

Bioacústica

Análisis de señales acústicas para su aplicación en ciencias biológicas

Lucía Ziegler

lucia.ziegler@gmail.com

Martín Rocamora

rocamora@fing.edu.uy

Programa de Formación de las Ciencias Básicas
PEDECIBA

Julio 9, 2021

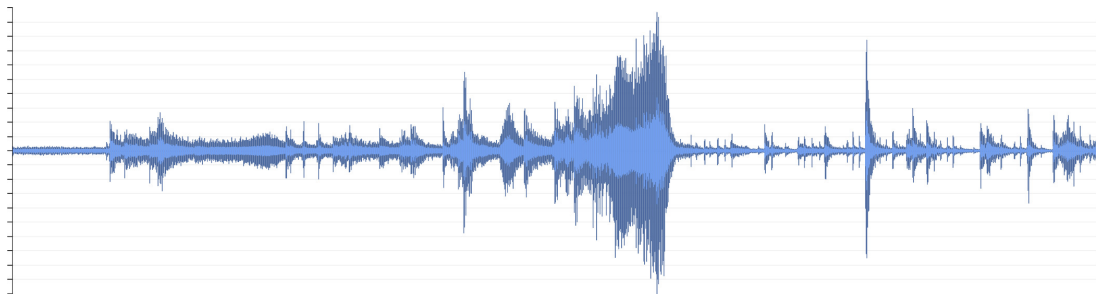


UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

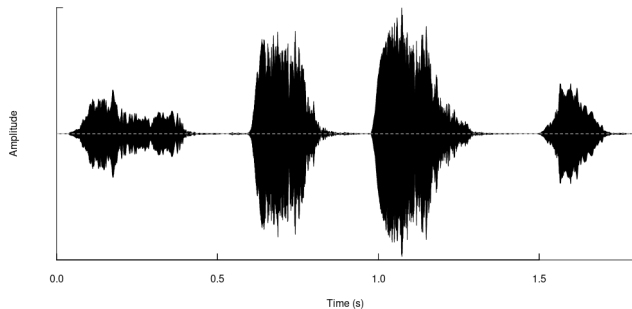
1. Forma de onda y oscilograma
2. Envolvente de amplitud
3. Análisis con ventanas deslizantes
4. Segmentación automática
5. Análisis de modulación de amplitud

- 1. Forma de onda y oscilograma**
2. Envolvente de amplitud
3. Análisis con ventanas deslizantes
4. Segmentación automática
5. Análisis de modulación de amplitud

La forma más común de representar un sonido es a través de su **forma de onda**, que consiste en un gráfico de la amplitud instantánea contra el tiempo, y que se denomina **sonograma**.



```
# importar biblioteca seewave  
library(seewave)  
  
# cargar Wave Object tico  
data(tico)  
tico  
  
# graficar oscilograma  
oscillo(tico)  
  
# escribir archivo de audio  
writeWave(object=tico,  
          filename="tico.wav")
```

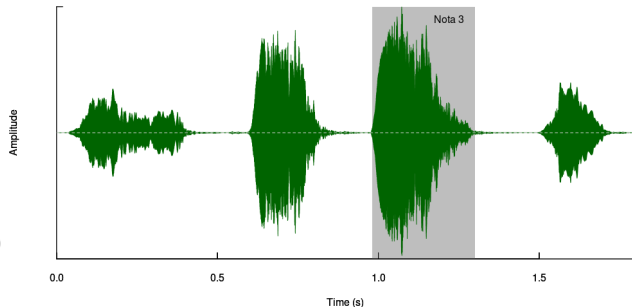


```
# importar biblioteca seewave
library(seewave)

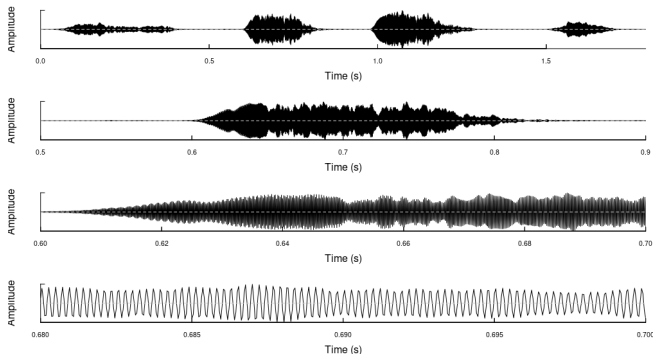
# cargar Wave Object tico
data(tico)

s <- tico@left
y <- max(abs(s))

oscillo(tico, type="n")
polygon(x=c(0.98,1.3,1.3,0.98),
        y=c(-y,-y,y,y),
        col="gray", border="NA")
text(x=1.22, y=y-y/10,
      labels="Nota 3")
par(new=TRUE)
oscillo(tico, colwave="darkgreen")
```



```
# importar biblioteca seewave  
library(seewave)  
  
# cargar Wave Object tico  
data(tico)  
  
# 4 líneas  
layout(matrix(1:4, nrow=4))  
  
# forma de onda completa  
oscillo(tico)  
  
# zoom 1  
oscillo(tico, from=0.5, to=0.9)  
  
# zoom 2  
oscillo(tico, from=0.6, to=0.7)  
  
# zoom 3  
oscillo(tico, from=0.68, to=0.70)
```

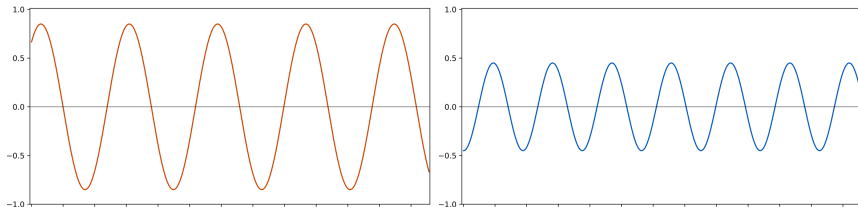


¿Preguntas?

1. Forma de onda y oscilograma
- 2. Envolvente de amplitud**
3. Análisis con ventanas deslizantes
4. Segmentación automática
5. Análisis de modulación de amplitud

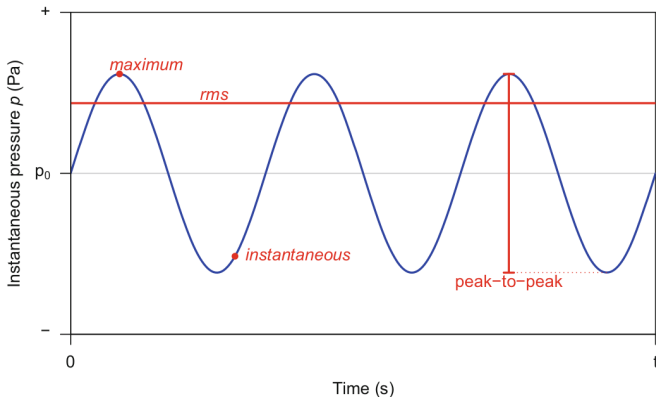
$$\begin{aligned}x(t) &= A \sin(\omega t + \varphi) \\ &= A \sin\left(\frac{2\pi}{T}t + \varphi\right) \\ &= A \sin(2\pi f t + \varphi)\end{aligned}$$

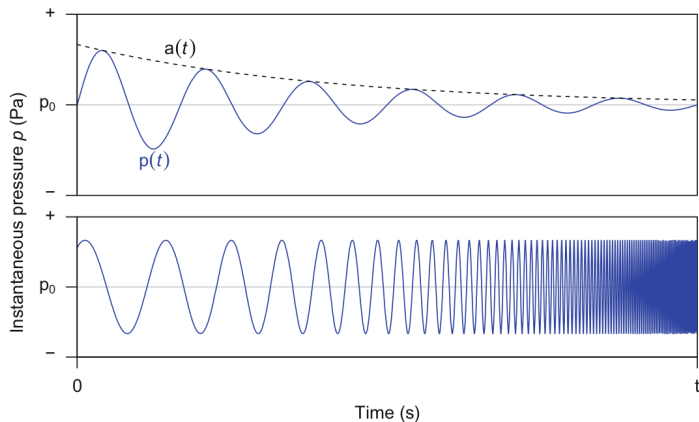
- A amplitud
- f frecuencia (Hz)
- φ fase inicial



- Amplitud máxima (A)
- Amplitud instantánea
- Amplitud pico a pico
- Amplitud RMS
(valor cuadrático medio)

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$



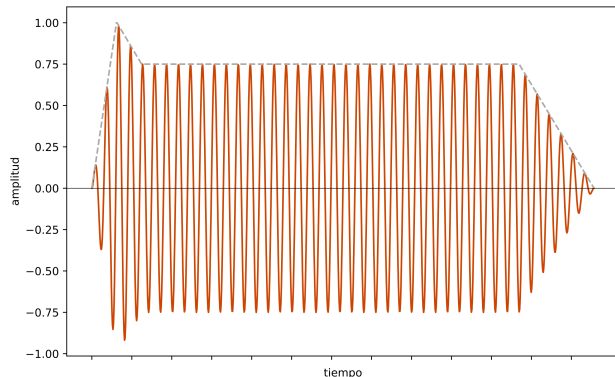


$$p(t) = Ae^{-\beta t} \sin(2\pi ft + \varphi)$$

$$p(t) = A \sin(2\pi(ft + ct^2) + \varphi)$$

La **envolvente** de una señal es la curva que une sus picos locales.

La **envolvente de amplitud** es la curva que representa cómo evoluciona la amplitud de una señal en el tiempo.



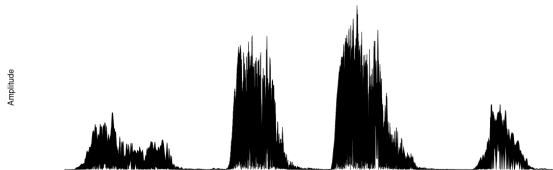
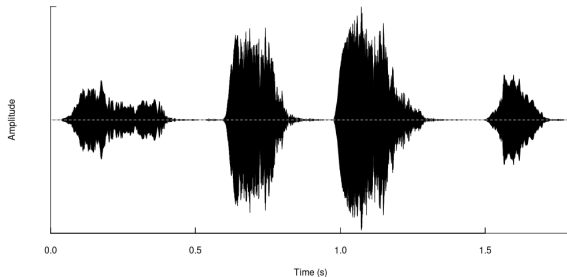
El valor absoluto de la forma de onda es una primera aproximación.

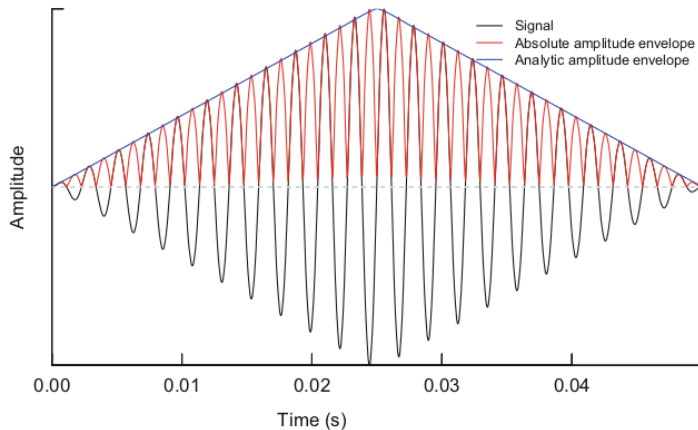
```
# importar biblioteca seewave  
library(seewave)
```

```
# cargar Wave Object tico  
data(tico)
```

```
# valor absoluto de la señal  
envelope <- abs(tico@left)
```

```
# gráfica de envolvente  
plot(envelope,  
      type="l",  
      frame.plot = FALSE,  
      xaxt="n", yaxt="n",  
      xlab="", ylab="Amplitude")
```





Comparación de valor absoluto y envolvente de amplitud calculada con transformada de Hilbert.

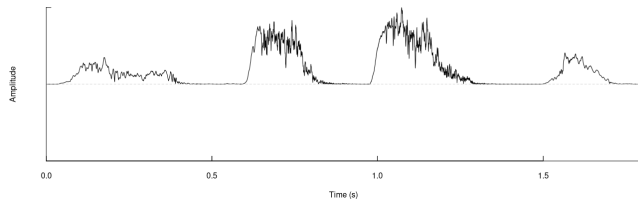
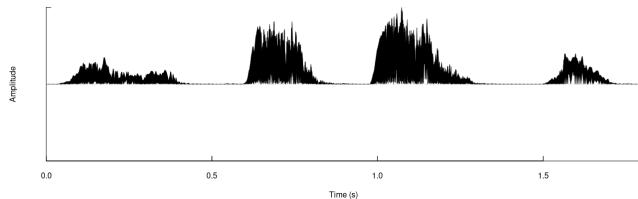
La envolvente calculada con la transformada de Hilbert es más apropiada, pero el proceso es más costoso computacionalmente.

```
# importar biblioteca seewave  
library(seewave)
```

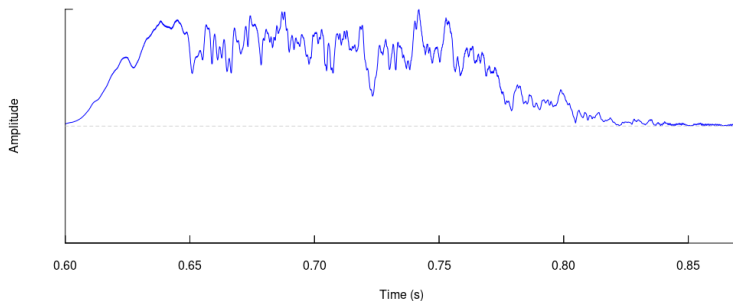
```
# cargar Wave Object tico  
data(tico)
```

```
# envolvente valor absoluto  
env(tico, envt="abs")
```

```
# envolvente con Hilbert  
env(tico, envt="hil")
```



En algunos casos puede ser necesario suavizar (o filtrar) la envolvente de amplitud.



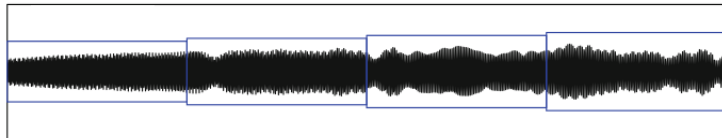
Detalle de la envolvente de amplitud de la segunda nota de la señal tico.

¿Preguntas?

1. Forma de onda y oscilograma
2. Envolvente de amplitud
- 3. Análisis con ventanas deslizantes**
4. Segmentación automática
5. Análisis de modulación de amplitud

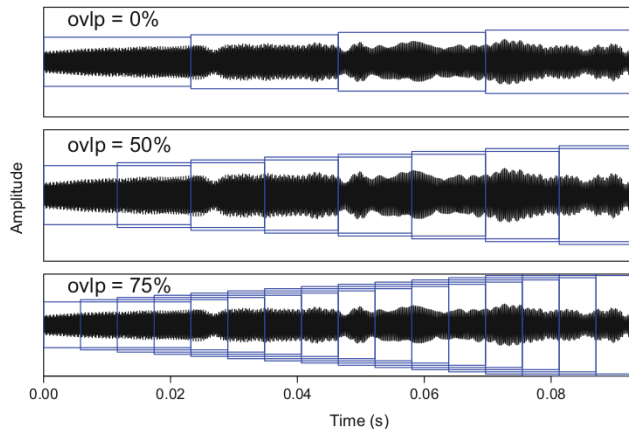
Consiste en segmentar la señal $s[n]$ con una ventana temporal $w[n]$ de la siguiente forma.

- Ventana de largo fijo dado por un número de muestras N o por una duración en segundos.
- La ventana se coloca primero al comienzo de la señal de la muestra 1 a la muestra N .
- Luego se desliza hacia adelante y se ubica de la muestra $N + 1$ a la muestra $2N$.
- El deslizamiento se repite hasta que la ventana llega al final de la señal $s[n]$.
- En cada paso se puede procesar (e.g. promediar) el tramo seleccionado y guardar su valor.
- La señal de salida es una nueva serie temporal de menor longitud que la señal de entrada.



Se reduce la resolución temporal respecto a la señal de audio.

- Acortar la ventana aumenta la resolución temporal, pero usa menos muestras para procesar (e.g. promedio menos muestras).
- Otra alternativa es usar ventanas solapadas en el tiempo, lo que aumenta la resolución temporal.

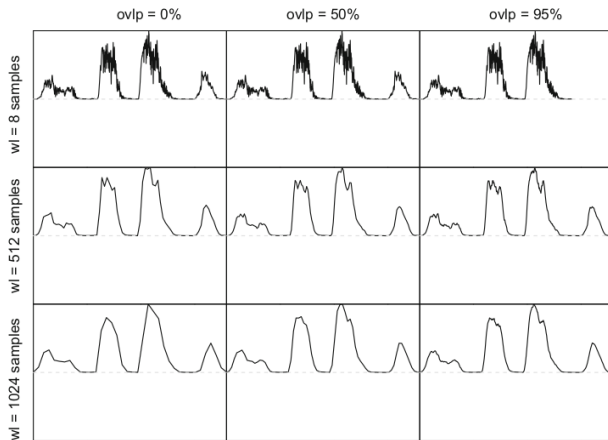


Media móvil: promedio de las muestras en ventanas deslizantes.

```
# importar biblioteca seewave  
library(seewave)
```

```
# cargar Wave Object tico  
data(tico)
```

```
# suavizado con media móvil  
env(tico, msmooth=c(512,50))
```

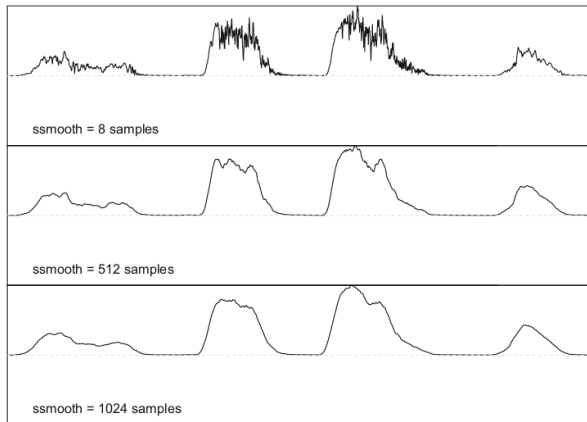


Suma móvil: suma de las muestras en ventanas deslizantes saltando de a una muestra (máximo solapamiento).

```
# importar biblioteca seewave  
library(seewave)
```

```
# cargar Wave Object tico  
data(tico)
```

```
# saltando de a una muestra  
env(tico, ssmooth=512)
```

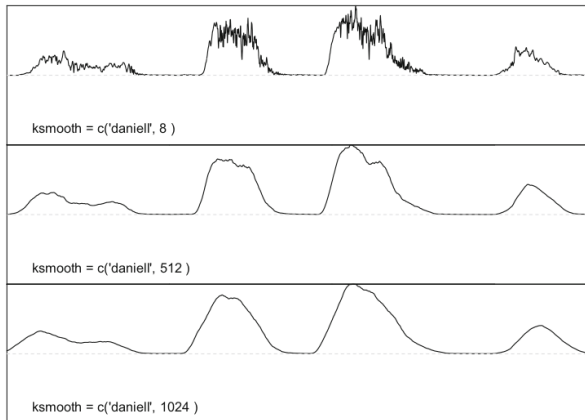


Kernel móvil: función de pesos (kernel) para promediar las muestras en ventanas deslizantes.

```
# importar biblioteca seewave  
library(seewave)
```

```
# cargar Wave Object tico  
data(tico)
```

```
# suavizado con kernel  
env(tico,  
    ksmooth=kernel("daniell",8))
```



¿Preguntas?

1. Forma de onda y oscilograma
2. Envolvente de amplitud
3. Análisis con ventanas deslizantes
- 4. Segmentación automática**
5. Análisis de modulación de amplitud

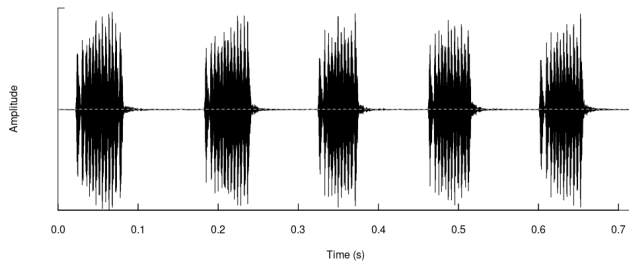
Ejemplo de *Cicada orni*.

```
# importar biblioteca seewave  
library(seewave)
```

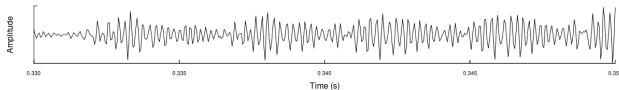
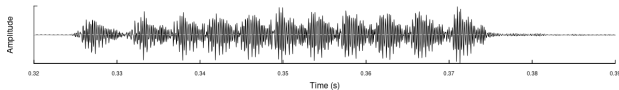
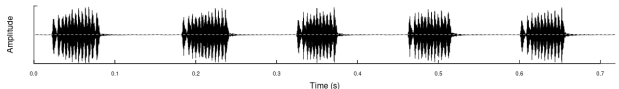
```
# cargar Wave Object orni  
data(orni)  
orni
```

```
# graficar oscilograma  
oscillo(orni)
```

```
# escribir archivo de audio  
writeWave(object=orni,  
          filename="orni.wav")
```



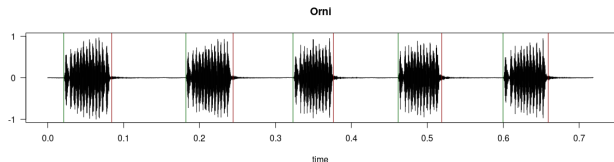
```
# importar biblioteca seewave  
library(seewave)  
  
# cargar Wave Object tico  
data(orni)  
orni  
  
# 3 líneas  
layout(matrix(1:3, nrow=3))  
  
# forma de onda completa  
oscillo(orni)  
  
# zoom 1  
oscillo(orni, from=0.32, to=0.39)  
  
# zoom 2  
oscillo(orni, from=0.33, to=0.35)
```



```
# leer anotaciones manuales
marcas <- read.csv(file = 'orni.
                  sep = "\t")

# leer inicios y finales
inicios <- marcas$Begin.Time..s.
finales <- marcas$End.Time..s.

# graficar forma de onda y anota
layout(matrix(1:2, nrow=2))
plot(normalize(orni), main='Orni
par(new=TRUE)
for (t in inicios)
{
  abline(v = t, col = "darkgreen
}
for (t in finales)
{
  abline(v = t, col = "darkred")
}
```



Una solución clásica y simple para detectar eventos en la señal es analizar las variaciones de amplitud a lo largo del tiempo en referencia a un cierto umbral de amplitud.

Para eso es posible usar la función `timer` del paquete `seewave`.

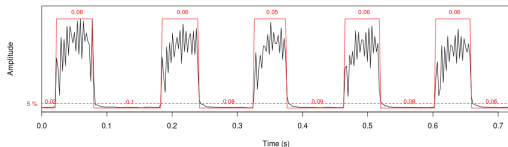
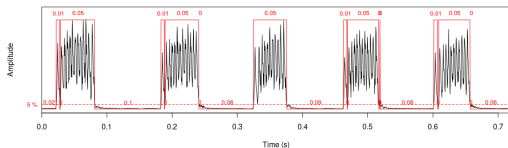
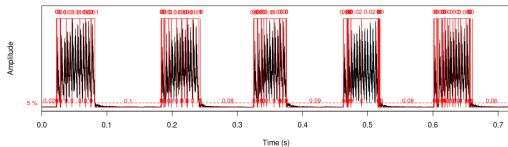
Argumentos de entrada:

`input` : wave, f
`envelope properties` :
envt, power, msmooth,
ksmooth, ssmooth, tlim,
`detection thresholds` :
threshold, dmin,
`graphical options` : plot,
plotthreshold, col,
colval, xlab, ylab.

Valores de salida:

`s` duration of signal event(s) in seconds,
`p` duration of pause event(s) in seconds,
`r` ratio between the signal and silence
events(s),
`s.start` start position(s) of signal event(s),
`s.end` end position(s) of signal event(s),
`first` whether the first event detected is a
signal or a pause.

```
# importar biblioteca seewave  
library(seewave)  
  
# cargar Wave Object orni  
data(orni)  
  
# segmentación automática  
res <- timer(orni,  
  threshold=5,  
  envt="hil")  
# msmooth=c(10,0)  
# msmooth=c(50,0)  
  
length(res$s)
```



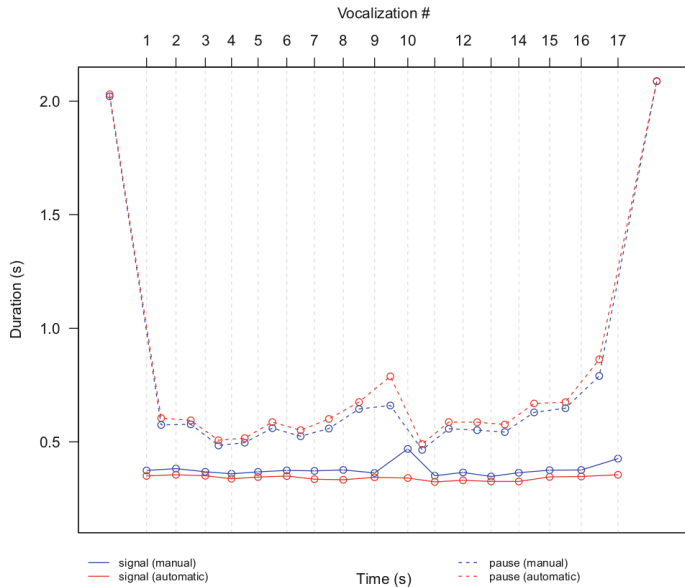
Actividad

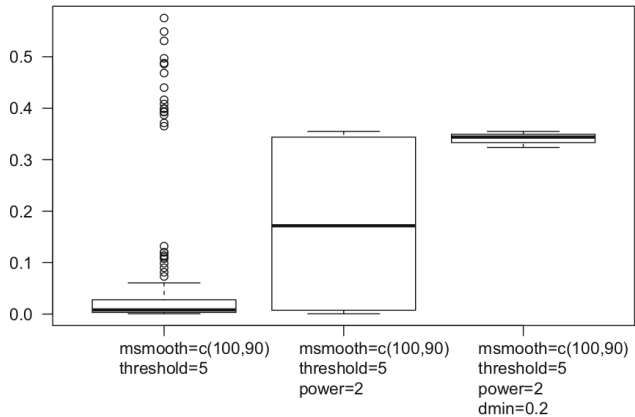
¿Que ocurre si se usa mucho más suavizado?

¿Cómo podríamos introducir información a-priori de la duración de las notas?

Calcular media y desviación estándar de las duraciones y las pausas.

Comparar estos resultados para la detección automática y la segmentación manual.

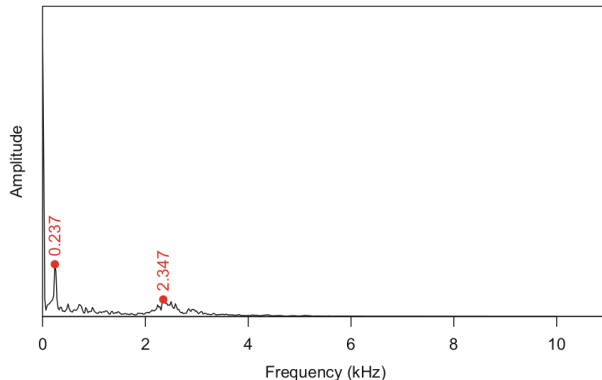




1. Forma de onda y oscilograma
2. Envolvente de amplitud
3. Análisis con ventanas deslizantes
4. Segmentación automática
- 5. Análisis de modulación de amplitud**

Es posible analizar modulaciones de amplitud a partir de la envolvente de amplitud.

```
# importar biblioteca seewave  
library(seewave)  
  
# cargar Wave Object orni  
data(orni)  
  
# modulación de amplitud  
ama(orni, wl=1024)
```



Actividad

¿A qué corresponde el pico de frecuencia inferior de la gráfica anterior?

¿A qué corresponde el pico de frecuencia superior de la gráfica anterior?

¿Preguntas?



J. Sueur, *Sound analysis and synthesis with R*.
Springer, 2018.