

# 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 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

Stretch Goal:

- 1) Unsupervised machine learning model to cluster and identify patterns in the EEG data

## 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 functionality 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 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.

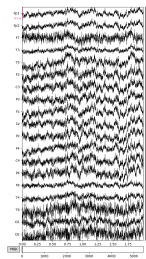


Figure 1: Example of a Standard Visualization for Two Seconds of 19 Channel EEG Raw Voltage Values

### 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.

For the unsupervised clustering method, we have tried both KMeans clustering model and Hidden Markov model to identify potential brain states in the data across different time periods. Brain states are considered as similar patterns of 19 electrodes over a period of time in the raw EEG data. The KMeans clustering method utilizes the KMeans model from Scikit-learn to separate data from each time periods into different clusters based on its distance from the center of the entire dataset. The Hidden Markov model method uses the GaussianHMM model from hmmlearn to learn the Gaussian distribution of the EEG signals and update the transmission probability matrix based on the changes of the distribution as it transitions from one time section to another. Furthermore, we looked into expanding our analysis to methods beyond what we didn't cover in Master of Data Science program such as Self Organizing Maps (SOM) and K-means and Dynamic Time Warping using `tslearn` package.

### 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

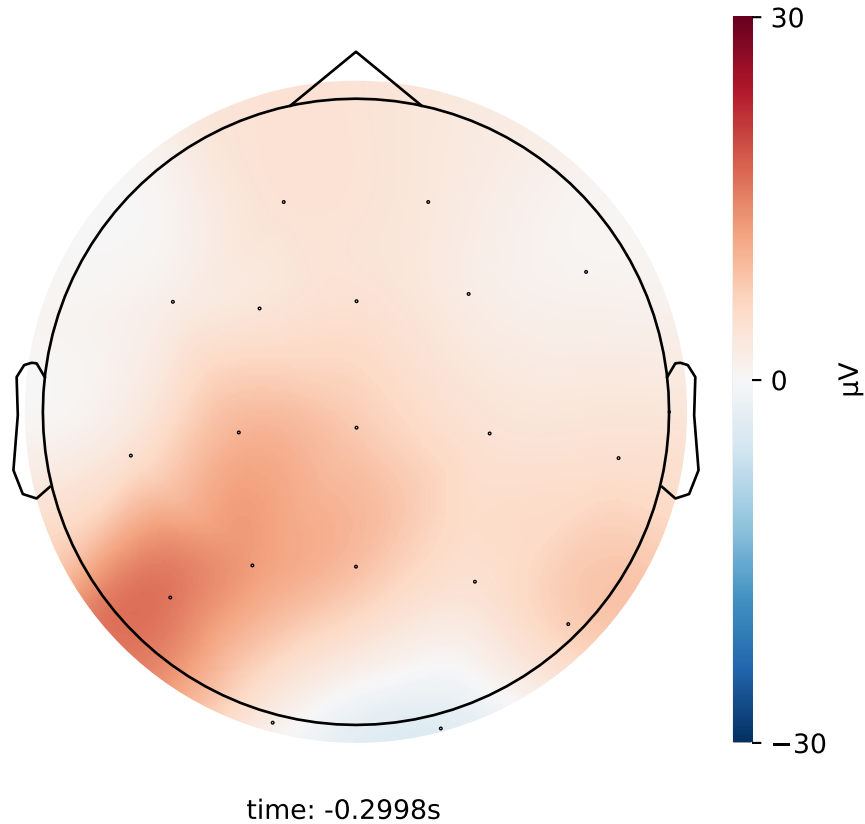
```
#raw_voltage.plot_voltage(epoch)
```

- 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)
```



- 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.

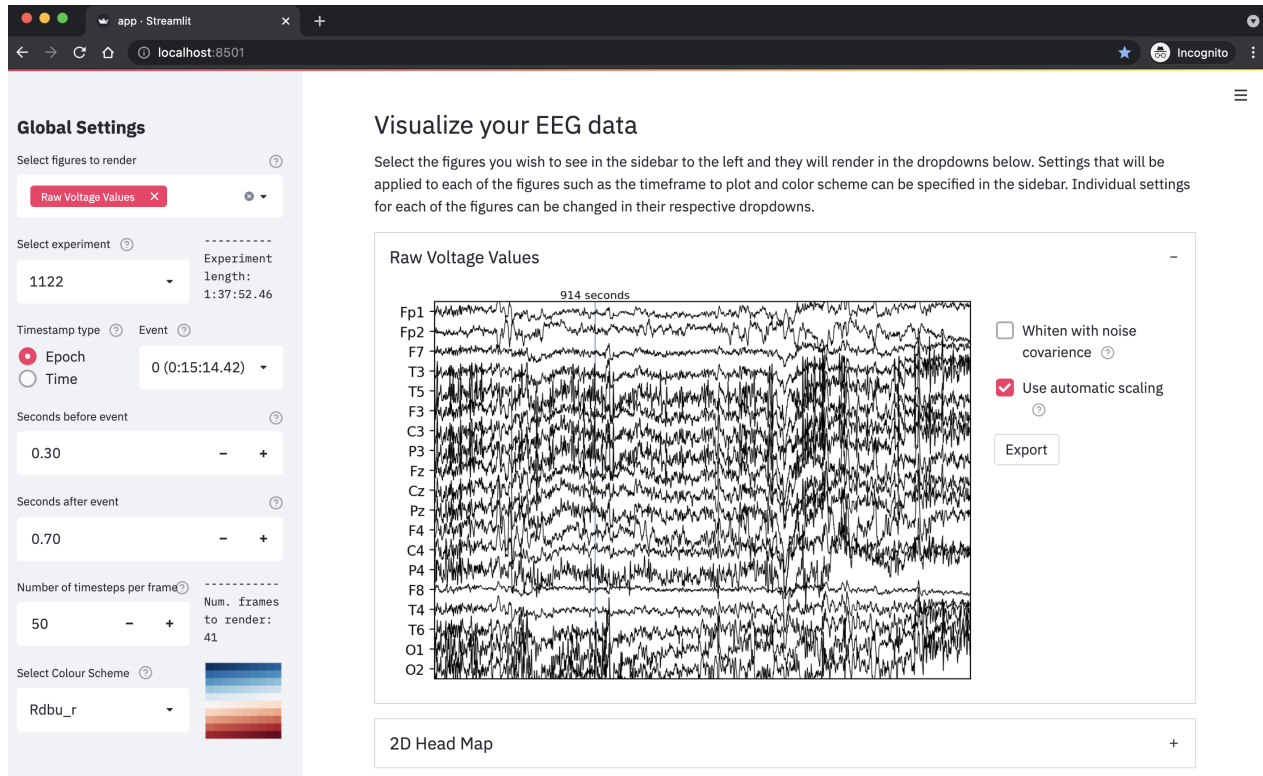


Figure 2: Streamlit User Interface

### 3.4.2 Stretch goal

Although we have tried different approaches for the stretch goal, due to the limited time and efforts we could allocate to the stretch goal so far, we haven't concluded any significant findings yet. However, we do think it would be useful to document all approaches we have tried so far to serve the partner as a good reference point to continue to work on the stretch goal in the future. Hence, we have decided to create a Jupyter notebook file for each approaches to outline the workflow from data preprocessing, model fitting to output visualization for each approaches. We have also included a section called **Next steps** in some applicable notebooks to let the partner know that there are some next steps potentially useful in finding the significant results but we haven't had time to implement yet, such as using rolling means and sliding windows as feature engineering or including a customized visualization function for visualizing clustering outcome.

## 3.5 Conclusions and Recommendations

## References

Asgari, Shadnaz PhD<sup>1,2</sup>; Adams, Hadie MD<sup>3</sup>; Kasprowicz, Magdalena PhD<sup>4</sup>; Czosnyka, Marek PhD<sup>3,5</sup>; Smielewski, Peter PhD<sup>3</sup>; Ercole, Ari MB BChir, PhD<sup>6</sup> 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