

EEG Signal Processing

Junaid Jawed Khan

Table of Contents

TABLE OF FIGURES	3
ABOUT DATASET.....	4
EXAMINING RAW DATA	4
<i>Removing DC Offset</i>	5
<i>Extracting Epochs and Re-referencing</i>	6
POWER-SPECTRAL DENSITY(PSD)	8
PREPROCESSING	10
FINITE IMPULSE RESPONSE (FIR) FILTER.....	10
<i>Remove Drift-Effect</i>	10
<i>Removing 60Hz Line Noise</i>	11
REMOVING BAD CHANNELS	11
INDEPENDENT COMPONENT ANALYSIS (ICA)	13
<i>Brief Theory</i>	13
<i>Mathematical Definition</i>	13
<i>FastICA Result</i>	15
<i>Cleaned EEG Data</i>	21
FEATURE EXTRACTION	21
TIME-FREQUENCY ANALYSIS	21
STFT	22
<i>STFT Window Impact</i>	23
WAVELET TRANSFORM	23
<i>Flowchart of CWT</i>	24
<i>Time-Frequency Analysis with EEGLAB</i>	25
<i>EEGLAB Time-Frequency Analysis Using Wavelet Transform</i>	26
<i>Coherence & Synchrony</i>	29
CONCLUSION	33
REFERENCES.....	34

Table of Figures

FIGURE 1: CHANNEL LOCATIONS OF THE SELECTED DATASET. (RIGHT: 10-20 SYSTEM NAME, LEFT: CHANNEL NUMBER)	4
FIGURE 2: IMPORTED SUBJECT 8 CHANNEL DATA PLOT.	5
FIGURE 3: CHANNEL DATA PLOT WITHOUT DC OFFSET.	5
FIGURE 4: INFORMATION ABOUT THE DATASET.....	6
FIGURE 5: RE-REFERENCING DATASET USING AVERAGE OF ALL CHANNELS CAPTURING BRAIN ACTIVITY.	7
FIGURE 6: EPOCHED AND AVERAGE REFERENCED CHANNEL DATA - (SHOWING ONLY 5 EPOCH WINDOW).....	8
FIGURE 7: POWER SPECTRAL DENSITY OF UNFILTERED CHANNEL DATA IN FIGURE 6.	9
FIGURE 8: CHANNEL DATA AFTER APPLYING HIGH-PASS FILTER AT 1HZ.	10
FIGURE 9: POWER SPECTRAL DENSITY OF CHANNEL DATA AFTER APPLYING NOTCH FILTER TO GET REMOVED 60HZ LINE NOISE.	11
FIGURE 10: REMOVING BAD CHANNELS M1 AND M2.....	12
FIGURE 11: INDEPENDENT SOURCES 1-35.	15
FIGURE 12: INDEPENDENT SOURCES 36-67.	16
FIGURE 13: EYE SOURCES.	17
FIGURE 14: CHANNEL NOISE SOURCES - PART 1	18
FIGURE 15: CHANNEL NOISE SOURCES - PART 2	19
FIGURE 16: CHANNEL NOISE SOURCES - PART 3	20
FIGURE 17: CLEANED CHANNEL DATA.....	21
FIGURE 18: STFT AND WAVELET TRANSFORM RESOLUTIONS [12].	22
FIGURE 19: STFT OUTPUT [13].	22
FIGURE 20: WAVELET TRANSFORM FLOW CHART [13][12].	24
FIGURE 21: CHANNEL LOCATIONS BY NAME AFTER REMOVING M1 AND M2.....	25
FIGURE 22: CHANNEL LOCATIONS BY NUMBER AFTER REMOVING M1 AND M2.....	26
FIGURE 23: EEGLAB TIME-FREQUENCY PARAMETERS.	27
FIGURE 24: TIME-FREQUENCY DECOMPOSITION OF CHANNEL O1.	28
FIGURE 25: TIME-FREQUENCY DECOMPOSITION OF CHANNEL OZ.	29
FIGURE 26: CROSS-CHANNEL COHERENCE PARAMETERS.....	30
FIGURE 27: PHASE SYNCHRONIZATION BETWEEN CHANNELS OZ AND O1.....	31
FIGURE 28: POWER SPECTRAL DENSITY OF CHANNELS.....	32

About Dataset

For EEG signal processing using MATLAB I chose to process **subject 8** in **Resting-state**[1]. The dataset is a 72-channel Electroencephalography (EEG) readout of a subject's brain activity at a sampling rate of 256 Hz in a wakeful resting state with their eyes open and close.

Each of the 72-channels is an electrode that is measuring synaptic activity of its underlying cortical neurons. The 10-20 electrode placement standard is used for electrode placement. The 10-20 system is based on total distance from front-to-back(nasion-to-inion) and left-to-right(left tragus-to-right tragus) of the skull. The 72 electrodes are placed based on 10% or 20% of the distance from either the inion, nasion, right tragus or left tragus[2]. The following figure shows the placement of the 72-electrodes used for measuring the brain activity of the dataset processed in this report.

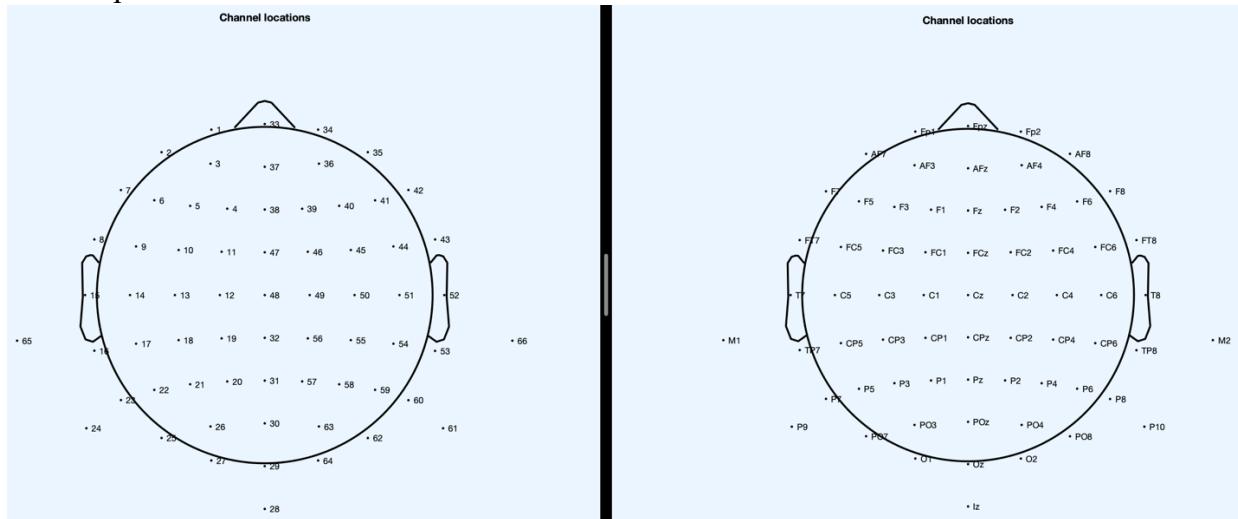


Figure 1: Channel locations of the selected dataset. (Right: 10-20 System Name, Left: Channel number)

Examining RAW Data

EEG is a non-invasive electric readout of brain activity. The electrodes are placed on the scalp and the measurements taken by the electrodes include undesired information called artifacts. The amplitude of artifacts in the captured EEG signals are often larger than the amplitude of brain data which potentially decreases the signal to noise ratio, biasing the data analysis and the potential results [3]. Possible artifacts to look for when analyzing raw EEG data are:

- Low frequency events such as eye movements
- Linear trend
- Transient high frequency events such as muscle contraction
- Line noise & electrode discontinuity [2]

Due to the existence of artifacts in the EEG signals captured by the electrodes, we need to clean the EEG data before we can extract desired features from them.

Figure 2 is the raw uncleaned EEG data of subject 8, imported into EEGLAB.

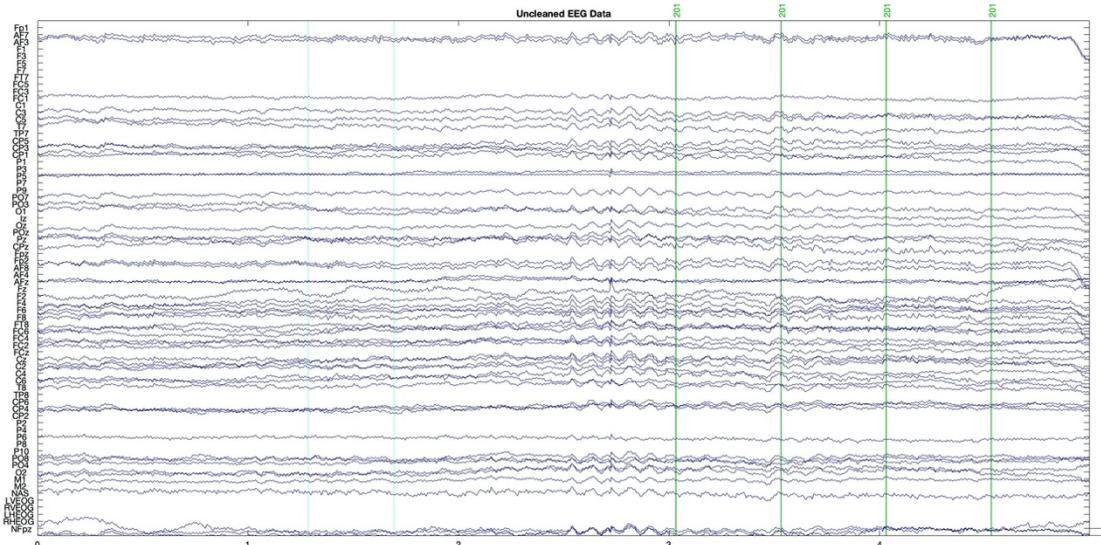


Figure 2: Imported Subject 8 channel data plot.

In Figure 2, we don't see some of the channel data because some of the channels are shifted by a DC offset. This DC offset introduces large artifacts at the boundaries of your signal. Therefore, before any preprocessing step, I removed the DC offset from my dataset.

Removing DC Offset

I removed the DC offset from my EEG dataset by plotting my channel data as shown in Figure 2, followed by Display->Remove DC Offset. This will remove the DC offset from all the channels. Figure 3 is the resulting plot of all channel data.

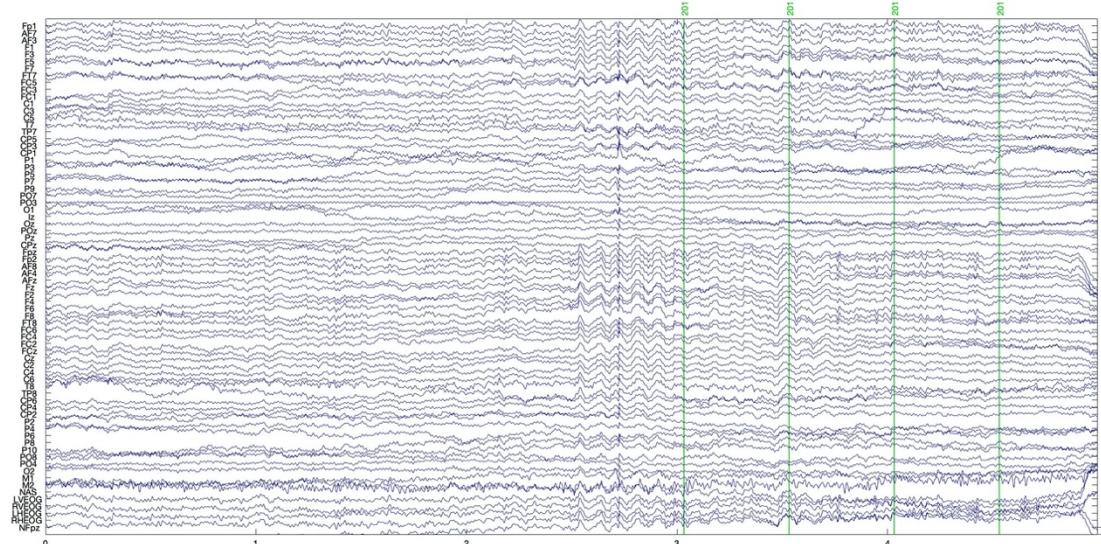


Figure 3: Channel data plot without DC offset.

Extracting Epochs and Re-referencing

After importing the raw data into EEGLAB, the information about the dataset showed no channel locations, about 960 events in this dataset, and Reference of the dataset was unknown. Figure 4 shows this information.

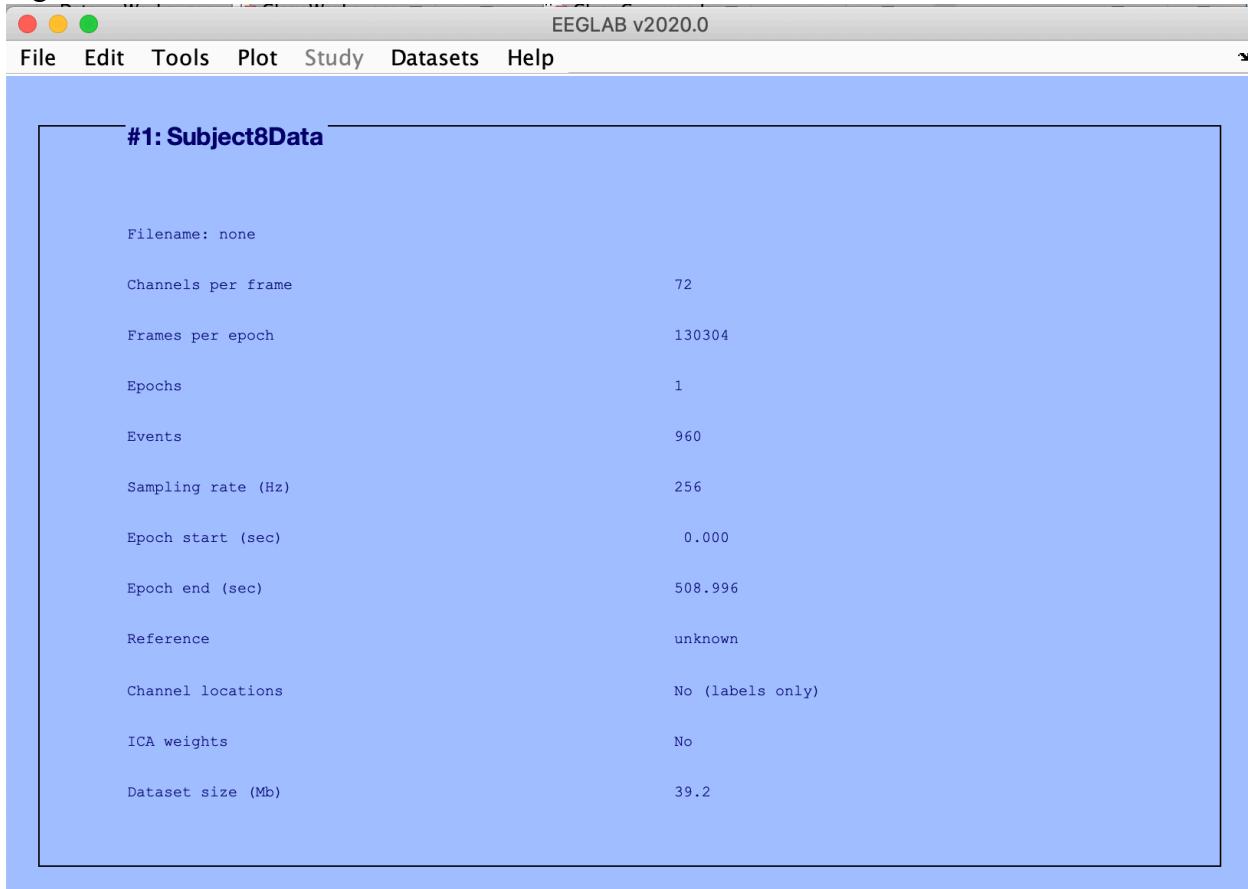


Figure 4: Information about the dataset.

Adding Channel Locations and Extracting Epochs

I added the channel locations by going to Edit -> Channel Locations -> OK. A continuous EEG signal can be divided into several small segments for analysis. These small segments are called Epochs. In Figure 4, there is a single epoch covering the entire continuous EEG data. The selected dataset has event markers as shown in Figure 2 and Figure 3. To study these events you can divide your continuous EEG data into epochs. To divide my continuous EEG data into epochs I went to Tools->Extract Epochs. In Figure 4, the imported dataset has unknown referencing for the channels.

Re-referencing

There are 3 referencing methods available for measuring EEG data. They are as follows:

- Common Reference
- Bipolar Montage
- Average Reference[2]

In Common Reference method, one of the electrodes on the scalp is used as a reference electrode. All the electrodes are measuring differential voltages with respect to the single common electrode. Usually, this common electrode is placed behind the earlobes [2].

In Bipolar Montage referencing, an electrode placed in another region of the brain is used as reference by all the electrodes. This method of reference distorts the measured signal preventing you from analyzing the bigger picture of the brain activity [2].

The Average Referencing method is the optimal referencing method and the most recommended method. The reference is the average of all the electrodes that are capturing brain activity (i.e. you exclude electrodes that are placed for capturing eye artifacts). By taking the average of all electrodes that are capturing brain activity, each electrode is essentially capturing differential voltage with respect to a zero line.

Since, average referencing is the most optimal referencing method employed for EEG signal processing, I decided to re-reference my imported data to the average of all the channels in my dataset except the channels that are capturing eye artifacts as shown in Figure 5.

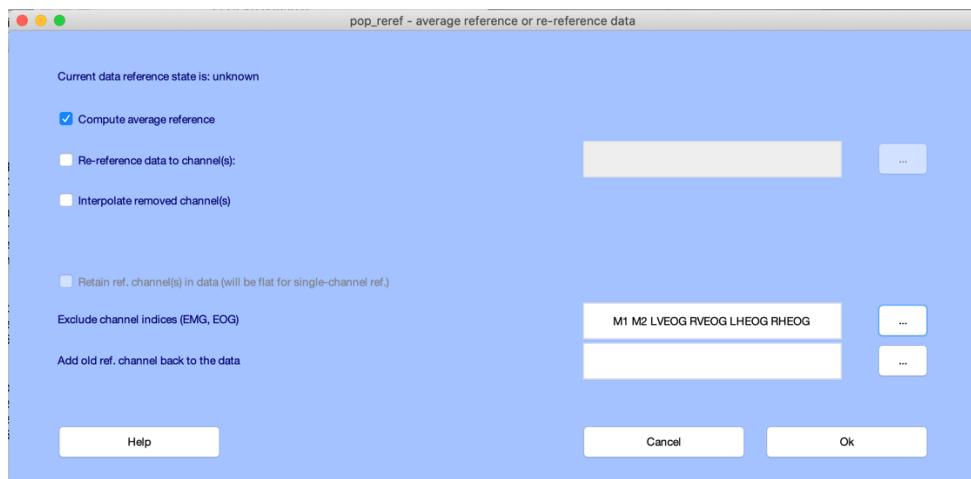


Figure 5: Re-referencing dataset using average of all channels capturing brain activity.

I excluded channels M1 and M2 from average calculations because their labels are not according to the 10-20 electrode naming convention, so I'm assuming they're not actually capturing brain activity. Figure 6 is the channel data plot after I added channel locations, divided my dataset into epochs and re-referenced by dataset using Average Referencing method that we learned in class.

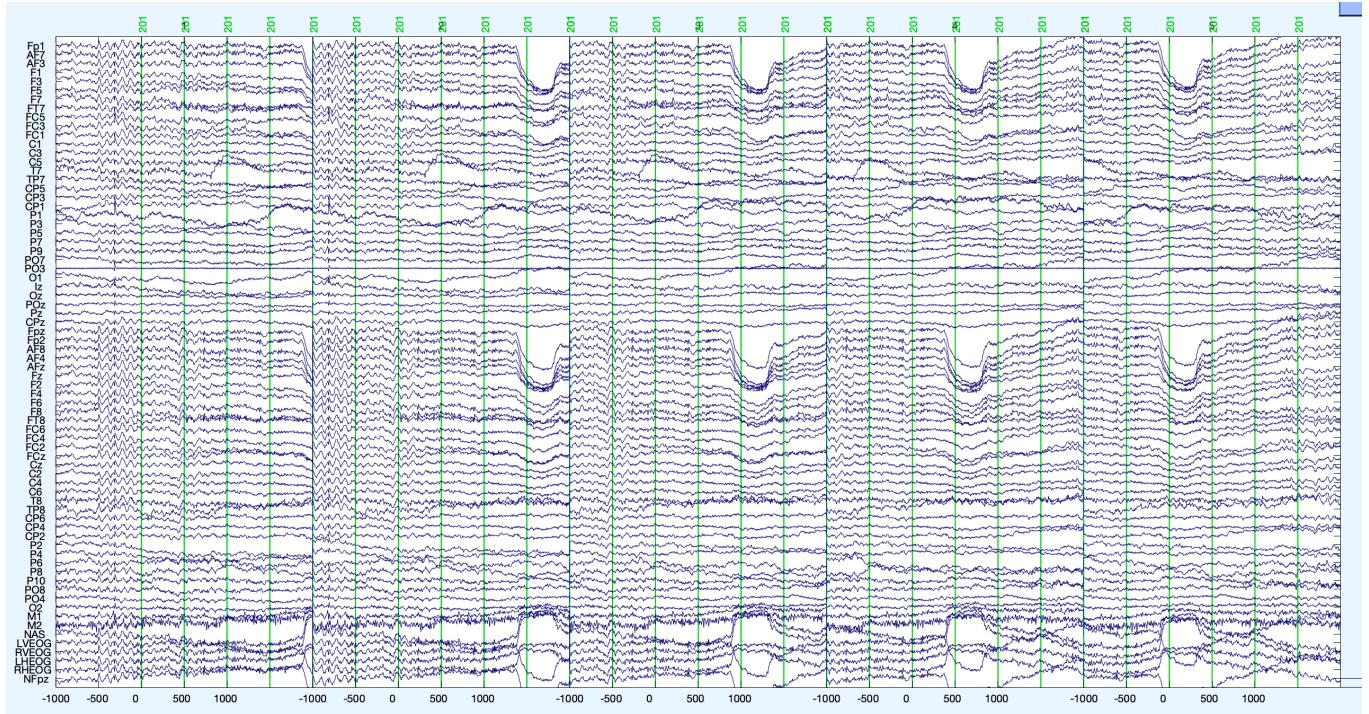


Figure 6: Epoched and Average Referenced Channel Data - (showing only 5 epoch window).

Another way to detect artifacts in your raw EEG is to look at the Power-Spectral Density(PSD) of your channel data.

Power-Spectral Density(PSD)

According to Fourier analysis, a signal can be decomposed into a sum of sinusoids of different frequencies[4]. A power spectrum describes the power distribution among these frequency components of a given signal[5]. To plot the PSD of the 72-channels, each channel's EEG signal is transformed from the time-domain to frequency domain, with the help of Fourier Transform[4].

Fourier transform is the formal name of the decomposition process to approximate a given signal in time-domain to a sum of sinusoids of its frequency content [2]. Equation 1 and Equation 2 are the mathematical formulas for calculating Fourier transform of a continuous time signal and discrete time signal, respectively.

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad \text{Equation 1}$$

$$X[k] = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\omega n} \quad k = 0, 1, 2, \dots, N-1 \quad \text{Equation 2}$$

For this assignment we have discrete time data for each of the 72-channels. Therefore, to transform the discrete-time signal to frequency domain, we employ Equation 2. Following the Fourier transform of each channel, we can calculate the power of each channel using Equation 3.

$$Power = \frac{X[k]^2}{2} \quad \text{Equation 3}$$

EEGLAB toolbox of MATLAB carries out the steps explained above to plot the PSD of the imported dataset. The PSD plot will help us further visualize any artifacts that we should remove from the dataset in the preprocessing section before we can extract features. Figure 7 is the PSD of the unfiltered channel data that we plotted in Figure 6.

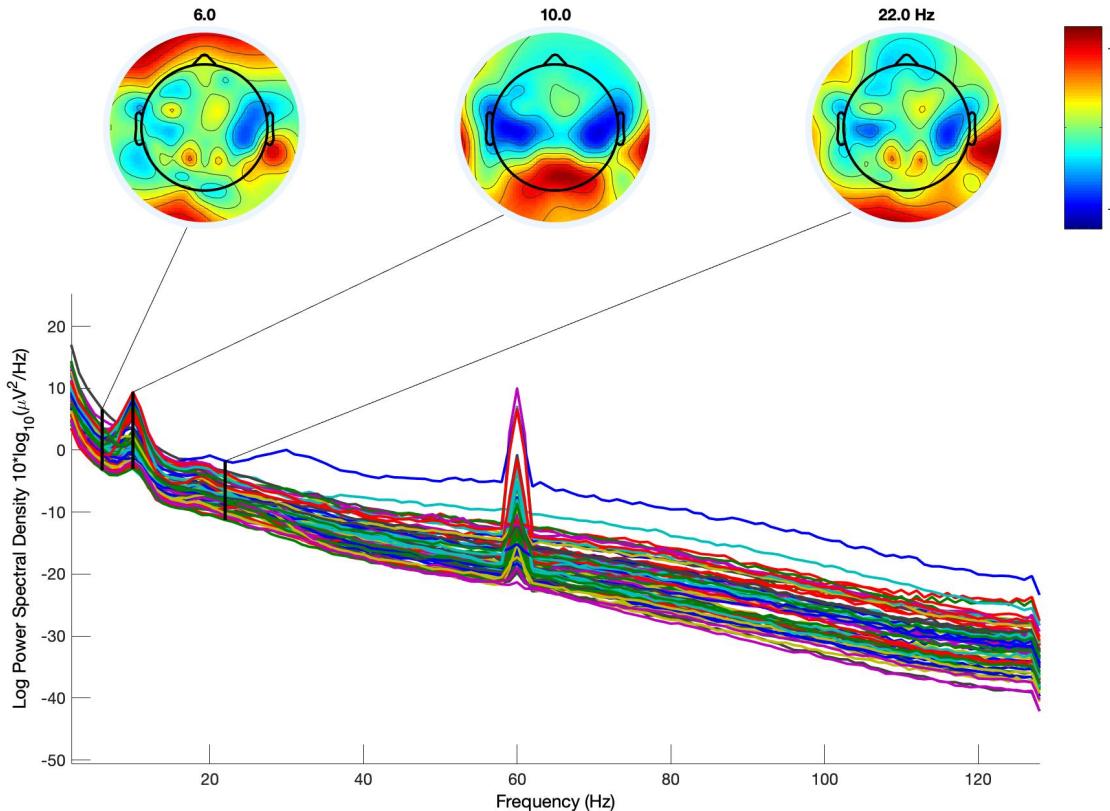


Figure 7: Power Spectral Density of unfiltered channel data in Figure 6.

The PSD shown in Figure 7 was plotted from 0-128Hz due to the sampling rate set at 256Hz as shown in Figure 4. According to Nyquist theorem, with a sampling frequency of 256Hz, you can at most digitize a 128Hz sinusoid. The 128Hz is called the Nyquist frequency [6].

A higher PSD value is proportional to higher activity at that particular frequency. The brain wave associated with resting is the Alpha wave and it typically has a frequency in the range of 8-12Hz. There are other brain waves that have frequencies around the Alpha waves, namely the Theta wave and Beta wave. Theta waves have a frequency in the range of 4-7Hz and Beta waves have

frequency in the range of 12-28Hz. Therefore, to extract features that confirm the subject is in resting state I decided to highlight frequencies 6Hz, 10Hz and 22Hz in my PSD plots, that are within the frequency ranges of Theta, Alpha and Beta bands.

In Figure 7, we can notice the existence of 60Hz line noise in all channels. In the preprocessing section of this report, we will remove this artifact along with others.

Preprocessing

Finite Impulse Response (FIR) Filter

A FIR filter is one of two types of filters used in Digital Signal processing. A FIR filter has no feedback when given a finite impulse and hence it eventually settles to zero without affecting the phase of the input signal [7]. These filters can be used to remove desired frequencies from a given input signal. I employ this basic filter to remove drifting artifact and the 60Hz line noise from my channels.

Remove Drift-Effect

First, I applied a high-pass filter at 1Hz to remove slow, possibly large amplitude, drifting effect seen in Figure 6. This will smoothen out EEG plot in time-domain and will also reduce the variance in PSD seen in Figure 7. Figure 8 shows the time-plot of my channel data after applying the high-pass filter at 1Hz.

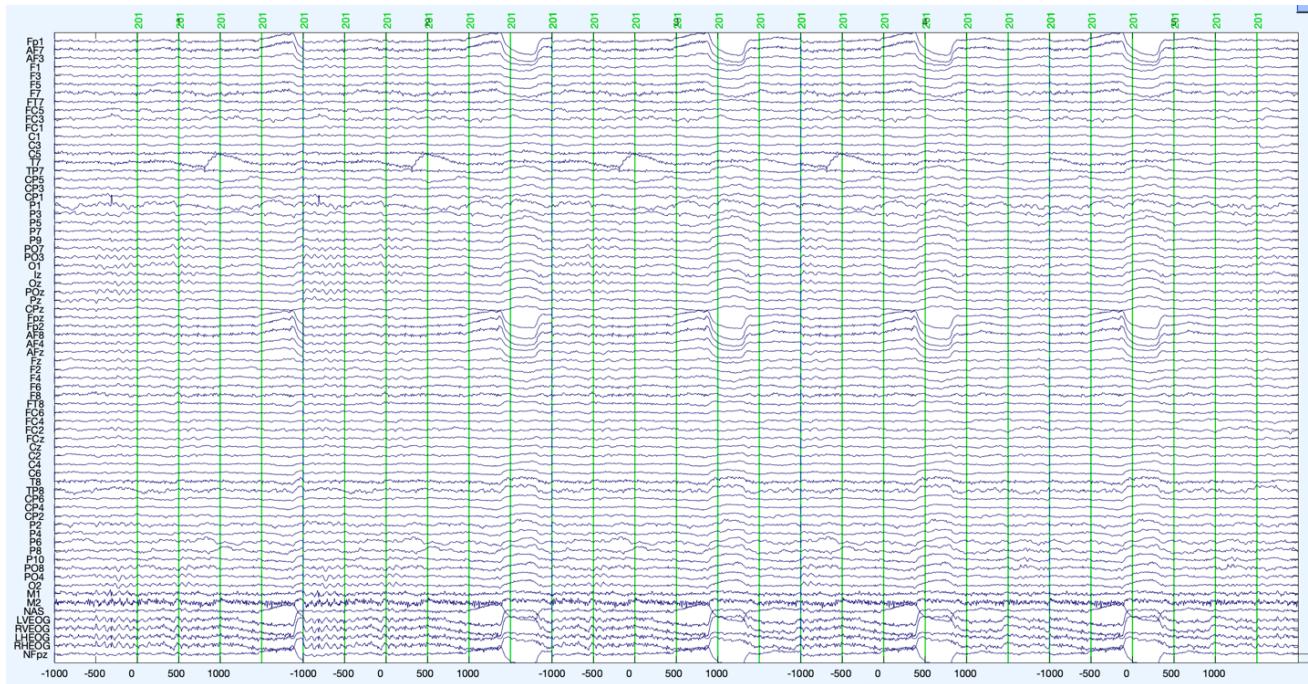


Figure 8: Channel data after applying high-pass filter at 1Hz.

As you can see in Figure 8, after applying the high-pass filter at 1Hz, the drift in each channel has been removed and channel data looks smoother in comparison to Figure 6.

Removing 60Hz Line Noise

Recalling that Figure 7 showed the existence of 60 Hz line noise in my dataset. The 60 Hz line noise can be removed from the dataset by employing one of the following methods:

1. A basic FIR filter with lower passband frequency set at 60 Hz.
2. Using a notch filter with lower passband frequency set at 58Hz and the upper passband frequency set at 62 Hz.

You could use either of the 2 mentioned filters to remove the 60 Hz line noise from your signal, I personally went with the notch filter. Figure 9 is the PSD of channel data after I applied a notch filter using eeglab gui.

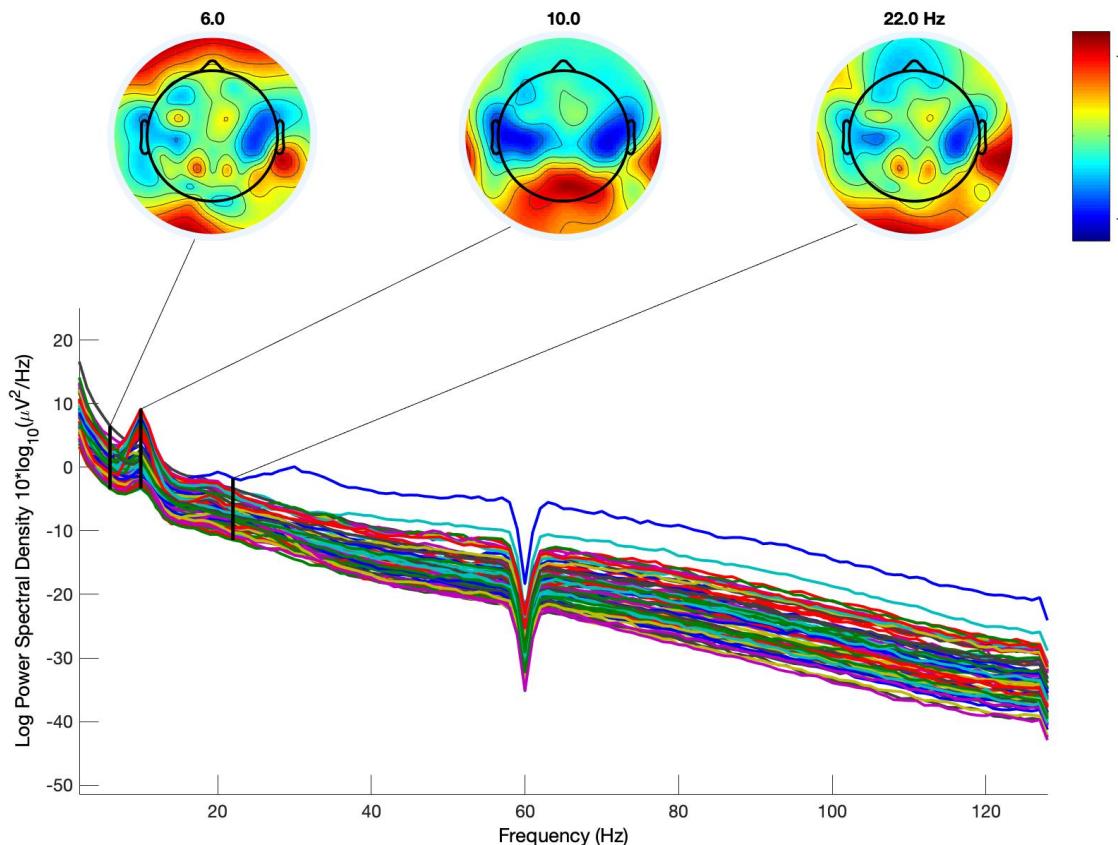


Figure 9: Power Spectral Density of channel data after applying notch filter to get removed 60Hz line noise.

Comparing Figure 9 with Figure 7, we can notice that we have removed power activity contributed by the 60Hz frequency to our channel data. Also, note the decrease in power variance due to the high-pass filter we applied at 1Hz, to remove the drifting effect.

Removing Bad Channels

The dark blue plot in the PSD of channel data shown in Figure 9 correspond to Channel M2. As I mentioned before, the channels M1 and M2 do not follow the electrode naming conventions of

the 10-20 standard system, so I assumed they're not recording brain activity. Also, looking at channel plot of M2 in Figure 8, we can notice the strange high frequency plot that doesn't correlate to the surrounding channel readings. Also, notice that M1 is a constant line plot throughout the 5 epochs shown in Figure 8.

Since M1 and M2 channel data does not resemble typical brain data, they can be removed from the dataset without affecting the actual brain data. Therefore, I decided to remove Channels M1 and M2 from the dataset by going to Plot->Select data -> Select Channels M1 and M2 from Drop down list -> Check Remove from Dataset. Figure 10 is PSD of channel data after removing channels M1 and M2.

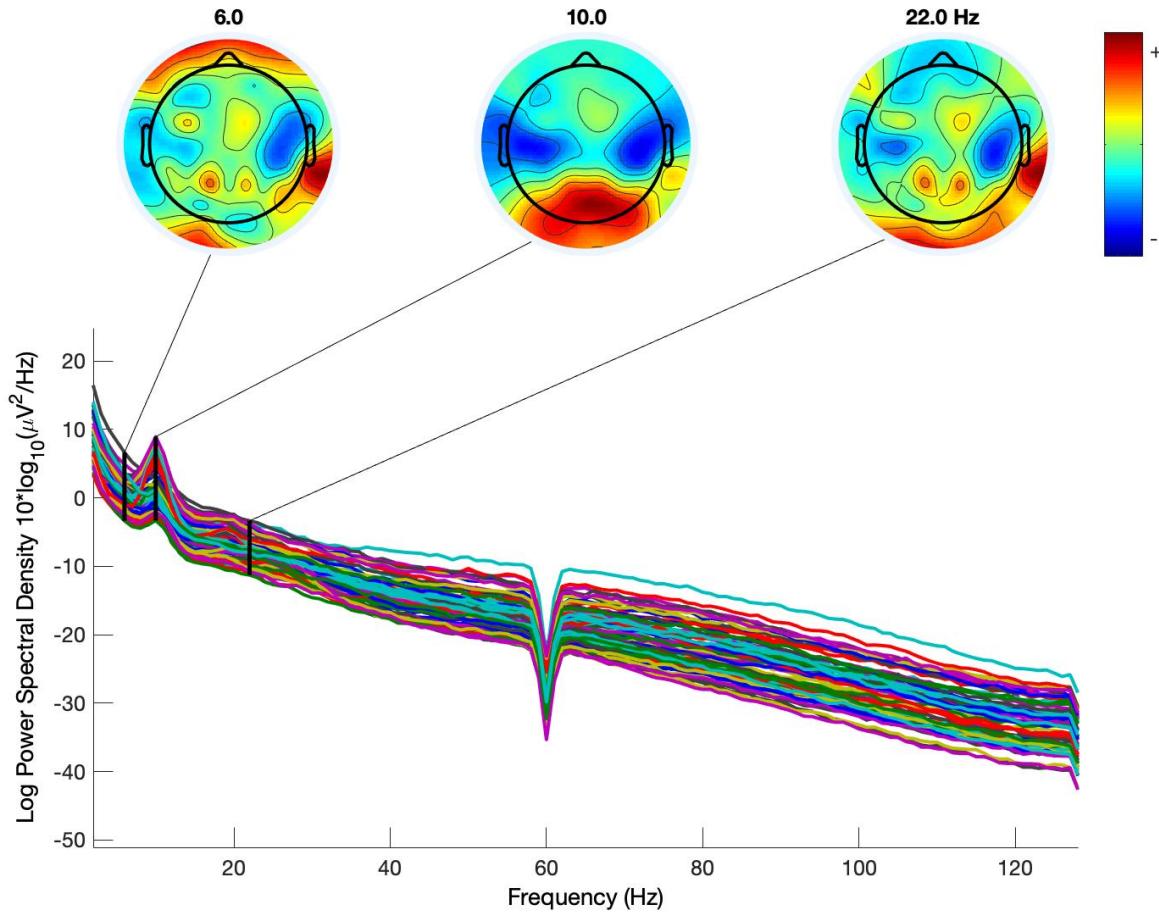


Figure 10: Removing Bad Channels M1 and M2.

After removing bad channels from my dataset, I employed a computationally expensive algorithm to remove artifacts that are not visibly affecting my dataset, these artifacts include eye movements and muscle contractions. The technique I employed for identifying these artifact is known as Independent Component Analysis(ICA). It is important to point out that ICA is sensitive to noise at low frequencies which is why it is crucial to remove the drifting effect in each channel before moving forward [8].

Independent Component Analysis (ICA)

Brief Theory

Independent Component Analysis is a signal processing method used to separate independent and uncorrelated sources in a mixed signal[9]. ICA is one of many techniques that can be employed to identify sources that are statistically independent and contribute to our measured EEG signal from the scalp [10]. By identifying the sources contributing to our measured EEG signal, we can recreate the EEG signal containing brain data important to us by excluding the sources that are interfering with our study.

For example, when measuring the brain activity of an individual in restful state, our electrodes pick up electrical information from eye movements & muscle contractions. If we identify these sources in our measured EEG signal, we can recreate the EEG signal without including these two sources. By doing so, our dataset will only have data corresponding to our study of resting state of an individual.

Mathematical Definition

Suppose we have n sources, defined as S_1, S_2, \dots, S_n and let X_1, X_2, \dots, X_j be a linear combinations of all these sources as shown in Equation 4. Equation 5 is the matrix form of the linear combination of these sources.

$$X_j = A_{11}S_1 + A_{12}S_2 + \dots + A_{1n}S_n \quad j = 1, 2, 3, \dots \quad \text{Equation 4}$$

$$\hat{X} = A \cdot \hat{S} \quad \text{Equation 5}$$

When we measure brain activity from the scalp, we have EEG signals captured by the electrodes we placed on the scalp. Each electrode is picking up a signal that has contributions from various sources. We denote each electrode's measured signal as X_j . Therefore, we only have access to vector \hat{X} , from Equation 5.

Our goal is to determine the vector \hat{S} , whose entries are the independent sources contributing to each electrode's measured EEG signal. If we had access to matrix A , which contains the coefficients of the linear combination of each source, and if this matrix was invertible, we could easily compute the vector \hat{S} , as shown in Equation 6.

$$\hat{S} = A^{-1} \cdot \hat{X} \quad \text{Equation 6}$$

However, we don't have access to A matrix. This where the ICA algorithm comes in to play. Equation 7 is essentially what ICA is trying to accomplish.

$$\hat{S} = W \cdot \hat{X} \quad \text{Equation 7}$$

Based on Equation 7, the ICA algorithm takes in the mixed signal \hat{X} , and returns a weight matrix W and a source vector \hat{S} [9]. The weight matrix W contains the coefficients that correspond to the contribution of each source in \hat{S} , to each electrode \hat{X} .

Once we have the weight matrix W and sources \hat{S} from the ICA algorithm, we can identify the sources that are interfering with our study and regenerate the EEG dataset by excluding those sources from each electrode's measured brain activity.

Equation 8 shows how a source is removed from your dataset after carrying out ICA and determining weight matrix W and source vector \hat{S} .

$$\hat{X}_{cleaned} = W^{-1} \cdot \hat{S} \quad \text{Equation 8}$$

After you've computed the inverse of the weight matrix W and you've identified the source you want to remove, create a matrix Y that has all the weights (i.e. rows) except the row corresponding to the source you want to remove.

Finally, you can multiply the source vector \hat{S} by matrix Y, the resulting mixed signals $\hat{X}_{cleaned}$, will have contributions from all the independent sources of source vector \hat{S} , except the one that we removed. Equation 9 illustrates this.

$$\hat{X}_{cleaned} = Y \cdot \hat{S} \quad \text{Equation 9}$$

Now, we apply the ICA algorithm to our cleaned dataset using eeglab to identify the different sources that are contributing to each electrode of my dataset. I ran the 'fastica' script in eeglab to determine all the independent sources in my dataset, followed by plotting the ICA components with labels. Figure 11 and 12 is a heat-map with all the independent sources in my dataset.

FastICA Result

Figures 11 and 12 are the different sources contributing to channel data.

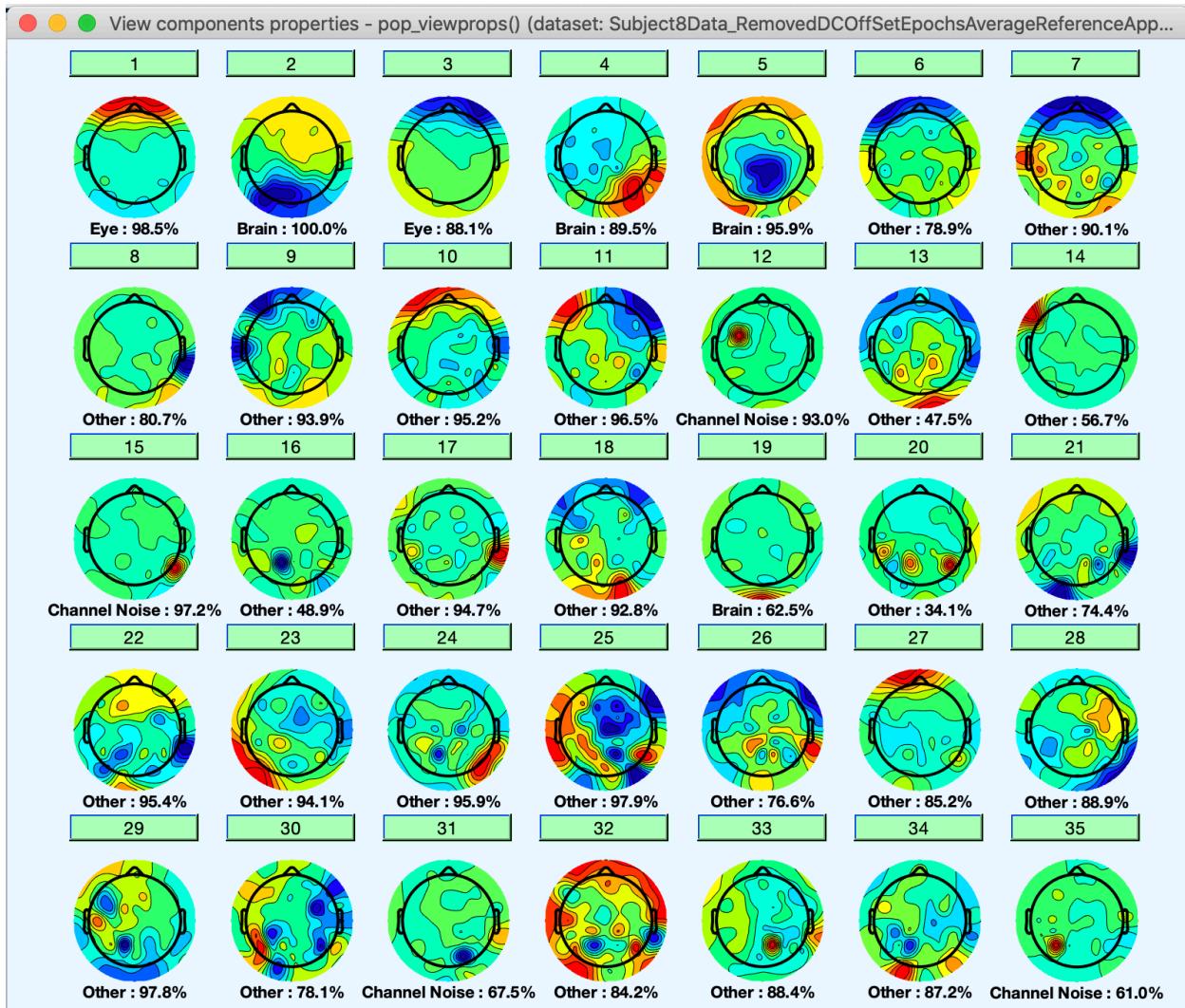


Figure 11: Independent sources 1-35.

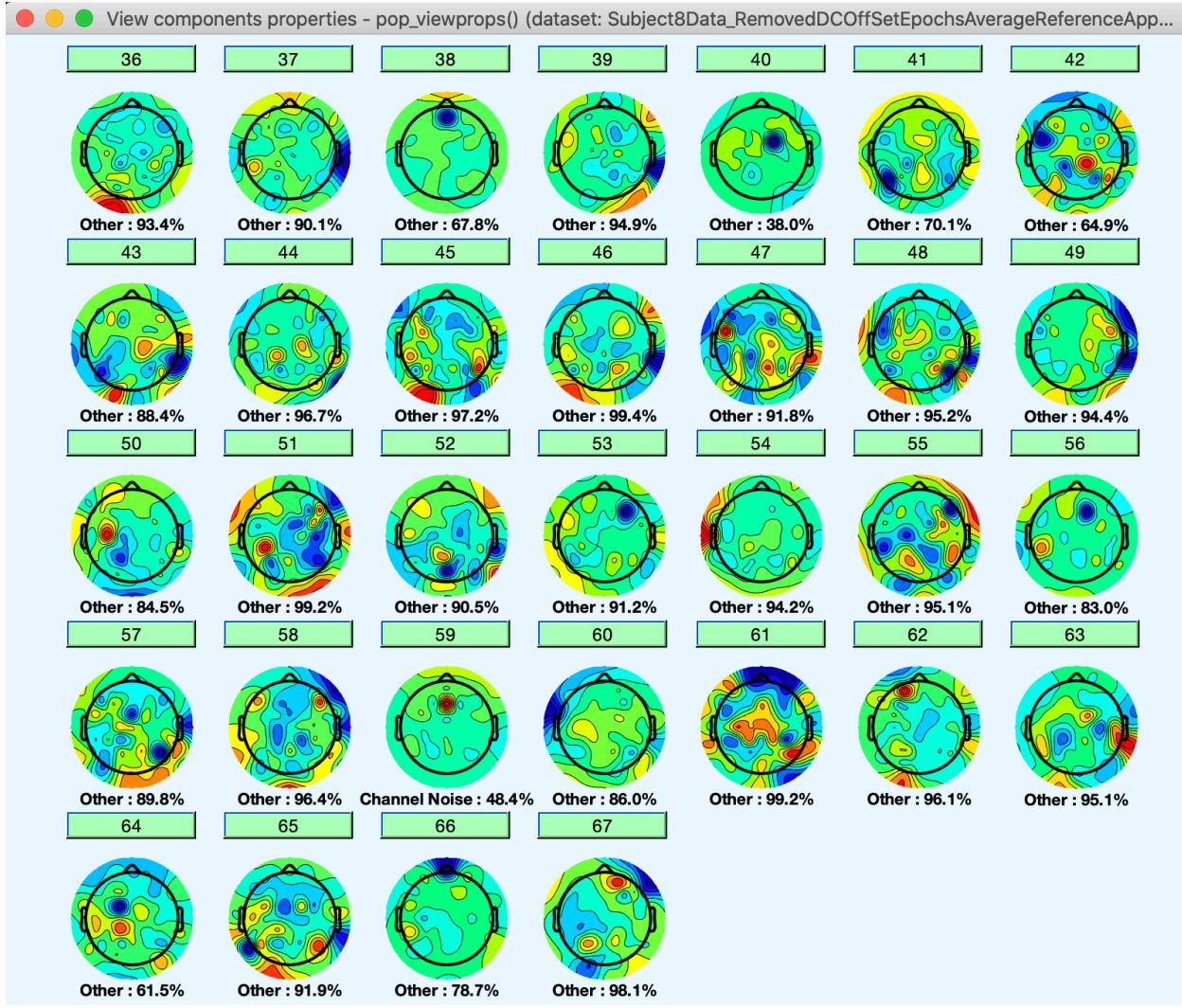


Figure 12: Independent sources 36-67.

ICA detected 67 independent sources in my dataset. The main purpose of ICA is to identify independent sources. The source predictions made by ICA algorithm are not always correct as ICA is mainly focused on trying to find the weight matrix W. It is up to the user to decide which source to remove from the dataset. To be on the safe side and not accidentally remove actual brain data, it's a common practice to only remove eye, muscle and channel noise components identified by ICA.

Therefore, based on this common practice and ICA component heat maps shown in Figure 11 and Figure 12, we can notice that the fastica algorithm is very confident that components 1& 3 correspond to eye movements and components 12, 15, 31, 35 & 59 are channel noise.

In the following text I provide justifications for the components that I decided to remove from my dataset.

Eye Artifacts

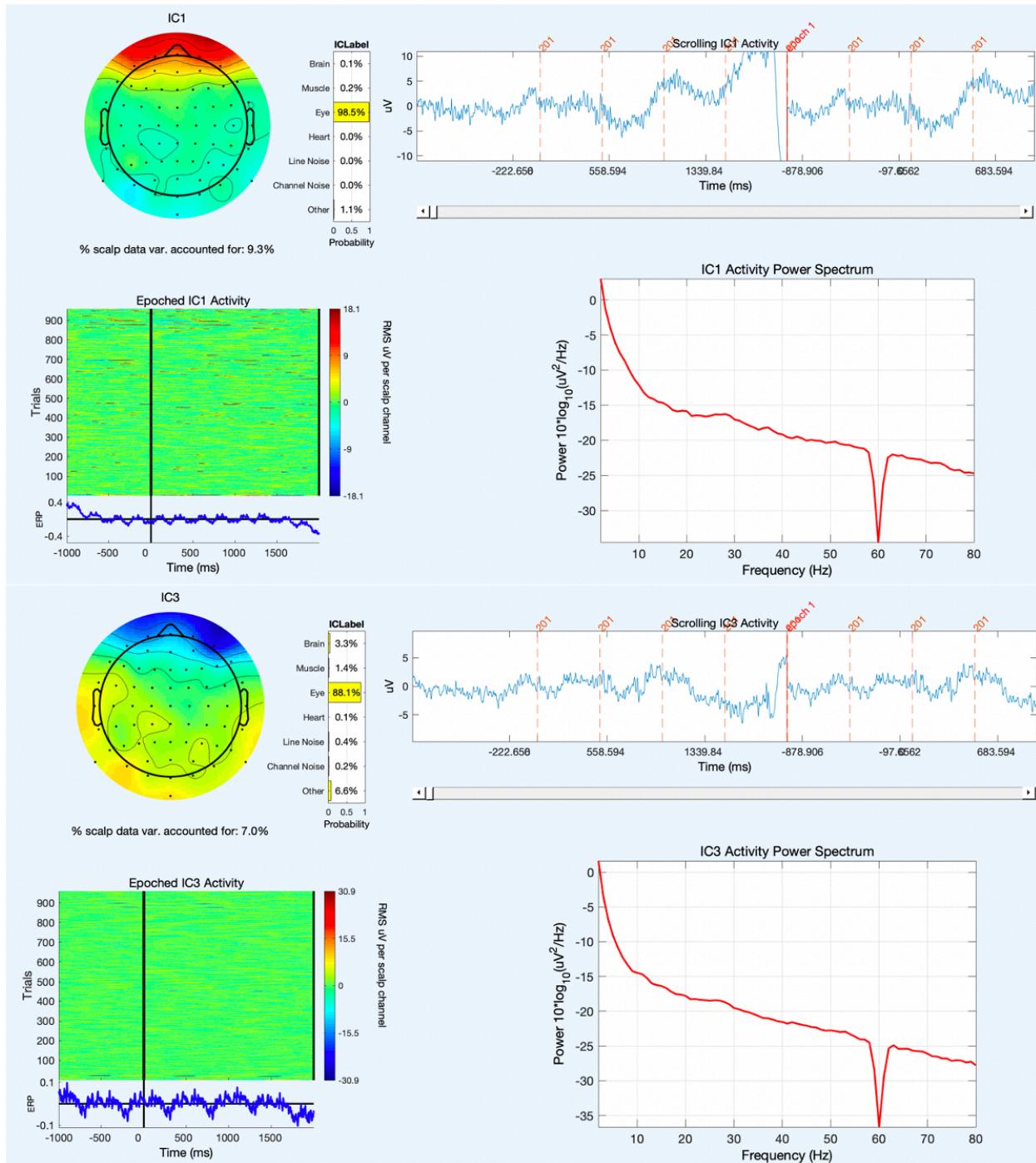


Figure 13: Eye Sources.

Based on the Power Spectrum of components 1 and 3 shown in Figure 13 , we can notice that these components have very low activity at the shown frequency range. Also, by looking at their individual activity plots with respect to time we can notice the continuous noisy behavior. The scalp map for each IC show that these two components are highly active at regions away from

brain(around the eyes). As these two components are not showing brain activity and ICA is over 80% confident that these are coming from the eyes, I've decided to remove these two sources from my channel data.

Channel Noise Artifacts

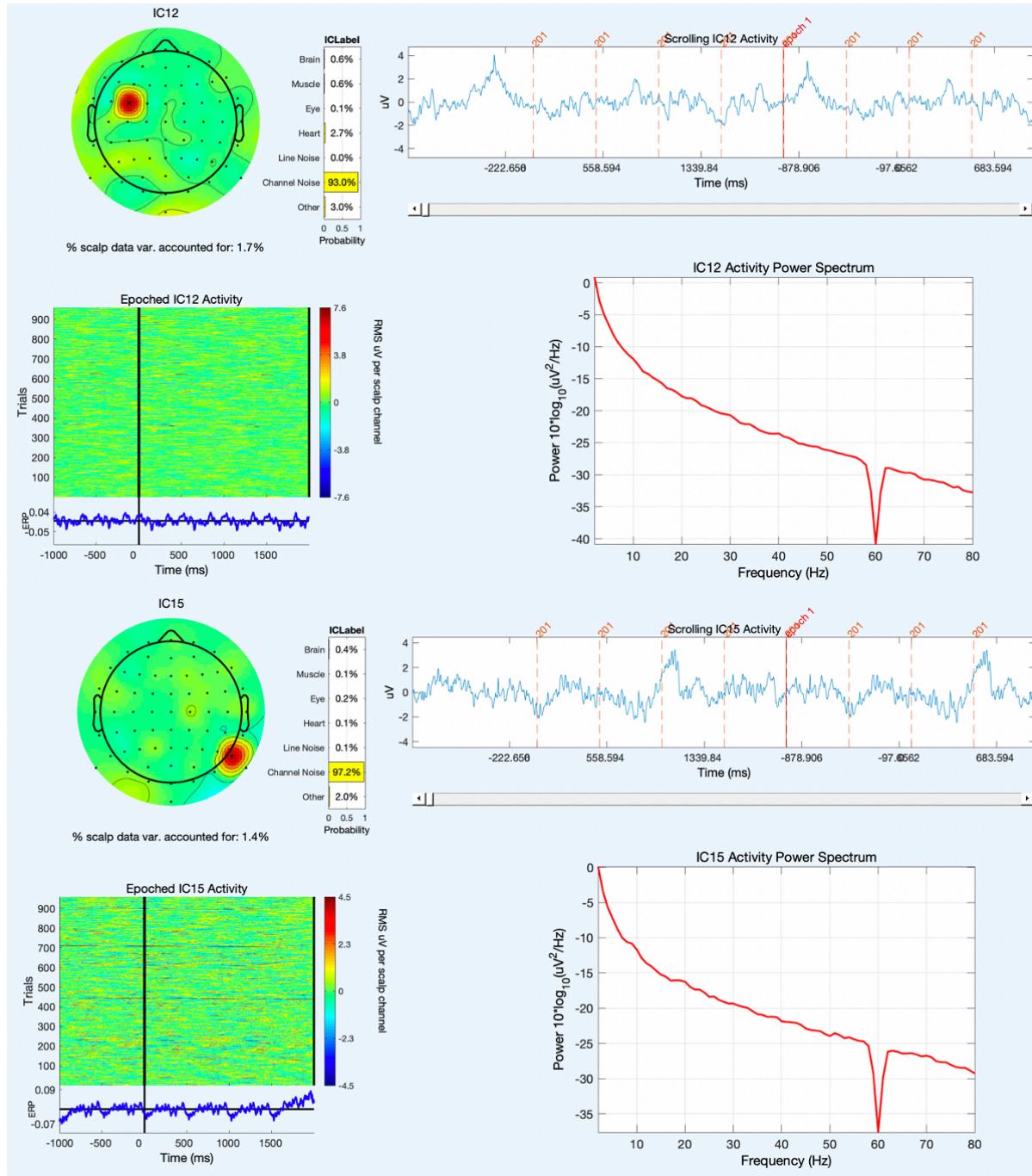


Figure 14: Channel Noise Sources - Part 1

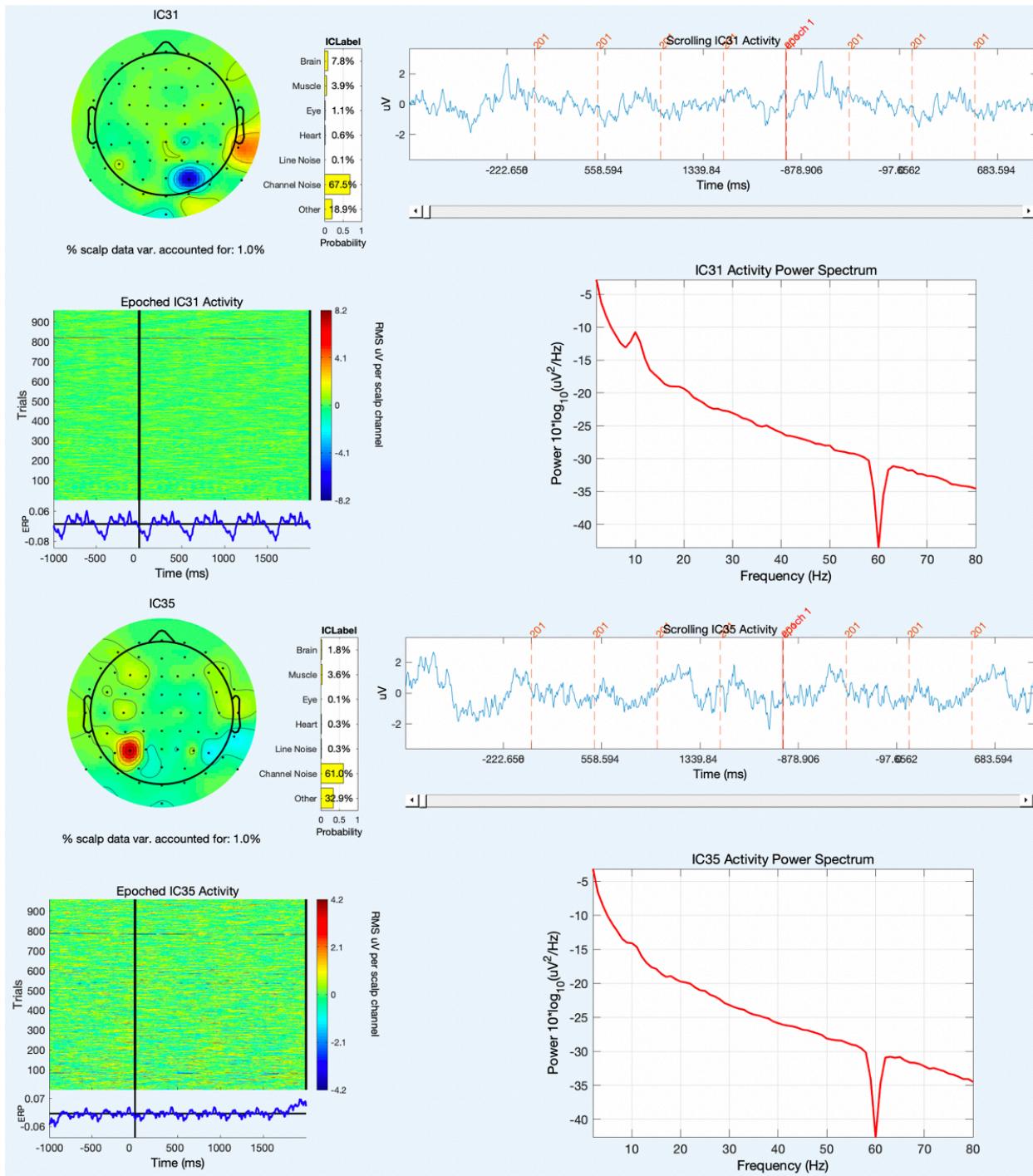


Figure 15: Channel Noise Sources - Part 2

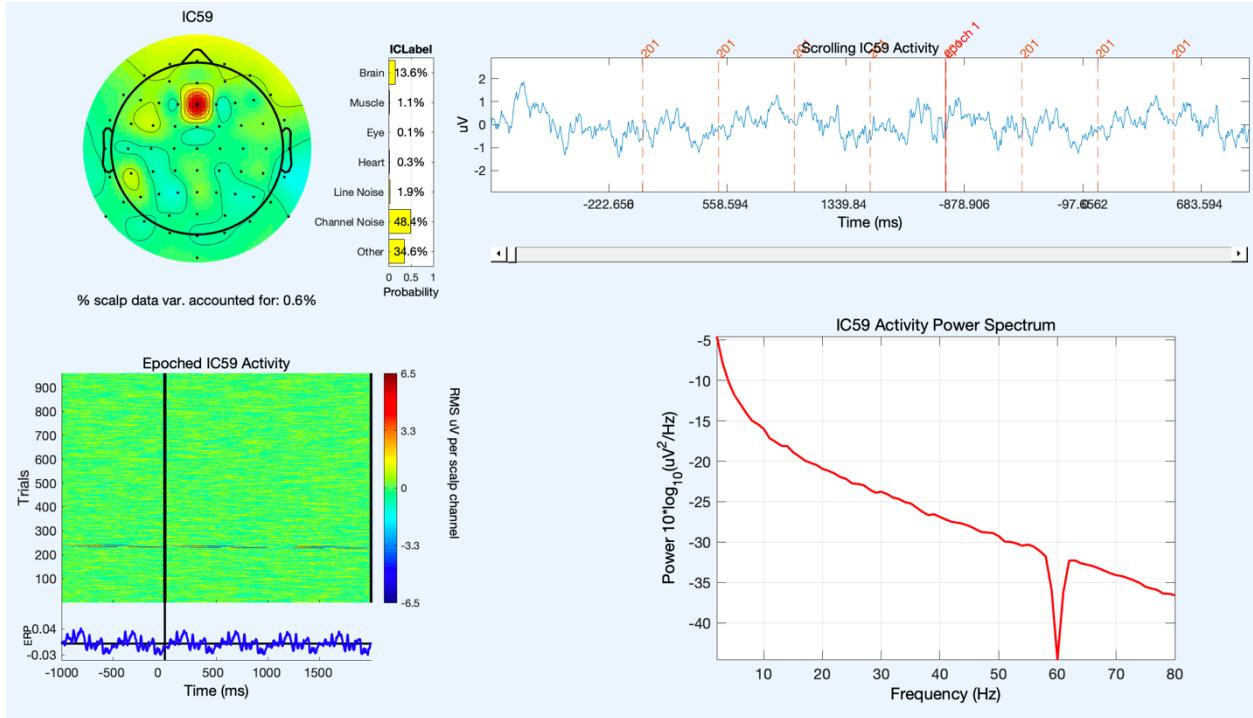


Figure 16: Channel Noise Sources - Part 3

Figures 14, 15 and 16 show details about components 12, 15, 31, 35 and 59. These sources were tagged by ICA algorithm as channel noise.

Based on Figure 14, we can notice the erratic behavior of components 12 and 15 in the voltage-time plot for a single epoch. The time-plots of these two components do not correspond to typical brain data, it corresponds to continuous noise. Also, based on the Activity Power spectrum of these two components, it is evident that they don't have any major activity at any of the known brain wave frequency ranges. As these two components are not showing any brain activity and ICA is over 95% confident that these sources are merely noise associated with channels, I've decided to remove these two components from my dataset.

Based on Figure 15 and 16, we can note the erratic behaviors of components 31, 35 and 59 in the voltage-time plot for a single epoch. The time-plots of these components do not correspond to typical brain data either. However, based on the Activity Power spectrum of components 31 and 35, it can be seen that these two components do have a bit of activity around the 10Hz frequency which is associated with resting state - more in Component 31 and very low in Component 59. Also, for component 59, ICA is more certain that it contains brain data as compared to Components 31 and 25. Therefore, there is no clear-cut conclusion on whether these 3 components can be classified as channel noise. Therefore, to be on the safe side and not remove actual brain data, I decided not to remove sources 31, 35 and 59.

All in all, I only removed components that didn't look like typical brain data and looked more like an artifact. Therefore, I only removed components 1,3, 12 and 15 from my channel data.

Cleaned EEG Data

Figure 17 is the channel data after removing the components 1, 3 , 12 and 15 from my channel data.

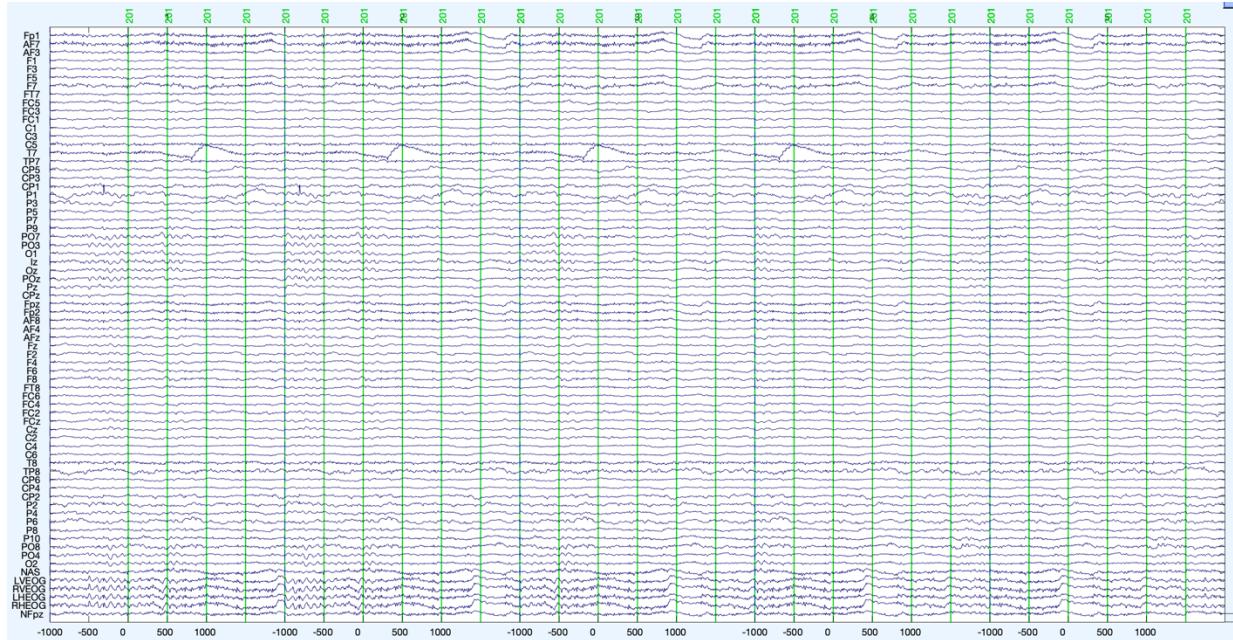


Figure 17: Cleaned Channel Data.

Feature Extraction

Now that we have removed all the suspicious noise in our dataset, we can analyze our cleaned data. I've decided to analyze my dataset using Power Spectral Density plots, Time-Frequency analysis and exploring functional connectivity between electrodes to justify my conclusions.

Time-Frequency Analysis

In signal-processing, time-frequency analysis is a technique to analyze signals whose statistics vary in time [11]. Time-frequency analysis allow visualization of signals in time-domain and frequency domain. The basic idea is to divide your time-domain signal into time-windows and compute the Fourier transform of each window, followed by plotting the power spectrum for each time window [10]. Doing so, allows you to identify the different frequencies that exist in your signal with respect to time.

Resolution is a key parameter in time-frequency analysis. It is a measure of the smallest feature that can be distinguished in the respective domain. In time-frequency analysis there is time-resolution and frequency-resolution [10].

Time-resolution corresponds to divisions of your time-domain. The frequency-resolution corresponds to the frequency represented by each data point for a given time-resolution. By setting either of the two, you will be constraint in the other (i.e. they're dependent on each other) [10].

Two of the most common time-frequency representations of signals are the Short-Time Fourier Transform(STFT) and Wavelet Transform [11]. The key difference between the two techniques is that STFT has a constant time and frequency resolution, i.e. time-resolution and frequency-resolutions are constant throughout the time-frequency analysis. Whereas in Wavelet Transform, the time and frequency resolutions are adapted for different frequencies [12]. Figure 18 illustrates this key difference.

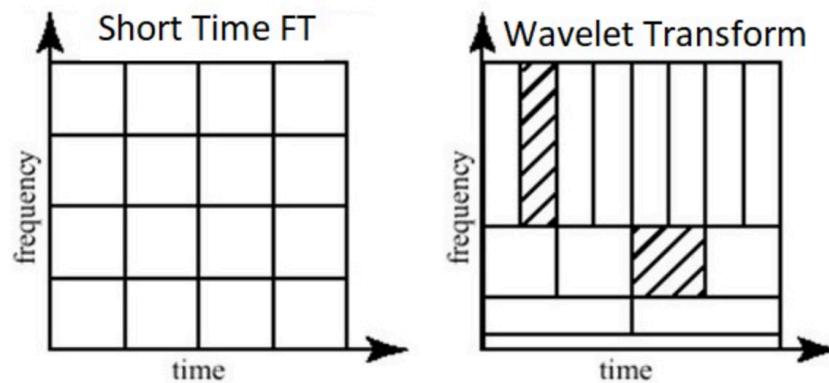


Figure 18: STFT and Wavelet Transform Resolutions [12].

STFT

Equation 10 is the mathematical representation of the STFT, where $x(t)$ is the time-plot of the signal and $\omega(t)$ is the time resolution, denoted as window in Figure 19 [13].

$$STFT_x(t', f) = \int [x(t) \cdot \omega(t - t')] \cdot e^{-j2\pi ft} dt$$

$\omega(t) = \text{Window Function}$

Equation 10

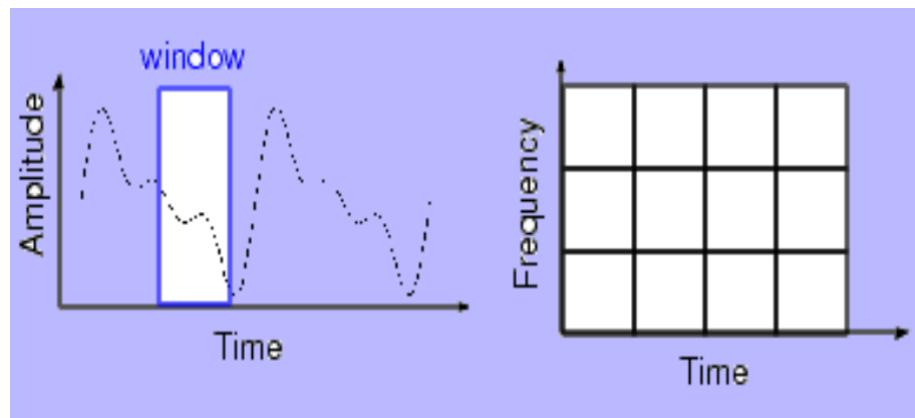


Figure 19: STFT Output [13].

STFT Window Impact

In STFT, with a short-time window, you have a higher time-resolution, but the frequency resolution (of the Fourier Transform of the signal) in each window will drop, this setup allows you to visualize where in time does a frequency occur in your signal, but it limits you from knowing the exact value of the frequency at that time [12].

Similarly, by increasing your time-window, you have a higher frequency-resolution but the time resolution in each window will drop, this setup allows you to visualize the exact frequencies that exist in your time-window, but it limits you from knowing the exact time-location of those frequency occurrences in your signal [12]. Hence, there is a trade-off between time-resolution and frequency resolution when carrying out time-frequency analysis using STFT.

Wavelet Transform

Wavelet Transform is an alternative approach to STFT [11]. It is employed to overcome the time and frequency-resolution dependency problem. Essentially, with Wavelet transform, a good time-resolution is used for signals that are more dynamic and good frequency-resolution is used for less dynamic signals. For example, considering EEG data as an example:

Gamma brain waves are more dynamic in nature as compared to Delta, Alpha, Theta and Beta brain waves because they have frequencies of over 30Hz. Due to their dynamic nature, we don't care if the frequency is 35Hz or 36Hz, we care about identifying the exact time-location of when they exist in the respective time-window. Therefore, to identify Gamma waves in a signal, good time-resolution is preferred over frequency resolution.

Similarly, Delta, Theta, Alpha and Beta brain waves have frequencies that are very close to each other, we want to be able to distinguish each one from the other. Therefore, to identify existence of Delta, Theta, Alpha and Beta bands in our signal good frequency resolution is preferred over time resolution.

A Continuous Wavelet Transform (CWT) is defined as shown by Equation 11 [13].

$$\begin{aligned} CWT_x(\tau, s) &= \psi_x(\tau, s) = \frac{1}{\sqrt{|s|}} \int [x(t) \cdot \psi \cdot \frac{(t - \tau)}{s}] dt \\ \psi \cdot \frac{(t - \tau)}{s} &= \text{Mother Wavelet} \\ s &= \text{scale}, \tau = \text{Translation}(Location of window) \end{aligned} \tag{Equation 11}$$

A wavelet is a small wave that represents the time-window and its localized in time, i.e. the window function has a finite length[13]. A Wavelet Transform splits the input signal into a bunch of signals by employing different wavelets that are generated using a prototype wavelet called the Mother Wavelet and is defined as shown in Equation 11 [12].

Each wavelet is a ‘dilated or compressed and shifted version’ of the Mother wavelet [13]. Each of these wavelets extract time and frequency information from the input signal via convolution with the signal. After each convolution, the wavelet is either dilated or compressed by the scaling factor ‘ s ’. The scaling of the wavelet affects the time-resolution, and this in-turn affects the frequency resolution. If $s>1$, the Mother Wavelet gets dilated and if $s<1$, the mother wavelet get compressed [13].

Flowchart of CWT

Figure 20 is a basic flow chart to illustrate how Wavelet Transform performs the Time-Frequency Analysis for a given Mother Wavelet.

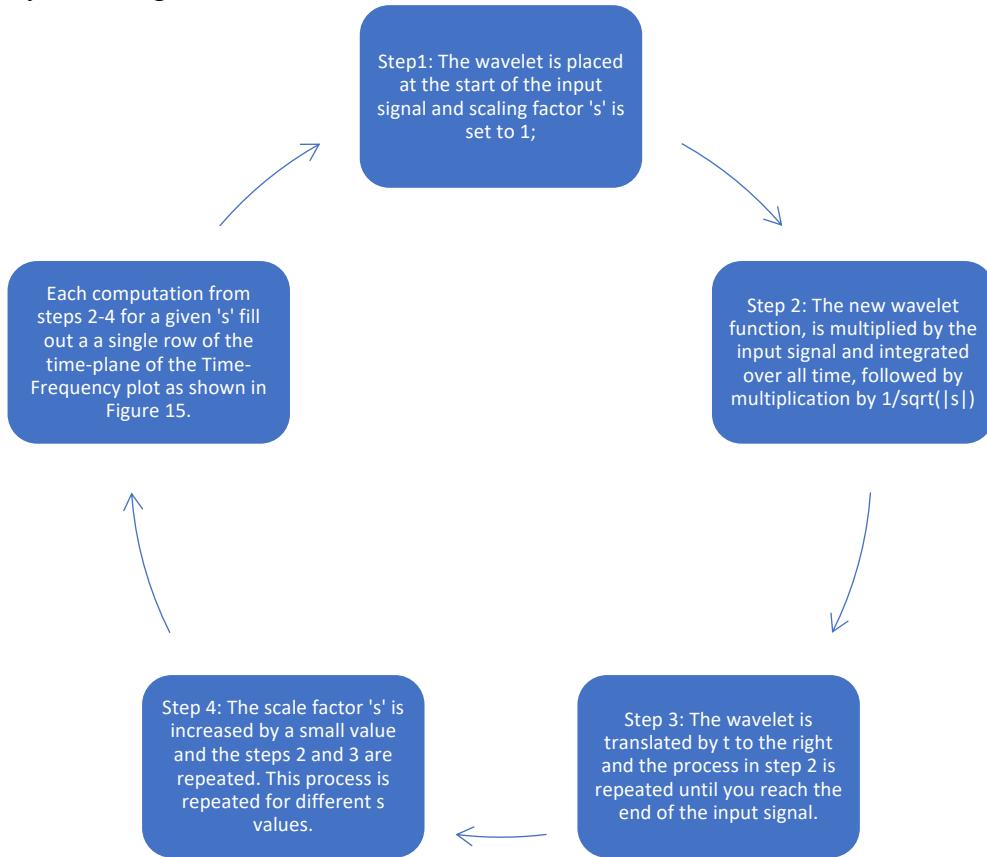


Figure 20: Wavelet Transform Flow Chart [13]/[12].

The CWT explained in this section can be implemented for Discrete time signals, the procedure is similar.

Mother Wavelet Size Impact

By stretching/dilating the Mother Wavelet in time-domain, you’re lowering the time-resolution. The convolution of this stretched wavelet with the input will allow analysis of smaller frequencies, i.e. you achieve a higher resolution in the frequency domain [13].

Similarly, by narrowing/compressing the Mother Wavelet in the time-domain, you’re improving the time-resolution. The convolution of this compressed wavelet with the input will allow

extracting finer details in the time-domain, i.e. you lose frequency resolution in the frequency domain [13].

Therefore, by changing the size of the Mother Wavelet after each convolution, the Wavelet Transform is able to adjust its time-window according to frequency bands, i.e. use a shorter-time window for higher frequencies, achieving good time-resolution, and use a longer-time window for lower frequencies, achieving good frequency resolution.

Time-Frequency Analysis with EEGLAB

Recalling that my channel data is from an individual in wakeful resting state with their eyes open and close. To confirm this, my goal is to extract information from time-frequency decomposition of the electrodes placed in the occipital region of the brain. Figure 21 is the updated channel location map of my dataset.

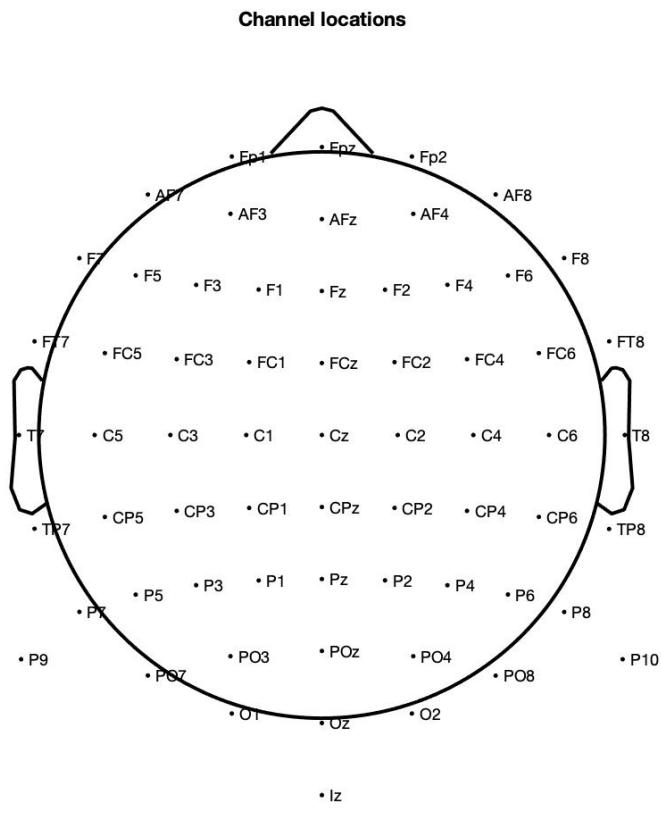


Figure 21: Channel Locations by name after removing M1 and M2.

I'm interested in extracting information about the resting state. Alpha bands are associated with relaxed brain state and they predominantly exist in the posterior of the brain [2]. Therefore, to analyze my subject's resting state, I decided to look at the electrodes connected to Occipital region of the brain.

The electrodes connected to the occipital region of the subject's brain are electrodes O2, Oz, O1, PO3, POz and PO4. Their corresponding channel numbers are 64, 29, 27, 26, 30 and 63 based on the updated channel location number map shown in Figure 22.

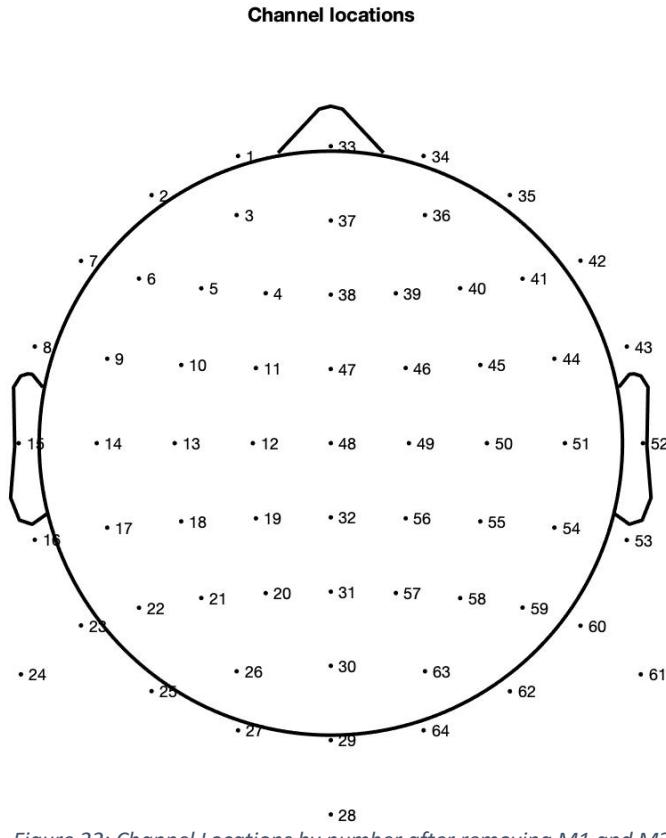


Figure 22: Channel Locations by number after removing M1 and M2.

To extract information about the resting state of the subject, I decided to carry out time-frequency analysis using wavelet transform for the channels 64, 29, 27, 26, 30 and 63 corresponding to O2, Oz, O1, PO3, POz and PO4.

The time-frequency decomposition of 64, 26, 30 and 63 channels didn't give me interpretable information, so in my report I've only included time-frequency decomposition of channels 29 & 27 as I found literature that I used to justify their time-frequency decomposition. Also, the reason why I chose wavelet transform is because 29 & 27 are connected to the brain region where you predominantly find alpha waves. Alpha waves are not dynamic in nature, so I want a good-frequency resolution to be able to distinguish alpha, beta and theta waves from each other.

In the following text I provide details on how I performed the Time-Frequency Analysis with wavelet transform using eeglab.

EEGLAB Time-Frequency Analysis Using Wavelet Transform

In eeglab, we don't have to worry about the steps shown in Figure 20, we just have to set the parameters in the eeglab gui and it carries out the steps in the flowchart shown in Figure 20. Figure 23 is the gui provided by eeglab to perform configure time-frequency parameters for a channel.

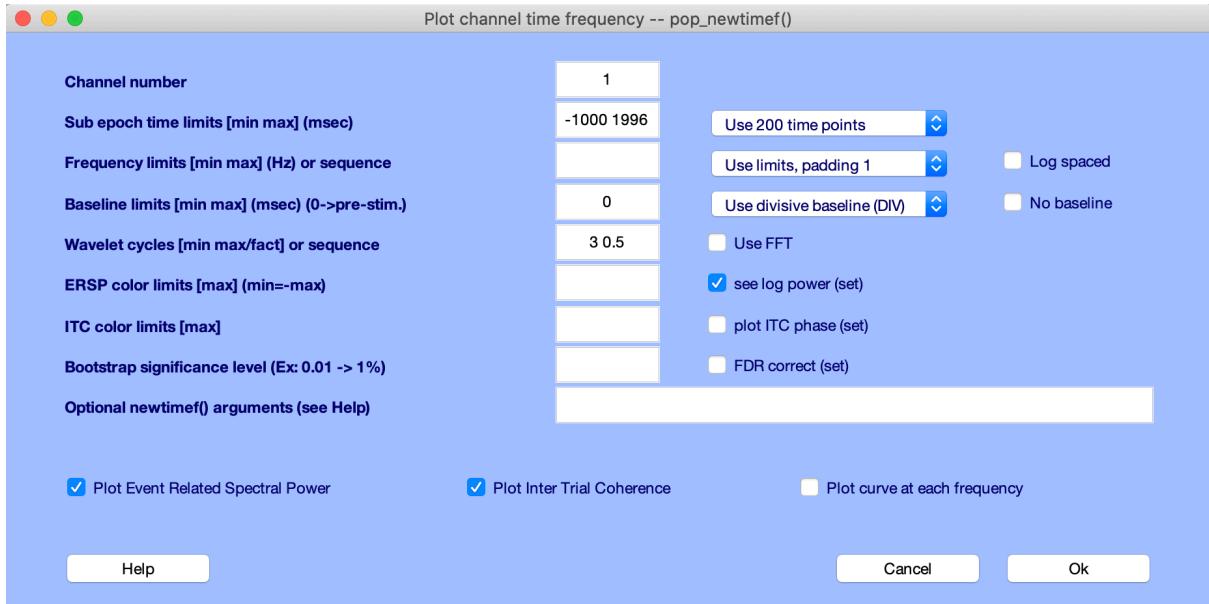


Figure 23: Eeglab time-frequency parameters.

Important Parameters

- Frequency limits: the upper and lower limit of frequencies you want to decompose your input signal to.
 - I set this to default, for a 3 Cycle wavelet, the lowest frequency that can be analyzed for a default 0.5 second window is $3/0.5 = 6\text{Hz}$ [14].
- Baseline Limits: Not doing any baseline correction.
 - I computed raw time-frequency decomposition for my channel, so I selected No-baseline box [14].
- Wavelet Cycle: The width of the wavelet used for each frequency F in the specified frequency limits
 - The width of the wavelet determines the spectral bandwidth and Wavelet Duration at a given Frequency F .
 - $\text{Spectral Bandwidth [Hz]} = (\frac{F}{\text{Wavelet Width}}) * 2$
 - $\text{Wavelet Duration [s]} = (\text{Wavelet Width}) / (\frac{F}{\pi})$
 - Making the width smaller will increase temporal resolution at the expense of frequency resolution for each F within the given frequency limits and vice versa.
 - By default for the lowest frequency eeglab uses a 3-cycle wavelet [14].
 - The second parameter indicates how the wavelet cycle/width changes as you go to higher frequencies [14].
 - If set to '1' -> same number of wavelet cycles used for each frequency as you go higher on the frequency axis.
 - If set to '0' -> Use same window size for all frequencies.
 - If this value lies within (0,1), the wavelet cycles increase linearly with frequency.
 - 'Use FFT' box is checked if you want to decompose the channel using STFT, i.e. fixed time and frequency resolution.

For my time-frequency decomposition, I used the default 3 wavelet cycle at the lowest frequency of 6Hz and the wavelet cycle increased linearly for higher frequencies by a factor of 0.5. The bootstrap significance parameter is typically set at 0.01, which masks regions of the time-frequency plot that are not showing significant power activity, this will smoothen the time-frequency decomposition [14].

Following are the time-frequency decompositions of channels Oz & O1 connected to the occipital region of the brain.

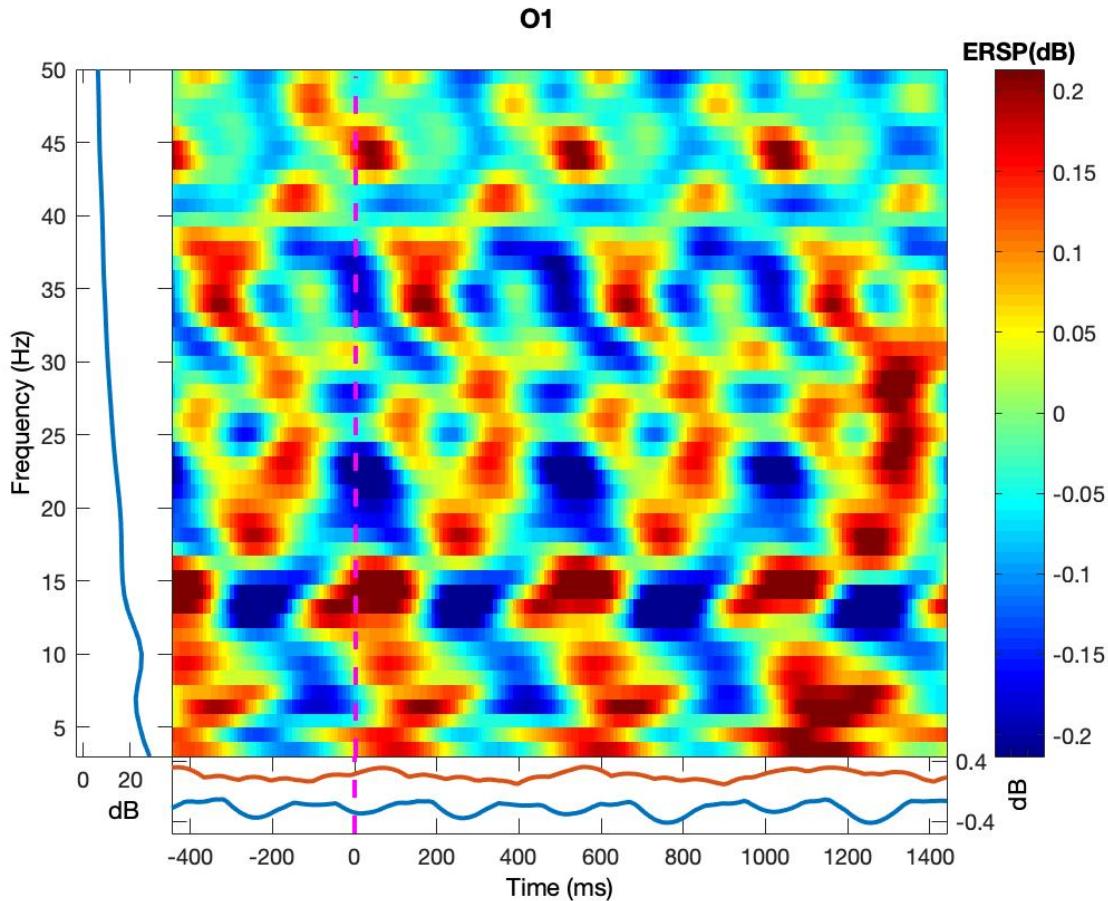


Figure 24: Time-frequency Decomposition of channel O1.

Note: Since the dataset doesn't mentioned if the recordings for 'wakeful resting stat with eyes open and close' were made in complete darkness or in light, I'm assuming complete darkness. Research has shown that the EEG activity completely differ in both conditions [15].

Based on Figure 24 and the assumption that the recordings were carried out in a dark room, we can notice the decrease in power in the beta bands after the event marker and an increase in power in the gamma band in occipital region. This power transition corresponds to transitioning from Resting Eyes Closed to Resting Eyes Open [15]. At time=610ms you can notice the significant decrease in beta band and an increase in gamma band in the occipital region. Therefore, knowing that the dataset used in this report was recorded from an individual in a resting wakeful state with their eyes open and close, the time-frequency decomposition of their

cleaned brain data confirms this, based on the assumption that the subject was in a complete dark room.

To further confirm this behavior, we look at the other channel located in the occipital region of the brain, which is in close proximity of channel O1.

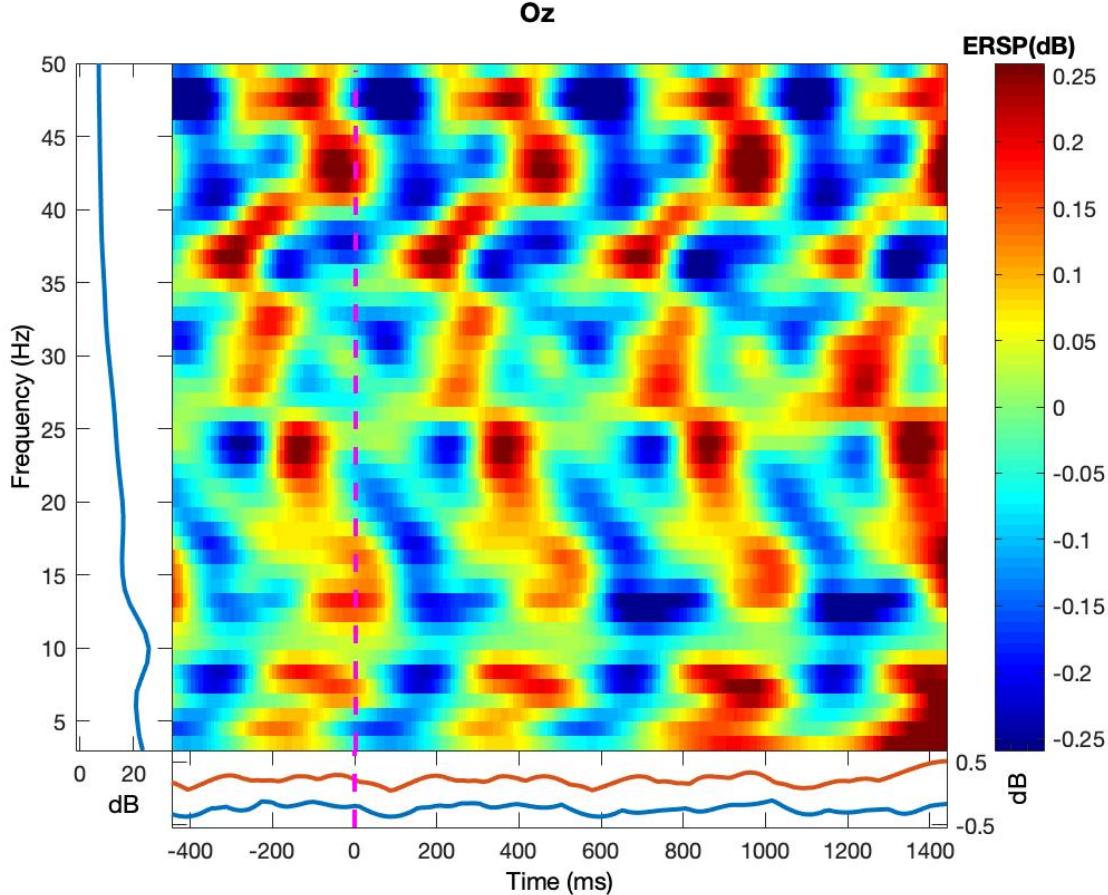


Figure 25: Time-Frequency Decomposition of Channel Oz.

Channel Oz is beside channel O1 and we expect the two channels to pick up the same information. Based on the assumption that the EEG recordings were made in a complete dark room, note the high-power activity in beta band at 400ms and then notice the significant increase in power in the gamma band at 420ms and decrease in power in the Beta band. This power transition corresponds to transitioning from Resting Eyes Closed to Resting Eyes Open [15].

We can also look at the synchronization of O1 and Oz to further confirm that the two channels are capturing the same brain activity as they're within close proximity of each other.

Coherence & Synchrony

The idea is to computing a phase vector that corresponds to the phase difference at different time points of two signals, followed by summing up all the phase vectors and normalizing their final value between 0 and 1 [16].

The closer the resultant phase vector is to 1 the greater phase synchronization between the two signal and this results in greater connectivity between the two signals . Similarly, the closer the resultant phase vector is to zero the lower the phase synchronization between the two signals which results in lower or almost no connectivity between the two signals [10].

This idea of phase synchronization and connectivity can be extended over different frequencies in a signal with the help of convolution of input signals with wavelets of different frequencies(i.e. Time-frequency decomposition). By doing so, it would allow determining if two independent signals have good phase synchronization or bad synchronization at different frequencies [10].

Further this idea of phase synchronization of signals at different frequencies can be extended to determine connectivity between two channels that are capturing brain activity. By showing a good phase synchronization between channels that are within close proximity to each we can show functional connectivity between captured eeg data to justify our conclusions about the subject.

The following pop-window is used to plot the inter-site coherence between two channels. I chose Channel 29 (Oz) and channel 27 (O1) to show their coherence around 10 Hz frequency which corresponds to alpha waves that are typically seen during resting state.

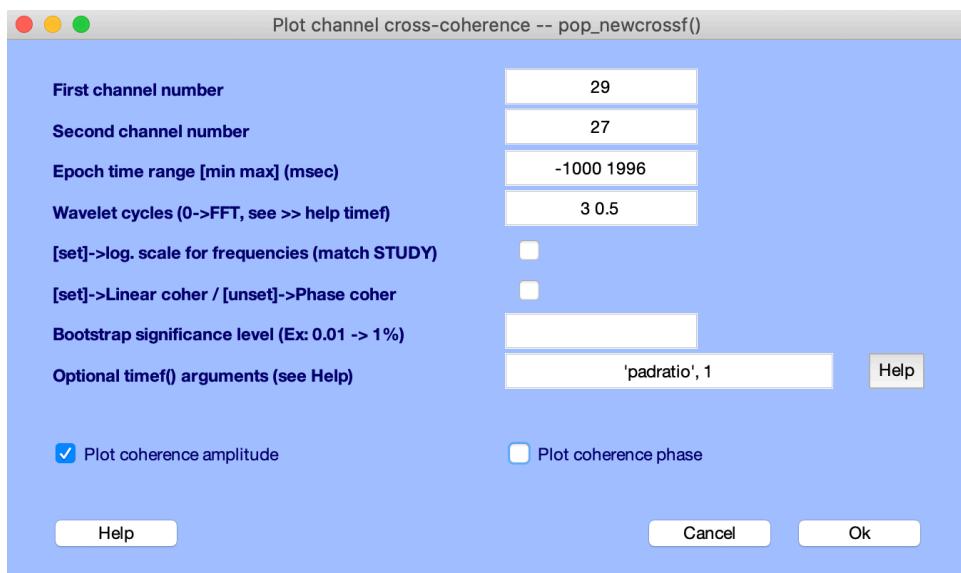


Figure 26: Cross-Channel Coherence Parameters

Recalling from time-frequency decomposition parameter section that the ‘Wavelet cycles’ parameter corresponds to the width of your wavelet and smaller the wavelet cycle is the greater the time resolution of your time-frequency decomposition. In my dataset I’m interested in the occipital region of the brain as it is known to be associated with resting state. To be able to distinguish between alpha, beta and theta waves, we would require a good frequency resolution, which is why I set 3 cycle wavelength at low frequency and increase the wavelet cycles linearly by a factor of 0.5 as we go higher in the frequency axis.

Figure 27 is the resulting inter-site coherence amplitude plot when comparing the phase at different time points for all frequencies of Oz and O1 channel.

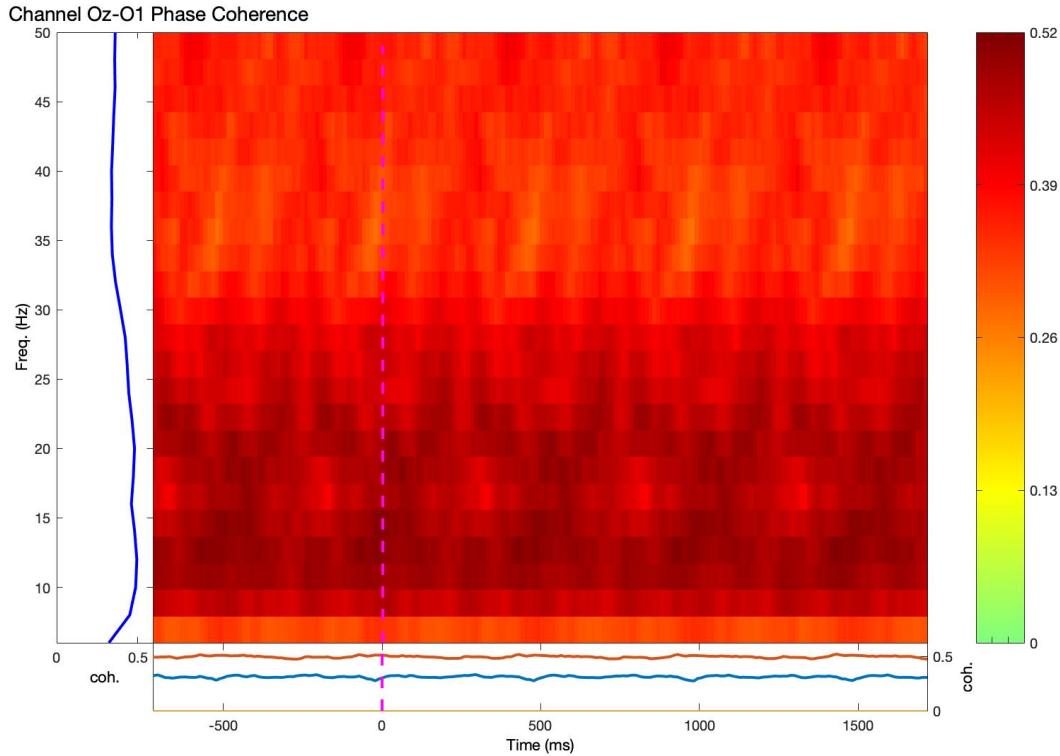


Figure 27: Phase Synchronization between Channels Oz and O1.

Based on the channel cross-coherence plot above, we can notice that channels O1 and Oz have good phase synchronization around the alpha and beta bands. By showing a strong coherency between the channels that are within close proximity, we can conclude functional connectivity between the two channels Oz and O1. They're both capturing the same brain activity in response to the event of eyes opening and closing.

We can also confirm that the subject is in a very relaxed state by looking at the Power Spectral Density of all the channels. Figure 28 is the power spectral density of the filtered channel data.

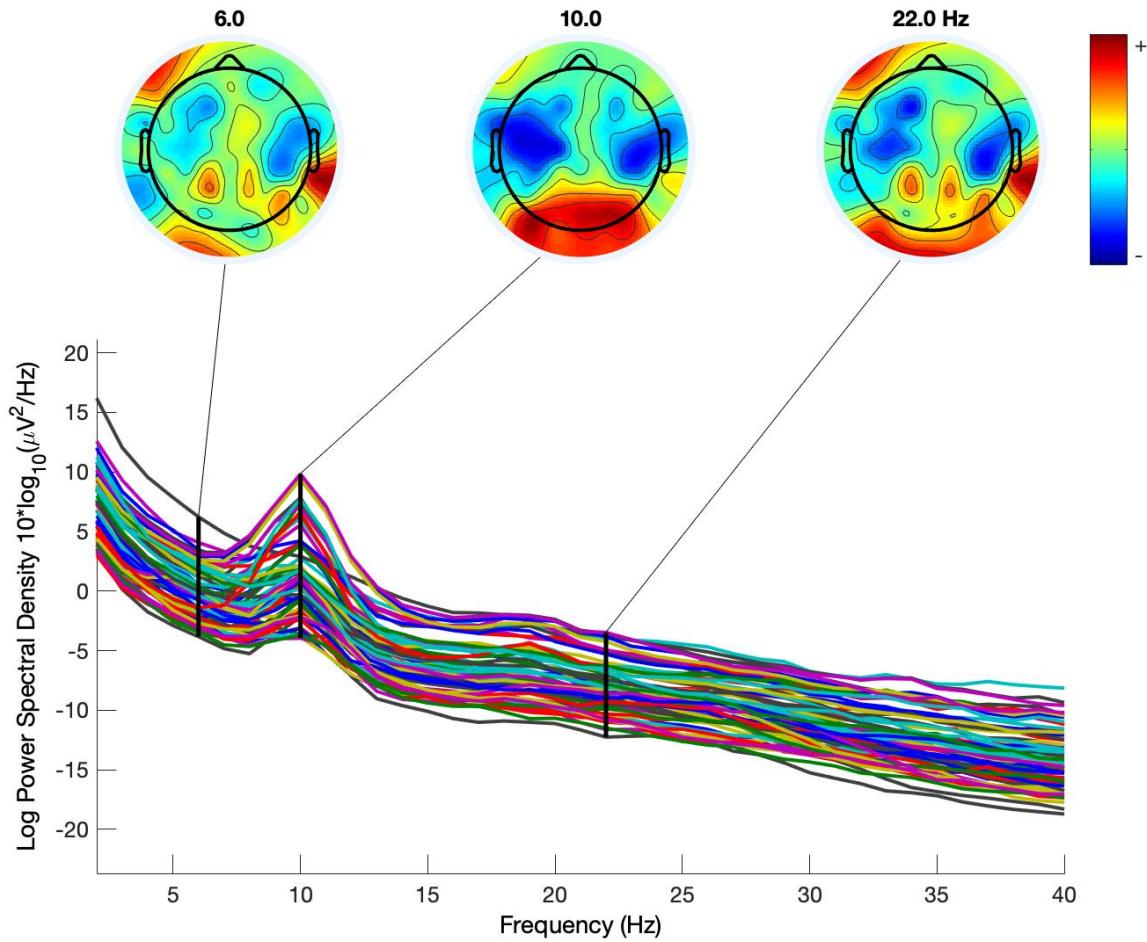


Figure 28: Power Spectral Density of Channels

Based on the plot in Figure 28, after filtering the subject's EEG data and using ICA to remove EOG and channel noise artifacts, the power spectral density revealed that the neural activity appears to be primarily in the occipital region as shown in the topographic plot in Figure 28. Also, noting that the power spectral density has the largest peak around the alpha band, which is predominantly found in the occipital region of the brain. The existence of strong alpha waves in the dataset correspond to the subject being in a very relaxed state.

Conclusion

Through EEG signal processing procedures and techniques that I learned in lecture and researching online, I was able to take raw EEG signals from an experiment through different stages of signal processing and extracting features.

The signal processing steps involved filtering the data using basic FIR filters, to removing artifacts using ICA and to finally employing Time-Frequency decomposition, Phase Synchronization and Power Spectral Density to make conclusions about the recorded EEG data.

The dataset I used for this assignment was a subject in a resting-state with their eyes open and close. I was able to show this behavior by analyzing the brain activity using time-frequency decomposition of the channels at the occipital region of the brain and also used phase synchronization to show functional connectivity between two channels that are recording the same brain activity in response to an event of eyes opening and closing.

References

- [1] Logan, T., 2020. *EEG_Cat_Study5_Resting_S8.Bdf - Trujillo (2019) Entropy Journal Article Dataverse*. [online] Dataverse.tdl.org. Available at: <<https://dataverse.tdl.org/file.xhtml?fileId=21617&version=1.1>> [Accessed 4 November 2020].
- [2]D. Farzan, "Module 2 - Technologies in Monitoring Activity of Nervous", SFU BB Collaborate Ultra, 2020.
- [3] A. Delorme, *EEGLAB preprocessing #1: Importing raw data*. 2020.
- [4] "Fourier analysis", *En.wikipedia.org*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Fourier_analysis. [Accessed: 04- Nov- 2020].
- [5] "Spectral density", *En.wikipedia.org*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Spectral_density. [Accessed: 04- Nov- 2020].
- [6]"Nyquist frequency", *En.wikipedia.org*, 2020. [Online]. Available: [https://en.wikipedia.org/wiki/Nyquist_frequency#:~:text=In%20signal%20processing%2C%20the%20Nyquist,rate%20\(samples%20per%20second\)](https://en.wikipedia.org/wiki/Nyquist_frequency#:~:text=In%20signal%20processing%2C%20the%20Nyquist,rate%20(samples%20per%20second)). [Accessed: 04- Nov- 2020].
- [7]F. Basics, "FIR Filter Basics - dspGuru", *dspGuru*, 2020. [Online]. Available: <https://dspguru.com/dsp/faqs/fir/basics/>. [Accessed: 17- Nov- 2020].
- [8]J. Anemüller, T. Sejnowski and S. Makeig, "Complex independent component analysis of frequency-domain electroencephalographic data", *Neural Networks*, vol. 16, no. 9, pp. 1311-1323, 2003. Available: [10.1016/j.neunet.2003.08.003](https://doi.org/10.1016/j.neunet.2003.08.003) [Accessed 17 November 2020].
- [9]"ICA for dummies – Arnaud Delorme", *Arnauddelorme.com*, 2020. [Online]. Available: http://arnauddelorme.com/ica_for_dummies/. [Accessed: 04- Nov- 2020].
- [10]F. Farzan, "Neuromodulation Technologies and Applications in Brain Health - Module 6", SFU BB Collaborative Ultra, 2020.
- [11]"Time–frequency analysis", *En.wikipedia.org*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Time%20frequency_analysis. [Accessed: 17- Nov- 2020].
- [12]A. Taspinar, "A guide for using the Wavelet Transform in Machine Learning", *Ahmet Taspinar*, 2020. [Online]. Available: <http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/>. [Accessed: 17- Nov- 2020].
- [13]B. D. Patil, "Introduction to Wavelet", Department of Electrical Engineering, Indian Institute of Technology, Bombay Powai, Mumbai. 400076, 2020.
- [14]"Chapter 11: Time/Frequency decomposition - SCCN", *Sccn.ucsd.edu*, 2020. [Online]. Available: https://sccn.ucsd.edu/wiki/Chapter_11:_Time/Frequency_decomposition. [Accessed: 17- Nov- 2020].
- [15]Y. Boytsova and S. Danko, "EEG differences between resting states with eyes open and closed in darkness", *Human Physiology*, vol. 36, no. 3, pp. 367-369, 2010. Available: [10.1134/s0362119710030199](https://doi.org/10.1134/s0362119710030199) [Accessed 17 November 2020].
- [16]"Phase synchronization", *En.wikipedia.org*, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Phase_synchronization#:~:text=Phase%20synchronization%20is%20the%20process,phase%20angles%20with%20each%20cycle. [Accessed: 17- Nov- 2020].