Joyce Rocha Garcia

Mestranda em Zoologia pela Unesp-campus de Botucatu. Minha linha de pesquisa é Ecologia de Animais Aquáticos com ênfase em crustáceos Decapoda.

Exercícios

exec

Trabalho Final

PROPOSTA A: Análise estrutural de uma comunidade

A ideia:

Criar uma função que faça uma análise exploratória da estrutura de uma comunidade. O reconhecimento de padrões em comunidades ecológicas é o mais antigo e, ainda, persistente desafio para as ciências ecológicas. Uma função que una alguns dos índices e curvas mais representativos para análise de comunidades representa maior facilidade para execução de uma tarefa bastante comum no âmbito científico.

Embasamento teórico:

O ponto de partida para a análise estrutural da composição de uma comunidade é a riqueza (S) da mesma, ou seja, o número total de espécies dentro da unidade amostral. Outro ponto que deve ser destacado é sobre a diversidade da comunidade, sendo o índice de Shannon-Weiner o mais utilizado, por incorporar tanto a riqueza, quanto a equitabilidade. A equitabilidade expressa a maneira pela qual o número de indivíduos está distribuído entre as diferentes espécies, isto é, indica se as diferentes espécies possuem abundância (número de indivíduos) semelhantes ou divergentes. O índice mais comumente utilizado para expressar tal informação é o de Pielou.

O cálculo da riqueza puramente só nos permite tirar algumas conclusões quando em comparativo ao longo do período de coleta, por exemplo. Dessa forma, é interessante ainda apresentar a riqueza aparente de uma comunidade (conhecida como índice de Shannon-Hill), que permite obter maior número de informações. Outras análises pertinentes são em relação à dispersão e à dominância. O primeiro diz respeito à distribuição espacial dos indivíduos, podendo ser categorizada como aleatória, agrupada ou uniforme. Já a dominância nos permite entender se há preponderância de alguma espécie dentro da comunidade.

A representação gráfica é de grande importância para observação de padrões ou singularidades da comunidade. Por meio da representação gráfica de riqueza por classes de abundância ao longo do tempo (períodos de coleta), podemos identificar um possível estresse ambiental a que a comunidade está sujeita. Isso porque, geralmente, o aumento dos níveis de estresse ambiental está relacionado

ao decréscimo da diversidade, da riqueza específica e da equitabilidade, com consequente acréscimo da dominância. No entanto, pode ocorrer da diversidade aumentar com o aumento do nível de estresse. Partindo de um dado momento onde o nível de estresse é mínimo, a diversidade é reduzida devido à exclusão competitiva entre as espécies. Se o nível de estresse continua aumentando, a competição diminui, resultando em um aumento da diversidade.

Porém, se o distúrbio chega a níveis elevados, as espécies começam a ser eliminadas e então a diversidade diminui novamente. Deste modo, a diversidade é máxima em níveis intermediários de estresse. Assim, mudanças na diversidade só podem ser analisadas através de comparações entre períodos de tempo. E um último ponto importante a ser destacado são as curvas de dominância-k, através das quais se podem determinar maiores informações acerca da diversidade e dominância dentro da comunidade.

Execução da função:

- Os dados de entrada na função deverão ser a abundância das espécies de uma comunidade em cada amostra, estruturados em uma matriz de abundância;

- A função calculará a riqueza específica (S) e aparente, o índice de diversidade de Shannon-Weiner (H'), o índice de equitabilidade de Pielou (J'), o índice de dominância de Berger-Parker (d) e o índice de dispersão de Morisita (ID);

- A função ainda deverá plotar gráficos representativos: gráfico de dispersão (evidenciando a categoria em que a dispersão dos organismos se enquadra), gráficos dos índices de Pielou e Shannon ao longo do tempo, gráfico da riqueza por classes de abundância também, ao longo de um período de tempo e a curva de dominância-k;

- A função deverá retornar ao usuário o valor de todos os índices e riquezas calculados, possivelmente, em forma de tabela, assim como os gráficos citados acima.

PROPOSTA B: Estimativa de tamanho populacional

A ideia:

Criar uma função que execute o Método de Schnabel para estimação de tamanho populacional. A estimação populacional foi escolhida para a criação desta função, pois é amplamente utilizada para análise de dados populacionais em ecologia, podendo abranger, conjuntamente, as áreas de zoologia, botânica etc. Desse modo, obteremos uma função útil e geral.

Embasamento teórico:

Como todo estimador probabilístico, o Método de Schnabel é embasado no processo de captura-marcação-recaptura (CMR), sendo que, as premissas que compõem este método são:

-População constante (fechada), onde não há nascimento, morte, imigração ou emigração;

- A captura dos animais é ao acaso;

- A marcação não é perdida durante o processo de estudo;

- A marcação não influencia a probabilidade de captura.

Execução da função:

- Os dados de entrada para a função deverão ser os dias (ou meses) de coleta, o número de indivíduos capturados (Ct) e o número de recapturados (Rt), estruturados em um data frame;

- Inicialmente, a função calculará alguns parâmetros (por exemplo, o n° de indivíduos novos marcados, dentre outros), que servirão para a estimação do tamanho populacional (N), que no Método Schnabel, é obtida pela razão entre a somatória de alguns dos parâmetros calculados previamente;

- A função ainda deverá realizar um teste de regressão para averiguar se as premissas foram atendidas, apresentando, o gráfico correspondente;

- A função deverá retornar ao usuário o valor de N, a densidade populacional, o intervalo de confiança (máximo e mínimo) e a saída gráfica do teste de regressão;

- Observação: um dos argumentos da função deverá conter o valor de t-crítico, obtido por meio de uma tabela de valores de t-crítico a partir do grau de liberdade (é um procedimento simples que deverá ser executado pelo usuário antes de utilizar a função). Isso se torna necessário para o cálculo do intervalo de confiança.

> Joyce, sua proposta A é bastante interessante, factível e útil. Você parece compreender bem os objetivos e o funcionamento da função. A proposta B também é bem interessante, mas por lidar com estimação de parâmetros parece ser de execução mais complexa. As duas propostas são boas, no entanto a opção A é mais segura, enquanto a opção B seria mais ousada. Devido à complexidade da B, talvez a opção A seja mais interessante neste momento. Qualquer dúvida, entre em contato.
>
> —-*Glaucia Del-Rio*
>
> Concordo com a Gláucia, vá pelo caminho mais seguro.
>
> — *Alexandre Adalardo de Oliveira 2014/04/25 19:06*

Execução da Função

PROPOSTA A: Análise estrutural de uma comunidade

Página de Ajuda

```
 Comunidade                package: unknown                        R
Documentation
```

Structural Analysis Of Communities

Description

Comunidade function creates a structural analysis of population communities. Including
calculation of diversity index (Shannon-Weiner), equitability index (Pielou), dominance
index (Berger-Parker) and dispersion index (Morisita). This function still calculates
specific richness and apparent richness. Comunidade function creates graphics using calculated
data in a period of time.

Usage

```
comunidade(Directory,Header,Extension)
```

Arguments

Directory	The address of the working directory according to getwd() (see details).
Header	If Header is TRUE, this means that selected files on the working directory contains the names of the variables as its first line.
Extension	The name of the file(s) in the working directory (see details).

Details

This function was created to facilitate structural analysis of communities.
Comunidade function reads files saved in the working directory, what allows reading
largest files. These files must to be saved with the same names for a correct reading
by the function. But it's not necessary specify the number complementing the name of
the files (see examples). The files must contain columms for each sample and another columm
of the species found. The files still must contain the abundance of each species in each
sample analysed by the researcher. Files must be saved in txt format. Each file must
represent a period of time, in other words, file 1 contains data of the first period
of time(first day or first month or first year), file 2 contains data of the second, and so on.

Value

```
    Comunidade returns a list of calculated indexes and richnesses in the R-
console and some
    graphics output.

Warning

    If files were not saved as txt format, messages of error in file will be
appear.

Author(s)
    Joyce Rocha Garcia
    joyce_garcia@ibb.unesp.br

References

    GOMES,A.S. 2004. Análise de dados ecológicos. Universidade Federal
Fluminense, Niterói, p.30

Examples

#For example 1, save the files "arquivo1.txt", "arquivo2.txt" and
"arquivo3.txt" (below) in your working directory. Save files with these
names in txt format.
#Specifications of the file: Species:5, samples:4, periods:3.
files <- getwd()
comunidade(files,TRUE,"arquivo")

#For example 2, download and save the file "periodo1.txt" and "periodo2.txt"
(below) in the working directory. Save files with these names in txt format.
#Specifications of the file: Species:6, samples:2, periods:2.
files <- getwd()
comunidade(files,TRUE,"periodo")
```

Código da função

```
comunidade <- function(Directory, Header, Extension)#Função comunidade e os
argumentos da função (especificados no help)#
{
arqs  <- sort(list.files(path = Directory, pattern = Extension))#A função se
inicia chamando um conjunto de arquivos nomeados no diretório de trabalho do
usuário e especificados no último argumento da função. E estes arquivos
foram ordenados pelo comando sort#
Local <- paste(Directory,arqs,sep = "\\")#Concatenando os arquivos
selecionados no diretório e atribuindo ao objeto Local#
matrizes <- lapply(Local, read.table, header = Header)#Neste passo, os
arquivos em Local estão sendo lidos e atribuidos ao objeto matrizes#
n.tempos <- length(arqs)#Como cada arquivo corresponde a um período de tempo
(a ser determinado pelo usuário - meses, anos ou dias), ao objeto n.tempos,
está sendo atribuído o comprimento do objeto arqs, ou seja, o n° de arquivos
(que correspondem aos períodos de tempo determinados pelo
usuário)(explicação detalhada no help)#
```

```
    Funcaoauxiliar <- function(data.file)#Aqui se inicia uma função auxiliar
para o cálculo dos índices em cada um dos arquivos selecionados#
        {
            data.file <- data.file[order(data.file[,1]),]#Ordenando as
linhas do objeto data.file (argumento da função auxiliar)#
            data.file <- data.file[,2:ncol(data.file)]#Organizando o n° de
colunas do objeto data.file#
            Amostra <- seq(1,ncol(data.file))#O n° de colunas será o n° de
amostras. Criando uma sequência de 1 até o n° de colunas (já que é o usuário
que decide o n° de amostras), atribuindo ao objeto Amostra#
        ################################################
        #INICIANDO O CÁLCULO DO ÍNDICE DE SHANNON-WEINER#
        ################################################
        x <- data.file[!is.na(data.file) & data.file!=0]#Retirando os zeros
e os dados faltantes e colocando tudo num novo objeto chamado x#
        p.i <- x/sum(x)#todos os dados de x divididos pela somatória de x,
colocados no objeto p.i#
        H <- -sum(p.i*log(p.i,base = exp(1))) #Esta é a fórmula para o
cálculo do índice: negativo da somatória de p.i vezes o logaritmo de p.i na
base e (logaritmo neperiano)#

        ########################################
        #INICIANDO O CÁLCULO DO ÍNDICE DE PIELOU#
        ########################################
        St <- nrow(data.file)#Colocando o número de linhas do arquivo no
objeto chamado St. O número de linhas corresponde ao número total de
espécies#
        Hmax <- log(St)#Calculando o logaritmo de St e atribuindo ao objeto
chamado Hmax#
        J <- H/Hmax#Esta é a fórmula para o cálculo do índice: índice de
Shannon dividido pelo Hmax#
        ################################################
        #INICIANDO O CÁLCULO DO ÍNDICE DE BERGER-PARKER#
        ################################################
        spp <- numeric(0)#Criando um objeto numérico vazio chamado spp para
o retorno dos resultados do for que será realizado abaixo#
        B <- nrow(data.file)#Atribuindo o número de linhas ao objeto B. O
número de linhas corresponde ao número total de espécies#
        for(i in 1:B)#Iniciando o comando for em que a repetição irá de 1
até B. Ou seja, o número de espécies do arquivo#
            {
            spp[i] <- sum(data.file[i,])#O objeto spp conterá a somatória
das linhas do arquivo#
            }
        ordem.decresc <- sort(spp,decreasing=TRUE)#Colocando os resultados
do for em ordem decrescente e atribuindo ao objeto ordem.decresc#
        Nmax <- ordem.decresc[1]#Escolhendo o primeiro resultado do objeto
ordem.decresc, ou seja, o maior valor e atribuindo ao objeto Nmax#
        Ntotal <- sum(data.file)#O objeto Ntotal contém a somatória das
abundâncias, ou seja, o número total#
```

```
        d <- Nmax/Ntotal #Esta é a fórmula para o cálculo do índice. O maior
valor dividido pelo número total#
        ########################################
        #INICIANDO O CÁLCULO DO ÍNDICE DE MORISITA#
        ########################################
        ID <- numeric(0)#Criando um objeto numérico vazio chamado ID para o
retorno dos resultados do for que será realizado abaixo#
        for(i in 1:ncol(data.file))#Iniciando o comando for em que a
repetição irá de 1 até o número de colunas existentes no arquivo#
            {
            A <- ncol(data.file)#Atribuindo o número de colunas ao objeto A.
Ou seja, o número de amostras#
              xi <- data.file[,i]#Atribuindo as colunas do arquivo ao objeto
xi. Ou seja,o número de indivíduos em cada amostra#
            ID[i] <- A*(sum(xi^2)-sum(xi))/(sum(xi)^2-sum(xi))#Esta é a
fórmula para o cálculo do índice. Somatória do n° de indivíduos (xi) ao
quadrado menos somatória de xi dividido pel somatória de xi ao quadrado
menos a somatória de xi. Tudo isso multiplicado pelo n° de amostras#
            }

        ################################
        #INICIANDO O CÁLCULO DA RIQUEZA#
        ################################
        ##Para o cálculo da riqueza será necessário transformar uma matriz
de abundância em uma de ocorrência. É o que segue abaixo##
        data.fileVF <- data.file>1 #Atribuindo ao objeto data.fileVF, todos
os dados maiores que 1. Transformando-o em um vetor lógico#
        sumVF <- sum(data.fileVF)#Atribuindo ao objeto sumVF a somatória
destes dados contidos em data.fileVF#
        if(sumVF>0)#Se os dados contidos em sumVF forem maiores que zero...#
            {
            data.file[data.fileVF] <- 1 #...atribuiremos 1 aos mesmos#
            }
        S <- apply(data.fileVF,2,sum) #O apply nos retornará a somatória das
colunas, ou seja, o número de espécies em cada amostra#
        ########################################
        #INICIANDO O CÁLCULO DA RIQUEZA APARENTE#
        ########################################
        SH <- S^H #A riqueza aparente é calculada elevando-se a riqueza(S)
ao índice de Shannon(H)#
                Flag_Conversao <- max(data.file[data.fileVF]) #Aviso da
conversão de matriz em abundância em matriz de ocorrência. Sempre que foi
necessária esta conversão#
return(list(data.frame(Amostra,ID,S,SH,Flag=rep(Flag_Conversao,length(ID))),
data.frame(H,J,d)))#A função irá retornar uma lista que contém: um data
frame que contém os dados de amostra, ID,S,SH e um flag de aviso da
conversão citada imediatamente acima. A função ainda retorna um data frame
com o cálculo dos índices H,J e d#
        } #FINAL DA FUNÇÃO AUXILIAR#

######################
```

```
#ORGANIZANDO OS DADOS#
######################
#Os resultados serão organizados em blocos. Bloco A (results_1 e indices_1): contém os índices calculados para cada amostras (ID, H, SH). Bloco B (results_2 e indices_2): contém os índices calculados em as amostras (H, J, d)#

for (i in 1:n.tempos)#Iniciando o for para organização dos índices calculados em cada um dos arquivos selecionados, indo de 1 até n.tempos (explicado na 4ª linha da função)#
{
if (i == 1)#Quando i for igual a 1, ou seja, referindo-se ao arquivo 1, ocorrerá o que segue abaixo#
 {
  ###BLOCO A###
  resulta   <- Funcaoauxiliar(matrizes[[i]])#Os índices calculados na Funcaoauxiliar para o arquivo 1 contido no objeto matrizes estão sendo atribuidos ao objeto results#
  resulta_1 <- data.frame(resulta[1])#Transformando results em um data frame#
  indices_1 <- cbind(Tempo = rep(i,nrow(resulta_1)), #Os resultados dos índices estão sendo organizados em colunas. O objeto indices_1 contém os índices ID, S, SH e o flag da conversão citada anteriormente#
             Amostra = resulta_1[,1], #As amostras estão na coluna 1#
             ID = resulta_1[,2], #Os valores de ID estão na coluna 2#
             S = resulta_1[,3], #Os valores de S estão na coluna 3#
             SH = resulta_1[,4], #Os valores de SH estão na coluna 4#
             Flag = resulta_1[,5]) #O flag está na coluna 5#
  row.names(indices_1) <- c(rep(gsub(".txt","",arqs[i]),nrow(resulta_1)))#Retirando o complemento ".txt" dos arquivos contidos em arqs para posterior nomeação das linhas dos resultados do bloco A#

  ###BLOCO B###
  resulta_2 <- data.frame(resulta[2])#O objeto resulta_2 contém o cálculo dos índices H,J e d em um data frame. Os resultados precisam ser divididos em 2 blocos, uma vez que o bloco A de índices é calculado para CADA amostra, enquanto o B não#
  indices_2 <- cbind(Tempo = i,H = resulta_2[,1],J = resulta_2[,2],d = resulta_2[,3]) #O bloco B de índices está sendo organizado em colunas, sendo o tempo o valor do contador. Objeto indices_2 contém os índices H, J e d#
  row.names(indices_2) <- c(gsub(".txt","",arqs[i]))#Retirando o complemento ".txt" dos arquivos contidos em arqs e atribuindo ao row.names(indices_2)#
    } #FIM DO PRIMEIRO IF#
  if(i > 1) #Quando o i for maior que 1, ou seja, os demais arquivos, ocorrerá o que segue abaixo#
    {
  ###BLOCO A###
  resulta <- Funcaoauxiliar(matrizes[[i]])#Os índices calculados na Funcaoauxiliar para os demais arquivos contidos no objeto arq.organizado
```

```
estão sendo atribuidos ao objeto results#
  resulta_1 <- data.frame(resulta[1])#Transformando results em um data
frame#
  nome.col <- c(row.names(indices_1))#Acrescentando indices_1 (criado acima)
ao objeto nome.col. Isso porque há 2 blocos de cada, que serão unidos no
final#
  indices_1 <- cbind(Tempo = c(indices_1[,1],rep(i,nrow(resulta_1))),Amostra
= c(indices_1[,2],resulta_1[,1]),ID = c(indices_1[,3],resulta_1[,2]),S =
c(indices_1[,4],resulta_1[,3]),SH = c(indices_1[,5],resulta_1[,4]),Flag =
c(indices_1[,6],resulta_1[,5]))#1° bloco de índices sendo organizado em
colunas#
  row.names(indices_1) <-
c(nome.col,rep(gsub(".txt","",arqs[i]),nrow(resulta_1)))#Concatenando
row.names(indices_1) com nome.col, também com o intuito de apresentar um só
bloco no final#
  ###BLOCO B###
  resulta_2 <- data.frame(resulta[2])#O objeto resulta_2 contém o cálculo
dos índices H,J e d em um data frame. Os resultados precisam ser divididos
em 2 blocos, uma vez que o 1° bloco de índices é calculado para CADA
amostra, enquanto o 2° bloco não#
  nome.col  <- c(row.names(indices_2))#Acrescentando indices_2 (criado
acima) ao objeto nome.col. Isso porque há 2 blocos de cada, que serão unidos
no final#
  indices_2 <- cbind(Tempo = c(indices_2[,1],i),H =
c(indices_2[,2],resulta_2[,1]),J = c(indices_2[,3],resulta_2[,2]),d =
c(indices_2[,4],resulta_2[,3]))#O 2° bloco de índices sendo organizado em
colunas#
  row.names(indices_2) <- c(nome.col,gsub(".txt","",arqs[i]))#Concatenando
row.names(indices_2) com nome.col, também com o intuito de apresentar um só
bloco no final#
     } #FIM DO SEGUNDO IF#
} #FIM DO FOR#

############################
#GRÁFICOS SIMPLES(BLOCO B)#
############################

n.row <- nrow(indices_2)#N° de linhas de indices_2 sendo atribuído ao objeto
n.row#

################
#SHANNON-WEINER#
################
plot(indices_2[,1],indices_2[,2], #Um gráfico com o tempo (os arquivos
organizados na 1ª coluna de indices_2)e os resultados de H em cada tempo
(arquivo)#
    type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col="green3",xaxt = 'n', #parâmetros do gráfico sendo
selecionados#
    xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos dos
eixos x e y#
```

```
    main="Diversidade de Shannon-Weiner (H)")#Título#
axis(side=1, at=seq(1, n.row, by=1))#Mais parâmetros do gráfico. Os pontos
do eixo x irão de 1 até o n° de linhas de indices_2, ou seja, o n° de
arquivos. Indo de 1 em 1#

########
#PIELOU#
########
windows() #Disponibilizando nova janela gráfica#
plot(indices_2[,1],indices_2[,3], #Gráfico do índice J (na coluna 3 de
indices_2 - 2°bloco) ao longo do tempo (cada arquivo)#
    type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col="green3",xaxt = 'n', #Parâmetros do gráfico#
    xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos do
eixos x e y#
    main="Equitabilidade de Pielou (J)")#Título#
axis(side=1, at=seq(1, n.row, by=1))#Mais parâmetros do gráfico. Os pontos
do eixo x irão de 1 até o n° de linhas de indices_2, ou seja, o n° de
arquivos. Indo de 1 em 1#

###############
#BERGER-PARKER#
###############
windows() #Disponibilizando nova janela gráfica#
plot(indices_2[,1],indices_2[,4], #Gráfico do índice d do 2° bloco (coluna
4) ao longo do tempo (cada arquivo)#
    type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col="green3",xaxt = 'n', #Parâmetros do gráfico#
    xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos do
eixos x e y#
    main="Dominância de Berger-Parker (d)") #Título#
axis(side=1, at=seq(1, n.row, by=1)) #Mais parâmetros do gráfico. Os pontos
do eixo x irão de 1 até o n° de linhas de indices_2, ou seja, o n° de
arquivos. Indo de 1 em 1#

################################
#GRÁFICOS COM AMOSTRAS(BLOCO A)#
################################

##########
#MORISITA#
##########
windows() #Disponibilizando nova janela gráfica#
par(mfrow=c(1,1),oma=c(0,0,0,0),pch=25, #Parâmetros do gráfico#
    mar=c(5,4,4,2) + c(0,0,0,6), #Parâmetros do gráfico. Ressaltando:
aumento da margem do gráfico#
    xpd=TRUE) #Parâmetros do gráfico. Ressaltando: plotando a legenda fora
da margem#
ymin.ID <- min(indices_1[,3])#Valor mínimo do índice ID sendo atribuído ao
objeto ymin.ID#
```

```
ymax.ID <- max(indices_1[,3])#Valor máximo de ID sendo atribuído ao objeto
ymax.ID#
names <- unique(indices_1[,2])#Retirando os valores repetidos da coluna 2 do
1° bloco de índices, sendo que esta coluna corresponde às amostras. Isto foi
necessário porque este índice NÃO É calculado para cada uma das amostras#
k.amostras <- length(names)#N° de amostras sem valores repetidos sendo
conferido pelo comando length e atribuído a k.amostras#
colors <- rainbow(k.amostras)#Atribuindo cores diferentes para cada uma das
amostras presentes#
plot(indices_1[,1][indices_1[,2] == 1],indices_1[,3][indices_1[,2] == 1],
#Gráfico do índice ID ao longo do tempo (coluna 1)#
     type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col=colors[1],xaxt = 'n', #Parâmetros do gráfico#
     xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos dos
eixos x e y#
     ylim=c(ymin.ID,ymax.ID), #Os limites do eixo y serão os valores mínimo
e máximo do indice ID#
     main="Dispersão de Morisita (ID)") #Título#
if (k.amostras > 1) #Se o n° de amostras for maior que 1, teremos o que
segue abaixo#
{
 for (i in 2:k.amostras)#Iniciando o for, de 2 até o n° total de amostras
presentes#
 {
 lines(indices_1[,1][indices_1[,2] == i],indices_1[,3][indices_1[,2] == i],
#Colocando o contador do for para tempo (coluna1), amostra (coluna2) e
ID(coluna3) para que apareçam linhas para todos os indices de todas as
amostras que o arquivo do usuário contiver#
      type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col=colors[i],xaxt = 'n', #Estabelecendo parâmetros#
      xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos dos
eixos x e y#
      ylim=c(ymin.ID,ymax.ID), #os limites do eixo y correspondem aos
valores máx e min de ID#
      main="Dispersão de Morisita (ID)")#Título#
 } #FIM DO FOR#
} #FIM DO IF#
axis(side=1, at=seq(1, n.row, by=1))#Mais parâmetros do gráfico. Os pontos
do eixo x irão de 1 até o n° de linhas de indices_2, ou seja, o n° de
arquivos. Indo de 1 em 1#
legend(1.05*n.row,(ymin.ID+ymax.ID)/2,yjust =
0.5,c(names),lty=c(rep(1,k.amostras)),lwd=c(rep(1,k.amostras)),col=c(colors)
,title = "Amostras",title.col = "darkblue",text.col = "black",box.col =
"darkblue",bg = "aliceblue") #Estabelecendo a legenda do gráfico#

#########
#RIQUEZA#
#########
windows() #Disponibilizando nova janela gráfica#
par(mfrow=c(1, 1),oma=c(0,0,0,0),pch=25, #Parâmetros do gráfico#
    mar=c(5,4,4,2) + c(0,0,0,6), #Parâmetros do gráfico. Ressaltando:
```

```
aumento da margem do gráfico#
    xpd=TRUE) #Parâmetros do gráfico. Ressaltando: plotando a legenda fora
da margem#
ymin.SH <- min(indices_1[,4]) #Valor mínimo calculado do índice S#
ymax.SH <- max(indices_1[,4]) #Valor máximo calculado do índice S#
plot(indices_1[,1][indices_1[,2] == 1],#Gráfico do índice S (coluna4) ao
longo do tempo (coluna1)(explicação referente a esta linha e à linha logo
abaixo)#
     indices_1[,4][indices_1[,2] == 1],
     type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col=colors[1],xaxt = 'n', #Parâmetros do gráfico#
     xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos dos
eixos x e y#
     ylim=c(ymin.SH,ymax.SH), #os limites do eixo y correspondem aos valores
máx e min de S#
     main="Riqueza (S)") #Título#
if (k.amostras > 1)#Se o n° de amostras for maior que 1, teremos o que segue
abaixo#
{
 for (i in 2:k.amostras)#Iniciando o for, de 2 até o n° total de amostras
presentes#
 {
 lines(indices_1[,1][indices_1[,2] == i],#Colocando o contador do for para
tempo (coluna1), amostra (coluna2) e ID(coluna3) para que apareçam linhas
para todos os indices de todas as amostras que o arquivo do usuário
contiver#
      indices_1[,4][indices_1[,2] == i], #A explicação acima refere-se a
esta linha também#
      type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col=colors[i],xaxt = 'n', #Estabelecendo parâmetros#
      xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos dos
eixos x e y#
      ylim=c(ymin.SH,ymax.SH),#os limites do eixo y correspondem aos valores
máx e min de S#
      main="Riqueza (S)") #Título#
 } #FIM DO FOR#
} #FIM DO IF#
axis(side=1, at=seq(1, n.row, by=1))#Mais parâmetros do gráfico. Os pontos
do eixo x irão de 1 até o n° de linhas de indices_2, ou seja, o n° de
arquivos. Indo de 1 em 1#
legend(1.05*n.row,(ymin.SH+ymax.SH)/2,yjust =
0.5,c(names),lty=c(rep(1,k.amostras)),lwd=c(rep(1,k.amostras)),col=c(colors)
,title = "Amostras",title.col = "darkblue",text.col = "black",box.col =
"darkblue",bg = "aliceblue") #Estabelecendo a legenda do gráfico#

##################
#RIQUEZA APARENTE#
##################
windows() #Disponibilizando nova janela gráfica#
par(mfrow=c(1, 1),oma=c(0,0,0,0),pch=25, #Parâmetros do gráfico#
```

```
    mar=c(5,4,4,2) + c(0,0,0,6), #Parâmetros do gráfico. Ressaltando:
aumento da margem do gráfico#
    xpd=TRUE) #Parâmetros do gráfico. Ressaltando: plotando a legenda fora
da margem#
ymin.SH <- min(indices_1[,5])#Valor mínimo calculado do índice SH#
ymax.SH <- max(indices_1[,5]) #Valor máximo calculado do índice SH#
plot(indices_1[,1][indices_1[,2] == 1],#Gráfico do índice SH (coluna5) ao
longo do tempo (coluna1)(explicação referente a esta linha e à linha logo
abaixo)#
     indices_1[,5][indices_1[,2] == 1],
     type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col=colors[1],xaxt = 'n', #Parâmetros do gráfico#
     xlab="Período de Coleta (Tempo)",ylab="Índice Calculado",  #Rótulos dos
eixos x e y#
     ylim=c(ymin.SH,ymax.SH), #os limites do eixo y correspondem aos valores
máx e min de SH#
     main="Riqueza Aparente (SH)") #Título#
if (k.amostras > 1)#Se o n° de amostras for maior que 1, teremos o que segue
abaixo#
{
 for (i in 2:k.amostras)#Iniciando o for, de 2 até o n° total de amostras
presentes#
 {
 lines(indices_1[,1][indices_1[,2] == i],#Colocando o contador do for para
tempo (coluna1), amostra (coluna2) e ID(coluna3) para que apareçam linhas
para todos os indices de todas as amostras que o arquivo do usuário
contiver#
      indices_1[,5][indices_1[,2] == i], #A explicação acima refere-se a
esta linha também#
      type="o",lwd=2,pch=1,col.lab = "darkblue",col.main =
"darkblue",col=colors[i],xaxt = 'n', #Estabelecendo parâmetros#
      xlab="Período de Coleta (Tempo)",ylab="Índice Calculado", #Rótulos dos
eixos x e y#
      ylim=c(ymin.SH,ymax.SH), #os limites do eixo y correspondem aos
valores máx e min de S#
      main="Riqueza Aparente (SH)") #Título#
 } #FIM DO FOR#
} #FIM DO IF#
axis(side=1, at=seq(1, n.row, by=1)) #Mais parâmetros do gráfico. Os pontos
do eixo x irão de 1 até o n° de linhas de indices_2, ou seja, o n° de
arquivos. Indo de 1 em 1#
legend(1.05*n.row,(ymin.SH+ymax.SH)/2,yjust =
0.5,c(names),lty=c(rep(1,k.amostras)),lwd=c(rep(1,k.amostras)),col=c(colors)
,title = "Amostras",title.col = "darkblue",text.col = "black",box.col =
"darkblue",bg = "aliceblue")#Estabelecendo a legenda do gráfico#
#
#Abaixo segue um título e uma nota explicando ao que se refere o Flag#
#
cat("
-------------------------------------------
--- Análise Estrutural de uma Comunidade ---
```

```
-------------------------------------------\n
Nota: Flag = 1 Implica na conversão: matriz
      de abundância em matriz de ocorrência\n\n")
#
return(list("Bloco A" = indices_1, "Bloco B" = indices_2))#A função retorna
uma lista contendo o valor dos índices calculados nos blocos A e B#
} #FIM DA FUNÇÃO comunidade#
```

ARQUIVOS

Exemplos do Help

Exemplo 1:

arquivo1.txt arquivo2.txt arquivo3.txt

Exemplo2:

periodo1.txt periodo2.txt

Código da função

Função comunidade

Código do Help

Help da função