BciPy: A Python Framework for Brain- Computer Interface Research

Tab Memmott[1], Aziz Kocanaogullari[2], Deniz Erdogmus[2], Steven Bedrick[1], Betts Peters[1], Melanie Fried- Oken[1], and Barry Oken[1]

1. Oregon Health and Science University 2. Northeastern University

Introduction: Interest in BCI has increased in recent years with productive collaboration across fields such as engineering, neuroscience, allied health, and computer science. Laboratories interested in exploring BCI must either construct a working system from scratch or use an off-the-shelf option. Technology that is able to solve the architectural and ease of access barriers to the BCI field will be essential in maintaining momentum and opening the field to even more diverse expertise. Here we mention popular software suites for BCI research and propose a new system, BciPy, to address existing gaps.

Material, Methods and Results:

There are BCI frameworks currently available, most notably the widely used BCI2000 and OpenViBE. These systems save time to set up or modify BCI experiments. They are free for research, generally written in lower level programming languages (C++) and have been used by many laboratories. However, many in the BCI field do not have training in lower level programming languages, if any at all. Code should be written in a language that is easily understood and widely used for programming tasks across disciplines. Python should be considered, as it's becoming the dominant language in many scientific fields and is increasing in usage yearly [1].

In this abstract, we present a modular, Python-based BCI framework, BciPy (See Figure 1 for a closed-loop view of framework).

This software:

- utilizes a higher-level programming language without comprising timing;
- outputs session data into immediately usable formats (.txt, .json, .csv, .pdf), with helper functions and full documentation;
- allows for closed-loop BCI control, with usage of modules outside of the loop; 4) is free;
- works on major OS;
- is architecturally modular;
- contains test and demo scripts; and
- has few outside dependencies, excepting those that are widely used, such as SciPy [2] and Psychopy [3].
- easily uploadable into major EEG processing software (ex. EEGLAB).
- prepares for the tools of the future (lack of parallel ports, eye-tracking, …)

The initial version is pre-packaged with an RSVPKeyboard paradigm [4] for BCI communication, data acquisition (for use with Wearable Sensing and Lab Streaming Layer supported devices), EEG signal and language modeling, display, and GUI with parameter editing.

Figure 1: Closed-Loop Module Flowchart

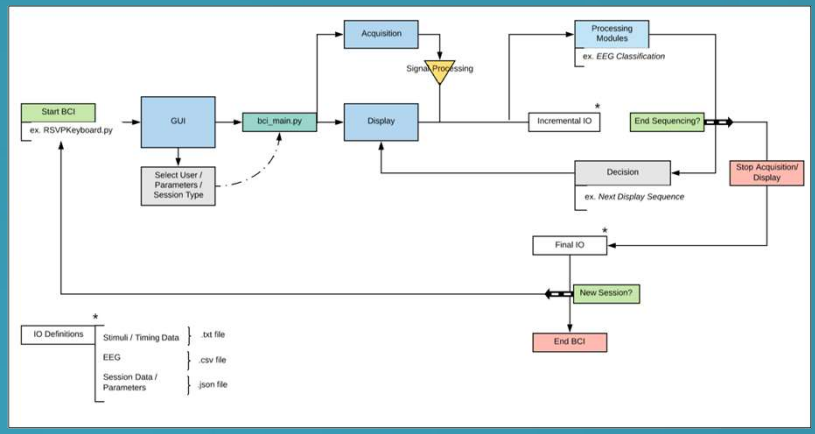


Figure 2: Modules and Implementation

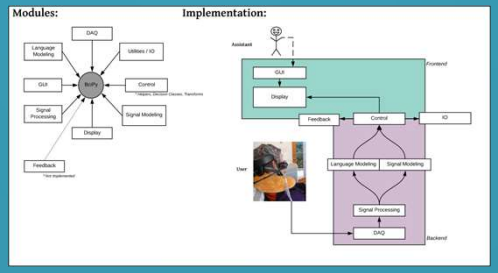
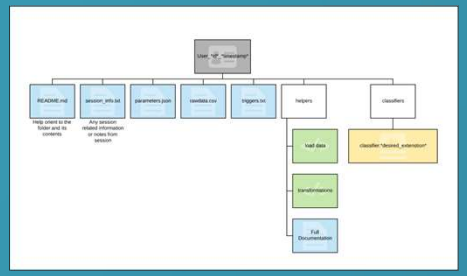


Figure 3: Data Save Structure



Discussion: There are still many features and enhancements to be added to the initial BciPy release to facilitate future BCI research. The code will go through 6-month development cycles. Contributions from the public will be encouraged and authorship granted to code integrated. Future developments are set to include additional user inputs (eye gaze, switches), enhancements to data acquisition, new language models and signal classifiers, and user interface enhancements.

Significance: A Python-based BCI framework will significantly reduce the barriers to contribute to the field and encourage participation from across disciplines. The code is approved for open source. V1.0 will be available to all in late Spring 2018. It is also freely available, with any published features, via request to the authors.

Acknowledgements: Dani Smektala for GUI work; Berkan Kadioglu, Andac Demir, Paula Gonzalez-Navarro for signal processing and classification work; Shiran Dudy and Shaobin Xu for language model integration; Matthew Lawhead for DAQ work; Brandon Eddy and David Smith for design contributions. Support provided by NIH (2R01DC009834) and NIDILRR (90RE5017).

Citations: [1] Robinson, David. (September 14, 2017). Why Is Python Growing So Quickly? https://stackoverflow.blog/2017/09/14/python-growing-quickly/ [2] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/. [3] doi:10.1016/j.jneumeth.2006.11.017. [4] doi:10.1109/ICASSP.2012.6287966.