



# Análise de classificação - Roteiro em R



Os métodos de classificação agrupam objetos conforme a similaridade entre eles. Se tomamos amostras de comunidades em que registramos a presença ou abundância de cada espécie, isso pode ser útil para verificarmos se as amostras formam conjuntos discretos. Para tanto, precisamos primeiro calcular um índice que nos diga o quanto cada amostra é similar às outras em termos de composição de espécies. A primeira parte dessa atividade explica como calcular medidas de similaridade e distância entre parcelas amostrais. Em seguida usamos um método de agrupamento dessas parcelas para finalmente apresentar os dados graficamente em um dendrograma.

## Similaridade & Distância

Existe uma infinidade de índices utilizados em ecologia para medir a similaridade ou dissimilaridade (distância) entre objetos. O primeiro passo para classificar comunidades é usar esses índices para expressar a diferença entre pares de amostras das comunidades. Em ecologia, essas amostras, em geral, podem ter dois tipos de dados: i) quais espécies estão presentes (dados de presença/ausência, ou binários), ou; ii) qual a abundância (que pode ser medida de diversas formas) de cada espécie presente (dados quantitativos).



A seguir apresentamos alguns dos índices mais simples para cálculo de similaridade para dados de presença/ausência e de abundância.

## Similaridade por presença e ausência

### Jaccard

O índice de Jaccard indica a proporção de espécies compartilhadas entre duas amostras em relação ao total de espécies. Uma forma de calculá-lo é:

$$J = \frac{S_{\text{com}}}{s_1 + s_2 - S_{\text{com}}}$$

O que é o mesmo que:

$$J = \frac{S_{\text{com}}}{S}$$

onde:

- $S_{\text{com}}$  é o número de espécies em comum nas duas amostras
- $s_1$  e  $s_2$  é o número total de espécies em cada uma das amostras
- $S$  é o total de espécies no conjunto de amostras

Vamos supor que foi feita uma amostra de duas parcelas de vegetação. O número de indivíduos das espécies de plantas encontradas em cada uma dessas parcelas fictícias foi:

	plot_1	plot_2
sp_1	2	8
sp_2	13	6
sp_3	0	5
sp_4	1	15
sp_5	20	0

Comparando a presença/ausência de espécies, você acha que as parcelas são muito parecidas ou muito diferentes? E comparando a abundância das espécies?

Vamos usar esses valores para o cálculo do índice de Jaccard no R. Para isso, copie os comandos abaixo e cole-os na linha de comando do R (obs.: as frases que iniciam com o símbolo “##” são apenas textos explicativos):

```
## Duas parcelas fictícias
(plot1=c(2,13,0,1,20))
(plot2=c(8,6,5,15,0))
## Criando o objeto //parcela// com as duas parcelas:
(parcelas=data.frame(plot1,plot2))
## Calculando o número de espécies em cada parcela
(nsp_1=sum(plot1>0))
(nsp_2=sum(plot2>0))
##Calculando o total de espécies em comum
(nsp_com=sum(plot1>0 & plot2>0))
## Índice de Jaccard
jacc= nsp_com/(nsp_1 + nsp_2 - nsp_com)
jacc
```

Qual a proporção de similaridade observada entre as duas parcelas fictícias? Anote essa informação.

## Similaridade por abundância

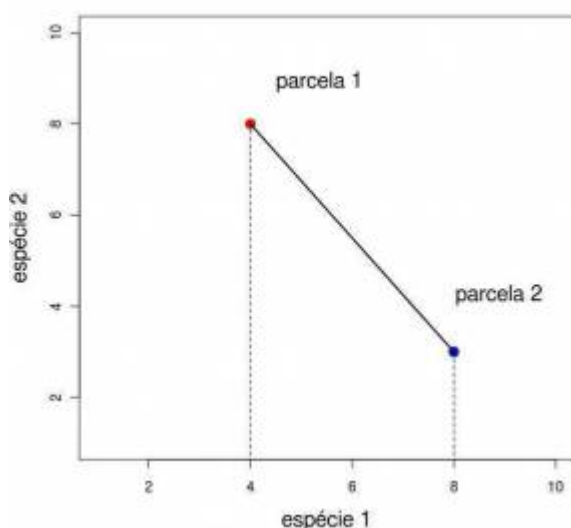


Quando queremos comparar amostras não só pelas presenças, mas também por suas abundâncias precisamos usar um índice quantitativo. Uma das maneiras de pensar nesses índices é pela medida de distância, ou seja pelo quanto uma parcela se diferencia da outra, medida pela diferença entre as abundâncias das espécies. A seguir dois dos índices quantitativos mais usados em ecologia de comunidades.

## Distância Euclidiana

O inverso ou complementar da similaridade são as medidas de distância. No exemplo anterior poderíamos dizer que a similaridade é de 60% entre as parcelas ou que a dissimilaridade (ou distância) entre elas é de 40%!

Uma das medidas de distância muito comum é a distância euclidiana, baseada na medida em um sistema de coordenadas cartesianas. Podemos usá-la para expressar a distância entre duas amostras de vegetação ao registramos as abundâncias de cada espécie. Se registramos apenas duas espécies, suas abundâncias podem ser representadas em um plano cartesiano, em que no eixo X temos a abundância de uma das espécies e no eixo Y a abundância da outra espécie. Os pontos no gráfico representam duas amostras/parcelas (Obs.: não são as nossas parcelas fictícias criadas acima).



Nesse caso a distância euclidiana pode ser descrita como a distância entre duas parcelas medidas por

unidades de indivíduos de duas espécies. Para calcular a distância utilizamos o teorema de Pitágoras e calculamos a hipotenusa, como representado na figura. Usamos a equação abaixo:

$$d_E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Onde  $x_1$ ,  $x_2$  são as abundâncias de uma espécie nas parcelas 1 e 2, e  $y_1$ ,  $y_2$  são as abundâncias da outra espécie nas mesmas parcelas.

Para um maior número de espécies fica difícil fazer uma representação gráfica, mas a lógica é a mesma e a equação é generalizada assim:

$$d_E = \sqrt{\sum_{i=1}^S (n_{i1} - n_{i2})^2}$$

Onde  $n_{i1}$  e  $n_{i2}$  são as abundâncias da  $i$ -ésima espécie na primeira e segunda parcela, e  $S$  é o total de espécies, como já definido.

Vamos calcular então a distância euclidiana entre as nossas duas parcelas fictícias do início dessa atividade:

```
eucl=sqrt(sum((plot1-plot2)^2))
eucl
```

Nesse caso, a distância entre as parcelas é de 26,6 indivíduos. E o que isso representa? A distância entre essas duas parcelas é pequena ou grande? Difícil responder isso com apenas um valor, não é?

Apesar da distância euclidiana ser muito fácil de ser entendida e muito útil para análises de classificação, ela não varia em um intervalo de 0 a 1 e, então, o valor obtido não é comparável aos valores obtidos por outras medidas de similaridade que variam entre 0 e 1. Então, para podermos comparar a medida de similaridade obtida para dados de presença/ausência (pelo índice de Jaccard), com uma medida de similaridade obtida para dados de abundância vamos usar uma outra medida chamada de “Bray-Curtis” em homenagem aos seus autores.

## Bray-Curtis

O índice de Bray-Curtis pode ser expresso como uma proporção de similaridade ou dissimilaridade (distância) na abundância das espécies. Em qualquer um dos casos seus valores vão de um máximo de um ao mínimo de zero. Essa padronização no intervalo entre um e zero facilita a interpretação e comparação.

## Similaridade

A similaridade de Bray-Curtis é:

$$\frac{2 \sum_{i=1}^S \min(n_{i1}, n_{i2})}{N}$$

Onde  $N$  é a soma de indivíduos de todas as espécies e parcelas, e  $\min(n_{i1}, n_{i2})$  é a menor

das duas abundâncias da espécie  $i$ , entre as duas parcelas. Como já definido,  $n_{i1}$  e  $n_{i2}$  são as abundâncias da  $i$ -ésima espécie na primeira e segunda parcela,  $S$  é o total de espécies.

### Distância

A distância de Bray Curtis é:

$$\frac{\sum_{i=1}^S |n_{i1} - n_{i2}|}{N}$$

Onde  $|n_{i1} - n_{i2}|$  é o **valor absoluto** da diferença das abundâncias da espécie  $i$  nas duas parcelas.

### Cálculo

Vamos calcular então a similaridade de Bray-Curtis entre as duas parcelas:

```
bc.sim=2*sum(apply(parcelas, 1, min))/(sum (parcelas))
bc.sim
```

Qual valor você observou? Ele é parecido com o valor do índice de Jaccard? Como você explicaria isso?

### Matriz de Similaridade

Para seguir em frente é necessário que você tenha na área de trabalho do R as amostras das comunidades virtuais montadas, como descrito no roteiro [padrões de gradientes em comunidades](#). Certifique-se disso com o comando `ls()`, que lista os objetos da área de trabalho do R. Na lista resultante devem estar os objetos `amost.cont` e `amost.disc`:

```
> ls()
[1] "amost"      "amost.cont" "amost.disc"
```

Agora vamos aplicar o índice de Bray-Curtis para o conjunto de parcelas [tomadas ao longo de um gradiente simulado](#). Para tanto temos que fazer uma função que calcule a similaridade entre cada par de parcelas. Ei-la:

```
sim<-function(dados, indice="bc")
{
  nplot=dim(dados)[2]
  similar=matrix(1,ncol=nplot,nrow=nplot)
  rownames(similar)<-paste("plot", c(1:nplot))
  colnames(similar)<-paste("plot", c(1:nplot))
```

```
for(i in 1:(nplot-1))
{
m=i+1
for(m in m:nplot)
{
if(indice=="jacc")
{
dados[dados>0]=1
co.oc=sum(dados[,i]>0 & dados[,m]>0)
total.sp=sum(dados[,i])+sum(dados[,m])-co.oc
similar[i,m]=co.oc/total.sp
similar[m,i]=co.oc/total.sp
}
if(indice=="bc")
{
bc.sim=2*sum(apply(dados[,c(i,m)], 1, min))/(sum
(dados[,c(i,m)]))
similar[i,m]=bc.sim
similar[m,i]=bc.sim
}
}
}
return(round(similar,3))
}
```

Agora vamos aplicar essa função para a nossa comunidade virtual contínua:

```
sim.cont1=sim(amost.cont, indice="bc")
```

Verifique a matriz resultante

```
sim.cont1
```

Qual o elemento da linha 1 coluna 4 desta matriz?

```
sim.cont1[1,4]
```

E da linha 4 e coluna 1?

```
sim.cont1[4,1]
```

Verifique o mesmo para linha 5 e 7 e coluna 5 e 7

```
sim.cont1[5,7]
sim.cont1[7,5]
```

O que significam os valores **1,00** na matriz resultante?

Por que os valores na posição [linha1,coluna4] são iguais aos da posição [linha4,coluna1]?

Consegue identificar padrões nos dados mostrados na matriz?

Faça o mesmo para amostras da comunidade discreta (objeto `amost.disc`), mas guarde o resultado em um objeto de nome `sim.disc1`. Chegamos nas matrizes de similaridades, o primeiro passo da nossa classificação.

## Agrupamento



Há vários métodos e algoritmos<sup>1)</sup> para agrupar os objetos em uma análise de classificação. Primeiro precisamos decidir se queremos iniciar juntando os elementos (análise aglomerativa) ou se queremos tomar todo o conjunto de objetos e ir separando em grupos (análise divisiva). No nosso caso, vamos iniciar com as parcelas como unidades e vamos montando grupos sucessivamente a partir daqueles já formados. Isso se chama análise de classificação (ou agrupamento) hierárquica aglomerativa. Apesar do nome feio, não há nada de complicado na lógica da análise. Vamos acompanhar passo a passo.

Qual o máximo de similaridade nesta matriz?

```
max(sim.cont1, na.rm=TRUE)
```

Vamos perguntar ao R quais valores da matriz são iguais a este máximo

```
sim.cont1==max(sim.cont1, na.rm=TRUE)
```

Na matriz produzida agora, o “TRUE” indica as posições em que os valores máximos são observados. Notem que todos os “TRUE” estão na diagonal principal da matriz produzida. Essas similaridades não nos interessam, pois na nossa matriz esses valores correspondem à similaridade de uma parcela com ela mesma. Não há porque agrupar algo com ele mesmo. Além disso, os valores se repetem acima e abaixo da diagonal<sup>2)</sup> Portanto, vamos retirar da nossa matriz de similaridade a diagonal, assim como as redundâncias (o triângulo superior da matriz).

```
nome.par=rownames(sim.cont1)
upper.tri(sim.cont1, diag=TRUE)
sim.cont1[upper.tri(sim.cont1, diag=TRUE)]=NA
sim.cont1
```

A função “upper.tri” seleciona todos os valores que estão acima da diagonal principal na matriz. E, em resumo, o que estamos fazendo aqui é criando uma nova matriz (mas vamos manter o mesmo nome “sim.cont1”) em que os valores da diagonal principal e os valores no triângulo acima da diagonal principal serão desconsiderados nas futuras buscas<sup>3)</sup>.

## Primeira Ligação

Agora que desconsideramos a diagonal, vamos procurar o maior valor de similaridade



```
max1=max(sim.cont1,na.rm=TRUE)  
max1
```

e então perguntamos ao R qual o par de parcelas que apresenta essa similaridade:

```
maior1=which(sim.cont1==max(sim.cont1,na.rm=TRUE), arr.ind=TRUE)  
maior1
```

Vamos tomar apenas o primeiro valor maior. Para garantir isso, caso haja outros, vamos fazer assim.

```
par1=nome.par[maior1[1,]]  
par1  
cat("\n\t 1a. ligação ", paste(par1[1],par1[2], sep=" x "), "; ligação = ",  
max1, "\n" )
```

Esse é nosso primeiro grupo: Agora temos que tomar uma outra decisão: como esse grupo formado agora irá se ligar às outras parcelas? Existem diferentes “Métodos de Ligação”. Podemos decidir que ele irá se ligar às outras parcelas pelo valor da máxima similaridade de seus componentes (ligação máxima), pela mínima, pela média, pelo centroide do grupo, etc. No nosso caso, vamos usar o método de ligação pela a média do grupo (sigla em inglês UPGMA).

No UPGMA, para definir a próxima parcela que será ligada usamos a média aritmética da similaridade entre a parcela que se quer incluir em um grupo a cada parcela já existente nesse grupo. A parcela é então atribuída ao grupo com o qual ela tem maior similaridade média com todas as parcelas (Valentin 2012).

Para fazer isso, criamos uma nova matriz de distâncias igual à anterior, com uma diferença: as duas parcelas agrupadas foram substituídas pelo grupo.

Essa nova matriz tem então as similaridades entre todas as parcelas que não foram agrupadas, e entre todas e o grupo criado. Como estamos usando o método UPGMA, a similaridade entre o grupo e as demais parcelas é a média das similaridades das parcelas do grupo com as demais parcelas.

O código abaixo faz isso passo a passo. Não se preocupe com os comandos de cada passo, mas tente entender o resultado final, que é exibido após o último comando:

```
## Preambulo: um rotulo para o grupo na matriz  
g1.n1=strsplit(nome.par[maior1[1,2]]," ")[[1]][2]  
g1.n2=strsplit(nome.par[maior1[1,1]]," ")[[1]][2]  
g1.nome=paste("g", paste(g1.n1,g1.n2, sep=","))  
g1.nome # este objeto apenas tem um novo rotulo para a nova matriz
```



```
## Nova matriz
mat.g1=sim.cont1[-maior1[1,],[-maior1[1,]]
g1a=apply(sim.cont1[maior1[1,],[-maior1[1,]],2,mean)
g1a[is.na(g1a)]=0
g1b=apply(sim.cont1[-maior1[1,],maior1[1,]],1,mean)
g1b[is.na(g1b)]=0
gr1=rbind(mat.g1,g1a+g1b)
grupo1=cbind(gr1,NA)
rownames(grupo1)<-c(nome.par[-maior1[1,]],g1.nome)
colnames(grupo1)[dim(grupo1)[2]]<-g1.nome
grupo1 # nova matriz com o primeiro grupo formado
```

## Segunda Ligação

Agora repetimos os mesmos passos. Primeiro procuramos o par de elementos mais próximos. Note que esses elementos podem ser:

- Duas outras parcelas diferentes do par já agrupado. Nesse caso, um novo grupo é criado.
- Uma parcela e o grupo já existente. Nesse caso a parcela será agregada ao grupo já existente.

Execute o código abaixo para descobrir o que ocorre no seu conjunto de dados

```
nome.par2=rownames(grupo1)
max2=max(grupo1,na.rm=TRUE)
max2
maior2=which(grupo1==max(grupo1,na.rm=TRUE), arr.ind=TRUE)
maior2
g2.n1=strsplit(rownames(grupo1)[maior2[1,2]]," ")[[1]][2]
g2.n2=strsplit(rownames(grupo1)[maior2[1,1]]," ")[[1]][2]
g2.nome=paste(paste("g",g2.n1,sep="_"),g2.n2, sep=",")
g2.nome
cat("\n\n\t 2a. ligação ",
paste(nome.par2[maior2[1,2]],nome.par2[maior2[1,1]], sep=" x "), "; ligação
= ", max2, "\n" )
```

Em criamos a nova matriz de similaridade:

- Se um novo grupo foi criado, repetimos a primeira ligação: as duas parcelas que o compõem serão retiradas da matriz de similaridade, e substituída pelo grupo. As similaridades são calculadas entre cada parcela e o novo grupo. Para isso, calcula-se a similaridade de cada parcela do grupo a cada parcela que não é do grupo. Em seguida calcula-se a média dessas similaridades, que então é usada como similaridade do grupo às parcelas.
- Se uma parcela é adicionada ao grupo já existente, a similaridade do grupo a cada parcela é recalculada. Isso é feito calculando-se a média da similaridade das parcelas do grupo (que agora são três) a cada parcela. Em seguida calcula-se a média dessas similaridades, que então é usada como similaridade do grupo às parcelas.

O código abaixo faz tudo isso. A última linha mostra a matriz resultante

```
mat.g2=grupo1[-maior2[1,],[-maior2[1,]]
```

```
g2a=apply(grupo1[maior2[1,], -maior2[1,]], 2, mean)
g2a[is.na(g2a)]=0
g2b=apply(grupo1[-maior2[1,], maior2[1,]], 1, mean)
g2b[is.na(g2b)]=0
gr2=rbind(mat.g2, g2a+g2b)
grupo2=cbind(gr2, NA)
rownames(grupo2)<-c(nome.par2[-maior2[1,]], g2.nome)
colnames(grupo2)[dim(grupo2)[2]]<-g2.nome
grupo2
```

Compare a matriz de similaridade após a segunda ligação com a obtida com a primeira ligação, digitando:

```
grupo1
```

## Terceira Ligação

Agora mais uma vez buscamos a maior ligação entre todas na nova matriz.

```
nome.par3=rownames(grupo2)
max3=max(grupo2, na.rm=TRUE)
max3
maior3=which(grupo2==max(grupo2, na.rm=TRUE), arr.ind=TRUE)
maior3
g3.n1=strsplit(rownames(grupo2)[maior3[1,2]], " ")[[1]][2]
g3.n2=strsplit(rownames(grupo2)[maior3[1,1]], " ")[[1]][2]
g3.nome=paste(paste("g", g3.n1, sep="_"), g3.n2, sep=",")
g3.nome
cat("\n\n\t 3a. ligação ",
paste(nome.par3[maior3[1,2]], nome.par3[maior3[1,1]], sep=" x "), "; ligação
= ", max3, "\n" )
```

E formamos um novo grupo.

```
mat.g3=grupo2[-maior3[1,], -maior3[1,]]
g3a=apply(grupo2[maior3[1,], -maior3[1,]], 2, mean)
g3a[is.na(g3a)]=0
g3b=apply(grupo2[-maior3[1,], maior3[1,]], 1, mean)
g3b[is.na(g3b)]=0
gr3=rbind(mat.g3, g3a+g3b)
grupo3=cbind(gr3, NA)
rownames(grupo3)<-c(nome.par3[-maior3[1,]], g3.nome)
colnames(grupo3)[dim(grupo3)[2]]<-g3.nome
grupo3
```

Se continuarmos a fazer a mesma operação teremos um único grupo no final, no qual existe a última ligação entre os grupos formados anteriormente. Engenhoso e didático, mas bastante tedioso!

Por sorte temos uma função no pacote básico do R que dá conta do recado para nós. Agora que

entendemos como é feito, podemos usar a função. Caso tenha ainda alguma dúvida sobre o processo de classificação hierárquica aglomerativa, refaça os passos anteriores ou peça socorro! Caso tenha entendido já pode perguntar numa rodinha na Vila Madalena: - "Vocês conhecem o algoritmo hierárquico aglomerativo?". Seu sucesso reprodutivo vai ficar em baixa, mas vale a pena para ver a cara dos(as) colequinhas...

## Todas as ligações de uma vez!

Vamos agora rodar a análise de classificação completa usando a nossa função de similaridade e a função básica do R `hclust()`. O argumento `method` permite usar outros métodos de ligação. Para saber mais cada método veja a seção *para saber mais*. Experimente a função com o método de ligação média:

```
clas.cont1=hclust(as.dist(1-sim.cont1), method="average")
```

Uma boa representação gráfica da análise de agrupamentos é o *dendrograma*. Crie este gráfico para a análise que você acabou de fazer com os comandos:

```
dend.cont1=as.dendrogram(clas.cont1, hang=-1)  
plot(dend.cont1)
```

Um dendrograma representa uma classificação hierárquica como a que fizemos. Ou seja, representa graficamente uma sequência de ligações de elementos a outros elementos. O resultado é uma hierarquia porque são grupos dentro de grupos dentro de grupos... Para interpretar um dendrograma, tome uma das parcelas, que são os ramos terminais do dendrograma. Siga até encontrar a primeira ligação. Este é o primeiro elemento que foi ligado à parcela. Esse "primeiro vizinho" pode ser outra parcela ou um grupo de parcelas. Na escala ao lado do dendrograma veja o valor no ponto da ligação. Essa é a distância (dissimilaridade) entre os dois elementos. Você pode fazer isso com quaisquer elementos do dendrograma.

Agora você pode criar e comparar os dendrogramas obtidos com análises de agrupamento com os quatro diferentes métodos de ligação:

```
par(mfrow=c(2,2))  
clas.cont1a=hclust(as.dist(1-sim.cont1), method="single")  
plot(as.dendrogram(clas.cont1a, hang=-1), ylab="Bray-Curtis", main="Ligação simples")  
clas.cont1b=hclust(as.dist(1-sim.cont1), method="complete")  
plot(as.dendrogram(clas.cont1b, hang=-1), ylab="Bray-Curtis", main="Ligação completa")  
clas.cont1c=hclust(as.dist(1-sim.cont1), method="average")
```

```
plot(as.dendrogram(clas.cont1c, hang=-1), ylab="Bray-Curtis", main="Ligação  
média")  
clas.cont1d=hclust(as.dist(1-sim.cont1), method="centroid")  
plot(as.dendrogram(clas.cont1d, hang=-1), ylab="Bray-Curtis", main="Ligação  
centroide", ylim=c(0,0.7))
```

Entendendo o que foi feito com código acima:

1. Para fazer vários gráficos em uma mesma janela foi alterado o parâmetro *mfrow* da função **par**

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

O primeiro valor indica o número de linhas e o segundo o número de colunas em que sua janela gráfica será dividida. Para retornar ao padrão inicial de apenas um gráfico por janela basta digitar

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

ou fechar a janela do gráfico atual. Quando você chamar outro gráfico uma nova janela no padrão inicial vai se abrir

2. Para usar diferentes métodos de ligação foi alterado o argumento *method* da função *hclust*. Veja exemplos nos códigos da seção [anterior](#).

Agora vamos fazer o dendrograma da análise de agrupamento para nossa amostra das comunidades discretas, usando o método de ligação média (UPMGA). Para isso:

**1. Calcule a matriz de similaridade Bray-Curtis e guarde em um objeto (sim.disc no caso):**

```
sim.disc=sim(amost.disc, indice="bc")
```

**2. Rode a análise de agrupamento e guarde em outro objeto do R (clas.disc no caso):**

```
clas.disc=hclust(as.dist(1-sim.disc), method="average")
```

**3. Faça o dendrograma, aplicando a função plot ao objeto criado com a análise:**

```
plot(as.dendrogram(clas.disc, hang=-1), ylab="Bray-Curtis", main="Ligação  
média")
```

E, por fim, você pode criar e comparar os dendrogramas obtidos com análises de agrupamento com os quatro diferentes métodos de ligação também para as comunidades discretas:

```
par(mfrow=c(2,2))  
clas.disc1a=hclust(as.dist(1-sim.disc), method="single")  
plot(as.dendrogram(clas.disc1a, hang=-1), ylab="Bray-Curtis", main="Ligação
```

```
simples")
clas.disc1b=hclust(as.dist(1-sim.disc), method="complete")
plot(as.dendrogram(clas.disc1b, hang=-1), ylab="Bray-Curtis", main="Ligação
completa")
clas.disc1c=hclust(as.dist(1-sim.disc), method="average")
plot(as.dendrogram(clas.disc1c, hang=-1), ylab="Bray-Curtis", main="Ligação
média")
clas.disc1d=hclust(as.dist(1-sim.disc), method="centroid")
plot(as.dendrogram(clas.disc1d, hang=-1), ylab="Bray-Curtis", main="Ligação
centroide", ylim=c(0,0.7))
```

## Agora é só você...



1. Explique as diferenças entre os resultados das análises de agrupamentos com diferentes métodos de ligação aplicados à mesma amostra, como as que fizemos na seção [anterior](#).
2. Compare os dendrogramas das comunidades contínuas e discretas usando um mesmo método de ligação e interprete as diferenças entre as duas comunidades em relação ao gradiente amostrado.



- Para responder a primeira pergunta você precisa entender os algoritmos de ligação. Veja a [o ótimo resumo](#) do curso de análises multivariadas de [Alan Fielding](#).

## Para saber mais

- [Análise de agrupamento na Wikipedia](#)
- [Método de centroide na Wikipedia](#)
- [Clustering and Classification methods for Biologists](#): site do excelente curso de [Alan Fielding](#).
- Manly, B. 2008. Métodos Estatísticos Multivariados: Uma Introdução. 3 Ed. Artmed, Porto Alegre. (*Uma das melhores introduções a técnicas multivariadas para biólogos*).
- Valentin, J. 2012. Ecologia Numérica: Uma Introdução à Análise Multivariada de Dados Ecológicos. Interciência, Rio de Janeiro. (*Outro ótimo texto introdutório*)

- Legendre, P., & Legendre, L. 2012. Numerical ecology. Elsevier, Amsterdam. *(o livro de referência de ecologia numérica. Completo e didático, mas é uma leitura mais avançada).*

1)

<http://pt.wikipedia.org/wiki/Algoritmo>

2)

o valor na célula [1,4] é o mesmo da célula [4,1], lembra-se?

3)

fazemos isso colocando valores NA nessas posições da matriz. NA é o código do R para dado faltante (*Not available*)

From:

<http://ecovirtual.ib.usp.br/> -

Permanent link:

[http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:comuni:comuni\\_classr](http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:comuni:comuni_classr)



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