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

UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Julio 30, 2021

Agenda

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1. Transformada de Fourier de Tiempo Corto
2. Seguimiento de frecuencia
3. Filtros de selección de frecuencias
4. Estimación y seguimiento de formantes

Agenda

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

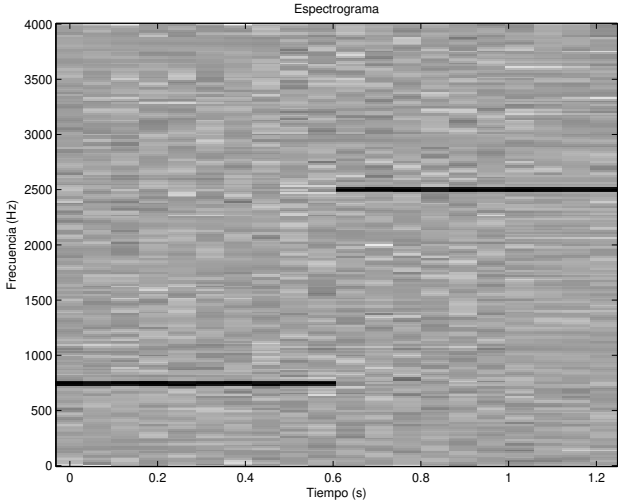
1. **Transformada de Fourier de Tiempo Corto**
2. Seguimiento de frecuencia
3. Filtros de selección de frecuencias
4. Estimación y seguimiento de formantes

Transformada de Fourier de Tiempo Corto

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Consiste en apilar transformadas de Fourier discretas (DFT) de ventanas sucesivas de pocas muestras.

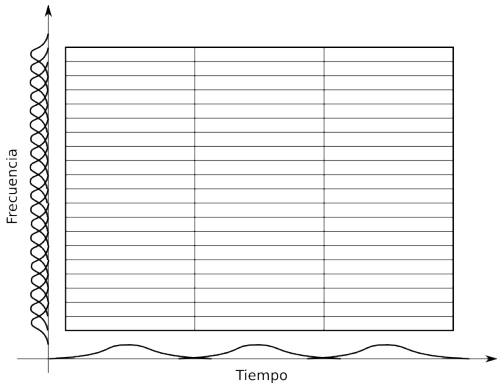
Espectrograma: es el módulo de la STFT (se descarta la fase).



Transformada de Fourier de Tiempo Corto

El largo de la ventana N determina el compromiso de resolución en el tiempo y frecuencia ($\Delta f = \frac{f_s}{N}$).

- Ventana larga: buena resolución en frecuencia pero baja resolución en el tiempo.
- Ventana corta: buena resolución en tiempo pero baja resolución en frecuencia.

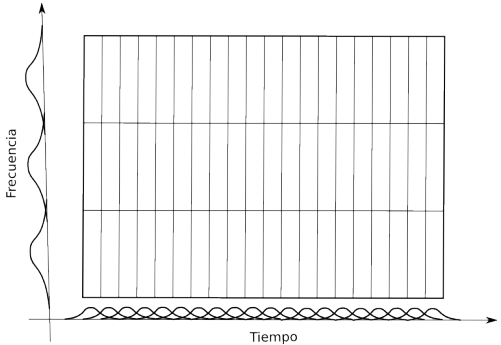


Para aumentar la resolución temporal manteniendo buena resolución en frecuencia se suele considerar cierto solapamiento entre ventanas.

Transformada de Fourier de Tiempo Corto

El largo de la ventana N determina el compromiso de resolución en el tiempo y frecuencia ($\Delta f = \frac{f_s}{N}$).

- Ventana larga: buena resolución en frecuencia pero baja resolución en el tiempo.
- Ventana corta: buena resolución en tiempo pero baja resolución en frecuencia.



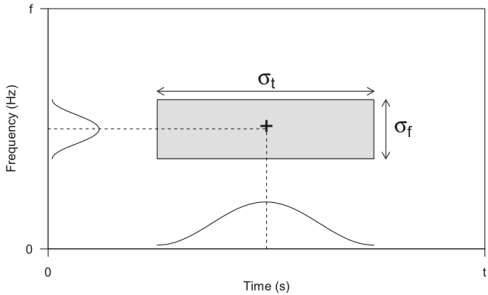
Para aumentar la resolución temporal manteniendo buena resolución en frecuencia se suele considerar cierto solapamiento entre ventanas.

Transformada de Fourier de Tiempo Corto

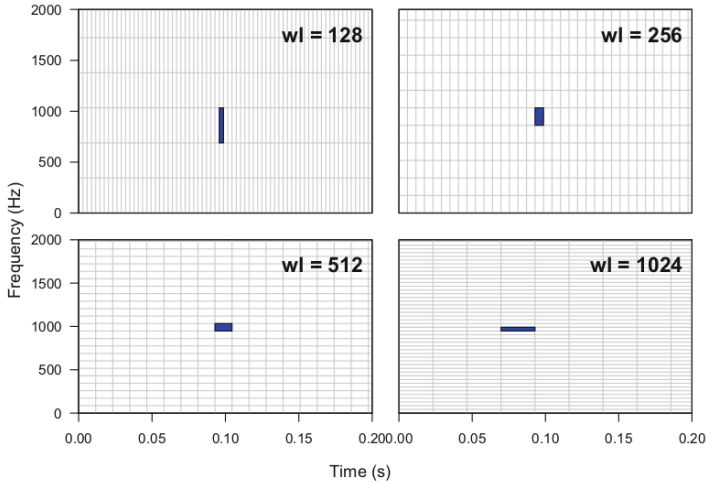
Compromiso de resolución tiempo-frecuencia.

- Principio de incertidumbre (Heisenberg).
- No se puede tener simultáneamente alta resolución en tiempo y en frecuencia.

$$\Delta t = \frac{1}{\Delta f} = \frac{\sigma_t}{f_s}$$

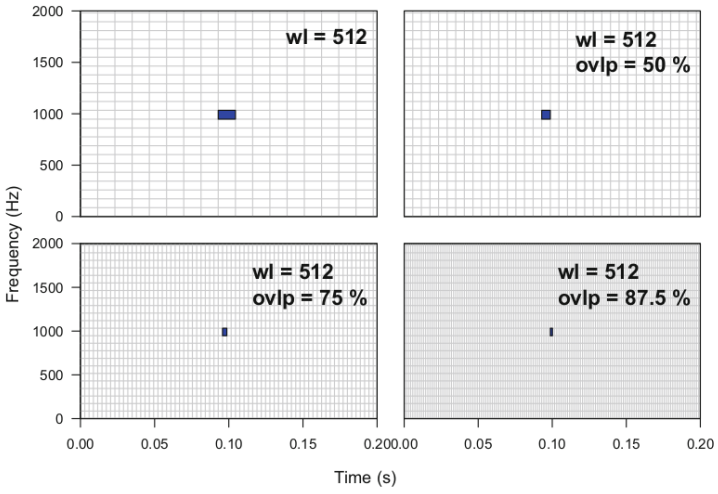


Transformada de Fourier de Tiempo Corto



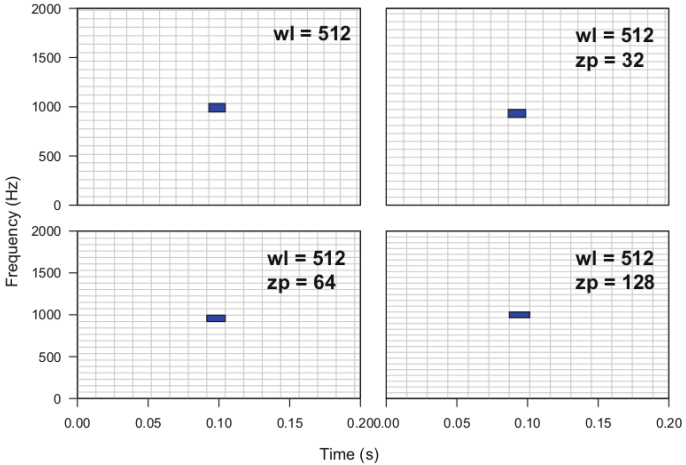
Compromiso resolución tiempo-frecuencia variando largo de ventana.

Transformada de Fourier de Tiempo Corto



Incremento de resolución temporal usando solapamiento entre ventanas.

Transformada de Fourier de Tiempo Corto

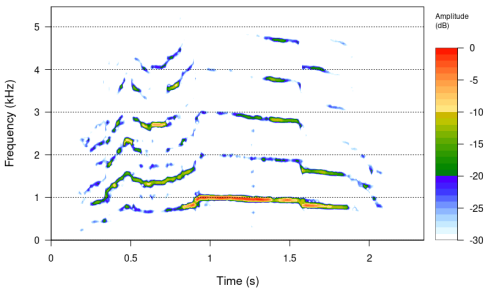


Incremento de resolución frecuencial usando relleno de ceros.

Transformada de Fourier de Tiempo Corto

Cálculo de la STFT usando la función `spectro`.

```
library(tuneR)
library(seewave)

cervus <- readWave("cervus-elaphus.wav")

# cálculo de la STFT (espectrogrma)
fspec <- spectro(cervus,
                 wl=256,
                 wn="hanning",
                 ovlp=50)
```



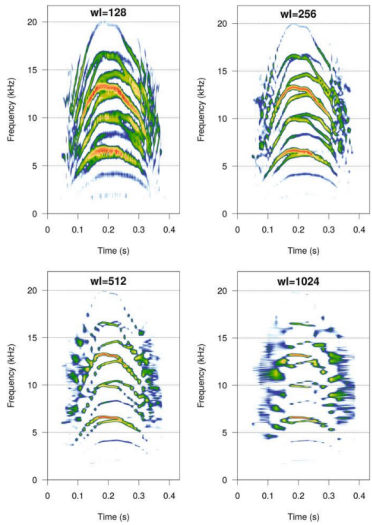
Transformada de Fourier de Tiempo Corto

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Argumentos de `spectro`

- wl: largo de ventana en muestras
- wn: tipo de ventana (hanning)
- ovlp: porcentaje de solapamiento
- zp: cantidad de relleno de ceros

Los valores adecuados de los parámetros dependen de la señal analizada.



Agenda

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

1. Transformada de Fourier de Tiempo Corto

2. **Seguimiento de frecuencia**

3. Filtros de selección de frecuencias

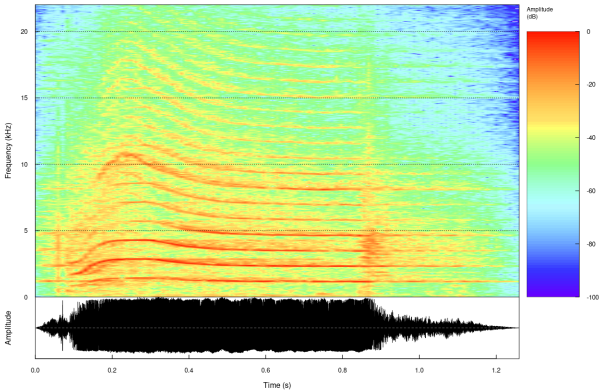
4. Estimación y seguimiento de formantes

Seguimiento de frecuencia

La STFT nos permite analizar cambios de frecuencia a lo largo del tiempo.

Esto consiste típicamente en:

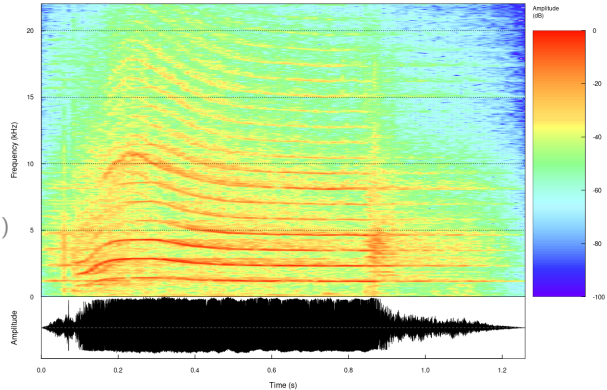
1. Calcular la STFT (espectrograma).
2. Procesar cada trama (e.g. obtener el máximo en frecuencia).
3. Integrar información temporalmente.



Seguimiento de frecuencia

Determinar parámetros adecuados para el cálculo de la STFT (wl, wn, ovl, zp).

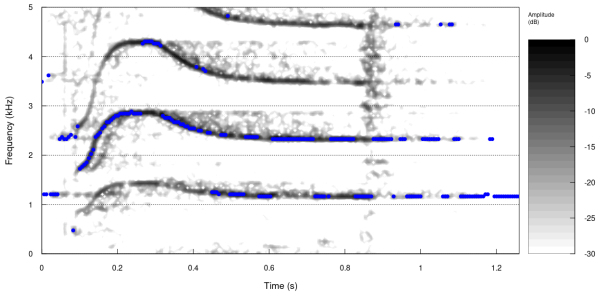
```
library(tuneR)
library(seewave)

orca <- readWave("orca2.wav")
# wav a 44100 Hz
orca

# cálculo de la STFT (espectrogrma)
fspec <- spectro(orca,
                 osc=TRUE,
                 wl=1024,
                 wn="hanning",
                 ovlp=50,
                 zp=1024,
                 collevels=seq(-100, 0, 1),
                 palette=temp.colors)
```



Seguimiento de frecuencia

Frecuencia dominante usando `dfreq`.

```
library(tuneR)
library(seewave)

orca <- readWave("orca2.wav")

# frecuencia dominante
wl <- 1024 ; ovlp <- 75
df1 <- dfreq(orca, ovlp=ovlp,
             wl=wl, plot=FALSE)


# gráfico de espectrograma
spectro(orca, wl=wl, ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE, flim=c(0,5))
# gráfico de frecuencia dominante
points(df1, col='blue', pch=19)
```

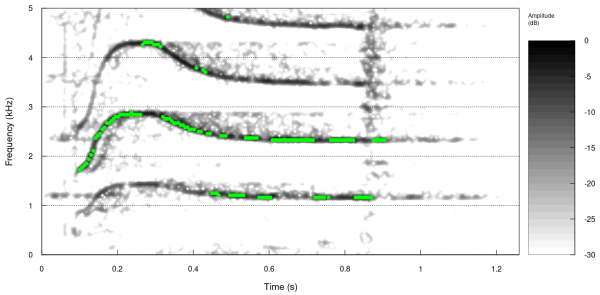


Seguimiento de frecuencia

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Frecuencia dominante usando `dfreq`.

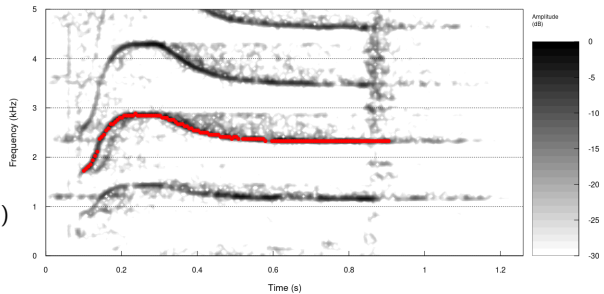
```
library(tuneR)
library(seewave)

orca <- readWave("orca2.wav")

# frecuencia dominante
wl <- 1024 ; ovlp <- 75
df2 <- dfreq(orca, ovlp=ovlp,
             wl=wl, plot=FALSE,
             clip=0.3)

# gráfico de espectrograma
spectro(orca, wl=wl, ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE, flim=c(0,5))
# gráfico de frequencia dominante
points(df2, col='green', pch=19)
```



Seguimiento de frecuencia

Frecuencia dominante usando `dfreq`.

```
library(tuneR)
library(seewave)

orca <- readWave("orca2.wav")

# frecuencia dominante
wl <- 1024 ; ovlp <- 75
df3 <- dfreq(orca, ovlp=ovlp,
             wl=wl, plot=FALSE,
             bandpass=c(1500, 3000)
             clip=0.3)

# gráfico de espectrograma
spectro(orca, wl=wl, ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE, flim=c(0,5))
# gráfico de frequencia dominante
points(df3, col='red', pch=19)
```

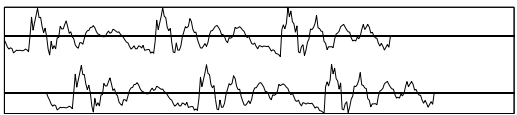


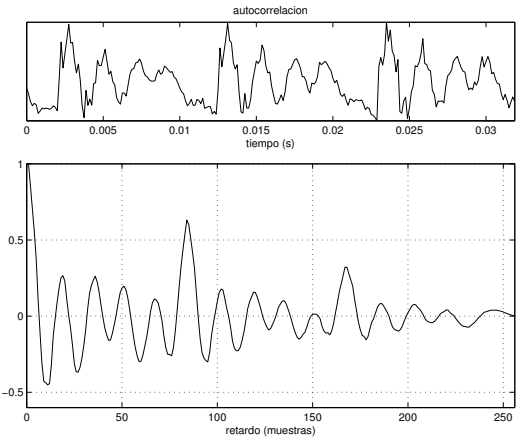
Seguimiento de frecuencia

Estimación de frecuencia fundamental usando autocorrelación.

- Para estimar el período comparo un tramo de señal con una versión retrasada.
- La correlación es máxima cuando el retraso coincide con el período.

$$R[k] = \sum_{m=0}^{M} x[m]x[m+k]$$



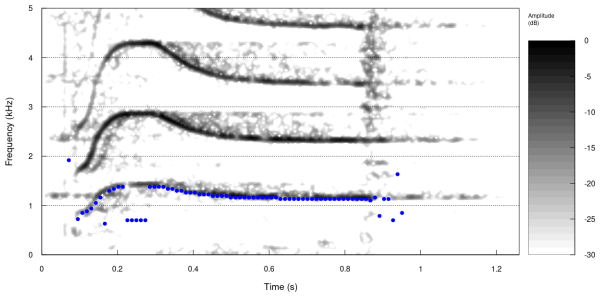


Seguimiento de frecuencia

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Frecuencia fundamental usando `autoc`.

```
library(tuneR)
library(seewave)

orca <- readWave("orca2.wav")

# frecuencia fundamental
ff1 <- autoc(orca,
             fmin=500,
             fmax=3000,
             threshold=50,
             plot=FALSE)

# gráfico de espectrograma
spectro(orca, wl=wl, ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE, flim=c(0,5))
# gráfico de frequencia fundamental
points(ff1, col='blue', pch=19)
```

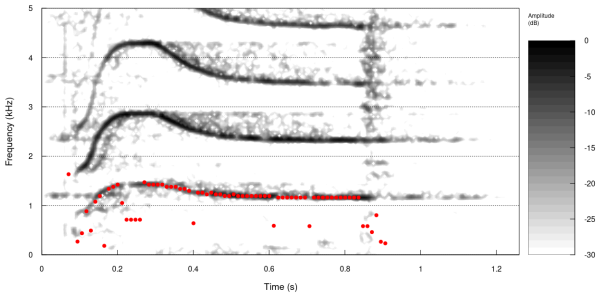


Seguimiento de frecuencia

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Frecuencia fundamental usando `fund`.

```
library(tuneR)
library(seewave)

orca <- readWave("orca2.wav")

# frecuencia fundamental
ff2 <- fund(orca,
            fmax=2000,
            threshold=65,
            plot=FALSE)

# gráfico de espectrograma
spectro(orca, wl=wl, ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE, flim=c(0,5))
# gráfico de frecuencia fundamental
points(ff2, col='red', pch=19)
```



¿Preguntas?

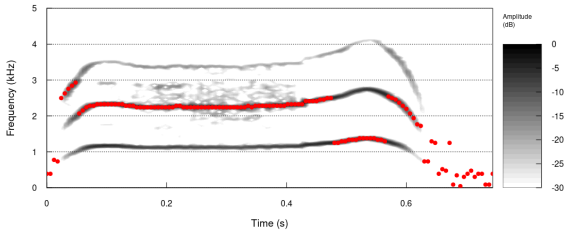
Actividad

1) Calcule el espectrograma para `peewit4.wav` seleccionando parámetros adecuados.
2) Estime la frecuencia dominante usando `dfreq` con parámetros por defecto.
3) Ajuste los parámetros para estimar la frecuencia dominante evitando valores espúreos.
4) Estime la frecuencia fundamental usando `autoc` y `fund` con parámetros por defecto.
5) Ajuste los parámetros para mejorar la estimación de frecuencia fundamental.

Seguimiento de frecuencia

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

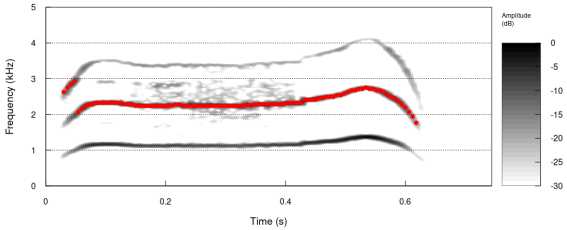
```
library(tuneR)
library(seewave)

peewit <- readWave("peewit4.wav")

# frecuencia dominante
wl <- 1024 ; ovlp <- 75
df <- dfreq(peewit, ovlp=ovlp,
            wl=wl, plot=FALSE)

# plot spectrogram
spectro(peewit,
        wl=wl,
        ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE,
        flim=c(0,5))
# plot dominant frequency
points(df, col='red', pch=19)
```



Seguimiento de frecuencia

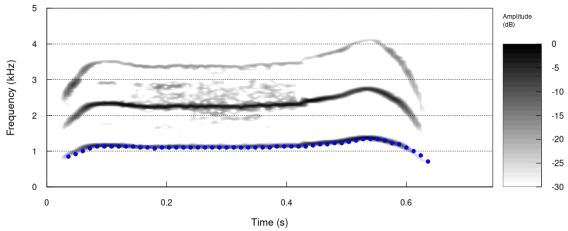
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

```
library(tuneR)
library(seewave)

peewit <- readWave("peewit4.wav")

# frecuencia dominante
wl <- 1024 ; ovlp <- 75
df <- dfreq(peewit, ovlp=ovlp,
            wl=wl, plot=FALSE,
            bandpass=c(1500, 3000),
            clip=0.1)

# plot spectrogram
spectro(peewit,
        wl=wl,
        ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE,
        flim=c(0,5))
# plot dominant frequency
points(df, col='red', pch=19)
```



Seguimiento de frecuencia

```
library(tuneR)
library(seewave)

peewit <- readWave("peewit4.wav")

# frecuencia fundamental
ff1 <- autoc(peewit,
             fmin=50,
             fmax=3000,
             threshold=5,
             plot=FALSE)

# gráfico de espectrograma
spectro(peewit, wl=wl, ovlp=ovlp,
        palette=reverse.gray.colors.2,
        norm=TRUE, flim=c(0,5))
# gráfico de frequencia fundamental
points(ff1, col='blue', pch=19)
```



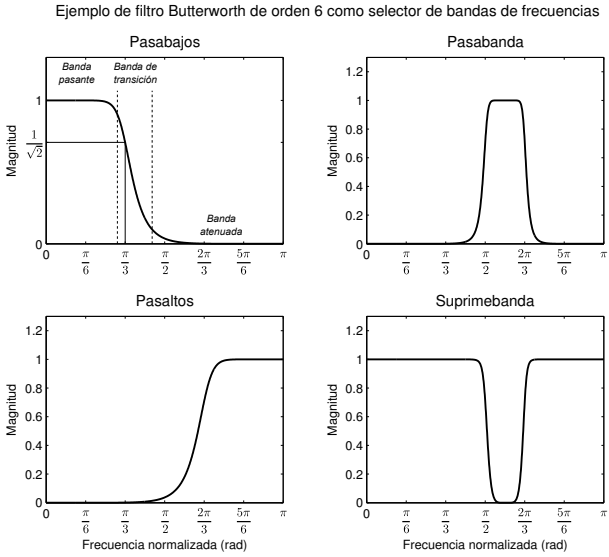
Agenda

Filtros de selección de frecuencias

El objetivo es permitir pasar inalterada cierta banda de frecuencias y bloquear completamente el resto.

Hay cuatro tipos básicos:

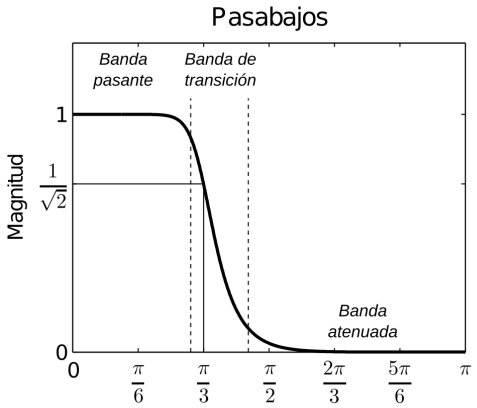
- pasa-bajos,
- pasa-altos,
- pasa-banda,
- y suprime-banda.



Filtros de selección de frecuencias

Clasificación de regiones de filtros selectores

- Banda pasante: Rango de frecuencias que el filtro permite pasar sin alterar.
- Banda atenuada: Rango de frecuencias que el filtro bloquea.
- Banda de transición: Región entre la banda pasante y la banda atenuada.
- Frecuencia de corte: Frecuencia entre la banda pasante y la banda de transición.

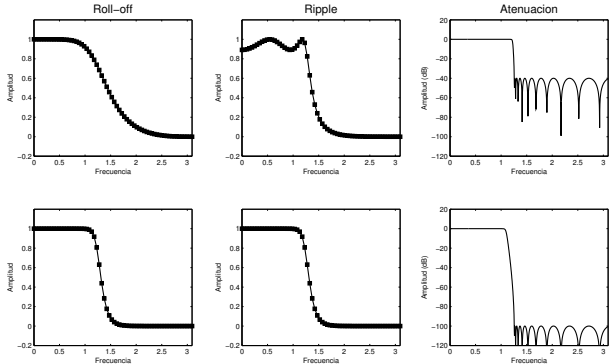


Filtros de selección de frecuencias

Parámetros que miden la calidad del filtro como selector de frecuencias.

- Roll-off: Es el ancho de la banda de transición. Un filtro de roll-off rápido significa que la banda de transición es angosta.

  Para separar componentes de frecuencia cercanos, el roll-off debe ser rápido.

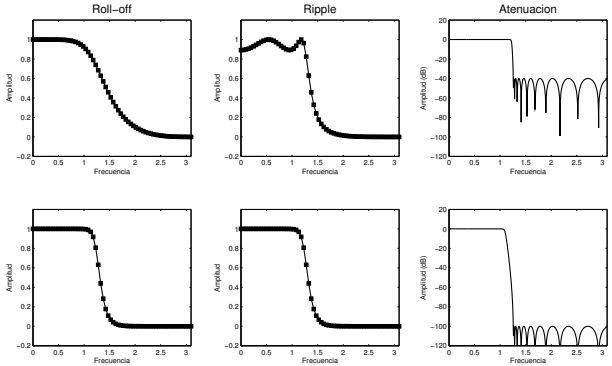


Filtros de selección de frecuencias

Parámetros que miden la calidad del filtro como selector de frecuencias.

- Ripple en la banda pasante: Amplitud de las oscilaciones en la banda pasante de la magnitud de la respuesta en frecuencia.

  Para no alterar la magnitud de los componentes espectrales de la banda pasante, el filtro no debe tener ripple.

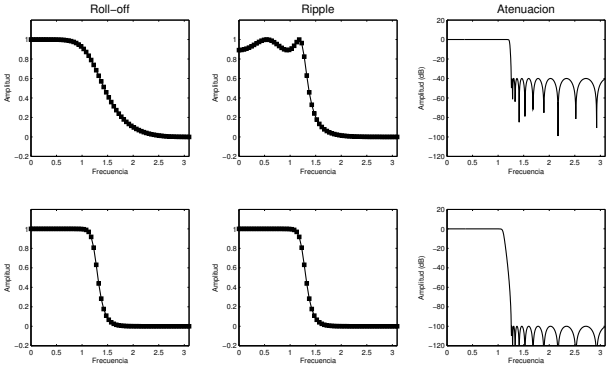


Filtros de selección de frecuencias

Parámetros que miden la calidad del filtro como selector de frecuencias.

- Atenuación en la banda atenuada: Cuánto son atenuados los componentes en la banda atenuada.

  Es deseable buena atenuación en la banda atenuada para eliminar los componentes espectrales en esa región.



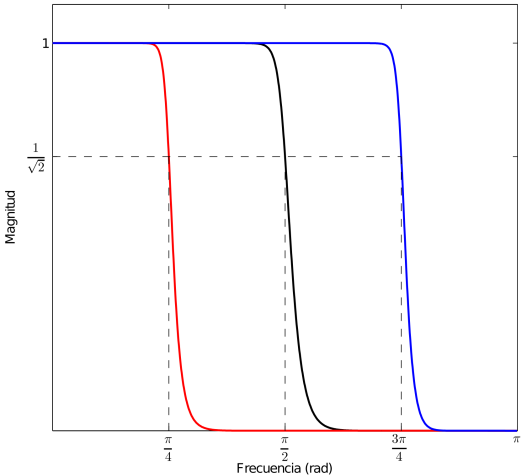
Filtros de selección de frecuencias

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Filtros Butterworth

Suelen ser la primera elección para implementar un filtro de selección de frecuencias de buen desempeño.

- Sin ripple en la banda pasante.
- Hay que especificar
  - el tipo (e.g. pasa-bajos)
  - la frecuencia de corte
  - y el orden.
- Aumentar el orden mejora el roll-off.



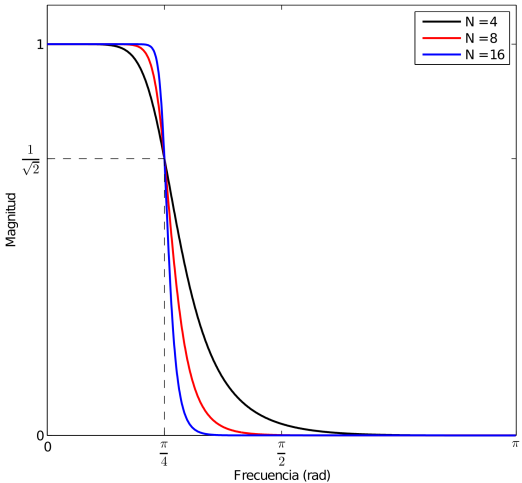
Filtros de selección de frecuencias

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Filtros Butterworth

Suelen ser la primera elección para implementar un filtro de selección de frecuencias de buen desempeño.

- Sin ripple en la banda pasante.
- Hay que especificar
  - el tipo (e.g. pasa-bajos)
  - la frecuencia de corte
  - y el orden.
- Aumentar el orden mejora el roll-off.



Filtros de selección de frecuencias

Filtros Butterworth con la función `bwfilter` del paquete seewave.

Los principales argumentos son:

- n: orden del filtro
- from: frecuencia de corte más baja (Hz)
- to: frecuencia de corte más alta (Hz)
- bandpass: si es pasa-banda o suprime-banda

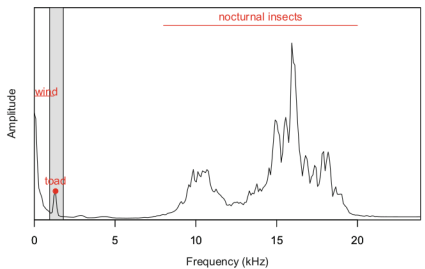
```
library(tuneR)
library(seewave)

ruido <- readWave("ruido2.wav")

# low-pass with a 5000 Hz cutoff frequency
res <- bwfilter(ruido, n=3, to=5000,
                output="Wave")

# high-pass with a 100 Hz cutoff frequency
res <- bwfilter(ruido, n=3, from=100,
                output="Wave")

# band-pass between 3000 and 5000 Hz
res <- bwfilter(ruido, n=3, from=3000, to=5000,
                output="Wave")

# band-stop between 3000 and 5000 Hz
res <- bwfilter(ruido, n=3, from=3000, to=5000,
                bandpass=FALSE, output="Wave")
```

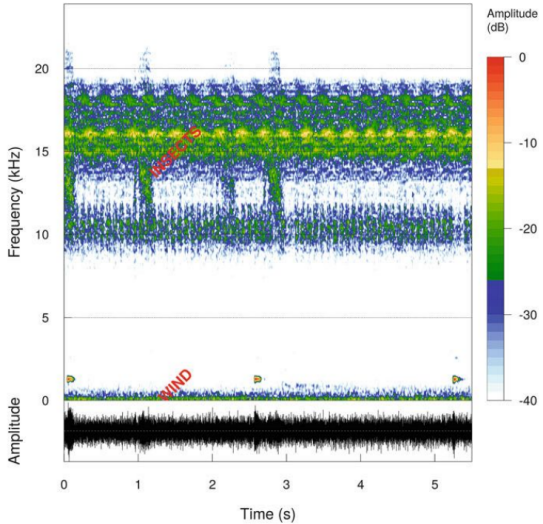
Filtros de selección de frecuencias

Filtro pasa-banda para mejorar la detección automática de vocalizaciones.

En esta grabación hay:

- viento en muy bajas frecuencias
- insectos en las altas frecuencias
- vocalizaciones de un sapo (toad) en la banda entre 0.95 y 1.8 kHz

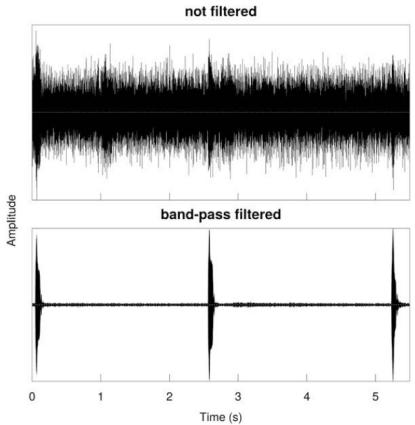




Filtros de selección de frecuencias

Con un filtro pasa-banda entre 0.95 y 1.8 kHz se puede seleccionar la señal de interés y hacer la detección automática.

```
library(tuneR)
library(seewave)

toad <- readWave("Alytes_obstetricans.wav")
toad

toad.filt <- bwfilter(toad,
                      from=950,
                      to=1800,
                      output="Wave")

res <- timer(toad.filt, envt="hil",
             threshold=4, ssmooth=400)
```



¿Preguntas?

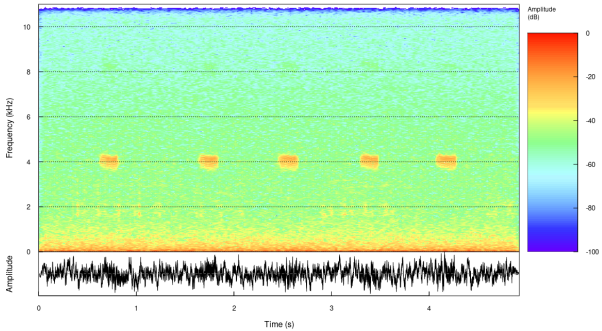
Actividad

1) Calcular espectrograma de la señal `ruido2.wav`.
2) Evaluar la posibilidad de una detección automática.
3) Determinar rango de frecuencia de la señal de interés.
4) Aplicar un filtro de selección de frecuencias.
5) Realizar la detección automática sobre la señal filtrada.
6) ¿Es adecuada? ¿Qué limitantes tiene el enfoque?

Filtros de selección de frecuencias

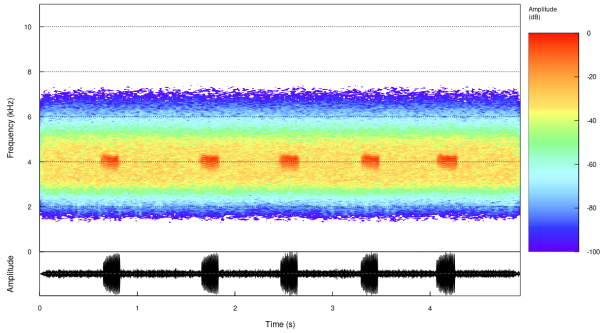
Filtro pasa-banda para mejorar la detección automática de vocalizaciones.

```
library(tuneR)
library(seewave)

ruido <- readWave("ruido2.wav")
# wav a 22050 Hz
ruido

# cálculo de la STFT (espectrogrma)
fspec <- spectro(ruido, osc=TRUE,
                 wl=1024, ovlp=50,
                 collevels=seq(-100, 0, 1),
                 palette=temp.colors)
```



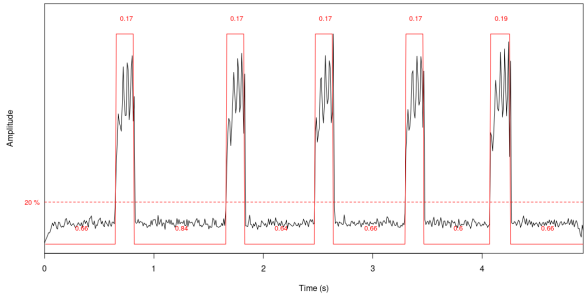
Filtros de selección de frecuencias

```
library(tuneR)
library(seewave)

ruido <- readWave("ruido2.wav")
# wav a 22050 Hz
ruido

# band-pass entre 3000 and 5000 Hz
out <- bwfilter(ruido, n=3,
                from=3000,
                to=5000,
                output="Wave")

# cálculo de la STFT (espectrogrma)
fspec <- spectro(out, osc=TRUE,
                 wl=1024, ovlp=50,
                 collevels=seq(-100, 0, 1),
                 palette=temp.colors)
```



Filtros de selección de frecuencias

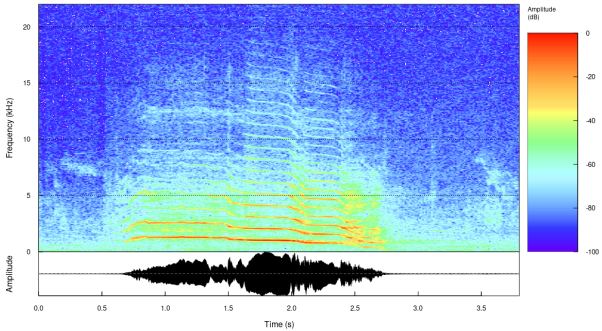
```
library(tuneR)
library(seewave)

ruido <- readWave("ruido2.wav")
# wav a 22050 Hz
ruido

# band-pass entre 3000 and 5000 Hz
out <- bwfilter(ruido, n=3,
                from=3000,
                to=5000,
                output="Wave")

res <- timer(out,
             threshold=20,
             msmooth=c(200,0))
```



Filtros de selección de frecuencias

Filtro pasa-banda para mejorar el seguimiento de frecuencia.

```
library(tuneR)
library(seewave)

cervus <- readWave("cervus7.wav")
# wav a 44100 Hz
cervus

# cálculo de la STFT (espectrogrma)
fspec <- spectro(cervus, osc=TRUE,
                 wl=1024, ovlp=50,
                 collevels=seq(-100, 0, 1),
                 palette=temp.colors)
```

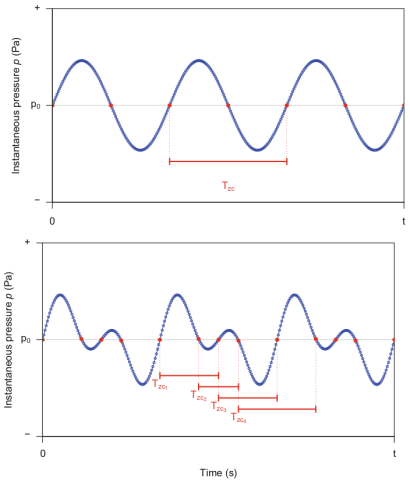


Filtros de selección de frecuencias

La tasa de cruces por cero (ZCR) es una forma muy simple de estimar la frecuencia instantánea de una señal.

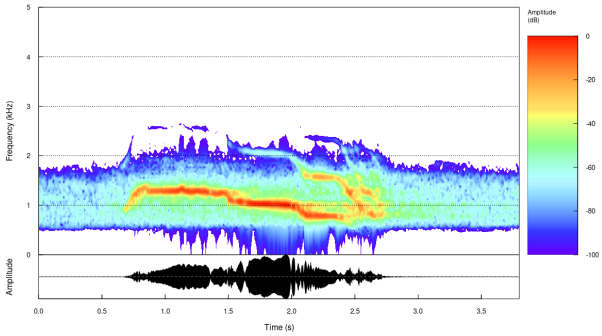
$$\text{zcr} = \frac{1}{2N} \sum_{n=0}^{N} |\text{sign}(s[n+1]) - \text{sign}(s[n])|$$

- Se cuentan la cantidad de cruces por cero por unidad de tiempo.
- Pero funciona solo para señales mono componentes.
- Se puede filtrar la señal para conservar un único componente.



Filtros de selección de frecuencias

```r
library(tuneR)
library(seewave)

cervus <- readWave("cervus7.wav")
# wav a 44100 Hz
cervus

# band-pass entre 650 and 1500 Hz
out <- bwfilter(cervus, n=5,
                from=650,
                to=1500,
                output="Wave")
```

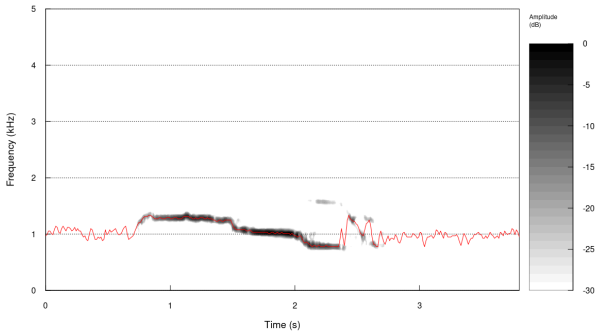



```r
# cálculo de la STFT (espectrogrma)
fspec <- spectro(out, osc=TRUE, flim=c(0,5),
                 wl=1024, ovlp=50,
                 collevels=seq(-100, 0, 1),
                 palette=temp.colors)
```

Filtros de selección de frecuencias

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

```
# tasa de cruces por cero
res <- zcr(out, plot=FALSE,
           ovlp=50, wl=1024)

# conversion a frecuencia (kHz)
f <- res[,2]*cervus@samp.rate/2000

# espectrograma y frecuencia
spectro(out, flim=c(0,5),
        wl=1024, ovlp=50,
        palette=reverse.gray.colors.2)
lines(res[,1], f, col="red")
```



Agenda

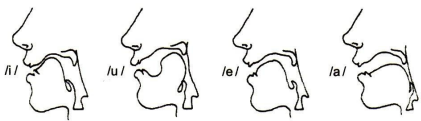
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

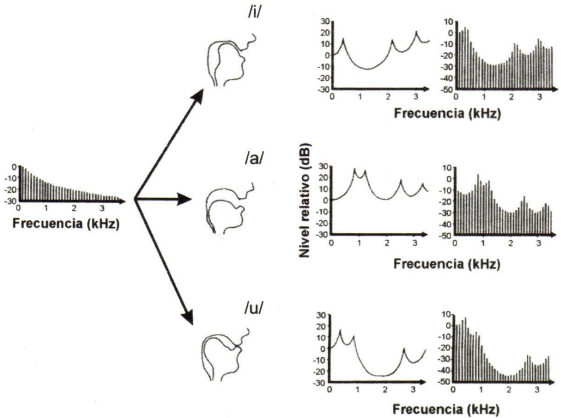
1. Transformada de Fourier de Tiempo Corto
2. Seguimiento de frecuencia
3. Filtros de selección de frecuencias
4. **Estimación y seguimiento de formantes**

Estimación y seguimiento de formantes

Mecanismo de producción de voz.

- Excitación: Señal periódica producida por la vibración de la cuerdas vocales.
- Tracto vocal: Tubos de sección no uniforme. Modifican contenido espectral de la excitación por su selectividad en frecuencia.
- Formantes: Frecuencias de resonancia del tracto vocal. Dependen de forma y dimensiones del tracto vocal, y la posición de sus partes móviles.



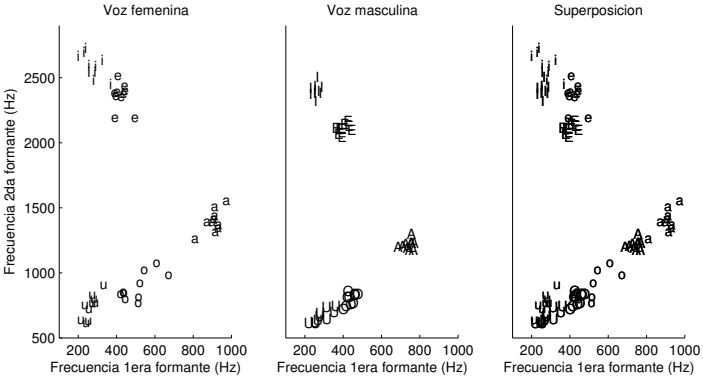


Estimación y seguimiento de formantes

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Formantes

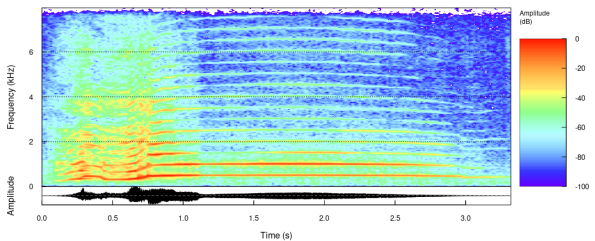
- Cada vocal corresponde a una combinación particular de formantes, y es posible distinguirlas incluso sólo a partir de las dos primeras.
- Varía entre diferentes personas y de acuerdo al género y edad.



Estimación y seguimiento de formantes

Algunos sonidos de animales también se puede caracterizar a través de sus formantes.

```
library(phonTools)
library(tuneR)
library(seewave)

cervus <- readWave("cervus2.wav")
# wav a 16000 Hz
cervus

# cálculo de la STFT
spectro(cervus, osc=TRUE,
        wl=512, ovlp=50,
        collevels=seq(-100, 0, 1),
        palette=temp.colors)
```

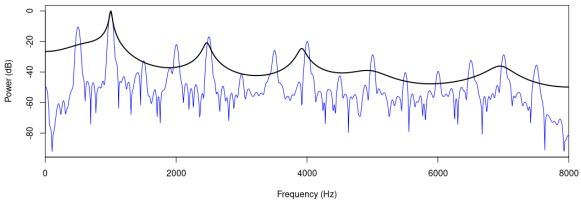


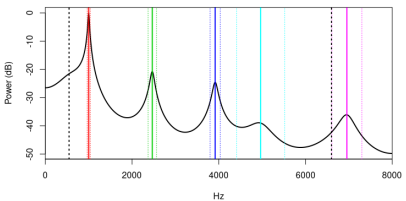
Estimación y seguimiento de formantes

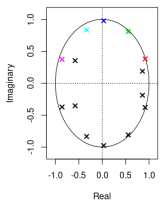
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Algunos sonidos de animales también se puede caracterizar a través de sus formantes.

```
# selección de un fragmento
sel <- cutw(cervus,
            from=1.225, to=1.275,
            output="Wave")

# cálculo de envolvente espectral
coeffs <- lpc(sel@left,
              fs=sel@samp.rate,
              order=14, show=TRUE)

# cálculo de formantes
findformants(coeffs=coeffs,
             fs=sel@samp.rate,
             showbws=TRUE)
```



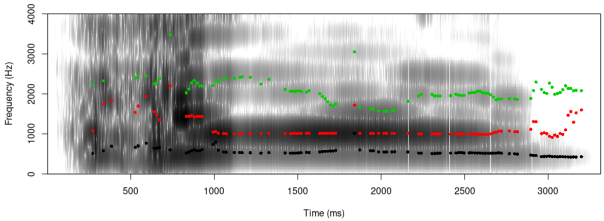




Estimación y seguimiento de formantes

Seguimiento de formantes a lo largo del tiempo.

```
library(phonTools)
library(tuneR)
library(seewave)

cervus <- readWave("cervus2.wav")
# wav a 16000 Hz
cervus

fs <- cervus@samp.rate
wl <- 512
res <- formanttrack(cervus@left, fs=fs,
                    windowlength=1000*wl/fs,
                    timestep=1000*wl/fs/2,
                    minformant=250, cutoff=4000, maxbw=800,
                    formants=3, periodicity=0.95,
                    show=TRUE, returnbw=TRUE)
```



¿Preguntas?

Referencias


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


S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*.
California Technical Publishing, 1997.