

# Proposta de trabalho final

Mapear a distribuição dos organismos nos ajuda a compreender melhor qual a relação destes com os ambientes favoráveis a sua ocorrência. Além disso, conhecer apropriadamente as áreas em que muitas espécies ocorrem em simpatria nos ajuda a otimizar nossos esforços destinados a conservação. No entanto, tanto análises espaciais em escala regional quanto em escala local são importantes para que possamos estabelecer estratégias eficazes e captar varialibilidade ambiental suficiente para beneficiar determinados grupos de espécies.

O objetivo do meu projeto é automatizar o mapeamento de conjuntos de diferentes áreas baseado na distribuição das espécies presentes nessas áreas. Por exemplo, somando as distribuições das espécies A, B e C, quero que possamos extrair de uma área maior (escala regional) a área ocupada por essas espécies, e assim sucessivamente para diferentes conjuntos de espécies. Essa função poderá ser usada para, por exemplo, extrair da área total de uma ecorregião (e.g. Mata Atlântica) áreas de endemismo obtidas por análises biogeográficas; Poderá também ser usada para extrair de uma área maior apenas a área de distribuição de um conjunto de espécies ameaçadas ou para definir os limites das áreas que devem ser amostradas para otimizar o encontro de um conjunto de espécies raras; Outra aplicação seria para estudos comparativos sobre área de ocupação de grupos de espécies que apresentam características semelhantes em detrimento da variação de cobertura em uma determinada área de estudo (escala local).

Inúmeras aplicações podem ser sugeridas uma vez que a função esteja operando adequadamente a depender exclusivamente das perguntas a serem respondidas e da criatividade de quem pretende responde-las.

Minha proposta está estruturada começando por um plano B (menos complexo) que está aninhado no plano A (mais complexo e dependente do funcionamento de B).

Para isso, a função pressupõem a existência de alguns objetos na área de trabalho:

 **OBS:** os argumentos de objetos que devem ser incluídos no código da função aparecem entre parenteses e aspas simples. ex.: ('objeto')

**1.** Um grid ('grid') que representa uma determinada área geográfica. Em que cada célula é codificada com um número de identificação.

**2.** Uma matriz de presença e ausência ('mpa'):

**Matriz:**

<b>mpa</b>	<b>Código das Células do Grid</b>				
<b>Espécie</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>sp1</b>	1	0	1	0	1
<b>sp2</b>	0	1	1	0	0
<b>sp3</b>	0	0	0	0	0

<b>mpa</b>	<b>Código das Células do Grid</b>				
<b>Espécie</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>sp4</b>	1	1	0	0	1
<b>sp5</b>	0	0	0	0	1
<b>sp6</b>	0	1	0	1	0
<b>sp7</b>	1	1	1	0	0
<b>sp8</b>	0	0	1	0	1
<b>sp9</b>	1	0	0	0	1
<b>sp10</b>	0	0	1	1	0

Em que as colunas são codificadas com o número de identificação de células do grid ('grid'); zero (0) representa ausência de uma dada espécie em uma determinada célula, e um (1) representa a presença; e

3. Um data.frame ('lsp') contendo em uma coluna uma lista de espécies e em outra um código que representará o agrupamento das espécies de interesse:

<b>Espécie</b>	<b>Cód. Área</b>
<b>sp1</b>	1
<b>sp2</b>	1
<b>sp3</b>	2
<b>sp4</b>	2
<b>sp5</b>	2
<b>sp6</b>	3
<b>sp7</b>	3
<b>sp8</b>	4
<b>sp9</b>	4
<b>sp10</b>	4

Cada código representará a área que a distribuição de espécies agrupadas ocupa.

Ao final, a função deve, sob escolha d@ usuári@ gerar um arquivo .shp para cada área.

Para isso, precisamos que o código:

1. selecione na lista de espécies ('lsp') as espécies pertencentes a cada área codificada:

```
lsp[,1][lsp[,2]=='cód']
```

2. selecione na matriz ('mpa') as linhas referentes a cada espécie e atribua a um objeto 'areaN':

```
areaN <- mpa[lsp[,1][lsp[,2]=='cód.areaN'],]
```

3. selecione no grid ('grid') apenas as células em que pelo menos uma das espécies esteja presente de acordo com o objeto areaN criado no passo anterior:

```
'I got it, believe me'
```

4. repita os passos de 1 a 3 para os outros códigos de área:

```
for(i in 1:length(unique(lsp[2])))  
{  
  tudo que está acima  
}
```

O *for* do passo 4 deve ter algumas peculiaridades: Precisamos que o ciclo adicione informações a um mesmo objeto enquanto a informação da coluna de código de áreas se repetir e, que crie um objeto novo cada vez que o R ler a coluna de código das áreas e encontrar algo diferente do que tinha encontrado antes

Ainda nessa proposta pretendo oferecer @ usuári@ o argumento lógico ('plot') que por padrão terá o valor 'FALSE'. Com o argumento 'plot' quero que a função plote no dispositivo gráfico o grid das áreas formadas. Caso @ usuári@ escolha TRUE, elx poderá incorporar à função o objeto ('shp') que nada mais é que o shapefile com o mapa original da área de estudo. Assim, o argumento plot terá que plotar o 'shp' e por cima dele (add.new=TRUE, da própria função plot) plotar as áreas criadas com o mapeamento.

Pode ser que @ usuári@ queira plotar apenas uma ou duas das áreas formadas por vez, nesse caso, o argumento 'plot=TRUE' deve chamar a função readline() solicitando que @ usuári@ escolha qual áreas pretende plotar na imagem. As possíveis respostas seriam, o número/nome da área ou uma resposta que finalize o plot.

Satisfeit@ com as áreas plotadas, outra readline() interrogará @ usuári@ sobre prosseguir para o passo 5 "salvar as áreas em arquivos .shp? (s/n)":

5. criar um objeto .shp para cada área a partir dos dados extraídos do grid original

Estou estudando pacotes de análises espaciais pra ver como fazer isso, mas certamente existe uma função semelhante a 'write.shp' para ser usada nesse caso

## Proposta mais complexa A1:

### DEPENDENTE DO FUNCIONAMENTO DA PROPOSTA B

Visando otimizar a visualização de informações importantes, seria interessante que @ usuári@ tivesse a opção de solicitar ao R que os grids das áreas mapeadas fosse exibido com suas células apresentando uma cor qualquer (de escolha do usuário) com transparências diferentes de acordo com a riqueza de cada célula em relação a riqueza total da área mapeada.

Para isso, precisamos que o código:

1. Faça o cálculo de proporção  $n/m$ , em que  $n$  é o número de espécies por célula e  $m$  é o número de espécies total de cada área;

```
n <- apply(areaN, 2, sum)
m <- length(rownames(areaN))
n/m
```

**2.** Atribua valores decrescentes de transparência para células: a) com mais de 70% das espécies da área; b) contendo entre 70% e 30% das espécies; e c) contendo menos que 30%; de modo que, células mais ricas (dentro desse gradiente de riqueza) sejam mais escuras.

Para essa proposta, pretendo adicionar à função o argumento lógico ('prop'), com padrão 'FALSE', em que 'TRUE' executaria o proposto acima.

## **Proposta mais complexa A2 - 'Striving for the best':**

### **DEPENDENTE PARCIALMENTE DA PROPOSTA A1; IGUALMENTE DEPENDENTE DA PROPOSTA B**

Para fins de praticidade seria interessante que todas as áreas criadas fossem agrupadas em um único arquivo .shp, o que possibilitaria que @ usuári@ poupasse tempo carregando apenas uma vez as áreas em um Sistema de Informação Geográfica (SIG) e otimizasse as imagens a serem apresentadas.

Para isso, precisamos que o código:

**1.** Agrupe todas as áreas formadas na proposta B em um único arquivo "areas.shp" de modo que elas possam ser tratadas como objetos distintos dentro do arquivo. Ou seja, possam ser editadas individualmente, selecionadas ou desselecionadas para serem projetadas em um SIG, etc.

**2.:** Atribua uma cor diferente para cada área com valores decrescentes de transparência para células: a) com mais de 70% das espécies da área; b) contendo entre 70% e 30% das espécies; e c) contendo menos que 30%; de modo que, células mais ricas (dentro desse gradiente de riqueza) sejam mais escuras.

Estou estudando arquivos similares pra entender como se organizam nesse quesito

Para a proposta A2, pretendo adicionar à função um argumento lógico ('grp'), 'TRUE' por padrão, em que @ usuári@ possa escolher se quer arquivos agrupados (TRUE) ou separados (FALSE).

Com o proposto até o momento nossa função fica:

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

demandando a entrada de três objetos, e com dois passos interativos

OBS: 'mapar' vem de 'map area'

## Resultado esperado:



Na imagem ao lado<sup>1)</sup>, **EE1** e **EE3** seriam equivalentes, por exemplo, a áreas codificadas como 1 e 3 no data.frame 'lsp', carregadas no arquivo "areas.shp", caso consigamos desenvolver a função até a proposta mais complexa A2 passando pela proposta mais complexa A1. Caso o resultado da proposta mais complexa A2 não seja alcançado, **EE1** seria o arquivo "area1.shp" e **EE3** seria o arquivo "area3.shp" ambos gerados pela proposta mais simples (B), mas com os parâmetros de transparência obtidos pela proposta mais complexa A1.

Essa função parece desafiadora, mas você também me parece preparado.

Entretanto, tome muito cuidado: Shape files no R são frequentemente difíceis de se lidar, e minha experiência (anedótica e rasa, não se compara à sua experiência mas mesmo assim releve) com shapes me diz que os pacotes que lidam com shape files são "temperamentais": as vezes não funcionam completamente, as vezes não funcionam de um PC pro outro, de uma versão do R pra outra etc...

A ideia da proposta B seria ter algo completamente independente, para caso a situação fique tão feia/tensa que seja necessário abandonar completamente a proposta A. Como você aninhou uma proposta na outra, isso te coloca em maus lençóis caso ocorra um problema grande e inesperado (p.e. os pacotes que lidam com shapes serem temperamentais e difíceis de lidar no curto prazo para preparação da sua função).

Eu sugiro que você faça a sua proposta B como proposta A, pois ela já apresenta a complexidade necessária. Ao mesmo tempo também te sugiro pensar numa nova proposta B completamente independente da proposta A.

Se quiser conversar mais sobre as funções (ou pedir conselhos, dicas etc) é só escrever aqui que pretendo revisar esta página dia 20/06. Vc tbm pode me mandar uma mensagem por whatsapp (11) 9-9199-3842.

Matheus Januario

Oi Matheus, obrigado pelas dicas. Fico feliz em responder que a função está rolando super bem até o momento. Já me informei sobre como lidar com arquivos de shape no R para o propósito da função e consigo executar (ainda não na forma de função) as propóstas até o nível de complexidade A1. Estou agora estudando para finalizar a ideia com a A2, e para rodar tudo que foi feito até o momento como função. Abraços, JP.

1)

[Retirada de Nogueira et al. 2011](#)

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

Permanent link:  
[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2019:alunos:trabalho\\_final:joaopaulo.valencar:prop](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:joaopaulo.valencar:prop) 

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