

Kate Maia



Sou aluna de mestrado do programa de pós graduação em Ecologia da USP. Meu projeto se intitula “Insetos eusociais e a estrutura das redes de polinização”.

Exercícios

[ex1.r](#)

[ex2.r](#)

[ex3.r](#)

[ex4.r](#)

[ex5.r](#)

[ex6.r](#)

[ex7.r](#)

[ex9.r](#)

Projeto final

Plano A:

O aninhamento é um padrão comum em redes ecológicas mutualísticas. É encontrado quando espécies que apresentam poucas interações em uma rede interagem com um subconjunto das espécies com quem as espécies que apresentam muitas interações interage. Uma forma de contabilizar esse aninhamento é através da métrica NODF. Com essa métrica também é possível calcular qual a contribuição de cada espécie para o aninhamento da rede como um todo. As redes do meu projeto de mestrado são bipartidas, ou seja, apresentam dois subgrupos de espécies: animais e plantas. Farei uma função que calcula a contribuição de cada espécie para o aninhamento total da rede. A função também criará um dataframe por rede e que terá as seguintes colunas: espécie, a contribuição para o aninhamento previamente calculada, contribuição para o aninhamento padronizada pelas outras espécies do grupo (animais ou plantas) e a proporção de espécies do grupo (animais ou plantas) que contribui menos para o aninhamento da rede do que a espécie em questão.

Plano B:

O aninhamento é um padrão comum em redes ecológicas mutualísticas. É encontrado quando espécies que apresentam poucas interações em uma rede interagem com um subconjunto das espécies com quem as espécies que apresentam muitas interações interage. Uma forma de

contabilizar esse aninhamento é através da métrica NODF. Porém para calculá-lo a matriz de interações de estar completamente preenchida com “0” e “1” além ordenada, da espécie com mais interações para a com menos interações. As redes do meu projeto de mestrado são bipartidas, ou seja, apresentam dois subgrupos de espécies: animais e plantas, e são representadas por matrizes. Porém muitas vezes células dessas matrizes estão vazias, ou preenchidas com números diferentes de “0” e “1”. Farei uma função que preencha as células vazias com “0”, substitua os valores diferente de “1” por “1” e ordene as espécies de animais e plantas (linhas e colunas) com mais interações (células diferentes de “0”) para a com menos interações.

Comentários

Sob a premissa que vc já sabe como fazer os cálculos do plano A, implementá-los em uma função de R é uma proposta bem dimensionada. Se vc ainda não conhece todos estes cálculos fazer a função é uma ótima maneira de aprendê-los, mas cuidado com o tempo.

O plano B seria a conversão de matrizes quantitativas para binárias? Só isto é muito simples, basta algo do tipo

```
objeto[objeto>0] <- 1
```

Não entendi a parte das células vazias, pois este conceito não existe no R. Seriam do tipo NA? De qualquer jeito, isto também é bem simples de resolver, na importação dos dados.

Um desafio interessante das duas propostas é como definir o objeto de entrada que armazena uma rede bipartida. Uma matriz é a escolha mais simples, mas alguns outros arranjos são possíveis, inclusive para garantir uso por outros pacotes de análise de redes não-bipartidas.

Alterações na proposta

Ao longo do desenvolvimento da função, algumas de minhas idéias se mostraram desnecessárias. Por exemplo, não era necessário ordenar minhas matrizes pois o programa ANINHADO já faz isso. Dessa forma, uma nova necessidade surgiu, de implementar um modelo nulo para as minhas redes, que será usado imediatamente no meu projeto. Portanto concluí que o desenvolvimento dessa função seria mais construtivo, devido a sua maior complexidade, e produtivo, devido a sua utilidade imediata.

Código da Função

```
modelonulo.kate <- function(path)
{
  setwd(path) # trabalha no diretorio indicado pelo usuario
  unlink(dir(pattern="mod")) # limpa da pasta os arquivos gerados pela
funcao
  unlink(dir(pattern="random")) # limpa da pasta os arquivos gerados pela
funcao
```

```
files <- list.files(path) # lista os arquivos do diretório indicado
# Para cada arquivo presente no diretório indicado pelo usuário:
for(p in 1:length(files))
{
  x <- read.table(files[p], sep="\t", header=FALSE) # ler arquivo na
posicao p
  rotulo <- files[p] # Atribui os nomes dos arquivos originais ao objeto
rotulo
  x[is.na(x) == TRUE] <- 0 # atribui 0 aos possíveis NA
  x[x > 0] <- 1 # atribui 1 a tudo que for maior que 0
  rownames(x) <- c(1:nrow(x)) # numera as linhas para que a identidade das
especies não se perca
  colnames(x) <- c(1:ncol(x)) # numera as colunas para que a identidade
das especies não se perca
  write.table(x, file=paste(rotulo, "modificado", ".txt", sep="_"), sep=" ")
# imprime na pasta indicada, em txt, o arquivo original, já com as
modificações realizadas
  k.rows <- apply(x, 1, sum) # soma o número de interações das linhas
  k.columns <- apply(x, 2, sum) # soma o número de interações das colunas
  x.prob <- matrix(, nrow=nrow(x), ncol=ncol(x)) # cria matriz vazia
  # Cria matriz de probabilidades de interação com base na matriz x
(original)
  for(i in 1:nrow(x))
  {
    for(j in 1:ncol(x))
    {
      x.prob[i,j] <- (k.rows[i]/ncol(x) + k.columns[j]/nrow(x))/2
    }
  }
  # Criando 100 matrizes aleatórias com base na x.prob
  for(k in 1:100)
  {
    matriz <- matrix(, nrow=nrow(x), ncol=ncol(x)) # cria matriz vazia que
será preenchida de acordo com as probabilidades calculadas anteriormente
    rownames(matriz) <- c(1:nrow(x)) # numera linhas para que a identidade
das especies não se perca
    colnames(matriz) <- c(1:ncol(x)) # numera colunas para que a
identidade das especies não se perca
    # Preenche a matriz vazia de acordo com as probabilidades calculadas
    for(i in 1:nrow(x))
    {
      for(j in 1:ncol(x))
      {
        matriz[i,j] <- sample(x = c(0,1), size = 1, replace = TRUE, prob=
c(1 - x.prob[i,j], x.prob[i,j]))
      }
    }
    # Linhas ou colunas sem interação (apenas com 0) devem ser eliminados
    a <- apply(matriz, 1, sum) # k das linhas
    b <- apply(matriz, 2, sum) # k das colunas
    if (sum(a[] == 0) >= 1) # esse if elimina linhas que tenham apenas 0
```

```
{  
  matriz <- matriz[-which(a[] == 0),]  
}  
if (sum(b[] == 0) >= 1) # esse if elimina colunas que tenham apenas 0  
{  
  matriz <- as.matrix(matriz) # se, apos a remocao das linhas com 0,  
sobrar apenas uma linha o R transforma a matriz em vetor, que deve ser  
retransformado em matriz  
  if (ncol(matriz) == 1) # ao transformar um vetor em matriz, ele  
transpoe transformando a unica linha em unica coluna  
  {  
    matriz <- t(matriz) # retranspoe  
    matriz <- matriz[, -which(b[] == 0)] # finalmente elimina as  
colunas com 0  
  }  
  matriz <- matriz[, -which(b[] == 0)]  
}  
# Manda pra pasta indicada pelo usuario a matriz[k] que foi gerada com  
base na matriz de probabilidades  
write.table(matriz, file=paste(rotulo, "random", k, ".txt", sep="_"),  
sep=" ")  
}  
}
```

Help da Função

modelonulo.kate

package:nenhum

R Documentation

Cria matrizes semi-aleatórias com base em uma matriz de interações empírica.

Description:

Lê arquivos .txt de uma pasta indicada pelo usuário que contenham matrizes de interação entre espécies ou indivíduos. Transforma as matrizes em binárias, substituindo todos os valores por 0 ou 1. Cria 100 matrizes semi-aleatórias com base na matriz original e em um modelo nulo, que considera a probabilidade de interação em cada célula. Essa probabilidade é calculada com base no número de interações presentes naquela linha e coluna da matriz original. A matriz original transformada em binária e as 100 matrizes geradas pela função, no formato .txt, são colocadas no mesmo diretório que foi indicado como sendo o diretório das entradas da função.

Usage:

```
modelonulo.kate(path)
```

Arguments:

path: Caminho de um diretório. De onde serão retirados os arquivos .txt originais para entrada da função, e onde serão depositados os novos arquivos .txt gerados.

Details:

As matrizes originais são lidas pela função, possíveis células não preenchidas recebem o valor zero, e todas as outras células que tenham valores maiores do que 1, recebem o valor 1. Uma matriz de probabilidade de interação é gerada com base na matriz original. Para cada célula (i,j) uma probabilidade é calculada da seguinte forma: $P_{ij} = ((k_i/C) + (k_j/L))/2$, onde k_i é o número de presenças de interação (1) na linha i, k_j é o número de presenças de interação (1) na coluna j, C é o número total de colunas e L é o número total de linhas. A probabilidade em cada célula é a probabilidade de, na matriz semi-aleatória, aquela célula ser preenchida por 1, e não por 0. Na criação de cada uma das 100 matrizes aleatórias, após todas as células terem sido preenchidas com 0 ou 1, as linhas ou colunas que apresentarem apenas 0, ou seja não possuírem nenhuma interação, serão eliminadas. Porém, é possível identificar qual linha ou coluna foi eliminada em decorrência da numeração de rótulo atribuída às linhas e colunas. Depois que as matrizes são criadas elas recebem um nome com um número de 1 a 100 e são salvas no diretório indicado pelo usuário.

Value:

Para cada matriz em .txt presente na pasta, 101 novas matrizes em .txt são geradas e colocadas nessa mesma pasta. Uma dessas matrizes é a matriz original transformada em binária, completamente preenchida com 0 e 1. As outras 100 matrizes são as semi-aleatórias geradas pelo modelo nulo.

Warning:

O diretório indicado no argumento deve conter inicialmente (antes da utilização da função) apenas os arquivos em .txt correspondentes às matrizes que devem ser modificadas e usadas para gerar matrizes semi-aleatórias.

Author(s):

Kate Maia

kate.spirogyra@gmail.com

References:

Bascompte, J., Jordano, P., Melián, C.J., Olesen, J.M. 2003. The nested assembly of plant-animal mutualistic networks. *Proceedings of the National Academy of Sciences*, 100:9383-9387.

Examples:

```
modelonulo.kate(path="Users/Kate/Documents/Disciplinar/redes")
```

From:

<http://labtrop.ib.usp.br/> - **Laboratório de Ecologia de Florestas Tropicais**

Permanent link:

http://labtrop.ib.usp.br/doku.php?id=cursos:ecor:05_curso_antigo:alunos2012:alunos:trabalho_final:kate.maia:start 

Last update: **2020/07/27 18:46**