# R as a sound system

**J. Sueur**[a]

[a]Muséum national d'Histoire naturelle
UMR CNRS-MNHN 7205 Origine Structure et Evolution de la Biodiversité
45 rue Buffon, 75005 Paris, France
sueur@mnhn.fr

**Mots clefs** : time series, acoustics, sound analysis, audio, time-frequency, sonification

Sound can be perceived everywhere at any time. In science, sound can be found in acoustics and in several other scientific and engineering disciplines, like musical acoustics, linguistics, speech and hearing sciences, psychoacoustics, bioacoustics, geology, noise control, and vibrations monitoring.

Handling sound with `R` is a rather easy task thanks to dedicated object classes directly deriving from `.wav` or `.mp3` audio file formats. It is possible to record, import, modify, and export audio objects with the packages `tuneR` [1] and `audio` [2]. Playing back a sound is mainly achieved by calling an external media player.

Sound can be analysed using `tuneR` and `seewave` [3] packages. These packages offer a set of complementary functions that can be used to extract and compare relevant amplitude, temporal, phase, and frequency parameters. For instance, Linear Predictive Coding, Fourier decomposition, Hilbert transform, cepstral analysis, or zero-crossing estimation are some of the techniques available to describe the frequency content of a sound. Figure 1 shows an example of a sound visualisation through a spectrographic representation with the tracking of both fundamental and dominant frequency bands. Correlation and distance functions can also be used to assess differences between pairs of amplitude envelopes or pairs of frequency spectral profiles. The use of the package `seewave` is now particularly important in bioacoustics – a discipline of life sciences focusing on animal sound – as it allows batch analyses of numerous audio files. This was in particular the case of studies that monitored the soundscapes of tropical forests and that returned up to 90,000 audio files to be analysed.

In addition to analysis, `tuneR` and `seewave` can generate sound by sinusoidal synthesis. Combined with arithmetic operations, frequency filtering and amplitude shaping, sinusoidal synthesis is simple but efficient and can produce signals that copy relatively well natural signals. Another side of sound synthesis is now available thanks to two new packages, `playitbyr` [4] and `audiolyzR` [5], that are dedicated to sonification [6]. Sonification is a type of auditory display aimed to communicate information. If visualization is the process of mapping data onto a graphic, then sonification can be seen as the process of transforming data into non-speech sound. In practical this means that you can listen to the crabs or iris data sets. `playitbyr`, which follows `ggplot2` syntax, maps data onto sonic parameters like pitch, tempo, and rhythm. Similarly, `audiolyzr` can generate audio representations of common plots accompanied with a pop-up panel to control interactively main sound parameters.

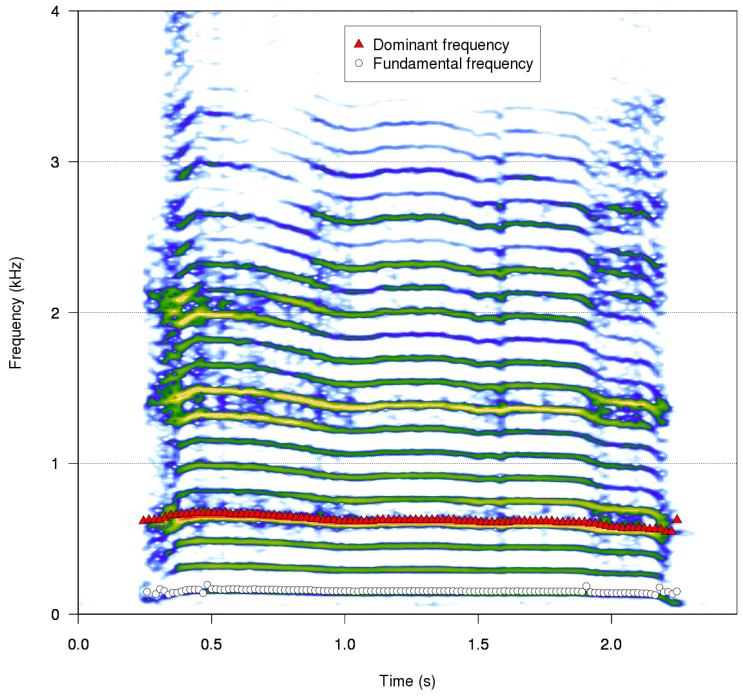


Figure 1: A call produced by a bird (*Vanellus vanellus*): a spectrographic representation with fundamental and dominant frequencies tracked. Produced with the `seewave` functions `spectro()`, `dfreq()`, and `fund()`. Electronic version with colours.

**Références**

[1] Ligges U. (2011). tuneR: analysis of music. R package version version 0.4-1. http://r-forge.r-project.org/projects/tuner/

[2] Urbanek S. (2011). audio: audio interface for R. R package version 0.1-4. http://CRAN.R-project.org/package=audio

[3] Sueur J., Aubin T., Simonis C. (2008). Seewave: a free modular tool for sound analysis and synthesis. *Bioacoustics*, **18**, 213-226.

[4] Brown E. (2012). playitbyr: representing and exploring data through sound. R package version 0.2-1. http://CRAN.R-project.org/package=playitbyr

[5] Stone E., Garrison J. (2013). audiolyzR: give your data a listen. R package version 0.4-9. http://CRAN.R-project.org/package=audiolyzR

[6] Hermann T., Hunt A., Neuhoff J. G. (2011). *The sonification handbook*. Logos Publishing House, Berlin, 586 p.