DOI: 10.61782/fa.2023.1106

forum acusticum 2023

LIVE-TRACKING ACOUSTIC PARAMETERS IN ANIMAL BEHAVIOURAL EXPERIMENTS: INTERACTIVE BIOACOUSTICS WITH PARSELMOUTH

Yannick Jadoul[1*] **Diandra Duengen**[1] **Andrea Ravignani**[1,2,3]

[1] Comparative Bioacoustics Group, Max Planck Institute for Psycholinguistics, Nijmegen, the Netherlands

[2] Department of Human Neurosciences, Sapienza University of Rome, Rome, Italy

[3] Center for Music in the Brain, Department of Clinical Medicine, Aarhus University & The Royal Academy of Music Aarhus/Aalborg, Denmark

ABSTRACT

Most bioacoustics software is used to analyse the already collected acoustics data in batch, i.e., after the data-collecting phase of a scientific study. However, experiments based on animal training require immediate and precise reactions from the experimenter, and thus do not easily dovetail with a typical bioacoustics workflow.

Bridging this methodological gap, we have developed a custom application to live-monitor the vocal development of harbour seals in a behavioural experiment. In each trial, the application records and automatically detects an animal's call, and immediately measures duration and acoustic measures such as intensity, fundamental frequency, or formant frequencies. It then displays a spectrogram of the recording and the acoustic measurements, allowing the experimenter to instantly evaluate whether or not to reinforce the animal's vocalisation.

From a technical perspective, the rapid and easy development of this custom software was made possible by combining multiple open-source software projects. Here, we integrated the acoustic analyses from Parselmouth, a Python library for Praat, together with PyAudio and Matplotlib's recording and plotting functionality, into a custom graphical user interface created with PyQt. This flexible recombination of different open-source Python libraries allows the whole program to be written in a mere couple of hundred lines of code.

Keywords: *behavioural experiments, vocal development, interactive software*

**Corresponding author*: *Yannick.Jadoul@mpi.nl.*

1. INTRODUCTION

When training animals, swift and correct feedback is of the utmost importance to achieve learning. A typical bioacoustics workflow however often includes a constraint conflicting with swift feedback: the data is analysed in batch, only after a full dataset has been collected. Most bioacoustics software tools will load the audio from a previously made recording before providing the opportunity to scrutinise the data and run the appropriate acoustic analyses.

For interactive experiments on vocal behaviour, this difference in approach between bioacoustics and operant conditioning animal training is a non-trivial challenge. Past research has dealt with this in various ways. One approach is to visually inspect a live spectrogram during training [1] or manually monitor a recording [2], and afterwards analyse the animal's recorded responses in full detail. A potential disadvantage here is that it can be hard or impossible to get measurements during training which are as precise as the following in-depth analysis. An alternative approach is to include the acoustic analysis as a part of a custom-tailored, fully-automated experimental setup [3–5]. Such an approach does not have the same

drawback, as vocalisations are measured and assessed automatically, and makes it easy to run many trials. However, it might not be feasible for all species and experimental settings. For example, a fully automatic methodology lacks the social interaction between trainer and animal, which might be beneficial — or even essential — for the animal's motivation to participate. Moreover, more upfront effort is necessary to make the experiment safe and fail-proof. This is especially the case for larger animal species, which also require a proportionally larger and more expensive setup.

Here, we present a third approach to perform interactive acoustics analyses during animal training, which delivers a compromise between the two previous approaches: We created a custom, interactive software tool to assist the experimenter and instantaneously analyse the recorded vocalisations. The core acoustic functionality of this tool is provided by Parselmouth.

2. PARSELMOUTH IN BIOACOUSTICS

Parselmouth is an open-source Python library for Praat [6, 7]. Praat is a popular software package originally developed to perform phonetic analysis and speech processing, but its wide range of acoustic algorithms and analyses have seen use in scientific fields beyond phonetics and linguistics. More specifically, Praat and Parselmouth have been used in a multitude of bioacoustics studies [8]. The algorithms implemented in Praat are particularly, though not exclusively, applicable to mammalian bioacoustics: Methods for tracking fundamental frequency, estimating formants, assessing voice quality, etc. are every bit as relevant to the field of bioacoustics as they are in phonetics.

As a Python library, one important advantage of Parselmouth is the possibility to combine Praat's functionality with the huge variety of other Python libraries. For example, the Python "scientific ecosystem", with packages such as NumPy [1], SciPy [2], Pandas [3], Matplotlib [4], and scikit-learn [5], provides a whole arsenal of tools which can complement Parselmouth in an acoustic data analysis pipeline. In addition, less obviously pertinent software packages such as PyQt [6] provide an accessible way to develop a simple graphical user interface (GUI). In combination with a specialised library for audio recording and playback (e.g., PyAudio [7]), all separate components are readily available to swiftly create a custom-created experimental software tool in Python.

3. PARSELMOUTH IN A BEHAVIOURAL EXPERIMENT

To study the vocal learning capabilities of harbour seals (*Phoca vitulina*) [9], we created a custom tool in Python to assist in the training procedure (Fig. 1). Our application complements the experimenter's role in three main steps:

1. At the start of a trial, the experimenter presses and holds down the keyboard's space bar or a USB foot pedal, then presents the animal the cue to vocalise. As long as the key is pressed, the application continuously records the microphone's audio (through PyAudio).
2. Once the space bar is released, Parselmouth takes the just-recorded audio fragment, filters out background noise with Praat's noise reduction algorithm [8], and detects the seal's call by finding the longest non-silent interval [9].
3. Finally, the acoustic parameter of interest is measured with Parselmouth (e.g., the change in fundamental frequency), and displayed to the experimenter.

In short, after the animal stops vocalising and the experimenter releases the recording button, the application displays all the necessary information to decide on whether to reward the animal or not. Throughout, the experimenter is in control of the procedure and keeps a one-on-one connection to motivate and reinforce the participating animal.

We added several additional pieces of functionality to the application, in order to further assist the experimenter. The application will check whether an extracted measurement reached a pre-set threshold and display this to the experimenter. After each trial, the recorded audio is saved,

[1] https://numpy.org/
[2] https://scipy.org/
[3] https://pandas.pydata.org/
[4] https://matplotlib.org/
[5] https://scikit-learn.org/
[6] https://www.riverbankcomputing.com/software/pyqt/
[7] https://people.csail.mit.edu/hubert/pyaudio/
[8] https://www.fon.hum.uva.nl/praat/manual/Sound__Remove_noise___.html
[9] https://www.fon.hum.uva.nl/praat/manual/Sound__To_TextGrid__silences____.html

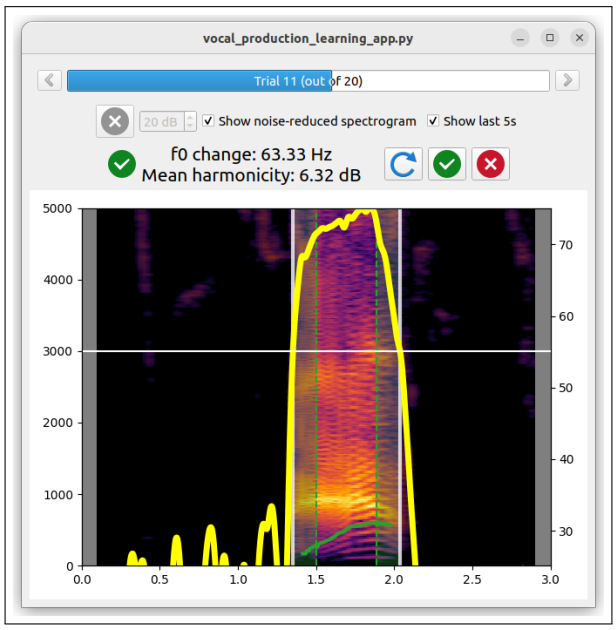

Figure 1. Our custom-written graphical application records and immediately analyses the acoustic parameter of interest. Here, using Parselmouth, the application detects the start and end of a harbour seal's recorded call, estimates the change in fundamental frequency throughout the call, and displays the call's spectrogram and estimated fundamental frequency curve.

and the result is recorded in a table, removing the need to manually keep track of the animal's performance during the training session.

In a closely related experiment [9, 10], we randomised the presentation order of stimuli with PyGellermann [11], and played back white noise through headphones with PyAudio to ensure double-blind presentation of the stimuli.

While we are currently still processing the collected data and subsequently analysing the results, our custom application facilitated the experimental setup and data collection. Whereas this qualitative judgment is inherently subjective and highly dependent on the experiment and experimenter, our experience with the above approach shows the possible advantages of developing custom, live-tracking experimental software. Moreover, it demonstrates the role that Parselmouth can play in implementing the bioacoustical aspects of such an application.

4. DISCUSSION

Acoustic analysis is a crucial aspect of experiments into the vocal behaviour of animals. Here, we have demonstrated how Parselmouth can enable the creation of new software tools for behavioural experiments by integrating Praat's acoustic processing into a larger Python context. In the presented case study, the acoustic analysis is closely integrated with libraries that provide a graphical user interface, audio recording, and plotting functionality. However, the possible combinations of Python libraries are endless. Mixing and matching already existing Python code and packages enables rapid and effective development of software which is fine-tuned to the experiment at hand.

5. ACKNOWLEDGMENTS

The authors wish to thank Robert, Elektra, and Jannik for being excellent beta testers with beautiful vocalisations. YJ, DD, and AR were supported by Max Planck Independent Group Leader funding awarded to AR.

6. REFERENCES

[1] A. L. Stansbury and V. M. Janik, "Formant modification through vocal production learning in gray seals," *Current Biology*, vol. 29, no. 13, pp. 2244–2249, 2019.

[2] M. M. Holt, D. P. Noren, R. C. Dunkin, and T. M. Williams, "Vocal performance affects metabolic rate in dolphins: implications for animals communicating in noisy environments," *The Journal of Experimental Biology*, vol. 218, no. 11, pp. 1647–1654, 2015.

[3] E. Z. Lattenkamp, S. C. Vernes, and L. Wiegrebe, "Vocal production learning in the pale spear-nosed bat, *Phyllostomus discolor*," *Biology letters*, vol. 16, no. 4, p. 20190928, 2020.

[4] M. S. Osmanski and R. J. Dooling, "The effect of altered auditory feedback on control of vocal production in budgerigars (*Melopsittacus undulatus*)," *The Journal of the Acoustical Society of America*, vol. 126, no. 2, pp. 911–919, 2009.

[5] K. Manabe, R. J. Dooling, and E. F. Brittan-Powell, "Vocal learning in budgerigars (*Melopsittacus undulatus*): effects of an acoustic reference on vocal matching," *The Journal of the Acoustical Society of America*, vol. 123, no. 3, pp. 1729–1736, 2008.

[6] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, pp. 1–15, 2018.

[7] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [Computer program]," 2021.

[8] Y. Jadoul, B. de Boer, and A. Ravignani, "Parselmouth for bioacoustics: Automated acoustic analysis in Python," in review.

[9] D. Duengen, "Phocal learning – are harbor seals vocal learners?," in *the Rhythm, Music, and Animal Communication Symposium*, (Nijmegen, the Netherlands), 19 January 2023.

[10] D. Duengen and A. Ravignani, "Social learning in a solitary pinniped: The harbor seal," in *the European Conference on Behavioural Biology (ECBB 2022)*, (Groningen, the Netherlands), 20–23 July 2022.

[11] Y. Jadoul, D. Duengen, and A. Ravignani, "PyGellermann: A Python tool to generate pseudorandom series for human and non-human animal behavioural experiments," *BMC Research Notes*, in press.