-
-
-

Metapopulações com chuva de Propágulos - Roteiro em R passo-a-passo

Neste exercício, iremos simular a dinâmica de uma metapopulação em que as probabilidades de extinção e de colonização de manchas pela nossa *espécie virtual* são constantes (modelo de chuva de propágulos). Neste modelo há sempre uma fonte constante de imigrantes que podem colonizar qualquer mancha vazia, por isso o nome “chuva de propágulos”.

Nesse modelo, a variação da fração de manchas ocupadas no tempo é descrita pela seguinte equação geral:

$$\frac{df}{dt} = I - E $$

onde I é a taxa de entrada de migrantes, e E a taxa de saída. A partir dele podemos definir um modelo simples para a dinâmica de ocupação de manchas que formam uma metapopulação:

$$\frac{df}{dt} = p_i(1 - f)- p_e f $$

onde p_i é a probabilidade de imigração ou colonização, p_e é a probabilidade de extinção e *f* é a fração de manchas ocupadas (número de manchas ocupadas / número total de manchas). Esses, portanto são os parâmetros do nosso modelo.

Apesar de ser matematicamente o modelo mais simples, historicamente esse modelo foi descrito mais recentemente que os outros modelos de metapopulações que iremos apresentar [1)]. Uma característica importante desse modelo é que ele não é fechado, o que biologicamente é mais complexo.

Simulando Metapopulações

Em primeiro lugar, vamos estabelecer a probabilidade de colonização de manchas vazias (p_c), a

probabilidade de extinção em manchas ocupadas (p_e) e a fração inicial de manchas ocupadas (f_i) como 30%, 15% e 40%, respectivamente.

```
pc=0.3
pe=0.15
fi=0.4
```

Agora que já temos os parâmetros do nosso modelo, vamos criar nossa paisagem virtual. Essa paisagem deve ser constituída por manchas de habitat que podem estar ocupadas ou não. Para tanto, criaremos uma matriz de 10 linhas e 10 colunas, sendo que cada célula dessa matriz representará uma mancha. Serão, portanto, 100 manchas.

```
paisag=array(0,dim=c(10,10,1))
paisag
```

Muito bem, mas temos somente a matriz no tempo inicial. Para acompanhar a dinâmica de ocupação de manchas é preciso criarmos uma terceira dimensão: o tempo. Por enquanto vamos criar 10 passos no tempo.

```
paisag=array(0,dim=c(10,10,10))
paisag
```

Ótimo! Agora temos nossa paisagem em 10 tempos diferentes.

A brincadeira aqui é como se tivéssemos um tabuleiro com 100 casas, representando nossa paisagem, com 40 delas ocupadas por peças vermelhas (representando nossa espécie) no início. A cada rodada, para cada casa do tabuleiro, sorteamos um número aleatório de 1 a 100 (um super dado com 100 lados! [2)]. Caso a casa esteja ocupada e o número sorteado estiver entre 1 e 15 (probabilidade de 0.15), retiramos a pedra; caso a casa esteja desocupada e o número sorteado for entre 1 e 30 (probabilidade de 0.3), colocamos uma nova peça. Caso essas condições não aconteçam mantemos a casa do tabuleiro da maneira que encontramos

Finalmente, antes de começarmos a brincadeira, precisamos definir quais manchas estarão ocupadas no tempo inicial. Vamos combinar que quando a mancha está ocupada ela recebe o valor 1 e quando está vazia recebe o valor 0. Do jeito que a matriz está, repleta de zeros, nenhuma mancha está ocupada. Então vamos preencher ao acaso algumas manchas, usando um f_i de 40%:

```
fi
1-fi
rep(1, fi*100)
ocor<-rep(1,fi*100)                    #ocupação
nocor<-rep(0,(1-fi)*100)               #não ocupação
nocor
c(ocor,nocor)
sample(c(ocor,nocor))                  #amostra aleatória
paisag[,,1]<-sample(c(ocor,nocor))     #aplicando sobre a paisagem
paisag[,,1]                            #paisagem inicial com ocupação
inicial aleatorizada
```

Agora sim, estamos prontos para a simulação.

O passo seguinte é fazer as coisas acontecerem! Mas para isso temos que ter em mente todas as possibilidades:

- **Manchas ocupadas**
  - podem permanecer ocupadas ($1-p_e = 0,85$)
  - podem sofrer extinção local ($p_e = 0,15$)
- **Manchas vazias**
  - podem permanecer vazias ($1-p_c = 0,7$)
  - podem ser colonizadas ($pc = 0,3$)

Calma, não se assuste. O monstrinho abaixo vai tratar apenas das manchas que estavam ocupadas no tempo inicial.

```
sum(paisag[,,1]) # numero de manchas ocupadas no tempo 1
paisag[,,2][paisag[,,1]==1]<-
sample(c(0,1),sum(paisag[,,1]),replace=T,prob=c(pe,1-pe))
paisag[,,2]
```

Se você observar atentamente, algumas manchas que estavam ocupadas (1) ficaram vazias (0):

```
paisag[,,1]         # paisagem no tempo inicial
paisag[,,2]         # paisagem depois de um passo no tempo
```

Estes foram eventos de extinção local. Note que o número de manchas ocupadas diminuiu.

```
sum(paisag[,,1])          # total de manchas ocupadas no tempo inicial
sum(paisag[,,2])          # total de manchas ocupadas no tempo 2
```

Mas essa é apenas metade da história, não? Estamos esquecendo que as manchas vazias podem ser ocupadas. Vamos simular a colonização com um outro monstrinho:

```
length(paisag[,,1])               ## esse é o tamanho da nossa paisagem ou o
número de manchas!
nmanchas=length(paisag[,,1])      ## vamos guardar esse valor que não muda
durante toda a simulação
paisag[,,2][paisag[,,1]==0]<-sample(c(0,1),nmanchas-
sum(paisag[,,1]),replace=T,prob=c(1-pc,pc))
paisag[,,2]
```

Compare a paisagem no tempo inicial e no tempo 2 e veja se no final o número de manchas ocupadas aumentou ou diminuiu.

```
paisag[,,1]
paisag[,,2]

sum(paisag[,,1])          # total de manchas ocupadas no tempo inicial
sum(paisag[,,2])          # total de manchas ocupadas no tempo 2
```

Note que algumas manchas que estavam ocupadas no tempo 1 ficaram vazias no tempo 2 e vice-

versa. Note também que algumas manchas continuaram vazias e outras continuaram ocupadas. O próximo passo é calcular a fração de manchas ocupadas inicialmente (f1) e no tempo 2 (f2) e depois a diferença entre elas:

```
nmanchas
f1=sum(paisag[,,1])/nmanchas         # fração das manchas ocupadas no tempo
1
f2=sum(paisag[,,2])/nmanchas          # fração das manchas ocupadas no tempo
2

f1
f2
f2-f1
```

Eis a dinâmica de metapopulações! Mas até aqui vimos apenas a variação do f do tempo 1 para o tempo 2. Restam ainda 8 passos para chegarmos no tempo 10, só que para não termos que repetir todo o trabalho 8 vezes usaremos um comando chamado **for**, que fará o serviço por nós:

```
paisag=array(0,dim=c(10,10,10))
nmanchas=length(paisag[,,1])
paisag[,,1]=sample(c(rep(0,((1-fi)*nmanchas)),rep(1,fi*nmanchas)))
resultado=numeric()
for(t in 2:10)
               {
           paisag[,,t][paisag[,,(t-1)]==1]<-
sample(c(0,1),sum(paisag[,,(t-1)]) ,replace=T, prob=c(pe,1-pe))
           paisag[,,t][paisag[,,(t-1)]==0]<-sample(c(0,1),nmanchas -
sum(paisag[,,(t-1)]), replace=T,prob=c(1-pc,pc))
           resultado[t-1]=sum(paisag[,,t])/nmanchas
          }
resultado
```

Vôa lá! Agora temos as frações de manchas ocupadas (f) ao longo do tempo. Só para deixar nossos resultados mais bonitos, vamos colocá-los em uma tabela:

```
resultado=data.frame(t=1:10,f=c(fi,resultado))
resultado
```

Agora, para ficar mais bonito ainda, vamos criar um gráfico com esses resultados:

```
plot(1:10,resultado$f,type="l",xlab="Tempo",ylab="Fração de manchas
ocupadas",
ylim=c(0,1),main="Dinâmica de ocupação de manchas",font.lab=2,lwd=2)
```

Como nossa filosofia de vida é (ou pelo menos deveria ser) melhorar sempre, vamos acrescentar a esse gráfico uma informação muito importante: a fração de manchas ocupadas no equilíbrio (F). O F é calculado da seguinte forma:

$$F=\frac{p_i}{p_i+p_e}$$

Portanto...

```
F=pc/(pc+pe)
F
```

Agora que conhecemos o F vamos colocá-lo no gráfico na forma de uma linha horizontal:

```
plot(1:10,resultado$f,type="l",xlab="Tempo",ylab="Fração de manchas
ocupadas",
ylim=c(0,1),main="Dinâmica de ocupação de manchas",font.lab=2,lwd=2)
abline(h=F,col=2,lwd=2,lty=2)
```

O serviço sujo está feito, agora é hora da diversão! Como somos bonzinhos, resolvemos criar uma função para facilitar a vida de vocês. Com essa função vocês poderão variar os parâmetros do nosso modelo à vontade, sem medo de ser feliz. Por favor, retribuam a gentileza e testem vários valores para cada parâmetro e vejam o que acontece com nossas metapopulações virtuais. Abaixo a função:

```
metapop=function(tf,cl,ln,fi,pc,pe){
    paisag=array(0,dim=c(ln,cl,tf))
        nmanchas=cl*ln
    paisag[,,1]=matrix(sample(c(1,0),nmanchas,prob=c(fi,1-fi),
replace=T),ln,cl)
    resultado=numeric()
    for(t in 2:tf){
           paisag[,,t][paisag[,,(t-1)]==1]<-
sample(c(0,1),sum(paisag[,,(t-1)]), replace=T, prob=c(pe,1-pe))
           paisag[,,t][paisag[,,(t-1)]==0]<-sample(c(0,1),cl*ln-
sum(paisag[,,(t-1)]), replace=T, prob=c(1-pc,pc))
           resultado[t-1]=sum(paisag[,,t])/(cl*ln)
          }

    F=pc/(pc+pe)

    plot(1:tf,c(fi,resultado),type="l",xlab="Tempo",ylab="Fração de manchas
ocupadas",
    ylim=c(0,1),main=paste("Chuva de Propágulos","\n cl=",cl," ln=",ln,"
fi=",fi," pc=",pc," pe=",pe),font.lab=2,lwd=2)
    abline(h=F,col=2,lwd=2,lty=2)
      return(paisag)
    }
```

Maravilha! A função acima tem 6 argumentos (`tf`, `cl`, `ln`, `fi`, `pc` e `pe`), mas para que a função funcione é preciso atribuir um valor para cada argumento, como no exemplo abaixo:

```
metapop(tf=100,cl=10,ln=10,fi=0.5,pc=0.3,pe=0.15)
```

Parâmetros da função metapop

- tf - número total de passos no tempo
- cl - número de colunas da matriz *paisag*
- ln - número de linhas da matriz *paisag*

- fi - fração de manchas ocupadas inicial
- pc - probabilidade de colonização
- pe - probabilidade de extinção

Perguntas

Agora é com vocês! Tentem responder às seguintes perguntas:

- Quando você rodar várias mais de uma vez a função com os mesmos parâmetros o resultado é o mesmo? Por quê?[3)]
- O que acontece com a variação na fração de manchas ocupadas quando:
  - o número de manchas é muito grande? E quando é muito pequeno?
  - e se as probabilidades de extinção e colonização são muito altas?
  - e se forem muito baixas?
- E quando a fração inicial (f_i) é muito diferente da fração em equilíbrio (F)?
- Quais condições levam à extinção da população na paisagem neste modelo?

DICA

Para responder às perguntas acima você precisa comparar resultados de diferentes simulações. Abrir espaço para mais de um gráfico na janela gráfica do R vai ajudar bastante. Para ter 4 gráficos digite na linha de comando do R:

```
par(mfrow=c(2,2))
```

Os dois números indicam o número de linhas e colunas desejadas, na janela gráfica. No caso, pedimos duas linhas e duas colunas, ou seja, espaço para quatro gráficos. Para voltar a um gráfico apenas na janela digite:

```
par(mfrow=c(1,1))
```

———

Para vocês que chegaram vivos até aqui, uma recompensa. Rode os comandos abaixo e descubra:

```
anima=function(dados){
        x11()
    for(i in 1:dim(dados)[3])
        {
    image(dados[,,i], main=("Ocupação de manchas"),sub=paste("simulação no.=
", i), col=c("white","red"), bty="n",xaxt='n',yaxt='n')
    grid(dim(dados)[1],dim(dados)[2])
    Sys.sleep(.2)
    }
}
```

```
simula1<- metapop(20,10,10,1.0, 0.4,0.2)
anima(simula1)
```

Para fazer rodar a animação você precisa apenas salvar o resultado da função *metapop* e chamar esse objeto na função anima. Para modificar parmâmetros do modelo é só rodar outra simulação.

```
simula2=metapop(tf=25,cl=100,ln=100,fi=.01,pc=0.2,pe=0.5)
anima(simula2)
```

Para saber mais

- **Gotelli, N. 2007. Ecologia.** Londrina, Ed. Planta. Capítulo 4.
- **Stevens, M. H. 2009. A primer of ecology with R.** New York. Springer.Capítulo 4.
- **Gotelli, N. 1991. Metapopulation models: the rescue effect, the propagule rain, and the core-satellite hypothesis.** The American Naturalist, 138: 768-776. pdf no site do autor

R, uma população, metapopulações, chuva de propágulos

1) Esse modelo foi descrito por Nicholas Gotelli em 1991, veja referências no final dessa página

2) Sim, isso existe!http://en.wikipedia.org/wiki/Zocchihedron

3) Você deve entender isto para responder várias das questões seguintes.

From:
http://ecovirtual.ib.usp.br/ -

Permanent link:
http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:metap_uma:metap_chuvar_passo

Last update: **2016/05/10 07:19**