

Juliete Costa de Oliveira



Doutoranda em Ecologia pelo Instituto de Biociências, USP. Integrante do Laboratório de Ecologia e Evolução (EACH).

Minha tese é sobre “Montagem de comunidades em poças temporárias artificiais”, sob orientação do Profº. Dr. Luís Schiesari.

Meus Exercícios

[Exercícios](#)

Proposta de Trabalho Final

PLANO A

Função “edge.effect” - Relacionando Efeito de borda e Diversidade de espécies

Contextualização

Conservar a biodiversidade tem sido uma das grandes preocupações deste milênio, cenário que vem se intensificando nos últimos anos, devido à elevação das ações antrópicas nos ambientes naturais e uma das consequências mais dramáticas das mudanças no uso do solo é a fragmentação da paisagem (Tabarelli & Gascon, 2005).

A fragmentação do habitat gera perdas das funções ecológicas pela diminuição dos habitats contínuos e pelo aparecimento de um mosaico composto por ambientes menores, o que resulta, além da redução em tamanho de uma mancha florestal, o aumento na quantidade de ambientes de transição entre o centro do fragmento e sua matriz circundante e um aumento do isolamento entre populações, influenciando na interação da diversidade local e afetando processos ecológicos entre populações e comunidades (Metzger, 2009).

O efeito de borda é talvez uma das consequências da fragmentação mais conhecidas. Ela gera uma zona entre o interior e a matriz que circunda o fragmento que tem grande influência sobre a qualidade do fragmento para as espécies que ali vivem.

Diante disso, a função infracitada tem como objetivo analisar a relação entre a diversidade de espécies e o efeito de borda, aqui obtido pela seguinte fórmula ($c = \text{perímetro} / \sqrt{\pi \cdot \text{área}}$).

Delineamento

* Informar para o usuário que a função depende da instalação prévia do pacote “Vegan”.

Pseudocódigo

Entrada

edge.effect (dados1, dados2, diversity = “shannon”, patton(c=perímetro / sqrt (pi*área), cor.test = “spearman”)

- dados1 = classe data.frame. Tabela com os dados de abundância das espécies (espécies nas colunas e localidades nas linhas).
- dados2 = classe data.frame. Tabela com os dados de perímetro (m) e área (ha) (perímetro e área nas colunas e localidades nas linhas).
- diversity = classe function. Retorna a diversidade (uma entre “shannon” e “simpson”).
- patton = classe formula. Retorna o índice de patton (efeito de borda).

cor.test = classe function. Retorna a correlação entre os dados (uma entre “pearson”, “spearman”, “kendall”).

Testes de premissa

- Se classe dos dados1 é diferente de data.frame, STOP: “Os dados devem estar num data.frame”.
- Se classe dos dados2 é diferente de data.frame, STOP: “Os dados devem estar num data.frame”.
- Se encontrar Na`s nos dados, Warning: Há Na`s nos seus dados. Os Na`s foram substituídos por 0”.
- Se valores de abundância não são números inteiros e se não é > ou = 0, STOP: “Os valores de abundância devem ser inteiros e > ou = 0”.
- Se valores de perímetro e área não são números > 0, STOP: “Os valores de abundância devem ser > 0”.
- Se índice de diversidade (diversity) é diferente de “shannon” ou “simpson”, STOP: “O índice de diversidade deve um entre “shannon” e “simpson”.
- Se teste de correlação (cor.test) é diferente de “pearson”, “spearman” ou “kendall”, STOP: “O teste de correlação deve ser um entre “pearson”, “spearman” ou “kendall”.

Pseudocódigo

1. Abrir uma janela gráfica usando X11().

2. Carregar o pacote “Vegan” usando função require.

3. Se diversity = “default” & cor.test = “default”,

- A. cria objeto “shannon” com o cálculo dos valores de diversidade de shannon usando os dados1.

- B. Cria um objeto “borda” com os valores do índice de efeito de borda, usando a fórmula ($c = \text{perímetro} / \sqrt{\pi \cdot \text{área}}$), usando dados2.
- Cria um objeto “cor.spearman” com o valor da correlação diversidade x borda.
- Plota um gráfico de “shannon~borda”.

4. Se diversity = “default” & cor.test = “pearson”,

- Repete passos A e B alterando os nomes dos objetos
- Cria um objeto “cor.pearson” com o valor da correlação diversidade x borda.
- Plota um gráfico de “shannon~borda”.

5. Se diversity = “default” & cor.test = “kendall”,

- Repete passos A e B alterando os nomes dos objetos
- Cria um objeto “cor.kendall” com o valor da correlação diversidade x borda.
- Plota um gráfico de “shannon~borda”.

6. Se diversity = “simpson” & cor.test = “default”

- A. cria objeto “simpson” com o cálculo dos valores de diversidade de simpson usando os dados1.
- B. Cria um objeto “borda” com os valores do índice de efeito de borda, usando a fórmula ($c = \text{perímetro} / \sqrt{\pi \cdot \text{área}}$), usando dados2.
- Cria um objeto “cor.spearman” com o valor da correlação diversidade x borda.
- Plota um gráfico de “simpson~borda”.

7. Se diversity = “simpson” & cor.test = “pearson”,

- Repete passos A e B alterando os nomes dos objetos
- Cria um objeto “cor.pearson” com o valor da correlação diversidade x borda.
- Plota um gráfico de “simpson~borda”.

8. Se diversity = “simpson” & cor.test = “kendall”,

- Repete passos A e B alterando os nomes dos objetos
- Cria um objeto “cor.kendall” com o valor da correlação diversidade x borda.
- Plota um gráfico de “kendall~borda”.

Saída

- Data.frame com os valores dos índices de diversidade e índice de patton.
- Gráfico da correlação “diversidade~borda” (x = borda, y = diversidade)

Referências

METZGER, J.P. (2009). **Conservation issues in the Brazilian Atlantic forest**. Biological Conservation, [s.l.], v. 142, n. 6, p. 1138-1140.

PATTON, D.R. (1975). **A diversity index for quantifying habitat “edge”**. Wildl. Soc. Bull. 3: 171-173.

TABARELLI, M.; GASCON, C. (2005). **Lições da pesquisa sobre fragmentação: aperfeiçoando políticas e diretrizes de manejo para a conservação da biodiversidade**. Megadiversidade, v. 1, n. 1, p. 181-188.

PLANO B

Função “diversity.index” – Índices de diversidade alfa, beta e gama

Contextualização

As diversidades alfa (diversidade local), beta (mudança de espécies ao longo de um gradiente ambiental) e gama (diversidade regional) são componentes espaciais da diversidade de espécies importantes para demonstrar a estrutura de uma comunidade e para compreender suas dinâmicas ecológicas.

A função infracitada objetiva facilitar o estudo das dinâmicas ecológicas de uma comunidade calculando a diversidade alfa, beta e gama de um conjunto de dados.

Delineamento

* Informar para o usuário que a função depende da instalação prévia do pacote “Vegan”.

Pseudocódigo

Entrada

diversity.index (dados, diversity.alfa = “shannon”, diversity.beta, diversity.gama)

- dados = Classe data.frame. Tabela com os dados de abundância das espécies.
- diversity.alfa = classe function. Retorna a diversidade alfa (uma entre “shannon” e “simpson”).
- diversity.beta = classe function. Retorna a diversidade beta (Índice de Jaccard).
- diversity.gama = classe formula. Retorna a diversidade gama (diversidade local x diversidade beta).

Testes de premissa

- Se classe dos dados é diferente de data.frame, STOP: “Os dados devem estar num data.frame”.
- Se encontrar Na`s nos dados, Warning: Há Na`s nos seus dados. Os Na`s foram substituídos por 0”.
- Se valores de abundância (dados) não são números inteiros e se não é > ou = 0, STOP: “Os valores de abundância devem ser inteiros e > ou = 0”.
- Se índice de diversidade alfa é diferente de “shannon” ou “simpson”, STOP: “O índice de diversidade alfa deve ser um entre “shannon” e “simpson”.

Pseudocódigo

1. Abrir uma janela gráfica usando X11().
2. Carregar o pacote “Vegan” usando função require.
3. Se diversity = “default”,
 - Cria objeto “shannon” com os dois valores de diversidade de shannon.
 - Cria um objeto as.matrix “Jaccard” com os valores do índice de similaridade de Jaccard.
 - calcula a matriz de distância (“vegdist”) e o cluster (“hclust”).
 - Plota um dendograma com a matriz de distâncias.
 - Cria um objeto “diversity.gama” com os valores de diversidade gama.
4. Se diversity = “simpson”,
 - Cria objeto “simpson” com os dois valores de diversidade de shannon.
 - Cria um objeto as.matrix “Jaccard” com os valores do índice de similaridade de Jaccard.
 - calcula a matriz de distância (“vegdist”) e o cluster (“hclust”).
 - Plota um dendograma com a matriz de distâncias.
 - Cria um objeto “diversity.gama” com os valores de diversidade gama.

Saída

- Data.frame com os valores dos índices de diversidade alfa e beta.
- Matriz com os valores de similaridade de Jaccard.
- Dendograma com a matriz de distâncias.

Referências

MAGURRAN, A.F. (2004). **Measuring Biological diversity**. Blackwell, Oxford.



— [Alexandre Adalardo de Oliveira](#) 2019/06/12 16:02 Oi Juliete,

As propostas me parecem um pouco básicas demais. A primeira, além disso, é bem restrita, um índice simples de borda e uma relação simples com a diversidade que será calculada pelo vegan. Além disso, tecnicamente não faz muito sentido comparar diversidade de fragmentos apenas baseado no efeito de borda, já que outros atributos do fragmento podem ser igualmente importantes ou mais que a borda. Por exemplo, o tamanho total ou o grau de isolamento. O que significa a relação de efeito de borda com diversidade para fragmentos de diferentes tamanhos e/ou graus de isolamentos? Aparentemente, não há um significado explícito, a menos que sejam comparados fragmentos de mesmo tamanho e com mesmo grau de isolamento, variando apenas a forma para que os fragmentos tenham índices de borda diferentes.

A segunda também é muito simples e está faltando uma informação importante na entrada dos dados, a localidade. A única forma de calcular beta diversidade é ter diferentes localidades para comparar a substituição de espécies (ou dissimilaridade). De qq forma, pelo que entendi, o cálculo em si dos índices seria feito pelas funções do vegan. Operar as funções de outro pacote apenas é uma proposta muito simples.

Já que está focada em diversidade, minha sugestão é:

Um dos problemas dos índices de diversidade é virem desacoplados de uma medida de precisão. É possível calcular uma variância associada, mas uma boa função poderia ser calcular o intervalo de confiança bootstrap para o índice de diversidade como Shannon e Simpson. Se fornecer o intervalo bootstrap dos índices na sua função, podemos fechar nela, mas acho que poderia se complicar. Então, minha proposta é focar em apenas uma escala de diversidade (alfa com Shannon e/ou Simpson) e calcular o intervalo de confiança dos índices.

Caso queira seguir outro caminho, modificando as propostas apresentadas, precisa colocar algo nelas que definam um desafio de programação suficiente.

DICAS para o intervalo bootstrap da diversidade

Fizemos intervalo bootstrap em aula em duas situações:

- No tutorial [bootstrap](#)
- Apresentei como exemplo na última aula sobre [função](#)

Isso vai ajudá-la a construir o intervalo de confiança para a sua função. Para o índice de Shannon/Simpson, precisa criar a reamostrar com reposição com o mesmo número de indivíduos da amostra (total de indivíduos), sendo que a probabilidade de cada espécie ser reamostrada depende da sua abundância. Para isso use a função `sample` para reamostrar o vetor de nome das espécies como dado, o `size` é o número total de indivíduos e usar o vetor de abundância no argumento `prob`.

Quando reorganizar a proposta me avise por email [Ale](#).

Escolha da função

Para a Função Final resolvi ficar com o Plano B com algumas modificações. O Plano B, inicialmente se propunha a criar uma função cuja finalidade seria devolver ao usuário, as diversidades alfa, beta e gama a partir de um `data.frame` de abundâncias das espécies por localidade. Na minha proposta de função final, fiquei apenas com a diversidade alfa, seguindo a orientação do prof^o Alexandre Adalardo. Adicionalmente, minha função também calcula o intervalo bootstrap para cada índice escolhido pelo usuário ("shannon" ou "simpson"). Ao final, a função devolve ao usuário um `data.frame` com os valores de índice de diversidade calculados para cada localidade, a média dos valores de índice calculados para cada reamostragem, o intervalo `qmin` e `qmax` da distribuição da frequência dos índices de diversidade.

Código da Função

```
boot.diversity <- function(dados, ind = c("shannon", "simpson"), qmin =  
0.025, qmax = 0.975, nboot = 1000) #Cria a função "boot.diversity"  
{  
  #TESTANDO AS PREMISSAS  
  
  if (class(dados) != "data.frame") #Verificando se os dados estão na classe  
correta  
  {stop ("Os dados devem estar num data.frame")} #Se os dados não estiverem  
na classe correta, interrompe a função e exibe a mensagem de erro  
  
  if (sum(is.na(dados)) != 0) #Verificando se os dados contém NA's  
  {warning ("Os dados continham NA's. Os NA's foram substituídos por 0 e  
todos os valores foram somados a 0,000000001 para serem operados")} #Se  
houver NA's nos dados, exibe mensagem para o usuário, mas não interrompe a  
função  
  
  if (sum(dados < 0, na.rm=TRUE) != 0) #Verificando se há dados negativos nos  
dados  
  {stop (" os valores de abundância devem ser maiores ou iguais a 0")} #Se  
houver dados negativos, interrompe a função e exibe a mensagem  
  
  if (ind != "shannon" & ind != "simpson") #Verificando se o usuário está  
escolhendo o índice corretamente  
  {stop("O índice de diversidade deve ser shannon ou simpson")} #Se o  
índice não estiver correto, interrompe a função e exibe a mensagem  
  
  if (qmin < 0 | qmax > 1 | qmax < qmin) #Verificando se os quantils mínimos  
e máximos estão dentro do intervalo correto para a função  
  {warning ("Quantils Incorretos. Calculando o Intervalo de Confiança  
bootstrap de 95%")  
  qmin = 0.025  
  qmax = 0.975  
  } #Se os quantils não estiverem corretos, exibe mensagem para o usuário,  
mas não interrompe a função  
  
  #LIDANDO COM OS NA's E OS 0's  
  
  if (sum(is.na(dados)) != 0) #Se houver NA's nos dados  
  {  
    dados[is.na(dados)] = 0 # onde, nos dados estiver NA, substitua por 0  
    dados <- (dados + 0.000000001) #Soma todos os valores dos dados por um  
valor infinitesimal (0.000000001) para conseguir operá-los dentro dos  
cálculos  
  }  
  #Lidando com os 0's  
  if(sum(dados == 0) != 0) #Se houver 0 nos dados  
  {  
    dados <- (dados + 0.000000001) #Soma todos os valores dos dados por um
```

valor infinitesimal (0.000000001) para conseguir operá-los dentro dos cálculos

```
}

#CALCULANDO ÍNDICE DE DIVERSIDADE "SHANNON"

if (ind == "shannon") #Se o usuário escolher o índice de shannon
{
  abund.rel <- matrix(NA, nrow=nrow(dados), ncol=ncol(dados)) #Criando um
objeto para guardar o resultado do "for" que irá calcular os valores das
abundâncias relativas com mesmo número de linhas e colunas dos dados
  abund.rel #Just checking...
  for(i in 1:nrow(dados)) #Ciclo em i para calcular abundâncias relativas
de todas as localidades
  {
    for(j in 1:ncol(dados)) #Ciclo em j para calcular abundâncias
relativas
    {
      a <- dados[i,j]/sum(dados[i,]) #Cálculo das abundâncias relativas
(abundâncias/somatório das abundâncias = abundâncias totais)
      abund.rel[i,j]=a #Guardando todos os resultados no meu objeto criado
inicialmente (data.frame preenchido com NA's com mesmo número de linhas e
colunas)
    }
  }
  abund.rel #Just checking...
  log.abund.rel <- log(abund.rel) #Calculo do ln (log natural) das
abundâncias relativas
  log.abund.rel #Just checking...
  abund.rel.log.abund.rel <- abund.rel*log.abund.rel #Multiplicando as
abundâncias relativas pelo log das abundâncias relativas
  abund.rel.log.abund.rel #Just checking...
  shannon <- - apply(abund.rel.log.abund.rel,1,sum) #Cálculo do índice de
shannon usando a formula  $H' = - \sum (n/N * \ln(n/N))$ 
  shannon #Checking

#BOOTSTRAP PARA "SHANNON"

boot.shannon <- matrix(data=NA,nrow=nboot,ncol=nrow(dados)) #Criando uma
matriz para guardar os índices de shannons das reamostragens
boot.shannon #Just checking...

for(j in 1:nboot) # "for" para cálculo das n matrizes de reamostragens
{

  repet <- vector() #Criando um vetor para alocar os índices de shannons
calculados das localidades reamostradas
  for(k in 1:nrow(dados)) # "for" para criar uma matriz de reamostragem
com os dados de entrada
  {
```



```

    A <- dados[k,]/sum(dados[k,]) #Criando vetor de probabilidades de
reamostragem
    linhas <- sample(x=dados[k,], size=ncol(dados), replace=TRUE,
prob=A) #Reamostrando os dados por localidade
    sh <-sum(linhas/sum(linhas)*log(linhas/sum(linhas))) #Calculando o
índice de shannon das localidades reamostradas
    repet[k] <- sh #Guardando o índice de shannon das localidades
reamostradas
  }
  boot.shannon[j,] <- repet #Guardando os cálculos dos índices de
shannons na matriz criada anteriormente
}
boot.shannon

B <-apply(boot.shannon,2,mean) #Calculando a média dos índices de cada
localidade
B
c <- vector() #Criando vetor para guardar o quantil qmin
d <- vector() #Criando um vetor para guardar o quantil qmax
for(i in 1:nrow(dados)) #Ciclo para calcular os quantis
{
  c[i] <- quantile(boot.shannon[,i],probs = qmin) #Calculando o quantil
qmin dos índices de shannons das reamostragens
  d[i] <- quantile(boot.shannon[,i],probs = qmax) #Calculando o quantil
q max dos índices de shannons das reamostragens
}

return(data.frame(shannon,Média.boot=B,ICinf=c,ICsup=d)) #Resultado da
função (um data.frame com os índices de shannons das localidades reais, as
médias dos índices de shannons das localidades reamostradas, o quantil
inferior e o quantil superior)
}

#CALCULANDO ÍNDICE DE DIVERSIDADE "SIMPSON"
else
{
  abund.rel.Q <- matrix(NA, nrow=nrow(dados), ncol=ncol(dados)) #Criando um
objeto para guardar o resultado do "for" que irá calcular os valores das
abundâncias relativas com mesmo número de linhas e colunas dos dados
  abund.rel.Q #Just checking...
  for(i in 1:nrow(dados)) #Ciclo para calcular abundâncias relativas por
localidade
  {
    for(j in 1:ncol(dados)) #Ciclo para calcular abundâncias relativas
    {
      a <- dados[i,j]/sum(dados[i,])^2 #Cálculo das abundâncias relativas
elevado ao quadrado (abundâncias/somatório das abundâncias = abundâncias
totais)^2
      abund.rel.Q[i,j]=a #Guardando todos os resultados no objeto criado
inicialmente (data.frame preenchido com NA's com mesmo n de linhas e

```

```
colunas)
}

}
abund.rel.Q #Just checking...
D <- apply(abund.rel.Q,1,sum) #Somando as abundâncias relativas para
calcular a Dominância em cada localidade
D #Just checking...
Simpson <- 1-D #Calculando 1 - os valores de dominância para cada
localidade para calcular a diversidade Simpson de acordo com a FÓRMULA  $1-D = 1 - \sum(n/N^2)$ 
Simpson #Checking

#BOOTSTRAP PARA "SIMPSON"

boot.simpson <- matrix(NA,nboot,nrow(dados)) #Criando uma matriz para
guardar os índices de simpsons das reamostragens
boot.simpson #Just checking...

for(j in 1:nboot) # "for" para
{

  repetit <- vector() #Criando vetor para guardar os índices de simpsons das
reamostragens
  for(k in 1:nrow(dados)) # for para criar uma matriz de reamostragem com
os dados
  {
    B <- dados[k,]/sum(dados[k,]) #Criando as probabilidades de
reamostragens das abundâncias
    linhas <- sample(x=dados[k,], size=ncol(dados), replace=TRUE, prob=B)
#Criando uma reamostragem dos dados de abundância das localidades
    simp <- 1-sum(linhas/sum(linhas)^2) #D =  $1 - \sum(n/N^2)$  #Calculando os
índices de simpsons as localidades reamostradas
    repetit[k] <- simp #Atribuindo os índices de simpsons para o vetor
criado anteriormente para entrar no "for" k
  }
  boot.simpson[j,] <- repetit # Atribuindo o vetor dos índices na matriz
criada para entrar no "for" j
}
boot.simpson #Just checking...

C <- apply(boot.simpson,2,mean) #Calculando a média dos índices de cada
localidade reamostrada
C ##Just checking...
e <- vector() #Criando um vetor para guardar o qmin
f <- vector() #Criando um vetor para guardar o qmax
for(i in 1:nrow(dados)) #"for" para calcular os quantis
{
  e[i] <- quantile(boot.simpson[,i],probs = qmin) #Calculando o quantil
qmin dos índices de simpsons das reamostragens
```

```
f[i] <- quantile(boot.simpson[,i],probs = qmax) #Calculando o quantil
qmax dos índices de simpsons das reamostragens
}

return(data.frame(Simpson,Média.boot=C,ICinf=e,ICsup=f)) #Resultado da
função (um data.frame com os índices de simpsons das localidades reais, as
médias dos índices de simpsons das localidades reamostradas, o quantil
inferior e o quantil superior da distribuição)

}
}
```

Página de Ajuda

Boot.diversity
R Documentation

package:unknown

BOOTSTRAP PARA ÍNDICES DE DIVERSIDADE

Description:

Função para calcular o intervalo Bootstrap dos índices de diversidade de Shannon e Simpson. Ao final, a função retorna um **data.frame** com os valores do índice escolhido (Shannon e Simpson) para cada localidade,