

# monitoraR

```
monitoraR <- function(lat,long, dia.inicio, dia.final, destfile, graph=T ){
  longitude= round(abs(long) +180, digits=1) #soma o valor de longitude a
  180 e arredonda para 1 casa decimal, pois eh requisito do modelo WaveWatch
  para baixar os dados - modelo usa longitude em graus leste
  lat= round(lat, 1) #arredonda latitude para 1 casa decimal - requisito do
  modelo
  if(lat>3 | lat<(-34)){
    stop('Latitude precisa ser um numero em graus decimais entre 3N e 33S.')
  } #verificando se os valores de latitude estao dentro dos limites norte e
  sul da costa brasileira
  #verificando se os valores de longitude estao na metade oeste do globo
  if (longitude <180 & longitude>270)
    stop("Longitude deve ser um valor para a costa brasileira")
  dia.inicio <- as.Date(dia.inicio, format="%Y-%m-%d") #le o argumento
  dia.inicio colocado pelo usuario como data no formato YYYY-MM-DD -
  necessario para baixar os dados
  dia.final <- as.Date(dia.final, format="%Y-%m-%d") #le o argumento
  dia.final colocado pelo usuario como data no formato YYYY-MM-DD - necessario
  para baixar os dados
  dif.datas <- dia.final - dia.inicio #calcula o intervalo de dias entre dos
  dias escolhidos pelo usu?rio
  if (dif.datas>6){
    stop("dia.inicio e dia.final devem ter intervalo de no maximo 6 dias")
  } #verifica se os intervalos de dados estao dentro do limite maximo do
  modelo
  message('Escolha os parametros dentre opcoes fornecidas, digitando
  diretamente no console (atentar letras maiusculas e minusculas), ao final
  das proximas 2 frases') #mensagem para facilitar o entendimento do usu?rio
  estado.BR <- readline("Escolha qual estado brasileiro se encontra a praia,
  dentre as opcoes fornecidas.\n
  (AM/PA/MA/PI/CE/RN/PB/PE/AL/SE/BA/ES/RJ/SP/PR/SC/RS)") #usuario escolhe
  dentre as opcoes dadas o estado brasileiro onde se encontra a praia de
  interesse
  if(estado.BR!="AM" & estado.BR!="PA" & estado.BR!="MA" & estado.BR!="PI" &
  estado.BR!="CE" & estado.BR!="RN" & estado.BR!="PB" & estado.BR!="PE" &
  estado.BR!="AL" & estado.BR!="SE" & estado.BR!="BA" & estado.BR!="ES" &
  estado.BR!="RJ" & estado.BR!="SP" & estado.BR!="PR" & estado.BR!="SC" &
  estado.BR!="RS"){
    stop('Sua praia DEVE ESTAR localizada em um estado brasileiro
    costeiro.')
  } #verificando se o usuario escolheu certo a sigla do estado dentre as
  opcoes oferecidas
  grao.carac <- readline("Escolha a caracteristica do grao na praia
  escolhida, dentre as opcoes fornecidas. \n (AreiaG/AreiaM/AreiaF/AreiaMF)
  \n" ) #usu?rio escolhe dentre as opcoes a caracteristica do grao da praia
  escolhida
  if(grao.carac!="AreiaG" & grao.carac!="AreiaM" & grao.carac!="AreiaF" &
```

```
grao.carac!="AreiaMF"){
  stop('Características do grão fora das opções possíveis.')
} #verificando se o usuário escolheu dentre as opções possíveis
#código efetivo da função:
link <-
paste(paste0("https://coastwatch.pfeg.noaa.gov/erddap/griddap/NWW3_Global_Be
st.csv?Tper[(",dia.inicio,"):1:(",dia.final,")][((0.0):1:(0.0)][(",lat,"):1:(
",lat,")][(",longitud,"):1:(",longitud,")]),
paste0("Thgt[(",dia.inicio,"):1:(",dia.final,")][((0.0):1:(0.0)][(",lat,"):1:
(",lat,")][(",longitud,"):1:(",longitud,")]), sep=",") #gerando link, a
partir dos argumentos inseridos pelo usuário, para baixar o dataframe direto
da internet
download.file(url= link, destfile=destfile) #baixando os dados e salvando
no destfile escolhido pelo usuário
Ondas<-read.csv(file=destfile, header=T, skip=2, sep=",", col.names =
c("data", "prof", "lat", "long", "T", "Hprof")) #reorganizando os dados
baixados e renomeando como Ondas
Ondas$Hrasa <- (9.8 * Ondas$Hprof)^0.5 #cria uma coluna com as
informações de altura de ondas em águas rasas a partir da equação de
conversão Hrasa = sqrt(gravidade * Hprof)
Grao <- data.frame(c("AreiaG", "AreiaM", "AreiaF", "AreiaMF"), c(0.1,
0.043, 0.024, 0.009)) #criação do data frame Grao que armazena as
velocidades de decantação dos diversos tipos de areia
colnames(Grao) = c("nomes", "Ws") #nomeia as colunas do dataframe Grao
MareBR <- data.frame(c("AM", "PA", "MA", "PI", "CE", "RN", "PB", "PE",
"AL", "SE", "BA", "ES", "RJ", "SP", "PR", "SC", "RS"), c(7, 6.5, 6.5, 4.25,
4.25, 4.25, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2, 1.5, 1.5, 1.5, 0.25)) #criação
do data frame MareBR que armazena a relação da amplitude de mare em cada
estado costeiro brasileiro
colnames(MareBR) = c("nomes.estados", "TR") #nomeia as colunas do
dataframe MareBR
Hrasa.m <- mean(Ondas$Hrasa) #calcula a média de altura de onda em águas
rasas para todo o intervalo de dias escolhido
RTR <- MareBR[MareBR$nomes.estados== estado.BR, 2] / Hrasa.m #cálculo da
relação entre amplitude de mare do estado em que se localiza a praia (a
partir de indexação) e altura de onda média calculada no intervalo
#controle de fluxo da função: praias dominadas por mare (RTR>10) não é
possível usar a função para calcular o estágio de praia e o fator de risco
de corrente de retorno, apenas para praias dominadas por ondas e dominadas
por ondas e mare
if (RTR>10){
  stop("Praia dominada por mare, não é possível calcular estágio praias")
} #caso RTR>10, para a função se a praia for dominada por mare
if (RTR<3){
  message('Praia dominada por onda, "\n')
} #retorna a mensagem para o usuário das forcantes que regem a praia a
depende do RTR
if (RTR>3 & RTR<10){
  message('Praia dominada por onda e mare', "\n")
} #retorna a mensagem para o usuário das forcantes que regem a praia, a
```

depende do RTR

```
Hrsa.m.dia <- rep(NA, dif.datas) #cria vetor de NA do tamanho do intervalo de datas para armazenar a altura media diaria de onda a ser calculada
```

```
T.m.dia <- rep (NA, dif.datas) #cria vetor de NA do tamanho do intervalo de datas para armazenar a media diaria do periodo de onda
```

```
omega <- rep(NA, dif.datas) #cria vetor de NA do tamanho do intervalo de datas para armazenar o estagio praial em cada dia do intervalo amostrado
```

```
for (i in 1:dif.datas){ #ciclo iterativo que vai de 1 ate o intervalo de dias
```

```
  for(j in 1: dif.datas) #ciclo iterativo que vai de 1 ate o intervalo de dias
```

```
  {
```

```
    Hrsa.m.dia[j]= mean(Ondas$Hrsa[((j*14)-13): (j*14)]) #calcula a media da altura de onda para cada dia no dataframe Ondas, sendo que cada dia corresponde a 24 linhas no dataframe Ondas, e armazena na posicao j do vetor Hrsa.m.dia
```

```
    T.m.dia[j] = mean(Ondas$T[((j*14)-13): (j*14)]) #calcula a media do periodo de onda para cada dia no dataframe Ondas, sendo que cada dia corresponde a 24 linhas no dataframe Ondas, e armazena na posicao j do vetor T.m.dia
```

```
  }
```

```
    omega[i]= (Hrsa.m.dia[i]^2)/ (9.8 * (T.m.dia[i]^2) * (Grao[Grao$nomes=="AreiaF", 2])) #calcula o estagio de praia a depender da media diaria da altura de onda e da media diaria do periodo de onda para aquele dia, de acordo com a equacao de Wright and Short, e armazena na posicao i do vetor omega
```

```
  }
```

```
nome.estagio <- rep(NA, dif.datas) ##cria vetor de NA do tamanho do intervalo de datas para armazenar o nome do estagio praial que corresponde ao valor de omega calculado em cada dia
```

```
for(k in 1:dif.datas){ #ciclo iterativo que vai de 1 ate o tamanho do intervalo de dias escolhido
```

```
  if (RTR<3 & omega[k]>=6) #retorna nome do estagio praial de cada dia a depender do RTR e do valor de omega calculado pra aquele dia
```

```
  {
```

```
    nome.estagio[k]= "Dissipativo"
```

```
  }
```

```
  if (RTR<3 & omega[k]<6 & omega[k]>=1)
```

```
  {
```

```
    nome.estagio[k]="Intermediario"
```

```
  }
```

```
  if (RTR<3 & omega[k]<1)
```

```
  {
```

```
    nome.estagio[k]="Reflexivo"
```

```
  }
```

```
  if (RTR>=3 & 10>=RTR & omega[k]>=5)
```

```
  {
```

```
    nome.estagio[k]= "Ultradissipativo"
```

```
  }
```

```
  if (RTR>=3 & 10>=RTR & omega[k]<5)
```

```
{
  nome.estagio[k]="Relexivo + Intermediario"
}
}
#calcular indice de corrente de retorno de acordo com Engle et al. 2002:
t.ripcurrent <- rep(NA, dif.datas) #cria vetor de NA de tamanho do
intervalo de dias para armazenar o fator de risco diario correspondente ao
periodo de onda medio por dia
for(l in 1:dif.datas){ #ciclo iterativo que vai de 1 ate o intervalo de
datas
  #calcula o fator de risco a depender da media do periodo de onda a cada
dia (cada dia corresponde a uma posicao do vetor T.m.dia)
  if(T.m.dia[l]<6){
    t.ripcurrent[l] = 1
  }
  if (6<=T.m.dia[l] & T.m.dia[l]<9){
    t.ripcurrent[l] = 1 + 0.5
  }
  if (9<=T.m.dia[l] & T.m.dia[l]<11){
    t.ripcurrent[l] = 1 + 1
  }
  if (11<=T.m.dia[l] & T.m.dia[l]<12){
    t.ripcurrent[l] = 1 +2
  }
  if (T.m.dia[l]>=12){
    t.ripcurrent[l] = 1 + 3
  }
}
ripcurrent <- rep(NA, dif.datas) #cria vetor para armazenar o fator de
risco de corrente de retorno levando em consideracao o fator calculado em
funcao do periodo (t.ripcurrent) e o fator correspondente a altura de onda
media por dia
for(m in 1:dif.datas){ #ciclo iterativo que vai de 1 ate o intervalo de
datas
  #calcula o fator de risco a depender da media da altura de onda a cada
dia (cada dia corresponde a uma posicao do vetor Hrasa.m.dia) somado ao
fator de risco associado ao periodo de onda (t.ripcurrent)
  if(Hrasa.m.dia[m]<1){
    ripcurrent[m] = t.ripcurrent[m]
  }
  if (1<=Hrasa.m.dia[m] & Hrasa.m.dia[m]<2){
    ripcurrent[m] = t.ripcurrent[m] + 0.5
  }
  if (2 <= Hrasa.m.dia[m] & Hrasa.m.dia[m]<3){
    ripcurrent[m] = t.ripcurrent[m] + 1
  }
  if (3<= Hrasa.m.dia[m] & Hrasa.m.dia[m]<5){
    ripcurrent[m] = t.ripcurrent[m] +2
  }
  if (5<= Hrasa.m.dia[m] & Hrasa.m.dia[m]<8){
```

```
    ripcurrent[m] = t.ripcurrent[m] +3
  }
  if (Hrasa.m.dia[m]>=8){
    ripcurrent[m] = t.ripcurrent[m] + 4
  }
}
#gerando o grafico, caso o usuario deseje esta saida (graph==T):
if(graph==T)
{
  plot(omega, bty="l", pch= 19, xlab="Dias") #plota o valor de omega em
cada dia
  lines(omega, lwd=2) #liga os pontos plotados com linhas
  par(new=T) #adiciona no mesmo plot
  plot(ripcurrent, axes=F, ann=F, pch=17, col="red") #plota valores do
fator de risco das correntes de retorno a cada dia
  lines(ripcurrent, lwd=2, col="red") #liga os valores do fator de risco
de corrente de retorno com linhas
  axis(4) #adiciona eixo correspondente ao valor do risco da corrente de
retrono
  message("Escolha o ponto no grafico (cique no grÃifico) para colocar a
legenda")
  legend(locator(1), legend= c("Omega", "Ripcurrent"), pch= c(19,17),
col=c("Black", "red")) #insere legenda no local escolhido pelo usuario no
grafico
}
#final da funcao: saida eh uma lista que contem as informacoes de altura
media de onda por dia, periodo medio de onda por dia, estagio praial e
respectivo nome a cada dia e o fator de risco de corrente de retorno na
praia a cada dia
return(list("Altura media de onda por dia"=Hrasa.m.dia, "Periodo medio de
onda por dia"= T.m.dia, "Estagio praial por dia"= omega, "Nome do estgio
praial por dia"= nome.estagio, "Fator de risco de corrente de retorno por
dia"= ripcurrent)) #retorna lista com dados calculados por dia
}
```

From:

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

Permanent link:

[http://ecor.ib.usp.br/doku.php?id=05\\_curso\\_antigo:r2019:alunos:trabalho\\_final:mariana.santos.martins:monitorar](http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2019:alunos:trabalho_final:mariana.santos.martins:monitorar)

Last update: 2020/08/12 06:04