$node.label
xx
}
```

Arquivos para download

[divercurve.r](#)

[branching.times.with.extinction.r](#)

[Carnivora_genus.csv](#)

[Raia_et_al_2013_tree.txt](#)

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

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:mauro.sugawara:start

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

