

Projeto Final: Tá em Área Protegida?

Alguns avisos importantes:

- Rodar linha por linha antes de rodar a função.
- Trata-se de uma função que utiliza *shapefiles* (informações especializadas).

A fonte para acesso a todos os *shapefiles* utilizados estão disponíveis abaixo. Foram utilizados os *shapes* do território brasileiro (disponível *online*) e das AP (conseguidos por solicitação ao Instituto Socioambiental (ISA), que dispõe de amplo acervo). É importante enfatizar que os *shapes* das AP foram cedidos pelo ISA especificamente para este fim e **qualquer outra utilização desses dados** requer contato diretamente com o Instituto.

A Função:

```
# Criando a função
taemap <- function(x, pasta, map=TRUE) #criando a função com os argumentos
"x" (nome do arquivo a ser comparado: "__.shp") e map (decisão de plotar ou
não o mapa com as AP e as áreas inseridas)
{
  # Acessando os pacotes requeridos para o desenvolvimento da função
  require(rgdal) # para acessar a função readOGR
  require(tmap) # plotar o mapa
  require(sf) # comparar dados sobreposições
  require(dplyr) # para acessar a função mutate
  # Se os pacotes não estiverem instalados
  if(("package:rgdal" %in% search()) == FALSE) # se o pacote rgdal não
tiver baixado..
  {
    install.packages("rgdal") # vai instalá-lo
    library(rgdal) # e acessá-lo
  }
  if(("package:tmap" %in% search()) == FALSE) # se o pacote tmap não tiver
baixado..
  {
    install.packages("tmap") # vai instalá-lo
    library(tmap) # e acessá-lo
  }
  if(("package:sf" %in% search()) == FALSE) # se o pacote sf não tiver
baixado..
  {
    install.packages("sf") # vai instalá-lo
    library(sf) # e acessá-lo
  }
  if(("package:dplyr" %in% search()) == FALSE) # se o pacote dplyr não
tiver baixado..
  {
    install.packages("dplyr") # vai instalá-lo
    library(dplyr) # e acessá-lo
  }
}
```

```
# Leitura dos dados de área inseridos pela/o usuária/o
pasta # lê o nome da pasta onde constam os shapefiles
diretorio <- getwd() # guarda a informação do diretório atual em um
objeto (para ao final da função, retorná-lo)
setwd(pasta) # define como diretório a pasta que contem os shapefiles
if(!(".shp" %in% dir()) == FALSE | (!".prj" %in% dir()) == FALSE | (!".bdf"
%in% dir()) == FALSE | (!".shx" %in% dir()) == FALSE) # Se alguma das
extensões dos shapefiles não estiverem na pasta
{message ("Todas as extensões do shapefile (.shp, .prj, .bdf, .shx, etc)
devem estar nomeadas em seu diretório com o mesmo nome")} # mostra uma
mensagem de alerta para garantir que, para que o arquivo seja lido, todas as
extensões do shape estejam na mesma pasta e com o mesmo nome
x #lê o nome do arquivo a ser comparado com as AP ("nome.shp") - deve
estar no diretório
insere <- readOGR(x) #cria o objeto "insere", designando-o a leitura do
arquivo inserido
#Verificação de parâmetro
#Cria objetos para conferir a classe do objeto "insere"
class.polygons <- "SpatialPolygonsDataFrame" %in% class(insere) #cria o
objeto "class.polygons" com a informação de se há o termo
"SpatialPolygonsDataFrame" na classe do objeto "insere"
class.lines <- "SpatialLinesDataFrame" %in% class(insere) #cria o
objeto "class.lines" com a informação de se há o termo
"SpatialLinesDataFrame" na classe do objeto "insere"
#Cria um if para parar a função caso o objeto "insere" não seja de
nenhuma das classes acima
if (class.lines == FALSE & class.polygons == FALSE) #cria uma condição:
se a classe do objedo "insere" não for dos formatos correspondentes a
shapefiles...
{stop ("o objeto 'insere' não é nem da classe 'SpatialPolygonsDataFrame'
nem 'SpatialLinesDataFrame'!")} #...a função para, pois não é possível
proceder.
# Leitura dos Shapes das Áreas Protegidas
# Unidades de Conservação Federais (UCF)
ucf <- readOGR("UCs_Federais.shp") #lê os shapes das Unidades de
Conservação Federais
#summary(ucf) #lendo o objeto ucf
#class(ucf) #mostra que trata-se de um objeto da classe
"SpatialPolygonsDataFrame"
#names(ucf) #verifica os nomes das colunas do objeto
# Unidades de Conservação Estaduais (UCE)
uce <- readOGR("UCs_Estaduais.shp") #lê os shapes das Unidades de
Conservação Estaduais
#summary(uce) #lendo o objeto uce
#class(uce) #mostra que trata-se de um objeto da classe
"SpatialPolygonsDataFrame"
#names(uce) #verifica os nomes das colunas do objeto
# Terras Indígenas (TI)
ti <- readOGR("Terra_Indígena.shp") #lê os shapes das Terras Indígenas
#summary(ti) #lendo o objeto ti
```

```
#class(ti) #mostra que trata-se de um objeto da classe
"SpatialPolygonsDataFrame"
#names(ti) #verifica os nomes das colunas do objeto
# Preparando os dados das AP para retornar um Data.frame único com todas
as informações de sobreposição todas as Áreas Protegidas
# Convertendo para sf (dados espaciais em data.frame) os dados a serem
sobrepostos
{
  insere.sf <- st_as_sf(insere) #convertendo o objeto "insere"
(shapefile) ao formato sf, legível para st_intersects
  ucf.sf <- st_as_sf(ucf) #convertendo o objeto "ucf" (shapefile) ao
formato sf, legível para st_intersects
  uce.sf <- st_as_sf(uce) #convertendo o objeto "uce" (shapefile) ao
formato sf, legível para st_intersects
  ti.sf <- st_as_sf(ti) #convertendo o objeto "ti" (shapefile) ao
formato sf, legível para st_intersects
}
#transformar ti.sf em um objeto com o mesmo número de colunas que
ucf.sf e uce.sf (para depois uní-los todos em um único objeto)
#cria um data.frame com repetidos NA (apenas pra coincidir o número
de colunas do objeto ha.sobrepos.uc) com o número da extensão do objeto
"extensao.ti"
uso = c(rep(NA,each=(length(ti.sf)))) # cria um objeto com
repetidos NA (apenas pra coincidir o número de colunas do objeto
ha.sobrepos.uc) com o número da extensão do objeto "ti.sf"
uso <- as.data.frame(uso) # transforma-o em data.frame
ti.sf <- merge(ti.sf,uso) # acrescenta mais uma coluna ao
data.frame ti.sf
#Juntando informações de todas as AP
uc.sf <- rbind(ucf.sf,uce.sf) #Juntando informações das UC
#padronizando nome das colunas para unir as informações de todas
as AP
names(uc.sf)[2] <- "nome_ap" #renomeia a coluna "nome_uc"
para "nome_ap"
names(ti.sf)[2] <- "nome_ap" #renomeia a coluna "nome_ti"
para "nome_ap"
ap.sf <- rbind(uc.sf, ti.sf) #Juntando informações das UC com TI =
AP
# Comparando espacialização das AP e da área inserida
ap.sobrepos <- st_intersection(insere.sf, ap.sf) #averigua a presença de
sobreposições das áreas protegidas e das áreas inseridas
#Averiguando a extensão da sobreposição
{
  #UCF
  sobrepos <- ap.sobrepos %>% #calcula a extensão da incidência da área
inserida nas áreas protegidas
  mutate(area_sobreposta = st_area(.) %>% as.numeric()) #cria a coluna
"area_sobreposta" onde a área sobreposta é inserida em formato numérico
  sobrepos #retorna todas as informações das AP, incluindo a sobreposição
com as áreas inseridas
}
```

```
# Fonte da etapa de averiguação das extensões:
https://gis.stackexchange.com/questions/140504/extracting-intersection-areas-in-r

#Plotando o mapa com as áreas
if (map) # Dado que um dos argumentos da função dá a alternativa de
escolher plotar ou não o mapa - cria-se um if para tanto (se map = TRUE:)
{
  br <- readOGR("UFEBRASIL.shp") #cria objeto com os shapes das
Unidades Federativas brasileiras para plotá-las no mapa
  #para o caso de "insere" conter shapefiles com polígonos
  if (class.polygons == TRUE) #Se o objeto "insere" for da classe
"SpatialPolygonsDataFrame" (polígonos)...
  {
    # Criar objeto de mapa com as informações de UCE, UCF, TI e área
inserida
    map.polygons <- tm_shape(br) + tm_fill(col = "white") +
tm_borders() + #informação de plotagem dos limites (contorno) do país e dos
estados
    tm_shape(ucf) + tm_fill(col = "green", alpha = 0.5) +
tm_borders() + #informação de plotagem das Unidades de Conservação Federais
em verde
    tm_shape(uce) + tm_fill(col = "yellow", alpha = 0.6) +
tm_borders() + #informação de plotagem das Unidades de Conservação Estaduais
em amarelo)
    tm_shape(ti) + tm_fill(col = "blue", alpha = 0.4) + tm_borders()
+ #informação de plotagem das Terras Indígenas (contorno preto e
preenchimento em marrom claro)
    tm_scale_bar(position = c("right", "bottom")) + #informação de
plotagem de escala (posicionada no canto inferior direito do mapa)
    tm_layout(title = "Incidência nas Áreas Protegidas brasileiras",
title.size=0.7, title.position = c("RIGHT", "TOP")) + #informação de
plotagem do título (tamanho especificado e posicionado no canto superior
direito do mapa)
    tm_compass(position = c("left","top"), size=0.5) + #informação
de plotagem da rosa dos ventos (tamanho especificado e posicionado no canto
superior esquerdo do mapa)
    tm_add_legend(type="fill", col =
c("yellow","green","blue","red"), #informação de plotagem da legenda
(sequência de cores)
    labels= c("UCE", "UCF", "TI","Dados inseridos"),
title="Legenda") + #informação de plotagem dos detalhes da legenda
(respectivas categorias associadas às cores e título da legenda)
    tm_shape(insere) + tm_borders(col = "red") + tm_fill(col =
"red", alpha = 0.4) #informação de plotagem das áreas inseridas (contorno em
vermelho e preenchimento em vermelho claro)
    #plota o mapa
    map <- print (map.polygons) # ..Plota o mapa com as configurações
de polígonos
  }
  #para o caso de "insere" conter shapefiles com linhas
```

```

    if (class.lines == TRUE) #Se o objeto "insere" for da classe
"SpatialLinesDataFrame" (linhas)...
    {
      # Criar objeto de mapa com as informações de UCE, UCF, TI e área
inserida
      map.lines <- tm_shape(br) + tm_fill(col ="white") + tm_borders() +
#informação de plotagem dos limites (contorno) do país e dos estados
      tm_shape(ucf) + tm_fill(col = "green", alpha = 0.5) +
tm_borders() + #informação de plotagem das Unidades de Conservação Federais
em verde
      tm_shape(uce) + tm_fill(col = "yellow", alpha = 0.6) +
tm_borders() + #informação de plotagem das Unidades de Conservação Estaduais
em amarelo)
      tm_shape(ti) + tm_fill(col = "blue", alpha = 0.4) + tm_borders()
+ #informação de plotagem das Terras Indígenas (contorno preto e
preenchimento em marrom claro)
      tm_scale_bar(position = c("right", "bottom")) + #informação de
plotagem de escala (posicionada no canto inferior direito do mapa)
      tm_layout(title = "Incidência nas Áreas Protegidas brasileiras",
title.size=0.7, title.position = c("RIGHT", "TOP")) + #informação de
plotagem do título (tamanho especificado e posicionado no canto superior
direito do mapa)
      tm_compass(position = c("left","top"), size=0.5) + #informação
de plotagem da rosa dos ventos (tamanho especificado e posicionado no canto
superior esquerdo do mapa)
      tm_add_legend(type="fill", col =
c("yellow","green","blue","red"), #informação de plotagem da legenda
(sequência de cores)
      labels= c("UCE", "UCF", "TI","Dados inseridos"),
title="Legenda") + #informação de plotagem dos detalhes da legenda
(respectivas categorias associadas às cores e título da legenda)
      tm_shape(insere) + tm_lines("red") #informação de plotagem das
áreas inseridas (contorno em vermelho)
      #plota o mapa
      map <- print (map.lines) # Plota o mapa com as configurações de
linha
    }
  }
  setwd (diretorio) # retorna o diretório ao diretório original
  # finaliza a função, mostrando o que retornará
  retorna <- c(sobrepos, map) # Criar objeto com lista dos dois objetos (o
data.frame com dados de todas as AP [incluindo as informações sobre as
sobreposições] e o mapa)
  return(retorna) # Retorna os dados das AP e o mapa
}

```

Help da Função

R

Documentation

taemap ()

Description:

~~ A função "taemap" visa comparar áreas fixas, especificamente as áreas protegidas (AP), ou seja, Unidades de Conservação (UC) e Terras Indígenas (TI), com alguma área inserida (ex. rodovias, áreas de desmatamento, mineração, etc.). A função retorna um data.frame com informações de sobreposição e mapa plotado com as AP e áreas inseridas. ~~

Usage:

~~ taemap(x, pasta, map=TRUE) ~~

Arguments:

~~ x - nome do arquivo shapefile: "____.shp" ~~

~~ pasta - nome da pasta em que encontra-se o shapefile

~~ map - decisão sobre plotagem de mapa (TRUE = plotagem) ~~

Details:

~~ O objetivo é avaliar incidência de pressão e ameaças nas APs e, por isso, há especificação sobre as AP. Mas as etapas da função podem servir para comparação de outras áreas quaisquer. ~~

Value:

~~ sobrepos: Data.frame com informações de sobreposição das as áreas protegidas ~~

~~ map: Mapa do Brasil com as AP e área inserida plotadas ~~

Warning:

~~ Todas as extensões do shapefile (.shp, .prj, .bdf, .shx, etc) devem estar nomeadas em seu diretório com o mesmo nome. ~~

Author

~~ Beatriz Moraes Murer: Mestranda em Ecologia no Instituto de Biociências da Universidade de São Paulo e analista de pesquisa socioambiental do Instituto Socioambiental ~~

beatriz.murer@usp.br / beatriz@socioambiental.org

References:

~~

<https://gis.stackexchange.com/questions/140504/extracting-intersection-areas-in-r> ~~

~~ <https://geocompr.robinlovelace.net/adv-map.html> ~~

~~ <https://keen-swartz-3146c4.netlify.com/plotting.html#class-intervals> ~~

Shapefiles:

~~ Território Brasileiro: <http://www.usp.br/nereus/?dados=brasil> ~~

~~ Áreas Protegidas: Instituto Socioambiental. Cedido em junho/2019 ~~

See Also:

~~ st_intersects()

~~ tm_shape()

Examples:

~~ x <- "Shapes/Insere.shp" #Rodovias do Plano Nacional de Logística (Fonte: <https://www.epl.gov.br/rede-georeferenciada-pnl-2025>)

~~ pasta <- "C:/Users/Bia/Documents/R/Shapes"

~~ taemap(x, pasta)

Shapefiles

Todos os shapes estão disponíveis em:

https://drive.google.com/drive/u/1/folders/1UWPBcv7Oy_FdL6xNFsZIW3cbP17BgocH

Separadamente é possível acessar em:

- Território Brasileiro: <http://www.usp.br/nereus/?dados=brasil>
- Áreas Protegidas:
https://drive.google.com/drive/u/1/folders/1UWPBcv7Oy_FdL6xNFsZIW3cbP17BgocH (arquivos no drive acessível por solicitação [só para controle], já que a cessão do ISA demanda restrição de uso)
- Plano Nacional de Logística (malha de transportes a ser construída no país até 2025):
<https://www.epl.gov.br/rede-georeferenciada-pnl-2025> - exemplo de shapes usados para serem comparados às APs

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:beatriz.murer:projeto



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