s(ound)lab: An easy to learn Python package for designing and running psychoacoustic experiments.

Marc Schönwiesner[*1, 2] **and Ole Bialas**[†1]

1 Institute of Biology, Faculty of Life sciences, Leipzig University, Germany **2** Institute of Psychology, Faculty of Arts and Sciences, University of Montreal, Canada

DOI: 10.21105/joss.03284

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @hadware
- @sneakers-the-rat

Submitted: 19 April 2021
Published: 25 June 2021

Summary

Slab enables researchers and students to prototype and implement psychoacoustic experiments quickly. Slab implements many of the procedures for psychoacoustic research and experiment control and is easily combined with other Python software. A secondary aim of slab is to enable researchers and students without prior training in computer programming and digital signal processing to implement and conduct these experiments. Slab provides building blocks rather than ready-made solutions, so that experimenters still need to carefully consider stimulation, sequencing and data management. This also makes slab very flexible and easy to customise. In the documentation (see slab.readthedocs.io), we provide tutorials suitable for beginners. We also provide experiments conducted in our lab as worked examples.

Slab can:

- generate and manipulate single- and multi-channel sounds
- analyse sound by extracting basic sound features
- aid experimental design through stimulus sequence management and response simulation
- calibrate the experimental setup (loudness calibration and frequency equalisation)
- display and manipulate head-related transfer functions

Below is an example script that estimates the detection threshold for a small change in the location of a sound source (minimum audible angle) with an amplitude-modulated pink noise and a staircase procedure. It illustrates the use of the `Binaural` and `Staircase` classes. The method `present_afc_trial` is a higher-level convenience function to present several sounds and acquire a response from the participant, but each of these steps can be performed separately in one or two lines when implementing non-standard paradigms.

```
# 1up-2down staircase, starting at 20° separation:
stairs = slab.Staircase(start_val=20, min_val=0, n_reversals=18)
for angle in stairs:
  # generate fresh noise in each trial:
  midline = slab.Binaural.pinknoise(duration=0.25)
  midline =  midline.am()  # apply amplitude modulation
  midline = midline.ramp()
  # get ITD equivalent to current angle:
  itd = slab.Binaural.azimuth_to_itd(angle, head_radius=10)
  stimulus_left = midline.itd(itd)  # apply the itd
  # apply smoothed head-related transfer function to
```

[*] co-first author, corresponding author
[†] co-first author

```
  # evoke externalized percept:
  midline = midline.externalize()
  stimulus_left = stimulus_left.externalize()
  # 3 alternatives (target and 2 distractors):
  stairs.present_afc_trial(target=stimulus_left, distractors=[midline]*2)
print(stairs.threshold(n=14))
```

Statement of need

Slab was written to address our own need for a Python package that allows incoming students and new researchers to implement their own experiments with clean and maintainable code. Experimenters should be able to write and understand the code that they are using. Many students in our lab have very quickly progressed from programming novices to completing psychoacoustic research theses using slab, and we think the package may be useful to others in the same situation. Our approach differs from existing software packages for running behavioural experiments, which provide a high level graphical user interface to customise the parameters of pre-made experiments (Carcagno, 2012; Peirce et al., 2019). In our experience, this leads to very little generalisable learning of Python and experimental control. Slab facilitates this learning by providing basic building blocks, implemented concisely in pure Python, that can be used to construct experiments of various levels of complexity. There is some overlap with librosa (McFee et al., 2015), a Python package for music analysis, but that package focuses on feature extraction and does not support psychoacoustic experimentation. Slab is also one of very few packages that features manipulation of head-related transfer functions and a simple API for reading a standard file format (SOFA) for such data. There is overlap with more recent implementations of the complete SOFA API (Lossner, 2019; Perez-Lopez, 2019), but these packages provide no methods for typical experimental manipulations of head-related transfer functions. We will likely use `pysofaconventions` internally for handling SOFA files within `slab` in the near future. The architecture of the `Signal` class and some of the sound generation methods in the `Sound` class are inspired on the 1.4 version of Brian.hears (Stimberg & Goodman, 2019), but we made several simplifications based on learning reports from students. For instance, signal objects do not implement buffering and do not inherit from Numpy arrays (Harris et al., 2020) directly, because these features significantly hindered students' understanding of the code.

Audience

Slab is directed towards students and researchers of all levels studying the perception of sound. Researchers and incoming students at our lab use it routinely in behavioural and neuroimaging experiments, and the package and has been used in several graduate courses psychophysics and auditory neuroscience.

References

Carcagno, S. (2012). Pychoacoustics. In *GitHub repository*. https://github.com/sam81/pychoacoustics; GitHub.

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Rio, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Lossner, J. (2019). Spatially oriented format for acoustics (SOFA) API for python. In *GitHub repository*. https://github.com/spatialaudio/python-sofa; GitHub.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*, *8*. https://doi.org/10.25080/majora-7b98e3ed-003

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

Perez-Lopez, A. (2019). Python implementation of the SOFA specification. In *GitHub repository*. https://github.com/andresperezlopez/pysofaconventions; GitHub.

Stimberg, M., & Goodman, D. (2019). Brian hears. In *GitHub repository*. https://github.com/brian-team/brian2hears; GitHub.