Laboratory Exercise 2.2

Multi-channel EEG: Analysing using EEGLAB

This intends to:

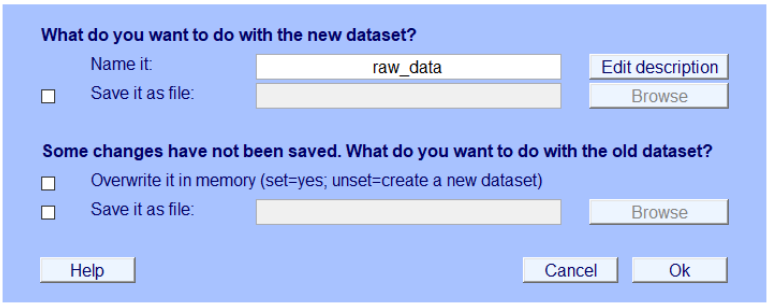
1. Familiarize you with EEGLAB opensource EEG analysis software
2. Analyze EEG in time and frequency domains including topological variation
3. Introduce you to tools related to independent component analysis (ICA)

Submission guidelines

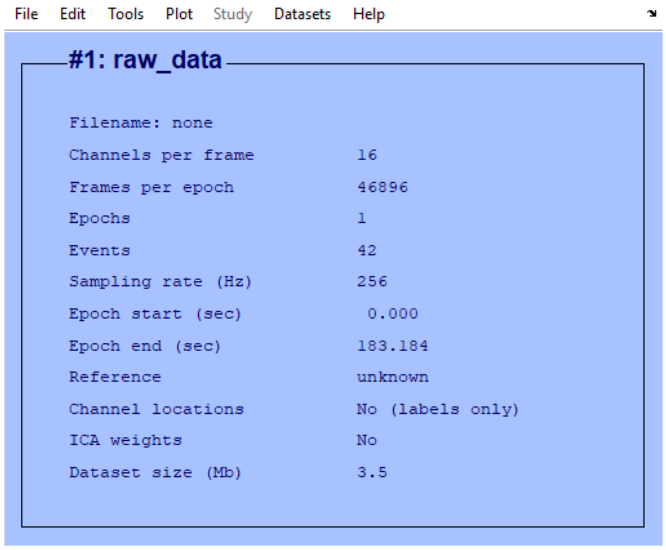
Submission document	**Submission method**	**Notes**
Report	Upload the softcopy to the Moodle	Should contain answers to questions in **boxes**. Should include observations and discussions with relevant plots to support your answers.
Viva voce examination		The submitted report will be examined.

Part 1: Installing EEGLAB and importing data

1. Install EEGLAB
   1.1. EEGLAB is an opensource MATLAB toolbox facilitating EEG signal processing. Download EEGLAB from https://sccn.ucsd.edu/EEGLAB/download.php and follow instructions.
2. Import data
   2.1. Install xdf import plugin (https://sccn.ucsd.edu/wiki/EEGLAB_Extensions)
   2.2. Import recorded data.
   File → Import data → Using EEGLAB functions and plugins → From .XDF or .XDFZ file
   2.3. Name the imported dataset as 'raw_data'

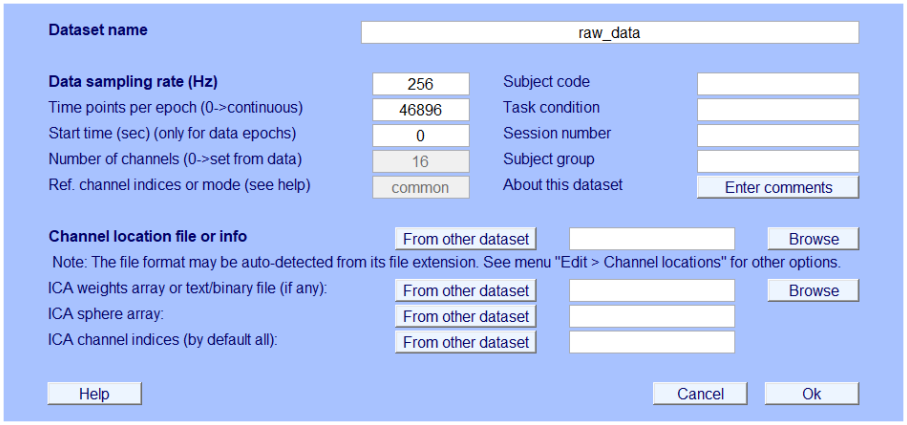


   2.4. Initial information of the dataset should look like:



Part 2: Data preparation, time and frequency domain analysis

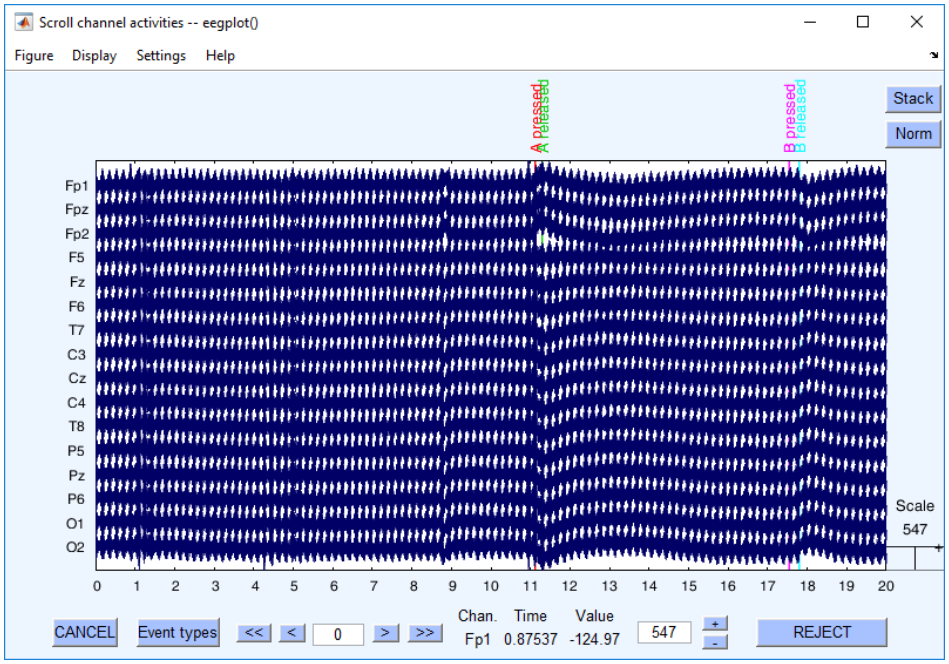
3. Channel labels
   3.1. Edit → Dataset info | Browse the 'Channel location file or info' | select 'electrode_locations.locs'



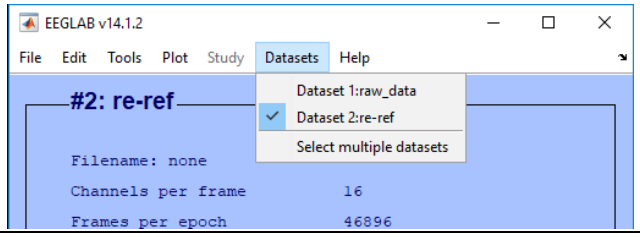
   3.2. Observe the 'Channel locations' on the dataset information GUI (section 2.4)

   3.3. Verify the channel locations by visualising: Plot → Channel locations → By name

4. Visualise data in time domain
   4.1. Plot → Channel data (scroll)
   4.2. Explore using provided button. To increase the time range to display: Setting → Time range to display → 20 s. Observe the keyboard event markers.

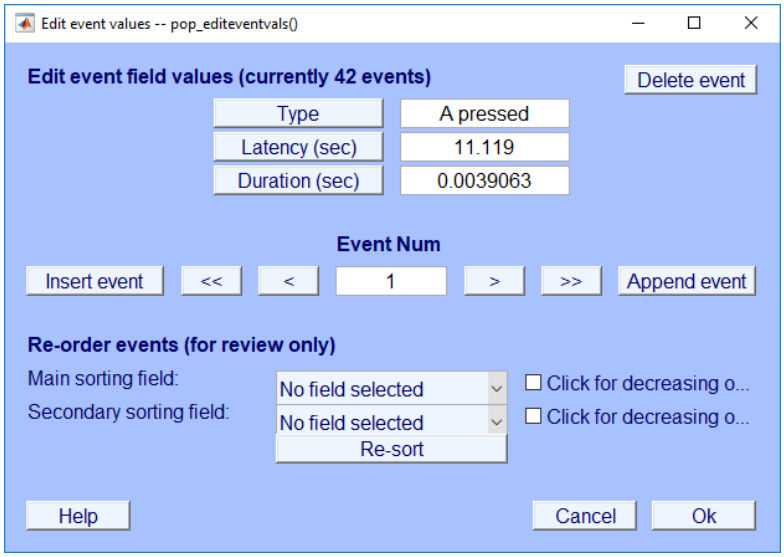


5. Re-referencing
   5.1. Tools → Re-reference | Compute average reference. Rename the dataset as 're-ref'
   5.2. Note that datasets are listed in the 'Datasets' tab



   5.3. Observe the time domain plots of raw and re-ref datasets and explain the effect of re-referencing. Normalize data if variance of amplitudes extensive (press the 'Norm' button).

6. Event renaming
   6.1. Edit → Event values



   6.2. Delete all the 'released' type of events (redundancy of event information).
   6.3. Rename all the 'pressed' type of events to corresponding meaningful events as listed in Table 2 of the 'Laboratory exercise 2 (EEG g.tec) – procedure.pdf'.
   6.4. Visualise the time domain plot with renamed events.

7. Segmenting data related to events
   7.1. Method 1: based on events
      7.1.1. Edit → Select data using events

7.1.2. Select an event (e.g. Blink)

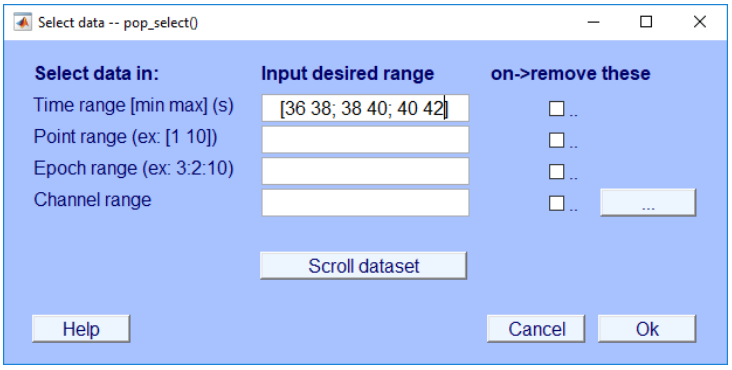
7.1.3. Set the start time limit to -0.5 from the event and enter an appropriate duration for the end time limit by manually observing a duration of a blink.

7.1.4. Rename the new data set as 'blinks_trial'. Observe the time domain plot and identify any issues.

7.2. Method 2: based on time

7.2.1. Edit → Select data

7.2.2. Use the time domain plot to identify start and end time points of each type of event and enter them in the desired range. Below figure shows the time points of the first three blinks.

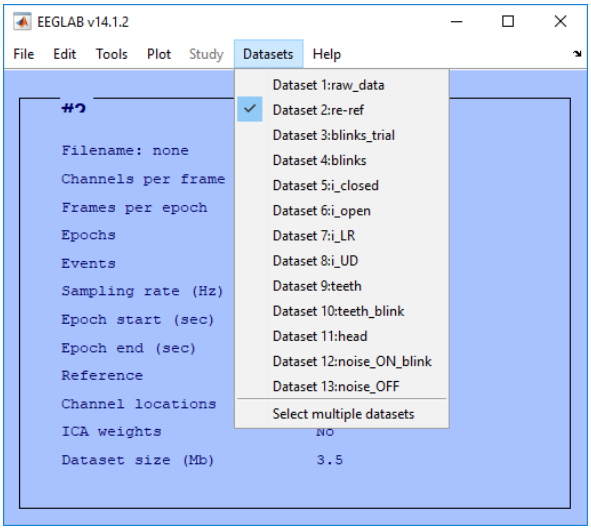


7.2.3. Save the new dataset as 'only_blinks'. Observe the new dataset in the time domain and compare with section 7.1.4.

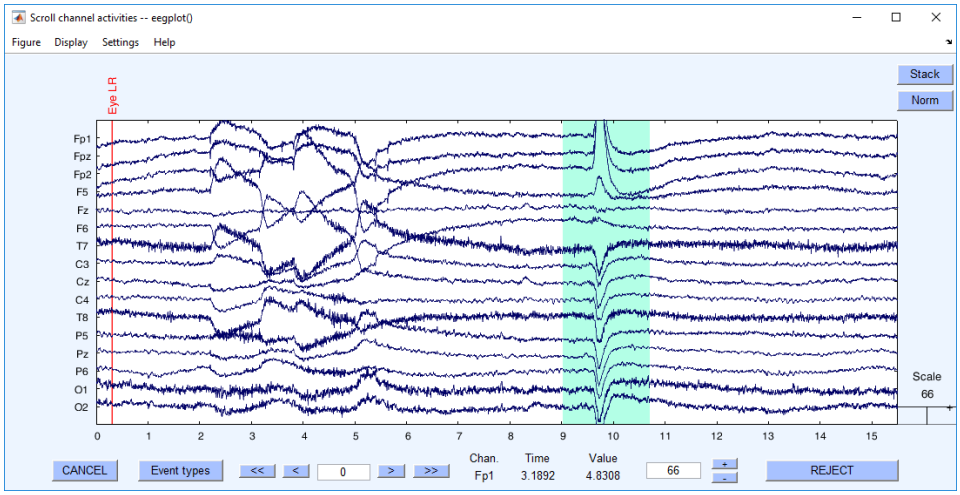
7.2.4. Using this method, create new datasets for each event type. All the datasets should be listed as per the figure below.

Note1: create separate datasets for simultaneous events (e.g. teeth clenching and blinking, noise source on and blinking).

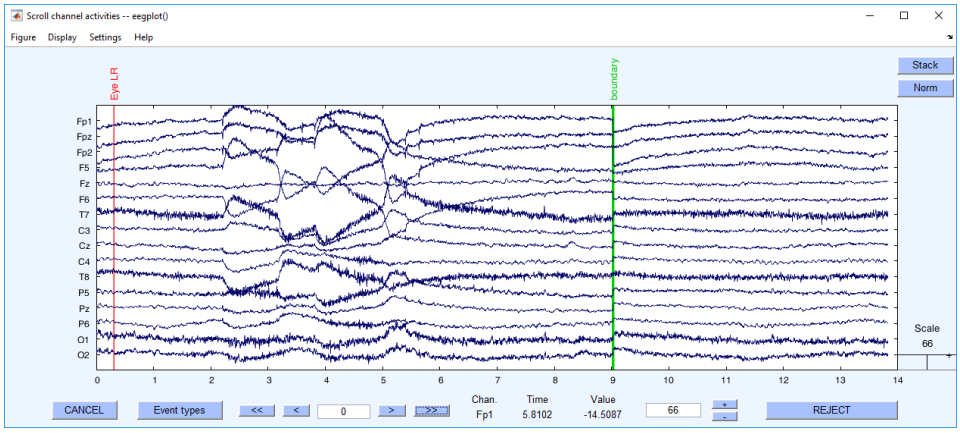
Note2: when creating new datasets, always select the re-ref dataset before executing section 7.2.1



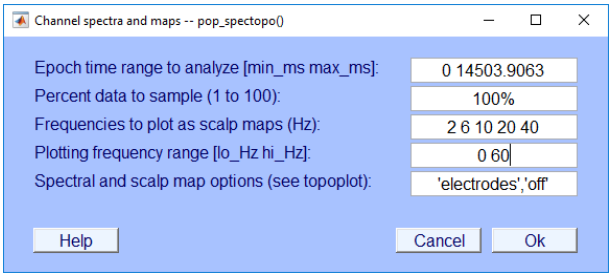
8. Saving datasets
   Each dataset should be saved separately. Select the required dataset from the 'Datasets' tab then, File → Save current dataset as.
9. Manual rejection of artifacts
   There may be artifacts within the datasets that you have created in the earlier step. For example, see the figure below. To manually remove such artifacts follow the following procedure.
   9.1. In the time domain plot, select (by clicking and dragging the cursor across) the range you want to reject (show in green) and press the 'Reject' button. You may override the dataset name.

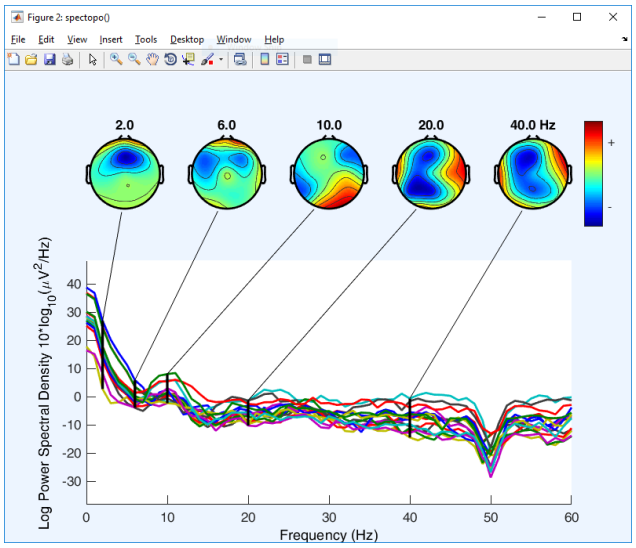


   9.2. Once the artifact is rejected, the time domain plot is shown in the figure below.



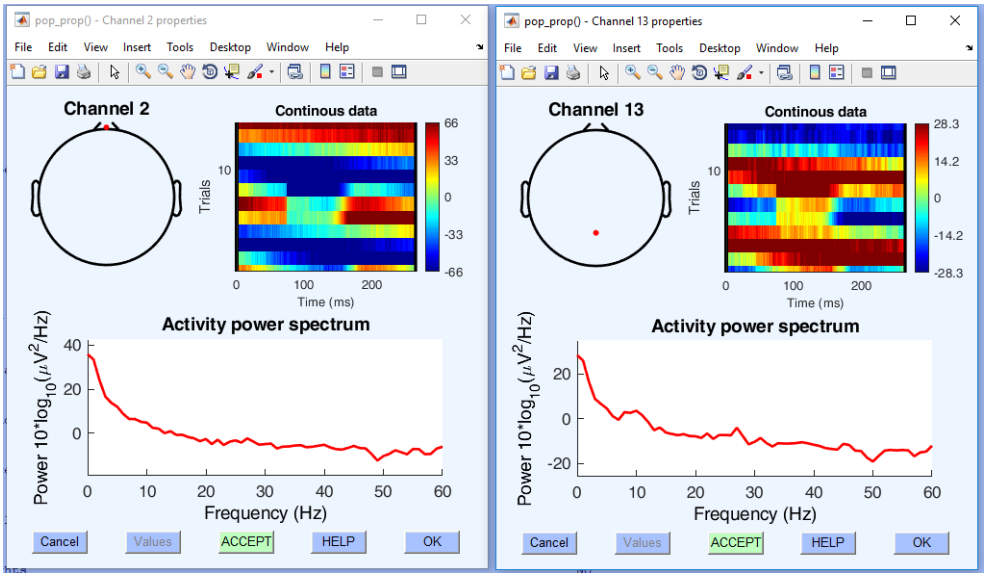
10. Channel spectra and scalp maps
    10.1. Select a dataset (e.g. i_closed)
    10.2. Plot → Channel spectra and maps
    10.3. Use all the data points to generate the spectrum (100%). Plot scalp maps at 2, 6, 10, 20, 40 Hz to investigate localization of spectral power. Plot the spectrum for a frequency range of 0 - 60 Hz.





10.4. To observe behaviour of specific channels: Plot → Channel properties

10.5. Include a channel at the anterior and posterior with a frequency range from 0 – 60



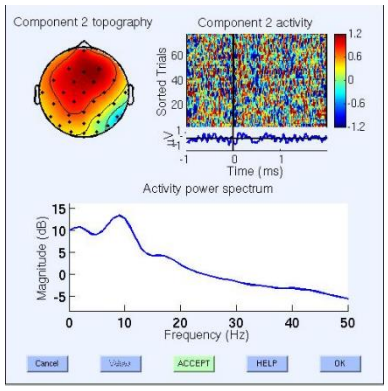
10.6. Interpret the conclusion that can be derived from both these types of plots for following events: i_closed, i_LR, i_UP, teeth, noise_ON without blinks. You may change any parameter except the frequency range (0-60 Hz) and percentage of data (100%).

Part 3: Using independent component analysis (ICA) of EEG data

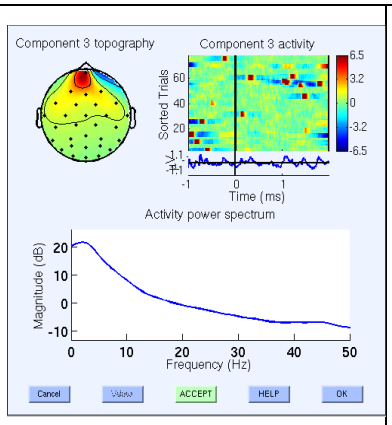
Introduction

Certain artefacts such as eye movements overlap with the frequency range of the signal of interest. Therefore, frequency domain filtering may not yield successful noise reduction. Independent Component Analysis (ICA) can be used to isolate such artefact components. Artefact components can be distinguished from EEG components using the following characteristics in general.

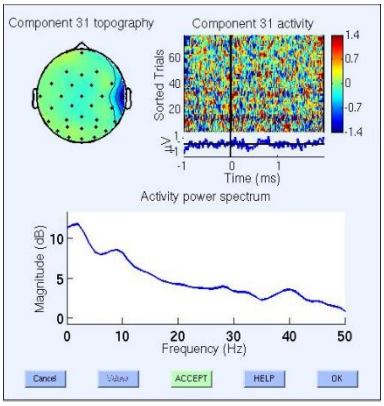
- Topography – EEG components have a generally distributed topography with balanced polarity, artefacts are usually localised (see figure below: example – eye movement components are localized in the front, muscle artefact components are localized close to the neck and ears, poor electrode connections appear as highly localized components)
- Characteristic of the frequency spectrum - 8 to 13 Hz activity is generally significant in EEG components, a smoothly decreasing spectrum is characteristic of eye movement artefact, high power at high frequencies (20-50Hz) are characteristic of muscle movement artefacts, high peaks at 50Hz are characteristic of power line noise (see figure below).
- Characteristic of the time domain signal – Eye movement components consist of large peaks in the mV range, sinusoidal components of high frequency are characteristic of power line noise (see figure below).



Brain-related component properties



Component of eye movement artefact



Component of muscle movement artefact

Note: Use the following tutorial as a guideline for ICA in EEGLAB.
https://sccn.ucsd.edu/wiki/Chapter_09:_Decomposing_Data_Using_ICA

11. ICA in EEGLAB

11.1. Select the 'blinks' dataset.

11.2. Perform ICA: Tools → Run ICA.

11.3. To visualize the time domain independent components, select Plot → Component activation (scroll). Identify which components could be eye movement components? Why?

11.4. To visualize the spectrum and topography of independent components, select Plot → Component properties. Select all 16 channels (1:16). From the 16 resulting output windows, observe both the topography and the spectrum.

11.5. Giving reasons, categorize these components into EEG and artefact (eye movement, muscle movement, power line noise, channel noise). You may practice this using the following website. https://labeling.ucsd.edu/tutorial/practice

11.6. Remove the identified noisy ICA components using Tools → Remove components.

11.7. Visualize the time domain channels the spectra after removing noise components. State your observations.

11.8. Perform similar analysis (step 11.1 to 11.7) on the following datasets: i_UD, i_LR, teeth_blink, noise_ON_blink, noise_OFF

12. ICA preprocessing using MARA (Extended Work)

From section 0, it can be understood that manually categorizing independent components is highly subjective and time consuming. One solution to this problem is to use a machine learning approach. MARA (Multiple Artifact Rejection Algorithm) which is an extension tool to EEGLAB is an example.

12.1. Install MARA from File → Manage EEGLAB Extensions → Data Processing Extensions.

12.2. Follow the guidelines and obtain the probabilities of each of the independent components being an artefact through MARA.
https://github.com/irenne/MARA/blob/master/MARAtutorial.pdf

12.3. Compare your results with section 0.