

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

Agosto 6, 2021

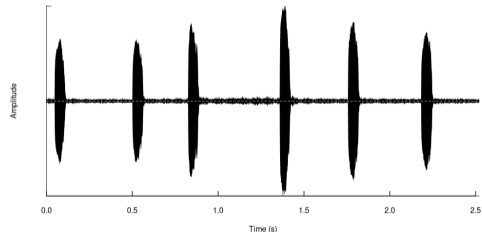
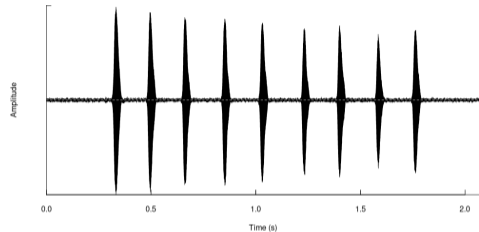


UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

La comparación de sonidos es una tarea muy habitual en **bioacústica**.

Un posible abordaje es el de utilizar **algoritmos** que comparan **series temporales**.

Entre otras cosas, pueden utilizarse para la **detección automática**.



1. **Correlación cruzada de series temporales**
2. **Deformación temporal dinámica**
3. **Correspondencia de patrones espectrales**

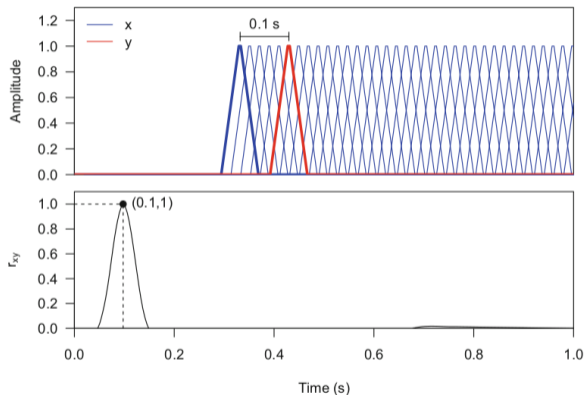
1. Correlación cruzada de series temporales

2. Deformación temporal dinámica

3. Correspondencia de patrones espectrales

La **correlación cruzada** de dos series temporales, $x[n]$ e $y[n]$, consiste en calcular una medida de similitud entre $x[n]$ y una versión retardada de $y[n]$.

Se puede emplear para identificar un sonido de referencia (plantilla) en un archivo de prueba automáticamente.

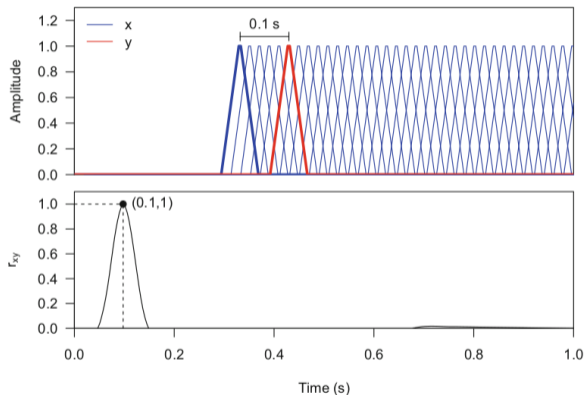


Distintas medidas de similitud pueden utilizarse como medida de correlación.

Una de las medidas más comunes es el **coeficiente de correlación normalizado**, $r_{xy}[m]$, se obtiene como:

$$r_{xy}[m] = \frac{\sum_{n=1}^{N-m} x[n]y[n+m]}{\sqrt{\sum_{n=1}^{N-m} x^2[n] \sum_{n=1}^{N-m} y^2[n]}}$$

El resultado de la **correlación cruzada** es también una serie temporal.

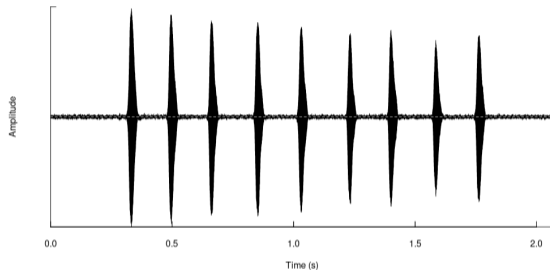


Comparación de notas de *Leptodactylus mystacinus* (rana de bigotes).

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

```
lepmys <- readWave("Lepmys.wav")  
# wav a 44100 Hz  
lepmys
```

```
# graficar oscilograma  
oscillo(lepmys)
```

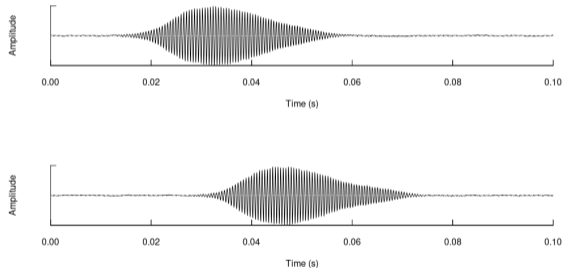


Consideramos las dos primeras notas.

```
# recortamos las dos primeras notas
note1 <- cutw(lepmys,
  from=0.30, to=0.40,
  output="Wave")

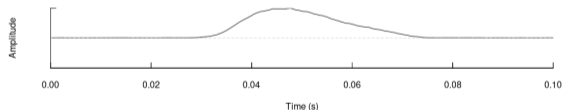
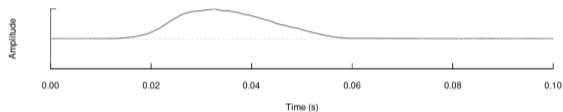
note2 <- cutw(lepmys,
  from=0.45, to=0.55,
  output="Wave")

# oscilograma de cada nota
layout(matrix(1:2, nc=1))
oscillo(note1)
oscillo(note2)
```



Calculamos sus envolventes de amplitud.

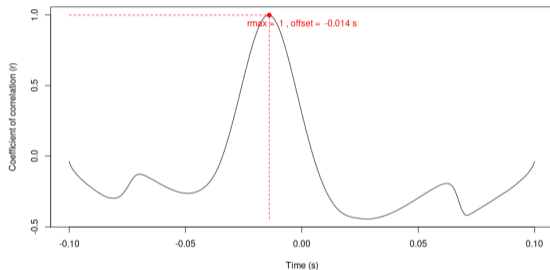
```
# envolventes de amplitud  
layout(matrix(1:2, nc=1))  
env1 <- env(note1,  
            msmooth=c(50,75))  
env2 <- env(note2,  
            msmooth=c(50,75))
```



La función `coenv` del paquete `seewave` permite calcular la correlación cruzada de envolventes de amplitud.

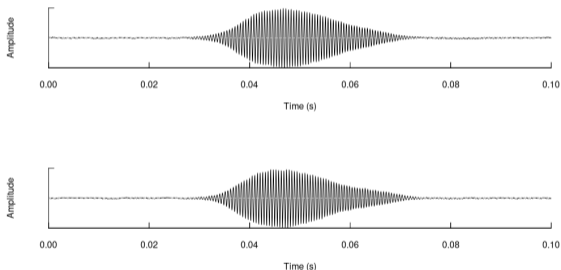
```
# correlación de envolventes  
res <- corenv(note1, note2,  
              msmooth=c(50,75),  
              method="pearson")
```

Utiliza un coeficiente de correlación estadística como medida de similitud (Spearman, Pearson o Kendall).



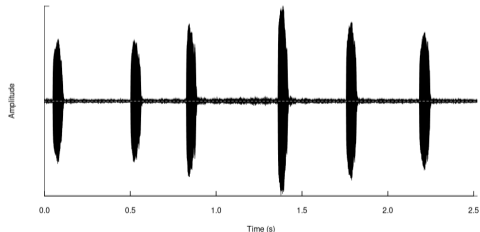
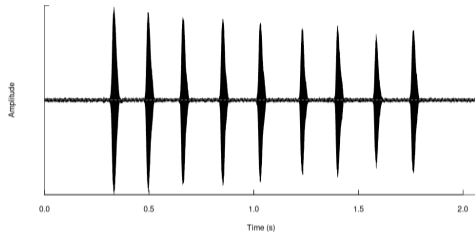
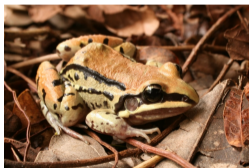
La correlación es máxima ($r_{max} = 1$) para un retraso de 14 milisegundos ($offset = -0.014$ s).

```
# alineamiento usando el  
# maximo de la correlacion  
dur <- 0.1  
note1_end <- cutw(note1,  
                  from=dur+res$t,  
                  to=dur,  
                  output="Wave")  
  
note1_ini <- cutw(note1,  
                 from=0,  
                 to=dur+res$t,  
                 output="Wave")  
  
note1_ali <- bind(note1_end,  
                 note1_ini)  
  
# comparar notas alineadas  
layout(matrix(1:2, nc=1))  
oscillo(note1_ali)  
oscillo(note2)
```



Comparación de notas de diferentes especies:
Leptodactylus mystacinus (rana de bigotes) y
Leptodactylus latinasus (rana piadora).

La envolvente temporal de sus notas es diferente. El valor de correlación nos da una medida de la similitud entre ellas.



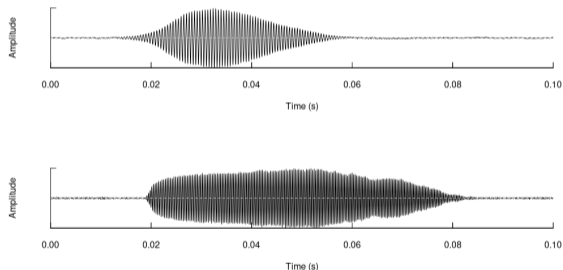
```
library(tuneR)
library(seewave)

lepmys <- readWave("Lepmys.wav")
leplat <- readWave("Leplat.wav")

# recortamos las dos primeras notas
note1 <- cutw(lepmys,
              from=0.30, to=0.40,
              output="Wave")

note2 <- cutw(leplat,
              from=0.03, to=0.13,
              output="Wave")

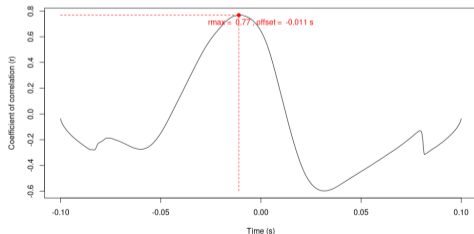
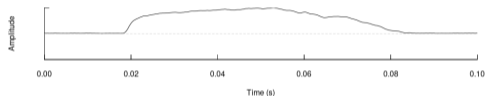
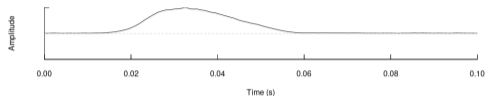
# oscilograma de cada nota
layout(matrix(1:2, nc=1))
oscillo(note1)
oscillo(note2)
```



```
# envolventes de amplitud
layout(matrix(1:2, nc=1))
env1 <- env(note1,
            msmooth=c(50,75))
env2 <- env(note2,
            msmooth=c(50,75))
```

```
# correlación de envolventes
res <- corenv(note1, note2,
              msmooth=c(50,75),
              method="pearson")
```

El valor máximo de correlación es ahora de $r_{max} = 0.77$. Esto nos podría permitir discriminar entre notas de *lepmys* y *leplat*.



¿Preguntas?

Actividad

Suponga que en una misma grabación coexisten cantos de las dos especies, *Leptodactylus mystacinus* (rana de bigotes) y *Leptodactylus latinasus* (rana piadora).

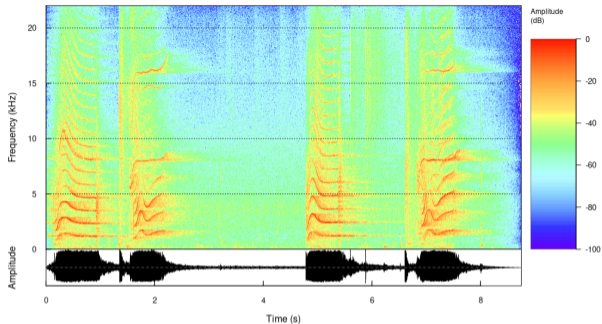
Proponga un método basado en correlación cruzada de series temporales para detectar los cantos de cada especie automáticamente. ¿Qué dificultades pueden presentarse?

El mismo procedimiento se podría aplicar a los contornos de frecuencia dominante.

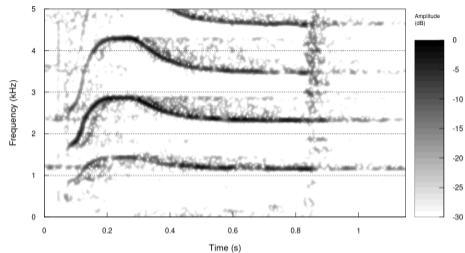
```
library(tuneR)
library(seewave)

orca <- readWave("orca.wav")

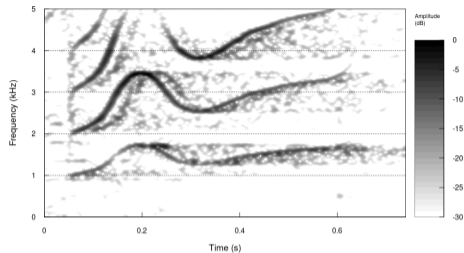
wl <- 1024
ovlp <- 87.5
# graficar oscilograma
spectro(orca, osc=TRUE,
        wl=wl, ovlp=ovlp,
        wn="hanning", zp=wl,
        collevels=seq(-100, 0, 1),
        palette=temp.colors)
```



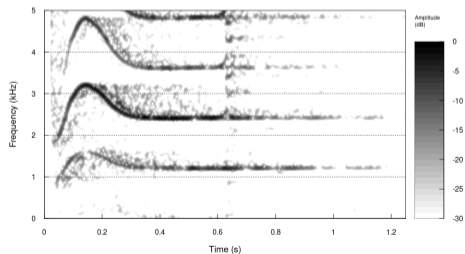
```
note1 <- cutw(orca, from=0.10, to=1.25, output="Wave")
```



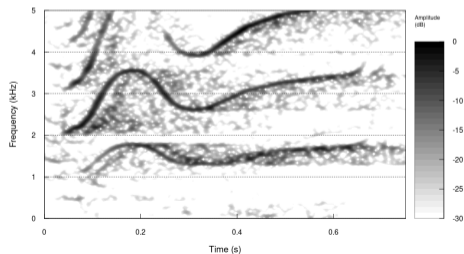
```
note2 <- cutw(orca, from=1.55, to=2.29, output="Wave")
```



```
note3 <- cutw(orca, from=4.75, to=6.00, output="Wave")
```



```
note4 <- cutw(orca, from=6.85, to=7.60, output="Wave")
```



Actividad

Suponga que se quieren identificar automáticamente dos tipos de notas en el canto de la orca: *note1* y *note3* *versus* *note2* y *note4*.

Se propone usar los contornos de frecuencia dominante. ¿Qué dificultades puede tener un enfoque basado en correlación cruzada para comparar esas series temporales?

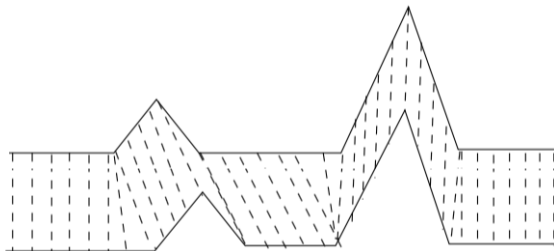
1. Correlación cruzada de series temporales

2. Deformación temporal dinámica

3. Correspondencia de patrones espectrales

La deformación temporal dinámica (DTW) es una técnica para comparar dos series temporales que pueden no estar en fase o tener duración diferente.

Los algoritmos de DTW consisten en estirar o comprimir de forma no lineal (i.e. deformar) las dos series temporales para encontrar su mejor alineamiento de modo de maximizar su similitud.



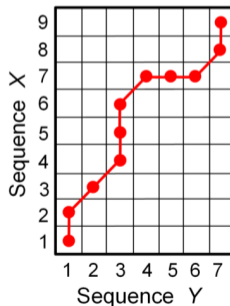
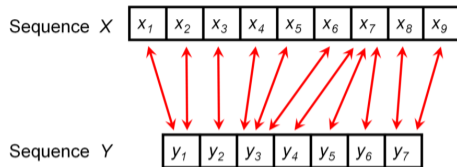
Encontrar el camino de deformación temporal que maximiza la similitud entre las secuencias:

$$x[n], \quad \text{con } n = \{1, \dots, N\}$$

$$y[m], \quad \text{con } m = \{1, \dots, M\}.$$

Para ellos se siguen los siguientes pasos:

1. Calcular la distancia entre cada par de puntos $x[n]$ e $y[m]$, y representarla como una **matriz de costo local**.
2. Calcular la **matriz de costo acumulado** al recorrer las secuencias desde inicio a fin.
3. Encontrar el **camino óptimo** en la matriz de costo acumulado, como el de menor costo.



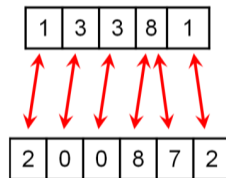
Cost matrix

1	1	1	1	7	6	1
8	6	8	8	0	1	6
3	1	3	3	5	4	1
3	1	3	3	5	4	1
1	1	1	1	7	6	1
	2	0	0	8	7	2

Accumulated cost matrix
and optimal warping path

1	10	10	11	14	13	9
8	9	11	13	7	8	14
3	3	5	7	10	12	13
3	2	4	5	8	12	13
1	1	2	3	10	16	17
	2	0	0	8	7	2

Resulting alignment

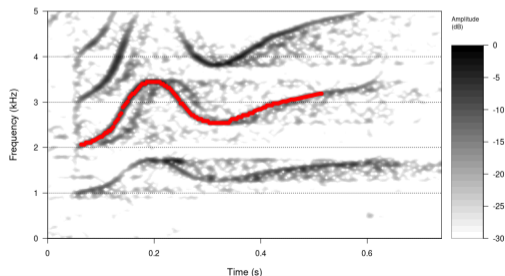
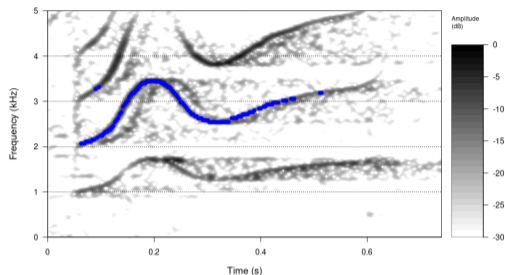


Estimación de frecuencia dominante.

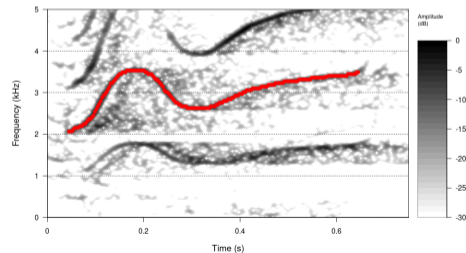
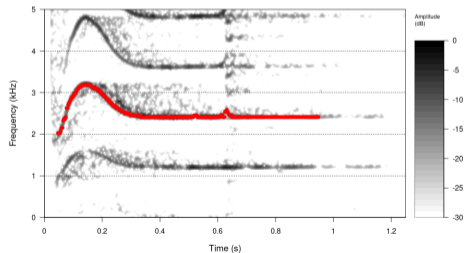
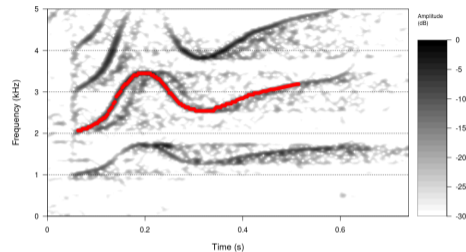
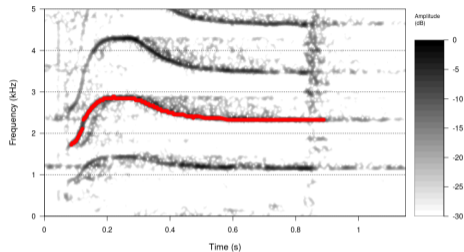
```
library(seewave)  
library(zoo)
```

```
# frecuencia dominante  
df2 <- dfreq(note2,  
             wl=wl, ovlp=ovlp,  
             clip=0.3,  
             bandpass=c(1900, 3600),  
             plot=FALSE)
```

```
# gráfico de frecuencia dominante  
points(df2, col='blue', pch=19)  
# puntos mal estimados se definen NA  
df2[31,2] <- NA  
df2[33,2] <- NA  
# interpolacion de valores NA  
df2_na <- na.approx(df2)  
points(df2_na, col='red', pch=19)
```



Deformación temporal dinámica

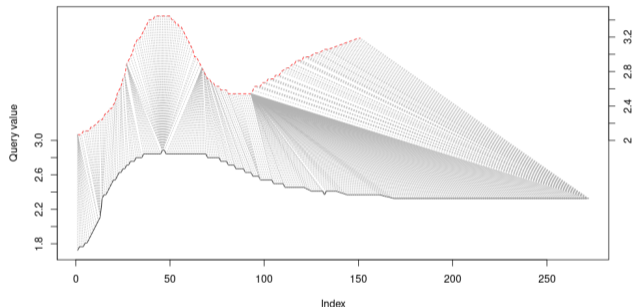


Alineamiento usando `dtw`.

```
# omitir valores NA
df1_ona = na.omit(df1_na)
df2_ona = na.omit(df2_na)

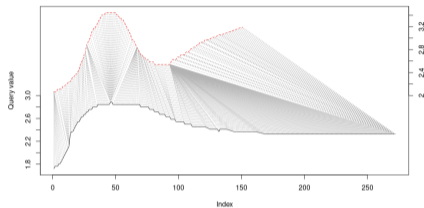
# solo valores de frecuencia
df1_flt <- df1_ona[,2]
df2_flt <- df2_ona[,2]

# alineamiento notas 1 y 2
dtw12 <- dtw(df1_flt, df2_flt,
              keep=TRUE)
plot(dtw12, type="twoway",
      offset=1)
dtw12$distance
```

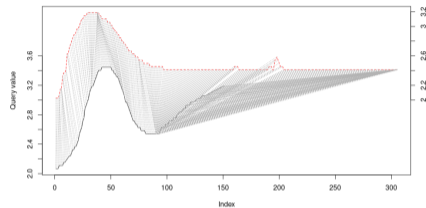


La **distancia DTW** es el costo acumulado del camino óptimo. Es una medida de similitud entre las secuencias.

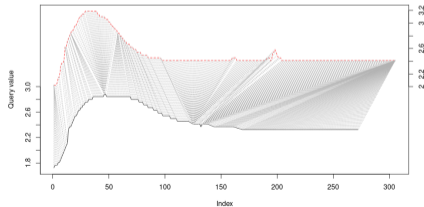
dtw12\$distance 84



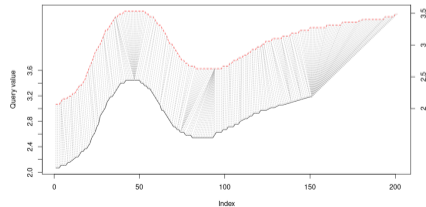
dtw23\$distance 63



dtw13\$distance 21



dtw24\$distance 13



¿Preguntas?

Actividad

Analice y ejecute el código provisto para comparar los contornos de frecuencia dominante de la grabación `orca.wav` usando DTW.

Compare las notas 3 y 4 de la grabación usando la distancia DTW. ¿Es razonable el resultado?

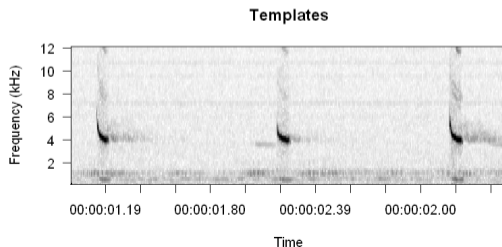
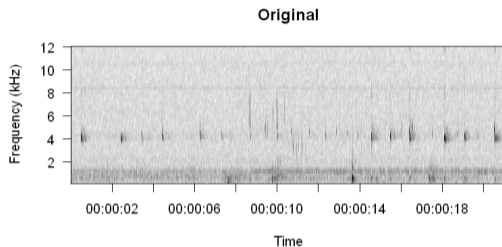
1. Correlación cruzada de series temporales

2. Deformación temporal dinámica

3. Correspondencia de patrones espectrales

Detección de vocalizaciones usando **correlación** entre ciertos **patrones espectrales** y el espectrograma de la señal.

Usamos el paquete **monitoR** y un ejemplo de **Pipistrellus kuhlii**.



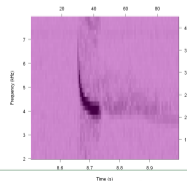
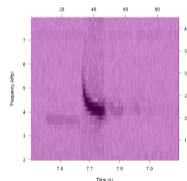
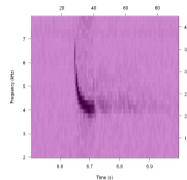
El procedimiento de detección automática consiste en los siguientes pasos:

1. Definir los **patrones espectrales** (templates) de los sonidos que se quiere detectar, como regiones específicas en el plano tiempo-frecuencia.
2. Calcular la **correlación cruzada** entre los patrones espectrales y el espectrograma de la señal de prueba, desplazando los patrones sobre el espectrograma.
3. Hacer una **detección de picos** en la función que resultante de la correlación, para identificar alta correlación entre el patrón espectral y la señal de prueba en esa región tiempo-frecuencia.

Seleccionamos los patrones espectrales (templates) como regiones específicas en el plano tiempo-frecuencia.

```
library(monitoR)  
library(seewave)
```

```
# se crean los templates  
temp1 <- makeCorTemplate('./templates.wav',  
                        t.lim = c(6.5, 7),  
                        frq.lim = c(2, 8),  
                        name = "temp1")  
temp2 <- makeCorTemplate('./templates.wav',  
                        t.lim = c(7.5, 8),  
                        frq.lim = c(2, 8),  
                        name = "temp2")  
temp3 <- makeCorTemplate('./templates.wav',  
                        t.lim = c(8.5, 9),  
                        frq.lim = c(2, 8),  
                        name = "temp3")
```



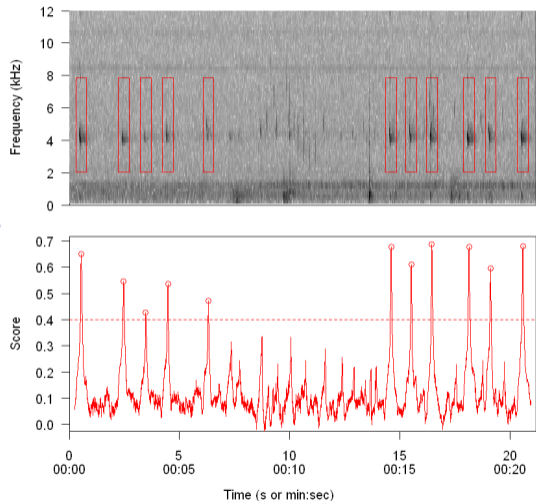
Correlación cruzada y detección de picos.

```
# se combinan los templates
ctemps <- combineCorTemplates(temp1,
                               temp2,
                               temp3)

# correlacion entre el espectro de
# la señal original y cada template
cscores <- corMatch('./pipistrellus.wav',
                    ctemps)

# se buscan picos en la correlacion
peaks <- findPeaks(cscores)

# grafico de las detecciones
plot(peaks,
     which.one="temp3",
     legend=FALSE,
     hit.marker="points",
     color='red')
```



¿Preguntas?

Actividad

Usando el código provisto compare las detecciones para cada uno de los tres patrones espectrales definidos para el ejemplo de *Pipistrellus kuhlii*.

¿En qué situaciones tiene sentido definir más o menos patrones espectrales?

¿Cuál es el parámetro que controla la cantidad de detecciones?

Pruebe modificarlo y analice los resultados obtenidos.

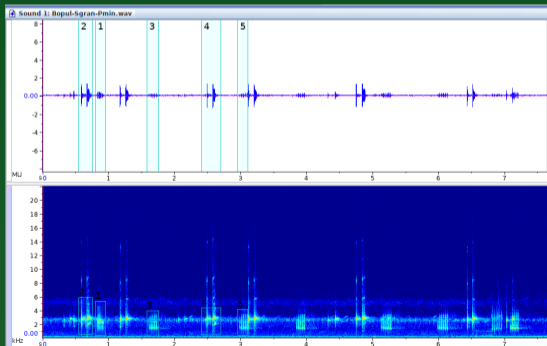
¿Qué limitantes tiene el enfoque de detección automática usando correspondencia de patrones espectrales?

Actividad

Analice la grabación Bopul-Sgran-Pmin.wav en Raven usando las anotaciones provistas.

¿Qué especies hay presentes? ¿Hay varias vocalizaciones de cada especie?

¿Cree que es viable hacer detección automática usando correspondencia de patrones espectrales? ¿Qué dificultades identifica?





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