

Guillermo Flórez Montero

Oi Guillermo! Acho que ambas propostas são viáveis. Ambas estão implementadas em pacotes do R, mas isso não importa, contanto que o código apresentado seja inteiramente seu. Aconselho a não olhar o código das funções dos pacotes para não ficar induzido pela lógica alheia e seguir o seu próprio algoritmo. Ambas têm desafios interessantes. Se optar pela segunda deixe mais claro qual método utilizará para o cálculo da elasticidade. É possível calcular todas os parâmetros utilizando apenas soluções analíticas de álgebra linear (eigen analysis), outra solução é através de métodos de simulação (perturbação na matriz). Sugiro que opte pelo método de simulação, por ser um desafio mais interessante para a programação. Para fechar: defina qual caminho irá seguir A ou B; se for B descreva o método de simulação que irá utilizar brevemente. E principalmente, cuidado para não utilizar funções e códigos já escritos para esse mesmo propósito.

```
--- //[[aleadalar@gmail.com|Alexandre  
Adalardo de Oliveira]] 2016/04/30 08:48//
```

Obrigado prof. Alexandre e Bruno, vou seguir o plano A.

Guillermo



Biólogo (Universidad de Carabobo, Venezuela) mestrando do programa em Evolução e Diversidade da Universidade Federal do ABC. A minha dissertação é sobre a influência de variáveis antrópicas e ambientais na variação da riqueza de espécies e guildas a escala regional, orientado pelo Dr. Márcio de Souza Werneck.

MEUS EXERCÍCIOS

Exercícios 1: [ex1usp.r](#)

Exercícios 4: [exercicio_4.r](#)

Exercícios 5: [exercicios_5_guillermo.r](#)

Exercícios 6.2: [graficos_de_anova_guillermo.r](#)

Exercícios 7: [107.2.r](#)

Exercícios 8:

Exercícios 9: [ex.9.2.r](#)

PROPOSTA DO TRABALHO FINAL: Selecionado o Plano A

Plano A. Função para fazer análise de rarefação em ecologia de comunidades.

Determinar a riqueza de espécies depende do esforço de amostragem, geralmente a representação da riqueza em função do esforço é a través de uma curva de acumulação de espécies que contem o número de espécies observadas em cada amostra e acumuladas na amostra seguinte. Mas isso não é suficiente para determinar quantas espécies estão presentes num local, para isso geralmente se realiza uma análise de rarefação, que fornece uma estimativa da riqueza calculada a partir dos dados de uma amostra aleatória. O pacote Vegan tem a função `rarefy` que estimam a riqueza baseado na equação de Hurlbert (1971) e uma curva de acumulação de espécies (`rarecurve`). A minha proposta é gerar uma função para fazer um análise de rarefação baseado na matriz de incidência de espécies e gere um plot (curva de acumulação com curva de extrapolação e intervalos de confiança) segundo o modelo de Colwell *et al* (2005, *Ecology*, 85(10): 2717 - 2727).

O objeto de entrada será uma matriz de incidência de espécies, onde os nomes das colunas são as espécies da localidade e os nomes das linhas é a unidade de esforço de coleta, a matriz é binária (0 se a Espécie *i* na amostra *j* está ausente e 1 se está presente). A função pode aceitar NA na matriz.

O objeto de saída será uma lista que contem: a riqueza observada (valor único), a riqueza estimada pelo modelo (Valor único), além disso vai gerar um plot($y \sim x$) onde o eixo X é o número de amostras e o eixo Y é o número de espécies, uma linha preta vai representar a acumulação observada de espécies, enquanto que uma linha vermelha vai representar o número estimado pelo modelo, umas linhas pontilhadas vão representar os intervalos de confiança de 95% do estimador.

Comentários Bruno

Oi Guillermo, Sua função é muito similar com a função `specaccum` do pacote `vegan`. Acho que você poderia pensar em fazer algo a mais, por exemplo, incluir alguns índices de diversidade.

Bruno

Mais uma pergunta

Oi Bruno, muito obrigado pela sugestão. Sim acabei de notar isso, o argumento `method = "exact"` faz a curva com o método de Colwell *et al.* (2005), mas acho que não faz interpolação, só calcula a riqueza média (acabei de descobrir a função). Então poderia acrescentar a saída com mais alguns índices de riqueza mesmo para comparar? Nesse caso a matriz de entrada deve ser de abundância. Agora, não sei se se é muito simples, você que achou da B e qual sugere desenvolver?

Guillermo

Plano B. Função para analisar sensibilidade numa matriz populacional com estrutura.

Numa população com estrutura, a matriz de Leslie (estrutura com classes de idade) ou de Leftkovich (estádios de desenvolvimento) é uma projeção do ciclo de vida da espécie, que contém um resumo das taxas vitais dessa população (as contribuições de cada classe às outras no tempo discreto). A partir dessa matriz é possível calcular a taxa intrínseca de crescimento da população (λ) e a estrutura estável das classes (W) que é a proporção de indivíduos de cada classe que se mantém estável no tempo.

O análise de sensibilidade permite determinar a importância dum parâmetro da matriz para a taxa de crescimento intrínseca (quanto sensível é λ a cada parâmetro da matriz); existe também uma análise de Elasticidade que é uma medida relativa (escalada) da Sensibilidade supondo que as taxas vitais são independentes entre elas. A proposta é gerar uma função para calcular as matrizes de sensibilidade e elasticidade a partir duma matriz de taxas vitais segundo Caswell (2001), a função estará condicionada com o argumento “análise” que terá dois valores possíveis (“sensibilidade” e “elasticidade”).

O objeto de entrada será uma matriz simétrica com valores de taxas vitais (Matriz de Leslie ou de Leftkovich) que não aceita NA.

O objeto de saída será uma lista que contém: o valor de λ , um vetor com os valores de W para cada classe (Estrutura estável) e a Matriz de Sensibilidade ou Elasticidade (segundo o argumento análise).

Código da função

```
##FUNÇÃO rarefacao()##

rarefacao = function(matriz, especies = "colunas", grafico = TRUE)
{
  matriz[is.na(matriz)] = 0 #A função pode lidar com Na, portanto devem ser
  substituidos por 0
  matriz[matriz > 1] = 1 #Se a matriz for de abundância deve ser
  substituida a matriz de incidência (só 0 e 1)
  if(especies == "linhas") #Nossa função pode aceitar matrizes com especies
  em linhas ou colunas,
  {
    #só é preciso especificar com o argumento
    "especies"
    com = t(matriz) #Se as espécies estiverem em linhas, é preciso
    transpor a matriz e se cria um objeto com a matriz.
  }
  else #Se não estão em linhas então assume o default =
  colunas então
  {
    com = matriz #Cria um objeto com a matriz original (espécies em
    colunas e amostras em linhas)
  }
  #Geramos os vetores para fazer a grafica de acumulação de especies
  tradicional, com os dados
  amostras = c(1:nrow(com)) # Cria um vetor com o numero de amostras.
```

```
cumula = apply(com, 2, cumsum) #Cria uma matriz que contem o cumulado de
especies por amostra.
Scumuladas = rowSums(cumula > 0) #Cria um vetor que soma as especies com
presença acumulada em cada amostra (>0),
#Nessa parte a função calcula os valores preditos para cada amostra,
segundo
#o modelo de Interpolação baseado em amostras, com o metodo de Colwell et
al. (2004) equação 5.
freq = apply(com, 2, sum) #soma de linhas, contem o numero de amostras nas
que esta presente cada especie.
contagem = rep(0, dim(com)[1]) #Um vetor para armazenar o contagem de
frequencias
for(i in 1:dim(com)[1]) #a contagem de frequencias depende do numero de
amostras então usamos for para contar a frequencia
{
  contagem[i] = sum(freq == i) #somando todas as frequencias que sejam
iguais (numero de especies em 1, 2, 3... n amostras)
}
#e armazenando no vetor na posição que
corresponde a i.
Sobs = sum(contagem) #O numero de especies observadas no conjunto
de amostras é a soma dos contagens.
#Calculando os coeficientes combinatórios: alpha
#alpha para cada amostra é função do numero de amostras e do cumulado de
amostras (h e j), portanto precisamos calcular uma matriz de valores de
alpha
#para h de 1 até o total de amostras (H) e j de 1 até o o total de
amostras (H).
alpha = matrix(0, nrow = dim(com)[1], ncol = dim(com)[1]) #Uma matriz de
tantas linhas e colunas como amostras no conjunto.
vetorj = c(1:dim(alpha)[2]) #Um vetor com o numero cumulado de amostras (j
= 1, 2, 3 até H amostras)
H = dim(alpha)[1] #Total de amostras
for(i in 1: dim(alpha)[1]) #Variando em h (linhas que representam cada
amostra do conjunto)
{
  alpha[i,] = ((factorial(H - i) * factorial(H - vetorj)) / (factorial(H -
i - vetorj) * factorial(H))) #O coeficiente alpha(i), para o vetor de j
amostras cumuladas
}
alpha[is.nan(alpha)] = 0 #O calculo anterior gera NaN quando h + j > H,
que segundo Colwell et al. (2004), alpha = 0 nesses casos.
betha = alpha%*%contagem #A somatoria de equacao implica uma multiplicacao
de matrizes.
Sinterpolacao = rep(0, dim(alpha)[1]) #Cria um vetor para armazenar a
riqueza interpolada para cada amostra
for(i in 1: dim(alpha)[1]) #Para cada amostra teorica do novo conjunto
{
  Sinterpolacao[i] = Sobs - betha[i] #aplicamos o modelo de interpolação
(Colwell et al. 2004 eq 5)
}
```

```

#Calculamos a variancia e os intervalos de confiança
Sest = Sobs + ((H - 1)*contagem[1]^2)/(2*H*contagem[2]) #Estimador Chao2 da
riqueza total (observada + não observada, Eq 7.)
var = rep(0, dim(alpha)[1]) #Vetor para armazenar a variancia
icinf = rep(0, dim(alpha)[1]) #vetor para o intervalo de confiança
inferior
icsup = rep(0, dim(alpha)[1]) #vetor para o intervalo de confiança
superior
for(i in 1: dim(alpha)[1]) #a variancia deve e calculada para cada valor
de riqueza interpolada
{
  var[i] = abs(sum(((1 - alpha[i])^2 * contagem[i]) -
(Sinterpolacao[i]/Sest))) #Calculo da variancia segundo a eq. 6 de Colwell
et al. (2004), valor absoluto por se tem valores negativos.
  icinf[i] = Sinterpolacao[i] - (1.96 * sqrt(var[i])) #Calculo do
intervalo de confiança inferior (95% de confiança z = 1.6)
  icsup[i] = Sinterpolacao[i] + (1.96 * sqrt(var[i])) #Calculo do
intervalo de confiança superior (95% de confiança z = 1.6)
}
#Criamos um dataframe com o resumo dos resultados, aos valores que se
graficarem colocamos 0 ao inicio para que a grafica comece em 0,0
Rarefacao = data.frame("amostras" = c(0, amostras), "Scumuladas" = c(0,
Scumuladas), "Sinterpolacao" = c(0, Sinterpolacao), "IC95.Inf" = c(0,
icinf), "IC95.Sup" = c(0, icsup), Sest, Sobs)
if(grafico == TRUE) #Se o argumento logico "grafico" for TRUE, vai gerar
um grafico
{
  x11() #Chamada de uma janela grafica.
  Y = Sobs + (Sobs/9) #Tive que criar esse valor para colocar de limite ao
exo Y porque dava problemas com riquezas grandes
  plot(Scumuladas~amostras, data = Rarefacao, bty = "l", pch = 16, xlim =
c(0, length(amostras)), ylim = c(0, Y), las = 1,
      ylab = "Numero de Especies ", xlab = "Numero de Amostras", family =
"serif") #Grafico dos pontos observados no conjunto empirico
  abline(h = Rarefacao$Sest, col= "black", lty = 3, lwd = 2) #Linha
pontada que contem o numero de Especies Estimado por Chao2
  par(new=TRUE) #Para sobrepor o grafico siguinte
  plot(Sinterpolacao~amostras, data = Rarefacao, type = "l", col = "red",
xlim = c(0, length(amostras)), ylim = c(0, Y),
      yaxt = "n", xaxt = "n", ylab = " ", xlab = " ") #Linha vermelha com
os valores interpolados pelo modelo.
  par(new=TRUE) #Para sobrepor o grafico siguinte
  plot(IC95.Inf~amostras, data = Rarefacao, type = "l", lty = 3, col =
"red", xlim = c(0, length(amostras)), ylim = c(0, Y),
      yaxt = "n", xaxt = "n", ylab = " ", xlab = " ") #Linha vermelha
pontada com o intervalo de confianca inferior
  par(new=TRUE) #Para sobrepor o grafico siguinte
  plot(IC95.Sup~amostras, data = Rarefacao, type = "l", lty = 3, col =
"red", xlim = c(0, length(amostras)), ylim = c(0, Y),
      yaxt = "n", xaxt = "n", ylab = " ", xlab = " ") #Linha vermelha
pontada com o intervalo de confianca superior

```

```
#Texto que indique que a linha horizontal representa o estimador Chao2
mtext("Sest (Chao2)", side = 2, las = 2, cex = 0.8, line = 0.4, at =
Sest, family = "serif")
return(Rarefacao) #0 resultado e o dataframe criado, alem da grafica.
}
else #Se o argumento grafico for FALSE (Se o usuario quer graficar ele os
resultados)
{
return(Rarefacao) #0 resultado e o dataframe criado, sem grafica.
}
}
}
##FIM!!!!
```

HELP da função

rarefacao package: nenhum R Documentation

DESCRIPTION:

Estima a riqueza de espécies pelo método de interpolação baseado em amostras a partir duma matriz de incidência.

USAGE:

```
rarefacao(matriz, especies = "colunas", grafico = TRUE)
```

ARGUMENTS:

matriz Matriz de incidência de espécies por local,

especies Posição dos nomes das espécies na matriz, "colunas" se as espécies estão nas colunas da matriz
"filas" se as espécies estão nas filas.

grafico Logico, TRUE para que a função grafique a curva de acumulação de especies, FALSE e a função só fornecerá os resultados numéricos da rarefação.

DETAILS:

Uma boa estimativa da riqueza de especies depende do esforço de amostragem. A representação da riqueza em função do esforço de amostragem é a traves duma curva de acumulação de especies que contem o numero de especies observadas em cada amostra e acumuladas na amostra seguinte. Más isso não é suficiente para determinar quantas especies estão presentes num local, para isso geralmente se realiza uma análise de rarefação, que fornece uma estimativa da riqueza calculada a partir

dos dados duma amostra aleatória. A função "rarefacao()" fornece um análise de rarefação baseado na matriz de incidência de e gere um plot (curva de acumulação, curva de rarefação e intervalos de confiança de 95%) segundo o modelo de Colwell et al (2004).

VALUES:

A função rarefacao gera um objeto de classe "data.frame" com os seguintes itens:

`amostras` Vetor que contem o numero de amostras da comunidade.

`Scumuladas` A riqueza (número de espécies) cumuladas por amostra, obtido do conjunto empírico.

`Sinterpolacao` A riqueza (número de espécies) por amostra, obtido por interpolação (Colwell et al. 2004, equação 5).

`IC95.Inf` Intervalo de confiança (inferior) de 95% para a riqueza interpolada (Colwell et al. 2004, equação 6).

`IC95.Sup` Intervalo de confiança (superior) de 95% para a riqueza interpolada (Colwell et al. 2004, equação 6).

`Sest` Riqueza total estimada pelo modelo Chao2 (Colwell et al. 2004, equação 7).

`Sobs` Riqueza total observada do conjunto de amostras.

Se o argumento `grafico = TRUE` gera um gráfico, com "amostras" no eixo X, "número de especies" no eixo Y e origem 0,0:

`Círculos pretos` Riqueza cumulada (`Scumulada`) do conjunto empírico.

`Linha vermelha sólida` Riqueza interpolada (`Sinterpolacao`) pelo modelo.

`Linhas vermelhas pontilhadas` Intervalos de confiança de 95%.

`Linha preta pontilhada` Riqueza estimada pelo modelo Chao2 (`Sest`)

WARNING:

Riquezas muito grandes podem gerar erros na saída gráfica, nesses casos o argumento `grafico = FALSE` é útil para desenvolver a gráfica passo a passo.

A função gera um `warnings()` devido a que no calculo, alguns NaN são obtidos, mas não representa um impedimento no funcionamento desta, já que são corrigidos antes de providenciar os resultados.

AUTHOR:

Guillermo Flórez Montero
gflorezmontero@gmail.com

REFERENCES:

Colwell RK, Mao CX, Chang J (2004) Interpolating, extrapolating, and comparing incidence-based species accumulation curves. *Ecology* 85:2717–27.

EXAMPLES:

```
library(vegan)
data(BCI)
rarefacao(BCI)
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:guillermo.florez:start 

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