



# Leticia Sicchieri

Aluna de doutorado do IPEN/USP.

Projeto: Utilização de novos biossensores para avaliação do risco cardiovascular.

[exec](#)

## PROPOSTA DE TRABALHO FINAL

**Plano A** Para analisar um espectro de fluorescência é necessário fazer um tratamento dos dados. Esse tratamento serve para corrigir os valores de base de intensidade, ou seja, subtrair um baseline. Isso pode ser feito através de um ajuste linear do ponto inicial e final do espectro, e a partir da equação de reta se obtém os valores da linha base para cada ponto no eixo x. Após essa subtração o ponto inicial e final ficam em zero, sendo possível em seguida comparar os espectros. A ideia da minha função será fazer a subtração desse “baseline” de cada espectro e em seguida calcular o valor da área integrada do espectro e armazenar em um data frame ou em uma matriz os valores obtidos.

**Plano B** Atualmente, a quantificação do colesterol ruim (LDL) é feita a partir de uma equação que leva em consideração os valores de Colesterol Total, HDL e Triglicerídeos. O risco de desenvolver uma doença cardiovascular é pensado levando em consideração vários fatores, dentre eles: IMC, diâmetro da cintura, concentração de LDL, valor da pressão. A proposta do plano B seria uma função que retornaria de maneira simplificada e não diagnostica, se a pessoa tem um risco baixo, moderado ou alto de desenvolver doença cardiovascular, para isso a pessoa informaria o valor de colesterol total, HDL, Triglicerídeos, IMC, diâmetro da cintura e função retornaria a concentração de LDL, e qual o risco de desenvolver doença cardiovascular. Lembrando que essa função tem como finalidade apenas mostrar de maneira muito simplificada esse risco e não pode ser considerada como uma forma de se obter um diagnostico.

Comentários: A proposta A está ótima. Ela está bem embasada, bem explicada, vai ser certamente útil para você e outros pesquisadores. Como ela deve ficar pronta rápido, pense em formas de incluir um pouco mais de funcionalidade: talvez algum aviso para detectar erros comuns, ou usar algum algoritmo mais interessante de [integração numérica](#) para calcular a área embaixo da curva.

A proposta B está muito simples. Pelo que eu entendi, ela faria uma conta e um “if”, é isso? —[André Chalom](#)

## Proposta ajustada

A proposta A retornará além da área integrada, o valor máximo de emissão e o comprimento de onda dessa intensidade, e talvez plotar o espectro de emissão. Para a proposta B, ela faria a conta, e para os outros fatores ela iria ver se a pessoa tem um item acima do permitido ou mais e diria dependendo

disso qual seria o risco, pensei assim, se o colesterol for acima do permitido ele armazena 1, se o IMC for acima do normal ele soma 1 ao que já foi armazenado do anterior, se a pressão for normal ele não faz nada e no final ele irá ver dependendo desse valor qual o risco.

**\*\*Plano A\*\***

HELP DA FUNÇÃO

area

package:unknown

R Documentation

Função para subtrair straight line de espectros de fluorescência e calcular a área integrada linear e mostrar intensidade de pico e seu comprimento de onda.

Descrição

A função área serve para subtrair o straight line de espectros de fluorescência a partir de um data.frame com duas colunas, na primeira os valores do eixo x (comprimento de onda) e na segunda os valores do eixo y (Intensidade) do espectro. Primeiro a função irá encontrar o ponto máximo de intensidade (pico de emissão) e a partir dele encontrar os pontos de mínimo do início e do final do espectro. Com esses pontos será obtido o baseline que será subtraído de todo espectro. Por fim o programa irá retornar um data.frame com o valor de pico de emissão, o comprimento de onda desse pico e o valor da área integrada.

Uso:

area(x)

Argumento:

x: um data.frame com duas variáveis, a primeira o eixo x do gráfico e a segunda o eixo y do gráfico.

Detalhes: A função irá calcular uma função de reta a partir dos pontos de mínimo do início e do final do gráfico e irá gerar a partir dos coeficientes do ajuste os novos valores de y. Em seguida, esses novos valores serão subtraídos dos valores de y iniciais, gerando um novo conjunto de dados. Por fim a função irá calcular a área integrada a partir de áreas de trapézios, e irá mostrar a intensidade de pico e em qual comprimento de onda está esse pico.

Valor:

A função retornara um data.frame com as seguintes informações nessa ordem:

lambda: Comprimento de onda do pico de emissão,

maximum: valor de intensidade máxima desse pico,

area: valor de área integrada.

Warning:

Os dados devem conter a mesma quantidade de valores nas duas colunas, ou a função não irá rodar corretamente.

Autor:

Leticia Bonfante Sicchieri

leticiabonfante@gmail.com

Referencia:

Calculo de área por trapézios:

[http://ww1.univap.br/spilling/CN/CN\\_Capt6.pdf](http://ww1.univap.br/spilling/CN/CN_Capt6.pdf)

Exemplo: Utilizando um espectro de fluorescência;

Carregar o conjunto de dados do arquivo txt chamado exemplo.txt

area(exemplo):

```
lambda maximum area
486 258773.6 18263544
```

Código da função

```
area = function(x)
{
  colnames(x)[1]="x1" #troca o nome da coluna para facilitar
  colnames(x)[2]="y1" #troca o nome da coluna para facilitar
  maior=0             #cria um vetor maior e armazeno 0 nele
  lambda=0           #cria um vetor lambda e armazeno 0 nele
  pos=0              #cria um vetor pos e armazeno 0 nele
  for (i in 1:length(x$y1)) # nesse for ele encontra o maior valor de y e
armazena o numero em maior e o valor de x em lambda e em pos a posição
desses valores
  {
    if (x[i,2]>maior)
    {
      maior=x[i,2]
      lambda=x[i,1]
      pos=i
    }
  }
  menor1=maior       #cria um vetor menor1 e guarda o valor que estava no
vetor maior
  menor2=maior       #cria um vetor menor2 e guarda o valor que estava no
vetor maior
  lambda1=0          #cria um vetor lambda1 e armazena 0
  lambda2=0          #cria um vetor lambda2 e armazena 0
  for (i in pos:length(x$y1)) #nesse for ele ira procurar o menor valor de y
a partir do ponto do maior valor e ira guardar esse valor em menor1 e o
valor de x correspondente em lambda1
  {
    if (x[i,2]<menor1)
    {
      menor1=x[i,2]
      lambda1=x[i,1]
    }
  }
  for (i in 1:pos) #nesse for ele ira procurar o menor valor de y do começo
até do ponto do maior valor e ira guardar esse valor em menor2 e o valor de
x correspondente em lambda2
  {
    if (x[i,2]<menor2)
    {
      menor2=x[i,2]
      lambda2=x[i,1]
    }
  }
  pri.ponto = c(lambda1,lambda2) # coloca em um vetor os valores de lambda
  ult.ponto = c(menor1,menor2) #coloca em um vetor os valores do vetor menor
  mod.linear = lm(ult.ponto~pri.ponto) # faz o modelo linear
```

```
coef = data.frame(coef(mod.linear)) # coloca em um data.frame os
coeficientes do ajuste
w = x$x1 # separa a primeira coluna dos dados
y = coef[1,] + w*coef[2,] # calcula os novos valores de y (da reta)
novo.y = data.frame(y) # coloca os valores em um data frame
res = data.frame(x$y1,novo.y) # forma um novo data.frame com os valores da
segunda coluna dos dados com os valores calculados pelo ajuste
dif = res[1] - res[2] #subtrai os valores das duas colunas
val.final = data.frame(x$x1,dif) # coloca junto os valores do eixo x com
os valores do eixo y calculados na linha de cima
colnames(val.final)[1] = "x2" #troca o nome das colunas para facilitar
colnames(val.final)[2] = "y2" #troca o nome das colunas para facilitar
maximum=0 #cria um vetor maximum e armazena 0
lambda=0 #cria um vetor lambda e armazena 0
for (i in 1:length(val.final$y2)) # esse for procura o maior valor dos
novos dados e armazena o valor de maximo em maximum e o respectivo x em
lambda
{
  if (val.final[i,2]>maximum)
  {
    maximum=val.final[i,2]
    lambda=val.final[i,1]
  }
}
base=rep(NA, length(val.final$x2)-1) #cria um vetor com NAs
for(i in 1:length(val.final$x2)-1) # faz um for para calcular o valor das
áreas
{
  base[i] = val.final$y2[i]+val.final$y2[i+1] #soma os valores de y a soma
das bases para o cálculo da área por um trapézio.
}
base.l=data.frame(base) # coloca os valores calculados anteriormente em um
data frame
cada.area=0.5*(diff(val.final$x2[-1])*base.l) # calcula a área de cada
trapézio
area = sum(cada.area) # soma as áreas dos trapézios
resul=data.frame(lambda,maximum,area) #coloca os resultados a serem
mostrados em um único data.frame
return(resul)
}
```

[exemplo.txt funcaofinal.r](#)

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2015:alunos:trabalho\\_final:a5115355:start](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2015:alunos:trabalho_final:a5115355:start) 

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