DSCI 591: Capstone Project – Proposal Report for Sensing in Biomechanical Processes Lab (SimPL)

Team members: Matthew Pin, Mo Garoub, Sasha Babicki, Zhanyi (Yiki) Su

Project mentor: Joel Ostblom

Date: May 14, 2021

3.1. Executive Summary

We are developing a Python package and dashboard for SimPL, a research lab that explores research questions concerning the human brain, to help them visualize EEG data and understand the functional state of the brain after sports-related head injuries. After learning about SimPL's problem with limited visualization methods, we are proposing the following deliverables:

1) A Python package for generating advanced EEG visualizations and metrics
2) An interactive web app to provide a user interface for the package

If time permits, we will expand our deliverables to include a stretch goal of building a data pipeline for detecting underlying structure in EEG data using unsupervised learning methods such as clustering.

3.2 Introduction

3.2.1 Background

Electroencephalograms (EEG) is an electrophysiological measurement method used to examine the electrical activity of the brain and represent it as location-based channels of waves and frequencies. EEG benefits from being inexpensive and unobtrusive, leading to its widespread use in diagnosing brain disorders such as epilepsy and brain damage from head injuries. EEG data is recorded with high dimensionality, so the use of visualizations and metrics is essential for the data to be easily interpreted by humans. Currently, the options for visualizing EEG data require the use of complicated packages or software and the functionally is often limited.

SimPL is a research lab in the department of Mechanical Engineering at UBC which focuses on developing quantitative and sensitive methods to evaluate the electrophysiological changes after sport head injuries. The underlying mechanisms of brain dysfunction are not fully understood, in part because concussion and brain injuries are generally invisible. EEG technology has proven particularly useful for their research purposes.

Our team was approached to design novel solutions and methods to simplify the process of extracting and visualizing the human brain state using EEG data. We intend to build tools which can extend the number of visualizations available to researchers with minimal programming background. Making multiple visualizations convenient to access and view simultaneously will allow for an intuitive understanding of the broad picture of brain function. Additionally, our machine learning stretch goal could uncover patterns in the data which could not be determined based on visualization alone.

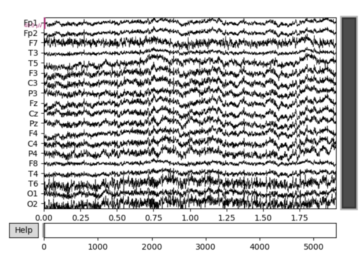

Figure 1: Example of a Standard Visualization for Two Seconds of 19 Channel EEG Raw Voltage Values

3.2.1 Main Goals

We will be designing a customized, well-documented Python package which will provide the ability to conveniently produce advanced visualizations and metrics for specified time ranges of EEG data. At minimum, the package will include the following functionality which was requested by the partner in order to help understand the state of brain:

1) Raw voltage values - produce raw voltage values to measure the EEG amplifier
2) Connectivity - calculate the correlation between nodes or groups of nodes for specified time ranges
3) 2D head map video - generate an animated 2D topographic heatmap of the voltage values recorded by each node of the EEG device. Includes the ability to take snapshots of power changes that represent the magnitude of the signal as a function of frequency
4) 3D skull map video - generate an animated topographic heatmap of voltage values mapped to a 3D model of skull
5) Interpolated 3D brain map - generate an animated topographic heatmap of voltage values mapped to a 3D model of the brain by interpolating voltage values to their presumed location in the brain.

We will also build an interactive web application to serve as a user interface (UI) for the package. The web application will be accessible by running a simple command and will provide all of the visualizations in one place. At a minimum, the widgets for customizing settings will provide the following options:

1) File selection
    - Depending on the preferences of the partner, this can be linked directly to their local files or files on cloud
    - If requested by the partner, a file upload option is also feasible
2) Two time selection options
    - The user may input epoch timing data and then select epochs to display
    - The user may input a specific start time and duration of the animation
3) Frame rate
    - The frame rate for the animations

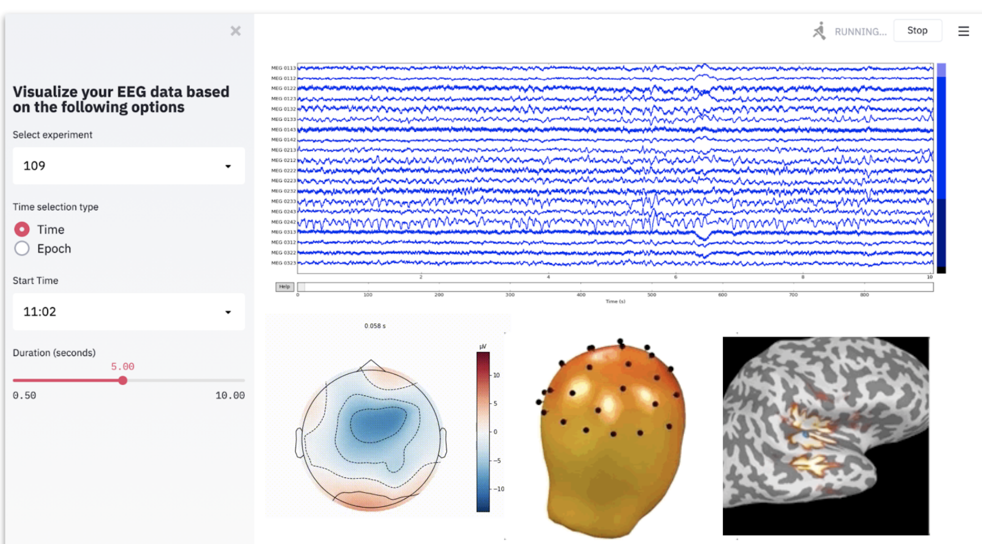

Figure 2: UI Mockup with 1. Raw voltage values (top), 2. 2D head map (left), 4. 3D skull map video (middle), 5. Interpolated 3D brain map (right)

3.2.2 Stretch goal

If sufficient time is available after completing the main deliverable we will complete an additional deliverable. Our stretch goal is to create a data pipeline for identifying patterns using clustering, which is an unsupervised learning method. The purpose is to distinguish structures of node signals and patterns in the brain indicating the effect of an impact at specific timestamps. It might involve decomposing the signal into alpha, beta, theta, and delta waves to look for structure, and then use a Markov model or hidden Markov model to carry out the clustering task.

3.3. Data Science Techniques

3.3.1 Source Data

We will be using cleaned EEG data from 8 provided experiments as input for our main goals.

Each experiment has the following files and attributes:

- `fixica.set` - metadata
- `fixica.fdt` - raw data
- `impact locations.mat` - impact timestamps
- `fixedareas.mat` - record of values that have been altered in cleaning process
- 19 channels, one for each electrode
- 33 head impacts per experiment including shams
- Sampling rate of 2048 Hz (samples per second)
- Approximate duration of 1.5 hours or less

Table 1: First Ten Rows and Columns of Example Source Data for Experiment 927

time	Fp1	Fp2	F7	T3	T5	F3	C3	P3	Fz
0	5.2500	6.6560	-4.4345	4.3041	17.6769	8.2750	5.2777	-5.2988	7.6702
0	5.2781	6.6954	-5.0002	4.0844	17.2995	8.1430	5.1968	-6.0566	7.6562
1	-0.7047	-2.4003	9.2241	4.5104	7.8099	4.4852	11.1368	20.4721	-2.4221
1	-0.6771	-2.3621	8.7526	4.3348	7.4980	4.3828	11.0669	19.8759	-2.4302
2	-0.8712	-3.7706	26.6630	4.5882	-3.3497	-3.0554	6.4791	19.8166	-6.2308
2	-4.6418	-7.0800	19.4258	0.4115	-15.8785	-6.1560	-1.2605	11.4035	-10.7741
3	-6.1009	-8.2283	20.6916	0.9807	-15.1462	-6.7221	0.1457	10.1121	-13.8165
3	-5.1490	-7.5291	15.2277	-1.6262	-17.0960	-8.2499	0.2797	7.6995	-11.8478
4	-3.2629	-6.3859	5.4595	-3.6072	-17.0394	-8.3204	0.1477	11.7201	-8.6690
4	-5.3662	-8.7086	-11.5473	-5.8980	-15.5566	-5.1943	0.7396	11.5066	-10.1058

3.3.2 Techniques

The Python visualization package will mainly be developed using the open source library MNE, which is designed for visualizing and analyzing human neurophysiological data. Custom visualizations may also be built with Matplotlib. Function development will be driven by the needs of the partner, improving ease-of-use compared to using MNE or Matplotlib directly. Clear documentation and code will be prioritized to allow package functionality to easily be updated following the completion of the Capstone. The main difficulties are expected to be processing and transforming the data into the format which MNE requires and building custom animations. We may also need to find solutions to reduce rendering times due to the high dimensionality of the data.

For the interactive UI we are planning to use an open source framework called Streamlit which is designed for creating web apps from Python scripts. Streamlit benefits from being lightweight and requiring no front-end experience. This will facilitate ease of updating in the future. The main difficulty will be to design a straightforward but informative UI with a large number of visualizations.

For the machine learning classification/clustering stretch goal, SciPy can be used to perform data wrangling and decompose data into frequency-specific bandwidths. We may use a Markov or Hidden Markov model for the clustering tasks, as recommended by our Capstone partner. Performing the clustering task using hidden Markov model could reduce the complexity of physiologic variables while retaining the significant signal structures (Asgari et al, 2019). Other researchers have historically used k-means clustering, support vector machine (SVM) or CNN models in the classification process, which are viable alternatives. The pipeline will be built using scikit-learn or PyTorch and can be delivered in either a Jupyter notebook or Python script. The main difficulty is that domain expertise is required for interpretation, so identifying the meaning of clusters will be difficult without significant assistance from the Capstone partner.

3.4 Timeline

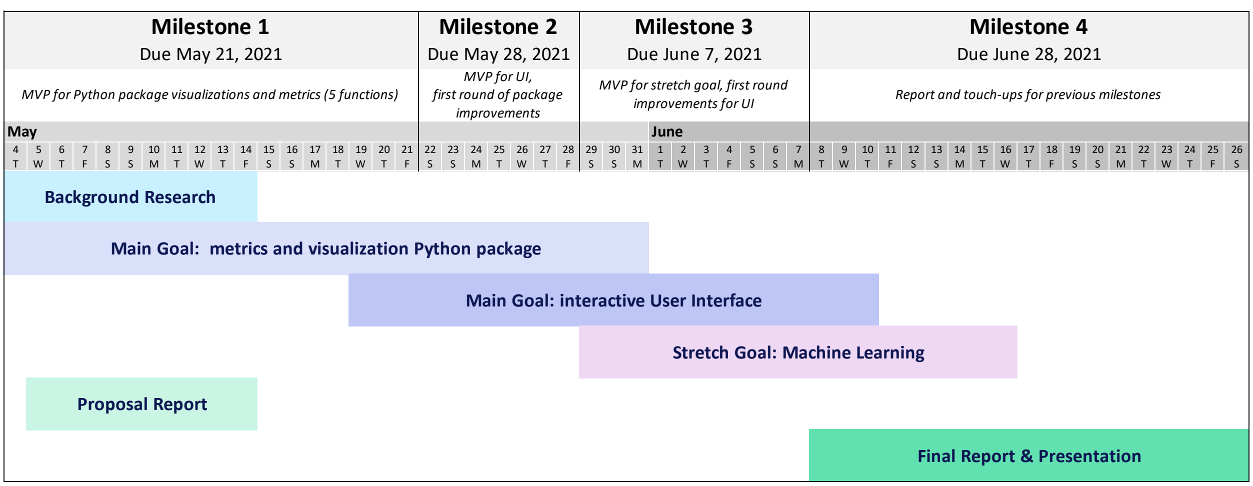

Figure 3: Timeline and Milestones

References

Asgari, Shadnaz PhD1,2; Adams, Hadie MD3; Kasprowicz, Magdalena PhD4; Czosnyka, Marek PhD3,5; Smielewski, Peter PhD3; Ercole, Ari MB BChir, PhD6 Feasibility of Hidden Markov Models for the Description of Time-Varying Physiologic State After Severe Traumatic Brain Injury, Critical Care Medicine: November 2019 - Volume 47 - Issue 11 - p e880-e885 doi: 10.1097/CCM.0000000000003966