Alexander Alvarez Rosario

Doutorando em Engenharia Mecânica, Ladin - Laboratório de Dinâmica e Instrumentação, Poli-USP.

Tema de doutorado: Detecção e identificação de eventos bioacústicos marinhos, orientado pelo Prof. Linilson R. Padovese.

Meus Exercícios

Exercício 1. F

Exercício 2 F

Exercício 3 F

Exercício 4 F

Exercício 5 F

Exercício 6 F

Exercício 7 P

Exercício 9 F

Proposta de Trabalho Final

Plano A. Criar uma função que leia um arquivo de som (Gravação de Bioacústica Marinha) e apresente de forma animada esse arquivo. A função deve executar as tarefas descritas a seguir: Importar arquivo de áudio (formato wav) – Criar matriz de dados Transformar esses dados espaço Amplitude – Tempo, para dados Frequência – Tempo. Criar um espectrograma dos dados Criar um gráfico que apresente a evolução desses dos sinais numa mesma janela Opcional: Aplicar uma Análise de Componentes Principais nos dados do espectrograma e escolher as componentes de possuem a maior variância acumulada dos dados do espectrograma. Criar um novo espectrograma a partir das componentes principais escolhidas (?) Fazer uma análise de clusters nos dados do espectrograma e tentar eliminar aqueles dados que apresentem maior dissimilaridade Criar um novo espectrograma com os dados após eliminação na fase anterior (?) Os arquivos de áudio podem ser faixas escolhidas onde há algum evento previamente identificado (sons de golfinhos).

Plano B. Para medição de um sinal magnético numa estrutura da industria petroleira, conhecida como riser, são feitas medidas simultâneas de até 32 sensores por cada rodada. Uma das variáveis controladas e a tensão aplicada na estrutura e geralmente correspondem com 3 níveis de tensão. Em cada medição ou rodada são feitas 10 gravações do sinal.

Nível de Tensão n = 1,...,3

```
Repetição i             i= 1,...,3
    Sensor k            k= 1,...,32
        Medição J        j= 1,...,10
```

Cada um dos anteriores níveis corresponde com uma pasta criada no computador, o problema é juntar os dados numa matriz multidimensional para posteriores cálculos, pois o número de arquivos resultantes em cada jornada de medição é 3 x 3 x 32 x 10 = 25920 arquivos e cada arquivo contêm aproximadamente 74000 linhas. A ideia é criar uma função que seja capaz de criar uma matriz multidimensional onde cada um dos “títulos” das pastas corresponda com um fator e a medição com o dado. A função deve avaliar possíveis outliers e fazer cálculo do valor RMS dos sinais além de outros parâmetros estatísticos.

Comentários

Olá Alexandre:

1. ambas propostas parecem bem desafiadoras e úteis. A primeira mais relacionada ao seu tema de trabalho, a segunda mais relacionada a um problema operacional de juntar uma grande quantidade de dados coletados para serem trabalhados. A proposta A tem uma complicação de tratamentos dos dados que não consigo avaliar completamente todas as dificuldades envolvidas; o plano B tem um problema de limitaçao do R: 74 mil linhas e 26 mil arquivos é um entrave de memoria no R (imagino que cada linha tenha 10 gravações do sinal.. ou seja, quase 2* 10^10 registros. Pela minha experiência o R náo vai ser dar bem com um objeto dessa dimensão.

Sugiro que divida sua primeira proposta e se comprometa apenas com parte dela... e se tiver tempo faça todos os passos da proposta A! Por exemplo:

- Importar arquivo de áudio (formato wav) – Criar matriz de dados
- Transformar esses dados espaço Amplitude – Tempo, para dados Frequência – Tempo.
- Criar um espectrograma dos dados
- Criar um gráfico que apresente a evolução desses dos sinais numa mesma janela

O resto, implementa se houver tempo.

— *Alexandre Adalardo de Oliveira 2012/04/02 19:52*

Página de Ajuda

```
ftempspec                package:unknown                R Documentation


Description:
```

Representa gráficamente um arquivo de audio (formato wav) em formato de onda e em espectro de frequência.
Uso

Usage:

ftempspec(x)

Arguments:

x Um arquivo de audio en formato WAV.

Details:

A partir de um arquivo de audio são calculadas as matrices de dados no domínio do tempo (uma columna para a variável Tempo e outra para a Amplitude do sinal nesse instante), e outra matriz com os valores em frequência (a partir do cálculo da Transformada de Fourier Rápida dos dados no domínio do Tempo). Com essas matrices são então feitos dois gráficos numa janela, apresentando o comportamento do sinal em uma janela correspondente com um segundo (1 s) de audio. Assím é possível avaliar o componente no espectro de frequência de maior participação no instante avaliado. Adicionalmente cria o espectrograma do arquivo de audio e o salva na pasta de trabalho (Espectrograma.png).

Value:

Um arquivo WAVE é um arquivo de tipo RIFF (Resource Interchange File Format) utilizado para o armazenamento de dados de áudio descompactados. Muitas vezes, é identificado pela extensão. WAV em DOS sistemas legados (como o Windows). Embora os arquivos WAVE podem conter dados comprimidos, a função acima só funciona com dados não comprimidos.
References

Warning:

....

Author(s):

Alexánder Alvarez Rosario

References:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language.

```
Wadsworth & Brooks/Cole.

Singleton, R. C. (1979) Mixed Radix Fast Fourier Transforms, in Programs for
Digital Signal Processing, IEEE Digital Signal Processing Committee eds.
IEEE Press.
Ver também


Examples:

#No exemplo apresentado a seguir é plotado um arquivo de audio de canto de
baleia.
#Para carregar o arquivo de exemplo fazer a descarga em:

www.ladin.usp.br/aarosario/exemplo.wav

dados <- load.wave("exemplo.wav")

# Para ver a representação dos dados usar a função:

ftempespec(dados)

# Para ver as matrices de dados nos domínios temporal e da frequência:
dados.tempo  # Dados no domínio do Tempo
dados.freq   # Dados no domínio da Frequência
num.amos     # Número de amostras
freq.amos    # Frequência de amostragem do arquivo de audio
```

Código da Função

```
require(seewave)
require(audio) # Para usar a função load.wave()
require(graphics)
require(Hmisc) #Para usar a função minor.tick() nos gráficos

ftempspec <- function (x){
#------------------------------------------------------------------------
# Carregando pacotes necessarios para execução da ftempspec
#------------------------------------------------------------------------
freq.amos <<- x$rate                            #Freq. Amostragem
num.amos <<-  length(x)                         #Num. de Amostras
temp.wav <- num.amos / freq.amos               #Tempo duração arquivo wav
temp.amos <- temp.wav / num.amos               #Tempo uma amostra
vet.temp <- seq(0, num.amos-1, 1)              #Criar de vetor tempo
vet.temp <- t(t(vet.temp * temp.amos))
dados.1<-t(t(x))
dados.tempo <- cbind(vet.temp, dados.1)        #Dados dominio amplitude-
```

```
tempo
colnames(dados.tempo) <- c("tempo", "amplitude")
dados.tempo <<- as.data.frame(dados.tempo)
#-------------------------------------------------------------------------
# Audio do dominio amplitude - tempo para frequência - tempo
#-------------------------------------------------------------------------
#
a <- fft(dados.tempo[,2])                          # Fast Fourier Transform
b <- Mod(a)                                             #Módulo da
transformada
c <- as.integer((num.amos)/2)
dados.freq <- rbind(t(t(b[1:c])), t(t(b[(c+1):num.amos])*0))
dados.freq <<- cbind(vet.temp,dados.freq)
rm(x,dados.1, temp.wav, temp.amos,a,b,c)
#-------------------------------------------------------------------------
# Grafico dos dados (sequência animada)
#-------------------------------------------------------------------------
tlim.min<-min(dados.tempo[,2])
tlim.max<-max(dados.tempo[,2])
for (i in seq(1, (num.amos - freq.amos), by = freq.amos))
{
a<-i
b<-i+freq.amos
sub.dadost<-dados.tempo[a:b,]
sub.dadosf<-fft(sub.dadost[,2])
sub.dadosf<-Mod(sub.dadosf)
sub.dadosf<-as.data.frame(sub.dadosf)
#.........................................................................
split.screen( figs = c( 2, 1))
#Screen 1
screen(1)
plot( sub.dadost[,1],sub.dadost[,2],
        col = "red", type="l", ylim=c(1,-1),
        main="Forma de Onda (Domínio Temporal)", ylab = "Amplitude
relativa", xlab="Tempo (s)",
        cex.axis=0.75, cex.lab=0.75, cex.main = 0.75, cex.sub= 0.75,
        lwd=2.0,
        mgp = c(1.5, 0, 0),
        tck=0.025,
        xaxs="i" )
box("figure", col="green")
abline(h=0,lty=1,col="black")
minor.tick(nx=5, ny=1, tick.ratio=0.5)
#.........................................................................
#Screen 2
screen(2)
plot( sub.dadosf[,1][1:(freq.amos/2)],
        col = "blue", type="l",
        main="Espectro de Frequência", ylab = "Amplitude", xlab="Frequência
(Hz)",
        cex.axis=0.75, cex.lab=0.75, cex.main = 0.75, cex.sub= 0.75,
```

```
        lwd=2.0,
        mgp = c(1.5, 0, 0),
        tck=0.025,
        xaxs="i" )
box("figure", col="green")
abline(h=0,lty=1,col="black")
minor.tick(nx=5, ny=1, tick.ratio=0.5)
close.screen(all = TRUE)
}
Sys.sleep(5)
spectro(dados.tempo[,2], f=freq.amos, tlab="Tempo (s)", flab="Frequência")
dev.print(png, filename="C:/Tfinal/Espectrograma.png", width=500,
  height=500)
Sys.sleep(5)
dev.off()
}
```

Arquivo da Função

Link externo