

Trabalho Final Juliana

Proposta 1

Trabalho com indicação de áreas prioritárias para conservação e utilizo técnicas de geoprocessamento como ferramenta. Minha proposta me ajudaria a escolher minhas paisagens de análise, mas também pode ajudar outras pessoas que precisam escolher áreas de estudo, mesmo que seu trabalho posterior seja diferente.

Minha proposta consiste em fazer uma função que indique áreas com base critérios estabelecidos. Essa função auxiliaria a escolha de áreas (paisagens) de análise/estudo.

Para o meu objetivo os critérios serão baseados em arquivos importados em formato de matriz (arquivos baseados em mapas). Não sei se vou conseguir fazer uma coisa geral onde cada um que utilizar possa incluir seu critério, senão minha função funcionará com a inclusão de um arquivo em matriz com informação de % de habitat ou cobertura vegetal e a partir de argumentos indicando como essa matriz deverá ser priorizada.

No meu caso gostaria de escolher 30 paisagens com 2kms de raio que possuam % de habitat contínuo. Então penso que o número de paisagem, o tamanho já podem ser bons argumentos. Tenho ainda que pensar melhor sobre como tratar com os critérios.

Proposta 2

Pensando...

Julianan, a proposta está interessante. Acredito que uma possível maneira de trabalhar com paisagens como input é transformando elas em uma matriz binária, em que o valor 1 pode ser considerado habitat e 0, não habitat. Aí então você poderia definir o critério de seleção de áreas. Acho que para o trabalho da disciplina você pode desenvolver uma função com apenas um critério (e.g. % cobertura contínua) ou dois.

— *Cristiane*

Função

```
#--Simulação de dados para função 1
#Se eu tenho uma tabela com hexagonos, espécies, n de indivíduos e cobertura
vegetal
# e eu quero uma tabela com riqueza por paisagem

#crie um objeto chamado temp e salve valores de uma distribuição poisson
```

```
temp<-rpois(n=10000, lambda=30)
#selecione dentro dos valores de temp os que sejam maiores que 30
#(estou fazendo isso simplesmente para tirar os zeros e tentar chegar mais
próxima a uma comunidade real)
temp<-subset(temp, temp>30)
#crie um objeto chamado HEX com valores de 1 a 500 com o comprimento do
objeto temp
HEX<-as.character(sample(1:500, replace=T, size=length(temp)))
#crie um objeto chamado especie e cole a palavra espécie com os valores do
objeto temp
ESPECIE<-as.character(paste("especie", temp))
#crie um objeto chamado dados e concatene em um data.frame os valores dos
objetos hex e especie
DADOS<-data.frame(cbind(HEX, ESPECIE))

#crie um objeto chamado hex com valores do 1 a 500
HEX<-(1:500)
#crie um objeto chamado cobveg e salve nele valores de 0 a 50 com
comprimento de n=500
COBVEG<-sample(0:50, 500, replace = T)
#crie um data.frame chamado x e salve nele os valores de hex e cobveg
x<-data.frame(cbind(HEX, COBVEG))

#Função preliminar
#função para criar tabela com dados de riqueza e % de habitat para cada
paisagem.
#Tabela para ser utilizada na próxima função se o usuário não tiver dados de
riqueza prontos.
f.riqueza<-function(x)
{
#crie um data.frame chamado a e faça um tabela de contingência com as
colunas hex e especie do objeto x
a<-data.frame(table(HEX=x$HEX, ESPECIE=x$ESPECIE))
#selecione somente as informações que tiverem valores maiores que 0 na
coluna de Freq e salve no objeto a
a<-subset(a, Freq>0)
#crie um data.frame chamado b e faça uma tabela de contingência com os
valores de hex
b<-data.frame(table(HEX=a$HEX))
#dê o nome de RIQ para a segunda coluna do objeto b
names(b)[2]<-("RIQ")

#crie um objeto chamado dados2 e junte os valores de b e x com base nos
nomes de b e x,
#sendo que não é para copiar os valores que não tiverem correspondência
DADOS2<-merge(b,x,by=intersect(names(b),names(x)),all=F )

#retorne dados2 para o usuário que utilizar essa função
return(DADOS2)
}
```

```
#Teste 1
f.riqueza(DADOS)

#Dados para função 2
#---Simulação de dados
#criação de vetor (simulando hexagonos - paisagens). Cenário simulado onde
teríamos 500 paisagens possíveis para serem analisadas.
HEX<-(1:500)
#criação de vetor (simulando riqueza de espécies).
RIQ<-sample(c((sample(150:250, 50,replace=T)),
(sample(100:150,100,replace=T)),
(sample(50:100,100, replace=T)), (sample(0:50,250,replace=T))))
#criação de vetor (simulando porcentagens de cobertura vegetal em cada
paisagem.
COBVEG<-sample(0:50, 500, replace=T)
#construção de um data.frame com concatenando os 3 vetores criados acima.
DADOS<-data.frame(cbind(HEX, RIQ, COBVEG))
#retirando todas as informações cuja riqueza for 0.
DADOS<-subset(DADOS, RIQ>0)

#FUNÇÃO PRINCIPAL
#---Função para seleção de paisagens com base em % de cobertura vegetal e nº
de espécies

f.select.paisagens<-
function(x,nminriq=1,nmaxriq=10000,quantil=T,probabilidades=NULL,nquebras=NU
LL,npaisagens=1)
{

#selecionando as paisagens com base na riqueza de espécies, de acordo com o
desejado pelo usuário.
xx<-subset(x,RIQ>=nminriq&RIQ<=nmaxriq)

#criando um objeto chamado faixa onde tenhamos duas informações: o valor
mínimo e máximo da coluna 3, relacionada, nesse caso,
#a % de cobertura vegetal.
    faixa=range(xx[,3])
#fazendo um condicional
#Se quantil for igual a TRUE
    if(quantil == T)
    {
#crie um objeto chamado CUTOFF e salve nele o resultado de: valores da
coluna 3 que sejam correspondentes
#as quebras de acordo com as probabilidades inseridas na função
        CUTOFF <- quantile(xx[,3], probs=probabilidades)
#Se não, ou seja, se quantil == T, for false.
    } else
```

```
{
#crie um objeto chamado CUTOFF e salve nele o resultado de: valores da
coluna 3 que sejam
#correspondentes a nquebras, ou seja, proporções iguais.
  CUTOFF <- quantile(xx[,3], probs=seq(from=0,to=1,
by=1/(nquebras+1)))
}
#Faça com que o último elemento de CUTOFF receba o valor de Inf
  CUTOFF[length(CUTOFF)]<-Inf

#crie uma coluna nova no data.frame xx com o nome de paisagens
#e copie todos os valores da coluna COBVEG do mesmo data.frame
xx$PAISAGENS<-xx$COBVEG
#transforme os valores em xx$paisagens em classes. E faça isso de acordo com
os valores de CUTOFF.
xx$PAISAGENS<-cut(xx$COBVEG, breaks=as.numeric(CUTOFF),
labels=1:(length(CUTOFF)-1), include.lowest = T, right = F)

#crie um objeto chamado seleção com nada dentro
selecao<-NULL
#faça um looping que rode para cada classe de xx$paisagens criado acima, e
faça isso em ordem.
for(i in sort(unique(xx$PAISAGENS)))
  {
#crie um objeto auxiliar e guarde nele uma seleção de xx onde os valores na
coluna de paisagem sejam igual a classe i
  AUX<-subset(xx, PAISAGENS==i)
#aleatorize as linhas desse objeto e salve nesse mesmo objeto auxiliar
somente as primeiras nlinhas(npaisagens) e todas as colunas
  AUX<-xx[ sample(1:nrow(xx), size=npaisagens) , ]
#salve no objeto seleção um data.frame que junte seleção com o auxiliar,
para que cada vez que rodar o for
#as informações sejam adicionadas a esse objeto
  selecao<-data.frame(rbind( selecao, AUX ))
  }
#crie um objeto chamado resultado que contenha uma lista com o objeto
seleção e CUTOFF
resultado<-list(selecao, CUTOFF)
#retorne para o usuário que utilizar essa função o objeto resultado
return(resultado)

}

#---Teste 1
f.select.paisagens(DADOS,quantil=F, nquebras=5)

#---Teste 2
f.select.paisagens(DADOS,nminriq=5,quantil=T,
probabilidades=c(0,0.1,0.3,0.5,0.8,1), npaisagens=2)
```

```
==== Documentação para a Função Principal criada ====
```

```
package:maptools          R Documentation
```

```
~~Function to select landscapes ~~
```

```
Description:
```

```
~~ Select landscapes based on percentage of forest cover. ~~
```

```
Usage:
```

```
~~f.select.landscape(x,quantil,probabilidades...)
```

```
##Default S3 method:
```

```
f.select.paisagens<-
```

```
function(x,nminriq=1,nmaxriq=10000,quantil=T,probabilidades=NULL,nquebras=NU  
LL,npaisagens=1)
```

```
~~
```

```
Arguments:
```

```
~~
```

```
x          data frames, matrix
```

```
nminriq    specifications of minimum value of some criterion/variable  
(richness for example).
```

```
nmaxriq    specifications of maximum value of some criterion/variable  
(richness for example).
```

```
quantil    logical; if TRUE, the data will be select based on especific  
probabilits to be add on probabilidades.
```

```
            If FALSE data will be select based n breaks of igual probabilits  
to be add on nquebras.
```

```
probabilidades  numeric vector of probabilities with value between 0 and 1.
```

```
nquebras      number of breaks witch the data will be select.
```

```
npaisagens    number of landscapes to be select on each band of probability.
```

```
~~
```

```
Details:
```

```
~~ x must have 3 coluns. x[,1] should have the landscapes, x[,2]  
richness or other numeric variable intended to be used as exclusion
```

criterion to select the

landscapes, x[,3], invariably, percentage of the variable witch is intended to be used as a continous or categorical criterion to select the landscapes (% forest

cover).

~~

Value:

~A List will be returned with:

comp1 : the breaks probabilities and respective values.

comp2 : subset of the DATA corresponding with the selecst landscapes.

Note:

~~This function was created to select landscapes based on a continous percentage of forest cover and a minimum value of richness, but it will work with other

variables. The second colomn included on the data must be a discrete variable which can be used to exclude a range of values. The tird column must be a

continuous variable which can be used to classify the landscapes, based on continous ou not, specified values or not, allowing to sample a n number of

landscapes randomly within the classes criated.

~~

Author(s):

~~Juliana Costa Coelho. Master's student at PPGI-applied ecology from the School of Agriculture "Luiz de Queiroz" (University of São Paulo).
Coelho, J. C. 2014.

~~

Examples:

```
HEX<- (1:500)
```

```
RIQ<- sample(c((sample(150:250, 50, replace=T)),
```

```
(sample(100:150,100, replace=T)),
```

```
(sample(50:100,100, replace=T)), (sample(0:50,250, replace=T))))
```

```
COBVEG<-sample(0:50, 500, replace=T)
```

```
DADOS<-data.frame(cbind(HEX, RIQ, COBVEG))
DADOS<-subset(DADOS, RIQ>0)

##Example 1 - with breaks
f.select.paisagens(DADOS,quantil=F, nquebras=5)

## Example 2 -with specified probabilities
f.select.paisagens(DADOS,nminriq=5,quantil=T,
probabilidades=c(0,0.1,0.3,0.5,0.8,1), npaisagens=2)
```

[funcaopreliminar.r](#)

[funcaoprincipal.r](#)

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2014:alunos:trabalho_final:coelho.ju.08:trabalho_final 

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