

```
lipidomics.process <- function(input,n,method,triplicate,group.names)
#inicio da funcao
{
#Inicio da funcao
input[,1] <-as.factor(input[,1])      #fazer as amostras como fator
input[,2] <-as.factor(input[,2])      #fazer os lipídios como fator
lista <- split(input, f = input[,1]) #passar as diferentes amostras para
lista
samples.names <- c(levels(input[,1]))#vetor com os nomes das amostras

#Etapa de checagem dos parametro
if(is.numeric(n) != TRUE){
  stop("inserir um valor de n inteiro") #Verificar se o numero de amostras
  eh um numero inteiro
}

if(is.factor(input[,1]) != TRUE || length(levels(input[,1])) != n ) #checar
se a primera coluna eh fator e se os levels eh igual ao numero de amostras
{
  stop("Checar se a primeira coluna se refere as amostras, olhar o exemplo
do help")
}

if(is.numeric(input[,3]) != TRUE || is.na(mean(input[,3])) != FALSE) #
checar se os valores de area sao numeros e se ha algum NA
{
  stop("Checar se a terceira coluna se refere aos lipídios e se todos os
valores sao numericos sem NA, olhar o exemplo do help")
}

#remover os diferentes tipos de ionizacao
for(i in 1:length(levels(input[,1])))
{
  lista[[i]][,2] <- sub(" NH4","",lista[[i]][,2]) #remover NH4 do nome dos
lipids
  lista[[i]][,2] <- sub(" H","",lista[[i]][,2])   #remover H do nome dos
lipids
  lista[[i]][,2] <- sub(" W","",lista[[i]][,2]) #remover W(H2O) do nome dos
lipids
  lista[[i]][,2] <- sub(" AcO","",lista[[i]][,2]) #remover AcO do nome dos
lipids
  lista[[i]][,2] <- sub(" Form","",lista[[i]][,2])#remover Form do nome dos
lipids
  lista[[i]][,2] <- sub(" Cl","",lista[[i]][,2])  #remover cL do nome dos
lipids
}

#somando as areas dos lipídios ionizados de forma diferentes, pois agora
estao com o mesmo nome
for(j in 1:length(levels(input[,1])))
```

```
{
  lista[[j]] <- aggregate(lista[[j]][,3], by=list(lista[[j]][,2]), FUN=sum)
#somando as areas dos lipidios com o mesmo nome
}

V1 <-c(rep(NA ,length(grep("IS",lista[[1]][,1])))) # Vetor com NA para criar
um dataframe que contenha os valores dos IS
is.df <- data.frame( V1, row.names = c(
lista[[1]][grep("IS",lista[[1]][,1]),1])) #Data frama com os valores de IS e
a coluna com cada tipo de IS

#salvando as areas do padrao internos em um objeto e removendo os IS da
lista

for(k in 1:length(levels(input[,1])))
{
  is.df[,k] <- lista[[k]][grep("IS",lista[[k]][,1]),2] #Adicionando os
valores de is em uma nova coluna
  lista[[k]] <- lista[[k]][-c(grep("IS",lista[[k]][,1])),] #Removendo os IS
da lista final
}

colnames(is.df) <- c(levels(input[,1])) # Colocar os nomes das amostras nas
colunas correspondente

#corrigindo pelo valor do padrao interno (IS) correspondente

List.of.IS <- read.csv(paste(getwd(),"IS.list.csv",sep = ""),as.is = TRUE,
header = TRUE) #objeto que sera usado para normalizar os PLs
List.of.IS[,3] <- as.numeric(List.of.IS[,3]) #faz com que a coluna dos
valores dos IS seja numerico

if(method == "neg")
{

for(l in 1:length(levels(input[,1])))
{
  lista[[l]][grep("PC ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PC
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[1,2]),l])* #Normalizar PC com IS 1
                                (List.of.IS[1,3]))
#Corrige PC com CF 1
  lista[[l]][grep("PE ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PE
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[2,2]),l])*#Normalizar PE com IS 2
                                (List.of.IS[2,3]))
#Corrige PE com CF 2
  lista[[l]][grep("PS ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PS
",lista[[l]][,1]),2]
```

```

/is.df[paste(List.of.IS[3,2]),l)]*#Normalizar PS com IS 3
                                (List.of.IS[3,3]))
#Corrige PS com CF 3
  lista[[l]][grep("PG ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PG
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[4,2]),l)]*#Normalizar PG com IS 4
                                (List.of.IS[4,3]))
#Corrige PG com CF 4
  lista[[l]][grep("PI ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PI
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[5,2]),l)]*#Normalizar PI com IS 5
                                (List.of.IS[5,3]))
#Corrige PI com CF 5
  lista[[l]][grep("PA ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PA
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[6,2]),l)]*#Normalizar PA com IS 6
                                (List.of.IS[6,3]))
#Corrige PA com CF 6
  lista[[l]][grep("CL ",lista[[l]][,1]),2]<- ((lista[[l]][grep("CL
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[7,2]),l)]*#Normalizar CL com IS 7
                                (List.of.IS[7,3]))
#Corrige CL com CF 7
  lista[[l]][grep("AEG ",lista[[l]][,1]),2]<- ((lista[[l]][grep("AEG
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[8,2]),l)]*#Normalizar AEG com IS 8
                                (List.of.IS[8,3]))
#Corrige AEG com CF 8
  lista[[l]][grep("PDME ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PDME
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[9,2]),l)]*#Normalizar PDME com IS 9
                                (List.of.IS[9,3]))
#Corrige PDME com CF 9
  lista[[l]][grep("PME ",lista[[l]][,1]),2]<- ((lista[[l]][grep("PME
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[10,2]),l)]*#Normalizar PME com IS 10
                                (List.of.IS[10,3]))
#Corrige PME com CF 10
  lista[[l]][grep("Cer ",lista[[l]][,1]),2]<- ((lista[[l]][grep("Cer
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[11,2]),l)]*#Normalizar Cer com IS 11
                                (List.of.IS[11,3]))
#Corrige cer com CF 11
  lista[[l]][grep("SM ",lista[[l]][,1]),2]<- ((lista[[l]][grep("SM
",lista[[l]][,1]),2]
/is.df[paste(List.of.IS[12,2]),l)]*#Normalizar SM com IS 12
                                (List.of.IS[12,3]))
#Corrige SM com CF 12
  }
}
if(method == "pos")

```

```
{
for(m in 1:length(levels(input[,1])))
{
  lista[[m]][grep("CE ",lista[[m]][,1]),2]<- ((lista[[m]][grep("CE
",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[1,2]),m))* #Normalizar CE com IS 1
                                     (List.of.IS[1,3]))
#Corrige CE com CF 1
  lista[[m]][grep("Cholesterol",lista[[m]][,1]),2]<-
((lista[[m]][grep("Cholesterol",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[2,2]),m))* #Normalizar Cholesterol com IS 2
                                     (List.of.IS[2,3]))
#Corrige Cholesterol com CF 2
  lista[[m]][grep("Sitosterol",lista[[m]][,1]),2]<-
((lista[[m]][grep("Sitosterol",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[3,2]),m))* #Normalizar Sitosterol ou Sitosterol
Ester com IS 3
                                     (List.of.IS[3,3]))
#Corrige Cholesterol com CF 3
  lista[[m]][grep("Campesterol",lista[[m]][,1]),2]<-
((lista[[m]][grep("Campesterol",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[4,2]),m))* #Normalizar Campesterol ou Campesterol
Ester com IS 4
                                     (List.of.IS[4,3]))
#Corrige Cholesterol com CF 4
  lista[[m]][grep("Q",lista[[m]][,1]),2]<-
((lista[[m]][grep("Q",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[5,2]),m))* #Normalizar coenzimas Q10, Q9 ou Q8 com
IS 5
                                     (List.of.IS[5,3]))
#Corrige Q10, Q9 ou Q8 com CF 5
  lista[[m]][grep("DAG ",lista[[m]][,1]),2]<- ((lista[[m]][grep("DAG
",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[6,2]),m))* #Normalizar DAG com IS 6
                                     (List.of.IS[6,3])) #Corrige
DAG com CF 6
  lista[[m]][grep("MAGE ",lista[[m]][,1]),2]<- ((lista[[m]][grep("MAGE
",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[7,2]),m))* #Normalizar MAGE com IS 7
                                     (List.of.IS[7,3])) #Corrige
MAGE com CF 7
  lista[[m]][grep("TAG ",lista[[m]][,1]),2]<- ((lista[[m]][grep("TAG
",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[8,2]),m))* #Normalizar TAG com IS 8
                                     (List.of.IS[8,3])) #Corrige
TAG com CF 8
  lista[[m]][grep("DAGE ",lista[[m]][,1]),2]<- ((lista[[m]][grep("DAGE
",lista[[m]][,1]),2]
/is.df[paste(List.of.IS[9,2]),m))* #Normalizar DAGE com IS 9
                                     (List.of.IS[9,3]))
```

```

#Corrige DAGE com CF 9
}
}
Finalmean <- data.frame(lista[[1]][,1]) #objeto que será guardado os valores
das medias
Finalsd <- data.frame(lista[[1]][,1]) #objeto que será guardado os valores
de desvio padrao
dir.create("Final") #diretorio onde serao guardados dos dados arrumados

if(triplicate == TRUE){
#Agrupando as replicatas em um mesmo dataframe guardados em uma lista(listb)
#Por enquanto so triplicata
listb <- as.list(V1)
for( p in 1:length(group.names))
{
  listb[[p]] <- data.frame(lista[[3*p-2]],
lista[[3*p-2)+1]][,2],lista[[3*p-2)+2]][,2]) #agrupando triplicatas, 3p-2
eh para ir mudando em triplicas:1,4,7,...
  listb[[p]][,5] <- rowMeans(listb[[p]][,2:4])
#Media da triplicata
  listb[[p]][,6] <- apply(listb[[p]][,2:4],1,FUN = sd)
#Desvio padrao da triplicata
  colnames(listb[[p]]) <- c(group.names[p], "1","2","3","Mean","SD")
#Nomes das colunas, indicando a replicata, media e desvio padrao
  write.csv(listb[[p]],
            file = paste(getwd(),"/Final/",group.names[p],".csv",
                        sep = ""), row.names = F)
#salvando cada dataframe da lista b em arquivos csv.
  Finalmean[,p+1] <- data.frame(listb[[p]][,5])
  Finalsd[,p+1] <- data.frame(listb[[p]][,6])
}
colnames(Finalmean) <- c("mean",c(group.names)) # alterando os nomes das
colunas do dataframe que contem as media
colnames(Finalsd) <- c("sd",c(group.names)) # alterando os nomes das colunas
do dataframe que contem as desvio padrao
write.csv(Finalmean, file = paste(getwd(),"/Final/Finalmean.csv",sep = ""),
row.names = F) #salvando a planilha que contem as medias
write.csv(Finalsd, file = paste(getwd(),"/Final/Finalsd.csv",sep = ""),
row.names = F) #salvando a planilha que contem as medias
} else{
  for(m in 1:n)
  {
    Finalmean[,m+1] <- data.frame(lista[[m]][,2]) #gravando os valores
corrigidos em um dataframe
  }
  colnames(Finalmean) <- c("",c(levels(input[,1]))) # alterando os nomes das
colunas do dataframe para os nomes das amostras
  write.csv(Finalmean, file = paste(getwd(),"/Final/Final.csv",sep = ""),
row.names = F) #salvando a planilha que contem as ares normalizadas
}

```

}

From:

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

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:rodrigo_faria:funcao 

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