Development of a MATLAB computational method for the bioacoustical characterization of anuran vocalizations

Antonio Palanca-Soler[1], Luz Calia Miramontes-Sequeiros[1], Nicolás Palanca-Castán[2]

[1]Animal Anatomy Laboratory Foundation, Department of Ecology and Animal Biology, Faculty of Biology, University of Vigo.
[2]Centro Interdisciplinario de Neurociencia de Valparaíso (CINV), Neuroscience of Circuits and Systems laboratory, Facultad de Ciencias, Universidad de Valparaíso. Chile.

Abstract

In this paper, we present a method for the bioacoustic analysis of Anuran vocalizations. We developed a MATLAB script in order to describe Anuran, taking as a study model Rana temporaria vocalizations. Using a total of 99 different variables we characterized the time and frequency domains of each individual croak.

INTRODUCTION

Anuran amphibians rely greatly on acoustic intraspecific communication to attract mates and demark and defend their territory. As a consequence of this, the central auditory system in anurans is highly specialized for detecting and processing conspecific vocalizations (review in Wilkzynski & Ryan 2010). The reproductive behavior in anurans is primarily based on sound and in most species; males aggregate at breeding sites and try to attract females using advertising calls. Male vocalizations and the associated mechanical and neural mechanisms are thus under a strong sexual selection. Traits subject to sexual selection are very labile and can undergo rapid changes that make them more exaggerated and elaborated or more innovative (Endler 2005). Such changes are fast enough to cause divergence and reproductive isolation between nearby populations and have been suggested as a cause of sympatric speciation (e.g. review in Panhuis *et al*. 2001). Although this paints a relatively simple picture of anuran vocalization, there are studies that suggest the roles; ranges and variation of intraspecific sound communication in these animals are rather wide. First, female anurans are known to respond to mating calls, as well as initiate their own (Emerson & Boyd 1999). In addition, frogs are known to engage in spontaneous vocalizations (Wells 2007), which have no obvious reproductive or territorial function. Lastly, a few studies that have investigated vocalizations in meticulous detail indicate that at least some anurans have significant individual variation in their particular calls, and can use calls to differentiate between individuals (Bee *et al.* 2001; Bee 2004).

The complexity of anuran vocal repertoire, as well as its relations with their ecology, behaviour and breeding habits, makes bioacoustic studies of these animals highly informative and valuable. Bioacoustic tools for ecological monitoring are powerful instruments for biodiversity assessment, allowing scientists to gather valuable data regarding species occurrence and richness, as well as using individual differences to obtain population abundance estimates (review in Blumstein *et al.* 2011). These tools are completely non-invasive, representing an improvement over traditional mark-and-recapture methods and can be passive, allowing flexibility in the spatial and temporal range of sampling campaigns. Frogs are animals that can be found, manipulated and recorded with relative ease, which contributes to their usefulness as a model animal to develop bioacoustic tools.

In this paper we present a computational model developed using MATLAB script to rapidly and easily process the vocalizations sampled from our study population. We are interested in separating individuals to allow for non-invasive mark-and-recapture population surveys, as well as developing tools that would help with identifying the position of a frog in a "bioacoustic population map" to study animal movements between the different valleys in our study area. More specifically we illustrate the methods described above with an analysis of the differences between the release calls of males and females, characterizing the bioacoustics and ecological relevance of induced, territorial and mating-related vocalizations in an alpine population of *Rana temporaria* from the Spanish central Pyrenees.

Identifying and analyzing the main variables that define these vocalizations, as well as differences between sexes, individuals and neighboring sub-populations will provide us with great insight into their behaviour and could result in the creation of fast, cost-efficient and non-invasive tools to identify individuals, sexual behavioral variants and subpopulations. Such tools could be useful in future ecological and behavioural studies.

METHODS

Samples

The samples used for the development of our analysis methodology were taken from individuals of *Rana temporaria* from the Central Spanish Pyrenees (Aragón, Spain) in the summer of 2013 and 2014. We aimed to record release calls, which are produced by both male and unreceptive female frogs when grasped by male frogs (Kentwood & Schwartz 2007). This type of vocalization can be easily induced by gently grabbing the frog with little damage or stress for the animal (Schmith 1972). Vocalizations were recorded using a TCD-D8 Portable Digital Audio Tape (Sony Corporation, Tokyo, Japan) a directional microphone (C568B, AKG acoustics, Vienna, Austria) with integrated amplifier and 20-20.000 Hz bandwidth. Sampling rate was 44.1 kHz.

From each recording we used SoundForge Pro 10.0 (Magix Software GmbH, Berlin, Germany) to separate individualized croaks that were saved in *.wav format (Fig. 1) and normalized by considering the largest peak as 100% amplitude (>>y=y/max(y);) in order to compared them.

Figure 1. Oscillograms of a vocalization. In the upper window there is a vocalization in two channels (right and left) and in the lower window, there is a separate individualized croak with SoundForge Pro 10.0 (Magix Software GmbH, Berlin, Germany).

Calculation of variables

We used MATLAB to calculate numerical variables transforming the oscillograms, spectrograms and periodograms, which define the sound and allowed us to compare numbers instead of graphs and figures. We selected the release call of *Rana temporaria* as an example to test our computational model. General definitions are standard physical terminology.

We only took into consideration the individual croaks within a vocalization regardless of the time between them, since we have observed that this last factor is highly dependent on air temperature, the time of recording, individual diet and other environmental factors (Larom *et al*. 1997) and it is hard to evaluate during high-mountain surveys. On the other hand, the variables we are studying for individual croaks are directly related to anatomical elements of the mouth, pharynx, vocal cords and lungs of each frog. This anatomical aspect is often overlooked in studies of frog vocalization, in contrast with studies of human speech, where anatomy is an important aspect.

We considered 95 primary sound-related variables to describe each single croak within a given vocalization.

Our MATLAB script is divided in seven parts of boxes. Each box and the variables it covers are described below.

Box 1: we developed the first part that opens the *.wav files corresponding to each individual and checks that no more of three consecutive croaks are missing. Each single croak is assigned a sample frog number followed by an individual number based on its ordinal position within the recorded vocalization. These numbers are reflected on the

```
% BOX 1 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%the first step is to ask for the folder where the
%vocalization files are saved, as well as the individual that
%we want to analyze
clear; % clear workspace
clc; % clear command window
file=input('Please specify the target folder: ','s');
name=input('Please input individual ID:','s');
rana=str2num(name);
%_____________________________________________________
%here we declare the variables to be used
salir=0 ; % Becomes 1 whenever an error is detected
estas=0; %variable that checks for errors
filas=0; % variable for the number of rows of the final result matrix
matriz=zeros(1,95); % create a matrix with a number of
%columns equal to the studied variables
%_____________________________________________________
%These files with consecutive numbers will be created by
%this script automatically using the following loop:
for n=1:99  % start of the “for” loop
  seguir=0; %declare variable to decide whether we
  %continue the loop or not
  %the file names of the .wav files containing the individual
  %vocalizations are formed by the frog ID, the vocalization
  %number and a V that indicates the recording has been
  %normalized (for example: “320_001_V.wav”).
  nstr=int2str(n); % we transform “n” to a string
  %this next piece of code will add one or two zeros before
  %the vocalization number for those vocalization numbers
  %that require it
  if (n<10) %(1*)
    numrana=strcat('00',nstr);
  else  % in case n is between 10 and 99
    numrana=strcat('0',nstr);
  end;%(1*)
  filename=strcat(file,'\',name,'_',numrana,'_','V.wav');
  filename=strrep(filename,' ',''); %removes any
     %superfluous empty spaces from the file name, which
     %can appear if the user copies and pastes the folder address
  try  %checks for the presence of the file
    [x,Fs]= audioread(filename); %tries to open the indicated file
  catch err  %if it can’t find the specified file
    estas=estas+1; %this value increases everytime a file
    %is not found, and the script carries on to the next
    %consecutive number
    seguir=1;
  end; %of “try”
  if (estas>3)break  % (2*)breaks the loop and ends the script if
  %there are more than three missing files
  else  %the script continues
  end;  %(2*)
```

corresponding file name. Finally, a *V* is added at the end of the file name to indicate that the croak has been normalized (Fig. 2) to avoid script errors.

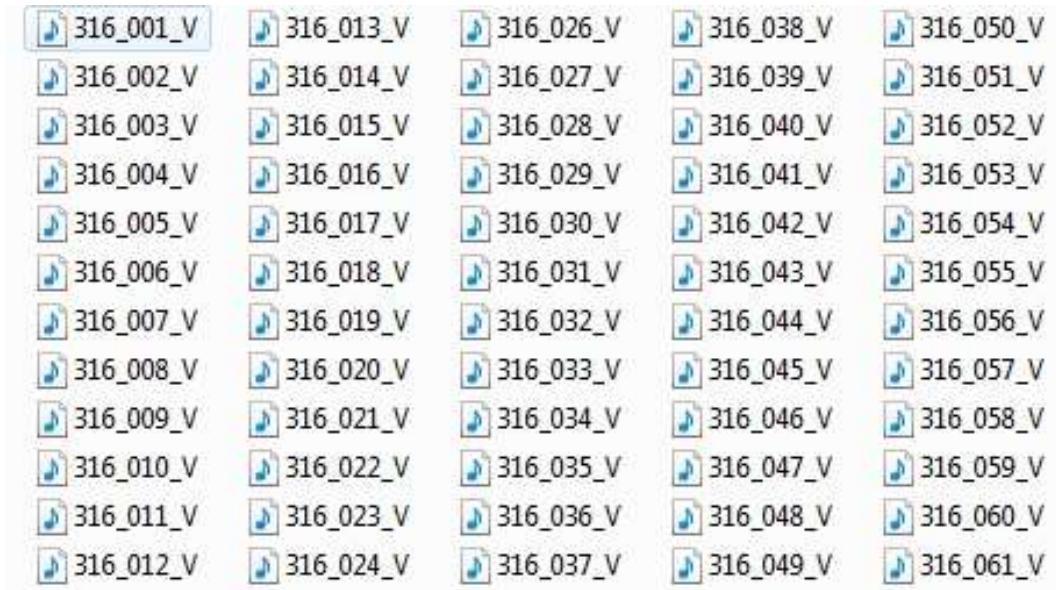


Figure 2. Example of files obtained from frog 316, where each of the 61 single croaks is separated, normalized (V) and ordered based on their file names.

<u>Box 2</u>: in the second part we cut the *.wav file, leaving only the length of each individual croak. We then calculated the values of variables 1 to 6 and save them in the general matrix.

Variable 1*:* Frog ID number. Each individual is assigned a unique number in the database, which will be part of the name of any file derived from its recording.

Variable 2*:* Croak number.

Variable 3: Sampling rate in Hz, as the number of times per second that the level of a digital signal changes (Fig. 3).

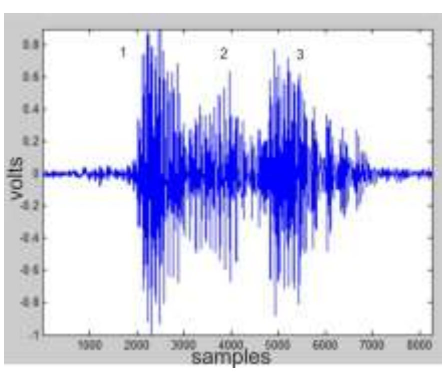


Figure 3. Representation of the amplitude-time course of a single croak. The x-axis shows the samples with the corresponding amplitude in volts in the y-axis. In this oscillogram,

```
% BOX 2   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

 if (seguir<1) estas=0; %(3*)checks for errors,
        %if there are no errors detected the script continues
    filas=filas+1;  %if everything is correct, the script
        %determines the position of the vocalization within the
        %rows of the results matrix. Rows will increase with the
        %number of vocalization files, but the vocalization
        %number and row number need not be the same, as a
        %missing vocalization file will result in an increase in "n"
        %without an increase in rows.
        %In the corresponding vocalization file is present, the
        %recording is saved in "x", and the three first columns of
        %the corresponding row are filled with frog ID,
        %vocalization number and sampling rate, respectively.

        %============== variables 1 to 3========
    matriz (filas,1)=rana; %frog ID
    matriz (filas,2)=n; %vocalization number
    matriz (filas,3)=Fs; % sampling rate
        %-----------Now the vocalization length is calculated
    Cx=midcross(x); %determines the bin where the signal as
        %measured in volts crosses the 0 line.
    lon=max(Cx)-min(Cx); %end of vocalization minus start of
        %vocalization.
        longitud=lon*1000/Fs; %length of vocalization in
        %milliseconds
        %============== variable 4 ===========
    matriz (filas,4)=longitud; %variable 4, vocalization length in ms
    R=[min(Cx),max(Cx)]; %this saves the cut part of the
        %vocalization between the first and last zero-line crossings.
        R=round(R); %the trace is converted to integer in order to
        %make it readable to the "audioread" function.
        %--------------------------------
    [xR,Fs]= audioread(filename, R); %xR saves the selected %part
        %of the trace
        P=pulseperiod (xR); %Divides the trace in pulses, periods
        %between two consecutive zero-line crossings.
        pul=size(P,1); %Copies the first column from P.
        if (pul==0)  %This prevents divide-by-zero errors down the
        %line.
                LP=0;
        elseif (pul > 0) LP=longitud/pul;
        else
        end
        %=============== variables 5 and 6========
        matriz (filas,5)=pul;  %Variable 5, number of pulses
        matriz (filas,6)=LP;  %Variable 6, pulses per ms.
```

three main modules, as clusters of pulses that increase and decrease integrating a group creating a wave in the oscillogram profile, are present (1, 2, 3).

Variable 4*:* Croak duration in milliseconds calculated from the number of samples and the sampling rate.
Variable 5*:* Number of pulses in the croak. A pulse is defined in MATLAB as the period between two consecutive zero line crossings by the recorded waveform (Fig. 4).
Variable 6: Pulses per millisecond, calculated from variables 4 and 5.

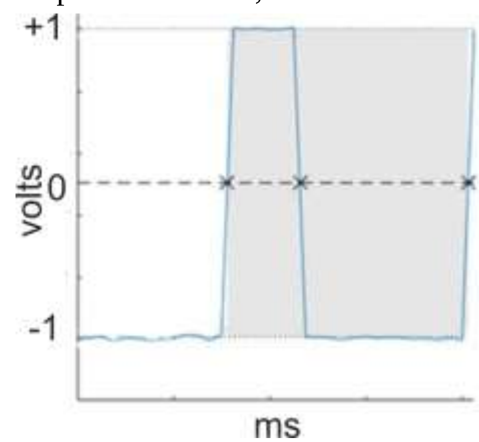


Figure 4. Illustration of a pulse defined in MATLAB as two consecutive zero line crossings by the recorded waveform. In the oscillogram, the x-axis shows recording time in ms and the y-axis shows waveform amplitude in volts.

Box 3: in the next steps we examined the spectral power estimates (SPEs), also known as croak periodograms or spectrograms (Fig.5). Using Burg's algorithm we could fit a curve in order to simplify the interpretation of the periodogram (Fig. 6).

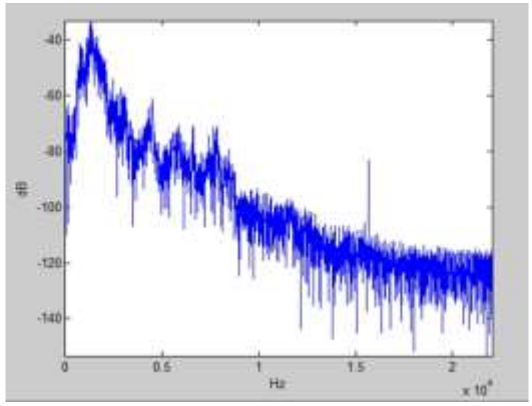


Figure 5. A croak periodogram, power spectral density estimate (PSD), obtained using a Fourier transform and a Hamming window.

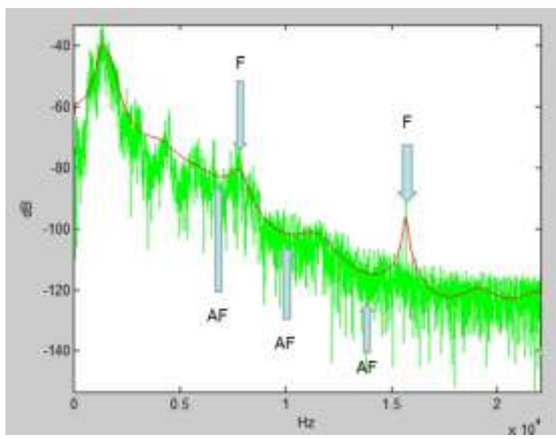


Figure 6. The green trace shows the periodogram obtained via a Fourier transform and Hamming window already shown in Figure 5. The red line is a fit using Burg's algorithm at 20 sensitivity, arrows point to formants (F) and anti-formants (AF).

Variable 7*:* Number of formants obtained subtracting the curves values of the Burg algorithm 50 sensitivity (Burg_50) from the one with Burg algorithm 2 sensitivity (Burg_2), which represents the frequencies modulation due to the inherent anatomical and physical properties of the sound organs and the fundamental frequency of a croak related to vibrating structures in the sound-producing organs respectively (Fig. 7). As a result we obtained a differential curve (Fig. 8) where peaks with a value over the fit larger than a threshold (5 in our case) can be identified, which are considered formants.

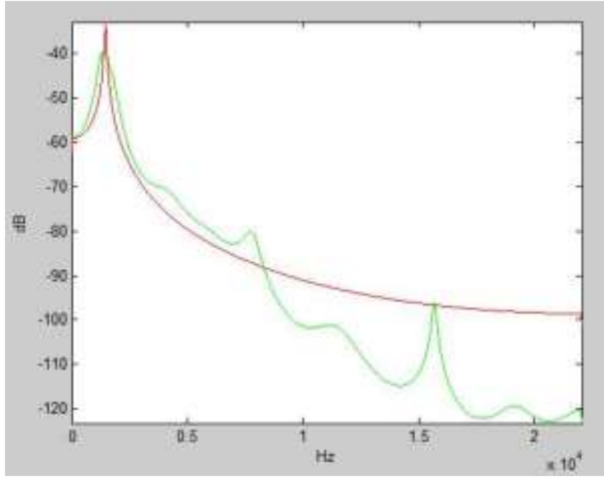


Figure 7. Fits to the periodogram of a single croak using Burg's algorithm. The red line represents the fit using a Burg algorithm 2 sensitivity, which represents the fundamental frequency of a croak related to vibrating structures in the sound-producing organs, while the green line represents the fit using a Burg algorithm 20sensitivity.

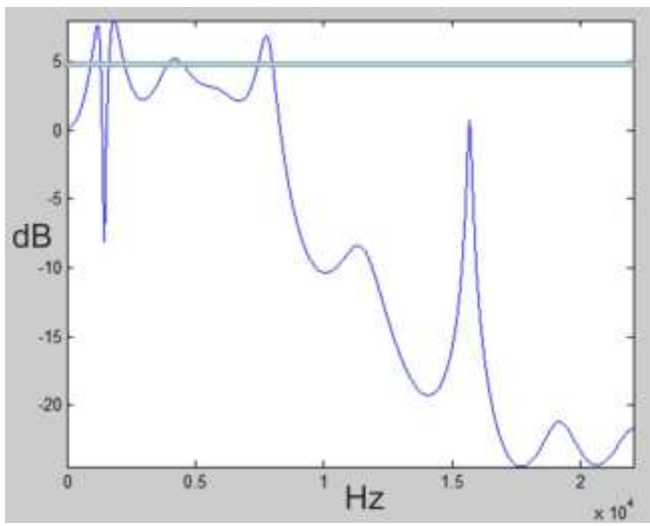


Figure 8. Difference curve between the fits obtained from the periodogram using Burg's algorithm at 2 and 50 sensitivities. The +5 dB threshold is depicted in order to highlight the formants.

Variables 8 to 12*:* Decibel (dB) value of the difference between the Burg_2 and Burg_50 fits of the periodogram for the 5 first formants.

Variables 13 to 17*:* Values in Hz of the first 5 formants.

Variable 18*:* Number of anti-formants. Calculating the difference between the periodogram fits obtained with Burg_2 and Burg_20 nets, a curve where peaks larger than a given threshold (5 in our case, Fig. 9) would be considered the anti-formants. Burg_20 is used for this process instead of the Burg_50 used for the formants, since Burg_50 tends to almost completely eliminate anti-formants.

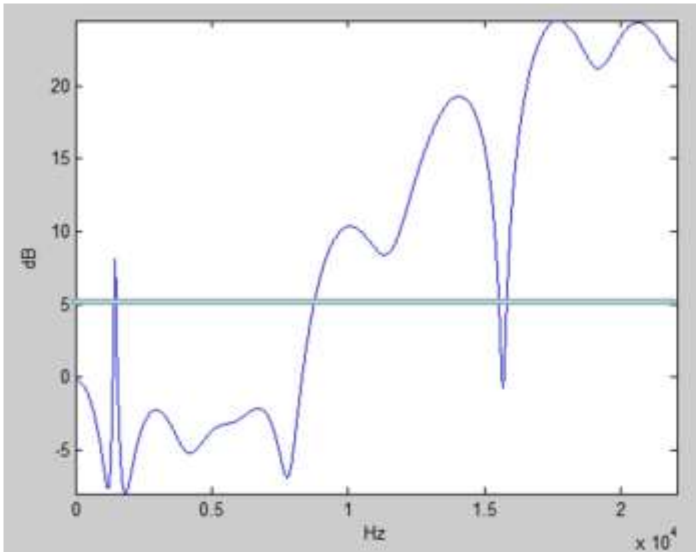


Figure 9. Curve representing the difference between the Burg_2 and Burg_20 fits to the periodogram. The +5 dB threshold value is shown to highlight the anti-formants.

Variables 19 to 23*:* Decibel value of the difference between the Burg_2 and Burg_20 fits of the periodogram for the 5 first anti-formants.
Variables 24 to 28*:* Value in Hz of the first 5 anti-formants.
Variables 29 to 33: Decibels of the equivalent x-axis position of the first 5 formants on the Burg_2 fit.
Variables 34 to 38*:* Decibels of the equivalent x-axis position of the first 5 anti-formants on the Burg_2 fit.

```
% BOX 3  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    %----We will use Burg's algorithm with sensitivities 2, 20 and
    % 50 with 2500 samples --------------------
    try  % checks for errors
        [xRB2,FB]=pburg(xR,2,2500,Fs); % Burg's algorithm
        %with 2 sensitivity
        curva2=10*log10(xRB2); %Obtain the value on the y
        %axis for each
        % point on the x axis
        [xRB20,FB]=pburg(xR,20,2500,Fs);  %Burg's
        % algorithm with 20 sensitivity
        curva20=10*log10(xRB20);  %y axis value
        [xRB50,FB]=pburg(xR,50,2500,Fs);  %Burg's
        % algorithm with 50 sensitivity
        curva50=10*log10(xRB50);% y axis values
        MH=max(FB); %Max frequency in Hz, which will be
        % used later as variable 88
    catch err
        salir=1;
        strcat('DELETE FILE = ',filename)
        %Error message indicating a defective file break
    end
    %********We create a provisional matrix named MATbase
    % with the 50 sensitivity curve for each row. This matrix will
    % be used at the end of the script **************
    if (filas ==1)
        nfila=size (FB,1); % Defining "nfila" for the first time
    else
    end
    MATbase(1:nfila,filas)=curva50;
    %Subscripted assignment dimension mismatch filas=4
    % -----Now we proceed with the formant calculation-----------
    formantes=curva50-curva2; %Subtract curve 2 from curve
    %50 in order to highlight the formants
    try
        [pksx,locsxf]=findpeaks(formantes,
'MINPEAKHEIGHT',5);
        %Remove peaks below threshold
        nf=length(pksx); %Determine formant number
    catch err %In case there are no formants
        nf=0;
    end
    %============ Variable  7========
    matriz (filas,7)=nf; %Formant number
    if(nf==0) %Detects when there are no formants
        pksx(1:5,1)=999999;  %Declares 999999 as "missing
        %value"
    elseif (nf<5) pksx(nf+1:5,1)=999999; %Creates a
    %matrix named "pksx" with at least 5 rows
    else
    end
```

```
pksx=pksx'; %Transpose the matrix
%============ Variables 8 to 12=================
matriz (filas,8:12)=pksx(1,1:5); %Difference value in
%dB of the first five formants
if (nf>0) Herzios=FB(locsxf); %Frequency values of the first
        %five formants
else
end
if(nf==0) Herzios(1:5,1)=999999;
elseif (nf<5) Herzios(nf+1:5,1)=999999;  %Increase rows
        %when needed
else
end
Herz=Herzios';  %Transpose matrix
%=============Variables 13 to 17=============
matriz (filas,13:17)=Herz(1,1:5);  %We introduce the
%frequency in Hz of the formants in the matrix
%  --------------Anti-formants---------------
antiformantes=curva2-curva20; %Find the anti-formants
try
        [pksx,locsxaf]=findpeaks(antiformantes,
'MINPEAKHEIGHT',5);
        nf=length(pksx); %Find the number of anti-formants
catch err
        nf=0;
end
%================Variable 18 ========
matriz (filas,18)=nf;  %Introduce number of anti-
%formants in variable matrix
if (nf==0) pksx(1:5,1)=999999;
elseif (nf<5) pksx(nf+1:5,1)=999999; %We créate a
%"pksx" matrix with at least 5 rows
else
end
pksx=pksx'; %transpose pksx
%============Variables 19 to 23=============
matriz (filas,19:23)=pksx(1,1:5); %Enter the difference
%in dB of the first 5 anti-formants into the matrix
if (nf>0)Herzios=FB(locsxaf); %Find the frequencies
%where the anti-formants are located
else
end
if(nf==0) Herzios(1:5,1)=999999;
elseif (nf<5) Herzios(nf+1:5,1)=999999; %Increase rows
%when necessary
else
end
Herz=Herzios';  %Transpose matrix
%=============Variables 24 to 28=============
matriz (filas,24:28)=Herz(1,1:5); %Enter the frequency in Hz
%of the first five anti-formants
```

```
%------Formant and anti-formant data from the Burg 2 fit----
nf=length(locsxf); %find number of formats
if (nf>0)dbnormalf=curva2(locsxf);  %Intensity in dB
%of the formants from the Burg 2 fit
else
end
if (nf==0)dbnormalf(1:5,1)=999999;
elseif (nf<5) dbnormalf(nf+1:5,1)=999999; %Creates
%the "dbnormalf" matrix with at least five rows
else
end
nf=length(locsxaf); %Find the number of anti-formants
if (nf>0)dbnormalaf=curva2(locsxaf); %Intensity in dB
%of the anti-formants on the normalized curve
else
end
if (nf==0) dbnormalaf(1:5,1)=999999;
elseif (nf<5) dbnormalaf(nf+1:5,1)=999999; %Creates
%the "dbnormalaf" matrix with at least 5 rows
else
end
dbnormalf=dbnormalf'; %transpose matrix
dbnormalaf=dbnormalaf'; %transpose matrix
%==============variables 29 to 38==============
matriz (filas,29:33)=dbnormalf(1,1:5);
matriz (filas,34:38)=dbnormalaf(1,1:5);
```

Box 4: using the *thd* MATLAB function we calculated the distortion (modification of harmonics by formants and anti-formants), intensity in dB and frequency of the harmonics, which form our next variable cluster.

Variable 39: Measures the distortion calculated using the *thd* MATLAB function.

Variables 40 to 44: Intensity in dB of the first five harmonics.

Variables 45 to 49: Frequency in Hz of the first five harmonics. The first harmonic is named fundamental frequency.

Variable 50: Average power, calculated using the *bandpower* MATLAB function (Fig.10).

Variables 51 to 55: Intensity in dB of the first five formants calculated from variables 8-12 and 29-33.

Variables 56 to 60: Intensity in dB of the first five anti-formants calculated from variables 19-23 and 34-38.

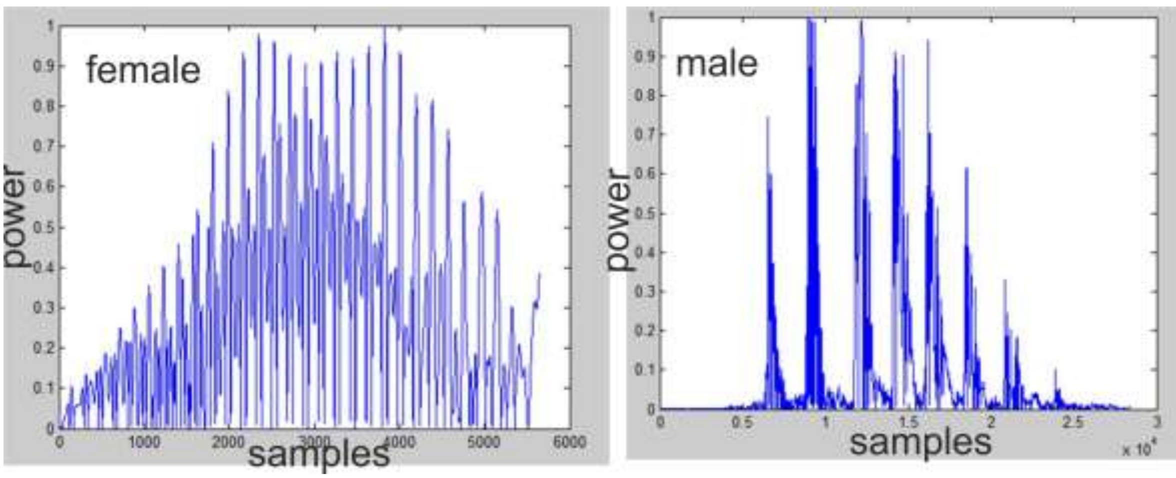


Figure 10. Oscillograms of a single croak with samples as the x-axis, and the power measurement (P) correspond to dB ([dB] = 10 log 10 [P]) as the y-axis. On the left hand side a female croak oscillogram with an average power, of 0.1454 P and on the right hand side a male croak oscillogram with an average power of 0.0391 P. Average power calculated using the "bandpower" MATLAB function (variable 50).

```
% BOX 4   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        %............Harmonics......................................
        [ Distorsion , harmpow , harmfreq ] = thd (xR, Fs,5 );
        %5 primeros armónicos
        %===============Variable 39===================
        matriz(filas,39)=Distorsion; %Harmonic distortion
        harmpow=harmpow';
        %==================Variables 40 to 44=========
        matriz(filas,40:44)=harmpow(1,1:5); %Intensity in dB
        %of the first five harmonics
        harmfreq=harmfreq';
        %===============Variables 45 to 49==============
        matriz(filas,45:49)=harmfreq(1,1:5); %Frequency in
        %Hz of the first five harmonics

        Potencia=bandpower(xR); %Average power
        %=============Variable 50===================
        matriz(filas,50)=Potencia; %Average power
        %....Data of formants and anti-formants of the Burg 50 fit ...
        nf=length(locsxf);
        if (nf>0)dbf=curva50(locsxf); %Intensity in dB of the
        %formants of the Burg 50 fit
        else
        end
        if (nf==0)dbf(1:5,1)=999999;
        elseif (nf<5) dbf(nf+1:5,1)=999999; %Creates the
        %matrix “dbf” of at least five rows
        else
        end
        nf=length(locsxaf);
        if (nf>0)dbaf=curva20(locsxaf); %Intensity in dB of
        %the anti-formants of the Burg 20 fit
        else
        end
        if (nf==0) dbaf(1:5,1)=999999;
        elseif (nf<5) dbaf(nf+1:5,1)=999999; %Creates the
        %matrix “dbaf” of at least five rows
        else
        end
        dbf=dbf'; %Transpose matrices
        dbaf=dbaf'; %Transpose matrices
        %==============Variables 51 to 60  ===========
        matriz(filas,51:55)=dbf(1,1:5);
        matriz (filas,56:60)=dbaf(1,1:5);
```

Box 5: we defined variables to complete the study of the croak oscillogram and spectrogram.

Variable 61*:* Total number of power spikes above 0.1 in order to eliminate noise, which represents 10% of maximum power (Fig.10).

Variable 62*:* Number of power spikes (variable 61) per time unit (ms).

Variables 63 to 65*:* Average of the difference in dB between the Burg_30 and Burg_2 curves on the low- (<7333 Hz), mid- (between 7334 and 14666 Hz) and high - frequency (between 14667 and 22000 Hz) sections of the periodogram (Fig. 11).

Variables 66 to 68*:* Differences between the average intensities in dB of the first and last third (Fig.12P1A and P1B) of the time course of the croak spectrogram applied separately to low-, mid- and high- frequencies (Fig. 13).

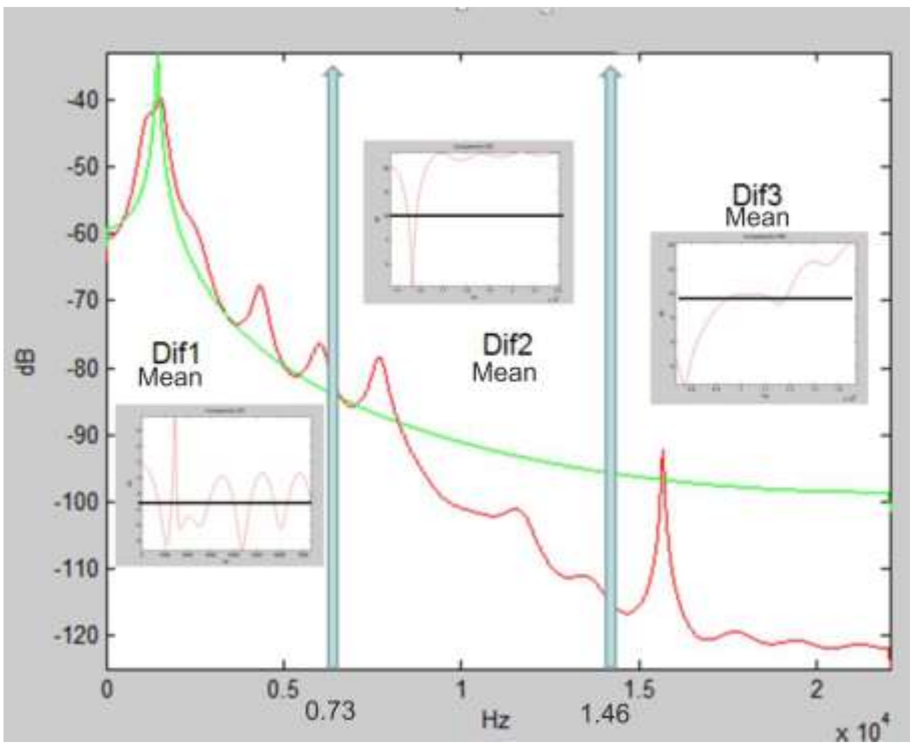


Figure 11. Representation of the Burg_2 (green) and Burg_30 (red) fits to the low-frequency (<7333 Hz), mid-frequency (between 7334 and 14666 Hz) and high-frequency (between 14667 and 22000 Hz) sections of the periodogram. Inset graphs show the average difference between Burg_2 and Burg_30 for the low-, mid- and high-frequency sections (Dif1, Dif2 and Dif3, respectively), which correspond to variables 63 to 65.

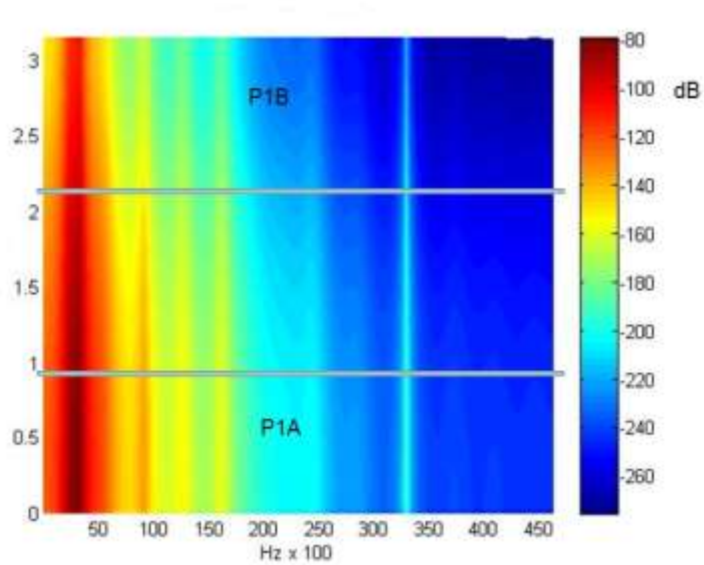


Figure 12. Spectrogram from a single croak, divided in three parts corresponding to the first, second and third parts of the recording time (y-axis). P1A and P1B mark the first and third of the recording time, respectively.

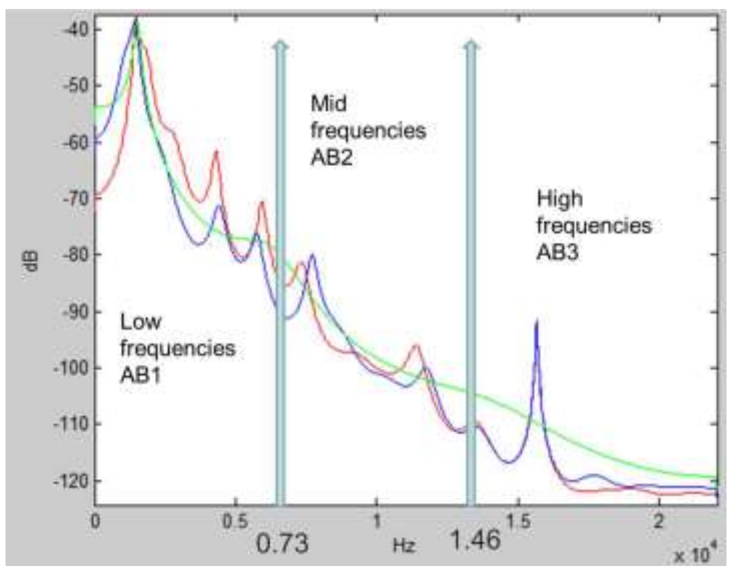


Figure 13. Comparison between the Burg_30 fits to the periodograms of the first (red) and third (blue) parts of the Fig.12 spectrogram's time course. A Burg_7 fit is shown in green for reference. Frequencies in the x-axis are divided in low, mid, and high (AB1, AB2 and AB3), which correspond to variables 66 to 68 respectively.

```
% BOX 5   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        %===============Power and peaks==========
        DB=db(xR); %Converts the recording from volts to dB
        POT=db2pow(DB); %Calculates power
        try
                [pk,loc]=findpeaks(POT, 'MINPEAKHEIGHT',0.1);
                msgloc=(loc*1000)/Fs; %Converts samples to ms
                npicos=size(msgloc,1);
                nmseg=max(msgloc);
                picosmseg=npicos/nmseg;%Calculates the
                %number of peaks per ms
        catch err
                npicos=0;
                picosmseg=0;
        end
        %============Variables 61 and 62 ======
        matriz(filas,61)=npicos;
        matriz(filas,62)=picosmseg;
        %  =========Average of the differences between Burg
        %30 and Burg 2. =============
        DIFCURVAS=curva20-curva2; %An average is also
        %calculated for each third of the recording
        DIF1=round(mean(DIFCURVAS(1:416,1)));
        DIF2=round(mean(DIFCURVAS(417:832,1)));
        DIF3=round(mean(DIFCURVAS(833:1250,1)));
        %===============Variables 63 to 65===========
        matriz(filas,63)=DIF1;
        matriz(filas,64)=DIF2;
        matriz(filas,65)=DIF3;
        %========Comparison of the first and
        %the third thirds of the recording========
        %====xR contains the individual vocalization======
        Rtercio=lon/3;
        R1=[min(Cx),min(Cx)+Rtercio];R1=round(R1);
        R3=[(min(Cx)+Rtercio+Rtercio),max(Cx)]; R3=round(R3);
        [xR1,Fs]= audioread(filename, R1);
        [xR3,Fs]= audioread(filename, R3);
        %-------Using Burg's algorithm with sensitivity set at 30 -----
        try
                [xR1B30,FB]=pburg(xR1,30,2500,Fs);
                curvaR1=10*log10(xR1B30);
                [xR3B30,FB]=pburg(xR3,30,2500,Fs);
                curvaR3=10*log10(xR3B30);
                DIFCURVAS=curvaR1-curvaR3;
                DIF1=round(mean(DIFCURVAS(1:416,1)));
                DIF2=round(mean(DIFCURVAS(417:832,1)));
                DIF3=round(mean(DIFCURVAS(833:1250,1)));
        catch err
                DIF1=999999;
                DIF2=999999;
                DIF3=999999;
```

```
end
%===============Variables 66 to 68===========
matriz(filas,66)=DIF1;
matriz(filas,67)=DIF2;
matriz(filas,68)=DIF3;
```

Box 6: because the preliminary observations of the croak shape in the oscillogram suggested there were differences between males and females, in this box we designed a new set of variables describing the amplitude modulation.
Variable 69*:* Number of modules (Fig.14). We defined a new variable in order to differentiate from pulses and peaks, which define a cluster of pulses that increase and decrease uniformly integrating a group creating a wave in the oscillogram profile (Fig. 3).
Variables 70 to 78*:* distance in ms from the beginning of the croak to module 1 to 9, considering the first nine if present, otherwise they are considered as a missing value.
Variables 79 to 87*:* Maximum power of the first nine modules, if present.
Variable 88*:* Maximum frequency of the periodogram calculated in BOX 3.

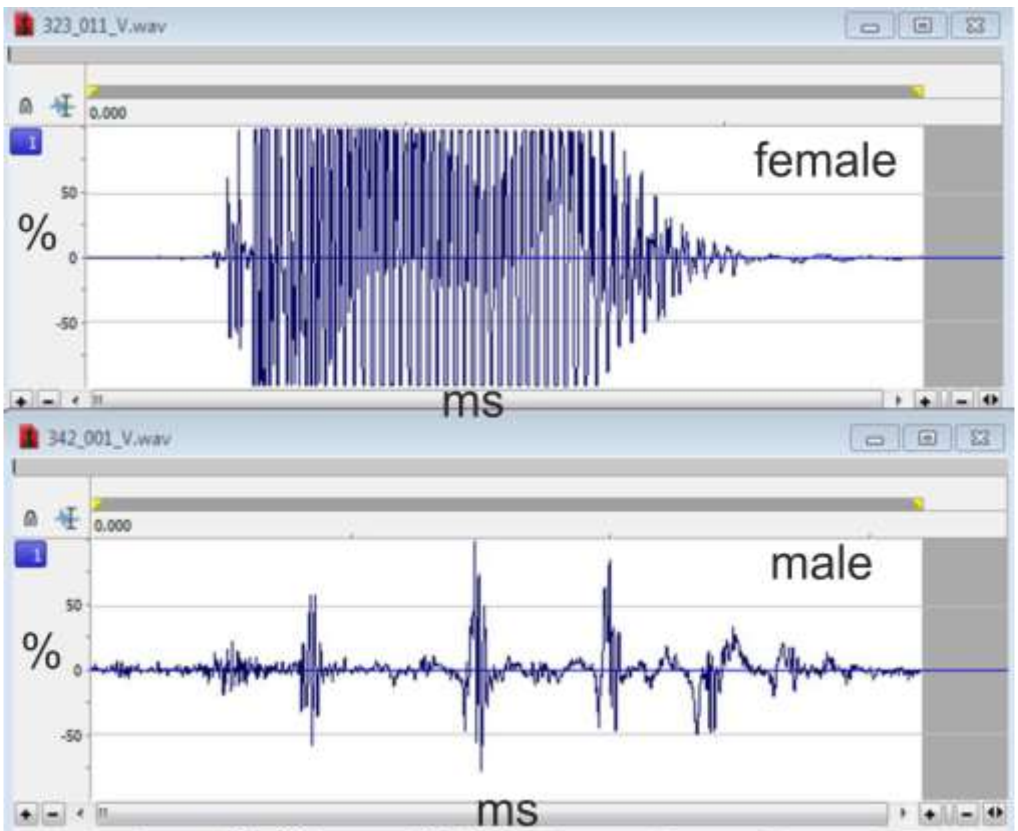


Figure 14. Oscillogram of female and male croaks, upper and lower graph respectively of individuals of Rana temporaria. Note the clearly differentiated modules compaction between sexes.

```
% BOX 6   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
        %=======count ridges and distances between them ====
        try  %Look for the rodges and find their position
           %within the trace in ms
                [pM,loM]=findpeaks(pk, 'MINPEAKHEIGHT',0.3);
                % pM represents the position of ridges within the
                %recording measured in samples
                %number of peak that corresponds to each of the
                %ridges
                ncrestas=size(loM,1); % variable 69
                SH=loc(loM)*1000/Fs; %Distance in ms since
                %the start of the vocalization
        catch err
                ncrestas=0;
        end;
        if (ncrestas==0) SH(1:9,1)=999999;
                pM(1:9,1)=999999;
        elseif (ncrestas <9) SH(ncrestas+1:9,1)=999999;
                %Creates the "SH" matrix with at least 9 rows
                pM(ncrestas+1:9,1)=999999;
        else
        end
        %===============Variables 69 to 88 ============
        matriz(filas,69)=ncrestas; % n ridges
        SH=SH';
        matriz(filas,70:78)=SH(1,1:9); %Distance in ms since
                %the start of the vocalization to ridges
        pM=pM';
        matriz (filas,79:87)=pM(1,1:9); %Power of the ridge
        matriz(filas,88)=MH; %Maximum frequency (see %Box3)
```

Box 7: finally, we added a final set of variables that can help define the croaks.

Variable 89: Dominant frequency calculated from the Fourier transform and a Hamming window. The dominant frequency is the frequency where the croak shows the highest intensity.

Variable 90: Intensity of the dominant frequency in dB.

Variable 91: Lowest frequency in the periodogram.

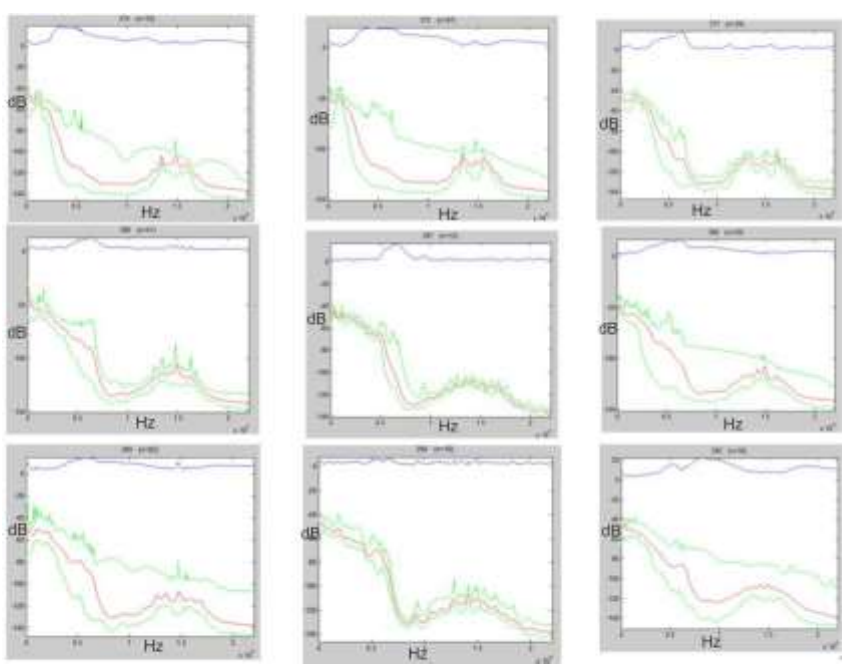
Variable 92: Intensity of the lowest frequency on the periodogram in dB.

Variable 93: Mean frequency calculated taking the values between the dominant frequency in dB and the frequency corresponding to 60 dB below the dominant frequency in the periodogram.

Variable 94: Signal-to-noise ratio (SNR) in dB, defined as the proportion between the power of the signal and the power of the background noise. All the harmonics, including the fundamental frequency, are excluded from the noise measurement. We used a modified periodogram with a Kaiser Window of β=38 for the calculation of SNR. This variable is important, as a large SNR ratio allows the transmission of vocalizations across larger distances.

Variable 95: Quality of the signal measured using the signal-to-noise and distortion ratio (SINAD). The SINAD is a parameter that measures the signal quality from disturbances like noise and distortion.

After the analysis, we created a complementary MATLAB matrix with the periodograms data of all the croaks from the same frog to examine the vocalization differences between specimens. We then built a series of periodograms for each specimen that included as the independent variable the values of the sampling rate in Hz (Variable 3, X axis, column 1), and as dependent variable the algorithm Bourgh_50 values in dB to calculate the mean intensity (column 2), the minimum intensity (column 3), the maximum intensity (column 4) and the intensity typical deviation (column 5) (Fig. 15).



```
% BOX 7   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          %-----Using the Fourier transform we get the dominant
          % frequency with the Kaiser modifier-------------
          [PerxR,FxR]=periodogram(xR,
kaiser(length(xR),38),(length(xR)),Fs);
          PxR=10*log10(PerxR);
          [pkfd,locfd]=findpeaks(PxR);
          M=[pkfd,FxR(locfd)];
          [i j]=find(pkfd==max(max((pkfd))));
          %===============Variables 89 to 92 ============
          matriz(filas,89)=M(i,2);  %Dominant frequency
          matriz(filas,90)=M(i,1); %Intensity in dB of the
                    %dominant frequency
          matriz(filas,91)=M(1,2); %Fundamental frequency
          matriz(filas,92)=M(1,1); %Intensity in dB of the
                    %fundamental frequency
          %============ Average frequency ==============
          CORTE=(M(i,1)-60)*(-1); %Intensity in dB plus 60 of the
                    %dominant frequency
          M1=[PxR,FxR];
          mconvertida=[(M1(:,1)+CORTE), M1(:,2)];
          mconvertida(mconvertida<0)=0; %Remove negative
          %values by converting them to 0
          NUM=sum(mconvertida(:,1).*mconvertida(:,2));
                                        %Multiple array
          DEN=sum(mconvertida(:,1));
          Fmedia=NUM/DEN;
          %===============Variables 93 to 95 ============
          matriz(filas,93)=Fmedia; %Average frequency
          matriz(filas,94)=snr(xR,Fs); %Signal-to-noise ratio
          matriz(filas,95)=sinad(xR,Fs); %Quality of the signal
          %in the presence of disturbances like nouse and distortion
          %_____________End of the process___________
   else % Of if number 3
   end % Of if number 3
          warning('off','all')
end %Of the first "for" loop
%_______Export the matrix to Excel in the absence of errors ____
if (filas >0)
if  (salir==0)
   filename2=strcat(file,'\',name,'_','matriz04.xls');
          %Save %the Excel file with the vocalization recordings
   filename2=strrep(filename2,' ',''); %Remove any
          %leftover blank spaces
   filename3=strcat(file,'\',name,'_','media_matriz04');
   filename3=strrep(filename3,' ','');
   xlswrite(filename2, matriz); %Exporting to Excel
   filename1=strcat(file,'\',name,'_','matriz04');
   filename1=strrep(filename1,' ','');
   save(filename1, 'matriz'); %Save matrix as Matlab matrix
```

```
        %******Creates “max”, “min” and “average” matrices ***
        %*******MATbase(1:nfila,1)=FB is the y axis******
        %****** MATbase(1:nfila,filas+1)=curva50; is the x******
  MATmedia(1:nfila,1)=FB;
  MATmedia(1:nfila,2)=mean(MATbase,2); %Average of the rows
  MATmedia(1:nfila,3)=min(MATbase,[],2);
  MATmedia(1:nfila,4)=max(MATbase,[],2);
  MATmedia(1:nfila,5)=std(MATbase,0,2);
  save(filename3, 'MATmedia'); %Saves data as a Matlab matrix
        nstr=int2str(n-4);
        if (n<10)
                numrana=strcat('00',nstr);
        else
                numrana=strcat('0',nstr);
        end;
  Nombre_del_fichero=strcat(file,'\',name,'_',numrana,'_','V.wav')
        %This gives us the last file opened
  filas  %”Rows” gives us the number of vocalizations that were
else
end;
elseif (filas==0)
        strcat('URL not found or the number of vocalization
incorrect')
else
end;
try
tit=strcat(name,'   (n=',int2str(filas),')');
plot(MATmedia(1:nfila,1),MATmedia(1:nfila,2),'r',MATmedia(1:nfil
a,1),MATmedia(1:nfila,3),'g',MATmedia(1:nfila,1),MATmedia(1:nfil
a,4),'g',MATmedia(1:nfila,1),MATmedia(1:nfila,5),'b');title(tit);axis
tight;
catch err
end
%________________ press enter ___________________________
```

Figure 15. Example of a series of periodograms which summarizes the vocalizations of different individuals of Rana temporaria. Mean intensity (red), the minimum intensity (green), the maximum intensity (green) and the intensity typical deviation (blue). The x-axis shows frequency in Hz whereas the y-axis shows intensity in dB. N is the croaks number in each vocalization.

The final result of the analysis was a matrix in both MATLAB and Excel formats with the data of the 95 studied variables.

We then processed all the variables with SPSS and calculated supplementary variables as the combination result of the 95 studied variables listed below:
Variable 69bis: modules (variable 69) divided by time ms (variable 4).
Variable 70bis: distance in ms between the first and second module of each croak (variable 71 and variable 70).
Variable 71bis: distance in ms between the second and third modules of each croak (variable 72 and variable 71).
Variable 96: distance in ms between the first and third module of each croak (addition of variable 70bis and 71bis).

In addition, we added seven other variables corresponding to the location, year, month and day of sampling as well as the species, snout-vent length and physical sex of the individual.

REFERENCES

Bee, M.A. (2004). Within-individual variation in bullfrog vocalizations: Implications for a vocally mediated social recognition system. *J. Acoust. Soc. Am.*, 116, 3770–3781.

Bee, M.A., Kozich, C.E., Blackwell, K.J. & Gerhardt, C.H. (2001). Individual variation in advertisement calls of territorial male green frogs, *Rana clamitans*: Implications for individual discrimination. *Ethology*, 107, 65-84.

Benade, A.H. (1976). *Fundamentals of musical acoustics*. London: Oxford University Press, New York, pp. (1-596).

Blumstein, D.T., Mennill, D.J., Clemins, P., Girod, L., Yao, K., Patricelli, G. *et al.* (2011). Acoustic monitoring in terrestrial environments using microphone arrays: Applications, technological considerations and prospectus. *J. Appl. Ecol.* 48, 758-767.

Duellman, W.E. & Trueb, L. (1994). *Biology of amphibians*. McGraw-Hill, New York, pp. (1-670).

Emerson, S.B., Boyd, S.K. (1999). Mating vocalizations of female frogs: Control and evolutionary mechanisms. *Brain Behav. Evol.*, 53, 187-197.

Endler, J.A., Westcott, D.A., Madden, J.R. & Robson, T. (2005). Animal visual systems and the evolution of color patterns: sensory processing illuminates signal evolution. *Evolution*, (NY) 59, 1795-1818.

Johnson, K. (2012). *Acoustic and Auditory Phonetics*. 3rd edition. Blackwell, Oxford 2003, pp. (1-232).

Ladich, F. & Winkler, H. (2017). Review: Acoustic communication in terrestrial and aquatic vertebrates. *J. Exp. Biol*., 220, 2306-2317.

Larom, D., Garstang, M., Payne, K., Raspet, R., Lindeque, M. (1997). The influence of surface atmospheric conditions on the range and area reached by animal vocalizations. *J. Exp. Biol*., 200, 421-31.

Panhuis, T.M., Butlin, R., Zuk, M. & Tregenza, T. (2001). Sexual selection and speciation. *Trends Ecol. Evol*., 16, 364-371.

Savage, R.M. (1962). *The ecology and life history of the common frog* (*Rana temporaria temporaria*). Hafner Pub. Co., New York, pp. (1-240).

Schrode, K. M., Buerkle, N. P., Brittan- Powell, E. F., Bee, M. A. (2004). Auditory brainstem responses in Cope's gray treefrog (*Hyla chrysoscelis*): effects of frequency, level, sex and size. *J. Comp. Physiol. A* 200, 221-238.

Wells, K.D. (2007). *The ecology and behavior of amphibians*. University of Chicago Press, Chicago, pp. (1-1148).

Wells, K.D. & Schwartz, J.J. (2007). *The behavioral ecology of anuran communication. In Hearing and sound communication in amphibians*. Springer, New York, NY, pp. (44-86).

Glossary of terms used in this study to describe Anuran vocalizations.

Formants	The peaks that are observed in the spectrum envelope (Benade 1976). Frequencies that are naturally enhanced within a vocalization due to the inherent anatomical and physical properties of the sound organs (Ladich & Winkler 2017).
Anti-formants	Also called anti-resonances, which show up as pronounced spectral valleys in the spectrum (periodogram) (Johnson 2012).
Fundamental frequency	Also known as first harmonic, it is the lowest frequency and highest intensity in a harmonic sound. The second and further harmonics are integer multiples of the fundamental frequency. The fundamental frequency of a vocalization is related to vibrating structures in the sound-producing organs (Ladich & Winkler 2017).
Croak	An acoustic unit of frog vocalization (croaking), a distinct sound; a

	croak is separated from other croaks by periods of silence. Savage (1962) also stated that the term call-note for the ordinary croak is not very appropriate.
Pulse	Defined in MATLAB as the period between two consecutive zero line crossings by the recorded waveform (Fig. 4).
Burg's algorithm	We can fit a curve in order to simplify the interpretation of the periodogram (Fig. 6). The Burg algorithm for segments provides more accurate models than any of the averaging methods (Waele & Broersen 2000). Burg algorithm 50 sensitivity (Burg_50) represents the frequencies modulation due to the inherent anatomical and physical properties of the sound organs. Burg algorithm 2 sensitivity (Burg_2) is the theoretical fundamental frequency of a vocalization related to vibrating structures in the sound-producing organs.
Distortion	The total harmonic distortion (THD MATALB function) is determined from the fundamental frequency and the first five harmonics using a modified periodogram of the same length as the input signal. The modified periodogram uses a Kaiser window with $\beta = 38$.
Module (new definition)	New variable defined in order to differentiate from pulses and peaks which defines a cluster of pulses (as defined in MATLAB) that increase and decrease uniformly integrating a group creating a wave in the oscillogram profile (Fig. 3). This is roughly equivalent to the concept of pulse as defined by Duellman and Trueb (1994). When ran by Sound Forge, each module sounds like a *Hyla chrysoscelis* click

	(Schrode 2014) so each croak is composed by several clicks.
Vocalization	Any kind of sound produced by animals by means of their respiratory system, typically by the action of vocal cords, independent of its categorization or structure
Mean frequency (Variable 89)	Calculated taking the values between the dB of the dominant frequency and the frequency corresponding to 60 dB below the dominant frequency in the periodogram.
Signal-to-noise ratio (SNR) (Variable 94)	Defined as the proportion between the power of the signal and the power of the background noise in dB. All the harmonics, including the fundamental frequency, are excluded from the noise measurement.