quali.ambi

```
quali.ambi<-function(dados,N) # criando a função "quali.ambi", o primeiro
argumento são os dados e o segundo o número de amostras
{
  if(class(dados)!="data.frame") # verificando se os dados estão em um
objeto dataframe
  {
    stop("Erro: O objeto de entrada deve ser um dataframe","\n") # se não, a
função para e essa será a mensagem de texto que aparecerá
  }
  if(ncol(dados)<2) # verificando se os dados estão em um objeto dataframe
com no mínimo duas colunas
  {
    stop("Erro:O objeto de entrada deve ter no mínimo duas colunas","\n") #
se não, a função para e essa será a mensagem de texto que aparecerá
  }
  if(class(dados[,1])!="character") # verificando se a primeira coluna do
objeto de entrada apresenta valores categóricos. Ou seja, se a tabela começa
com os nomes dos taxa.
  {
    stop("Erro: O objeto de entrada deve conter a lista de taxa na primeira
coluna (classe: character)","\n") # se não, a função para e essa será a
mensagem de texto que aparecerá
  }
  if(N!=ncol(dados)-1) # verificando se o valor de N informado é igual ao
número de colunas numéricas (ou seja, tirando a coluna com a lista de taxa)
do objeto de entrada
  {
    stop("Erro: N deve ser igual ao número de amostras do objeto de
entrada","\n") # se não, a função para e essa será a mensagem de texto que
aparecerá
  }
  else # caso as premissas acima estejam ok, a função segue
  {
    tabela.base<-read.table(file="tabela_base.txt",sep=",",as.is=T) #
buscando a tabela base no diretório de trabalho do usuário e guardando em um
objeto na área de trabalho
    grupos<-rep(NA,dim(dados)[1]) # criando um vetor (inicialmente com "NAs"
e com tamanho igual ao número de linhas do objeto de entrada) para
armazenamento dos dados com o grupo bioindicador de cada taxa informado nas
linhas do objeto de entrada.
    for(i in 1:dim(dados)[1]) # criando o ciclo com número de voltas igual
ao número de linhas do objeto de entrada
    {
      grupos[i]<-abs(sum(which(dados[i,1]!=tabela.base[,1]))-sum(1:423)) # o
contador "i" representará cada taxa (cada linha) constante no objeto de
entrada e a partir deste calculo obteremos um vetor com as posições deles na
tabela base
```

```
    }
  if(length(grupos[which(grupos==0)])>=1) # verificando quantos taxa que não foram encontrados na tabela base e tiveram posição igual a "0" no cálculo anterior.
  {
    cat("Aviso:",length(grupos[which(grupos==0)]),"NA(s) atribuído(s) para o(s) taxa não classificado(s) como bioindicador(es)","\n","ou por estar(em) com erro de escrita no objeto de entrada","\n","\n") # aviso que ocorrerá no caso de existirem taxa não encontrados na tabela base
    grupos[which(grupos==0)]<-NA # os taxa que não forem encontrados na tabela base terão posição igual a "0". Para estes casos serão atribuídos "NAs"
  }
  grupos<-tabela.base[grupos,2] # o vetor resultado "grupos" conterá o grupo bioindicador equivalente a cada taxa de acordo com suas respectivas posições calculadas para a tabela base
  dados[,N+2]<-grupos # adicionando o vetor "grupos" como uma nova coluna no objeto de entrada, contendo o grupo bioindicador de cada taxa
  colnames(dados)[c(1,N+2)]<-c("Taxa","Grupo bioindicador") # dando nomes para a primeira e última coluna do objeto de entrada
  legenda<- c("GRUPO I: Sensíveis à poluição","GRUPO II: Indiferentes à poluição", "GRUPO III: Tolerantes à poluição", "GRUPO IV: Oportunistas de 2ª ordem", "GRUPO V: Oportunistas de 1ª ordem") # criando uma legenda para a tabela final
  tabela<-list(dados,legenda) # criando uma tabela final com a classificação de cada taxa em seu grupo bioindicador
  dados<-na.exclude(dados) # excluindo os NAs formados para os grupos não classificados a fim de que os próximos cálculos não sejam prejudicados
  dados[c(rep(dim(dados)[1],5)+1:5),]<-c(rep("sp",5),rep(0,N*5),"I","II","III","IV","V") # adicionando espécies fictícias com abundância 0 (uma para cada grupo bioindicador) com o intuito de que, no caso dos dados de entrada não possuírem espécies dos 5 grupos, as espécies fictícias vão acrescentar abundância relativa igual a zero para os grupos não encontrados. Isso é necessário para que nos gráficos finais apareçam todos os grupos bioindicadores, inclusive aqueles com o valor 0 de abundância relativa
  dados$`Grupo bioindicador`<-as.factor(dados$`Grupo bioindicador`) # transformando a coluna com os grupos bioindicadores em "fator" com os 5 níveis (5 grupos)
  porcentagem<-matrix(NA,nrow=N,ncol=5,byrow=T) # criando uma matriz para armazenar os dados de abundância relativa por grupo
  for(j in 2:(N+1)) # criando o ciclo com número de voltas que englobe desde a segunda até a última coluna do objeto de entrada (ou seja, todas as amostras)
  {
    dados[,j]<-as.numeric(dados[,j]) # transformando os dados de cada amostra dos dados de entrada em "numéricos"
    porcentagem[j-1,1:5]<-tapply(dados[,j],dados$`Grupo bioindicador`,sum)*100/sum(tapply(dados[,j],dados$`Grupo bioindicador`,sum)) # calculando a porcentagem de cada grupo bioindicador por amostra
```

```
(abundância relativa)
    colnames(porcentagem)<-c("I","II","III","IV","V") # dando o nome dos
grupos às colunas da matriz
    rownames(porcentagem)<-c(colnames(dados)[2:(N+1)]) # dando o nome das
amostras nas linhas
  }
  }
X11() # abrindo janela gráfica
if(N<=3) # se o número de amostras for igual ou menor que 3
{
  par(mfrow=c(1,N)) # esse será o padrão mais adequado e selecionado para a
representação dos gráficos
}
if(N==4) # se o número de amostras for igual a 4
{
  par(mfrow=c(2,2)) # esse será o padrão mais adequado e selecionado para a
representação dos gráficos
}
if(N>4&N<=6) # se o número de amostras estiver entre 4 e 6
{
  par(mfrow=c(2,3)) # esse será o padrão mais adequado e selecionado para a
representação dos gráficos
}
if(N>6&N<=9) # se o número de amostras estiver entre 6 e 9
{
  par(mfrow=c(3,3)) # esse será o padrão mais adequado e selecionado para a
representação dos gráficos
}
if(N>9) # se o número de amostras for maior que 9
{
  par(mfrow=c(1,1)) # esse será o padrão mais adequado e selecionado para a
representação dos gráficos
}
CB<-matrix(NA,nrow = 1,ncol = N) # criando uma matriz para armazenar os
valores do coeficiente biótico para cada amostra
colnames(CB)<-colnames(dados)[2:(N+1)] # dando o nome das amostras às
colunas da matriz
rownames(CB)<-"CB" # dando nome para a linha da matriz com os coeficiêntes
bióticos
IB<-matrix(NA,nrow = 1,ncol = N) # criando uma matriz para armazenar os
valores do índice biótico para cada amostra
colnames(IB)<-colnames(dados)[2:(N+1)] # dando o nome das amostras às
colunas da matriz
rownames(IB)<-"IB" # dando nome para a linha da matriz com os índices
bióticos
for(z in 1:N) # Criando um ciclo para fazer os gráficos e o cálculo dos
índices por amostra
{
  barplot(porcentagem[z,],xlab = "Grupos Bioindicadores",ylab = "Abundância
Relativa (%)",main = rownames(porcentagem)[z]) # código para a criação do
gráfico para cada amostra
```

```
  CB[,z]<-
((0*porcentagem[z,1])+(1.5*porcentagem[z,2])+(3*porcentagem[z,3])+(4.5*porce
ntagem[z,4])+(6*porcentagem[z,5]))/100 # cálculo do coeficiente biótico a
partir da fórmula de Borja et al. (2000) para cada amostra
##########################################################################
#####################################

# As próximas linhas selecionarão o melhor Índice Biótico, a classificação
da área e a saúde da comunidade bentônica a partir dos valores obtidos para
o coeficiente biótico, seguindo os critérios de Borja et al. (2000)

  if(0<=CB[,z]&CB[,z]<0.2) # para valores de CB de 0 a 0.2
  {
    IB[z]<-0 # o valor de IB é 0
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como não poluída","\n","\t","Saúde da
comunidade bentônica normal","\n","\n") # a área é classificada como não
poluída e a saúde da comunidade como normal
  }
  if(0.2<=CB[,z]&CB[,z]<1.2) # para valores de CB de 0.2 a 1.2
  {
    IB[z]<-1 # o valor de IB é 1
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como não poluída","\n","\t","Saúde da
comunidade bentônica pobre","\n","\n") # a área é classificada como não
poluída e a saúde da comunidade como pobre
  }
  if(1.2<=CB[,z]&CB[,z]<3.3) # para valores de CB de 1.2 a 3.3
  {
    IB[z]<-2 # o valor de IB é 2
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como levemente poluída","\n","\t","Saúde da
comunidade bentônica instável","\n","\n") # a área é classificada como
levemente poluída e a saúde da comunidade como instável
  }
  if(3.3<=CB[,z]&CB[,z]<4.3) # para valores de CB de 3.3 a 4.3
  {
    IB[z]<-3 # o valor de IB é 3
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como de nível médio de
poluição","\n","\t","Saúde da comunidade bentônica transicional para
perturbada","\n","\n") # a área é classificada como de nível médio de
poluição e a saúde da comunidade como transicional para perturbada
  }
  if(4.3<=CB[,z]&CB[,z]<5) # para valores de CB de 4.3 a 5
  {
```

```
    IB[z]<-4 # o valor de IB é 4
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como de nível médio de
poluição","\n","\t","Saúde da comunidade bentônica perturbada","\n","\n") #
a área é classificada como  de nível médio de poluição e a saúde da
comunidade como perturbada
  }
  if(5<=CB[,z]&CB[,z]<5.5) # para valores de CB de 5 a 5.5
  {
    IB[z]<-5 # o valor de IB é 5
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como de nível alto de
poluição","\n","\t","Saúde da comunidade bentônica transicional para muito
perturbada","\n","\n") # a área é classificada como de nível alto de
poluição e a saúde da comunidade como transicional para muito perturbada
  }
  if(5.5<=CB[,z]&CB[,z]<6) # para valores de CB de 5.5 a 6
  {
    IB[z]<-6 # o valor de IB é 6
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como de nível alto de
poluição","\n","\t","Saúde da comunidade bentônica muito
perturbada","\n","\n") # a área é classificada como de nível alto de
poluição e a saúde da comunidade como muito perturbada
  }
  if(6<=CB[,z]) # se o valor de CB de 6 para cima
  {
    IB[z]<-7 # o valor de IB é 7
    cat("\t","Coeficiente Biótico", colnames(CB)[z], "=", CB[,z],
"\n","\t","Índice Biótico", colnames(CB)[z], "=", IB[z], "\n","\t","Área",
colnames(CB)[z], "classificada como extremamente poluída","\n","\t","Saúde
da comunidade bentônica azóica","\n","\n") # a área é classificada como
extremamente poluída e a saúde da comunidade como azóica
  }
}
##########################################################################
#######################################

if(N>1) # se os dados de entrada possuirem mais de uma amostra, um gráfico a
mais será construído na sequência
{
  X11() # se abre uma janela gráfica
  par(mfrow=c(1,1)) # volta o padrão gráfico ao normal (um gráfico por vez)
  indices<-matrix(c(CB[1,],IB[1,]),nrow = N,ncol = 2,byrow = F) # junta os
coeficiêntes e índices bióticos em uma mesma matrix
  rownames(indices)<-colnames(CB) # da nome às linhas da matix
  colnames(indices)<-c(rownames(CB),rownames(IB)) # da nome às colunas da
matriz
```

```
  dotchart(indices,pch=c(rep(1,N),rep(16,N)),xlab="Valores dos índices") #
cria um gráfico de pontos com os valores dos índices por amostra
  lines(x=c(indices[,1]),y=c((N+3):(N*2+2))) # cria uma linha para ligar os
pontos dos diferentes  coeficientes bióticos
  lines(x=c(indices[,2]),y=c(1:N),lty="dotted") # cria uma outra linha para
ligar os pontos dos diferentes índices bióticos
  return(tabela) # retorna a tabela final criada la no ínicio, a qual virá
acompanhada dos gráficos e dos resultados de classificação das áreas, a
saúde da counidade e índices
}
else # se os dados possúem apenas uma amostra, esse último gráfico não será
produzido
{
  return(tabela)# retorna a tabela final criada la no início, a qual virá
acompanhada do gráfico de abundância relativa e dos resultados de
classificação da área, a saúde da comunidade e índices
}
}
```

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:ivanlaurino:quali.ambi

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