

# Hamanda Cavalheri



Sou mestranda em Ecologia, USP, trabalho no laboratório de ecologia, evolução e conservação de vertebrados, orientada por Marcio Martins. Minha pesquisa busca entender como é a estrutura filogenética e fenotípica de comunidades de serpentes do Brasil e quais fatores interferem nessa estrutura.

exec

## Função - Propostas

### Plano A Estrutura filogenética de comunidades

A estrutura filogenética de comunidades pode ser influenciada por dois processos: filtro ambiental ou competição. Se o processo principal for filtro ambiental, então a estrutura filogenética que encontramos será agregada, mostrando que o ambiente impede que algumas espécies ocorram na comunidade por não possuírem os atributos necessários. Portanto, esperamos que espécies próximas filogeneticamente partilhem os atributos requeridos pelo ambiente. No entanto, se o processo for competição, então a estrutura filogenética será dispersa, indicando que espécies próximas e similares no uso de recursos competem e não ocorrem juntas na comunidade. Existem dois índices que calculam a estrutura filogenética de comunidades, os quais foram propostos por Webb et al. 2002. Para calcular os índices é necessário ter uma filogenia do pool regional de espécies e um data.frame com resposta binária para espécies do pool que ocorrem na comunidade. O primeiro índice é NRI ("net relatedness index") é uma medida padronizada da média da distancia filogenética (número de nós) entre par de espécies. O segundo índice é o NTI ("nearest taxa index") que é uma medida padronizada da distancia filogenética de uma espécie em relação a mais próxima. NRI está relacionado a padrões de agregação ou dispersão em toda filogenia, enquanto o NTI é mais sensível a padrões próximos aos terminais. Para saber se o padrão encontrado é diferente do esperado pelo acaso é possível comparar o padrão encontrado com um modelo nulo, o qual será obtido pela randomização das espécies de uma comunidade nula gerada ao acaso no pool regional.

### Plano B Representação gráfica da diferença de tamanho entre comunidades

Uma análise já realizada na minha dissertação mostra que existe diferença de tamanho do corpo entre as espécies das comunidades estudadas. Portanto, proponho fazer uma função em que nos terminais das filogenias aja uma marcação indicando o tamanho do corpo da espécie. A fim de comparar o tamanho entre as comunidades dividiria o tamanho do corpo em classes de tamanho utilizando uma escala de cinza para marcar as classes. Quanto maior o tamanho mais escura a marcação no terminal da espécie. Como existem espécies sem dados morfológicos incluiria uma marcação especial para indicar que aquela espécie ocorre na comunidade. Assim, as figuras geradas mostrariam o qual o pool regional de espécies para cada comunidade, além das espécies que ocorrem e a classe de tamanho delas.

---

Tenho dados para oito comunidades de serpentes do Brasil, incluindo as filogenias dos pools regionais

de espécies. Além disso, se for possível poderia criar uma figura para a primeira proposta usando a ideia da segunda de mostrar o pool regional e as espécies que ocorrem nas comunidades.

Webb, C.O.; Ackerly, D.D.; McPeck, M.A. & Donoghue, M.J. Phylogenies and Community Ecology. *Annu. Rev. Ecol. Syst.* , 33(2002), pp. 475-505.

## Comentários

### PI

As duas propostas são muito boas. A segunda depende de vc conhecer as funções de R que geram cladogramas, e como alterar estes gráficos, que não ensinamos. Se está á vontade com isto, ok.

Já a primeira pode ser feita com o que ensinamos na disciplina. O modelo nulo não ficou muito claro. Seria um sorteio do número observado de espécies da comunidade, sem reposição? Antes de começar, defina claramente o que entra e o que sai de sua função, que são as seções 'arguments' e 'value'.

### Sara

Muito legal a proposta!

Veja o pacote `picante` para as medidas de estrutura filogenética, tem algumas randomizações de árvores e espécies em localidades, mas você pode trabalhar para modificá-las. Pode ser um bom caminho. Para o modelo nulo vale pensar no modelo browniano, pois os modelos já implementados são baseados em randomizações dos braços sem levar em conta qualquer modelo de evolução. Talvez complique um pouco... mas vamos vendo!

Para plotar filogenias tem o pacote `ape`. Na documentação dos dois pacotes tem exemplos legais de plot de árvores a partir dos quais você pode elaborar sua função.

Recomendo você dar uma olhada no [livro do Emanuel Paradis](#), um dos desenvolvedores do `ape`. O livro combina o básico de manipulação de filogenias e método comparado no R, com exemplos de códigos e tudo.

## Função - Final

### HELP

`graph.phylo`  
R Documentation

package: unknown

Representação gráfica da diferença de atributos entre comunidades

Description

Esta função retorna uma filogenia com informações contínuas ou categóricas para cada terminal.

#### Uso

```
graph.phylo(x, y, phy, bks="FD", color="gray",
            direction="rightwards", type="phylogram",
            legend="sem dados", position1="topleft",
            position2="bottomleft", show.tip.label=TRUE,
            label.offset=0.3, cex.tip=3, pch.tip=19,
            cex.leg=1, pch.nd=20, cex.nd=1.5, cex.leg.nd=1,
            bty="n", col.nd="black", show.node.label=FALSE,
            edge.color="black", edge.width=1,
            edge.lty=1, font=3, no.margin=TRUE)
```

#### Argumentos

**x** data frame que contém na primeira coluna o nome dos terminais e nas seguintes as variáveis (as quais podem ser da classe "numeric" ou "factor").

**y** o número da coluna que corresponde a variável que será mostrada nos terminais da filogenia.

**phy** objeto da classe "phylo".

**bks** divide uma variável contínua em classes segundo a fórmula de Sturges, Scott ou Freedman-Diaconis. É especificado como: "Sturges", "Scott" e "FD", mas aceita também um vetor com as quebras das classes.

**color** indica qual escala de cor será usada para mostrar a variável nos terminais da filogenia. As opções são: "gray" (default), "heat" e "rainbow".

**direction** especifica a direção da filogenia. São quatro possibilidades: "rightwards" (default), "leftwards", "upwards" e "downwards".

**type** especifica o tipo de filogenia que será desenhada, deve ser uma das seguintes opções: "phylogram" (default), "cladogram", "fan", "unrooted" ou "radial".

**legend** especifica o que aparecerá na legenda dos terminais que não contém dados.

**position1** posição da legenda relativa aos dados mostrados nos terminais da filogenia.

**position2** posição da legenda relativa aos terminais sem dados.

**show.tip.label** argumento lógico que indica se os nomes dos terminais devem aparecer na filogenia (default é TRUE, i.e. os nomes serão mostrados).

**label.offset** argumento numérico que indica o espaço entre os nós e os terminais da filogenia e os nomes dos terminais correspondentes. Esta opção não tem efeito se o type = "unrooted".

**cex.tip** argumento numérico que indica a escala dos símbolos dos terminais da filogenia.

**pch.tip** argumento numérico que fornece o tipo de símbolo que será mostrado nos terminais da filogenia.

**cex.leg** argumento numérico que fornece a escala da legenda dos dados.

**pch.nd** argumento numérico que indica o tipo de símbolo será

usado para os terminais que não possuem dados.

`cex.nd` argumento numérico que indica a escala dos símbolos usados para indicar terminais sem dados.

`cex.leg.nd` argumento numérico que indica a escala da legenda relativa aos terminais sem dados.

`bty` o tipo de caixa que será desenhada ao redor da legenda. Os valores permitidos são: "o" (default) e "n".

`col.nd` cor atribuída ao símbolo dos terminais sem dados.

`show.node.label` argumento lógico que indica se os nós da filogenia devem ser mostrados (default é FALSE, i.e. os nós não serão mostrados).

`edge.color` um vetor de caracteres que fornece as cores que serão usadas nos ramos da filogenia. As cores devem estar na mesma ordem que os componentes `edge` da filogenia. Se poucas cores forem fornecidas, elas serão recicladas.

`edge.width` um vetor numérico que indica a largura da linha dos ramos da filogenia. As cores devem estar na mesma ordem que os componentes `edge` da filogenia. Se poucas larguras forem fornecidas, elas serão recicladas.

`edge.lty` o mesmo que o argumento anterior, mas para tipos de linhas. As opções são 1: contínua, 2: tracejada, 3: pontilhada, 4: pontos e traços, 5: tracejado longo, 6: dois tracejados.

`font` argumento numérico que especifica o tipo de fonte usada nos nós e nos terminais se estes forem TRUE. As seguintes opções são permitidas: 1 (texto simples), 2 (negrito), 3 (itálico, default) ou 4 (negrito itálico).

`no.margin` argumento lógico. Se TRUE, as margens serão zero e a figura usará todo o espaço do dispositivo gráfico.

## Detalhes

Este gráfico é útil para comparar atributos entre comunidades, pois permite usar filogenias de pool regional de espécies, onde NA indica espécies que não ocorrem na comunidade e as espécies que ocorrem são reconhecidas pela presença de dados. Além de permitir, também, que espécies que ocorrem na comunidade, mas que não possuem dados sejam marcadas pela sua ocorrência. Estas espécies devem ter os dados das variáveis contínuas e categóricas como zero.

Outro aspecto importante na comparação entre atributos para comunidades diferentes é que a função permite escolher as quebras das classes das variáveis contínuas. Então, o usuário pode escolher classes das variáveis contínuas de acordo com a amplitude de todas as espécies de todas as comunidades. Isso é possível porque a função não possui restrição de quebra de classes de acordo com o mínimo e o máximo dos dados. As variáveis categóricas devem ser especificadas por diferentes letras e não números.

Para variáveis categóricas a escala de cor "heat" e "rainbow" retorna uma escala de cores sem padrão.

Se a variável escolhida for categórica, então, independente de qualquer especificação no argumento `bks`, a função usará os níveis dos fatores nos terminais da filogenia, os quais estarão especificados na legenda.

Se em `bks` for fornecido um vetor e os valores das quebras das classes não

abrangerem o máximo valor dos dados, então os valores que não estiverem compreendidos nas classes serão especificados como da classe que contém os maiores valores.

### Valores

graph.phylo retorna ao usuário uma figura no dispositivo gráfico ativado que contém a filogenia com a variável escolhida evidenciada nos terminais, de acordo com as especificações de cor, tamanho e tipo. Além de legendas que especificam as classes das variáveis contínuas e os níveis dos fatores das variáveis categóricas.

### Cuidado

É necessário instalar e carregar na área de trabalho o pacote ape para usar a função graph.phylo.

### Autor

Hamanda Badona Cavalheri (hbcavalheri@gmail.com).

### Ver também

plot.phylo, tiplabels for the package ape.

### Exemplos

```
phy <- read.tree(text = "((sp1,sp2),(((sp3,sp4),sp5),(((sp6,sp7),sp8),
(sp9,sp10)))));")
phy
plot(phy)
tam <- c(100,0,84,NA,67,NA,44,58,0,60)
sp <- c("sp1", "sp2", "sp3", "sp4", "sp5", "sp6", "sp7", "sp8", "sp9",
"sp10")
habitat <- c("F",0, "A", NA, "A", NA, "Q", "F", 0, "Q")
tam <- data.frame(tam, habitat)
rownames(tam) <- sp
graph.phylo(tam,1,phy) ### dados contínuos ###
graph.phylo(tam,2,phy, color="heat") ### dados categóricos ###
graph.phylo(tam,1,phy, bks=c(43,66,87,100), col.nd="green",
type="cladogram") ### definindo a quebra das classes ###
```

## FUNÇÃO - CÓDIGO

```
graph.phylo <- function(x, y, phy, bks="FD",color="gray",
direction="rightwards", type="phylogram",
legend="sem dados", position1="topleft",
position2="bottomleft", show.tip.label=TRUE,
label.offset=0.3, cex.tip=3, pch.tip=19,
cex.leg=1, pch.nd=20, cex.nd=1.5, cex.leg.nd=1,
bty="n", col.nd="black", show.node.label=FALSE,
edge.color="black", edge.width=1,
edge.lty=1, font=3, no.margin=TRUE)
{
direction <- match.arg(direction, c("rightwards", "leftwards", "upwards",
"downwards"))
```

```
color <- match.arg(color, c("gray", "heat", "rainbow"))
type <- match.arg(type, c("phylogram", "cladogram", "fan",
                        "unrooted", "radial"))
position1 <- match.arg(position1, c("bottomright", "bottom", "bottomleft",
"left", "topleft", "top", "topright", "right", "center"))
position2 <- match.arg(position2, c("bottomright", "bottom", "bottomleft",
"left", "topleft", "top", "topright", "right", "center"))
if(class(phy)== "phylo"){
  if(class(x[,y])=="numeric")
  {
    x1 <- x[-which(x==0),y]
    x1 <- na.exclude(x1)
    if(is.character(bks))
    {
      if(bks=="FD") ## FD
      {
        res <- hist(x1, breaks=nclass.FD(x1))$breaks
      }
      if(bks=="Sturges") #Sturges
      {
        res <- hist(x1)$breaks
      }
      if(bks=="Scott") #Scott
      {
        res <- hist(x1, breaks=nclass.scott(x1))$breaks
      }
    }
    if(is.numeric(bks)) #vetor com numeros
    {
      res <- bks
    }
    x <- data.frame(x, "class"=rep(NA, times=dim(x)[1]))
    for(i in 1:(length(res)-1))
    {
      x$class <- replace(x$class, list=(res[i]<=x[,y] & x[,y]<res[i+1]),
length(res)-i)
      x$class <- replace(x$class, list=(range(res)[2]<=x[,y]), 1)
    }
    pas <- c(rep(NA, length(res)))
    for(l in 1:length(res))
    {
      pas[l] <- paste(res[l],res[l+1],sep="-")
    }
    pas <- pas[-length(pas)]
    if(color=="gray")
    {
      plot.phylo(phy, show.tip.label=show.tip.label,
label.offset=label.offset, direction=direction,
type=type,show.node.label=show.node.label,
                        edge.color=edge.color, edge.width=edge.width,
                        edge.lty=edge.lty, font=font, no.margin=no.margin)
```

```

    for(i in 1:dim(x)[1])
    {
        tiplabels(tip=i, pch=pch.tip, cex=cex.tip, bg=gray(seq(0.1, 0.9,
length=length(pas)))[x[i,"class"]],
                col=gray(seq(0.1, 0.9, length=length(pas)))[x[i,
"class"]])
    }
    legend(position1,legend=pas[length(pas):1], col=gray(seq(0.1, 0.9,
length=length(pas))), pch=pch.tip, cex=cex.leg, bty=bty)
}
if(color=="heat")
{
    plot.phylo(phy, show.tip.label=show.tip.label,
label.offset=label.offset, direction=direction,
type=type,show.node.label=show.node.label,
            edge.color=edge.color, edge.width=edge.width,
            edge.lty=edge.lty, font=font, no.margin=no.margin)
    for(i in 1:dim(x)[1])
    {
        tiplabels(tip=i, pch=pch.tip, cex=cex.tip,
bg=heat.colors(length(pas))[x[i,"class"]],
                col=heat.colors(length(pas))[x[i,"class"]])
    }
    legend(position1,legend=pas[length(pas):1],
col=heat.colors(length(pas)), pch=pch.tip, cex=cex.leg, bty=bty)
}
if(color=="rainbow")
{
    plot.phylo(phy, show.tip.label=show.tip.label,
label.offset=label.offset, direction=direction,
type=type,show.node.label=show.node.label,
            edge.color=edge.color, edge.width=edge.width,
            edge.lty=edge.lty, font=font, no.margin=no.margin)
    for(i in 1:dim(x)[1])
    {
        tiplabels(tip=i, pch=pch.tip, cex=cex.tip,
bg=rainbow(length(pas))[x[i,"class"]],
                col=rainbow(length(pas))[x[i,"class"]])
    }
    legend(position1,legend=pas[length(pas):1],
col=rainbow(length(pas)), pch=pch.tip, cex=cex.leg, bty=bty)
}
for (m in which(x[,y]==0))
{
    tiplabels(tip=m, pch=pch.nd, cex=cex.nd, col=col.nd, bg=col.nd)
}
legend(position2,legend=legend, col=col.nd, pch=pch.nd,
cex=cex.leg.nd, bty=bty)
}
if(class(x[,y])=="factor")
{ x1 <- which(x[,y]!=0)

```

```
if(color=="rainbow"|color=="heat")
{
  plot.phylo(phy, show.tip.label=show.tip.label,
label.offset=label.offset, direction=direction,
type=type,show.node.label=show.node.label,
          edge.color=edge.color, edge.width=edge.width,
          edge.lty=edge.lty, font=font, no.margin=no.margin)
  for(i in x1)
  {
    tiplabels(tip=i, pch=pch.tip, cex=cex.tip, bg=unique(x[i,y]),
              col=unique(x[i,y]))
  }
  legend(position1,legend=unique(x[which(x[,y]!=0),y]),
col=match(unique(x[which(x[,y]!=0),y]), names(table(x[,y]))), pch=pch.tip,
cex=cex.leg, bty=bty)
}
if(color=="gray")
{
  plot.phylo(phy, show.tip.label=show.tip.label,
label.offset=label.offset, direction=direction,
type=type,show.node.label=show.node.label,
          edge.color=edge.color, edge.width=edge.width,
          edge.lty=edge.lty, font=font, no.margin=no.margin)
  for(i in x1)
  {
    tiplabels(tip=i, pch=pch.tip, cex=cex.tip, bg=gray(seq(0.1, 0.9,
length=length(unique(x[,y]))))[x[i,y]],
              col=gray(seq(0.1, 0.9,
length=length(unique(x[,y]))))[x[i,y]])
  }
  legend(position1,legend=unique(x[which(x[,y]!=0),y]),
col=gray(seq(0.1,0.9,length=length(unique(x[,y]))))[match(unique(x[which(x[,y]!=0),y]), names(table(x[,y])))]), pch=pch.tip, cex=cex.leg, bty=bty)
}
for (m in which(x[,y]==0))
{
  tiplabels(tip=m, pch=pch.nd, cex=cex.nd, col=col.nd, bg=col.nd)
}
legend(x=position2,legend=legend, col=col.nd, pch=pch.nd,
cex=cex.leg.nd, bty="n")
}
}
else
{
  cat("objeto phy não é da classe phylo")
}
}
```

[funcao\\_hamanda.r](#) [help\\_hamanda.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:hamandabcavalheri:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:hamandabcavalheri:start) 

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