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: June 22, 2021

3.1. Executive Summary

Our partner SimPL is a research lab that explores research questions concerning the human brain. The purpose of our project was to 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 proposed 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

We also proposed 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

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 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. Our goal was to extend the number of visualizations available to researchers, including those with a 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, future iterations of our machine learning stretch goal could uncover patterns in the data which could not be determined based on visualization alone.

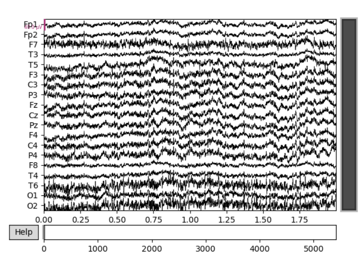


Figure 1: Example of a Standard Visualization for Two Seconds of 19 Channel EEG Raw Voltage Values

3.3. Data Science Methods

The Python visualization package was mainly developed using the open source library MNE, which is designed for visualizing and analyzing human neurophysiological data. Custom visualizations were built with Matplotlib and Plotly. The functions in our package were developed to improve ease-of-use over using MNE, Matplotlib, and Plotly directly. Clear documentation was generated using JupyterBook and the code is tested and documented to allow package functionality to easily be updated following the completion of the Capstone.

For the interactive user interface (UI) we used 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, which will facilitate ease of updating in the future. The downsides to Streamlit is reduced flexibility, only certain types of figures can be used and customization of styling is limited. However, for the purposes of this project we believe that the simplicity and ease of maintenance outweighs the need for additional features.

———-UPDATE———- 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 Data Product and Results

3.4.1 Python Package

The `simpl_eeg` Python package is able to produce advanced visualizations for specified time ranges of EEG data. The following visualization types are available:

1) Raw voltage plot - visualize raw voltage values changes over time for each node

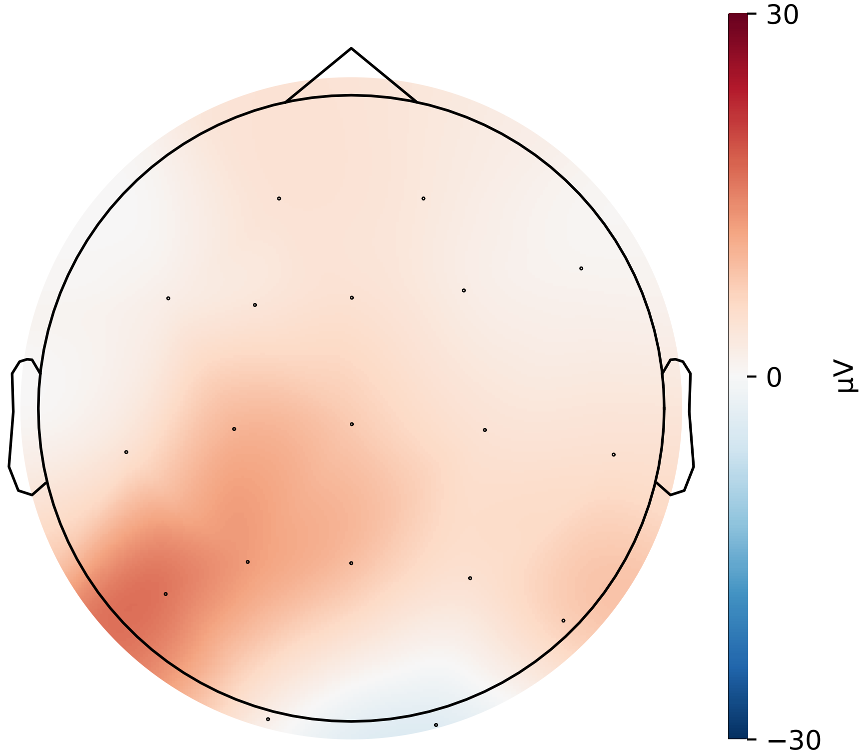
```
#raw_voltage.plot_voltage(epoch)
```

2) 2D head map - visualize a topographic heatmap of the voltage values mapped to a 2D model of a skull

```
#topomap_3d_head.topo_3d_map(epoch, 0)
```

3) 3D head map - visualize a topographic heatmap of voltage values mapped to a 3D model of skull

```
topomap_2d.plot_topomap_2d(epoch)
```



time: -0.2998s

4) 3D brain map - visualize a topographic heatmap of interpolated voltage values mapped to their presumed position on a 3D model of a brain

```
#topomap_3d_brain.plot_topomap_3d_brain(epoch, backend='matplotlib')
```

5) Connectivity - visualize pairwise connectivity measurements between nodes for specified time ranges

```
#connectivity.plot_connectivity(epoch)
```

6) Connectivity Circle - alternative perspective for visualizing pairwise connectivity measurements

```
#connectivity.plot_conn_circle(epoch)
```

With the exception of the Raw voltage plot, each visualization can be created as an animation to view changes over time or as a standalone plot. Detailed instructions on how to use the package can be found in our documentation.

3.4.2 User Interface

In addition to the package we built an interactive web application to serve as a UI for the package. The UI requires no coding experience, and is accessible by running a simple command. It also has the benefit of providing all the main visualizations in one place, so you can easily look at your data from several different perspectives. Customizing settings is made easy through the widgets available, which include the most widely used options for each package function.

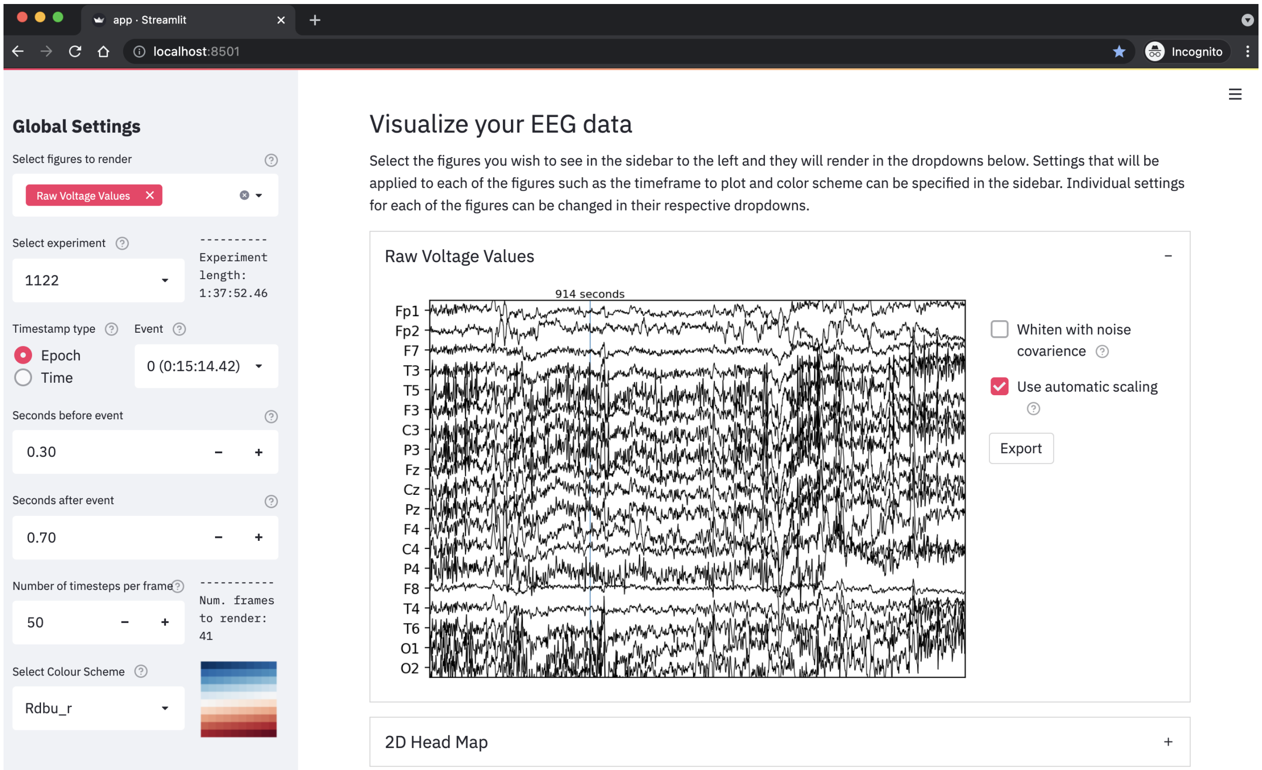


Figure 2: Streamlit User Interface

3.4.2 Stretch goal

———-UPDATE———- 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.

3.5 Conclusions and Recommendations

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