

[trabalhofinal1.txt](#)[trabalhofinal2.txt](#)

Comentários: A proposta A está MUITO legal, mas parece bastante trabalhosa. Veja se você consegue dividir a tarefa em funções menores: uma função que gera a matriz A, uma função que verifica se uma dada matriz tem uma estrutura estável, etc; e a sua função final vai simplesmente chamando essas funções auxiliares.

A proposta B precisaria ser um pouco mais detalhada. O que é um IVI, por exemplo? —[André Chalom](#)

```
str.stable <- function (matriz, vetor,nsim)#insira a tabela com o número de
espécies próximas/focal (matriz); o vetor de composição inicial das espécies
focais (vetor); e o número de vezes que a matriz deve ser multiplicada pelos
vetores, o qual deve ser suficiente para estabilizar o vetor de composição
estável (nsim)
{
  vetor<-as.matrix(vetor, ncol=1)#transformando o vetor em vetor coluna,
para que a multiplicação de matrizes seja feita corretamete (linha por
coluna).
  if ( ncol(matriz) != nrow(matriz))#restringindo os cálculos para dados
organizados em matrizes quadradas.
  {
    warning("Matriz deve ser quadrada: nº de colunas = nº de linhas. Toda
espécie 'focal' deve ser também 'próxima'")# O modelo prevê que os dados
estejam distribuídos assim, para que a diagonal principal seja sempre a
probabilidade da espécie focal ser substituída por ela mesma.
  }

  if ( ncol(matriz) != nrow(vetor))# restringindo os cálculos para haja
informação do número de espécies atual para toda espécie focal.
  {
    warning("Nº de elementos do vetor de composição atual deve ser igual ao nº
de colunas da matriz : deve haver informação do número de espécies atuais
para todas as espécies 'focais'.")#
  }

  sumcol<-apply(X=matriz, MARGIN=2, FUN=sum )#somando os valores de cada
coluna da matriz e atribuindo um nome ao vetor de valores resultante.
  dadosmat=matriz#criando um data frame para guardar os dados de abundância
relativa das espécies que serão calculados a partir da divisão de cada
célula pela soma de sua respectiva coluna.
  dadosmat[]=NA#substituindo os valores do data frame base para NA.
  for(i in 1: length(sumcol))# para todos os resultados da somatória das
colunas da 'matriz'
  {
    for(j in 1:length(matriz [,1]))# e para todos os valores da 'matriz':
```

```
{
  dadosmat[j,i]<-matriz[j,i]/sumcol[i]#divida cada valor da coluna 'i'
(matriz) pelo total da coluna 'i'(sumcol) e assim sucessivamente(abundâncias
relativas de espécies 'próximas' de cada 'focal').
}
}
dadosmat<-as.matrix(dadosmat)#transformando os valores de abundância
relativa obtidos para a classe "matrix"

mat.stable<- matrix(NA, ncol=nsim, nrow=length(vetor))# criando uma matriz
para guardar os resultados obtidos da multiplicação da 'matriz' pelos
diferentes vetores. Sendo que o número de multiplicações será informado pela
usuária ou usuário (argumento nsim).

mat.stable[,1]<-dadosmat%%vetor#multiplicar a matriz de probabilidades pelo
vetor de composição atual e guardar o resultado na primeira coluna do objeto
'mat.stable'.
for(i in 1:(nsim-1))# criando um loop que irá guardar os vetores
resultantes das multiplicações de matrizes pelo vetor. Como a primeira
multiplicação foi feita fora do loop, agora restam 'nsim'-1 multiplicações
para serem feitas.
{
  mat.stable[, (i+1)]<-round(dadosmat%%mat.stable[,i],digits=6)#guardar o
resultado da multiplicação da matriz pelo vetor guardado na posição 'i' do
objeto 'mat.stable', na posição 'i+1. Restringindo o número de dígitos para
seis casas depois da vírgula.
  if (i>=6)# quando a matriz 'mat.stable' for preenchida por pelo menos
seis valores de vetores de composição:
  {
    if(mat.stable[,i]==mat.stable[, (i-1)]&&
mat.stable[,i]==mat.stable[, (i-2)]&&mat.stable[,i]==mat.stable[, (i-3)]&&mat.
stable[,i]==mat.stable[, (i-4)]&&mat.stable[,i]==mat.stable[, (i-5)])#
verifique se os resultados da multiplicação da matriz 'dadosmat' pelos
vetores estabilizaram, ou seja, os valores dos vetores não se alteram mais.
Para diminuir a chance da igualdade de vetores não ter sido ao acaso,
verifique se o resultado (vetor) se mantém igual por cinco ciclos
consecutivos de multiplicações.
    {break} #quando os valores dos vetores estabilizarem, pare as
multiplicações.

  }
}
mat.stable <- mat.stable[,colSums(is.na(mat.stable))<nrow(mat.stable)]#
retire as colunas onde so existem NAs. Situações que ocorrerão sempre e,
caso não ocorram (número de multiplicações não tenham sido suficientes para
estabilizar o vetor de composição), argumento nsim deve ser aumentado.

if(mat.stable[,length (mat.stable [1,])]!=mat.stable[, (length (mat.stable
[1,])-1)]&& mat.stable[,length (mat.stable [1,])]!=mat.stable[, (length
(mat.stable [1,])-2)]&&mat.stable[,length (mat.stable
```

```
[1,)]!=mat.stable[, (length (mat.stable [1,])-3)]&&mat.stable[,length
(mat.stable [1,)]!=mat.stable[, (length (mat.stable
[1,])-4)]&&mat.stable[,length (mat.stable [1,)]!=mat.stable[, (length
(mat.stable [1,])-5)])# caso os vetores coluna nao se estabilizem,
{
  warning("0 número de multiplicações 'nsim' não foi suficiente para
estabilizar os valores do vetor coluna")# avisar que o número de
multiplicações não foi suficiente para estabilizar o vetor de composição.
Porém a função irá retornar o valor obtido, de qualquer forma, pois @
usuári@ pode querer saber a composição em determinado 'nsim' mesmo que nao
seja estável.
}

nsim.stable<-(length (mat.stable [1,]))#verificando quantas linhas tem a
matriz 'mat.stable' após a retirada das colunas com NA e atribuindo um nome
a este valor.
last.col<-mat.stable[,nsim.stable]#chamando a última coluna da matriz
'mat.stable' (vetor de composição estável) e atribuindo-lhe um nome.
last.col<-as.matrix(last.col,ncol=1)# colocando 'last.col' na vertical para
melhor visualização da composição estável da comunidade.
colnames(last.col)<-"Stable.Composition"#atribuindo um nome ao vetor de
composição estável.
rownames(last.col)<-rownames(matriz)# atribuindo os nomes correspondentes as
espécies do vetor de composição estável.
stable<-list(Nsim_for_stabilize=nsim.stable, Stable.Composition=last.col)#
juntando os valores que desejo que a função mostre.
return(stable)# retornando o número de multiplicações realizadas para
atingir o vetor coluna estável e os valores absolutos das espécies previstos
em uma composição estável da comunidade a longo prazo.
```

```
str.stable {base}
```

R Documentation

Function to estimate the stable structure of a community, that is, to check what are the relative abundances of each species on the stability (or climax)

Description:

From a table that contains:

1. the species of interest (focal) existents in an area;
2. the sampling of species that occur "nearby" to focal species;

And a vector that contains:

3. the current numbers of the focal species in the community, the function calculates the probability of an individual of a focal species be replaced

by another individual of the same or other species.

Usage:

```
str.stable(matriz = NA, vetor = NA, nsim=100)
```

Arguments:

matriz a squared table where the columns labels are the focal species and the row labels are the nearby species. Every focal species must be too a

nearby species, and vice versa.

vetor a vector with the current numbers of the focal species in the community.

nsim numbers of times that the matrix should be multiplied by the vectors.

Details:

'Nsim' must be sufficient to the column vector achieve the stability, whereas the parameters to determine 'nsim' will depend of the organism and the situation analized. For trees communities, for example, a suitable 'nsim' value is obtained through the sum of the averages of survival of each tree species

(Horn, 1975).

Value:

Gives a new column vector with the stable composition (absolute numbers) expected to the community in long term, if the environmental conditions

and the relationships between species do not change. In the group '_for_Stabilize', gives the number of multiplications necessary to stabilize the column vector.

Note:

The list of "nearby" species will depend on the characteristics of the organism and of the environment analised. As a rule, the "nearby" species must be

species with potential capacity of replace the "focal" if the last one release the place where it lives.

Species with the lowest representation can be treated as "other".

Author:

Aline Luisa Mansur, graduate student. University of Campinas (UNICAMP)

References:

Horn, H.S. (1975). Markovian properties of forest successions. In Cody, M.L. & Diamond J.M.(Eds.), Ecology and Evolution of Communities. Belknap Press.

Cambridge, MA, 196–211.

Enright, N. & Ogden, J. (1979). Applications of transition matrix models in forests dynamics: Araucaria in Papua New Guinea and Nothofagus in New

Zealand. Australian Journal of Ecology 4,3-23.

Usher, M.B. & Parr, T.W. (1977). Are there sucessional changes in arthropodes decomposer communities? J. Env. Manag. 5,151-160.

Examples:

```
matriz <- matrix(1:28, nrow=7,ncol=7)#creating a matrix with the same  
number of row and columns  
vetor<- (1:7)#creating a vector with the current number of focal species  
str.stable(matriz, vetor, nsim=20)
```

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:aline.mansur:trabalho_final



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