Melina Oliveira Melito

Estou no início do doutorado em Ecologia pela USP e irei pesquisar sobre estabilidade de paisagens dinâmicas em frente a diferentes regimes de distúrbios.

Exercícios

- exec

Trabalho final

Plano A

O plano A seria o mais “ousado” e com uma forte aplicação para meu projeto de doutorado, espero conseguir executa-lo! A idéia é construir uma função para simulação espacial e temporal de elementos (categorias) da paisagem. O input da função seriam mapas de uma série temporal das classes de vegetação de uma paisagem e também mapas topográficos. A função então calcularia a probabilidade de transição ou permanência de cada pixel nas categorias de vegetação da paisagem considerando a posição geográfica do pixel. Desta forma, o output da função seria uma lista contendo a simulação espacial e temporal da vegetação na paisagem e a matriz de transição para cada um desses períodos.

Plano B

Esse é engraçadinho e pode ser uma ferramenta vital para a difícil escolha de qual bandejão (restaurante universitário) o estudante da USP deve comer! A função estimaria a qualidade do bandejão a partir de parâmetros como verba recebida para compra dos alimentos, número de usuários por dia (ou semana, mês) do bandejão, tempo médio de espera na fila, diversidade de pratos, número de dias de atendimento (dias úteis). O output seria uma escala de 0 a 10 da qualidade do bandejão.

Comentários

A proposta A é boa, parece viável, mas acho que é importante explicitar como seria o formato do mapa a ser inserido e o cálculo das probabilidades de transição se você já tem isso em mente. Se não, me parece que essa é a parte mais desafiadora e que tomará mais tempo. Sua proposta B é engraçadinha, se tiver esses dados disponíveis ok, mas me parece muito simples também. Há um salto muito grande entre A e B!

Sara Mortara

Resposta

Valeu pelas considerações Sara! Os mapas serão em formato txt, assim a entrada é uma matriz em que cada categoria da paisagem é representado por um número inteiro! Ainda estou vendo como irei calcular as probabilidades de transição... Conversei com o Alê e ele me sugeriu simplificar a função do Plano A. Assim, pretendo fazer a simulação espacial e temporal dos elementos da paisagem sem considerar a posição geográfica. O output continua sendo o plot da simulação e uma lista com as matrizes de probabilidade de transição de cada período.

Plano A - executado

A função executada foi um pouco diferente da proposta. Não deu tempo de inserir a simulação mas eu acrescentei à função o cálculo de matrizes de transição considerando a posição do pixel (célula) de vegetação (estágio inicial ou avançado de sucessão) em relação ao fragmento, ou seja, reclassificando as células de vegetação em borda ou interior do fragmento florestal. O output da função retorna uma lista com as matrizes de transição “normal” e as matrizes após a reclassificação da célula de vegetação.

Matrizes de probabilidade de transição entre categorias de paisagem

```
mapas.mat<- function(mapas, entorno, limite, cat.veg)
{
  total.mapas<- lapply(mapas, table)          # contando o total de celulas
em cada categoria do mapa
  matrizes.transicao<- list()                 # lista das matrizes de
transicao entre cada periodo de tempo (mapa)
  for(i in 1:(length(mapas)-1))               # criando as matrizes para
guardar as probabilidades de transicao
  {
    matrizes.transicao[[i]]<- matrix(NA, nrow=length(total.mapas[[1]]),
ncol=length(total.mapas[[1]]))
  }
  names(matrizes.transicao) <- paste("transicao.t",
1:length(matrizes.transicao), sep="")   # nomeando os objetos da lista com
as matrizes de transicao
  for(i in 1:length(matrizes.transicao))                                  #
nomeando as linhas e colunas dos objetos da lista com as matrizes de
transicao
  {
   colnames(matrizes.transicao[[i]]) <- paste(names(total.mapas[[1]]))
   rownames(matrizes.transicao[[i]]) <- paste(names(total.mapas[[1]]))
  }
  for(l in 1:length(matrizes.transicao))                  # loop para cada
```

```
matriz da lista de matrizes
  {
    for(i in 1:ncol(matrizes.transicao[[l]]))             # loop para cada
variável da coluna da matriz
    {
      for(j in 1:nrow(matrizes.transicao[[l]]))           # loop para cada
variável da linha da matriz
      {
        matrizes.transicao[[l]][j,i] <- table(mapas[[l]]==i &
mapas[[l+1]]==j)[2]/total.mapas[[l]][i]   # cálculo das probabilidades de
transicao
      }
    }
  }
  #veg inicial
  posicoes1<- list()             # lista para guardar as posicoes (linha e
coluna do mapa) da categoria de vegetacao selecionada
  for(i in 1:length(mapas))      # loop para armazenar as posicoes
  {
    posicoes1[[i]]<- which(mapas[[i]]==cat.veg[1], arr.ind=T)
  }
  entorno1<- list()               # lista com uma matriz para cada mapa
onde serao guardados a porcentagem de vegetacao do entorno de cada celula
  for(e in 1:length(mapas))
  {
    entorno1[[e]]<- matrix(NA, ncol=ncol(mapas[[1]]), nrow=nrow(mapas[[1]]))
  }
  # % de vegetacao (inicial e avancada) no entorno para categoria de
vegetacao inicial dos mapas
  for(j in 1:length(mapas))
  {
    lim.line<- nrow(mapas[[1]])-entorno       # limite da borda (pela linha)
do mapa; especificando as linhas que nao sao alvo do loop
    lim.col<-  ncol(mapas[[1]])-entorno       # limite da borda (pela linha)
do mapa; especificando as linhas que nao sao alvo do loop
    for(j in 1:length(posicoes1))
      {
        for(i in 1:dim(posicoes1[[j]])[1])          # loop que se move pelo
dimensao linha da matriz de posicoes
        {
          linha <- posicoes1[[j]] [i,1]             # linha i pela coluna 1
que e a posicao linha do mapa original
          coluna <- posicoes1[[j]] [i,2]            # linha i pela coluna 1
que e a posicao coluna do mapa original
          if( !(linha<entorno | linha>lim.line | coluna<entorno |
coluna>lim.col))    # especificando os limites (bordas) da matriz em que o
loop naoo roda nestas posicoes
          {
            entorno.veg.in<- (sum(mapas[[j]] [(linha-
entorno):(linha+entorno), (coluna-entorno):(coluna+entorno)]==cat.veg[1],
na.rm=T)-1)            # somando as celulas iguais a vegetacao inicial
```

```
            entorno.veg.mad <-  (sum(mapas[[j]]  [(linha-
entorno):(linha+entorno), (coluna-entorno):(coluna+entorno)]==cat.veg[2],
na.rm=T))          # somando as celulas iguais a vegetacao madura
            entorno1[[j]] [linha,coluna] <- (entorno.veg.in +
entorno.veg.mad)/ ((2*entorno+1)*(2*entorno+1)-1)
          }
        }
      }
   }
  # veg avancada
  posicoes2<- list()
  for(i in 1:length(mapas))
  {
    posicoes2[[i]]<- which(mapas[[i]]==cat.veg[2], arr.ind=T)
  }
  entorno2<- list()
  for(e in 1:length(mapas))
  {
    entorno2[[e]]<- matrix(NA, ncol=ncol(mapas[[1]]), nrow=nrow(mapas[[1]]))
  }
  # % de vegetacao (avancada) no entorno para categoria de vegetacao
avancada dos mapas
  for(j in 1:length(mapas))
  {
    lim.line<- nrow(mapas[[1]])-entorno
    lim.col<-  ncol(mapas[[1]])-entorno
    for(j in 1:length(posicoes2))
    {
      for(i in 1:dim(posicoes2[[j]])[1])
      {
        linha <- posicoes2[[j]] [i,1]
        coluna <- posicoes2[[j]] [i,2]
        if( !(linha<entorno | linha>lim.line | coluna<entorno |
coluna>lim.col))
        {
          entorno2[[j]] [linha,coluna] <- (sum(mapas[[j]] [(linha-
entorno):(linha+entorno), (coluna-entorno):(coluna+entorno)]==cat.veg[2],
na.rm=T)-1)/ ((2*entorno+1)*(2*entorno+1)-1)           # somando as celulas
iguais a vegetacao inicial
        }
      }
    }
  }
  mapas.bi<- mapas           # criando os mapas para reclassificar celulas
de vegetacao nas categorias borda e interior
  for(h in 1:length(mapas.bi))
  {
    mapas.bi[[h]][which(entorno1[[h]] < limite)] <- (cat.veg[1])+0.1
    mapas.bi[[h]][which(entorno1[[h]] >= limite)] <- (cat.veg[1])+0.2
    mapas.bi[[h]][which(mapas.bi[[h]] == (cat.veg[1]))] <- (cat.veg[1])+0.1
```

```
    mapas.bi[[h]][which(entorno2[[h]] < limite)] <- (cat.veg[2])+0.1
    mapas.bi[[h]][which(entorno2[[h]] >= limite)] <- (cat.veg[2])+0.2
    mapas.bi[[h]][which(mapas.bi[[h]] == (cat.veg[2]))] <- (cat.veg[2])+0.1
  }

  total.mapas.bi<- lapply(mapas.bi, table)           # contando o numero de
celulas para cada categoria

  matrizes.transicao.bi<- list()                     # criando as matrizes
para guardar as probabilidades de transicao
  for(i in 1:(length(mapas.bi)-1))
  {
    matrizes.transicao.bi[[i]]<- matrix(NA,
nrow=length(total.mapas.bi[[1]]), ncol=length(total.mapas.bi[[1]]))
  }
  names(matrizes.transicao.bi) <- paste("transicao.bi.t",
1:length(matrizes.transicao.bi), sep="")
  for(i in 1:length(matrizes.transicao.bi))
  {
    colnames(matrizes.transicao.bi[[i]]) <-
paste(names(total.mapas.bi[[1]]))
    rownames(matrizes.transicao.bi[[i]]) <-
paste(names(total.mapas.bi[[1]]))
  }
  categoria<- as.numeric(rownames(matrizes.transicao.bi[[1]]))       # vetor
das novas categorias para o loop
  for(l in 1:length(matrizes.transicao.bi))
  {
    for(i in 1:ncol(matrizes.transicao.bi[[l]]))
    {
      for(j in 1:nrow(matrizes.transicao.bi[[l]]))
      {
        matrizes.transicao.bi[[l]][j,i] <- table(mapas.bi[[l]]==categoria[i]
& mapas.bi[[l+1]]==categoria[j])[2]/total.mapas.bi[[l]][i]
      }
    }
  }
  return(c(matrizes.transicao, matrizes.transicao.bi))
}
```

Help da função

```
mapas.mat                                package:nenhum
R documentation



                               Matrizes de probabilidade de transição entre
categorias de paisagem
```

Descrição
Calcula uma matriz com as probabilidades de transição entre as categorias da paisagem para cada par de mapas (tempo 1 e tempo 2). Também calcula a matriz de transição considerando a posição (borda ou interior do fragmento) do pixel de vegetação .

Uso
mapas.mat(mapas, entorno, limite, cat.veg)

Argumentos
mapas Lista contendo os mapas em formato de matrizes e ordenados temporalmente.
entorno Integer. Número de células a serem consideradas para o cálculo da porcentagem de vegetação presente no entorno da célula alvo.
limite Numerical. Limite para classificação da célula alvo em borda e interior a partir da porcentagem de vegetação no entorno da célula alvo. Abaixo desse valor a vegetação é classificada como de borda. Valores iguais ou maiores que o limite classificam a célula como de interior do fragmento.
cat.veg Vetor especificando as duas categorias de vegetação da matriz. A primeira posição do vetor é para vegetação em estágio inicial e a segunda posição para vegetação em estágio intermediário/avançado.

Detalhes
As matrizes devem estar ordenadas na lista com o primeiro objeto a matriz do tempo 1 até a matriz n do tempo n. As categorias de classificação dos elementos do mapa devem ser integers. Utilizar preferencialmente a categoria 9999 para células com dados faltantes, entretanto NAs são aceitos .
A função exige que sejam utilizadas duas categorias sucessionais de vegetação para o cálculo das matrizes. O cálculo para a porcentagem de vegetação no entorno da célula alvo difere entre as duas categorias da vegetação (inicial e intermediária/avançada). O entorno de uma célula alvo de vegetação inicial considera tanto as células de vegetação inicial como intermediárias/avançadas. Para células alvo de vegetação intermediária/avançada apenas células da mesma categoria são consideradas para o cálculo do entorno.

Valores
Retorna uma lista com as matrizes de transição sem e com a reclassificação das células de vegetação pela condição do entorno.

Autora
Melina Oliveira Melito
melina_m12@hotmail.com / mel.melito@usp.br

Referências
Crawley, M. 2007. The R book. John Wiley & Sons Ltd, England.

```
Exemplos
# Exemplo 1
exemplo1.t1<- matrix(round(runif(10000, 1,5)), 100, 100)
exemplo1.t2<- matrix(round(runif(10000, 1,5)), 100, 100)
exemplo1<- list(t1=exemplo1.t1, t2=exemplo1.t2)

mapas.mat(exemplo1, entorno=2, limite=0.6, cat.veg=c(3,4))


# Exemplo 2
a<-rep(c(3,4,5), 5000)
b<-rep(c(1,2), 5000)

exemplo2.t1<- matrix(sample(c(a,b)) , 100, 100)
exemplo2.t2<- matrix(sample(c(a,b)) , 100, 100)
exemplo2<- list(t1=exemplo2.t1, t2=exemplo2.t2)

mapas.mat(exemplo2, entorno=2, limite=0.5, cat.veg=c(1,2))


# Exemplo 3
n.floresta<-rep(c(3), 5000)
floresta<-rep(c(1,2), 5000)

n.floresta1<-rep(c(3), 6000)
floresta1<-rep(c(1,2), 4000)

n.floresta2<-rep(c(3), 8000)
floresta2<-rep(c(1,2), 2000)

n.floresta3<-rep(c(3), 4000)
floresta3<-rep(c(1,2), 6000)

exemplo3.t1<- matrix(sample(c(n.floresta,floresta)) , 100, 100)
exemplo3.t2<- matrix(sample(c(n.floresta1,floresta1)) , 100, 100)
exemplo3.t3<- matrix(sample(c(n.floresta2,floresta2)) , 100, 100)
exemplo3.t4<- matrix(sample(c(n.floresta3,floresta3)) , 100, 100)

exemplo3<- list(t1=exemplo3.t1, t2=exemplo3.t2, t3=exemplo3.t3,
t4=exemplo3.t4)

mapas.mat(exemplo3, entorno=1, limite=0.9, cat.veg=c(1,2))
```

Código

- Função mapas.mat
- Script dos exemplos do help