

# Fernanda Tibério



Sou doutoranda do Programa de Ecologia e Recursos Naturais, UFSCar.

Atualmente meu tema de pesquisa é a diversidade funcional de comunidade vegetais em estuários. Me interesso por diversos campos da ecologia geral e vegetal e por projetos de educação. Já trabalhei em projetos de ecologia de populações e comunidades vegetais, invasões biológicas e ecologia humana além de projetos de educação presencial e a distância.

Meu e-mail: nandatiberio@gmail.com

## Meus Exercícios

Para acessar meus exercícios resolvidos clique em [exec](#)



## Trabalho Final

### Plano A

Construir uma função que automatize o cálculo de estimativas de riqueza e de diversidade taxonômica, estimando intervalos de confiança para as estimativas a partir de bootstrap (reamostrando as parcelas). O input seria uma data.frame de abundância de espécies (plots x espécies). O output seria uma lista de valores e intervalos de confiança para cada uma das estimativas.

### Plano B

Construir uma função que execute uma simulação da relação entre duas medidas de diversidade funcional (DF-1: incluindo abundâncias; DF-2: excluindo abundâncias das espécies), de acordo com a variação o número de traços funcionais incluídos. O input seria um data.frame de abundância de espécies e de traços funcionais por espécie. O output seria o valor da medida de DF-1 e DF-2 para cada número de traços e/ou uma saída gráfica tipo plot(). Pensei numa variação pra este plano. Ao invés de comparar duas medidas de DF, simularia a relação entre DF (variando o números de traços) e diferentes riquezas da comunidade. pode ser mais interessante...

---

...ps.: Fui e voltei com as idéias e mesmo depois de conversar com os monitores e ser aconselhada por eles, cheguei em casa encontrei outros problemas. Acabei mudando os planos de novo! De qualquer forma, queria agradecer pela ajuda e me desculpar por não seguir com as idéias que

definimos. ...

## Comentários

Oi Fernanda

Seu plano A me parece bem simples e factível, mas não entendi bem o que você vai reamostrar no seu bootstrap. Você poderia detalhar um pouco mais essa proposta?

No plano B também faltou um pouco de detalhamento para entender sua proposta:

1) Não entendi direito como são seus dados de entrada... você teria dados para várias localidades? Ou nesse caso você está interessada em uma só localidade?

2) Você vai calcular as medidas DF1 e DF2 incluindo diferentes quantidades de traços funcionais, mas como vc vai determinar quais traços serão incluídos em cada calculo de DF? Por exemplo, na hora de calcular DF só com 2 traços, quais 2 você ia escolher? Imagino que não só QUANTOS, mas também QUAIS traços vc usa para calcular DF deve influenciar bastante no resultado de DF, não?

Resumindo, a proposta B ainda está bem confusa, mas acho que talvez ela seja mais interessante.

## Ale

Acho que entendi o bootstrap do plano A. Siga com esse, com a Debora comentou é factível, e apesar de simples um desafio interessantes. — [Alexandre Adalardo de Oliveira](#) 2013/03/24 21:58

---

Ok, sigo com o plano A então!

---

## Trabalho Final

### A função estima()

Esta função automatiza os cálculos de estimadores de riqueza e índices de diversidade e determina intervalos de confiança para cada valor através de bootstrap.

### Código da Função

```
estima=function(dados, n=1000 ,prob=c(0.025,0.975))
{
  #parametros para a funcao
  x=dados
```

```

linhas=1:nrow(x) #vetor com numero de parcelas (linhas)
abund=function(tab){apply(tab, 2, sum, na.rm=T)} #função para abundancias
(soma das colunas)
s.obs=function(vetor){length(vetor[vetor>0])} #numero de especies
observado
n.sp=function(vetor){sum(vetor, na.rm=T)}
p=function(vetor){na.omit(vetor)/sum(na.omit(vetor))} #função para pi
f1=function(vetor){sum(vetor==1, na.rm=T)}
f2=function(vetor){sum(vetor==2, na.rm=T)}

abundancias=matrix(data=NA, nrow=n, ncol=ncol(x), byrow=T,
dimnames=list(1:n, colnames(x))) #matriz para guardar as abundancias
(linhas) de cada reamostragem
for(i in 2:n)
{
  abundancias[1,]=abund(x)
  abundancias[i,]=abund(x[(sample(linhas, replace=T)),])
}
#RIQUEZA
#CHAO 1
c1=function(vetor){round(s.obs(vetor)+f1(vetor)*(f1(vetor)-1)/(2*(f2(vetor)+
1)), digits=4)}
c1.reamostra=apply(abundancias,1,c1)
quantis.c1=quantile(c1.reamostra, probs=prob)
r1=c("Chao 1", c1.reamostra[1], quantis.c1[1], quantis.c1[2])
#ACE abundancia
y=abund(x)
s.raro=function(vetor){length(vetor[vetor<=10])} #numero de especies
raras
s.abund=function(vetor){length(vetor[vetor>10])} #numero de especies
abundantes
n.raro=function(vetor){sum(vetor[vetor<=10],na.rm=T)} #numero de
indivíduos das spp raras
c.ace=function(vetor){1-(f1(vetor)/n.raro(vetor))} #proporção de
indivíduos que não são singletons
y.ace=function(vetor){max((s.raro(vetor)/c.ace(vetor))*((sum((vetor[vetor<=1
0])*((vetor[vetor<=10]-1))*((vetor[vetor<=10]/sum(vetor[vetor<=10])))))/(n.rar
o(vetor)*(n.raro(vetor)-1)-1))} #coeficiente de variação de ACE -
vetor=k=numero de indivíduos de cada espécie
ace.ab=function(vetor){round(s.abund(vetor)+(s.raro(vetor)/c.ace(vetor))+(f1
(vetor)/c.ace(vetor))*y.ace(vetor), digits=4)}
ace.ab.reamostra=apply(abundancias,1,ace.ab)
quantis.ace.ab=quantile(ace.ab.reamostra, probs=prob)
r2=c("ACE", ace.ab.reamostra[1], quantis.ace.ab[1], quantis.ace.ab[2])
#JACKKIFE 1A ORDEM - considera singletons
j.ab1=function(vetor){round(s.obs(vetor)+f1(vetor), digits=4)} #função
para jackknife
j.ab1.reamostra=apply(abundancias,1,j.ab1)
quantis.j.ab1=quantile(j.ab1.reamostra, probs=prob)
r3=c("Jackknife 1a ordem", j.ab1.reamostra[1], quantis.j.ab1[1],
quantis.j.ab1[2])

```

```
#JACKKNIFE 2A ORDEM
j.ab2=function(vetor){round(s.obs(vetor)+2*f1(vetor)-f2(vetor),
digits=4)}
j.ab2.reamostra=apply(abundancias,1,j.ab2)
quantis.j.ab2=quantile(j.ab2.reamostra, probs=prob)
r4=c("Jackknife 2a ordem", j.ab2.reamostra[1], quantis.j.ab2[1],
quantis.j.ab2[2])
#DIVERSIDADE
#Shannon
H=function(vetor) #função para calcular H de cada linha da matriz de
abundancias
{
  vetor=vetor[vetor>0]
  -round(sum(p(vetor)*log(p(vetor))), digits=4)
}
H.reamostra=apply(abundancias,1,H)
quantis.H=quantile(H.reamostra, probs=prob)
r5=c("Shannon (H)", H.reamostra[1], quantis.H[1], quantis.H[2])
#exp(H_shannon)=Hill - H2
H2.reamostra=round(exp(H.reamostra), digits=4)
quantis.H2=quantile(H2.reamostra, probs=prob)
r6=c("Exponencial de H", H2.reamostra[1], quantis.H2[1], quantis.H2[2])
#Simpson
D=function(vetor){round(1/(sum(p(vetor)^2)), digits=4)}
D.reamostra=apply(abundancias,1,D)
quantis.D=quantile(D.reamostra, probs=prob)
r7=c("Simpson", D.reamostra[1], quantis.D[1], quantis.D[2])
#Div. de Hurlbert
Dh=function(vetor){round(1-sum(p(vetor)*((vetor-1)/(n.sp(vetor-1)))),
digits=4)}
Dh.reamostra=apply(abundancias,1,Dh)
quantis.Dh=quantile(Dh.reamostra, probs=prob)
r8=c("D de Hurlbert", Dh.reamostra[1], quantis.Dh[1], quantis.Dh[2])
#Div.Margalef
s.mar=function(vetor){round((s.obs(vetor)-1)/log(n.sp(vetor)), digits=4)}
s.mar.reamostra=apply(abundancias,1,s.mar)
quantis.s.mar=quantile(s.mar.reamostra, probs=prob)
r9=c("Diversidade de Margalef", s.mar.reamostra[1], quantis.s.mar[1],
quantis.s.mar[2])
#Div.Menhinick
s.men=function(vetor){round(s.obs(vetor)/sqrt(n.sp(vetor)), digits=4)}
s.men.reamostra=apply(abundancias,1,s.men)
quantis.s.men=quantile(s.men.reamostra, probs=prob)
r10=c("Diversidade de Menhinick", s.men.reamostra[1], quantis.s.men[1],
quantis.s.men[2])
#Chao infinite sample
d.chao=function(vetor){round(s.obs(vetor)+(f1(vetor)^2)/(2*(f2(vetor))),
digits=4)}
d.chao.reamostra=apply(abundancias,1,d.chao)
quantis.d.chao=quantile(d.chao.reamostra, probs=prob)
```

```
r11=c("Diversidade de Chao", d.chao.reamostra[1], quantis.d.chao[1],
quantis.d.chao[2])
tab.rich=data.frame("Estimador"=rep(NA,5),"Valor
real"=rep(NA,5),"ICmin"=rep(NA,5),"ICmax"=rep(NA,5))
tab.rich[1,]=r1
tab.rich[2,]=r2
tab.rich[3,]=r3
tab.rich[4,]=r4
tab.rich[5,]=r11
tab.div=data.frame("Índice"=rep(NA,6),"Valor
real"=rep(NA,6),"ICmin"=rep(NA,6),"ICmax"=rep(NA,6))
tab.div[1,]=r5
tab.div[2,]=r6
tab.div[3,]=r7
tab.div[4,]=r8
tab.div[5,]=r9
tab.div[6,]=r10

lista=list(c("Para dados de abundância", "número de
espécies"=s.obs(abund(x)), "número de observações"=nrow(x)),
RIQUEZA=tab.rich, DIVERSIDADE=tab.div)
return(lista)
}
```

## Página de Ajuda

estima package:unknown R Documentation

Função para o cálculo de estimativas de riqueza e índices de diversidade.

### Description:

A função estima calcula os valores de estimadores de riqueza e índices de diversidade, e seus respectivos intervalos de confiança por bootstrap, a partir de dados de abundância de uma comunidade.

### Usage:

```
estima(dados, n, prob)
```

### Arguments:

**dados** data.frame com dados de abundância para espécies (colunas) em cada amostra (linhas).

**n** número de aleatorizações para cálculo dos intervalos de confiança por bootstrap. Default: n=1000

**prob** intervalo de confiança desejado. Default: prob=c(0.025,0.975).

## Details:

Os dados devem necessariamente ser inseridos com espécies nas colunas e amostras (ou plots) nas linhas. Os estimadores de riqueza usados são Chao 1, ACE, Jackknife 1a ordem, Jackknife 2a ordem e Chao para amostra infinita. Os índices de diversidade são Shannon (H), exponencial de Shannon ( $e^H$ ), Simpson (1/D), Hurlbert, Margalef e Menhinick.

## Value:

Se todos os dados forem de abundância, a função retorna uma lista com duas tabelas. A primeira mostra as riquezas estimadas para a comunidade e a segunda os valores de diversidade segundo cada índice.

## Note:

A descrição e o uso dos estimadores e índices utilizados podem ser obtidos em Magurran e McGill (2011).

## Author(s):

Fernanda C. S. Tibério  
(nandatiberio@gmail.com)

## References:

MAGURRAN, A. E. & MCGILL, B. J. 2011. Biological diversity : frontiers in measurement and assessment, Oxford ; New York, Oxford University Press.

## Examples:

```
estima(x, 10000, c=(0.025,0.975))
```

## Arquivo para download

### [Função estima](#)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2013:alunos:trabalho\\_final:nandatiberio:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:nandatiberio:start)

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