Thaís Nícia Azevedo

Mestrado: Efeito da expansão da cultura de cana-de-açúcar na estrutura da paisagem do estado de São Paulo. Orientador - Jean Paul Metzger

exec

Trabalho Final:

A idéia desta função veio de uma pergunta relacionada ao meu projeto de mestrado que analisará a conectividade funcional de paisagens, no meu caso, paisagens em que a matriz é a cana-de-açúcar.

Áreas com cultivo de cana-de-açúcar são realizadas em talhões com grandes extensões, levando à supressão de árvores isoladas e pequenos fragmentos de vegetação florestal.

Visto o apresentado, queremos responder se árvores isoladas em uma matriz são importantes para manter a conectividade funcional da paisagem. O estudo irá fornecer informações mais aprofundadas sobre a importância de árvores isoladas na matriz e sua influência na conectividade da paisagem.

Os dados de entrada no R serão previamente trabalhados em um software de informações geográficas, o qual sorteará 1.000 pequenos polígonos aleatórios que representarão as árvores de uma paisagem simulada. A partir desta paisagem, o software irá preparar uma tabela com todos os fragmentos da paisagem. As árvores serão consideradas fragmentos com uma pequena área e indicadas como inexistentes na tabela. Serão três colunas: código do fragmento / árvore; área do fragmento e existência (1 para fragmentos existentes e 0 para inexistentes - árvores). Ex:

Código do Fragmento - Área - Exist

```
1       13      0
2           13    0
3           13    0
4           13    0
5           13    0
6         2492    1
7        32084    1
8        12764    1
```

Note que os números de 1 a 5 representam as árvores simuladas pelo programa, com área muito baixa e não existentes na paisagem. A função a ser desenvolvida no R irá simular várias paisagens, sorteando diversas quantidades de árvores (dentre as 1.000) que passarão a ser consideradas como existentes, (mudarão na tabela de 0 para 1). Serão realizados diversos sorteios, para 10, 20, 30, 40, 50 e assim por diante até totalizar 1.000 árvores. Para cada sorteio, um data.frame será gerado. Os data.frames serão os objetos de saída.

Os objetos serão posteriormente analisados no software de Informações Geográficas, no qual dados de conectividade funcional serão gerados e plotados com a quantidade de árvores. Deste modo, a partir de certo número de árvores, podemos encontrar o ponto em que a retirada de uma árvore

afetará sensivelmente a conectividade da paisagem. Pode-se, então definir um número crítico de árvores necessárias para manter a conectividade funcional.

Plano B Função Sudoku

Comentário Leandro

Thais, você precisa explicar melhor qual o objetivo da função e como serão os dados de entrada e saída. Da forma como está não é possível saber se a função poderá ser feita no tempo disponível e qual o nível de complexidade dela.

Comentário Ale O leandro trabalha com vcs... imagina eu! Não consegui entender o que a função fará e nem tampouco o que significa a saída desse "software" de paisagem simulada!

Comentário Thaís

O comentário do Leandro foi em cima de um primeiro plano de trabalho que eu havia colocado anteriormente e que apaguei.

Mas como havíamos conversado em aula, as dúvidas da função foram esclarecidas. O que tornou o Plano A viável.

Explicando algumas dúvidas da função: ela irá simular diversas paisagens com o sorteio de diferentes quantidades de árvores, a partir de uma tabela vinda do ambiente SIG. Cada linha desta tabela representa um polígono da paisagem. Dentre estes polígonos têm-se fragmentos existentes na paisagem real e polígonos sorteados aleatoriamente em SIG que representam locais de uma possível árvore. O "R" irá simular a "entrada" de árvores sorteadas nos possíveis locais da paisagem.

A função proposta inicialmente foi realizada e algumas melhorias foram feitas. A função foi elaborada para que o argumento com o número de sorteios de árvores ("ntree", na função), pudesse ser um vetor. Assim, simulações com diferentes números de sorteios de árvores podem ser realizadas utilizando a função somente uma vez. Ou seja, pode-se calcular 1000 simulações (por exemplo) para o sorteio de diferentes números de árvores ao mesmo tempo, a partir de um vetor concatenado para o argumento: ntree=c(x,y,z...). No final, seguindo o exemplo, teremos 1000 simulações para "x", 1000 para "y", 1000 para "z" e assim sucessivamente.

Além do objeto de saída, que ficou melhor elaborado como sendo um objeto da classe "list" (ao invés de data.frame), uma tabela em formato "txt" é gerada para cada simulação. O formato "txt" é o que melhor se adapta para posteriores análises em um ambiente SIG, que também aceita linhas de comando. O diretório de saída destas tabelas em "txt" é um argumento que pode ser definido pelo usuário.

Página de ajuda

```
sample.tree             package:unknown                R Documentation


Description:
```

```
    A função simula a existência de árvores em uma paisagem, a fim de
calcular
a conectividade funcional em ambiente SIG. Tem como saída no “R” um objeto
da classe
“list” com todas as simulações realizadas e também arquivos em formato
“txt”, para
cada simulação.

Usage:

sample.tree (tab,nsim,ntree,dir)

Arguments:

tab: Tabela de dados produzida a partir de mapeamento de fragmentos, cruzada
com o
sorteio de pequenos polígonos aleatórios da paisagem, não sobrepostos aos
polígonos
de fragmentos em ambiente SIG. Para esta simulação, os pequenos polígonos
são tratados
como possíveis árvores da paisagem. A tabela deve ser composta por três
colunas:
Código do Fragmento – Área – Exist;

nsim: número de simulações a serem realizadas. Se este argumento for
omitido, serão
realizadas 10 simulações;

ntree: número de árvores que serão sorteadas. Indica quantas árvores serão
consideradas
existentes em cada sorteio. O argumento “ntree” pode ser um vetor, simulando
diferentes
quantidades de árvores a serem consideradas existentes em somente uma
execução da função.
Ao ser omitido, será atribuído o valor “1”;

dir: endereço de saída para os arquivos “txt” gerados pela função. Se
omitido, os arquivos
serão salvos no diretório da área de trabalho do “R”.

Details:
    As linhas da tabela de entrada representam os polígonos da paisagem,
tanto os
fragmentos quanto as árvores. Na primeira coluna, tem-se o código do
polígono e
na segunda, a área. A terceira coluna da tabela de entrada de dados é a
coluna
“Exist”, nas quais as linhas que irão simular a presença de árvores recebem
a
priori o valor “0”, visto que ainda serão sorteadas. Os fragmentos recebem o
valor “1”, o que indica sua existência na paisagem.
```

```
A função sorteará diversas quantidades de árvores, as quais serão
consideradas
existentes, mudando de “0” para “1” na tabela final. Os arquivos “txt”
gerados
poderão ser posteriormente analisados com o pacote “Conefor Sensinode” em
ambiente
SIG, a fim de calcular a conectividade funcional das árvores na paisagem.

Value:
    A função retorna um objeto da classe “list”, chamado “sampletree”.
    Os componentes da lista deste objeto são as diferentes simulações e têm
        como indexador o número da simulação.
    Além disso, a função gera um arquivo “txt” para cada simulação
realizada.
        Os arquivos são nomeados da seguinte maneira:
        “sample_tree_(número de árvores sorteadas)_sim_(número da
simulação)”
        Ex: “sample_tree_100_sim1”, no qual foram sorteadas 100 árvores na
primeira simulação.
Warning:
    O argumento “ntree” não pode ser maior que o número de linhas
disponíveis a
        serem sorteadas.
        A função não pode eleger uma amostra de possíveis árvores maior do
que os
        valores previstos com “Exist=0” na coluna. Não há reposição de
árvores
        durante a simulação da paisagem.


Author(s):

Thaís Nícia Azevedo
thaisnicia@yahoo.com.br

References:

http://eco.ib.usp.br/lepac/paisagem.htm

restore.strat
http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:05_curso_antigo:r2010:
alunos:trabalho_final:letambosi:start

http://www.conefor.org/


Examples:
cod=seq(1:30)
area=c(20,30,45,60,75,35,120,80,26,44,rep(13,20))
```

```
exist=c(rep(1,10),rep(0,20))
ex=data.frame(cod,area,exist)
sample.tree(ex,5,10)

cod=seq(1:30)
area=c(20,30,45,60,75,35,120,80,26,44,rep(13,20))
exist=c(rep(1,10),rep(0,20))
ex2=ex=data.frame(cod,area,exist)
ntree=(seq(1:10)) # ntree=vetor, para diferentes valores de árvores a serem
sorteadas,
#neste caso pode-se simular o sorteio de 1, 2, 3, assim sucessivamente até
10 árvores,
#conforme o vetor definido.
sample.tree(ex2,8,ntree)# Serão feitas oito simulações para cada número de
árvores
#a serem sorteadas.
```

Código da função

```
sample.tree<-function(tab,nsim=10,ntree=1,dir="")
    {
        list()->> sampletree # Objeto vazio, criado para o resultado final,
objeto de retorno
        tab[tab[,3]==1,]->u # Separa da tabela original as linhas com "1"
        tab[tab[,3]==0,]->z # Separa da tabela original as linhas com "0"
    for(j in 1:length(ntree)) ## O for irá gerar um ciclo para diversos
sorteios de árvores simuladas
    {
    for(i in 1:nsim)
        {
    tab[tab[,3]==0,]->z # A tabela de zeros, feita acima, é repetida para o
"z" ser renovado, assim não ocorrerá um sorteio em uma tabela já sorteada.
    z[(sample(seq(1:nrow(z)),ntree[j])),3]<-1 ## Sorteio de árvores da
primeira linha, até a última e substituição dos valores sorteados por "1".
    final <- rbind(u,z) ## Aqui as tabelas com "1" e "0" serão "coladas".
    write.table(final,
file=paste(dir,"sample_tree_",ntree[j],"_sim_",i,".txt",sep=""),sep=" ")#
Imprime o arquivo de saída "txt".
    final ->> sampletree[[i]]
        }
    }
    return(sampletree)
    }
```

Arquivo da função

sample_tree.r