Função trabalho final

Arquivo da função

Função mapar

Código da função

```
################################################################################
##############################
############################## FUNCAO MAPAR   POR:  JP VIEIRA-ALENCAR JUN/2019
##############################
################################################################################
##############################

mapar <- function(grid, mpa, lsp, plot=FALSE, shp=NULL, prop=FALSE,
grp=FALSE) #nomeando a funcao, indicando os argumentos e defaults
{
  #Modificacoes nos dados essenciais para o funcionamento da funcao:
  colnames(mpa) <- grid@data[["id"]] #atribuindo 'ids' do grid as colunas da
matriz
  lsp[1] <- as.character(lsp[,1]) #atribuindo classe 'character' a coluna
com nome das especies
################################################################################
###############
  ###################################### If tests
############################################
################################################################################
###############
  ##### if's da matriz #####

  if(is.null(row.names(mpa))) #Verificando se ha nomes nas linhas da matriz
    {
    stop("Nomes das linhas de mpa ausentes, corrija para prosseguir") #para
a funcao e exibe uma mensagem de erro
    }
  if(sum(is.na(mpa))>=1) #Verificando se existem NA's na matrix
    {
    stop("NA's detectados na matriz, corrija isso para prosseguir") #para a
funcao e exibe uma mensagem de erro
    }
  ##### if's da lista de spp e areas #####
  if(class(lsp)!= 'data.frame') #veriricando se a classe da lista de
spp/areas e um data.frame
    {
    stop("lsp nao e um objeto da classe data.frame, corrija para continuar")
#para a funcao e exibe mensagem de erro
    }
```

```
  ##### if's conjuntos #####
  if(sum(sort(row.names(mpa)) != sort(lsp[,1]))>=1) #verificando se o nome
das spp na matrix e no grid sao iguais
    {
    stop("Nomes das especies diferem entre mpa e lsp, corrija para
prosseguir") #para a funcao e exibe uma mensagem de erro
    }
  if(sum(as.numeric(colnames(mpa))!= grid@data[['id']])>=1) #comparando
codigo das celulas entre objetos grid e mpa
    {
    stop("Codigos das celulas diferem entre o grid e a matriz, corrija para
prosseguir") #para a funcao e exibe uma mensagem de erro
    }
##############################################################################
##############
  ################################## Desenrolar da funcao
####################################
##############################################################################
##############
    areas <- list() #cria uma lista vazia com o nome areas
    map.ar <- list()#cria uma lista vazia com o nome map.ar
    for(i in 1:nrow(unique(lsp[2]))) #preenche as listas areas e map.ar
baseado nos codigos unicos das areas em lsp
    {
    areas[[i]] <- mpa[lsp[,1][lsp[,2] == unique(lsp[,2])[i]],] #retira da
matriz de presenca e ausencia os dados das spp com codigo da posicao 'i' e
os coloca as info de cada area separada na lista areas
    map.ar[[i]] <- grid[which(apply(areas[[i]], 2, sum)!=0),] #cria grids
individuais para cada codigo de area e os coloca na lista map.ar
    }
##############################################################################
################################
  ################################## interacao com @ usuari@: plot das areas
####################################
##############################################################################
################################
    if(plot==TRUE) #condicional para plotar ou nao um preview das areas em
um dispositivo visual do R
      {
      if(is.null(shp)) #conferindo se @ usuari@ informou o shp da area de
estudo para ser associado ao plot
        {
        stop("shape file nao indicado para realizar o plot, corrigir isso
antes de continuar") #Parando a funcao e solicitando o .shp da area de
estudo
        }else{
      Q2='s' #definindo padrao para entrar no while
      while(Q2 == 's'| Q2 == 'S') #condicional para o desejo d@ usuari@ de
plotar outra area
        {
```

```
        Q1 <- readline(prompt=cat('\n', length(map.ar), '\ Areas formadas.
Qual area gostaria de visualizar? ')) #Informando quantas areas foram
formadas e perguntando qual area @ usuari@ gostaria de visualizar
        if(prop==TRUE) #condicional para criar a lista com porcentagem de
spp por celula do grid das areas
        {
          per <- list() #Criando lista vazia para receber os vetores de
porcentagem de spp por celula em cada area
          for(i in 1:length(areas)) #criando looping com comprimento igual
ao numero total de areas
          {
            n <- apply(areas[[i]], 2, sum) #somando numero de spp por celula
da area 'i'
            m <- length(rownames(areas[[i]])) #atribuindo a m o valor da
quantidade total de spp na area i
            per[[i]] <- n[n!=0]/m #preenchendo a lista vazia com vetores
numericos da proporcao de spp/area em cada celula da area 'i'
          }
          for(i in 1:length(per)) #looping para ajustar a proporcao de
spp/area a valores desejados de 0, 0.3 e 0.7
          {
            per[[i]][per[[i]]<0.3] <- 0.05 #todos valores de per abaixo de
0.3 transformados em 0.05
            per[[i]][per[[i]]>=0.3 & per[[i]] < 0.7] <- 0.3 #todos valores
de per maiores ou iguais 0.3 e menores que 0.7 transformados em 0.3
            per[[i]][per[[i]]>=0.7] <- 0.7 #todos valores maiores ou iguais
a 0.7 transformados em 0.7
          }
            par(mar=c(0.1,0.1,1,0.1)) #ajustando margens do plot
            plot(shp, main=paste('Area', unique(lsp[,2])[as.numeric(Q1)],
sep=' '), cex.main=1.3) #plota area de estudo
            plot(map.ar[[as.numeric(Q1)]], add=TRUE, col=rgb(0,0,1,
per[[as.numeric(Q1)]])) #plota grid da area escolhida COM proporcao de
spp/area em cada celula
        }else{
            per <- NULL #criando per = NULL para evitar mensagem de erro
'per not found' no return
            par(mar=c(0.1,0.1,1,0.1)) #ajustando margens do plot
            plot(shp, main=paste('Area', unique(lsp[,2])[as.numeric(Q1)],
sep=' '), cex.main=1.3) #plotando area de estudo
            plot(map.ar[[as.numeric(Q1)]], add=TRUE, col=rgb(0,0,1, 0.3))
#plota grid da area escolhida SEM proporcao de spp/area em cada celula
          } #Fechando looping de plot SEM porcentagens de spp por celula
          Q2 <- readline(prompt=cat('\nGostaria de visualizar outra area?
s/n ')) #Da opcao de plotar outra area e aponta as respostas possiveis
          if(Q2 != 's' & Q2 != 'n' & Q2 != 'S' & Q2 != 'N') #condicional
sobre as respostas possiveis para Q2 considerando maiusculas
          {
            stop("Resposta invalida, mapar finalizada.") #para a funcao e
exibe uma mensagem de erro
          }
```

```
        } #fecha o while de plotagem
      } #fechando else da presenca de shapefile para o plot
    }else{
      per <- NULL #criando per = NULL para evitar mensagem de erro 'per not
found' no return
    } #fechando o condicional plot == TRUE
################################################################################
##############################
    ######################################## salvando areas em .shp
###############################################
################################################################################
##############################
     if(grp == TRUE) #condicional para agrupar todas as areas em um unico
.shp
        {
          allar <- map.ar[[1]] #cria objeto allar e atribui a primeira
posicao de map.ar, em que podemos adicionar mais informacoes com o for
          for(i in 2:length(map.ar)) #looping de 2 ao length de map.ar
          {
          allar <- rbind(allar, map.ar[[i]]) #combina os
SpatialPoligonsDataFrame mantendo as propriedades dessa classe de objeto
atraves do rbind
          }
          df.info <- c(rep(paste(unique(lsp[,2])[1]),
times=length(map.ar[[1]]))) #cria um vetor com a repeticao do codigo da area
1 com comprimento equivalente a quantidade de celulas do grid da area 1
          for(i in 2:length(map.ar)) #looping de 2 ao lenght de map.ar
          {
          df.info <- c(df.info, rep(paste(unique(lsp[,2])[i], sep=' '),
times=length(map.ar[[i]]))) #cria vetores com as repeticoes do codigo de
todas as areas com comprimento equivalente a quantidade de celulas do grid
de cada area
          }
          allar@data$cod.areas <-  df.info #cria uma nova coluna no
dataframe do SpatialPoligonsDataFrame das areas combinadas atribui os dados
criados acima a uma nova coluna chamada 'cod.areas'
          writeOGR(allar, dsn=getwd(), layer = 'map.areas', driver="ESRI
Shapefile") #salva o objeto formado em formato shp com uma coluna para
classificar e editar cada area separadamente baseado no codigo das areas
          paste(cat('\nTodas as areas foram salvas em um unico arquivo
chamado map.areas.shp \n\nMapar finalizada\n\n')) #informa que um arquivo
shp foi criado e que a funcao esta finalizada
        }else{
        Q3 <- readline(prompt=cat('\nGostaria de salvar alguma area em .shp?
s/n ')) #Da opcao de salvar areas em .shp e aponta as respostas possiveis
        if(Q3 != 's' & Q3 != 'n' & Q3 != 'S' & Q3 != 'N') #condicional sobre
as respostas possiveis para Q3 considerando maiusculas
          {
            stop("Resposta invalida, mapar finalizada.") #para a funcao e
exibe uma mensagem de erro
```

```
          }
        if(Q3 == 's') #condicional para o desejo d@ usuari@ de salvar alguma
area em .shp
          {
          Q4 <- readline(prompt=cat('\nQual area gostaria de salvar em .shp?
')) #oferece a opcao de salvar areas individualmente
            if(Q4 == 'todas' | Q4 == 'TODAS' | Q4 == 'Todas') #condicional
para salvar todas as areas separadamente
            {
            for(i in 1:length(map.ar)) #cria looping de comprimento igual ao
numero de areas
            {
            writeOGR(map.ar[[i]], dsn=getwd(), layer = paste('Area',
unique(lsp[,2])[i], sep=' '), driver="ESRI Shapefile") #cria arquivos .shp
individuais para cada area criada pela funcao
            }
            paste(cat('\nTodas as areas foram salvas separadamente em .shp
\n\nMapar finalizada\n\n')) #informa que um arquivo shp foi criado para cada
area e que a funcao esta finalizada
            }else{
            writeOGR(map.ar[[as.numeric(Q4)]], dsn=getwd(), layer =
paste('Area', unique(lsp[,2])[as.numeric(Q4)], sep=' '), driver="ESRI
Shapefile") #cria arquivo .shp com a area escolhida pel@ usuari@
            paste(cat(paste('\nArea', unique(lsp[,2])[as.numeric(Q4)], sep='
'), 'salva em .shp', '\n\nMapar finalizada\n\n')) #informa qual objeto foi
salve e exibe mensagem de finalizacao da funcao
            }
          }else{
          cat('\nMapar finalizada\n\n') #exibe mensagem de finalizacao da
funcao
          }
        }
##############################################################################
################################
    ################### finalizando a funcao e selecionando os objetos que
devem ser retornados ###############
##############################################################################
################################
    if(is.null(per)) #condicional para determinar quais objetos retornar no
final da funcao
    {
    return(list(areas, map.ar)) #finalizando a funcao e retornando os
objetos areas e map.ar
    }else{
    return(list(areas, map.ar, per))  #finalizando a funcao e retornando os
objetos areas, map.ar e per
    }
}

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

```
  ############################################## Fim da Funcao
##############################################
################################################################################
################################
```

Arquivo da página de ajuda da função

Help da função mapar

Página de ajuda da função

```
mapar                package:disciplina.do.R                R Documentation

Mapeamento de áreas

Description :

A função mapar extrai de um grid espacial subconjuntos de grids formados a
partir da distribuição geográfica de um conjunto de espécies selecionado
pel@ usuári@. Os subconjuntos de grids podem ser visualizados no dispositivo
visual do R e/ou podem ser salvos em arquivos .shp para edição em outros
softwares (eg.: Sistemas de Informação Geográfica, QGIS, ArcGis, etc…)

Usage :

mapar(grid, mpa, lsp, plot=FALSE, shp=NULL, prop=FALSE, grp=FALSE)

Arguments :

grid    Objeto da classe SpatialPolygonsDataFrame contendo um conjunto de
polígonos interligados que sobrepõem totalmente uma área de estudos de
interesse. Cada polígono do grid deve conter um código de identificação
relacionado na coluna “id” do slot @data do objeto.

mpa Objeto da classe matrix representando uma matriz de presença e ausência
em que as colunas representam cada polígono (célula) do objeto grid e as
linhas representam as espécies encontradas na área de estudo. Zero
representa a ausência de uma espécie em um determinado polígono, e um
representa a presença. Os nomes das colunas devem corresponder com as
informações relacionadas na coluna “id” do slot @data do objeto grid. A
própria função mapar atribui aos nomes das colunas da matriz as informações
de “id”, mas @ usuário deve garantir que a sequência de colunas da matriz
estejam na mesma sequência de informações dos “ids” do objeto grid.

lsp Objeto da classe data.frame contendo duas colunas, a primeira contendo o
nome das espécies de interesse, e a segunda contendo um código de área que
funciona como fator de agrupamento das distribuições geográficas das
```

```
espécies.

plot    Argumento lógico. Se plot=TRUE a função informa quantos subconjuntos
de grid foram formados e se torna interativa aguardando input d@ usuári@
selecionando qual subconjunto gostaria de visualizar. Após a exibição do
primeiro subconjunto selecionado pel@ usuári@ a função aguarda um segundo
input sobre o interesse em continuar visualizando os diferentes subconjunto
formados pela função. Se @ usuári@ alterar par(mfrow) antes de executar a
função, múltiplos subconjuntos de grid aparecerão no dispositivo visual do
R.

shp Objeto da classe SpatialPolygonsDataFrame que representa a área de
estudo de interesse. Objeto requisitado caso @ usuári@ opte por visualizar
no dispositivo visual do R os subconjuntos de grid formados pela função.

prop    Argumento lógico. Se prop=TRUE a função calcula a proporção n/m em
que n representa o número de espécies em um determinado polígono e m
representa o número total de espécies no subconjunto de polígonos. prop
retorna um conjunto de valores de zero a um por célula do subconjunto de
grid que é aplicado em alpha no argumento col=rgb(r, g, b, alpha) e atribui
transparência diferentes de acordo com a proporção n/m calculada para cada
célula. Células com menos de 30% de todas as espécies do subconjunto terão
alpha=0.05; Células com 30%~69% de todas as espécies do subconjunto terão
alpha=0.3; Células com mais 70% ou mais de todas as espécies do subconjunto
terão alpha=0.7.

grp Argumento lógico. Se grp=TRUE a função salva automaticamente todas os
subconjuntos de grid formados em um arquivo .shp e cria a coluna cod.areas
no arquivo para que o shapefile possa ser classificado através dessa coluna
e a visualização dos subconjuntos possa ser individualizada em um programa
de Sistema de Informação Geográfica (e.g.: QGIS, ArcGis).


Details :

A função mapar apresenta uma parte interativa que visa ampliar as opções de
escolha d@ usuári@ para visualização dos subconjuntos de polígonos (áreas de
distribuição das espécies de interesse) formados e para salvar
individualmente os subconjuntos de interesse em formato .shp. As interações
que demandam respostas do tipo sim/não ('s/n') aceitam apenas essas
respostas para continuar ou interromper a função, enquanto que as interações
relacionadas aos subconjuntos formados dependem da quantidade de
subconjuntos formados. Quando grp=FALSE @ usuári@ tem a opção de salvar
áreas individualmente digitando um número inteiro que representa a posição
do subconjunto na lista de subconjuntos formados, ou respondendo 'todas'
para salvar todos os subconjuntos mas em arquivos .shp separados.

Value :

Se prop=FALSE, mapar retorna uma lista com dois componentes:
```

```
  comp1 : Lista de comprimento=N contendo as matrizes de presença e ausência
individuais para cada subconjunto de polígonos (áreas de interesse) apenas
com as espécies selecionadas para formar as respectivas N áreas

  comp2 : Lista de comprimento=N contendo objetos da classe
SpatialPolygonsDataFrame representando os N subconjuntos de polígonos (áreas
de interesse) formados

Se prop=TRUE, mapar retorna uma lista contendo os dois componentes acima
mais:
  comp3 : Lista de comprimento=N contendo vetores com os valores de alpha
(col=rgb(r, g, b, alpha) por polígono do grid formado, respeitando a
proporção n/m descrita em prop (acima)

Warning :

a) mapar atribui a colnames(mpa) os 'id' do objeto grid, é responsabilidade
d@ usuári@ garantir que a matriz mpa tenha sido elaborada a partir do grid
usado na função ou as áreas formadas podem não representar o resultado
esperado pel@ usuári@

b) Ao salvar um subconjunto de polígonos em .shp, mapar uma uma nomenclatura
padrão, caso existam outros arquivos com o mesmo nome no diretório de
trabalho a área não será salva e uma mensagem de erro padrão será exibida

Author(s) :

        Vieira-Alencar, João Paulo dos Santos
    Laboratório de Ecologia, Evolução e Conservação de Vertebrados,
Departamento de Ecologia, Universidade de São Paulo, São Paulo/SP - BRASIL


See also :

     class: SpatialPolygonsDataFrame

Examples :

     ## Arquivos de exemplo disponibilizados no wiki do autor.
```

Arquivos de exemplo para teste da função mapar

Mapa da área: Shapefile Cerrado
Grid da área: Shapefile Grid
Matriz de presença e ausência: mpa
Lista de espécies/área: lsp

OBS: Arquivos de shapefile podem ser lidos usando a função: “readOGR(dsn=getwd(), layer='nome do arquivo sem extensão')” do pacote rgdal. Em dsn basta indicar o diretório que contém os arquivos depois de descomprimi-los.