

# André Francisco Bucci



Mestrando em Oceanografia Biológica (Instituto Oceanográfico - IOUSP) Projeto “Efeitos de morfotipos de fitoplâncton no comportamento espectral da absorção da luz, e possíveis implicações para a determinação de carbono particulado por sensoriamento remoto.”

Desenvolvo meu projeto de mestrado no Centro de Biologia Marinha (CEBIMar-USP) sob orientação da Profa. Dra. Áurea Maria Ciotti. Basicamente, trabalho com dados bio-óticos e de contagem de fitoplâncton em microscópio provenientes de cruzeiros oceanográficos realizados na costa brasileira nos anos de 2010 e 2011.

bucci.andre@usp.br

## Exercícios

[exercicio01\\_FINAL](#)

[exercicio02\\_FINAL](#)

[exercicio03\\_FINAL abs\\_costarj\\_dataout.txt](#)

[exercicio04\\_FINAL](#)

[exercicio05\\_FINAL](#)

[exercicio06\\_FINAL](#)

[exercicio07\\_FINAL](#)

[exercicio09\\_FINAL](#)

## Trabalho Final

### Plano A (Principal)

Produzir uma função que seja capaz de agrupar dados (colunas) de múltiplos arquivos provenientes de espectrofotômetros (pós-processamento) em um único dataframe e salvar esses dados em um arquivo único.

Especificamente:

1. Todos os arquivos possuem as mesmas variáveis e exatamente na mesma ordem dentro do arquivo (são arquivos padrões)
2. Os arquivos lidos devem estar no mesmo diretório

3. A rotina deverá ser rodada para cada coluna desejada (sempre será utilizada a mesma coluna para todos os arquivos)
4. As colunas dos múltiplos arquivos serão salvos em um dataframe e depois salvo em um arquivo (.txt de preferência)

## Plano B

Produzir uma função que realize a normalização de curvas de absorção de luz e plot o resultado (múltiplas curvas) em um único gráfico.

Especificamente:

1. Cada coluna no arquivo representa os valores que irão compor a curva
2. As curvas serão normalizadas (divididas) pela média dos valores de absorção em um intervalo de comprimento de onda
3. Os valores resultantes serão salvos em um novo dataframe
4. As novas curvas serão postadas em um único gráfico

## Comentário

Olá André, Pode tocar a primeira proposta, mas amplie ela um pouco. Tipo, poder escolher uma única ou várias variáveis (inclusive todas) e juntando cada uma delas em arquivos diferentes, algo assim! Tente usar também funções interativas como `choose.files()` para facilitar a vida de outros usuários. — [Alexandre Adalardo de Oliveira](#) 2012/04/03 20:52

## Função Final

### Com texto comentado

```
agrupa.col=function(files, col, filename, wave=TRUE){  
  
  if (missing(files)){files=list.files()}  
  else {files=files}  
  #importando os arquivos fornecidos pela lista em "files"  
  datalist=lapply(files,function(files){read.table(files,header=TRUE,  
as.is=TRUE, sep="\t", dec=".", fill=TRUE)})  
  #transforma a lista de arquivos importados em dataframe, dividindo-os em  
  colunas  
  data=data.frame(datalist);  
  
  a=seq(0,length(files));  
  b=ncol(data.frame(datalist[1]))  
  
  coluna=(a*b)+col; #seleciona todas as colunas da mesma variável de acordo  
  com o fornecido em "col"
```

```
coluna=coluna[1:(length(coluna)-1)] #remove o último elemento do vetor para
que
# nao seja maior que o numero maximo de colunas do dataframe inicial

#seleciona os dados desejados
dataout=data[coluna];

# idealmente a funcao foi criada para agrupar arquivos de espectrofotometro
# da forma que idealmente, a primeira coluna deve ser os comprimento de onda
# de leitura do aparelho (wave=TRUE), mas pode ser usada para qualquer tipo
de arquivo
# descartando a primeira coluna de comprimentos de onda (wave=FALSE)
if (wave==TRUE){
  wave=data[1]
  dataout=data.frame(wave,dataout);
  #os nomes das colunas do arquivo de saida sera os nomes dos arquivos da
lista inicial "." nome da coluna da variavel
  # ex. "arquivo1.coluna1", "arquivo2.coluna1", com a primeira coluna sendo
o comprimento de onda que o espectrofotometro usou
  names(dataout)=c("wavelength",paste(files, names(data[coluna[1]])));
}
else
{
  names(dataout)=c(paste(files, names(data[coluna[1]])));
  dataout=data.frame(dataout);
}
# cria um arquivo com as colunas desejadas
write.table(dataout, file=paste(filename, ".txt", sep=""), quote=FALSE,
sep="\t", col.names=names(dataout), row.names=FALSE, dec=".")

  return(dataout)
}
```

## Sem texto comentado

```
agrupa.col=function(files, col, filename, wave=TRUE){
  if (missing(files)){files=list.files()}
  else {files=files}
  datalist=lapply(files,function(files){read.table(files,header=TRUE,
as.is=TRUE, sep="\t", dec=".", fill=TRUE)})
  data=data.frame(datalist);
  a=seq(0,length(files));
  b=ncol(data.frame(datalist[1]))
  coluna=(a*b)+col;
  coluna=coluna[1:(length(coluna)-1)]
  dataout=data[coluna];
  if (wave==TRUE){
    wave=data[1]
    dataout=data.frame(wave,dataout);
    names(dataout)=c("wavelength",paste(files, names(data[coluna[1]])));
  }
}
```

```
}  
else  
{  
  names(dataout)=c(paste(files, names(data[coluna[1]])));  
  dataout=data.frame(dataout);  
}  
  
write.table(dataout, file=paste(filename, ".txt", sep=""), quote=FALSE,  
sep="\t", col.names=names(dataout), row.names=FALSE, dec=".")  
return(dataout)  
}
```

## Script e Arquivos

[scriptfuncao\\_agrupa.col](#)  
[arquivoteste01](#)  
[arquivoteste02](#)  
[arquivoteste03](#)  
[arquivoteste04](#)  
[arquivoteste05](#)

## Página Ajuda

[agrupa.col HELP!](#)

agrupa.col  
R Documentation

package:unkonw

Multiple Data Input

Description

Reads multiple files in table format of the same length and creates a data frame and a .txt file from the desired common column (e.g. 2nd column of all files), with cases corresponding to lines and variables to the columns selected. This function was designed to be used with spectrophotometers data files after processing. By changing its arguments, it can be used with any type of file.

Usage

```
read.table(files, col, filename, wave=TRUE)
```

## Arguments

`files` a list of the files which the data are to be read from. Each row of the table appears as one line of the file.

If `<files>` is not specified, `<agrupa.col()>` will automatically list all files in current directory `<getwd()>` and use these to import the data.

`col` a integer positive value indicating which column of the files should be used to group the data.

`<agrupa.col()>` only works for one specified column at a time. Multiple columns are not supported and each file generated will have the same column position from the multiple files. For more than one column selected, user must run `<agrupa.col()>` for each different column.

`filename` the name of the output file without any extensions. `<filename>` must be used under quote (e.g. "myfile").

`wave` logical, if TRUE (default), then the first column of the output file will be the wavelength values for the spectrophotometer files. Assuming a standard spectrophotometer file to have its first column as the reading wavelengths. If FALSE, the output file will be saved without the wavelengths.

## Details

This function is an auxiliary way of reading spectrophotometer data into R.

Unless `<files>` is specified, all files from the current directory will be read in. `<files>` can be a list containing the files names `<list.files()>` or a single file, in this case `<agrupa.col()>` will save only the correspondent column from `<col>`.

All files provided in `<files>` must have the same number of column and rows, since the data will be coerced in a data frame `<data.frame()>`. For all effects, input data should be tabulation separated and with "." (dot) as decimal separator since `<read.table()>` is used to read the input files into the function.

If `<filename>` isn't quoted, `<agrupa.col()>` will not work and a warning message will follow.

## Values

A data frame `<data.frame()>` containing the specified column `<col>` from all files read.

Also, `<agrupa.col()>` will save a .txt file into the working directory with

the name provided in <filename>. The file will have separation based on tabulations, and decimal separator as ".".

### Warning

If <col> is 0 the following error message will appear:

```
Error in names(dataout) = c(paste(files, names(data[coluna[1]]))) :  
'names' attribute [1] must be the same length as the vector [0]
```

If <col> is not provided, the following error message will appear:

```
Error in (a * b) + col : 'col' is missing
```

If <filename> is not quoted, the following error message will appear:

```
Error in paste(filename, ".txt", sep = "") : object 'object' not found
```

### Note

This is a function to rapidly access spectrophotometer data from multiple files and could be used with any readable file, while attending the premisses provided above and changing <agrupa.col()> arguments.

### References

Crawley, M. J. (2007). Data Input. Chapter 3 of The R Book. eds M. J. Crawley. John Wiley & Sons, Ltd.

### See Also

`read.table()`, `list.files()`, `write.table()`, `data.frame()`

### Author

Andre Bucci  
bucci.andre@usp.br

### Examples

```
test1=c(rep(1,3), rep(2,3), rep(3,3))  
test2=c(rep("a",3), rep("b",3), rep("c",3))  
test1=matrix(test1,3,3)  
test2=matrix(test2,3,3)  
write.table(test1, "test1.txt", sep="\t", dec=".")  
write.table(test2, "test2.txt", sep="\t", dec=".")
```

```
files=list.files(pattern="test")
agrupa.col(files, col=2, filename="agrupatest", wave=FALSE)
```

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2013:alunos:trabalho\\_final:bucci.andre:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:bucci.andre:start) 

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