Measuring a call

Eleutherodactylus Research Studio · Fall 2026

Tom Devitt

2026-09-14

This week runs without me

I am at posit::conf Monday and Wednesday. Class still happens, on your own schedule, and the work is due on Canvas.

- **Monday:** read this deck (ten minutes), then do worksheet 06 on the pod. It walks you from a wav file to a measured, summarized call. Due **Tuesday Sep 15, 11:59 pm**
- **Wednesday:** team session. Each team measures its own recordings using the same steps. Directions are in Wednesday's Canvas assignment
- **Stuck?** Email me. I will answer between conference sessions, usually within a few hours. Ask your teammates first; four people reading the same error message solve it faster than one

From picture to number

Last Wednesday you read call duration, pulse count, and dominant frequency off a figure by eye. Today you get the same numbers from code, so that anyone can rerun them and get the same answer.

Two R packages do the work, both already installed on the pod:

- **seewave** does one step at a time and shows you each one. You will use it to learn what the numbers mean
- **warbleR** wraps the same steps into one function that returns a table. You will use it once you trust it

Raven Lite, a point-and-click program, does the same job on your laptop. It is optional this semester. Everything we grade comes from R on the pod.

The five steps

Every call description in the literature, including the one your team will write, comes down to this:

1. **Read** the recording into R
2. **Look** at it as a spectrogram and oscillogram
3. **Select** each call: write down where it starts and ends
4. **Measure** each selection: duration, interval, dominant frequency, pulses
5. **Summarize** across calls and across males: mean ± SD (range), *n*

Steps 1, 2, 4, and 5 are code and run the same way every time. Step 3 is judgment. It is where two people measuring the same recording disagree, so it is the step you document.

Step 1: read

```
library(tidyverse)
library(tuneR)
library(seewave)
library(warbleR)

w <- readWave("marnockii-series.wav")
w
duration(w)
```

A `Wave` object holds the column of numbers (44,100 per second) plus the sample rate and bit depth. `duration()` returns seconds. For an mp3, swap in `readMP3()`; everything after that is identical.

Step 2: look

```
spectro(w, wl = 512, ovlp = 75, flim = c(0, 5), osc = TRUE, scale = FALSE)
```

- `wl` is the window length from last week: 512 samples resolves frequency, 256 resolves pulses
- `flim` clips the frequency axis to where frogs live
- `osc = TRUE` adds the oscillogram underneath

To zoom in on one call, cut the piece out first and then plot the piece:

```
one <- cutw(w, from = 1.9, to = 2.4, output = "Wave")
spectro(one, wl = 256, ovlp = 85, flim = c(0, 5), osc = TRUE, scale = FALSE)
```

(Passing `from` and `to` straight to `spectro()` with `osc = TRUE` trips a bug in seewave. Cut first.)

Step 3: select

A **selection table** is a tidy table with one row per call and two columns you fill in from the spectrogram, start and end in seconds:

```
calls <- tibble(
  call  = 1:4,
  start = c(0.42, 1.99, ...),
  end   = c(0.76, 2.32, ...)
)
```

Reading these off the plot is the whole skill. Zoom until the pulses are individually visible and put start at the first pulse that stands clear of the background, end at the last. Then **write your rule down** in a sentence. Teammates who measure the other half of the recordings must use the same rule, or your numbers cannot be pooled.

Step 4: measure, by hand

```r
calls <- calls |>
  mutate(
    duration_s   = end - start,
    interval_s   = lead(start) - start,
    rate_per_min = 60 / interval_s
  )
```

Duration and interval are subtraction. `lead()` looks at the next row, so the last call has no interval (that is correct, not an error).

Dominant frequency needs the spectrum of each selection. A small function does it once; `map2_dbl()` applies it to every row:

```r
dom_freq_khz <- function(wave, from, to) {
  piece <- cutw(wave, from = from, to = to, output = "Wave")
  spec  <- meanspec(piece, wl = 512, plot = FALSE)
  fpeaks(spec, nmax = 1, plot = FALSE)[1, 1]
}
calls <- calls |>
  mutate(dom_khz = map2_dbl(start, end, \(s, e) dom_freq_khz(w, s, e)))
```

Step 4: measure, with warbleR

```r
sel <- selection_table(
  X = data.frame(sound.files = "marnockii-series.wav", selec = calls$call,
                 start = calls$start, end = calls$end),
  path = "~"
)
sa <- spectro_analysis(sel, path = "~", bp = c(1, 4), wl = 512)
sa |> select(selec, duration, meanpeakf, meandom, dfrange)
```

Same selection table, 27 columns back. The ones that map to Köhler's list:

- `duration`, the same subtraction you did
- `meanpeakf`, the dominant frequency (kHz), should match your `dom_khz`
- `meandom`, `dfrange`, how the dominant frequency moves within the call

`bp = c(1, 4)` is a band-pass filter in kHz: it tells warbleR to ignore the insects below 1 kHz. Report it in your methods, like the window length.

Step 5: summarize

```
calls |>
  summarise(n = n(),
            across(c(duration_s, dom_khz),
                   list(mean = mean, sd = sd, min = min, max = max)))
```

which becomes a sentence:

> Call duration 0.35 ± 0.01 s (0.33–0.36 s; *n* = 4 calls from 1 male); call rate 37.5 ± 0.8 calls/min; dominant frequency 2.15 kHz.

Two *n*s, always: calls and males. Four calls from one male tell you about that male. Your team's description needs several males, so Wednesday you split the recordings among you and pool with one shared rule.

Rules of the road

- **Document every knob.** Window length, band-pass, your start/end rule. Someone else must be able to reproduce your table
- **wav vs. mp3.** From mp3 files, report only durations, intervals, rates, counts, and dominant frequency. Skip amplitude and fine spectral measures
- **Temperature** goes in the table whenever it exists. A rate without a temperature is hard to compare
- **Recordings stay on the pod.** Measure there; export tables and figures, never audio
- **Listening.** The pod has no speakers. The practice clip plays on the Sep 9 deck. You can measure a call you have never heard; the spectrogram is the instrument. Ask me before moving any other audio off the pod

Monday's deliverable

Worksheet 06, on the pod:

```
file.copy("/stor/work/FRI_Bio206_TD_Spring2026/06-measuring-calls-worksheet.qmd", "~/")
```

You will measure the four calls in the practice clip, check yourself against warbleR, then measure **ten consecutive calls** from a 3-minute reference recording of a Texas *E. marnockii* male and write the summary sentence.

Submit the rendered HTML and the .qmd to Canvas by **Tuesday Sep 15, 11:59 pm**. Wednesday's assignment assumes you can do this.