



Signals and Systems Project

INSTRUCTOR: PROF. HAMID AGHAJAN

SHARIF UNIVERSITY OF TECHNOLOGY

**Analysis of Phase Locking Value during Olfactory Stimulation
as a Biomarker for Alzheimer's Disease in EEG Signals**

Authors:

Mohammad Matin Mirzababaei

mm.mirzababaei@ee.sharif.edu

Spring 2023

Preface

Notes on the project:

- Due date: 1402/04/10
- The project must be done individually. Each individual will present his results in an online session on 1402/04/11.
- Please submit your project report as a `.pdf` file. Include all outputs and final results in the report. Make sure to list the practice text questions and provide a concise explanation of your problem-solving approach in each section.
- Ensure that all codes are provided in a separate `.m/.py/.ipynb` file. If a code cannot be tested accurately upon submission, the reported results will be considered invalid, and no points will be awarded in such cases.
- You have the flexibility to utilize either **MATLAB** or **Python** for your project. However, please be aware that **MATLAB** is recommended since certain aspects of the project rely on **MATLAB** toolboxes.
- Ensure that you save all files, including your report, codes, helper functions, and any additional outputs, if required, in a compressed file format such as `.zip` or `.rar`. This compressed file should then be uploaded to the Coursework CW submission platform.
- Your file names must be in the following format:

`Project_#StudentID.zip/.rar/.pdf/.m/.py/.ipynb`

- The details of the grading system of this project will be provided in the coming days. Generally, the project is worth a total of 1 point, with an additional 1 point allocated for the bonus section. Part 5.1 carries 0.5 points, and another 0.5 points are assigned to part 5.2.
- In this project, it is essential to uphold the principles of academic integrity and refrain from any form of cheating or copying. Cheating undermines the learning process, diminishes personal growth, and compromises the trust placed in us as students/researchers/professionals. It is crucial to recognize that engaging in dishonest practices not only tarnishes our own reputation but also has serious consequences, both ethically and academically. We want to emphasize that if anyone is found to have cheated, their results will not be accepted in this project, and they will receive a zero mark.

Contents

1	Introduction	3
1.1	Neurodegenerative Diseases	3
1.2	Olfactory Dysfunction	3
1.3	Goal of the Project	3
2	Electroencephalography (EEG)	4
2.1	What is EEG?	4
2.2	Alzheimer's Disease	6
2.3	Frequency Bands of EEG	6
2.4	Sampling frequency	8
3	EEG Signal Processing	9
3.1	Task Definition	9
3.2	Data Description	9
3.3	Pre-Processing	10
3.4	Phase Locking Value (PLV)	24
4	Results	26
4.1	Values	26
4.2	Distributions	29
4.3	Statistical Significance	32
4.4	Phase Difference	33
4.5	Heatmaps	35
5	*Bonus	39
5.1	Mild Cognitive Impairment (MCI)	39
5.1.1	Additional Information	39
5.1.2	MCI Data Processing	40
5.2	Phase-Amplitude Coupling (PAC)	44
5.2.1	Metrics	44
5.2.2	Implementation	45
6	Conclusion	46

1 Introduction

1.1 Neurodegenerative Diseases

Neurodegenerative diseases, including Alzheimer's Disease (AD) and Mild Cognitive Impairment (MCI), pose significant challenges to individuals, families, and healthcare systems worldwide. These conditions are characterized by progressive deterioration of cognitive functions, leading to severe impairment and loss of independence. Understanding the importance of these diseases and the urgency to find reliable biomarkers for their early detection and accurate diagnosis is critical for effective intervention and disease management and has become a priority in the field of neurodegenerative research.

Alzheimer's disease, the most common form of dementia, affects millions of individuals globally, and its prevalence is expected to rise with the aging population. MCI, often considered a transitional stage between normal aging and AD, is characterized by subtle cognitive decline that does not severely impact daily functioning. Scientists recognize that the combination of a person's genes and environment contributes to their risk of developing a neurodegenerative disease. For example, someone might have a gene that makes them more susceptible to Parkinson's disease, but their environmental exposures can affect whether, when, and how severely they are affected. [3]

1.2 Olfactory Dysfunction

The sense of smell is today one of the focuses of interest in aging and neurodegenerative disease research. In several neurodegenerative diseases, such as Parkinson's disease and Alzheimer's disease, the olfactory dysfunction is one of the initial symptoms appearing years before motor symptoms and cognitive decline which manifests as a decreased ability to detect, identify, or differentiate odors and thus, being considered a clinical marker of these diseases' early stages and a marker of disease progression and cognitive decline. [4]

One of the primary reasons olfactory dysfunction is prominent in neurodegenerative diseases is the presence of pathological changes in the olfactory system. In AD, for example, amyloid plaques and neurofibrillary tangles, the hallmark pathological features of the disease, are found not only in brain regions associated with memory and cognition but also in areas involved in olfaction, such as the olfactory bulb and olfactory cortex.

1.3 Goal of the Project

Understanding the significance of olfactory dysfunction in neurodegenerative diseases is important as it can serve as a potential biomarker for early detection and help unravel underlying disease mechanisms. The study of olfactory dysfunction in neurodegenerative diseases is an active area of research. Researchers are investigating the potential of olfactory testing as a diagnostic tool and exploring the mechanisms underlying olfactory dysfunction. They are also examining the role of olfactory dysfunction in disease progression and exploring therapeutic interventions targeting the olfactory system.

In this project, we want to identify early biomarkers for related brain disorders through olfactory stimulus.

2 Electroencephalography (EEG)

2.1 What is EEG?

There are different tools for collecting data from the brain. One of the methods of capturing brain signals is called Electroencephalography (EEG). These signals are changes in voltage level caused by changes in brain signals captured by some electrodes. These voltages are microVolt-level, so they can be sensitive to small noises.

One of the EEG advantages compared to other methods is its high temporal accuracy (i.e. high sampling frequency) while it suffers from low spatial accuracy. Another benefit of EEG devices is their smaller size compared to other devices like fMRI (functional Magnetic Resonance Imaging). While fMRI devices occupy the whole room, you can use EEG via portable devices.

EEG headsets are devices built to save EEG signals. These headsets could contain many electrodes. One internationally recognized electrode placement method is the **10-20 system**. This method was developed to maintain standardized testing methods ensuring that a subject's study outcomes (clinical or research) could be compiled, reproduced, and effectively analyzed and compared using scientific methods. It is called 10-20 because the distance between adjacent electrodes is 10% or 20% of the skull's total front-back or right-left distance.

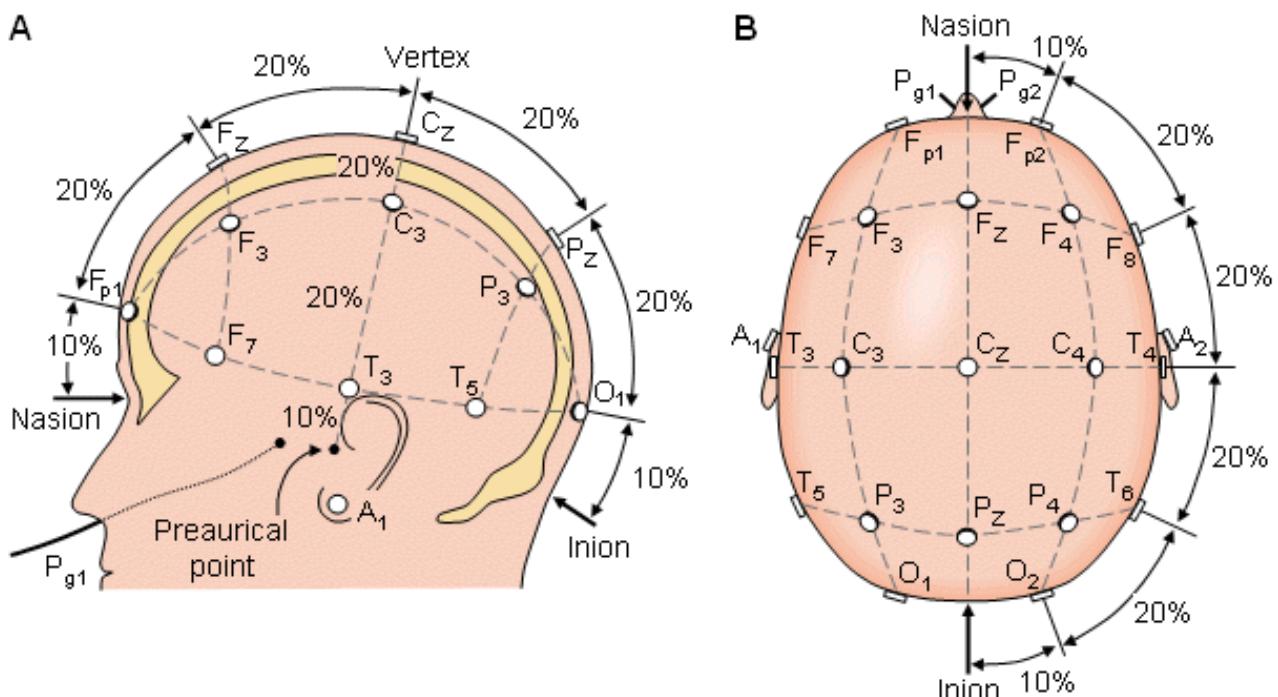


Figure 1: EEG 10-20 Electrode Placement System

Based on the picture above, What does each electrode's name stand for? Explain the naming method used in the 10-20 EEG system.

What is the 10-20 System?

The 10-20 system of the international federation is an internationally recognized method that uses anatomical landmarks to standardize the placement of electroencephalography (EEG) electrodes. The system is based on the relationship between electrode location and the underlying area of the cerebral cortex whilst ensuring that all brain regions are covered. Each electrode's name stands for the specific brain region that the electrode is placed upon. Some of them are as follows:

- Fp (pre-frontal or frontal pole)
- F (frontal)
- C (central line of the brain)
- T (temporal)
- P (parietal)
- O (occipital)

The numbers “10” and “20” refer to the distances between adjacent electrodes, which are either 10% or 20% of the total distance (front-back or right-left) of the skull. For example, the Fp1 is placed at 10% of the total distance from the nasion, and Fz is placed at 20% of the total distance from Fp1.

The number of the electrode gives information about the distance from the electrode to the midline of the brain. At the midline, the electrodes are labeled with a ‘z’ to represent zero. The electrode numbers increase as you move away from the midline. Odd numbers represent electrodes on the left hemisphere and even numbers represent electrodes on the right hemisphere.

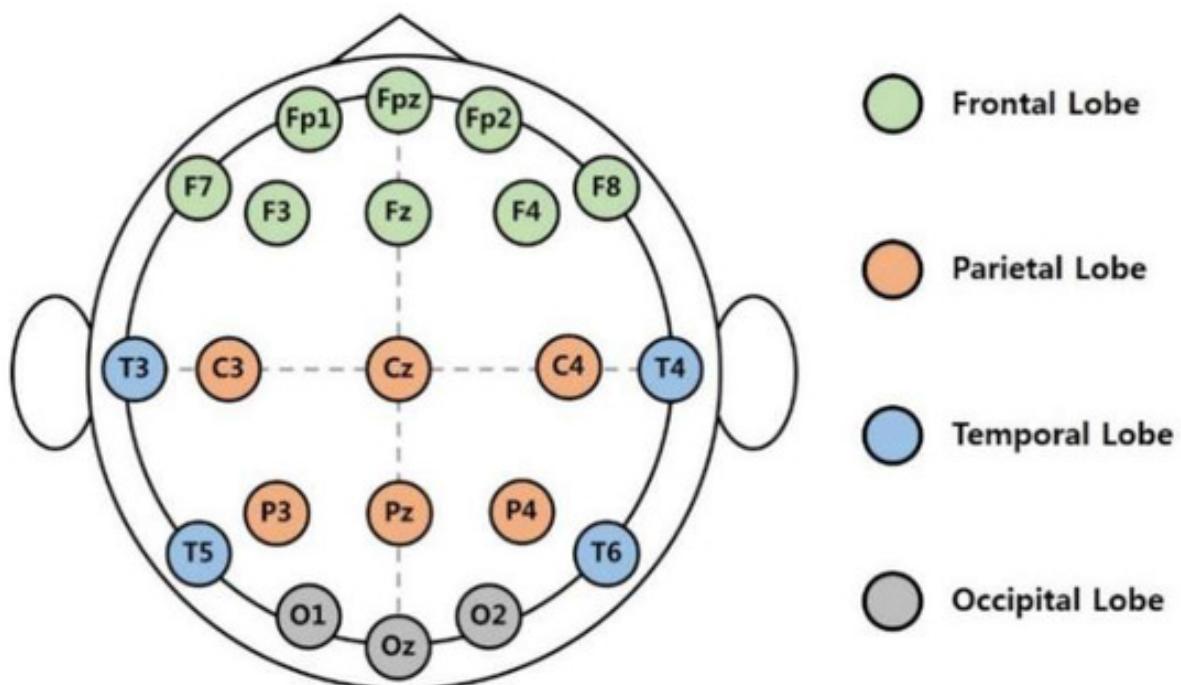


Figure 2: Electrode's name

2.2 Alzheimer's Disease

Alzheimer's Disease (AD) is a progressive and irreversible neurological disorder that affects the brain, primarily causing problems with memory, thinking, and behavior. It is the most common cause of dementia, a general term for a decline in cognitive ability severe enough to interfere with daily life.

The exact cause of Alzheimer's disease is not yet fully understood, but it is believed to involve a combination of genetic, lifestyle, and environmental factors. The staging of the AD is associated with the accumulation of Amyloid- beta ($A\beta$) proteins in the brain. These depositions cause synaptic and neuronal loss, which leads to major cognitive dysfunction in the advanced levels of the disease.

While EEG is not currently used as a primary treatment for Alzheimer's disease, it can be a valuable tool in the diagnosis and monitoring of the disease. EEG can help in the diagnosis of Alzheimer's by detecting abnormal patterns of brain activity that are characteristic of the disease. In individuals with AD, EEG often shows changes such as a reduction in certain brainwave frequencies and an increase in others. These patterns can aid in differentiating Alzheimer's from other types of dementia or cognitive disorders.

2.3 Frequency Bands of EEG

In the frequency domain, EEG signals are divided into 5 bands. [2]

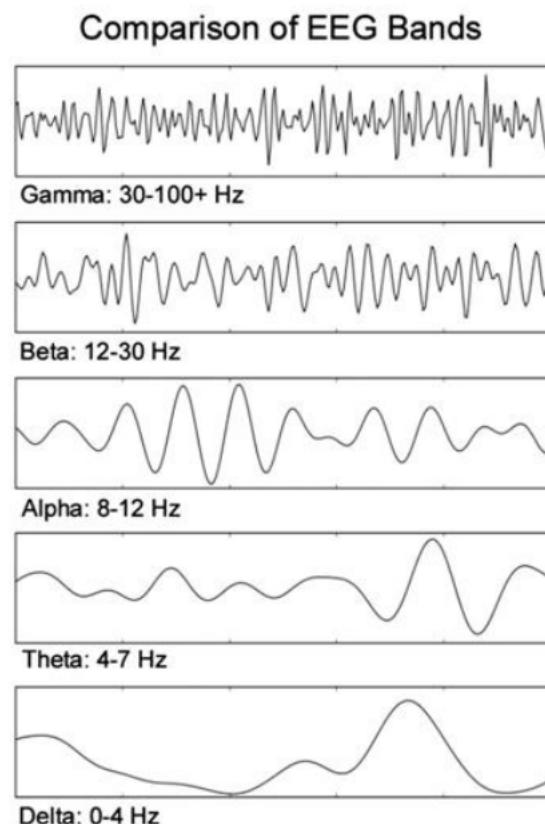


Figure 3: EEG Frequency Bands

Determine the activities each frequency band is associated with.

EEG Bands

Throughout daily tasks, brain is firing with neuronal activity. The neurons can fire randomly or in a synchronized manner. These synchronized, rhythmic electrical pulses from communicating neurons are called neural oscillations, or brain waves. For these signals to be detected from the cerebral surface, a large number of neurons must be synchronously active. When the signals' amplitude is high enough, they can be detected from the cerebral surface with electroencephalography (EEG). There are different categories of brain waves, which are characterized by their differences in frequency and amplitude. The four main brain waves that are recognized are: alpha, beta, theta, and delta waves, which are shown in Figure 4.

EEG bands	Frequency (Hz)	Distribution	State of mind
Delta	0.5–4	Central cerebrum and parietal lobes	Deep sleep, non-REM sleep
Theta	4–8	Frontal, parietal and temporal lobes	Drowsiness, first stage of sleep
Alpha	8–13	Most prominent at occipital and parietal lobe	Relaxed wakefulness with eyes closed
Mu	8–12	Central electrodes, over motor and somatosensory cortex	Shows rest state motor neurons
Beta	13–30	Frontal and central regions	Highly alert and focused
Gamma	>30	Very localized	Higher mental activity, including perception and consciousness [30]

Figure 4: different frequency bands in the brain signal treats

Characteristics of alpha waves

Alpha waves have a frequency of 8-12 Hz. They can be seen in states when a subject is awake, relaxed, and resting. These waves are seen most intensely in the occipital region (visual cortex) when a subject has their eyes closed. In addition to the occipital region, these waves can also be recorded from the parietal and frontal scalp regions.

Characteristics of beta waves

Beta waves typically occur at a frequency of 14-30 Hz. These waves are small and faster and can be seen in states of intellectual activity, focus, and alertness. Beta waves can be separated into two subgroups: beta I and beta II. They are most prominent in the parietal and frontal regions of the scalp, as this is where the abovementioned mental tasks take place. Beta waves are less easy to identify in EEG compared to the other types of brain waves, as they do not have a regular signature waveform.

Characteristics of theta waves

Theta waves have a frequency of 4-7 Hz. They can be seen in states of daydreaming or light sleep. These waves can be seen in the parietal and temporal regions. They are more

pronounced in children or adolescents and less prominent in adults. Functionally, they have been linked to coordinating the process of memory storage.

Characteristics of delta waves

Delta waves have a frequency of 0.5-3 Hz. These waves have a low frequency and relatively high amplitude. They can mainly be seen in states of deep sleep. Delta waves only occur within the cortex, independent of activities in the lower regions of the brain.¹ Delta waves during sleep are thought to play a role in transferring learning and long-term memory storage.

2.4 Sampling frequency

Based on frequency bands and Nyquist criterion, which sampling frequencies are preferred for EEG signals?

Sampling Challenge!

For EEG signals, the preferred sampling frequency depends on the frequency range of interest and the Nyquist criterion.

The Nyquist-Shannon theorem states that "if a signal containing no frequency higher than f_N is sampled at a frequency at least twice the f_N ?", the signal can be reconstructed without loss from its samples". The frequency f_N is called the Nyquist frequency. Practical rule following from the theorem can be formulated as "a signal sampled at a sampling rate f_s can be fully reconstructed from its samples without loss if the signal contains only frequencies lower than $f_N = f_s/2$ ". Otherwise, the signal cannot be precisely recovered due to the aliasing phenomenon.

EEG signals typically range from 0.5 Hz to 70 Hz, with the most clinically relevant frequency range being 0.5 Hz to 30 Hz. To satisfy the Nyquist criterion, the sampling frequency should be at least twice the highest frequency in this range, which is 60 Hz. This ensures that signals up to 30 Hz can be accurately represented without loss of information. However, higher sampling frequencies may be used in certain cases where higher frequency components are of interest, such as in sleep studies or neuroscientific research.

It's worth noting that specific experiment requirements, signal processing techniques, and the nature of the EEG application may influence the choice of sampling frequency.

Thus, to answer this question, the sampling rate of 200 Hz is too sufficient to analyse any EEG frequencies below Nyquist frequency of 100 Hz ($= 200/2$) if, before sampling, the EEGs were low-pass filtered to guarantee that the signals did not contain any frequencies higher than 100 Hz. Practically speaking, it should be checked that the EEGs were appropriately filtered before digitization, and if they were, the sampled data is certainly good to analyze the 30Hz beta range.

3 EEG Signal Processing

In this section, firstly you would get familiar with the task and the structure of the data.

3.1 Task Definition

[7] To identify the effect of olfactory dysfunction among different brain health states, the following task was performed to collect the data. The same sequence of stimuli was presented to all participants. The stimulation sequence was composed of two different odors, one occurring frequently (standard) with a probability of 0.75 and the other presented rarely (deviant) with a probability of 0.25. Each trial consisted of a 2s stimulus presentation followed by 8s of rest (pure water vapor). The odors were delivered to the participants using a laboratory olfactometer. The experiment involved 120 trials in which 90 frequent and 30 rare stimulation cycles were presented in a predetermined, randomized order. Lemon essence was used as the frequent odorant and rose essence was used as the rare odorant. These odors were selected to avoid trigeminal system activation as the olfactory and trigeminal systems are interconnected and may interact with each other during exposure to certain stimuli [8]. The duration of odor presentation was set at 2s to enable regular breathing cycles for the participants.

3.2 Data Description

[7] The dataset consists of three files as follows:

- **AD.mat**: Contains data for Alzheimer's disease patients.
- **Normal.mat**: Contains data for healthy elderly participants.
- **MCI.mat**: Contains data for mild cognitive impairment patients. (Described in part 5.1)

The structure of the files is the same. Each file is organized as a structure array, in which each row contains information of one participant and the three columns correspond to the “epoch”, “odor” and “noisy” fields as described in Table 1.

Field	Description
epoch	This is a 3D array structured as $4 \times 600 \times \text{Num_trials}$. The first dimension indicates EEG channels respectively from the first column as Fp1, Fz, Cz, and Pz. The second dimension contains EEG samples from 1 s pre stimulus to 2 s post stimulus, which at a 200 Hz sampling rate amounts to 600 samples. The last dimension shows the number of trials. This could be different for each participant as some trials were deleted during preprocessing.
odor	This is a 2D binary array shaped as $\text{Num_trial} \times 1$. This array shows the odorant type (lemon/rose) the participant was exposed to in each trial. The value = 1 indicates the rose odor and the value = 0 indicates the lemon odor.
noisy	This is a 2D array with the size $1 \times \text{Num_noisy}$. This array indicates noisy trials identified based on comparing the instantaneous and average trial amplitudes. These noisy trials can be ignored in processing and were included for the dataset completeness.

Table 1: Description of each structure array (.mat file) in the dataset.

3.3 Pre-Processing

Using a standard pipeline in EEG signal preprocessing is crucial for ensuring consistency, reproducibility, and objectivity in research. It reduces bias, enhances the reliability of results, and provides established best practices for addressing common challenges. A popular and widely used pipeline for EEG signal preprocessing is Makoto's pipeline ([Makoto's preprocessing pipeline - SCCN](#)).

The collected raw data from all participants were preprocessed following the full pipeline of Makoto with the use of EEGLAB and posted as a dataset, as described in the following steps:

1. Apply 1 Hz high pass filter to remove baseline drifts.
2. Apply relevant notch filter to remove the 50 Hz line noise.
3. Reject bad channels as a critical step before average referencing with the use of `clean_rawdata()` EEGLAB plugin.
4. Interpolate the removed channels.
5. Re-reference the data to the average of all channels to obtain a good estimate of reference-independent potentials.
6. Apply `clean_rawdata()` for cleaning the data by running artifact subspace reconstruction(ASR).
7. Re-reference the data to the average again to compensate for any potential changes in the data caused by the previous step.

8. Run independent component analysis (ICA) to identify EEG sources as well as the sources associated with noise and artifacts.
9. Fit single and bilateral (if available) current dipoles.
10. Further clean the data by source (dipole) selection using `IClabel()` plugin in EEGLAB.

In the **Dataset/Preprocess** folder you can find the raw data for 2 subjects with the corresponding additional information provided. In this section you are required to preprocess these data and save your final preprocessed cleaned data.

However, there is no need to fully implement the Makoto's pipeline and a simplified version of this is as follows; follow the instructions below and provide the required results in each step:

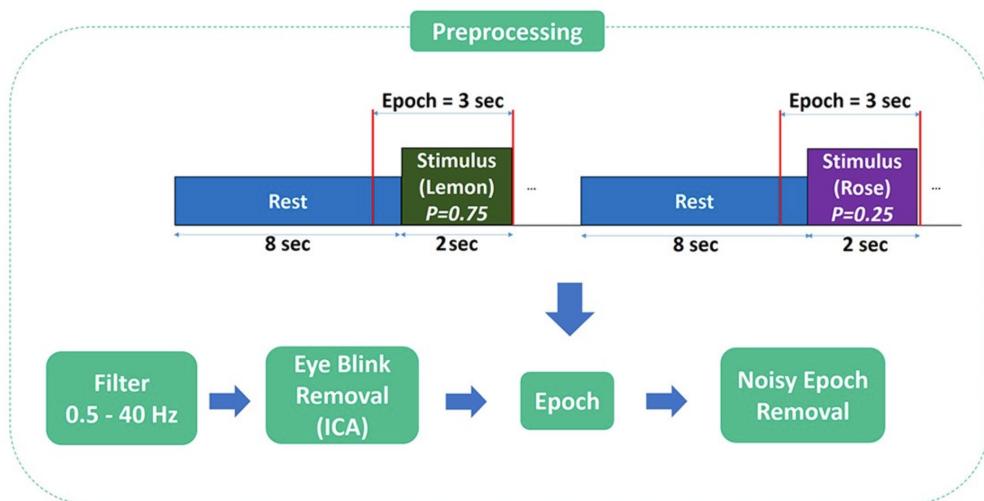


Figure 5: Task and Preprocessing Steps [5]

Note

Since the same procedure should be done on the two subjects, the whole procedure would be done on the first subject and also explained. To sum up, we will only bring the results of the second objects.

Primary Setup

It is not possible to simply import the given 'Subject.mat' into the EEGLab GUI app. It is necessary to load the named file into matlab workspace and save it as a Matlab Variable. First, we transform the data to an 2D array. Since the data is given the form of [Sample,Channel], it is our duty to transpose it and import the proper form into EEGLab([Channel,Sample]). On the other hand, the 20th channel should be omitted from the subject, because it has no new important data for us and 19 Channel Standard system of electrodes' location is required.

Function 'setupSubject' below is designed to meet this need and is used to set both of the subjects up:

```
function newSub = setupSubject(Subject)
% This function sets the subjects up in order to be imported
in EEGLAB

newSub = transpose(table2array(Subject));
% It should be transposed to reshape:
% rows -> Channels % columns -> Samples
newSub = newSub(1:19,:); % Channel-19 is neglected
end
```

Now, it is convenient to import the 'newSub' into EEGLab for further pre-processing activities. Raw data and the default settings are shown in the figure below:

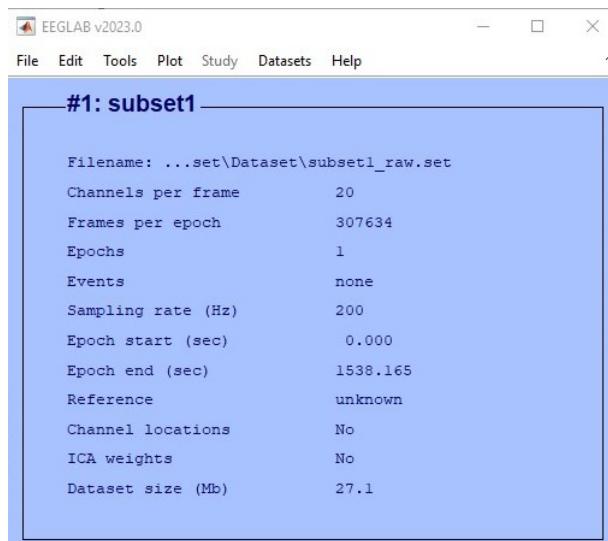


Figure 6: Subject1 - Raw EEG Data

- **Step 1:** To preprocess using EEGLAB, first re-reference data to the mean of the channels. Then use a bandpass filter to filter 0.5 - 40.5 Hz frequencies. As we have filtered to 40.5 Hz, there is no need to apply a 49.9 - 50.1 Hz notch filter to remove the line noise (However, keep this step in mind as this is a crucial step in EEG signal preprocessing!). Using FFT function or EEGLAB, plot the frequency spectrum of Fz channel data. (Just to note, your data will be saved at EEG struct in MATLAB workspace.)

Step 1

Firstly, the starting time([0 14]s) should be neglected from the data. In order to do this, we select the mentioned data and remove it from all channels. Secondly, we have to check whether bad channels are available in our data or not. This is done

using `clean_rawdata()` in EEGLab, shown below:

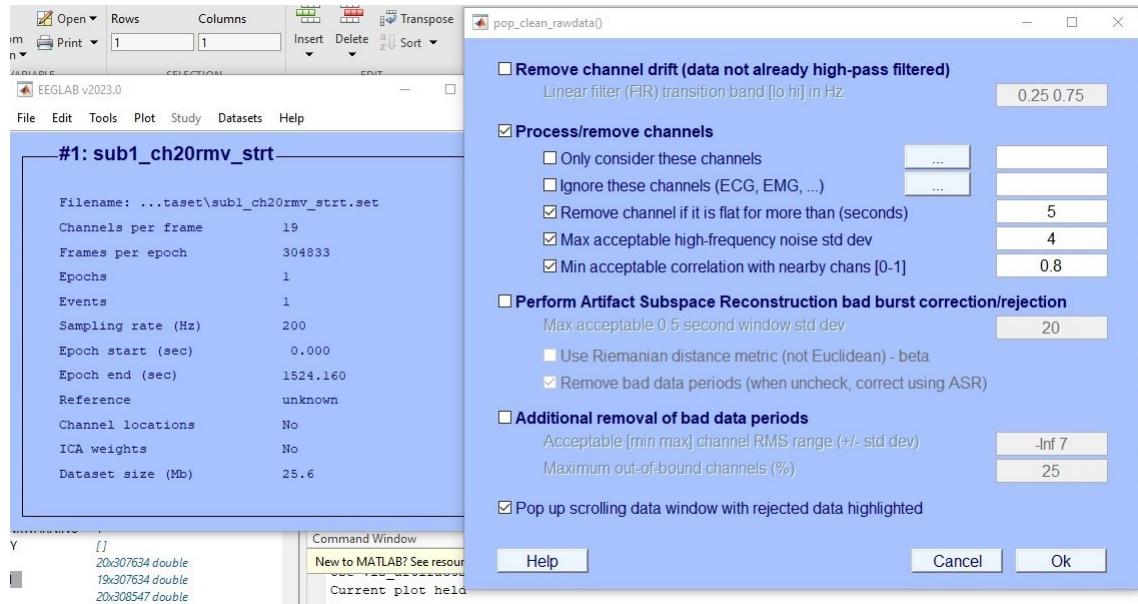


Figure 7: Checking for possible bad channels

Luckily, no bad channel existed in our data.

Thirdly, we have to re-reference the electrodes' voltages to a rational value. Here, this value would be the average of the all channels. This hugely helps us to remove the bad drifts in the data channel waves.

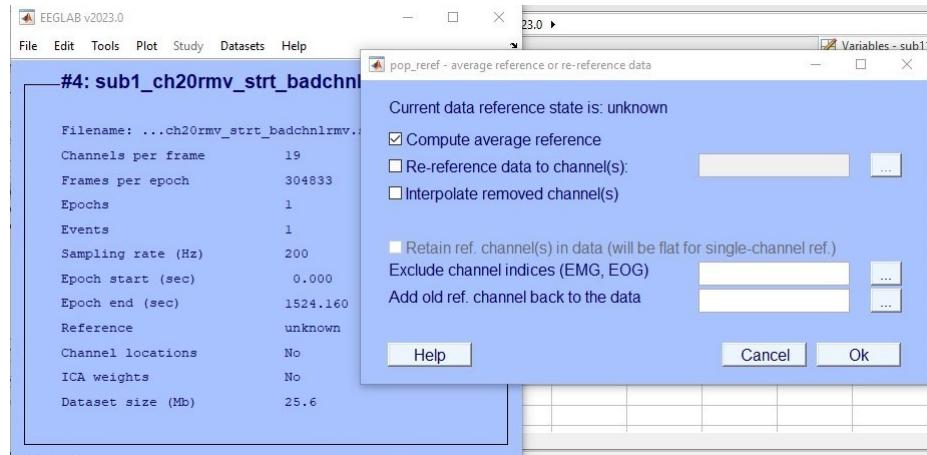


Figure 8: Re-referencing the EEG data to mean

Fourthly, the EEG data is filtered by a 0.5 to 40.5 Hz band-pass filter. Thus, there would be no need to apply the notch filter to remove power-line frequency any more.

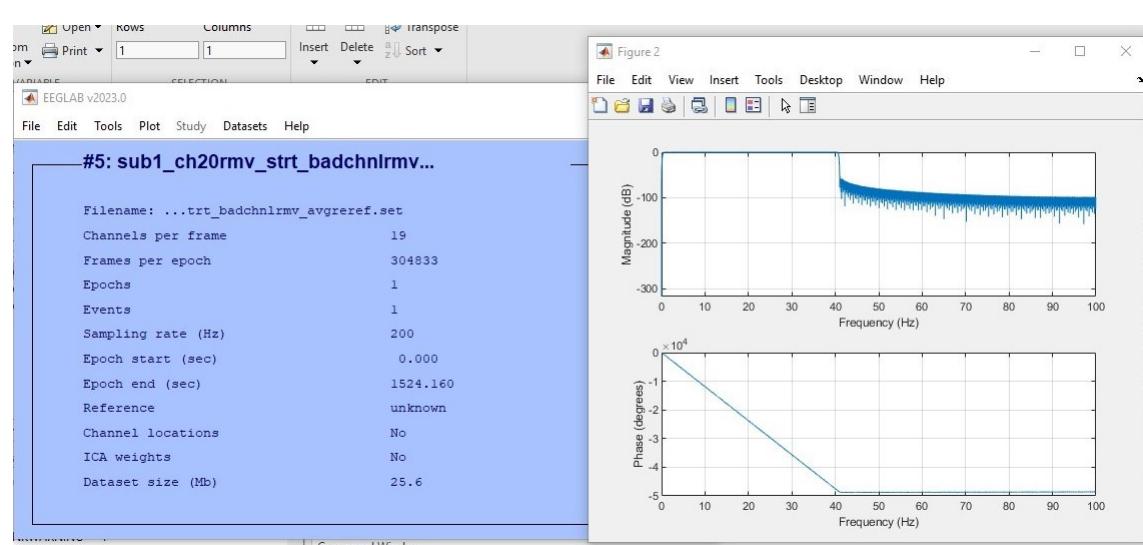


Figure 9: Band-pass filter

We plot the channels data again to see the difference,

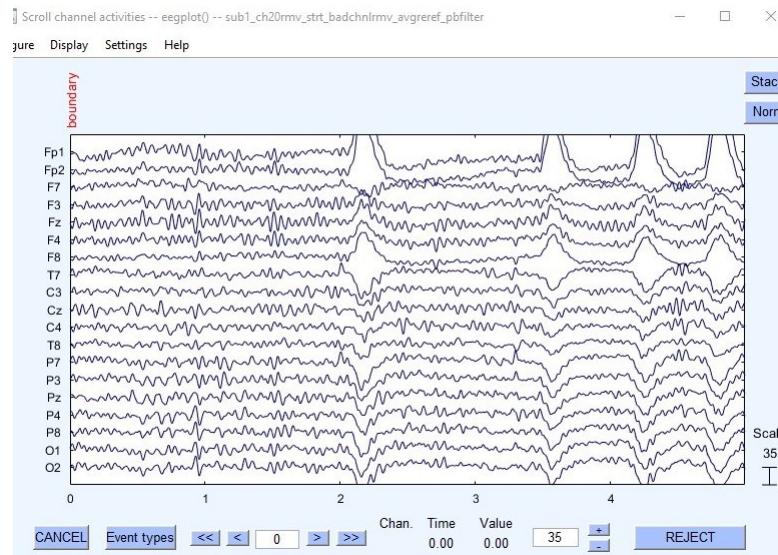


Figure 10: Subject1 - Before ICA

In this at last, we are asked to plot the frequency spectrum of Fz in both Subjects. Suitable function to plot is provided here:

```

freqPlotter(EEG.data(5,:),200,1000) % Channel Fz ->
data(5,:)

function freqPlotter( signal , fs , samples)
%fs -> Sampling frequency
Y = fft ( signal ,samples);
f = fs *(0 : samples -1) / samples;

```

```

% We use fftshift() to cover the negative frequencies
% as well
y = fftshift (Y);
figure
plot (f-fs /2, abs (y))
xlabel ('frequency (hz)');
ylabel ('Magnitude (v)');
title("Sample1 Channel Fz frequency Spectrum")
grid on
end

```

It plots the frequency spectrum of the EEG.data of the current(running) dataset in EEGLab. Results are as follows:

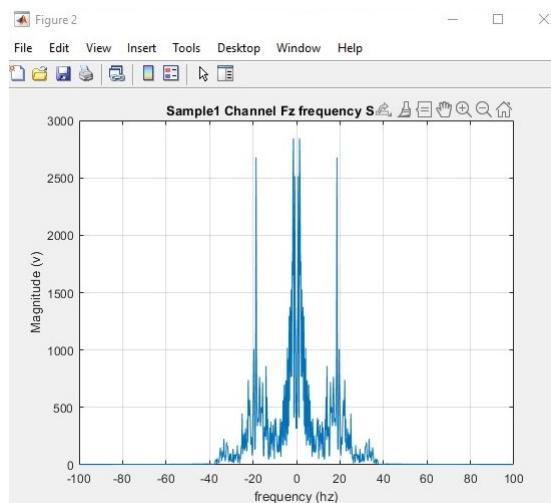


Figure 11: Subject1 - frequency spectrum of channel 'Fz'

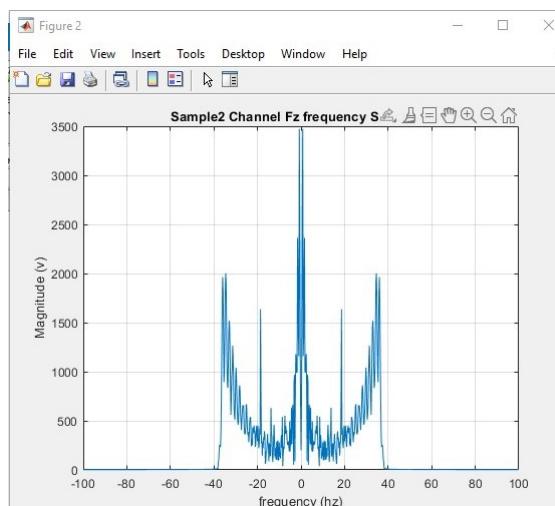


Figure 12: Subject2 - frequency spectrum of channel 'Fz'

- **Step 2:** In this part you would remove the artifacts of the signal. Artifacts include blinking, eye movement, muscle movement, heart rate and etc. For this, load your data at EEGLAB. Now load `Standard-10-20-Cap19.loc` file from `edit-channel loacations` menu that contains locations of channels. Then run ICA (Independent Component Analysis) algorithm from `tools-decompose data by ICA` menu. Please note that this part would probably takes more time. Then you will have the a figure like [Figure 4](#) by running `tools → classify components using ICLabel-label components`. By clicking on

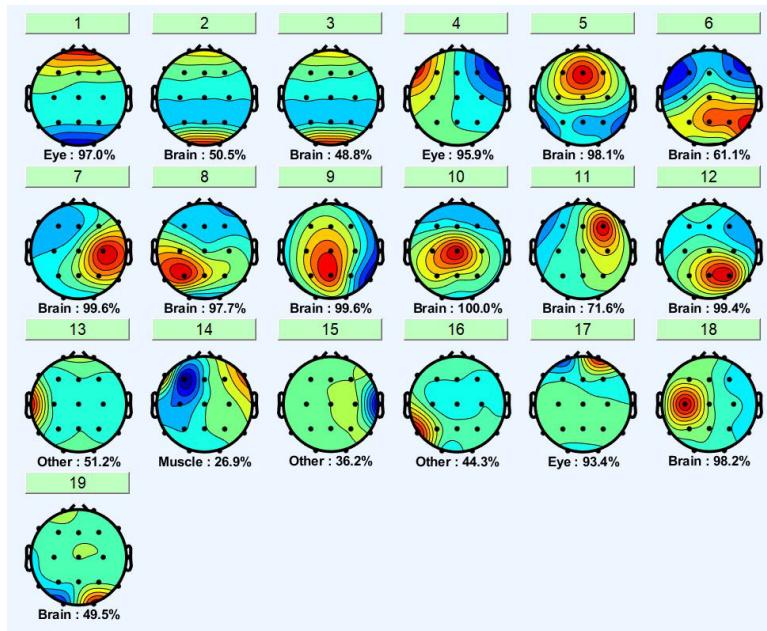


Figure 13: An example of ICA components

each component, you can some details about it as well. Present a figure from one of the brain components with its details.

Now remove all non-brain components. For this purpose, from `Tools-remove components from data` enter the number of components that must be removed.

Step 2

First, we load the Standard electrodes' location to EElab ('Standard-10-20-Cap19.loc').

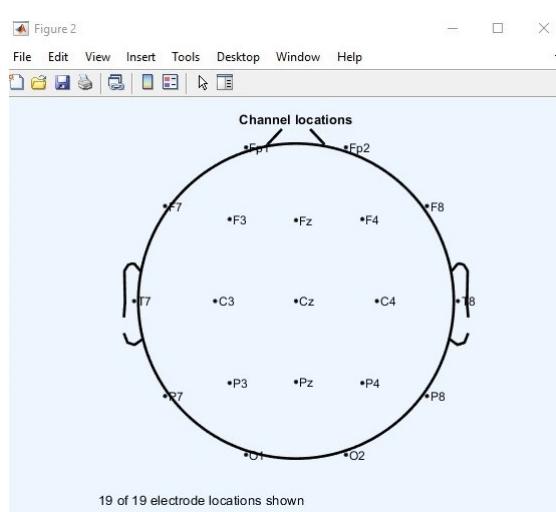


Figure 14: Location of the electrodes around brain

ICA can provide us the matrix that relates the electrodes' signal(X) and the source signal inside the brain(S) and helps to find an estimator for the source signal. This is useful to differentiate between different sources in the EEG data signal. This is done mostly in order to detect the non-brain components(such as muscle, eye-blink or others) Thus, we run the ICA by default settings of EEGlab(We take notice of the rank of the desired matrix). After waiting for a while, result of ICA is ready and we can plot and see different components:

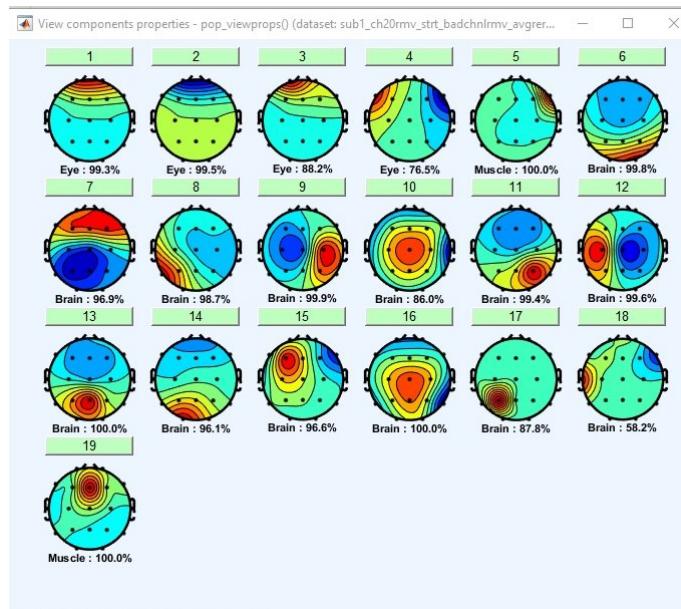


Figure 15: Subject1-Components found followed by their occurring probabilities

Non-brain components should be removed from the components to have a pure-brain wave. Figure below compares the components before and after removing non-brain components:

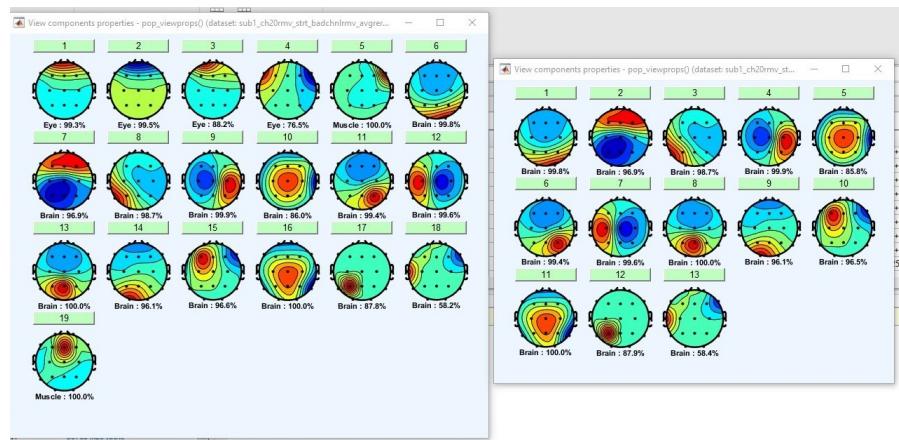


Figure 16: Subject1-Before and after removing non-brain components

Moreover, a detailed figure of a brain wave is presented here:

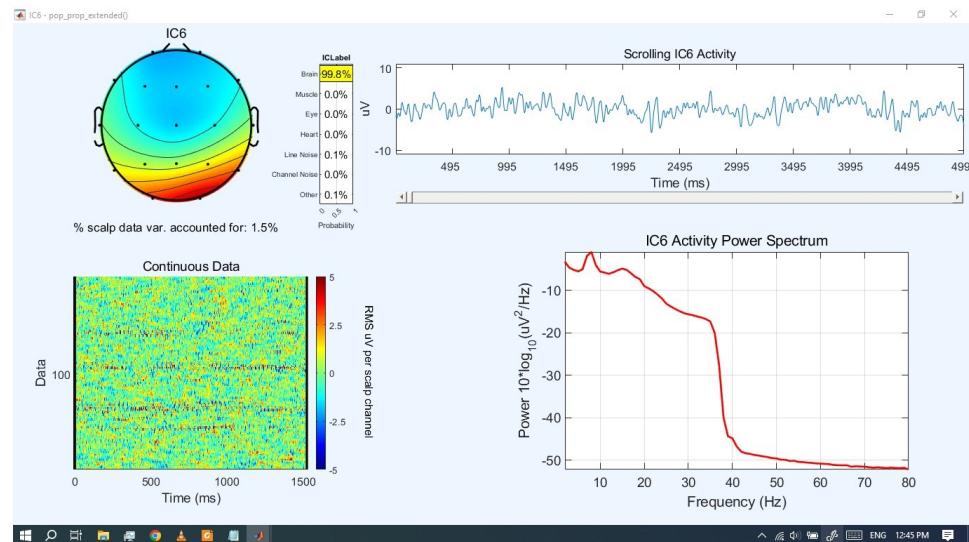


Figure 17: Subject1 - A brain component in detail

Same procedure was done on the Subject2 and here we can observe the corresponding result,

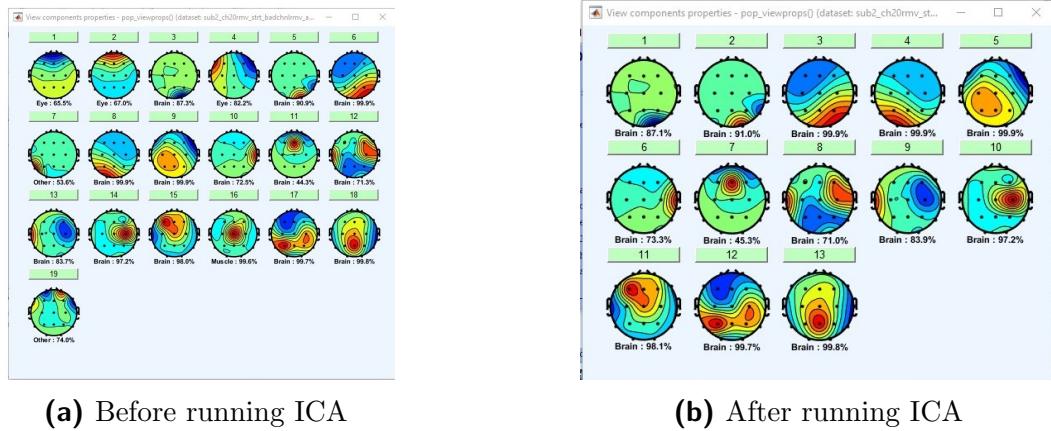


Figure 18: Subject2 Components

Besides, a detailed figure of a brain wave is presented here for subject2:

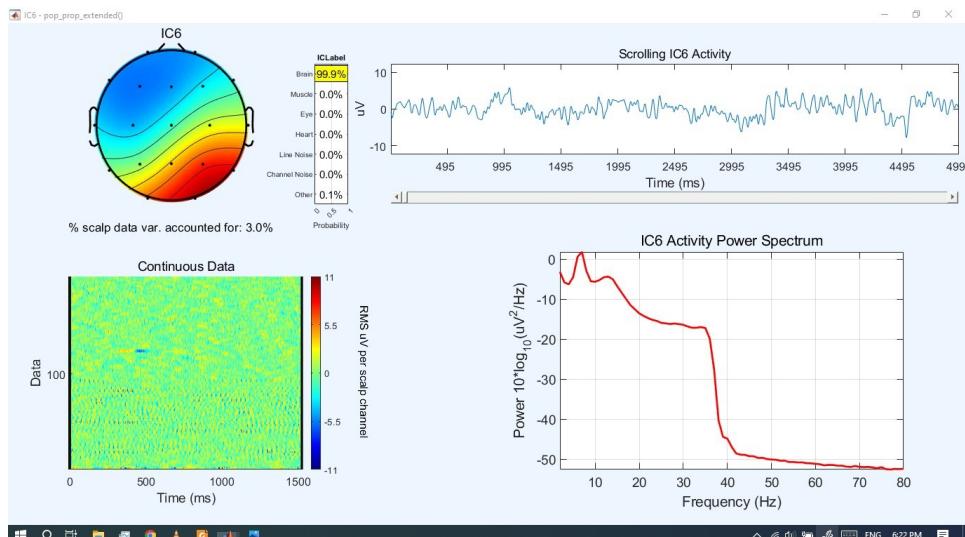


Figure 19: Subject2 - A brain component in detail

- **Step 3:** Epoch the data of each subject. Epoch is a 3D matrix of the shape $\{\text{Num_Channels} \times \text{Samples} \times \text{Num_Trials}\}$. In fact, all data must be reshaped as the following figure suggests:

For epoching the data, starting point of the experiment is required. This is provided in the `help` file for each subject. Please note that you must epoch the data by considering this time as the start. Also, the data after 120 trials should be neglected as well.

Step 3

Whole data of the running programs in Matlab is saved in workspace, since we can obtain the EEGLab Pre-processed data by a field of a struct called 'EEG':

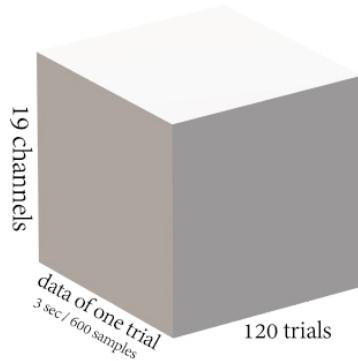


Figure 20: Epoch

”EEG.data” In order to epoch this data, following function was designed.(Note that the timing effect of the starting-time was considered)

```

function epochMatrix = epoch(eegData)
%This function epochs the pre-processed data in the
%desired format

for j = 1 : 19
    for i = 1 : 120
        epochMatrix(j,:,:,i) = eegData(j,(i*2000-599):(i
            *2000));
    end
end

end

```

The epoched data is also attached.(outputEpoch1.mat and outputEpoch2.mat)

- **Step 4:** In this step, you need to remove noisy trials. There are two ways to achieve this:

1. Observe data at EEGLAB and remove any trial that seems noisy. (PREFERRED!)
2. Using power spectrum of each trial, remove trials that their standard deviation of their power spectrum is bigger than 3.5 .Create a 3D matrix by each trial’s power spectrum for each channel using `pspectrum` in MATLAB. You can use the following commands to find noisy trials:

```

vr = sum(nanstd(p,[],2).^2,2);
noisy_trials = find(abs(zscore(vr))>3.5);

```

In these commands, `p` is a matrix of frequency spectrums of all trials of a channel. `noisy_trials` contains the number of noisy trials of that channel. These commands

must run for each channel individually and the resultant noisy trials must be accumulated over all channel. Then remove all **noisy_trials** from your epoch.

Step 4

Using EEGLab, again we load the epoched form of the data and create a new dataset.(Mentioned datasets are saved in output folders) EEGLab wisely detects that this an epoched data and lets us remove the noisy trial(s).We check all 120 trials manually and highlight them. Note that the received signal was very clean and contained just a few noisy trials.

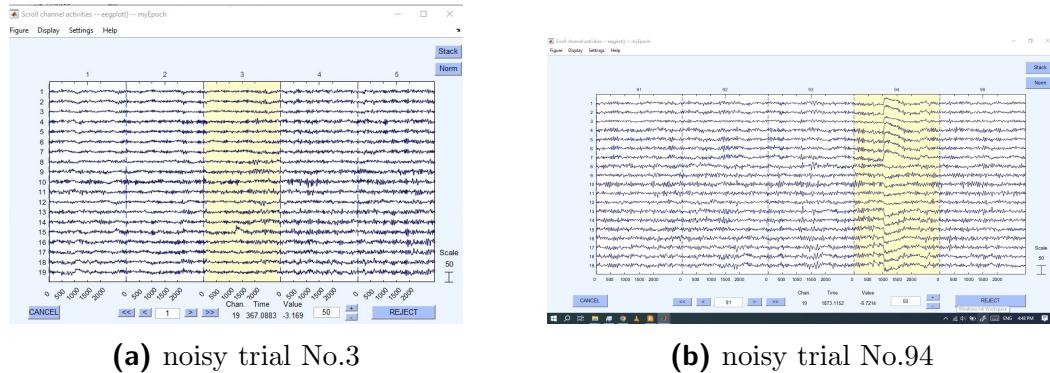


Figure 21: Subject1 - Noisy Trial Remove

The same action is taken for the subject2 as well,

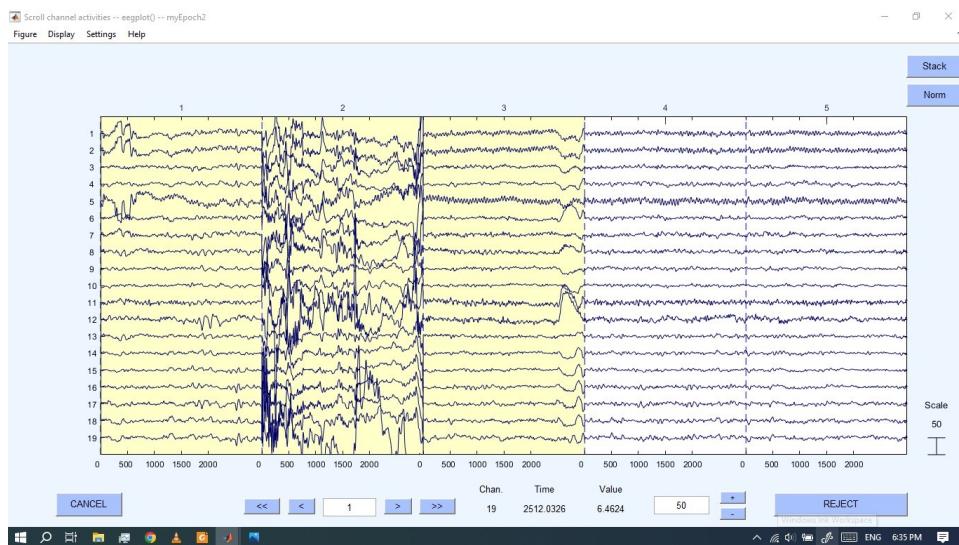


Figure 22: Subject2 - Noisy Trial Remove

- **Step 5:** In the final step, only subsample the data corresponding to the Fp1, Fz, Cz & Pz channels. You can find the channels' orders in the **Channels.jpg**.

Do these 5 steps for each subject and save the final data through an **struct** with the same

format as described in [Table 1](#). Also, consider the order of `odor` being the same as the ones used for normal participants.

Step 5

Channels' order are as follows:

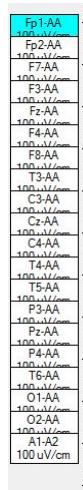


Figure 23: Order of channels

According to these orders, the following channels are ought to be sub-sampled: - Fp1(1) - Fz(5) - Cz(10) - Pz(15)

The necessary function is provided here:

```
function subSampledEpoch = subSample(UnsampledEpoch)
% This function prepare subsamples of the epoch

subSampledEpoch(1,:,:,:) = UnsampledEpoch(1,:,:,:); %Fp1
subSampledEpoch(2,:,:,:) = UnsampledEpoch(5,:,:,:); %Fz
subSampledEpoch(3,:,:,:) = UnsampledEpoch(10,:,:,:); %Cz
subSampledEpoch(4,:,:,:) = UnsampledEpoch(15,:,:,:); %Pz

end
```

We also note the removed trials' No in order to mention in output's noisy field. Odors also are sorted like 'Normal.mat' struct.

1x1 <u>struct</u> with 3 fields	
Field	Value
epoch	4x600x118 <i>single</i>
oder	118x1 <i>double</i>
noisy	[3,94]

(a) Sample1 - Output struct1

1x1 <u>struct</u> with 3 fields	
Field	Value
epoch	4x600x117 <i>single</i>
oder	117x1 <i>double</i>
noisy	[1,2,3]

(b) Sample2 - Output struct2

Figure 24: Output Structs of subsampled epoches

3.4 Phase Locking Value (PLV)

Phase Locking Value (PLV) is a metric used to quantify the degree of phase synchronization or phase consistency between two oscillatory signals. It assesses the relationship between the phases of two signals at a specific frequency range. PLV is commonly used in the analysis of neural signals, including electroencephalography (EEG) and magnetoencephalography (MEG), to investigate the synchronization of oscillatory activity between different brain regions or across different frequency bands within a single region.. It provides insights into the functional connectivity and coordination of neural activity.

PLV ranges from 0 to 1, where a value of 1 indicates perfect phase synchronization, while a value close to 0 represents a lack of synchronization. High PLV values suggest that the phases of the two signals are consistently aligned or coupled, indicating strong synchronization. This synchronization can reflect functional interactions between brain regions or coordinated activity within a network. In contrast, low PLV values indicate weaker or desynchronized activity, suggesting less functional coupling between the signals.

- What does phase synchronization indicate from a functional point of view? Discuss its importance with valid references.
- Formulate the definition of PLV and briefly discuss the mathematical tools needed to calculate it.
- Implement a function which finds the PLV between two channels in a specific frequency range. This function is going to be needed in the [section 4](#). (NOTE: You are allowed to define this function with any required input arguments.)

PLV Calculator Function

Two functions below can be used based on our purpose. The following function can handle two signals(Channels) and calculate the plv of them in out desired frequency range by using a band-pass filter(Matlab's function).

```
function plv = PLV( sig1, sig2, freqRange )
fs = 200; % Sampling Rate
sig1 = bandpass(sig1,freqRange,fs);
sig2 = bandpass(sig2,freqRange,fs);
% Hilbert Transform
phase_sig1 = angle(hilbert(sig1));
phase_sig2 = angle(hilbert(sig2));
phaseDiff = phase_sig1 - phase_sig2;

% Can also be calculated in this way:
% z1 = sig1 + 1i*hilbert(sig1);
% z2 = sig2 + 1i*hilbert(sig2);
% phaseDiff = angle(z1.*conj(z2));
```

```

Ntrials = 600;
% compute PLV
e = exp(1i*(phaseDiff));
plv = abs(sum(e) / Ntrials );

end

```

The second implemented function, accepts a full patient's EEG struct and calculate the plv using the built-in special filter fo this case:

```

function plv = myPLV(patientEEG, type, freqRange)
samples = 600;
SR = 200; %Sample Rate
epoch = patientEEG.epoch;
Ntrials = size(epoch,3);
freqRange = [35 40];
%b = fir1(n,Wn) uses a Hamming window to design an nth-
%order lowpass,
% bandpass, or multiband FIR filter with linear phase.
% The filter type
% depends on the number of elements of Wn.
b = fir1(50, 2/SR*freqRange);
% y = filter(b,a,x,zi,dim) acts along dimension dim. For
% example, if x
% is a matrix, then filter(b,a,x,zi,2) returns the
% filtered data for
% each row.
filterData = filter(b,1, epoch, [], 2); %fir1 -> Designs
% filter

hilberted(1,:,:,:) = hilbert(filterData(2, :, :)); %Fz
hilberted(2,:,:,:) = hilbert(filterData(3, :, :)); %Cz

odors = patientEEG.odor;
nTrial = 0;
for trial = 1:Ntrials
    if odors(trial)==type
        nTrial = nTrial+1;
        phi = angle(hilberted(1,:,trial).*conj(hilberted
            (2,:,trial)));
        plvSum(trial) = abs(sum(exp(1i*phi))/samples);
    end
end
plv = sum(plvSum)/nTrial;
end

```

4 Results

In this section, you need to present the required results to assess the difference of Phase Locking Values (PLV) among two groups, namely AD and Normal in the slow gamma frequency range, which is 35 to 40 Hz.

To fairly compare your results in this part, you do not need to use your preprocessing data from section 3.3 and the preprocessed data of 15 healthy (normal) (age = 69.27 ± 6.65 , female = 53.33%) individuals and 13 AD patients (age = 75.31 ± 9.90 , female = 61.54%) are available through `Dataset/Normal.mat` and `Dataset/AD.mat`.

4.1 Values

Find the PLV for all participants of both groups on both frequent and rare odors between the **Fz** and **Cz** channels using the function you implemented in section 3.4 .

4.1

First we calculated the PLV using the first function:

```
load Normal.mat
load AD.mat
% PLV for normal

fs = 200;
sumNormalFrequentPLV = 0; %15 members
sumNormalRarePLV = 0; %15 members
Frequent = 0;
Rare = 0;
for i = 1:15
    %SignalChannel2=[];
    matrix1 = normal(i).epoch(2,:,:);
    SignalChannel2 = reshape(matrix1,1,[]); % convert matrix
    to row vector

    matrix2 = normal(i).epoch(3,:,:);
    SignalChannel3 = reshape(matrix2,1,[]); % convert matrix
    to row vector

    BPS1 = bandpass(SignalChannel2,[35 40],fs);
    BPS2 = bandpass(SignalChannel3,[35 40],fs);
    [normal1Size,~] = size(normal(i).odor);
```

```

matrix1 = reshape(BPS1, 600, normal1Size); %use 10
    columns and as many rows as necessary
matrix2 = reshape(BPS2, 600, normal1Size); %use 10
    columns and as many rows as necessary

for j = 1: normal1Size
    if (normal(i).odor(j) == 0) %lemon - Frequent
        Frequent = Frequent + 1;
        sumNormalFrequentPLV = sumNormalFrequentPLV +
        PlvOf2Signal(matrix1(:, j), matrix2(:, j));

    else %Rose - Rare
        Rare = Rare + 1;
        sumNormalRarePLV = sumNormalRarePLV +
        PlvOf2Signal(matrix1(:, j), matrix2(:, j));

    end
end

normalFrequentPLV(i) = sumNormalFrequentPLV/Frequent ;
normalRarePLV(i) = sumNormalRarePLV/Rare;
end
%%%%%%%%%%%%%
% PLV for AD

sumADFrequentPLV = 0; %13 members
sumADRAREPLV = 0; %13 members
Frequent = 0;
Rare = 0;
for i = 1:13
    %SignalChannel2=[];
    matrix1 = AD(i).epoch(2, :, :);
    SignalChannel2 = reshape(matrix1, 1, []); % convert matrix
    to row vector

    matrix2 = AD(i).epoch(3, :, :);
    SignalChannel3 = reshape(matrix2, 1, []); % convert matrix
    to row vector

    BPS1 = bandpass(SignalChannel2, [35 40], fs);
    BPS2 = bandpass(SignalChannel3, [35 40], fs);
    [AD1Size, ~] = size(AD(i).odor);

```

```

matrix1 = reshape(BPS1, 600, AD1Size ); %use 10 columns
    and as many rows as necessary
matrix2 = reshape(BPS2, 600, AD1Size ); %use 10 columns
    and as many rows as necessary

for j = 1:AD1Size
    if (AD(i).odor(j) == 0) %lemon - Frequent
        Frequent = Frequent + 1;
        sumADFrequentPLV = sumADFrequentPLV +
            PlvOf2Signal(matrix1(:,j),matrix2(:,j));
    else %Rose - Rare
        Rare = Rare + 1;
        sumADRAREPLV = sumADRAREPLV + PlvOf2Signal(
            matrix1(:,j),matrix2(:,j));
    end
end

ADFrequentPLV(i) = sumADFrequentPLV/Frequent ;
ADRAREPLV(i) = sumADRAREPLV/Rare;

end

```

The corresponding results are saved at 'PLV1.mat'. Results poss a very low standard deviation(i.e very close to mean-value).Thus, they seem slightly unreliable.Therefore, We use the second function below:

```

load Normal.mat
load AD.mat

% PLV of Normal Group
group = normal;
patientsNum = size(group,2);
for patient = 1 : patientsNum
    NFplv(patient) = myPLV(group(patient),0); % Freq
    NRplv(patient) = myPLV(group(patient),1); % Rare
end

% PLV of AD Group
group = AD;
patientsNum = size(group,2);
for patient = 1 : patientsNum
    ADFplv(patient) = myPLV(group(patient),0); %Freq
    ADRplv(patient) = myPLV(group(patient),1); % Rare

```

```

end

function plv = myPLV(patientEEG, type)
samples = 600;
SR = 200;
epoch = patientEEG.epoch;
Ntrials = size(epoch,3);
freqRange = [35 40];
%b = fir1(n,Wn) uses a Hamming window to design an nth-
%order lowpass, bandpass, or multiband FIR filter with
%linear phase. The filter type depends on the number of
%elements of Wn.
b = fir1(50, 2/SR*freqRange);
% y = filter(b,a,x,zi,dim) acts along dimension dim. For
% example, if x is a matrix, then filter(b,a,x,zi,2)
% returns the filtered data for each row.
filterData = filter(b, 1, epoch, [], 2); %fir1 -> Designs
% filter
hilberted(1,:,:,:) = hilbert(filterData(2, :, :)); %Fz
hilberted(2,:,:,:) = hilbert(filterData(3, :, :)); %Cz

odors = patientEEG.odor;
nTrial = 0;
for trial = 1:Ntrials
    if odors(trial)==type
        nTrial = nTrial+1;
        phi = angle(hilberted(1,:,trial).*conj(hilberted
            (2,:,trial)));
        plvSum(trial) = abs(sum(exp(1i*phi))/samples);
    end
end
plv = sum(plvSum)/nTrial;
end

```

The corresponding results are saved at 'PLV2.mat'. Results poss a rational standard deviation(i.e there is some outlier data). Thus, they seem more realistic. This would be discussed in detail in the Conclusion.

4.2 Distributions

Draw the box plots of PLVs you found in the previous part among two groups and two odors. Also, fit a gaussian distribution on these PLVs and present you results. You need to specify the corresponding **p-values** to evaluate the statistical significance of your findings.

4.2

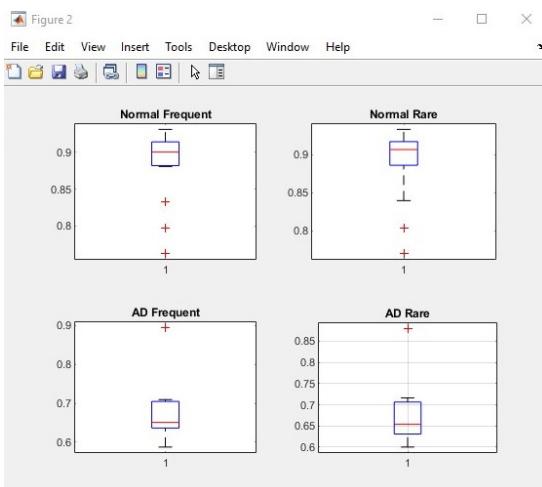
Using the code below, box plots are plotted:

```

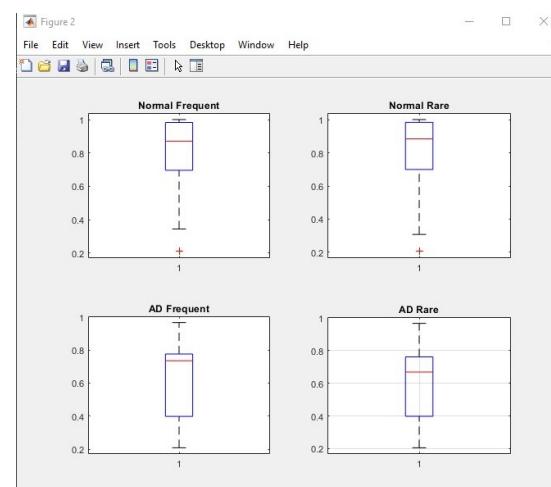
figure
subplot(2,2,1)
boxplot(NFplv);
title("Normal Frequent");
grid on;
subplot(2,2,2)
boxplot(NRplv);
title("Normal Rare");
grid on;
subplot(2,2,3)
boxplot(ADFplv);
title("AD Frequent");
grid on;
subplot(2,2,4)
boxplot(ADRplv);
title("AD Rare");
grid on;

```

box plots corresponding to the PLV1 and PLV2 are as follows:



(a) Box-plot For PLV1



(b) Box-plot For PLV2

Figure 25: Desired Box-plots

From now using PLV2, Now we fit a guassian distribution:

```

x_values = -0.1:0.01:1.8;
figure
subplot(2,2,1)
NFgf = fitdist ('NFplv' , 'Normal');

```

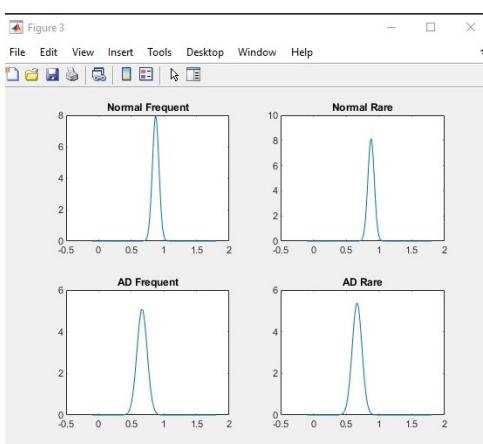
```
% NFgf = fitdist ( normalFrequentPLV' , 'Normal');
plot(x_values, pdf(NFgf, x_values));
title("Normal Frequent");

subplot(2,2,2)
NRgf = fitdist ( NRplv' , 'Normal');
% NRgf = fitdist ( normalRarePLV' , 'Normal');
plot(x_values, pdf(NRgf, x_values));
title("Normal Rare");

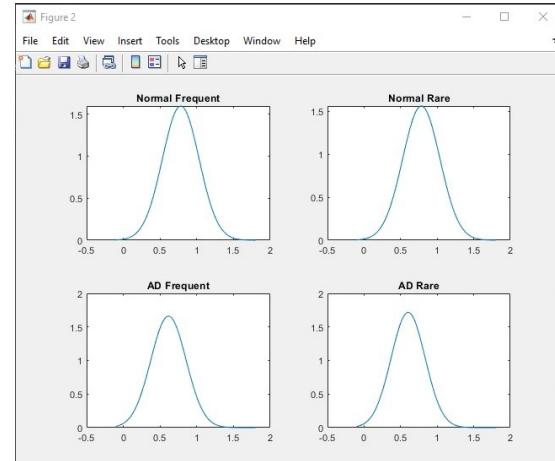
subplot(2,2,3)
ADFgf = fitdist ( ADFplv' , 'Normal');
% ADFgf = fitdist ( ADFrequentPLV' , 'Normal');
plot(x_values, pdf(ADFgf, x_values));
title("AD Frequent");

subplot(2,2,4)
ADRgf = fitdist ( ADRplv' , 'Normal');
% ADRgf = fitdist ( ADRarePLV' , 'Normal');
plot(x_values, pdf(ADRgf, x_values));
title("AD Rare");
```

Results are presented here:



(a) Gaussianfit For PLV1



(b) Gaussianfit For PLV2

Figure 26: Gaussian Fit

P-value is calculated using `ttest2()` built-in function of Matlab based on PLV2. (Significance level of 5%)

```
[H0 ,Pvalue] = ttest2 ( NRplv ,ADFplv);
[H02 ,Pvalue2] = ttest2 ( ADRplv ,NRplv);
```

Thus:

- P-value = 0.090507971827578 :

The returned value of $H0 = 0$ indicates that ttest2 does not reject the null hypothesis at the default 5% significance level.

- P-value2 = 0.068679703189846 :

The returned value of $H02 = 0$ indicates that ttest2 does not reject the null hypothesis at the default 5% significance level.

4.3 Statistical Significance

Based on the **p-values** you founded in the previous part, discuss whether we could state that the "PLV is significantly different among AD and Normal subjects in the slow gamma frequency range".

4.3

PLV evaluates the spread of the distribution of phase differences, and the connectivity estimation is linked to this spread. The narrower the distribution of the phase difference, the higher the PLV value, which ranges between zero (no phase dependence) and one (complete phase dependence). Here we briefly describe the main steps involved.

The most common method to calculate PLV is based on the instantaneous phase of the signals obtained using the Hilbert or the wavelet transforms. In both cases, the calculations are fast and reliable, which has undoubtedly contributed to its use. This is especially important in the case of M/EEG, distributed source-space analysis, where such number ranges from the thousands to the hundreds of thousands.

Lachaux (Lachaux et al.) defined the PLV as a time-dependent connectivity measured tailored to study evoked activity. The idea behind their definition is that the stimulus resets the phase of the neural oscillators so that signals connected in a given time should have a stable phase-difference along trials. In mathematical terms:

$$PLV_{i,j}(t) = \frac{1}{N} \left| \sum_{n=1}^N e^{-i(\phi_i(t,n) - \phi_j(t,n))} \right|$$

where N is the number of trials and (\cdot) is the instantaneous phase for signal i in trial n at time t . This definition can be extended to resting-state data, by assessing phase locking as a stable phase difference over time, thereby obtain the so-called MPC1:

$$PLV_{i,j}(t) = \frac{1}{T} \left| \sum_{n=1}^T e^{-i(\phi_i(t) - \phi_j(t))} \right|$$

where T is the data length. In either case, one has to extract the instantaneous phase (\cdot) of each signal. Besides, for the phase to be physically meaningful, it is necessary that only one oscillator is present in each signal. This is achieved, e.g., by means of a narrow-band pass filtering or, equivalently, the convolution with a narrow band complex wavelet such as that of Morlet (Bruns 2004). After the filtering process, we obtain a band-pass version

of the Hilbert analytical signal:

$$x_{BP,H}(t) = x_{BP}(t) + ix_{BP}^*(t) = B(t) \cdot e^{-i\phi_i(t)} \tilde{x} = Hilbert(x)$$

The instantaneous phase is usually extracted from this analytical signal, the phase difference estimated and, finally, the exponentiation calculated to get the unit phase difference vector. However, these two operations (phase extraction and exponentiation) are computationally expensive, but, as we will show, they can be easily circumvented by using the properties of exponentials. First, let us obtain the oscillatory part of the analytical signal by normalizing (3):

$$\dot{x}_{BP,H,i}(t) = \frac{x_{BP,H,i}(t)}{|x_{BP,H,i}(t)|} = e^{-i\phi_i(t)}$$

So phase difference can be calculated:

$$\dot{x}_{BP,H,i}(t) \cdot (\dot{x}_{BP,H,i}(t))^* = e^{-i(\phi_i(t) - \phi_i(t))}$$

4.4 Phase Difference

Draw a polar histogram of the phase difference between Fz and Cz channels during frequent odor trials for a random subject in each group and compare the results. Also, plot the mean value of this quantity among all the subjects of each group and discuss the results.

4.4

Here, first we assume two random people from the two groups and then we find phase difference using formulas stated in the last section.

```

figure
subplot(2,2,1);
polarhistogram(double(phaseCalc(normal(1))))
title('patient 1 Normal frequent')
subplot(2,2,2);
polarhistogram(double(phaseCalc(AD(1))))
title('patient 1 AD frequent')

subplot(2,2,3);
group = normal;
meanPH_N = 0;
for patient = 1:15
    meanPH_N = meanPH_N+phaseCalc(group(patient));
end
meanPH_N = meanPH_N/15;
polarhistogram(double(meanPH_N))
title('mean Normal frequent')

```

```
subplot(2,2,4);
meanPH_AD = 0;
group = AD;
for patient = 1:13
    meanPH_AD = meanPH_AD+phaseCalc(group(patient));
end
meanPH_AD = meanPH_AD/13;

polarhistogram(double(meanPH_AD))
title('mean AD frequent')

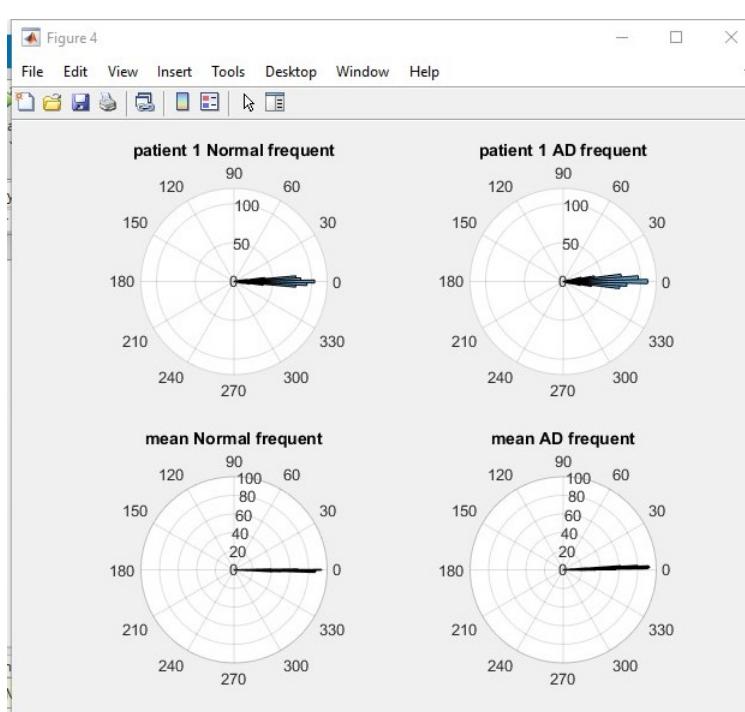
function phDif = phaseCalc(patientEEG)
    SR = 200; % Sample Rate
    epoch = patientEEG.epoch;
    odors = patientEEG.odor;
    trials = size(epoch,3);
    filterRange = [35 40];
    filterData = filter(fir1(50, 2/SR*filterRange),1,
        epoch, [], 2);
    %fir1 -> Designs filter

    hilberted(1,:,:,:) = hilbert(filterData(2, :, :)); %Fz
    hilberted(2,:,:,:) = hilbert(filterData(3, :, :)); %Cz

    nTrial = 0;
    phDif = 0;

    for trial = 1:trials
        if odors(trial)== 0
            nTrial = nTrial+1;
            phase = angle(hilberted(1,:,:).*conj(
                hilberted(2,:,:)));
            phDif = mean(phase,3);

        end
    end
end
```

**Figure 27:** Phase difference

4.5 Heatmaps

Now you need to plot a heatmap which has the PLVs between each pair of the channels. Find whether PLV between other channel pairs are significantly different among two groups in the slow gamma frequency range and test your results. (NOTE: You need to provide p-values for your hypothesis if you found any significantly different channel pairs apart from (Fz,Cz).)

4.5

In order to check whether PLV between other channel pairs are significantly different among two groups, it is recommended to form two tables for each group. Each table has 6 rows ($\binom{4}{2}$) and 15 columns for normal people group and 13 columns for AD people group. Comparing these two table and the heatmap we would be able to decide. Code:

```

load Normal.mat
load AD.mat

% PLV of Normal Group
group = normal;
normalNum = size(group,2);

for normalPerson = 1 : normalNum

```

```
n1 = 1;
for ch1 = 1 : 4
    for ch2 = (ch1+1) : 4
        Nplv(normalPerson, n1) = myPLV(group(normalPerson
            ), 0, ch1, ch2); % Freq
        n1 = n1 + 1;
    end
end

% PLV of AD Group

group = AD;
patientsNum = size(group, 2);

for patient = 1 : patientsNum
    n2 = 1;
    for ch1 = 1 : 4
        for ch2 = (ch1+1) : 4
            ADplv(patient, n2) = myPLV(group(patient), 0, ch1,
                ch2); %Freq
            n2 = n2 + 1;
        end
    end
end

figure
subplot(1, 2, 1)
h1 = heatmap(Nplv);
xlabel("Channels (In order: FpFz - FpCz - FpDz - FzCz - FzDz - CzDz )")
ylabel("normal people (In order: 1 - 15 )")
title("Normal");
subplot(1, 2, 2)
h2 = heatmap(ADplv);
xlabel("Channels (In order: FpFz - FpCz - FpDz - FzCz - FzDz - CzDz )")
ylabel("AD people (In order: 1 - 13 )")
title("AD");

function plv = myPLV(patientEEG, type, Channel1, Channel2)
    samples = 600;
    SR = 200;
    epoch = patientEEG.epoch;
    Ntrials = size(epoch, 3);
    freqRange = [35 40];
```

```
%b = fir1(n,Wn) uses a Hamming window to design an nth-
order lowpass, bandpass, or multiband FIR filter with
linear phase. The filter type depends on the number of
elements of Wn.
b = fir1(50, 2/SR*freqRange);
% y = filter(b,a,x,zi,dim) acts along dimension dim. For
% example, if x is a matrix, then filter(b,a,x,zi,2)
% returns the filtered data for each row.
filterData = filter(b ,1, epoch, [], 2); %fir1 -> Designs
% filter

hilberted(1,:,:,:) = hilbert(filterData(Channel1, :, :)); %
Fz
hilberted(2,:,:,:) = hilbert(filterData(Channel2, :, :)); %
Cz

odors = patientEEG.odor;
nTrial = 0;
for trial = 1:Ntrials
    %if odors(trial)==type
    %nTrial = nTrial+1;
    %phi = angle(hilberted(1,:,trial).*conj(hilberted
    %(2,:,trial)));
    %plvSum(trial) = abs(sum(exp(1i*phi))/samples);
    %end
end
plv =sum(plvSum)/nTrial;
end
```

The result is provided down here:

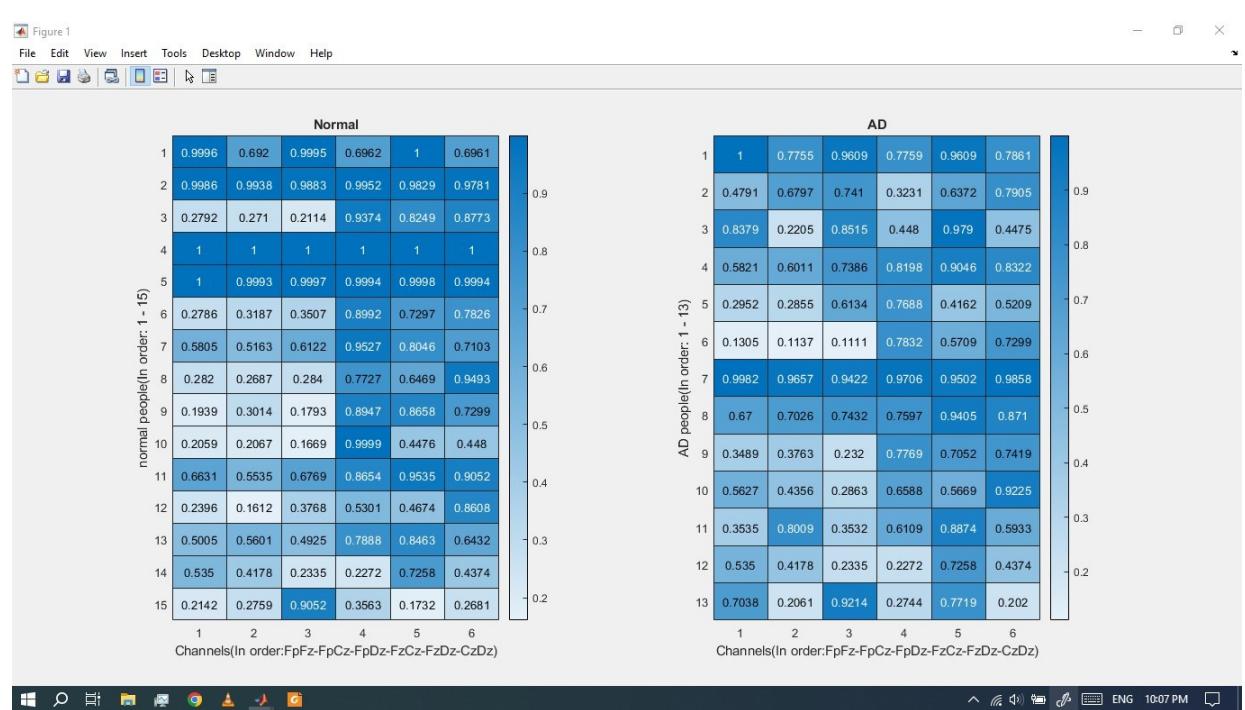


Figure 28: Heatmap

Comparing two groups, apparently there are no other channels(Apart from Fz-Cz) to be worry about and this is a good news for us.

5 *Bonus

5.1 Mild Cognitive Impairment (MCI)

Mild Cognitive Impairment (MCI) is the stage between the expected decline in memory and thinking that happens with age and the more serious decline of dementia. MCI may include problems with memory, language or judgment. People with MCI may be aware that their memory or mental function has slipped. [1]

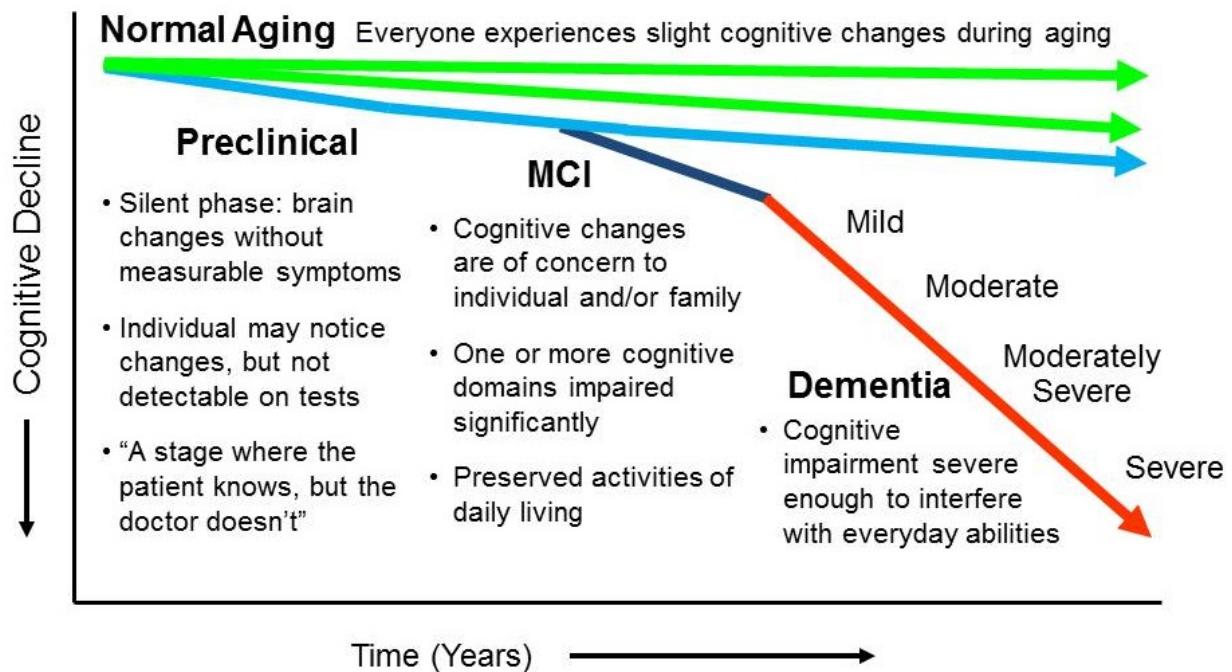


Figure 29: Normal Aging to Dementia Process

5.1.1 Additional Information

Describe the relationship between MCI and AD. Explain whether MCI would always result in AD and briefly investigate the causes of MCI.

About MCI

The symptoms of MCI are not as severe as those of Alzheimer's disease or a related dementia. People with MCI can usually take care of themselves and carry out their normal daily activities.

People with MCI are at a greater risk of developing Alzheimer's disease or a related dementia. Estimates vary as to how many people who have MCI will develop dementia. Roughly one to two out of 10 people age 65 or older with MCI are estimated to develop

dementia over a one-year period. However, in many cases, the symptoms of MCI stay the same or even improve.

Many factors can cause problems with memory and thinking. There is no single cause of MCI, and it's more likely to occur as someone ages. Estimates vary, but roughly 10% to 20% of people over age 65 have MCI, with the risk increasing as someone gets older. Other factors like genetics and certain conditions — including diabetes, depression, and stroke — may affect a person's risk for MCI.

5.1.2 MCI Data Processing

In the provided dataset, you can find `MCI.mat` file. This dataset contains preprocessed cleaned EEG recording of the same task described in sections 3.1 and 3.2 for 7 MCI patients.

Based on the significantly different coupled channels you found for differentiation between AD and Normal groups, find the Phase-Locking-Value (PLV) for the MCI subjects and provide the required results by comparing all the the 3 states (Normal, MCI, AD). Your findings must include the significance testing by providing the corresponding `p-values`.

MCI Data processing

The same procedure of section 4 is done on the the new dataset 'MCI.mat'.First we find the PLV :

```

load Normal.mat
load MCI.mat

% PLV of Normal Group
group = normal;
patientsNum = size(group,2);
for patient = 1 : patientsNum
    NFplv(patient) = myPLV(group(patient),0); % Freq
    NRplv(patient) = myPLV(group(patient),1); % Rare
end

% PLV of AD Group
group = MCI;
patientsNum = size(group,2);
for patient = 1 : patientsNum
    MFplv(patient) = myPLV(group(patient),0); % Freq
    MRplv(patient) = myPLV(group(patient),1); % Rare
end

function MCIPlv = myPLV(patientEEG, type)
    samples = 600;

```

```

SR = 200;
epoch = patientEEG.epoch;
Ntrials = size(epoch,3);
freqRange = [35 40]; % Gamma Range
%b = fir1(n,Wn) uses a Hamming window to design an nth-
% order lowpass, bandpass, or multiband FIR filter with
% linear phase. The filter type depends on the number of
% elements of Wn.
b = fir1(50, 2/SR*freqRange);
% y = filter(b,a,x,zi,dim) acts along dimension dim. For
% example, if x is a matrix, then filter(b,a,x,zi,2)
% returns the filtered data for each row.
filterData = filter(b ,1, epoch, [], 2); %fir1 -> Designs
% filter

hilberted(1,:,:,:) = hilbert(filterData(2, :, :)); %Fz
hilberted(2,:,:,:) = hilbert(filterData(3, :, :)); %Cz

odors = patientEEG.odor;
nTrial = 0;
for trial = 1:Ntrials
    if odors(trial)==type
        nTrial = nTrial+1;
        phi = angle(hilberted(1,:,trial).*conj(hilberted
            (2,:,trial)));
        plvSum(trial) = abs(sum(exp(1i*phi))/samples);
    end
end
MCIpIvlv =sum(plvSum)/nTrial;
end

```

The corresponding results are saved at 'MCIpIvlvs.mat'.

Now, we box plot:

```

figure
subplot(2,2,1)
boxplot(NFplv);
title("Normal Frequent");
grid on;
subplot(2,2,2)
boxplot(NRplv);
title("Normal Rare");
grid on;
subplot(2,2,3)
boxplot(MFplv);

```

```

title("MCI Frequent");
grid on;
subplot(2,2,4)
boxplot(MRplv);
title("MCI Rare");
grid on;

```

Result:

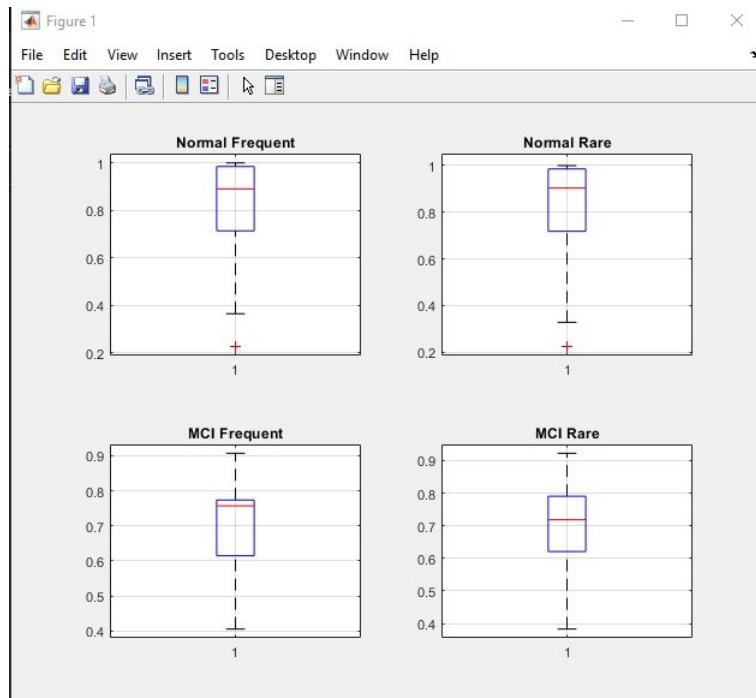


Figure 30: MCI box plot

Now, we fit a guassian distribution:

```

x_values = -0.1:0.01:1.8;
figure
subplot(2,2,1)
NFgf = fitdist ( NFplv' , 'Normal');
% NFgf = fitdist ( normalFrequentPLV' , 'Normal');
plot(x_values, pdf(NFgf, x_values));
title("Normal Frequent");

subplot(2,2,2)
NRgf = fitdist ( NRplv' , 'Normal');
% NRgf = fitdist ( normalRarePLV' , 'Normal');
plot(x_values, pdf(NRgf, x_values));
title("Normal Rare");

```

```

subplot(2,2,3)
MFgf = fitdist ( MFplv' , 'Normal');
% ADFgf = fitdist ( ADFrequentPLV' , 'Normal');
plot(x_values, pdf(MFgf, x_values));
title("MCI Frequent");

subplot(2,2,4)
MRgf = fitdist ( MRplv' , 'Normal');
% ADRgf = fitdist ( ADRarePLV' , 'Normal');
plot(x_values, pdf(MRgf, x_values));
title("MCI Rare");

```

Result:

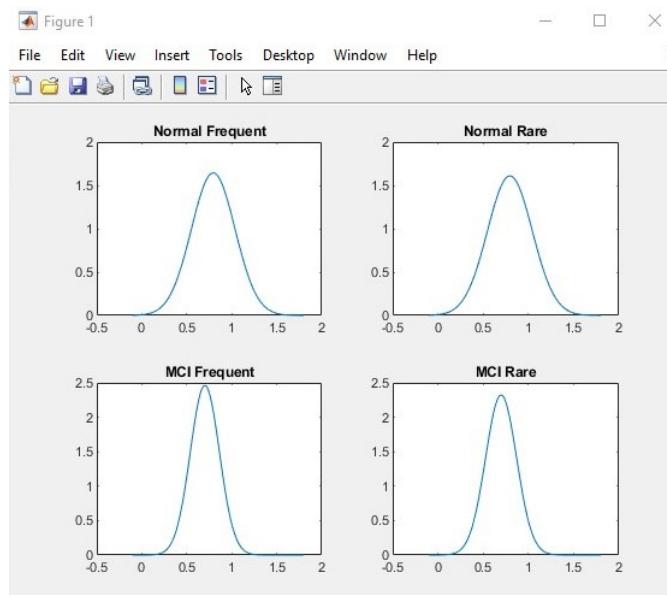


Figure 31: MCI gaussian fit

P-value is calculated using `ttest2()` built-in function of Matlab based on MCI-PLV.(Significance level of 5%)

```

[HOM ,PvalueM] = ttest2 ( MFplv ,ADFplv);
[H02M ,Pvalue2M] = ttest2 ( MRplv ,NRplv);

```

Thus:

- P-valueM = 0.5119539 :

The returned value of $H_0 = 0$ indicates that `ttest2` does not reject the null hypothesis at the default 5% significance level.

- P-value2M = 0.3622465 :

The returned value of $H_0 = 0$ indicates that `ttest2` does not reject the null hypothesis at the default 5% significance level.

5.2 Phase-Amplitude Coupling (PAC)

PLV was just one instance of the Phase-Amplitude Coupling (PAC) metrics. PAC is a form of cross-frequency coupling where the amplitude of a high frequency signal is modulated by the phase of low frequency oscillations. PAC is the most-studied type of cross-frequency coupling and is thought to be responsible for integration across populations of neurons. Low frequency brain activity controls the information exchange between brain regions by modulating the amplitude of the high frequency oscillations. [6]

5.2.1 Metrics

Conduct a search about other PAC measures and briefly provide an explanation about two of them.

PAC Metrics

Measuring Phase-Amplitude Coupling

To calculate phase-amplitude coupling, first, raw data is band-pass filtered in the frequency bands of interest. Second, the real-valued band-pass filtered signal is transformed into a complex-valued analytic signal. Finally, phase or amplitude is extracted from the complex-valued analytic signal.

Phase-Locking-Value as Used in Mormann et al. (2005)

Note that we explained this metric comprehensively in the previous sections.

Mean Vector Length by Canolty et al. (2006)

For the phase-amplitude coupling measure MVL, introduced by Canolty et al. (2006), phase is extracted from the low frequency filtered analytic signal and amplitude is extracted from the high frequency filtered analytic signal. MVL utilizes phase angle and magnitude of each complex number (i.e., each data point) of the corresponding analytic signal in a quite direct way to estimate the degree of coupling. Each complex value of the analytic time series is a vector in the polar plane.

Averaging all vectors creates a mean vector with a specific phase and length. The length of this vector represents the amount of phase-amplitude coupling. The direction represents the mean phase where amplitude is strongest. When no coupling is present, all vectors cancel each other out and the mean vector will be short. Then its direction does not represent any meaningful phase. In mathematical literature:

$$MVL = \left| \frac{\sum_{t=1}^n a_t e^{i\theta_t}}{n} \right|$$

n : the total number of data points,

t : data point,

at : amplitude at data point t,

t : phase angle at data point t

Modulation Index by Tort et al. (2008)

For calculating the MI according to Tort et al. (2008), all possible phases from -180° to 180° are first binned into a freely chosen amount of bins. Tort et al. (2008) established to use 18 bins of 20° each, which many authors follow. The amount of bins can influence the

results, as will be explained below. The average amplitude of the amplitude-providing frequency in each phase bin of the phase-providing frequency is computed and normalized by the following formula:

$$p(j) = \frac{\bar{a}}{\sum_{k=1}^N \bar{a}_k}$$

\bar{a} : average amplitude of one bin, k : running index for the bins, N : total amount of bins
 p : vector of N values.

Also we know that, Shannon entropy is computed by the following formula:

$$H(p) = - \sum_{j=1}^N p(j) \log(p(j))$$

Phase-amplitude coupling is defined by a distribution that significantly deviates from the uniform distribution. Kullback–Leibler distance, a measure for the disparity of two distributions is calculated by the following formula:

$$KLdist(U, X) = \log(N) - H(p)$$

U : uniform distribution,

X : distribution of the data,

N : total amount of bins,

$H(p)$: Shannon entropy

Finally, MI is calculated in this way:

$$MI = \frac{KLdist(U, X)}{\log(N)}$$

5.2.2 Implementation

Implement one of the metrics mentioned earlier as a biomarker for distinguishing between AD and Normal groups. Present the relevant results through plots and provide a discussion regarding the efficacy of the selected metric.

MI metric

Due to lack of proper time, we just managed to implement the function by a lot of research:

```
function [MI, KL] = modulationIndex(phase, amplitude, nBins)
    amplQ = ones(1, nBins) ./ nBins;
    amplP = binAmp(nBins, phase, amplitude);
    KL = sum(amplP .* log(amplP ./ amplQ));
    MI = KL ./ log(nBins);
end
```

```

function amplP = binAmp(nBins,phase,amplitude)

binEdges=linspace(-pi,pi,nBins+1);
[~,binIdx]=histc(phase,binEdges);

amplBin=zeros(1,nBins);
for bin=1:nBins
    if any(binIdx==bin)
        amplBin(bin)=mean(amplitude(binIdx==bin)); % Only
            bin Index
    end
end
amplP=amplBin/sum(amplBin);

end

```

This function is ready to be used to calculate MI metric.

6 Conclusion

In this section, you are required to thoroughly examine and analyze the results you have obtained throughout this project. You must provide a comprehensive discussion of your findings, highlighting their significance and relevance to the research question. You should also present any limitations or weaknesses in your study and suggest possible areas for future research. Overall, this section is critical to demonstrating the quality and validity of your research and should be approached with careful attention to detail and clarity of expression.

Cocclusion

First of all, we try to make a summary of what we have been through on this project. First, we got to know the Neurodegenerative Diseases, their relationship to Olfactory Dysfunction and we clarified the main purpose of this project. Further on this way, we got to know what EEG signals exactly are, their pros and cons, methods of measurements, sampling frequency and Moreover, We did some research on how brain emits these signals and common ways of analysing them.

Then we did some practise of data pre-processing and perceived why it is so important in order to have a successful processing procedure. EEGLab tool helped us a lot on this way. Using different methods of pre-processing such as ICA, Bad channel removing and... we became capable of extract the pure-brain signal for our processing studies.

Afterwards, we entered the processing stage where we used different statistical tools for diagnosing the diseases. We used some PAC metrics like PLV, MI and They contributed to accept or reject our hypotheses. We implemented two functions for PLV calculations with different STDs. One of them resulted in a low unpleasant PLV which was neglected.

At last, We generalized our studies by taking MCI people into consideration. Besides, we conducted a search for other Phase-Amplitude Coupling especially MI , MVL and... .Having researched, Other metrics seemed to be very useful and more research needs to be done in this field.

References

- [1] Mild cognitive impairment (mci). <https://www.mayoclinic.org/diseases-conditions/mild-cognitive-impairment/symptoms-causes/syc-20354578#:~:text=Overview,mental%20function%20has%20slipped.%22>. Last Reviewed: Jan. 18, 2023.
- [2] Neural oscillations – interpreting eeg frequency bands. <https://imotions.com/blog/learning/best-practice/neural-oscillations/>.
- [3] Neurodegenerative diseases. <https://www.niehs.nih.gov/research/supported/health/neurodegenerative/index.cfm>. Last Reviewed: June 09, 2022.
- [4] Marin C, Vilas D, Langdon C, Allobid I, López-Chacón M, Haehner A, Hummel T, and Mullol J. Olfactory dysfunction in neurodegenerative diseases. In *Curr Allergy Asthma Rep*, volume 18, 2018 Jun 15.
- [5] Sedghizadeh MJ, Hojjati H, Ezzatdoost K, Aghajan H, Vahabi Z, and Tarighatnia H. Olfactory response as a marker for alzheimer's disease: Evidence from perceptual and frontal lobe oscillation coherence deficit. *PLOS One*, 15(12), December 15, 2020.
- [6] T.T.K. Munia and S. Aviyente. Time-frequency based phase-amplitude coupling measure for neuronal oscillations. In *Scientific Reports* 9, 12441, 27 August 2019.
- [7] Mohammad Javad Sedghizadeh, Hamid Aghajan, and Zahra Vahabi. Brain electrophysiological recording during olfactory stimulation in mild cognitive impairment and alzheimer disease patients: An eeg dataset. *Data in Brief*, 48:109289, 2023.
- [8] Cécilia Tremblay and Johannes Frasnelli. Olfactory and Trigeminal Systems Interact in the Periphery. *Chemical Senses*, 43(8):611–616, 07 2018.