



Parselmouth for bioacoustics: automated acoustic analysis in Python

Yannick Jadoul, Bart de Boer & Andrea Ravignani

To cite this article: Yannick Jadoul, Bart de Boer & Andrea Ravignani (13 Oct 2023): Parselmouth for bioacoustics: automated acoustic analysis in Python, Bioacoustics, DOI: 10.1080/09524622.2023.2259327

To link to this article: <https://doi.org/10.1080/09524622.2023.2259327>



© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



View supplementary material [↗](#)



Published online: 13 Oct 2023.



Submit your article to this journal [↗](#)



Article views: 773



View related articles [↗](#)



View Crossmark data [↗](#)

Parselmouth for bioacoustics: automated acoustic analysis in Python

Yannick Jadoul ^{a,b}, Bart de Boer ^{b#} and Andrea Ravnani ^{a,c,d#}

^aComparative Bioacoustics Group, Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands; ^bArtificial Intelligence Lab, Vrije Universiteit Brussel, Elsene, Belgium; ^cCenter for Music in the Brain, Department of Clinical Medicine, Aarhus University & The Royal Academy of Music Aarhus/Aalborg, Aarhus, Denmark; ^dDepartment of Human Neurosciences, Sapienza University of Rome, Rome, Italy

ABSTRACT

Bioacoustics increasingly relies on large datasets and computational methods. The need to batch-process large amounts of data and the increased focus on algorithmic processing require software tools. To optimally assist in a bioacoustician's workflow, software tools need to be as simple and effective as possible. Five years ago, the Python package Parselmouth was released to provide easy and intuitive access to all functionality in the Praat software. Whereas Praat is principally designed for phonetics and speech processing, plenty of bioacoustics studies have used its advanced acoustic algorithms. Here, we evaluate existing usage of Parselmouth and discuss in detail several studies which used the software library. We argue that Parselmouth has the potential to be used even more in bioacoustics research, and suggest future directions to be pursued with the help of Parselmouth.

ARTICLE HISTORY

Received 3 May 2023
Accepted 22 August 2023

KEYWORDS

Acoustic analysis; data processing; software; Praat; Python

Introduction

A bioacoustician's job lies at the intersection of many different skill sets. As for any empirical scientist, collecting data and designing experiments to test hypotheses are essential. The nature of acoustic data also requires a bioacoustician to be technically adept, both when choosing experimental equipment and when analysing the collected recordings. Especially the latter part has become increasingly important: advances in computational power and storage capacity have created new possibilities to collect larger acoustic datasets and apply increasingly complex analyses. Key skills in bioacoustics now include managing and processing lots of data and automating these processes through scripting. With some exceptions, these skills are typically not the main training of the classical empirical scientist and considered incidental to the actual research question. As such, any help in automating and smoothing out the data processing aspect can help a bioacoustician focus on the scientific questions at hand.

Five years ago, Jadoul, Thompson, and de Boer released Parselmouth¹ (Jadoul et al.

CONTACT Yannick Jadoul  Yannick.Jadoul@mpi.nl

[#]Co-senior authors

This article has been corrected with minor changes. These changes do not impact the academic content of the article.

 Supplemental data for this article can be accessed online at <https://doi.org/10.1080/09524622.2023.2259327>

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

2018), an open-source Python library for Praat (Boersma 2001, Boersma & Weenink 2023). Praat is a speech analysis software package which is widely used within phonetics and other linguistics disciplines. Crucially, the wide range of acoustic algorithms in Praat includes analyses which are just as relevant and applicable to other bioacoustics research. These analyses include, for example, estimating fundamental frequency and formants, or determining a signal's intensity or harmonics-to-noise ratio. Various past bioacoustics studies have successfully relied on Praat for acoustic analyses (e.g. McComb et al. 2014; Fitch et al. 2016).

From the onset, Parselmouth's goal has been to provide a full-fledged Python library that integrates into the larger Python scientific ecosystem easily and efficiently. This way, Parselmouth aims to both allow experienced Praat users to combine their work with the large range of scientific tools available for Python, as well as provide access to Praat's functionality for any user of Python unfamiliar with Praat. Python is a reasonably simple, easily learnable (Bogdanchikov et al. 2013; Mészárosóvá 2015), cross-platform, and extensible programming language with many built-in libraries. These strengths make it well-suited to act as a *glue language* between different software libraries and programming languages.^{2,3} Parselmouth creates the possibility to further integrate Praat's functionality with other libraries and thereby simplify and optimise workflows in a single programming environment. At the same time, each released version of Parselmouth directly corresponds to a specific Praat version,⁴ and produces the exact same numerical results.

There is a considerable amount of overlap between the typical challenges faced during data analysis by phoneticians and by bioacousticians. For example, researchers from both fields often need to perform the same measurements on all audio fragments in a dataset, segment recordings, cluster and classify audio fragments into categories. As a result, tools originally developed in the context of phonetics – such as Praat and Parselmouth – are relevant beyond human speech. This is particularly the case for one taxonomic group: mammalian vocal production mechanisms are, almost always, similar to ours (Elemans et al. 2015). And since both fields have moved beyond manual extraction and copy-pasting of data into spreadsheets, solutions for batch processing of data are not only useful, but crucial (Rocha et al. 2015; Beguš et al. 2023).

In this article, our ambition is threefold. First, we aim at demonstrating the broad range of applications and suitability of Parselmouth for bioacoustics research. Second, we describe in which contexts it most often gets used, and in which others it could be fruitfully used. Third, we want to evaluate whether and how Parselmouth has reached the aims set out during its conception and development (Jadoul et al. 2018). We present a handful of practical, hands-on examples of past (bio)acoustics projects that have used Parselmouth, and show which Praat functionality these projects accessed via Python. We then use these projects as case studies to evaluate three main goals of Parselmouth, and answer the following questions:

- Was the development of research methods or analyses more time efficient because of Parselmouth?
- Did the development benefit from the integration of Praat functionality with Python libraries and tools?
- Was the computation or data access more efficient because of the interface between Praat and Python?

The diversity of bioacoustics research is such that it cannot be appropriately captured by a couple of case studies. Therefore, after presenting case studies which provide a practical, in-depth perspective on the use of Parselmouth in bioacoustics, we give a broad, synthetic overview of other published research which used Parselmouth.

Case studies

Batch-analysis of large datasets: 'Vocal plasticity in harbour seal pups'

The algorithm to track the fundamental frequency is one of Praat's key pieces of functionality (Boersma, 1993). Torres Borda et al. (2021) used Parselmouth to test for vocal plasticity of harbour seal (*Phoca vitulina*) pups, and in particular the effect of background noise on the fundamental frequency (f_0) of their vocalisations. Environmental noise from the Wadden Sea was band-pass filtered between 250 and 500 Hz to mask the seal pups' typical range of f_0 . This noise was played back at different intensity levels in randomised orders of 5-minute chunks, and meanwhile the pups' vocalisations were recorded. In response to the high intensity noise playback, the pups adapted their calls by lowering the fundamental frequency. Moreover, some individuals showed an increase of their call amplitude and a shift in the spectral tilt of the calls.

Accessed Praat functionality

Parselmouth was used to analyse the acoustic properties of the seal pup calls recorded during the experiment. More specifically, sound recordings consisted of long files containing multiple calls. The onset and offset of calls were manually annotated with Praat TextGrids, using Praat's graphical user interface (GUI). Then, a Python script processed all recordings and saved acoustic parameters into a table. In this script, Parselmouth took up the role of reading

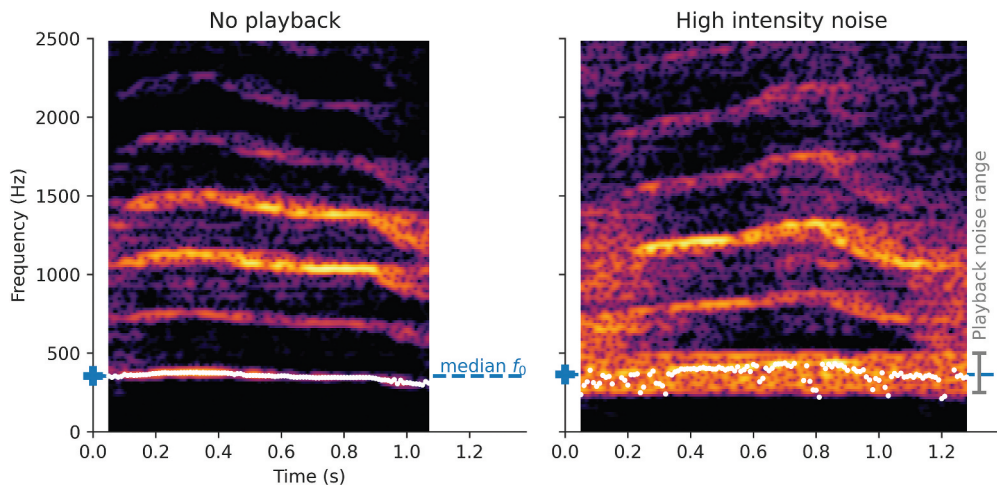


Figure 1. A plot of the fundamental frequency estimated through Parselmouth, overlaid on two seal call spectrograms, shows Praat's ability to correctly track the fundamental frequency, even in the presence of noise. Automated generation of such plots was used to swiftly assess the tracking quality, as to ensure correctness of the results.

the audio files, of extracting the individual calls and their duration based on the annotations, and – most importantly – of estimating the calls’ mean f_0 (Figure 1).

The central part of the Python script for analysis revolves around the application of the `parselmouth.Sound.to_pitch_ac()` method to each call’s sound fragment:

```
def calculate_pitch(snd_part):
    pitch = snd_part.to_pitch(time_step=TIME_STEP,
                              pitch_floor=200, pitch_ceiling=800)
    pitch_track = pitch.selected_array
    pitch_track = pitch_track[pitch_track['frequency'] != 0]

    if len(pitch_track) == 0:
        return np.nan, np.nan, np.nan
    else:
        median_pitch = np.nanmedian(pitch_track['frequency'])
        median_pitch_i = np.argmin(abs(pitch_track['frequency'] -
median_pitch))
        median_pitch_strength = pitch_track['strength'][median_pitch_i]
        pitch_iqr = scipy.stats.iqr(pitch_track['frequency'])
        assert median_pitch >= 200 or np.isnan(median_pitch)
        return median_pitch, median_pitch_strength, pitch_iqr
```

Code fragment 1: A short Python function wraps the functionality to estimate the median fundamental frequency using Parselmouth. Note the seamless interaction between Praat functionality (e.g. `snd_part.to_pitch(...)`), pure-Python syntax (e.g. the function definition or if-statement), and other Python libraries (i.e. NumPy’s `np.nanmedian` and SciPy’s `scipy.stats.iqr`) (see Supplemental online material, `calculate_pitch.py`, https://figshare.com/articles/dataset/Parselmouth_for_bioacoustics_automated_acoustic_analysis_in_Python/24307391?file=42678629).

This function is called repeatedly, as the main code loops over a table, loads the audio fragments from disk, and combines all pitch analysis results into a single table of results.

Note that the usage of `to_pitch_ac()` directly corresponds to the ‘Sound: To Pitch (ac)...’⁵ action in the Praat GUI or in a Praat script: this Python function accepts the same parameters as Praat, and produces exactly the same results (Parselmouth uses Praat’s underlying C/C++ code; Jadoul et al., 2018). One advantage of this is that it allows for a hybrid workflow combining the strengths of both Praat and Python. One can first use Praat to manually investigate a limited subset of vocalisations, finetune the pitch analysis parameters to the species or even individuals, and ensure the fundamental frequency can be adequately tracked. Thanks to this direct correspondence between Parselmouth and Praat, the right parameter settings, once determined, can easily be transferred into a larger Python analysis script to process all data in batch.

The use of Parselmouth benefited this project in the following ways:

- **Development efficiency:** Decomposing the whole script into various custom Python functions reduced the development complexity. Similarly, the use of Python data structures such as lists and dictionaries made development easier. While this script could have been written in the Praat scripting language, the ability to use the Python data and control structures made development subjectively more efficient.
- **Python integration:** The Pandas Python library was used to both read the table with the recordings’ metadata, and to write the results. Reading and writing tables is perfectly feasible within a Praat script. However, previous knowledge of the widely-

used Pandas library sped up development, allowed to focus on the problem at hand, and eliminated the risk of errors when learning to use new tools.

- **Computational efficiency:** Given the simple nature of the analysis script, Parselmouth and Python did not offer substantial gains in efficiency compared to using the Praat scripting language. Python and Pandas, being specialised software, are implemented more efficiently than the corresponding Praat functionality, but as the acoustic analyses made up most of the computation, any difference due to file writing and bookkeeping would be imperceptible. However, in a situation where the analysis takes much longer to run, Python's built-in multiprocessing module would be able to run the analysis in parallel with only minimal extra effort, something which (at the time of writing) is impossible to do in Praat.

Torres Borda et al. (2021) did not use Parselmouth to prepare the experiment's playback. However, Parselmouth could have easily filtered and normalised the recorded noise, and created the randomised playbacks in a single go. We are currently performing the same experiment by Torres Borda et al. (2021) in another pinniped species, the grey seal (Leonetti et al. 2022). Setting up this new experiment has shown us the value of programming the automatic generation of stimuli: An existing script would have saved time when adapting the experiment to a second species. In addition, a script to generate playbacks would have provided an exact record of the stimulus manipulation, increasing reproducibility and traceability of a scientific experiment.

The code to generate the playbacks in (Leonetti et al. 2022) is not complex. It only uses Parselmouth to load, create, concatenate, and save audio files, something any other audio library could do. In its simplicity, the script still combines Parselmouth with two other Python libraries (NumPy⁶ and Pandas⁷) and makes the stimulus generation reproducible and reusable.

```
NOISE_PARTS = {
    'silence': parselmouth.Sound("silence_5minutes.wav"),
    'low': parselmouth.Sound("low_noise_5minutes.wav"),
    'high': parselmouth.Sound("high_noise_5minutes.wav")
}
START_SOUND = parselmouth.Sound("start.wav")
START_SILENCE_DURATION = 2 * 60
N_REPETITIONS = 3
TARGET_SAMPLING_FREQUENCY = 16000

def contains_consecutive_elements(a):
    return np.any(a[:-1] == a[1:])

def next_playback_order(rng):
    noise_part_names = list(NOISE_PARTS.keys())
    noise_part_idx = np.repeat(np.arange(len(noise_part_names)), N_REPETITIONS)
    while contains_consecutive_elements(noise_part_idx):
        rng.shuffle(noise_part_idx)
    return [noise_part_names[i] for i in noise_part_idx]
```

```

def create_next_playback(order):
    sf = START_SOUND.sampling_frequency
    extra_start_silence_duration = START_SILENCE_DURATION -
START_SOUND.duration
    n_extra_start_silence_samples = int(sf * extra_start_silence_duration)
    extra_start_silence_samples = np.zeros(n_extra_start_silence_samples,
dtype=float)
    extra_start_silence = parselmouth.Sound(extra_start_silence_samples, sf)
    sounds = [NOISE_PARTS[name] for name in order]
    concatenated = parselmouth.Sound.concatenate(
        [START_RECORDING_SOUND, extra_start_silence] + sounds)
    resampled = concatenated.resample(TARGET_SAMPLING_FREQUENCY)
    return resampled
rng = np.random.default_rng(42)
playback_orders = pd.DataFrame()
for i in range(N_PLAYBACKS):
    playback_order = next_playback_order(rng)
    playback_sound = create_next_playback(playback_order)
    playback_sound.save(f"playback_{i}.wav", 'WAV')

    playback_order_df = pd.DataFrame({
        'playback_name': f"playback_{i}",
        'noise_part_i': list(range(len(playback_order))),
        'noise_part': playback_order
    })
    playback_orders = playback_orders.append(playback_order_df,
ignore_index=True)

playback_orders.to_csv("playback_orders.csv", index=False)

```

Code fragment 2: The use of Python and Parselmouth renders stimulus generation reusable and reproducible. Python functions help to compartmentalise the code, and Python's built-in data structures (i.e. lists and dictionaries) allow the programmer to reuse existing programming experience and paradigms in combination with Parselmouth's functionality (see Supplemental online material, `generate_playbacks.py`, https://figshare.com/articles/dataset/Parselmouth_for_bioacoustics_automated_acoustic_analysis_in_Python/24307391?file=42678626).

This case study demonstrates two main points. First, it illustrates one feature where Parselmouth was not used but it could have been: namely, the authors could have generated playback stimuli with a script, and Parselmouth could have easily performed the audio data wrangling. In fact, this choice would have made repeating the experiment and generating stimuli much easier when testing a second species. Second, for the analysis part in the second species the existing code by Torres Borda et al. (2020) could be reused, saving time. Automated analysis is not a requirement to do bioacoustics research (for approaches relying more on manual analyses rather than scripting see e.g. Rose et al. 2018; Abur et al. 2018; Wirth and Warren 2020; Nunes et al. 2021); nonetheless, our case study suggests how the upfront investment of writing a Python or Praat script for automated analysis can pay off as the amount of collected data increases or more species get tested.

Evaluation of computational simulations: 'Discovering articulatory speech targets from synthesized random babble'

Most mammalian vocalisations can be modelled by the source-filter theory of sound production. Originally developed to model human speech, the framework explains how the pitch and type of vowel in human speech can vary independently, to a first approximation. The source-filter theory has also been applied to study non-human

vocalisations, where formants and formant range are important bioacoustic measures (Riede et al. 2005; Taylor and Reby 2010; Taylor et al. 2016; Fitch et al. 2016; Boë et al. 2017). Here, we discuss a study on formants in human voice.

Rasilo and Jadoul (2020) used Parselmouth to apply Praat's formant-tracking algorithm (Burg algorithm; Childers 1978) to assess the range of formants in synthesised speech. More specifically, the study first proposed a learning algorithm to explore the articulatory space of a randomly babbling 2D vocal tract model (the LeVI acoustic model; Rasilo et al. 2013), aiming to cover as much as possible of the associated acoustic space of the synthesised random utterances. This algorithm does indeed result in a more acoustically varied synthesised speech, as the observed range of the first two formants (F1 & F2) was significantly larger than that of a non-learning, randomly babbling instance of the same vocal tract model.

Accessed Praat functionality

Parselmouth was used to estimate F1 and F2 frequencies of all the vowels of the synthesised speech, immediately calculate the 2D convex hull encompassing all (F1, F2) points in each sample of speech, and calculate the area of each convex hull. The following fragment shows how little code is necessary to link together these different parts of the analysis:

```
LEARNED = 'LEARNED'
RANDOM = 'RANDOM'
N = 5

memory = joblib.Memory('cache/')

def get_audio_fn(i, what):
    if what == LEARNED:
        return f'data/final_speech_{i+1}.wav'
    elif what == RANDOM:
        return f'data/final_speech_random{i+1}.wav'
    raise ValueError(what)

def get_times():
    return [0.45 + i * 6.41 + j * 0.6 for i in range(200) for j in range(10)]

@memory.cache
def get_formants(i, what):
    fn = get_audio_fn(i, what)
    formant = parselmouth.Sound(fn).to_formant_burg()
    times = get_times()
    return [tuple(formant.get_value_at_time(f, t) for f in (1, 2)) for t in
            times]

for what in [LEARNED, RANDOM]:
    for i in range(N):
        formants = get_formants(i, what)
        F1, F2 = list(map(list, zip(*formants)))
        hull = scipy.spatial.ConvexHull(formants)
        print(what, i, hull.area)
```

Code fragment 3: Parselmouth's formant analysis and SciPy's convex hull calculations are combined in a handful of lines of Python code to calculate the 2-dimensional formant range of synthesised speech fragments (see Supplemental online material, `plot_for_mant_triangles.py`, https://figshare.com/articles/dataset/Parselmouth_for_bioacoustics_automated_acoustic_analysis_in_Python/24307391?file=42678623).

- **Development efficiency:** Glueing together the formant analysis, convex hull calculation, and plotting (not shown in the above fragment) made development easier by not having to switch context between different scripts. Moreover, the Python Joblib⁸ library provides an easy, plug-in solution which can cache the result of running a function on a set of parameters (known as ‘memoization’). Through use of this functionality in the Joblib Python library, the same slow calculation (i.e. the formant extraction) did not get repeated over and over again when updating other parts of the code during development. During the typical alternation of editing and running code, this improved efficiency while programming.
- **Python integration:** Calculating the convex hull of a set of points is a non-trivial operation to implement, but common enough for a scientific Python library such as SciPy⁹ to provide. Parselmouth makes it possible to directly feed Praat’s results into the convex hull library. In a different part of the project, SciPy’s `scipy.io.loadmat` provided easy access to the MATLAB files generated by the LeVI vocal tract model.
- **Computational efficiency:** As it is using the C++ code underlying Praat, the formant analysis is equally efficient when using Parselmouth as when using Praat directly. However, the Joblib library’s memoization cache makes successive calls during development (see above) much more efficient, eliminating repetitive calculation in the computationally most expensive part (i.e. the formant analysis). Being able to do the formant analysis, convex hull calculation, and stats and plotting in the same Python script facilitates data access and management by not having to save and read a file with formant values.

To conclude, by dealing with formant extraction in human voice, this example showcases Parselmouth’s potential in working with acoustic parameters key in mammalian vocalisations (formants) while using the ‘model species’ for which formants are best studied and understood (humans). Notice that here we deal with formants in speech, but formants are also extremely important in human non-speech bioacoustics, such as song, laughter, etc (Pisanski et al. 2016a; Pisanski et al. 2016b; Keller et al. 2017; Anikin et al. 2022). This example also shows Praat’s and Parselmouth’s potential in extracting acoustic features from synthesised, rather than recorded, sounds.

Finally, this case study again illustrates the importance of automatic and replicable pipelines. In this specific case of two interacting steps, simulating data and analysing data, a more traditional ‘point and click’ analysis pipeline would have resulted in immense loss of time, as it would have had to be performed again for every change in parameters in the data simulation step. On the contrary, a single automated pipeline makes it easy to update results and plots when a part of the analysis changes. Testifying to the advantages of reusable scripts, we reran the the 3-years-old script on, changed the colours and orientation of the plots, and generated a new figure (see Figure 2), all within a handful of minutes.

A scatterplot of the values of the first two formants tracked by Praat, accompanied by the convex hull encompassing all the points. A single Python script performed the formant analysis through Parselmouth, calculated the convex hull with SciPy, and plotted the result with the matplotlib library. To demonstrate the advantages of having scripted the analysis, we swiftly recreated the plots with new data and tweaked the figure’s aesthetics (cfr. Rasilo and Jadoul 2020).

Adaptive playbacks in interactive experiments: ‘Gibbs sampling with people’

As humans divide the world into categories and concepts, it can be surprisingly hard to say which features make perception fall into a certain category. A bioacoustics parallel would be to test which of the many acoustic features encodes aspects of meaning: a signaller’s identity, physical state, potential referential meaning, etc (Seyfarth and Cheney 2003; Townsend et al. 2013). One approach to linking sound and meaning consists in applying correlations and inferential statistics on a large dataset. In an experimental setting however, targeted trials can be used to more precisely probe which features and values are important for a participant to distinguish perceptual categories. As one such approach, Harrison et al. (2020) propose ‘Gibbs Sampling with People’ (GSP). Analogously to ordinary Gibbs sampling, GSP samples from a multidimensional probability distribution by repeatedly sampling the full conditional probability distribution of each single dimension. In GSP, instead of a mathematical formulation of this condition distribution, the participant’s perceptual judgement is used to repeatedly sample a new one-dimensional value. When this procedure is applied correctly, the mathematical derivation underlying Gibbs sampling ensures convergence to the multi-dimensional probability distribution associated with the perceptual category.

The second, online experiment by Harrison et al. (2020) applies GSP to the emotional prosody in human speech, and is a good case study where Parselmouth fits into a larger, interactive experimental setup. Following GSP, participants in the experiment repeatedly altered one of seven selected prosodic measures of the speech fragments to make them sound more ‘happy’, ‘angry’, or ‘sad’. After several iterations of this procedure, each of the three emotional categories became associated with a distinct set of typical values for each of the seven prosodic dimensions.

On a technical level, this study shows flawless integration of Parselmouth into large-scale software platforms. All experiments were run online, through a combination of Amazon Mechanical Turk¹⁰ and a Heroku¹¹ back-end. The actual experiments were implemented in Python, building on top of the PsyNet¹² and Dallinger¹³ frameworks. The full technical setup of these experiments is described in detail in the article’s accompanying supplementary material, which includes the experiment’s full Python code: <https://osf.io/rzk4s/>. As apparent from the supplementary material and Python code, Parselmouth fits right into this technical setup, providing a small but crucial piece of functionality within the larger Python framework for online experiments.

Accessed Praat functionality

The study used Parselmouth to, on-the-fly, change several prosodic features of previously recorded sentences and synthesise a new version of the sentence. More specifically, the participants could adapt the sentences’ pitch level, pitch range, pitch slope, jitter, duration, and intensity (modulation frequency and depth) of the audio fragments (i.e. several acoustic parameters that also often get used in animal bioacoustics). Concretely, in Praat and Parselmouth, this corresponds to the different aspects of a ‘Manipulation’ object.¹⁴

- **Development efficiency:** It would be possible to implement the manipulation as a Praat script and call Praat as a subprocess from Python. This case study, however,

Table 1. A wide variety of research fields and studies have made use of Parselmouth. A majority of these studies use Parselmouth to perform acoustic feature extraction (AFE).

Macroarea & Title of the paper	Keywords (as reported in the original paper)	Parselmouth functionality
Bioacoustics		
Large-scale unsupervised clustering of Orca vocalizations: a model for describing Orca communication systems (Poupard et al. 2019)	none	Acoustic feature extraction (AFE): pitch curve extraction
Scaredy-cat: Assessment of individual differences in response to an acute everyday stressor across development in the domestic cat (Urrutia et al. 2022)	• stress • social separation • individual differences • vocalisation • thermography • <i>Felis silvestris catus</i>	AFE: mean and standard deviation of fundamental frequency
Ontogeny of vocal rhythms in harbor seal pups: an exploratory study (Ravignani et al. 2019)	• bioacoustics • pinnipeds • rhythm • timing • vocal development	AFE: maximum-intensity peaks, inter-peak intervals
Machine learning		
Neural analysis and synthesis: Reconstructing speech from self-supervised representations (Choi et al. 2021)	• speech analysis • speech synthesis • voice conversion • self-supervised • information perturbation	Speech manipulation and resynthesis (formant shifting, pitch randomisation, PSOLA)
What do audio transformers hear? Probing their representations for language delivery & structure (Singla et al. 2022)	• interpretability • probing • pre-trained acoustic representations	AFE: mean pitch, local jitter, local shimmer, voiced to unvoiced ratio
Speech pathology & medicine		
Deep learning-based classification of posttraumatic stress disorder and depression following trauma utilizing visual and auditory markers of arousal and mood (Schultebrucks et al. 2022)	• computer vision • deep learning • depression • digital biomarker • emergency department • landmark feature • posttraumatic stress • resilience • voice analysis	AFE: audio expressivity index, audio intensity, fundamental frequency, harmonics-to-noise ratio, glottal to noise excitation ratio, voice frame core, formant frequency variability, intensity variability, pitch variability, normalised amplitude quotient
Artificial intelligence-based voice assessment of patients with Parkinson's disease off and on treatment: machine vs. deep-learning comparison (Costantini et al. 2023)	• speech • Voice • Parkinson's disease • artificial intelligence • deep learning • CNN • SVM • L-Dopa • F0	AFE: 96 vocal formants-related features
Vocal markers from sustained phonation in Huntington's disease (Riad et al. 2020)	• Huntington's disease • phonation • pathological speech processing • dysarthria • modulation power spectrum	AFE: airflow insufficiency, aperiodicity, irregular vibration of vocal folds, signal perturbation, increased noise, vocal tremor, articulatory deficiency (in combination with pre-existing 'tremor' Praat script; Brückl 2012)
Do you have COVID-19? An artificial intelligence-based screening tool for COVID-19 using acoustic parameters (Vahedian-Azimi et al. 2021)	none	AFE: fundamental frequency, fundamental frequency perturbation, harmonicity, vocal tract function, airflow sufficiency, and periodicity

(Continued)

Table 1. (Continued).

Macroarea & Title of the paper	Keywords (as reported in the original paper)	Parselmouth functionality
Formant-aware spectral analysis of Sustained vowels of pathological breathy voice (Ikuma et al. 2023)	• voice disorder • breathiness • acoustic measurements • linear regression • harmonic modeling • vowel effect	AFE: fundamental frequency, acoustic breathiness index
Emotional analysis & prosody		
Gender, candidate emotional expression, and voter reactions during televised debates (Boussalis et al. 2021)	<i>none</i>	AFE: average per-second fundamental frequency
Modelling individual and cross-cultural variation in the mapping of emotions to speech prosody (van Rijn and Larrouy-Maestri 2023)	<i>none</i>	Pre-processing: sentence segmentation
Does the Lombard effect improve emotional communication in noise? – Analysis of emotional speech acted in noise (Zhao et al. 2019)	• Lombard effect • noise • emotional speech • database recording • auditory detection • acoustic analysis	AFE: fundamental frequency, sound intensity, harmonics-to-noise ratio, first and second formants, spectral tilt
Human-robot interaction & social robotics		
Tell me more! Assessing interactions with social robots from speech (Laban et al. 2020)	• social robots • human-robot interaction • self-disclosure • voice assistant • voice analysis • embodiment	AFE: mean pitch, mean harmonicity, mean intensity, energy, duration
Towards a real-time measure of the perception of anthropomorphism in human-robot interaction (Tsfasman et al. 2021)	• multi-modal • human-robot interaction • prosody • acoustic-prosodic entrainment	AFE: pitch and RMS-intensity
Informal caregivers disclose increasingly more to a social robot over time (Laban et al. 2022)	• caregiving • informal caregivers • human-robot interaction • HRI • social robot • self-disclosure • intervention • ecological momentary intervention • EMI • stress	Speech segmentation: extracting disclosure duration
Music & song		
Large-scale iterated singing experiments reveal oral transmission mechanisms underlying music evolution (Anglada-Tort et al. 2023)	• cultural transmission • oral transmission • cultural evolution • iterated learning • social learning • singing • music evolution • melody • pitch perception • online experiments	AFE: pitch (from song)
Acoustic analysis of the influence of warm-up on singing voice quality (Pórolniczak and Kramarczyk 2023)	• signal analysis • singing voice • singing quality • acoustic analysis • warm-up	AFE: jitter and shimmer

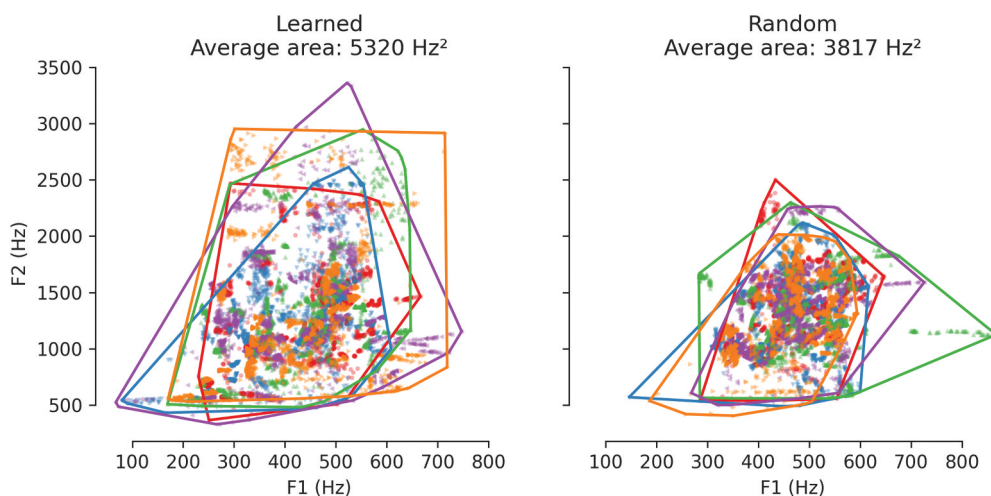


Figure 2. A scatterplot of the values of the first two formants tracked by Praat, accompanied by the convex hull encompassing all the points. A single Python script performed the formant analysis through Parselmouth, calculated the convex hull with SciPy, and plotted the result with the matplotlib library. To demonstrate the advantages of having scripted the analysis, we swiftly recreated the plots with new data and tweaked the figure’s aesthetics (cfr. Rasilo and Jadoul 2020).

showcases two potential advantages of Parselmouth over this approach. First and foremost, it provides a ready-to-use Python library and gets rid of a lot of technical boilerplate code and error-handling the subprocess approach would entail. Secondly, it also eliminates the context-switching when scripting in multiple languages and their associated development environments.

- **Python integration:** The implementation takes advantage of the integration with Python on two different levels. On an architectural level, Parselmouth provides a straightforward way to change relevant prosodic features in resynthesised speech within the Python frameworks for online experiments (i.e. PsyNet and Dallinger). At the implementation level, the published code also shows a tight integration with the NumPy¹⁵ and SciPy¹⁶ scientific computing Python libraries, when modifying and interpolating the audio’s pitch curve.
- **Computational efficiency:** A naive alternative to the current workflow could have been to generate and save in advance all possible combinations of sounds to be played to participants. In the given framework of Gibbs sampling where participants continuously adjust each parameter, it would however be nearly impossible to generate all possible combinations of parameters before the experiment. Even when only very coarsely subdividing the range of each variable into, say, 10 discrete values, this would still result in 10^7 (i.e. 10 million) possible parameter combinations and stimuli to generate. Given the combinatorial and continuous nature of the experimental stimuli, Parselmouth makes it easy to achieve a crucial gain in computational efficiency by generating the stimuli during the experiment.

The case study is a perfect example of how Parselmouth can be combined with many different existing Python software packages and facilitate new research. As we outlined

above, it would have been possible to implement the experiment purely with Praat, but using Parselmouth takes care of all technical details and enables researchers to focus on the actual experiment at hand.

This paper is also a good example of a different experimental pipeline involving Parselmouth. This study did not use Parselmouth to acoustically analyse collected data, but rather to generate stimuli on-the-fly during an interactive experiment. The application of Parselmouth in a large-scale Python experimental framework is a real-life example of the use case anticipated by Jadoul et al. (2018, ‘Integration into experimental design’ subsection), which demonstrated combined usage of Parselmouth in an interactive PsychoPy experiment (Peirce et al. 2019). Past research has used interactive experimental testing in non-human animals (e.g. baboons; Fagot and Paleressompoulle 2009; Fagot and Bonté 2010; Grainger et al. 2012; Claidiere et al. 2014) and shown its feasibility for visual experiments. We suggest that Parselmouth, in combination with PsychoPy and other libraries, could provide flexible solutions to interactive experiments across species involving sound.

Other studies using Parselmouth

Table 1 shows a selection of other published articles which have used the original Parselmouth paper (Jadoul et al. 2018). These studies showcase the broad range of scientific fields using Parselmouth, and the variety of Praat functionality accessed through it.

Table 1 shows the typical contexts in which Parselmouth has proven useful. Firstly, Praat’s fundamental frequency analysis is possibly the single most popular feature. This is perhaps unsurprising, as this algorithm was developed by one of Praat’s authors (Boersma, 1993). Parselmouth provides direct access to the main implementation of said algorithm in Praat, and makes it available from Python.

Another trend in Table 1 is the prevalence of studies where the interpretability of measurements and results is important. The phonetic measurements of Praat typically have a close correspondence to the production process (i.e. related to articulation and voice characteristics); for example, fundamental frequency directly corresponds to the vocal folds’ frequency of vibration, and formants correspond to the upper vocal tract’s resonance frequencies. From this perspective, it is understandable that Parselmouth gets used to calculate phonetic measures in medical and speech pathology studies, in research into explainable AI and in machine learning, or in more exploratory studies.

Discussion

For more than 30 years Praat has been, and will likely continue to be, a unique tool to study human speech. Its applications have however reached beyond phonetics, and several of its analyses are routinely used in bioacoustics. Praat is free and open source, and has been kept up to date to be used on modern computational infrastructure. As argued when Parselmouth was first released, a tighter integration of Praat into the Python scientific ecosystem can benefit both Python and Praat users (Jadoul et al., 2018). Experienced Praat users can more easily combine their expertise and existing Praat scripts with all scientific Python libraries. Meanwhile, Python users from various disciplines can access Praat functionality and automate their acoustic analyses without the need to learn a new scripting language.

Here, we have evaluated the range of scientific applications of Parselmouth, describing three distinct case studies in detail. We suggest there is still some untapped potential of Praat and Parselmouth, especially in the field of bioacoustics. The selection of scientific studies using Parselmouth in [Table 1](#) and the preceding case studies show a particular focus on acoustic feature extraction. Praat's fundamental frequency estimation algorithm is the most popular feature accessed through Parselmouth, but also its analyses of formants and other voice parameters (i.e. glottal pulses, voice breaks, jitter, shimmer, harmonic-to-noise ratio) are often used. So far, a large majority of articles using Parselmouth involve the vocalisations of one animal species, human speech. However, already existing bioacoustics studies using Praat and Parselmouth clearly demonstrate the applicability of these methods to a much broader range of species and needs. In this paper we have focused on mammals; how Parselmouth can be fruitfully used in avian, reptilian, amphibian, piscine, or arthropodean bioacoustics remains an open question, which we hope to see answered by those who study non-mammalian species venturing into Parselmouth as new users.

Given our overview of studies using Parselmouth and bioacoustics studies with Praat, several specific suggestions come to mind where and how Parselmouth could be used in future research. Formant extraction and quantification is often used in bioacoustic work across species. Given the non-trivial, powerful formant extraction algorithm in Praat, we suggest that researchers previously unfamiliar with Praat or Praat scripting could use Parselmouth to access the algorithm in animal studies. The on-the-fly resynthesis and generation of new stimuli, such as demonstrated by Harrison et al. (2020), could also be extended to animal research. The study is a prime example of how Parselmouth can play a role in a larger interactive experimental setup, e.g. live monitoring animal vocalisations or adaptively generating stimuli during a perceptual experiment. The application of Parselmouth to synthesised speech (Rasilo and Jadoul 2020) raises the opportunity to perform computational modelling and synthesis techniques on non-human animals' vocalisations, and illustrates the role that Parselmouth can play in analysing such data. Here and in other scientific studies, adopting Praat and Parselmouth provides a tool for direct cross-species comparisons by processing data with the exact same pipeline.

The collection of bioacoustics-related Python packages keeps expanding, resulting in a whole ecosystem of packages and tools that can interact and work together (Rhinehart et al. 2022). A few recent examples show the variety of available software. The crowsetta library (Nicholson 2023) provides a single Python interface to read and process different bioacoustics annotation formats. OpenSoundscape (Lapp et al. 2023), Koogu (Shyam 2022), and vak (Nicholson and Cohen 2022) are Python libraries for training custom machine learning models on animal vocalisation. The scikit-maad (Ulloa et al. 2021) package consists of functionality necessary to process ecoacoustics data. On top of these bioacoustics-specific packages, Python features a vast array of general scientific computing libraries. It is impossible to list all potentially useful Python packages, but the case studies above have shown some of the most pervasive ones: NumPy, SciPy, and Pandas. Libraries also exist to link Python and R (notably, rpy2¹⁷ and reticulate¹⁸), another programming language with plenty of libraries dedicated to bioacoustics, such as warbleR (Araya-Salas et al. 2017) or Seewave (Sueur et al. 2008). In this landscape, Parselmouth aims to fill one missing piece of the puzzle, making Praat more easily accessible from Python and combinable with all these other libraries.

The usual caution when working with bioacoustics data is still necessary when using Parselmouth. The Python interface aims to simplify the integration of acoustic analyses into a larger workflow, but Parselmouth cannot magically determine which parameter values or Fourier window length are appropriate for a particular species or case at hand. On the contrary, it is important to realise that Praat was designed to deal with human speech, and that its default parameters have been fine-tuned for this one species. Given the similarity of production mechanisms, Praat algorithms and parameter settings will be most applicable in other mammalian species (potentially even non-laryngeal ones; e.g. Madsen et al. 2023; Ravignani and Herbst 2023). This means that, as when applying any methodology, when using Praat and Parselmouth researchers should always keep an eye on the methods' suitability and find appropriate parameter settings with regard to the species of interest.

When exploring methods and parameter values, Parselmouth and Praat perfectly complement each other: example recordings and different parameter settings can be explored in Praat's graphical user interface (GUI), the noise conditions can be inspected, and the spectrogram and results should be scrutinised before automating the resulting workflow (either as a traditional Praat script or in Python with Parselmouth). The direct correspondence between Parselmouth and Praat versions ensures reproducibility of results between software packages.¹⁹ We believe this to be an accessible way of applying the classical good practices in bioacoustics research. Additionally, Parselmouth can also be part of a more automated parameter exploration: given a sample of manually or semi-automatically analysed training data, a script can loop over multiple parameter combinations in order to find the optimal parameter values.

To conclude, we hope the examples presented in this article have shown the benefits that Parselmouth can bring to a data-focussed bioacoustics project. Several studies over the past years have already integrated Parselmouth as a component of their experiment or analysis. And while no two projects or researchers have the exact same needs, the variety of studies shows Parselmouth's potential to facilitate existing bioacoustics research pipelines and enable new, more complex experimental setups. Finally, by adding to the collection of available Python bioacoustic software packages, Parselmouth can hopefully play a role in the shared effort to make bioacoustics analyses more efficient, reusable, and reproducible.

Notes

1. Available from <https://github.com/YannickJadoul/Parselmouth>, documentation at <https://parselmouth.readthedocs.io/>
2. <https://www.python.org/doc/essays/omg-darpa-mcc-position/>
3. <https://numpy.org/doc/stable/user/c-info.python-as-glue.html>
4. The corresponding Praat version can be accessed as `parselmouth.PRAAT_VERSION`.
5. https://www.fon.hum.uva.nl/praat/manual/Sound__To_Pitch__ac____.html
6. <https://numpy.org/>
7. <https://pandas.pydata.org/>
8. <https://joblib.readthedocs.io/>
9. <https://scipy.org/>
10. <https://www.mturk.com/>
11. <https://www.heroku.com/>

12. <https://psynetdev.gitlab.io/PsyNet/>
13. <https://dallinger.readthedocs.io/>
14. <https://www.fon.hum.uva.nl/praat/manual/Manipulation.html>
15. <https://numpy.org/>
16. <https://scipy.org/>
17. <https://rpy2.github.io/>
18. <https://rstudio.github.io/reticulate/>
19. In scientific publications, we advise to report the exact version of both Praat and Parselmouth used to obtain the results.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The Comparative Bioacoustics Group is funded by Max Planck Group Leader funding to A.R. The Center for Music in the Brain is funded by the Danish National Research Foundation [DNRF117].

ORCID

Yannick Jadoul  <http://orcid.org/0000-0003-0540-3135>

Bart de Boer  <http://orcid.org/0000-0001-9429-6129>

Andrea Ravignani  <http://orcid.org/0000-0002-1058-0024>

References

- Abur D, Lester-Smith RA, Daliri A, Lupiani AA, Guenther FH, Stepp CE, Larson CR. 2018. Sensorimotor adaptation of voice fundamental frequency in Parkinson's disease. *PloS One*. 13(1): e0191839. doi: [10.1371/journal.pone.0191839](https://doi.org/10.1371/journal.pone.0191839).
- Anglada-Tort M, Harrison PM, Lee H, Jacoby N. 2023. Large-scale iterated singing experiments reveal oral transmission mechanisms underlying music evolution. *Curr Biol*. 33(8):1472–1486. e12. doi: [10.1016/j.cub.2023.02.070](https://doi.org/10.1016/j.cub.2023.02.070).
- Anikin A, Pisanski K, Reby D. 2022. Static and dynamic formant scaling conveys body size and aggression. *R Soc Open Sci*. 9(1):211496. doi: [10.1098/rsos.211496](https://doi.org/10.1098/rsos.211496).
- Araya-Salas M, Smith-Vidaurre G, Golding N. 2017. warbleR: an R package to streamline analysis of animal acoustic signals. *Methods Ecol Evol*. 8(2):184–191. doi: [10.1111/2041-210X.12624](https://doi.org/10.1111/2041-210X.12624).
- Beguś G, Leban A, Gero S. (2023). Approaching an unknown communication system by latent space exploration and causal inference. *arXiv preprint arXiv:2303.10931*.
- Boë L-J, Berthommier F, Legou T, Captier G, Kemp C, Sawallis TR, Becker Y, Rey A, Fagot J, Reby D. 2017. Evidence of a vocalic proto-System in the baboon (*Papio papio*) suggests pre-hominin speech precursors. *PloS One*. 12(1):e0169321. doi: [10.1371/journal.pone.0169321](https://doi.org/10.1371/journal.pone.0169321).
- Boersma P. 1993. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the institute of phonetic sciences* (Vol. 17, No. 1193, p. 97–110).
- Boersma P. 2001. Praat, a system for doing phonetics by computer. *Glott International*. 5(9):341–345.
- Boersma P, Weenink D. 2023. Praat: doing phonetics by computer [Computer program]. Version 10.3.17. [accessed 2023 Sept 10]. <http://www.praat.org/>.
- Bogdanchikov A, Zhaparov M, Suliyev R. 2013. Python to learn programming. *J Phys*. 423(1):12027. 1.

- Boussalis C, Coan TG, Holman MR, Müller S. 2021. Gender, candidate emotional expression, and voter reactions during televised debates. *Am Polit Sci Rev.* 115(4):1242–1257. doi: [10.1017/S0003055421000666](https://doi.org/10.1017/S0003055421000666).
- Brückl M. 2012. Vocal tremor measurement based on autocorrelation of contours. *INTERSPEECH 2012 ISCA's 13th Annual Conference Portland*; Sep 9–13; Portland, OR, USA; p. 715–718. doi: [10.21437/Interspeech.2012-223](https://doi.org/10.21437/Interspeech.2012-223).
- Childers DG. 1978. *Modern spectrum analysis*. IEEE Press.
- Choi HS, Lee J, Kim W, Lee J, Heo H, Lee K. 2021. Neural analysis and synthesis: reconstructing speech from self-supervised representations. *Adv Neural Inf Process Syst.* 34:16251–16265.
- Claidiere N, Smith K, Kirby S, Fagot J. 2014. Cultural evolution of systematically structured behaviour in a non-human primate. *Proc R Soc B.* 281(1797): 20141541.
- Costantini G, Cesarini V, Di Leo P, Amato F, Suppa A, Asci F, Saggio G. 2023. Artificial intelligence-based voice assessment of patients with Parkinson's disease off and on treatment: machine vs deep-learning comparison. *Sensors.* 23(4):2293. doi: [10.3390/s23042293](https://doi.org/10.3390/s23042293).
- Elemans CP, Rasmussen JH, Herbst CT, Düring DN, Zollinger SA, Brumm H, Švec JG. 2015. Universal mechanisms of sound production and control in birds and mammals. *Nat Commun.* 6(1):8978. doi: [10.1038/ncomms9978](https://doi.org/10.1038/ncomms9978).
- Fagot J, Bonté E. 2010. Automated testing of cognitive performance in monkeys: use of a battery of computerized test systems by a troop of semi-free-ranging baboons (*Papio papio*). *Behav Res Methods.* 42(2):507–516. doi: [10.3758/BRM.42.2.507](https://doi.org/10.3758/BRM.42.2.507).
- Fagot J, Paleressompoulle D. 2009. Automatic testing of cognitive performance in baboons maintained in social groups. *Behav Res Methods.* 41(2):396–404. doi: [10.3758/BRM.41.2.396](https://doi.org/10.3758/BRM.41.2.396).
- Fitch WT, de Boer B, Mathur N, Ghazanfar AA. 2016a. Monkey vocal tracts are speech-ready. *Sci Adv.* 2(12):e1600723. doi: [10.1126/sciadv.1600723](https://doi.org/10.1126/sciadv.1600723).
- Grainger J, Dufau S, Montant M, Ziegler JC, Fagot J. 2012. Orthographic processing in baboons (*Papio papio*). *Sci.* 336(6078):245–248. doi: [10.1126/science.1218152](https://doi.org/10.1126/science.1218152).
- Harrison P, Marjeh R, Adolff F, van Rijn P, Anglada-Tort M, Tchernichovski O, Larrouy-Maestri P, Jacoby N. 2020. Gibbs sampling with people. *Adv Neural Inf Process Syst.* 33:10659–10671.
- Ikuma T, McWhorter AJ, Oral E, and Kunduk M. 2023. Formant-aware spectral analysis of Sustained vowels of pathological breathy voice. *J Voice.* doi:[10.1016/j.jvoice.2023.05.002](https://doi.org/10.1016/j.jvoice.2023.05.002).
- Jadoul Y, Thompson B, De Boer B. 2018. Introducing Parselmouth: a python interface to Praat. *J Phon.* 71:1–15.
- Keller PE, König R, Novembre G. 2017. Simultaneous cooperation and competition in the evolution of musical behavior: sex-related modulations of the singer's formant in human chorusing. *Front Psychol.* 8:1559. doi:[10.3389/fpsyg.2017.01559](https://doi.org/10.3389/fpsyg.2017.01559).
- Laban G, George JN, Morrison V, Cross ES. 2020. Tell me more! Assessing interactions with social robots from speech. *Paladyn.* 12(1):136–159. doi: [10.1515/pjbr-2021-0011](https://doi.org/10.1515/pjbr-2021-0011).
- Laban G, Morrison V, Kappas A, Cross ES. 2022. Informal caregivers disclose increasingly more to a social robot over time. *CHI '22: CHI Conference on Human Factors in Computing Systems*; 29 April 2022–5 May 2022; New Orleans, LA, USA; p. 1–7. doi: [10.1145/3491101](https://doi.org/10.1145/3491101).
- Lapp S, Rhinehart T, Freeland-Haynes L, Khilnani J, Syunkova A, Kitzes J. 2023. OpenSoundscape: an open-source bioacoustics analysis package for Python. *Methods Ecol Evol.* 14(9):2321–2328. doi: [10.1111/2041-210X.14196](https://doi.org/10.1111/2041-210X.14196).
- Leonetti S, Jadoul Y, Torres Borda L, de Reus K, Rasilo H, Salazar Casals A, Ravignani A. 2022. Noise-dependent vocal plasticity in harbour seal and grey seal pups. *The European Conference on Behavioural Biology (ECBB 2022)*; July 20– 23; Groningen, the Netherlands.
- Madsen PT, Siebert U, Elemans CP. 2023. Toothed whales use distinct vocal registers for echolocation and communication. *Science.* 379(6635):928–933. doi: [10.1126/science.adc9570](https://doi.org/10.1126/science.adc9570).
- McComb K, Shannon G, Sayialel KN, Moss C. 2014. Elephants can determine ethnicity, gender, and age from acoustic cues in human voices. *Proc Natl Acad Sci.* 111(14):5433–5438.
- Mészárosóvá E. 2015. Is python an appropriate programming language for teaching programming in secondary schools. *Int J Inf Commun Technol Educ.* 4(2):5–14. doi: [10.1515/ijicte-2015-0005](https://doi.org/10.1515/ijicte-2015-0005).
- Nicholson D. 2023. Crowsetta: a Python tool to work with any format for annotating animal vocalizations and bioacoustics data. *J Open Source Softw.* 8(84):5338. doi: [10.21105/joss.05338](https://doi.org/10.21105/joss.05338).

- Nicholson D, Cohen Y. 2022. vak (0.6.0). Zenodo. doi: [10.5281/zenodo.6808839](https://doi.org/10.5281/zenodo.6808839).
- Nunes CEP, Nevard L, Montealegre-Z F, Vallejo-Marín M. 2021. Variation in the natural frequency of stamens in six morphologically diverse, buzz-pollinated, heterantherous *Solanum* taxa and its relationship to bee vibrations. *Bot J Linn Soc.* 197(4):541–553. doi: [10.1093/botlinnean/boab044](https://doi.org/10.1093/botlinnean/boab044).
- Peirce J, Gray JR, Simpson S, MacAskill M, Höchenberger R, Sogo H, Lindeløv JK, Lindeløv JK. 2019. PsychoPy2: experiments in behavior made easy. *Behav Res Methods.* 51(1):195–203. doi: [10.3758/s13428-018-01193-y](https://doi.org/10.3758/s13428-018-01193-y).
- Pisanski K, Jones BC, Fink B, O'Connor JJ, DeBruine LM, Röder S, Feinberg DR. 2016a. Voice parameters predict sex-specific body morphology in men and women. *Anim Behav.* 112:13–22. doi: [10.1016/j.anbehav.2015.11.008](https://doi.org/10.1016/j.anbehav.2015.11.008).
- Pisanski K, Mora EC, Pisanski A, Reby D, Sorokowski P, Frackowiak T, Feinberg DR. 2016b. Volitional exaggeration of body size through fundamental and formant frequency modulation in humans. *Sci Rep.* 6(1):34389. doi: [10.1038/srep34389](https://doi.org/10.1038/srep34389).
- Pórolniczak E, Kramarczyk M. 2023. Acoustic analysis of the influence of warm-up on singing voice quality. *J Voice.* doi: [10.1016/j.jvoice.2023.02.017](https://doi.org/10.1016/j.jvoice.2023.02.017).
- Poupard M, Best P, Schlüter J, Symonds H, Spong P, Lengagne T, Soriano T, Glotin H. 2019. Large-scale unsupervised clustering of Orca vocalizations: a model for describing Orca communication systems. 2nd International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots; Aug 29–30; London, UK.
- Rasilo H, Jadoul Y. 2020. Discovering articulatory speech targets from synthesized random babble. In *Proc. Interspeech OCT 25–29, 2020*, 3715–3719. Shanghai, China; p. 3715–3719. doi: [10.21437/Interspeech.2020-3186](https://doi.org/10.21437/Interspeech.2020-3186).
- Rasilo H, Räsänen O, Laine UK. 2013. Feedback and imitation by a caregiver guides a virtual infant to learn native phonemes and the skill of speech inversion. *Speech Commun.* 55(9):909–931.
- Ravignani A, Herbst CT. 2023. Voices in the ocean. *Sci.* 379(6635):881–882. doi: [10.1126/science.adg5256](https://doi.org/10.1126/science.adg5256).
- Ravignani A, Kello CT, de Reus K, Kotz SA, Dalla Bella S, Méndez-Aróstegui M, Rapado-Tamarit B, Rubio-Garcia A, de Boer B. 2019. Ontogeny of vocal rhythms in harbor seal pups: an exploratory study. *Curr Zool.* 65(1):107–120. doi: [10.1093/cz/zoey055](https://doi.org/10.1093/cz/zoey055).
- Rhinehart T, Lapp S, Kitzes J. 2022. Identifying and building on the current state of bioacoustics software. *J Acoust Soc Am.* 151(4):A27–A27. doi: [10.1121/10.0010544](https://doi.org/10.1121/10.0010544).
- Riad R, Titeux H, Lemoine L, Montillot J, Bagnou JH, Cao XN, Bachoud-Lévi AC. 2020. Vocal markers from sustained phonation in Huntington's disease. *Proc. Interspeech 2020*; p. 1893–1897. <https://dblp.org/rec/conf/interspeech/RiadTLMBCDB20.html>.
- Riede T, Bronson E, Hatzikirou H, Zuberbühler K. 2005. Vocal production mechanisms in a non-human primate: morphological data and a model. *J Hum Evol.* 48(1):85–96. doi: [10.1016/j.jhevol.2004.10.002](https://doi.org/10.1016/j.jhevol.2004.10.002).
- Rocha LH, Ferreira LS, Paula BC, Rodrigues FH, Sousa-Lima RS. 2015. An evaluation of manual and automated methods for detecting sounds of maned wolves (*Chrysocyon brachyurus* illiger 1815). *Bioacoustics.* 24(2):185–198. doi: [10.1080/09524622.2015.1019361](https://doi.org/10.1080/09524622.2015.1019361).
- Rose SJ, Allen D, Noble D, Clarke JA. 2018. Quantitative analysis of vocalizations of captive Sumatran tigers (*Panthera tigris sumatrae*). *Bioacoustics.* 27(1):13–26. doi: [10.1080/09524622.2016.1272003](https://doi.org/10.1080/09524622.2016.1272003).
- Schultebras K, Yadav V, Shalev AY, Bonanno GA, Galatzer-Levy IR. 2022. Deep learning-based classification of posttraumatic stress disorder and depression following trauma utilizing visual and auditory markers of arousal and mood. *Psychol Med.* 52(5):957–967. doi: [10.1017/S0033291720002718](https://doi.org/10.1017/S0033291720002718).
- Seyfarth RM, Cheney DL. 2003. Meaning and emotion in animal vocalizations. *Ann N Y Acad Sci.* 1000(1):32–55. doi: [10.1196/annals.1280.004](https://doi.org/10.1196/annals.1280.004).
- Shyam M. (2022). shyamblast/Koogu: v0.7.1 (v0.7.1). Zenodo. doi: [10.5281/zenodo.7275319](https://doi.org/10.5281/zenodo.7275319).
- Singla YK, Shah J, Chen C, Shah RR. 2022. What do audio transformers hear? Probing their representations for language delivery & structure. 2022 IEEE International Conference on Data

- Mining Workshops (ICDMW); November; 28 Nov 2022–01 Dec 2022; Orlando, FL, USA; p. 910–925. doi: [10.1109/ICDMW58026.2022.00120](https://doi.org/10.1109/ICDMW58026.2022.00120).
- Sueur J, Aubin T, Simonis C. 2008. Seewave, a free modular tool for sound analysis and synthesis. *Bioacoustics*. 18(2):213–226. doi: [10.1080/09524622.2008.9753600](https://doi.org/10.1080/09524622.2008.9753600).
- Taylor AM, Charlton BD, Reby D. 2016. Vocal production by terrestrial mammals: source, filter, and function. In: Suthers R, Fitch W, Fay R, and Popper A, editors. *Vertebrate sound production and acoustic communication*. Springer handbook of auditory research. Vol. 53. Cham: Springer. doi: [10.1007/978-3-319-27721-9_8](https://doi.org/10.1007/978-3-319-27721-9_8).
- Taylor AM, Reby D. 2010. The contribution of source–filter theory to mammal vocal communication research. *J Zool*. 280(3):221–236. doi: [10.1111/j.1469-7998.2009.00661.x](https://doi.org/10.1111/j.1469-7998.2009.00661.x).
- Torres Borda L, Jadoul Y, Rasilo H, Salazar Casals A, Ravignani A. 2021. Vocal plasticity in harbour seal pups. *Philos Trans R Soc B*. 376(1840):20200456.
- Townsend SW, Manser MB, Hauber M. 2013. Functionally referential communication in mammals: the past, present and the future. *Ethology*. 119(1):1–11. doi: [10.1111/eth.12015](https://doi.org/10.1111/eth.12015).
- Tsfasman M, Saravanan A, Viner D, Goslinga D, De Wolf S, Raman C, Oertel C. 2021. Towards a real-time measure of the perception of anthropomorphism in human-robot interaction. *Proceedings of the 2nd ACM Multimedia Workshop on Multimodal Conversational AI*; p. 13–18.
- Ulloa JS, Haupert S, Latorre JF, Aubin T, Sueur J. 2021. Scikit-maad: an open-source and modular toolbox for quantitative soundscape analysis in Python. *Methods Ecology Evol*. 12(12):2041–210X.13711. doi: [10.1111/2041-210X.13711](https://doi.org/10.1111/2041-210X.13711).
- Urrutia A, Bánszegi O, Szenczi P, Hudson R. 2022. Scaredy-cat: assessment of individual differences in response to an acute everyday stressor across development in the domestic cat. *Appl Anim Behav Sci*. 256:105771. doi:[10.1016/j.applanim.2022.105771](https://doi.org/10.1016/j.applanim.2022.105771).
- Vahedian-Azimi A, Keramatfar A, Asiaee M, Atashi SS, Nourbakhsh M. 2021. Do you have COVID-19? An artificial intelligence-based screening tool for COVID-19 using acoustic parameters. *J Acoust Soc Am*. 150(3):1945–1953. doi: [10.1121/10.0006104](https://doi.org/10.1121/10.0006104).
- van Rijn P, Larrouy-Maestri P. 2023. Modelling individual and cross-cultural variation in the mapping of emotions to speech prosody. *Nat Hum Behav*. 7(3):1–11. doi: [10.1038/s41562-022-01505-5](https://doi.org/10.1038/s41562-022-01505-5).
- Wirth C, Warren JD. 2020. Spatial and temporal variation in toadfish (*Opsanus tau*) and cusk eel (*Ophidion marginatum*) mating choruses in eelgrass (*Zostera marina*) beds in a shallow, temperate estuary. *Bioacoustics*. 29(1):61–78. doi: [10.1080/09524622.2018.1542631](https://doi.org/10.1080/09524622.2018.1542631).
- Zhao Y, Ando A, Takaki S, Yamagishi J, Kobashikawa S. (2019). Does the Lombard Effect Improve Emotional Communication in Noise? — Analysis of Emotional Speech Acted in Noise. *Proc. Interspeech* September 15–19, 2019, 3292–3296, doi: [10.21437/Interspeech.2019-1605](https://doi.org/10.21437/Interspeech.2019-1605).