Signals and Systems Project

Instructor: Prof. Hamid Aghajan

Sharif University of Technology

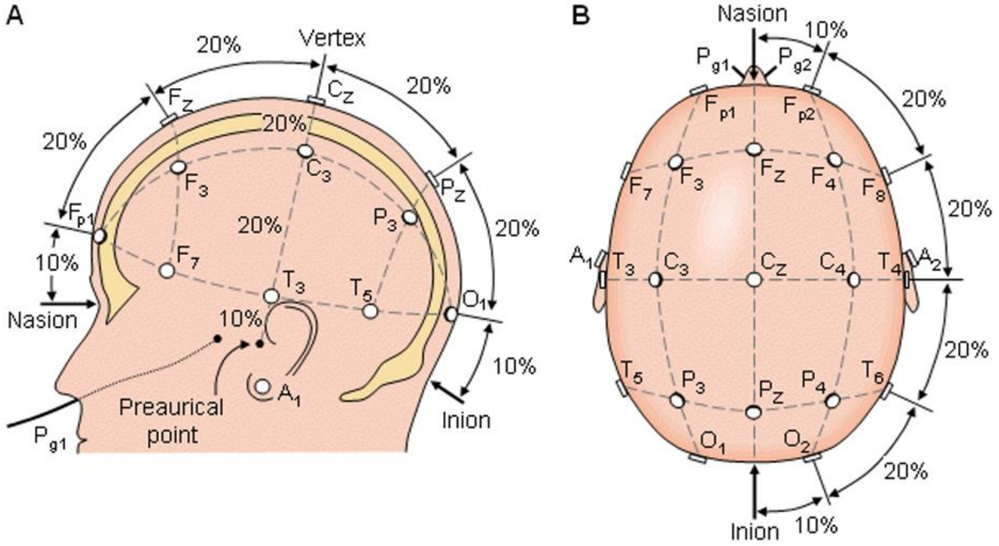
Epileptic Seizure Prediction Using Spectral Entropy-Based Features of EEG

Authors:

Alireza Gargoori Motlagh AmirReza Imani
S. Moienadin Makkiyan

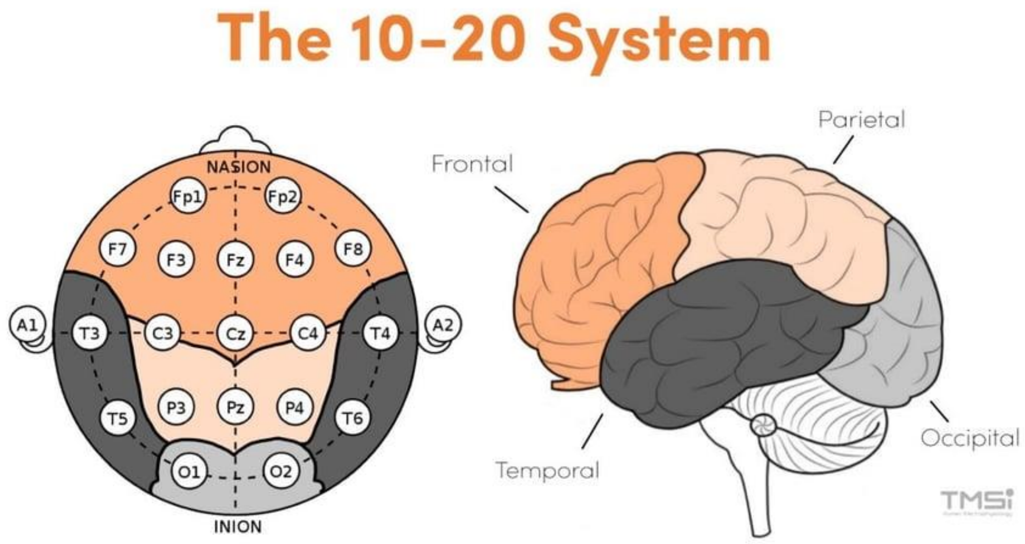
Project by:
Abbas Naimi (4011102723)

Part 1) Understanding EEG

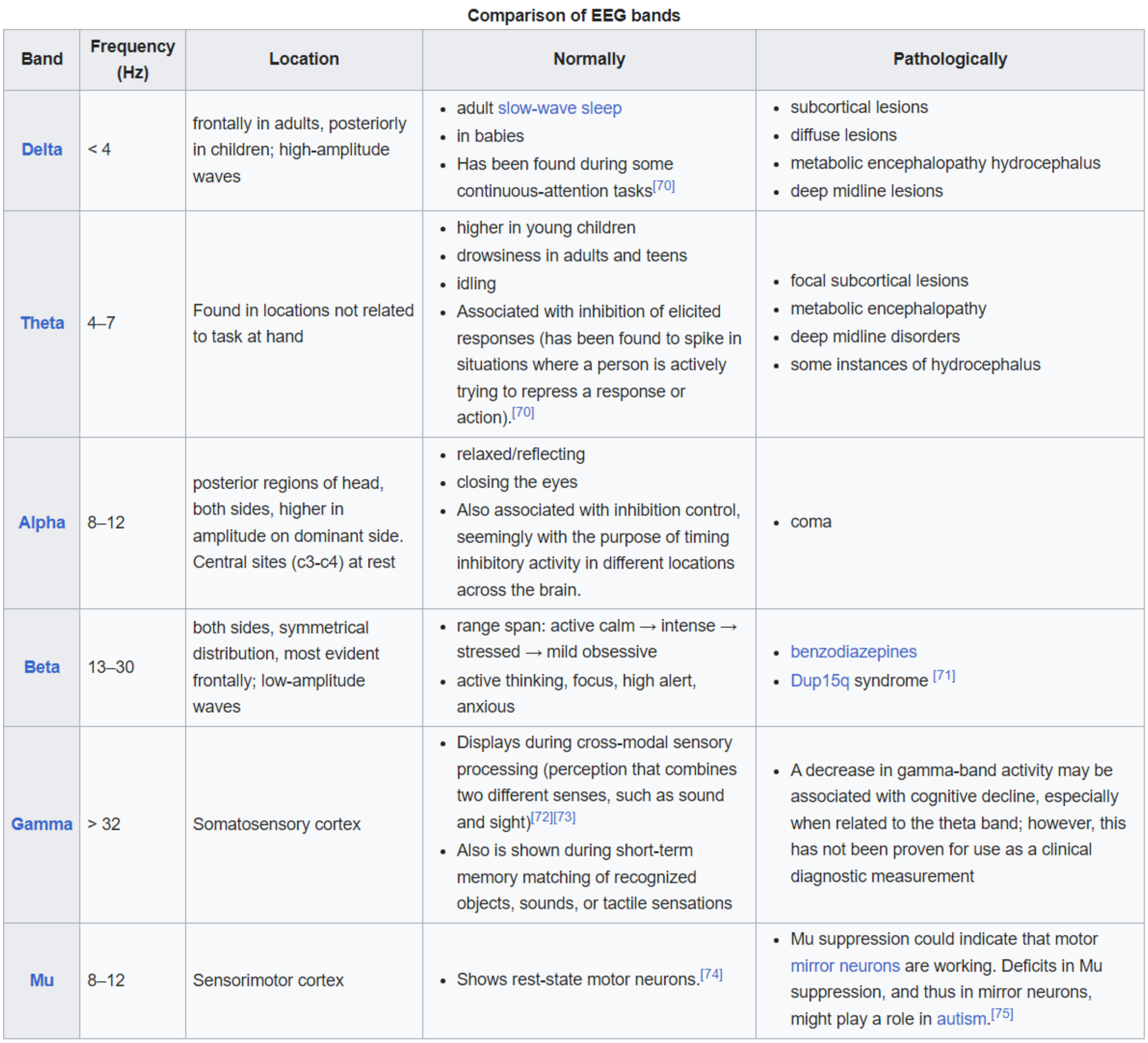


2-1) Question: Based on the picture above, what does each electrode's name stand for? Explain the naming method used in the 10-20 EEG system.

- Each electrode placement site has a letter to identify the lobe, or area of the brain it is reading from: pre-frontal (Fp), frontal (F), temporal (T), parietal (P), occipital (O), and central (C). Note that there is no "central lobe"; due to their placement, and depending on the individual, the "C" electrodes can exhibit/represent EEG activity more typical of frontal, temporal, and some parietal-occipital activity, and are always utilized in polysomnography sleep studies for the purpose of determining stages of sleep. There are also Z points, these are electrodes placed on the midline of the skull (FpZ, Fz, Cz, Oz). They're mainly used for reference or measurement points. They don't necessarily reflect or amplify activity from either side of the brain as they're placed over the corpus callosum, a part of the brain that connects the two hemispheres. They're often used as 'grounds' or 'references' in sleep studies and EEGs meant to diagnose seizures or brain death. The last set of electrodes visible in the image are A, these refer to the prominent bone usually found just behind the outer ear. In basic sleep studies, F3, F4, Fz, Cz, C3, C4, O1, O2, A1, A2 (M1, M2), are used. Cz and Fz are 'ground' or 'common' reference points for all EEG and EOG electrodes, and A1-A2 are used for referencing all EEG electrodes from the opposite side.



2-3) Determine the activities each frequency band is associated with.



Comparison of EEG bands

Band	Frequency (Hz)	Location	Normally	Pathologically
Delta	< 4	frontally in adults, posteriorly in children; high-amplitude waves	• adult slow-wave sleep • in babies • Has been found during some continuous-attention tasks[70]	• subcortical lesions • diffuse lesions • metabolic encephalopathy hydrocephalus • deep midline lesions
Theta	4–7	Found in locations not related to task at hand	• higher in young children • drowsiness in adults and teens • idling • Associated with inhibition of elicited responses (has been found to spike in situations where a person is actively trying to repress a response or action).[70]	• focal subcortical lesions • metabolic encephalopathy • deep midline disorders • some instances of hydrocephalus
Alpha	8–12	posterior regions of head, both sides, higher in amplitude on dominant side. Central sites (c3-c4) at rest	• relaxed/reflecting • closing the eyes • Also associated with inhibition control, seemingly with the purpose of timing inhibitory activity in different locations across the brain.	• coma
Beta	13–30	both sides, symmetrical distribution, most evident frontally; low-amplitude waves	• range span: active calm → intense → stressed → mild obsessive • active thinking, focus, high alert, anxious	• benzodiazepines • Dup15q syndrome [71]
Gamma	> 32	Somatosensory cortex	• Displays during cross-modal sensory processing (perception that combines two different senses, such as sound and sight)[72][73] • Also is shown during short-term memory matching of recognized objects, sounds, or tactile sensations	• A decrease in gamma-band activity may be associated with cognitive decline, especially when related to the theta band; however, this has not been proven for use as a clinical diagnostic measurement
Mu	8–12	Sensorimotor cortex	• Shows rest-state motor neurons.[74]	• Mu suppression could indicate that motor mirror neurons are working. Deficits in Mu suppression, and thus in mirror neurons, might play a role in autism.[75]

2-4) Based on frequency bands and Nyquist criterion, which sampling frequencies are preferred for EEG signals?

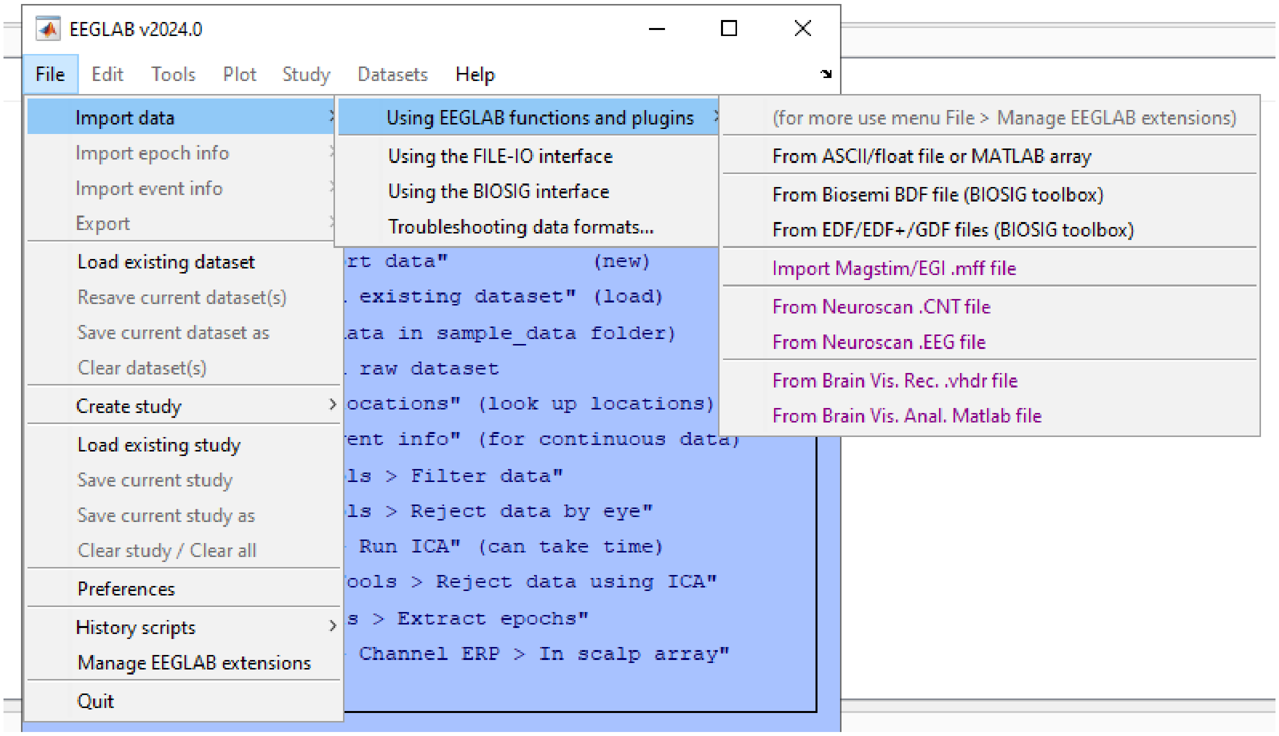
in previous parts we discovered that there are different ranges of frequencies in EEG signals but the maximum is about 100Hz. Higher sampling requires more storage to record the data, more expensive equipment to operate at a higher clock rate and etc., so we need to make sure we are setting a sampling frequency that is not too low so aliasing happens but not too high so the higher frequencies remain unused. Simply taking 100 Hz as max frequency Nyquist criterion states we need at least a sampling frequency of 200 Hz, so the 256 Hz given in the instructions has a safe margin and isn't too high either.

Part 2) EEG Signal processing with eeglab:

From this point on we will be using eeglab matlab toolbox to process the dataset which was provided for this phase. Eeglab uses an HTML GUI but the code for each part can be accessed using the code history in the GUI.

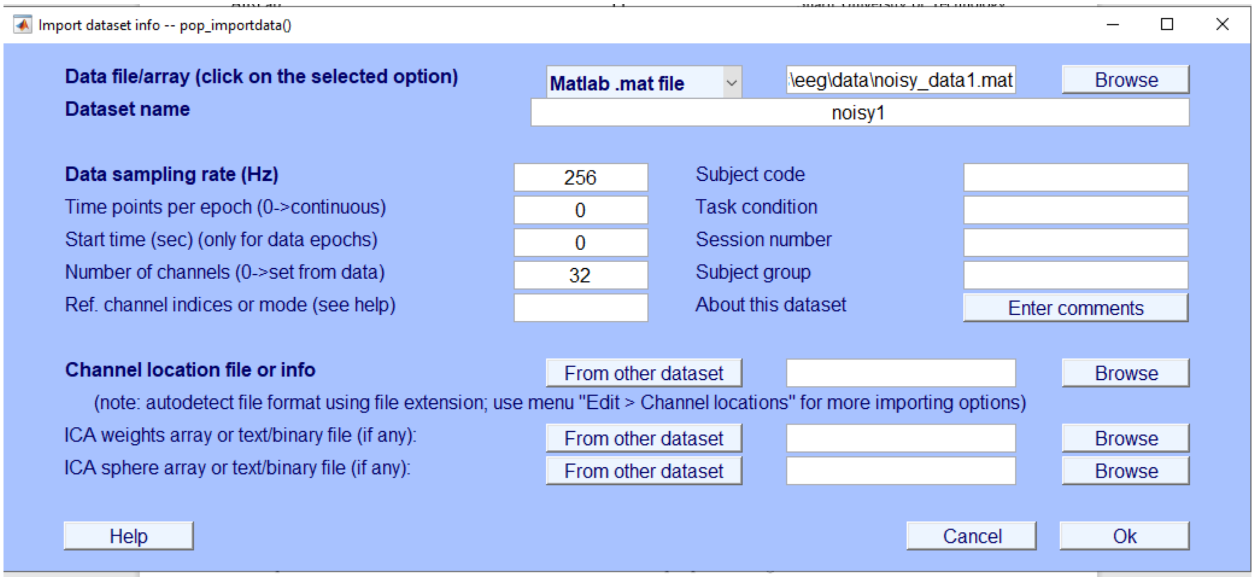
4.3.1) loading data:

Datasets are provided as two separate .mat files which contain the EEG results in a matlab struct. Eeglab natively supports importing matlab files so we use the utility:


EEGLAB v2024.0
File
Edit
Tools
Plot
Study
Datasets
Help
Import data
Import epoch info
Import event info
Export
Load existing dataset
Resave current dataset(s)
Save current dataset as
Clear dataset(s)
Create study
Load existing study
Save current study
Save current study as
Clear study / Clear all
Preferences
History scripts
Manage EEGLAB extensions
Quit
Using EEGLAB functions and plugins
Using the FILE-IO interface
Using the BIOSIG interface
Troubleshooting data formats...
(for more use menu File > Manage EEGLAB extensions)
From ASCII/float file or MATLAB array
From Biosemi BDF file (BIOSIG toolbox)
From EDF/EDF+/GDF files (BIOSIG toolbox)
Import Magstim/EGI .mff file
From Neuroscan .CNT file
From Neuroscan .EEG file
From Brain Vis. Rec. .vhdr file
From Brain Vis. Anal. Matlab file

```
rt data"          (new)
. existing dataset" (load)
ata in sample_data folder)
. raw dataset
ocations" (look up locations)
ent info" (for continuous data)
ls > Filter data"
ls > Reject data by eye"
 Run ICA" (can take time)
ools > Reject data using ICA"
s > Extract epochs"
 Channel ERP > In scalp array"
```

Information about the tests were provided such as sampling rate and number of channels so we input them here:



Import dataset info -- pop_importdata()

Data file/array (click on the selected option): Matlab .mat file — \eeg\data\noisy_data1.mat

Dataset name: noisy1

Data sampling rate (Hz): 256

Time points per epoch (0->continuous): 0

Start time (sec) (only for data epochs): 0

Number of channels (0->set from data): 32

Ref. channel indices or mode (see help):

Subject code:

Task condition:

Session number:

Subject group:

About this dataset: Enter comments

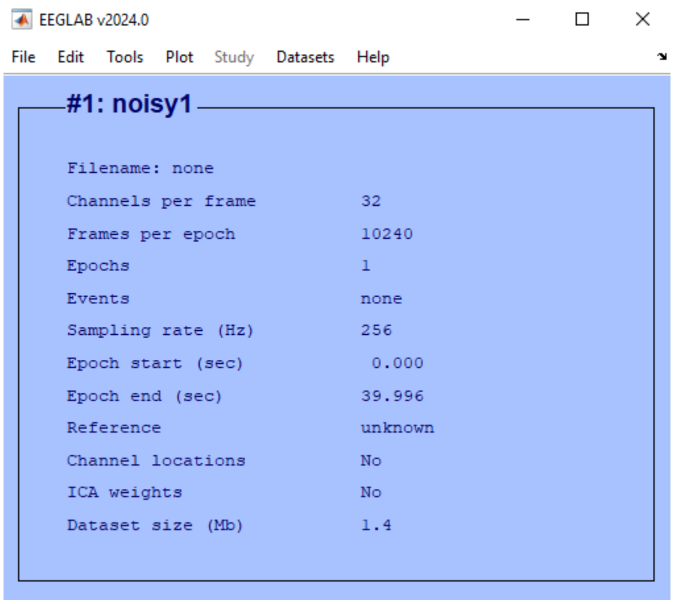
Channel location file or info: From other dataset

(note: autodetect file format using file extension; use menu "Edit > Channel locations" for more importing options)

ICA weights array or text/binary file (if any): From other dataset

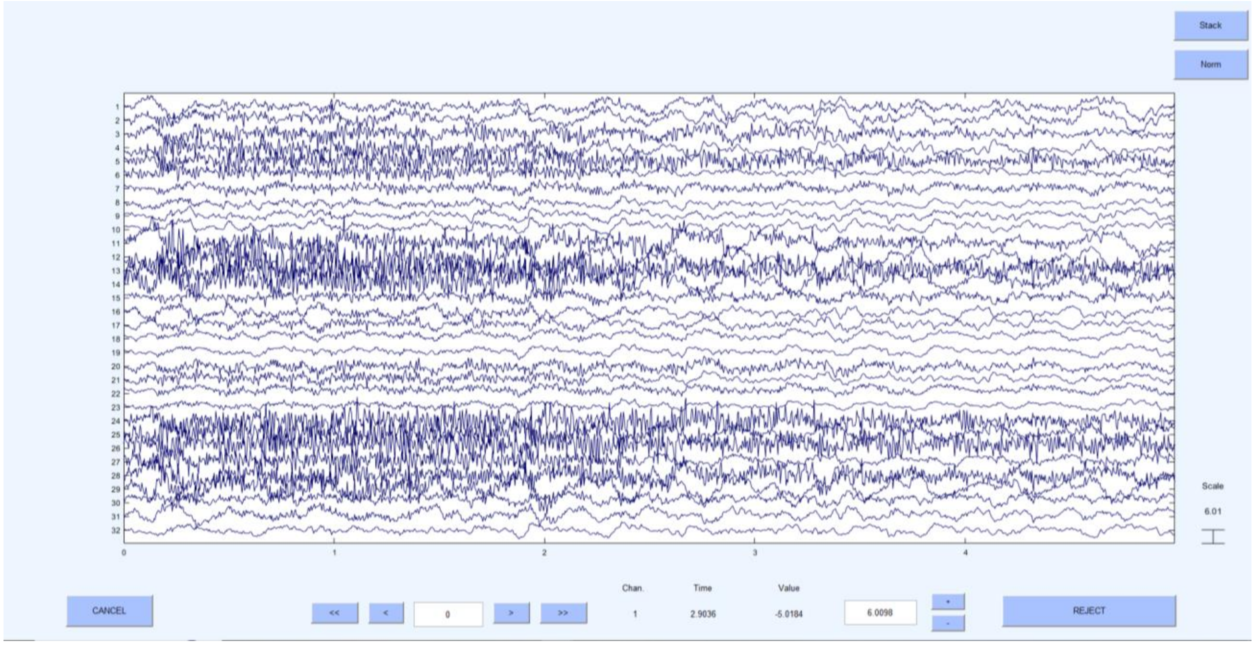
ICA sphere array or text/binary file (if any): From other dataset

After completing the steps above we have imported a raw matlab struct into eeglab but in order to understand and analyze the data we need to know where each channel was placed and what data was recorded. Right now the channels only have indexes:

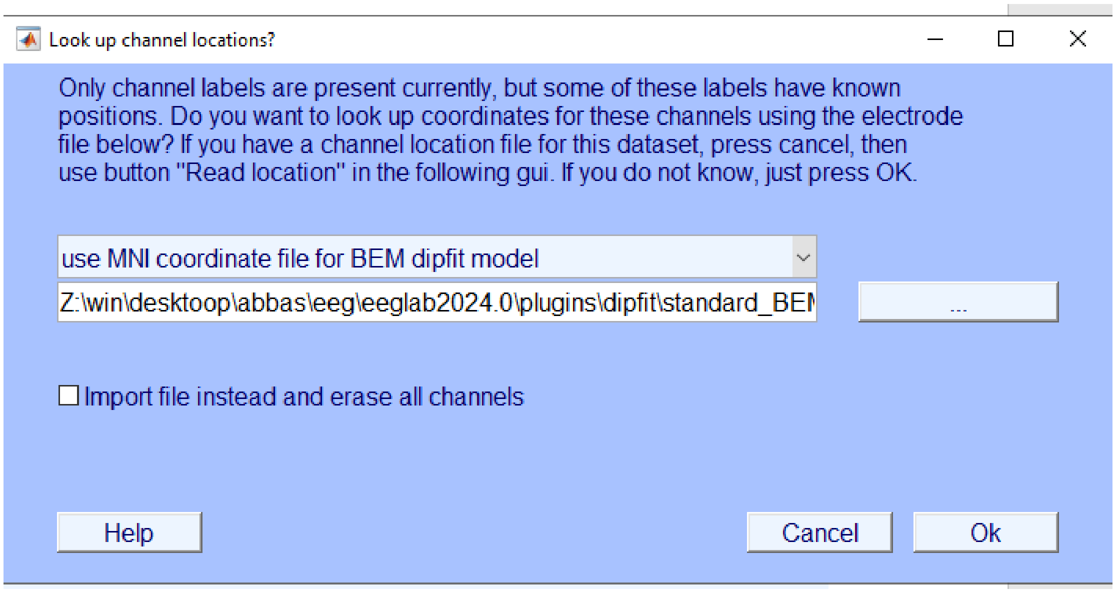


#1: noisy1

```
Filename: none
Channels per frame          32
Frames per epoch            10240
Epochs                      1
Events                      none
Sampling rate (Hz)          256
Epoch start (sec)            0.000
Epoch end (sec)             39.996
Reference                   unknown
Channel locations           No
ICA weights                 No
Dataset size (Mb)           1.4
```



So as the next step we import the location data which was provided to make our data complete:



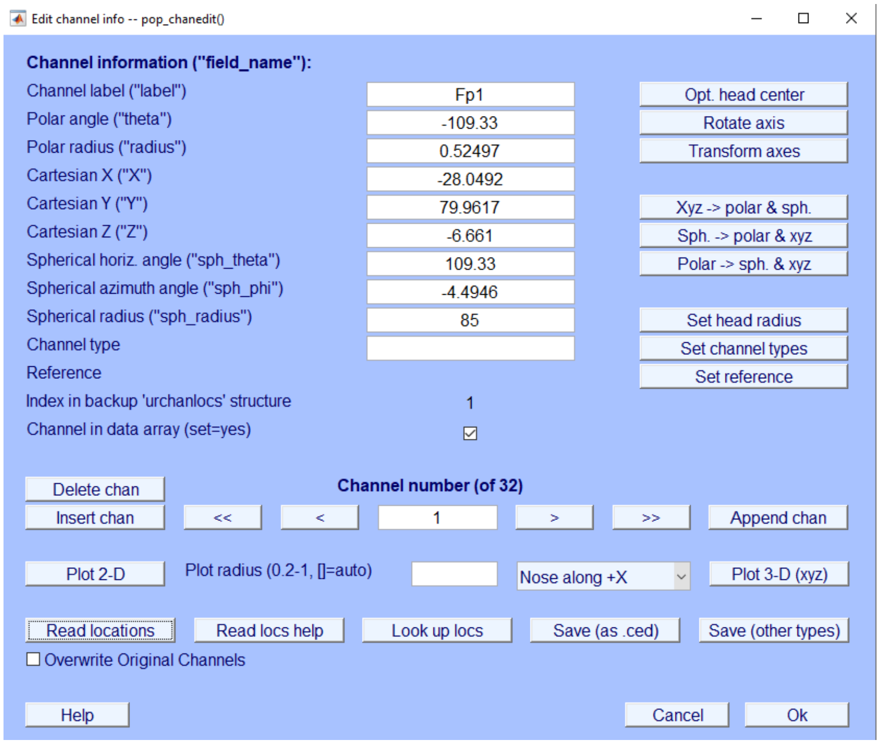
Only channel labels are present currently, but some of these labels have known positions. Do you want to look up coordinates for these channels using the electrode file below? If you have a channel location file for this dataset, press cancel, then use button "Read location" in the following gui. If you do not know, just press OK.

use MNI coordinate file for BEM dipfit model

Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\dipfit\standard_BEI

☐ Import file instead and erase all channels

There are two options to use, one is using one of preset channel location datasets that are built into eeglab and the other one is using our own location file. The location file is provided in this project so we pressed “no” in the above dialog and in the following menu imported our own data:


Edit channel info -- pop_chanedit()

Channel information ("field_name"):

Channel label ("label") Fp1

Polar angle ("theta") -109.33

Polar radius ("radius") 0.52497

Cartesian X ("X") -28.0492

Cartesian Y ("Y") 79.9617

Cartesian Z ("Z") -6.661

Spherical horiz. angle ("sph_theta") 109.33

Spherical azimuth angle ("sph_phi") -4.4946

Spherical radius ("sph_radius") 85

Channel type

Reference

Index in backup 'urchanlocs' structure 1

Channel in data array (set=yes) ☑

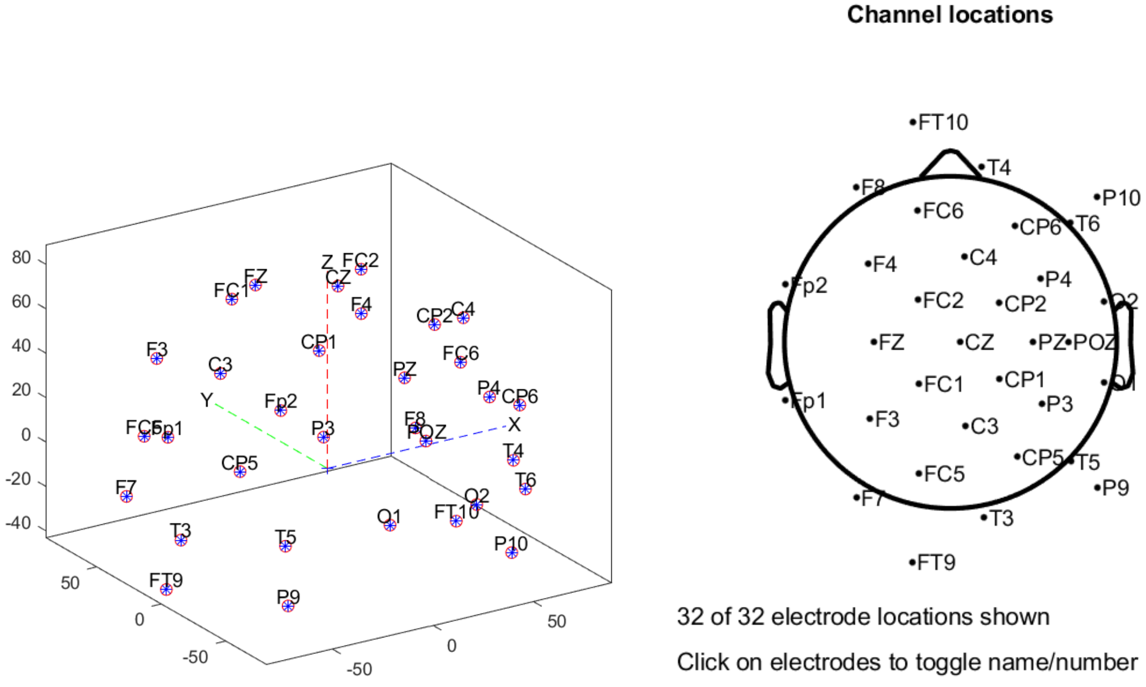
Channel number (of 32) 1

Plot radius (0.2-1, []=auto)

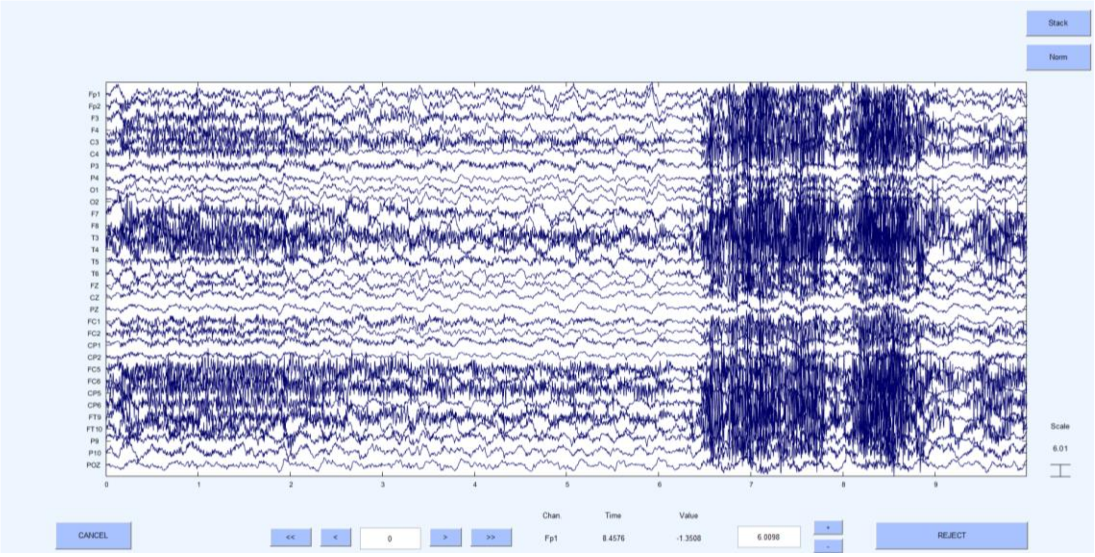
Nose along +X

□ Overwrite Original Channels

Inspecting the location file:

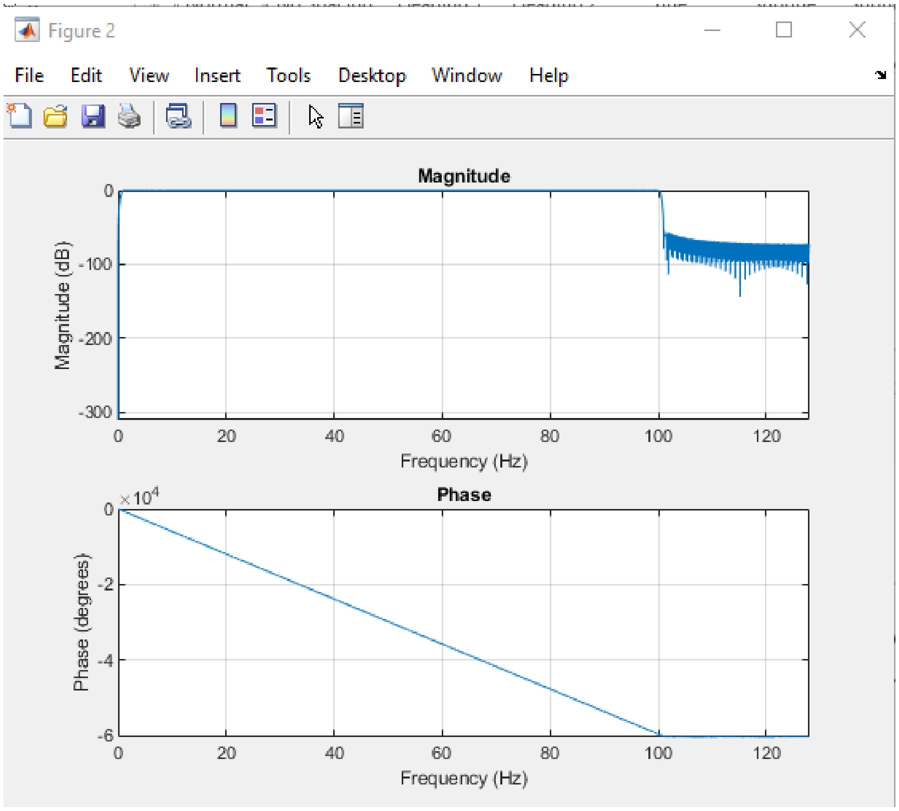


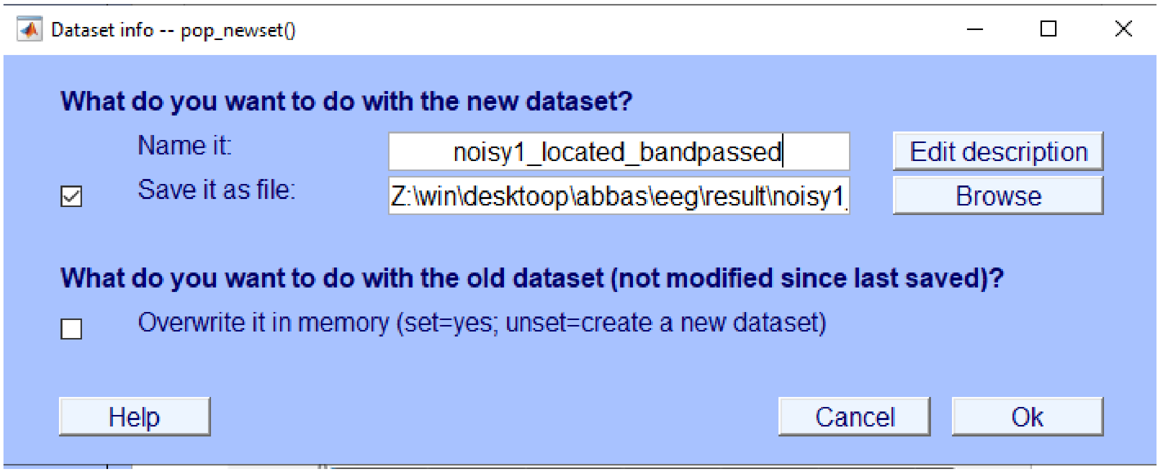
By completing this step, we can see the data now uses tags to indicate the location of each channel:



Chan. Fp1
Time 8.4576
Value -1.3508

At this point we start to preprocess the data to reduce noise, remove non brain signal and etc. the first step here is to use a 1hz highpass filter to remove very low frequency changes in the data. These changes can be caused by many factors such as head movements, electrode movement and other changes that happened during the EEG session but are not meant to be studied here. At the same time as we discussed earlier in this file we are not expecting signals above 100 Hz in our brain data so we set the higher cutoff frequency as 100 Hz to remove any higher frequency from the data. The final filter is provided below:





What do you want to do with the new dataset?

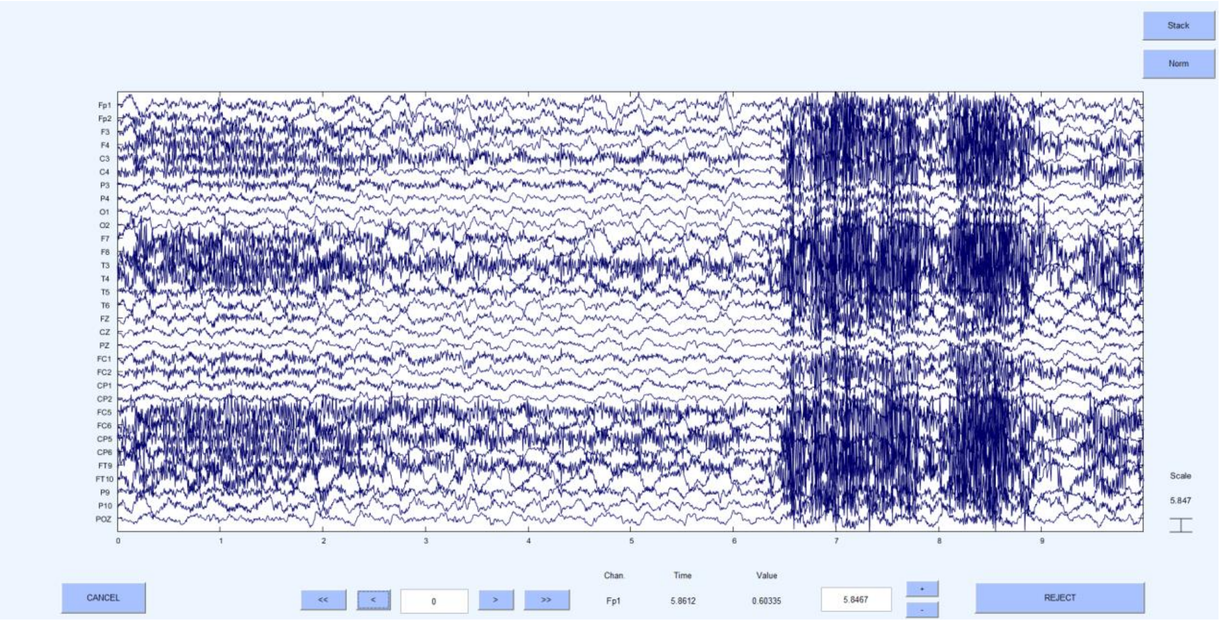
Name it: noisy1_located_bandpassed

☑ Save it as file: Z:\win\desktoop\abbas\eeg\result\noisy1

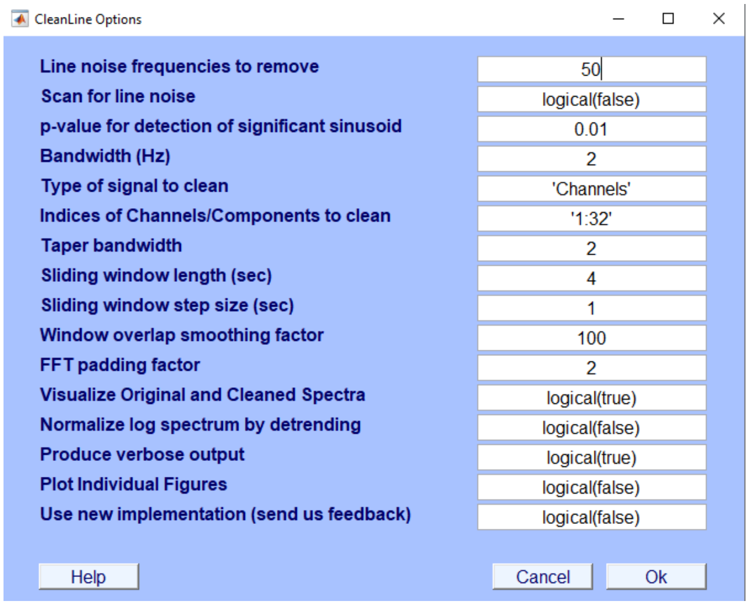
What do you want to do with the old dataset (not modified since last saved)?

☐ Overwrite it in memory (set=yes; unset=create a new dataset)

Now a new copy of our dataset is created with the filter applied. We can plot the new dataset to check if we can spot any difference from the original data:

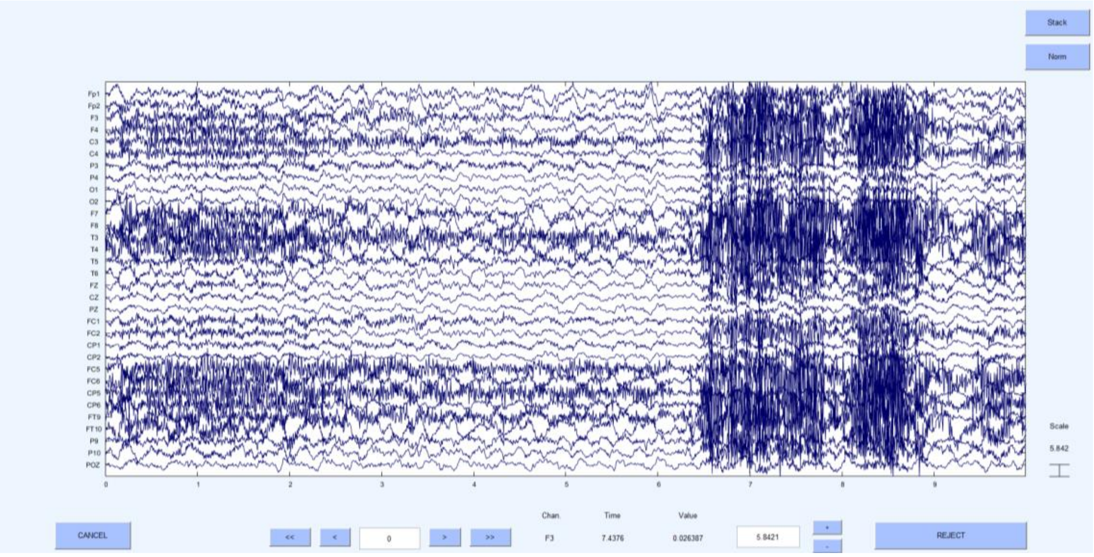


After removing the unrelated frequencies, we need to remove power supply noise at 50Hz too. There are many different ways to do so, because 50 Hz is a frequency which contains useful information so it’s ideal to find a way to remove only 50 Hz supply but keep 50Hz signal. As indicated by the instructions we are not going to go that deep and use a simple notch filer for 50Hz, but in the tutorial videos we there introduced to a plugin that could give better results that simple notch filter. Using the Cleanline utility:



Line noise frequencies to remove	50
Scan for line noise	logical(false)
p-value for detection of significant sinusoid	0.01
Bandwidth (Hz)	2
Type of signal to clean	'Channels'
Indices of Channels/Components to clean	'1:32'
Taper bandwidth	2
Sliding window length (sec)	4
Sliding window step size (sec)	1
Window overlap smoothing factor	100
FFT padding factor	2
Visualize Original and Cleaned Spectra	logical(true)
Normalize log spectrum by detrending	logical(false)
Produce verbose output	logical(true)
Plot Individual Figures	logical(false)
Use new implementation (send us feedback)	logical(false)

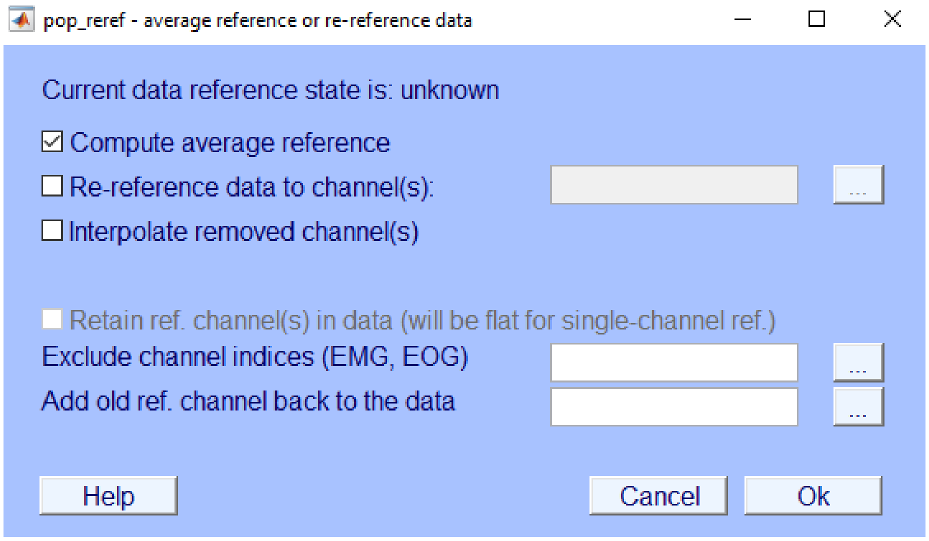
Plotting the results:



Chan. F3
Time 7.4376
Value 0.026387

The difference is minimal but when checked side by side we can see a strong harmonic has been removed.

After saving the results in a separate dataset now we have blocked the frequencies that are unrelated for sure. As for the next step we need to filter noise in the accepted frequency range. One easy but effective step is referencing the data to average of all channels. It was explained in the first part that there are different methods for capturing channel data like referencing all voltages to some electrodes that are used for reference (locations that contain less brain signal), or checking voltage of each electrode with respect to the next electrode. Each method has its own advantages but in order to remove noises that are not part of brain activity re-referencing to channels average is an effective method that is computationally easy.

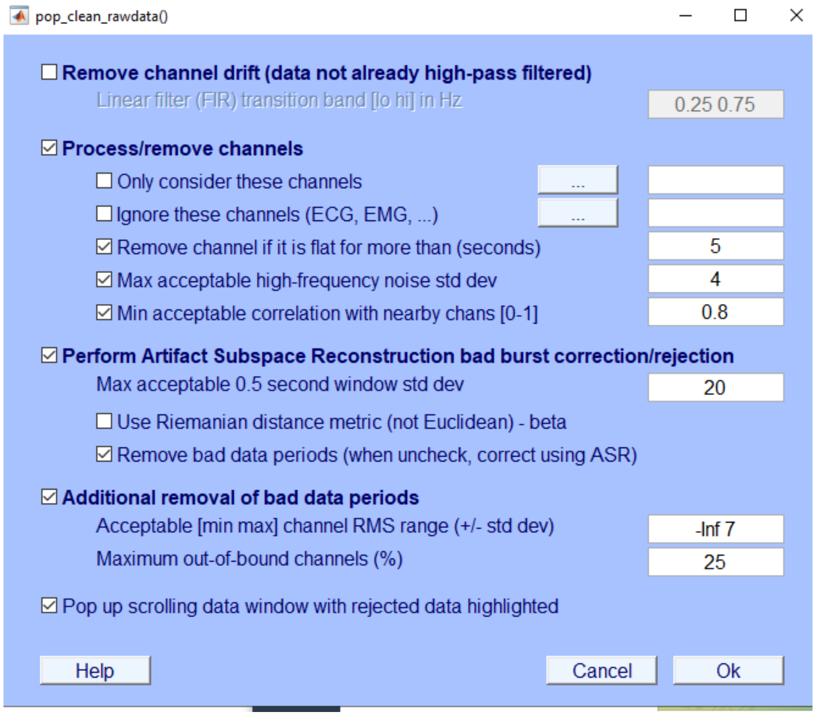

pop_reref - average reference or re-reference data

Current data reference state is: unknown

☑ Compute average reference
☐ Re-reference data to channel(s):
☐ Interpolate removed channel(s)

☐ Retain ref. channel(s) in data (will be flat for single-channel ref.)
Exclude channel indices (EMG, EOG)
Add old ref. channel back to the data

And then we use 'clean rawdata()' to remove the parts that are clearly not brain signal.


pop_clean_rawdata()

☐ **Remove channel drift (data not already high-pass filtered)**
Linear filter (FIR) transition band [lo hi] in Hz: 0.25 0.75

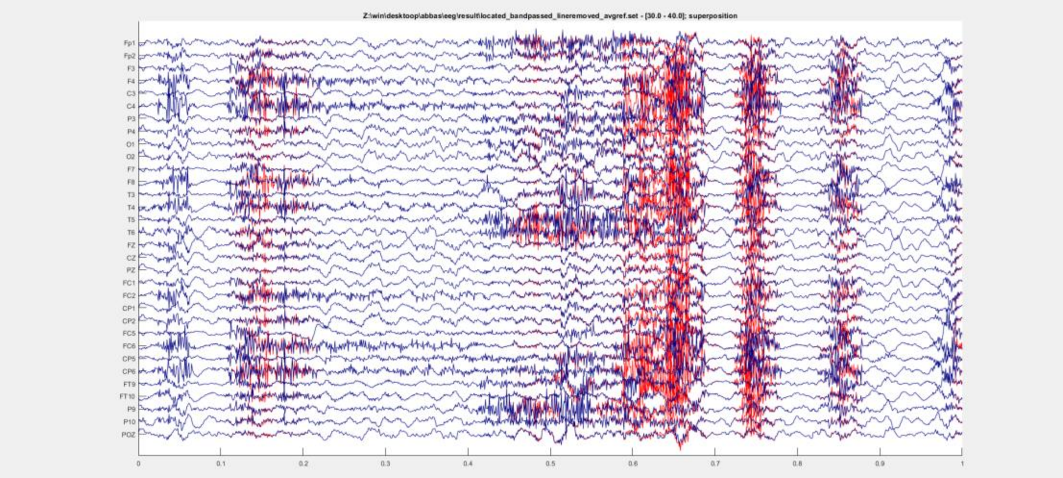
☑ **Process/remove channels**
☐ Only consider these channels
☐ Ignore these channels (ECG, EMG, ...)
☑ Remove channel if it is flat for more than (seconds): 5
☑ Max acceptable high-frequency noise std dev: 4
☑ Min acceptable correlation with nearby chans [0-1]: 0.8

☑ **Perform Artifact Subspace Reconstruction bad burst correction/rejection**
Max acceptable 0.5 second window std dev: 20
☐ Use Riemanian distance metric (not Euclidean) - beta
☑ Remove bad data periods (when uncheck, correct using ASR)

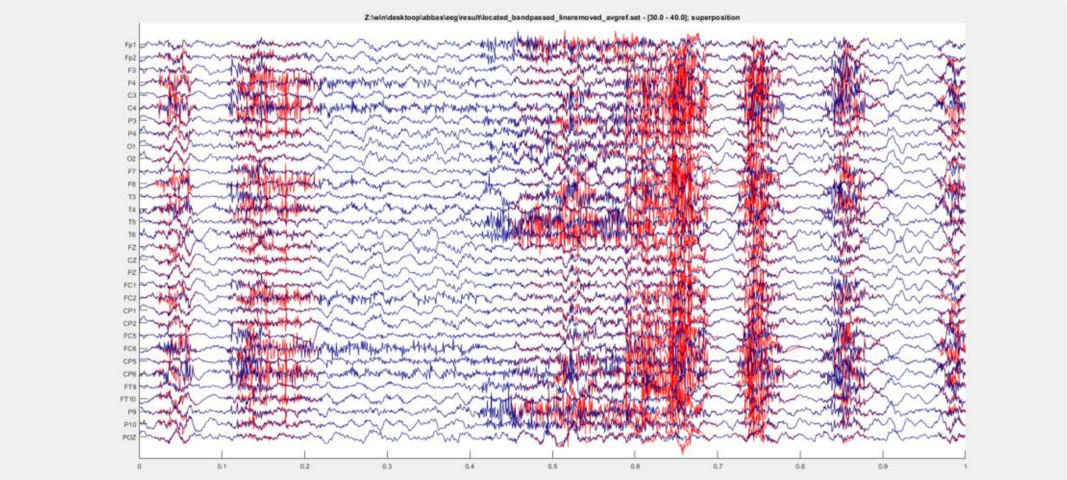
☑ **Additional removal of bad data periods**
Acceptable [min max] channel RMS range (+/- std dev): -Inf 7
Maximum out-of-bound channels (%): 25

☑ Pop up scrolling data window with rejected data highlighted

With this method there are different techniques used for removing noise. One is highpass filtering to remove channel drift which is the low frequency changes that we discussed and removed using a simple filter. The second option is removing channels that meet certain criteria like being flat for a period of time or having no correlation with other channels that are located near them. The instructions didn’t mention about using this section but if we wanted, we could use these before re-referencing. The third part is the one we are going to use, to reduce artifacts. We can either remove the time period containing the artifact or correct it using ASR. For this part we use ASR correction. In order to find a suiting value for window STD dev we checked a few different values to find which one removes noise better but keeps the signal intact. Ideally, we need to know what type of data we are dealing with and when certain events happened so we don’t remove some important change in the data mistakenly but since we have no data about the event, we just make sure eye blinks and confirmed noise patterns are removed.

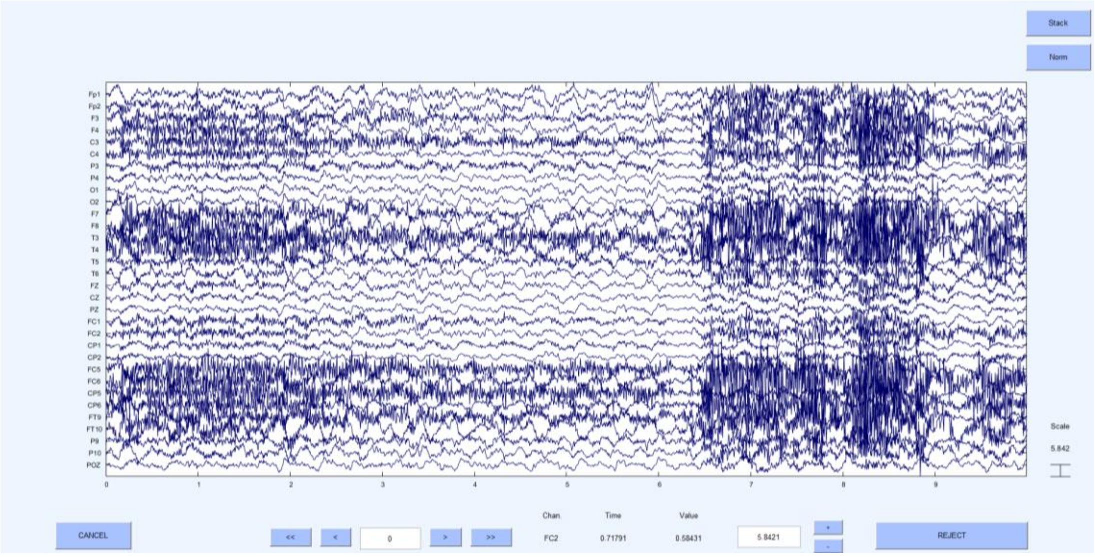


STD dev = 20, some parts are corrected but the rest is intact.

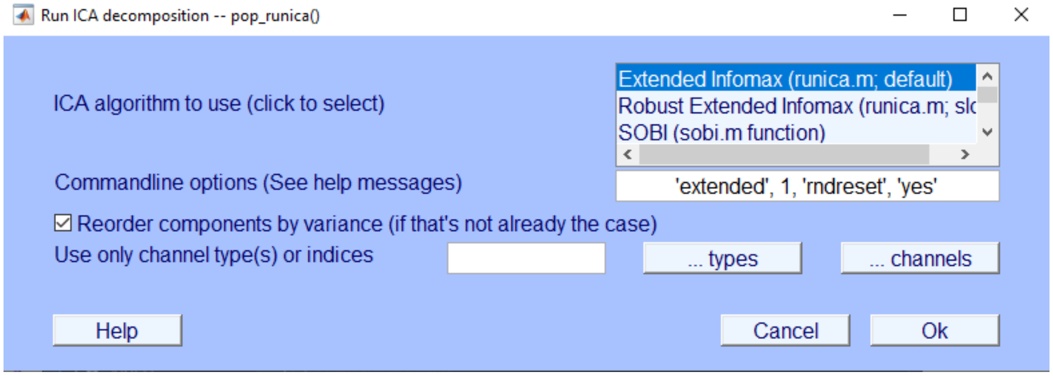


STD dev = 10, the signal is changed more. Its obscure because we might remove useful data.

Using STD dev = 20 we continue and save the data. Re-referencing again we can see a lot has changed:



At this point we have removed frequencies that were unrelated, and also reconstructed timeframes that contained noise. One problem that we need to address now is signals that are in the 1~100Hz range but their source isn't brain related like muscle movements, signals from heart and other sources. For this part we use Independent Component Analysis(ICA) which is basically a blind source separation technique to separate brain signal from other sources. This separation can be carried out by means of identifying source location, patterns and many other factors. Eeglab has functionality built in for this sort of separation so we use it:



Run ICA decomposition -- pop_runica()

ICA algorithm to use (click to select): Extended Infomax (runica.m; default) / Robust Extended Infomax (runica.m; slc / SOBI (sobi.m function)

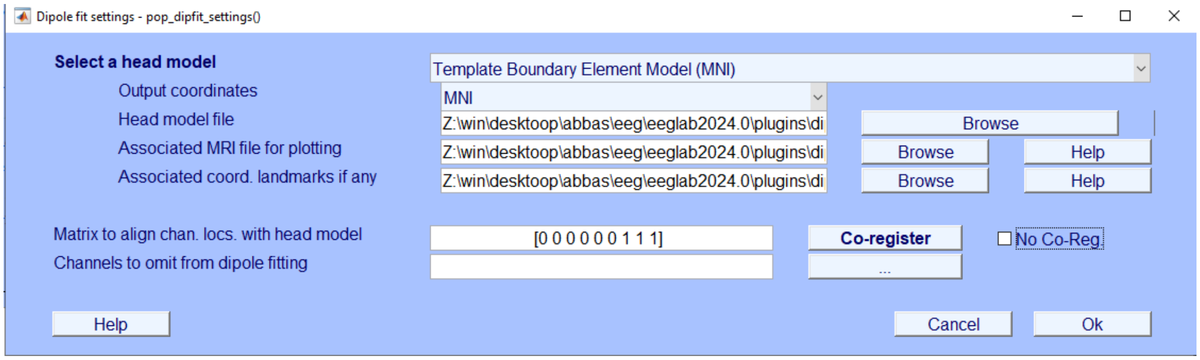
Commandline options (See help messages): 'extended', 1, 'rndreset', 'yes'

☑ Reorder components by variance (if that's not already the case)

Use only channel type(s) or indices

There are many different algorithms to use for ICA but we go with the default which is runica. At this point we might see a dialog indicating the rank of our data is less than the number of channels. This could be caused by the reconstructions we used or the re-referencing we did but anyway we can still get 31 different sources.

After this we need to set a head model to find source locations more accurately. There are different head models with different levels of accuracy. The data provided uses BESA but in the tutorial video it was indicated we can get better results using MNI so we need to co register locations from one model to another.



Dipole fit settings - pop_dipfit_settings()

Select a head model: Template Boundary Element Model (MNI)

Output coordinates: MNI

Head model file: Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\di

Associated MRI file for plotting: Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\di

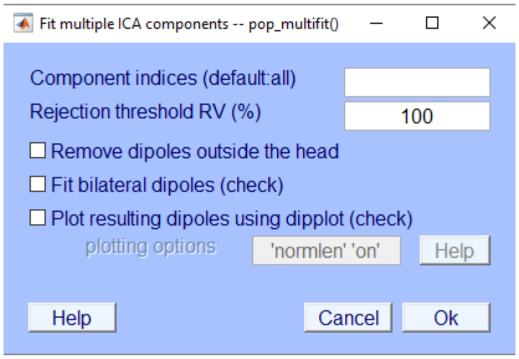
Associated coord. landmarks if any: Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\di

Matrix to align chan. locs. with head model: [0 0 0 0 0 0 1 1 1]

Channels to omit from dipole fitting

□ No Co-Reg.

After this part that the head model is set we use the automatic method to fit the sources to the locations in head to match the recorded channels. There are options to do it with more controls but we use the auto method.



Fit multiple ICA components -- pop_multifit()

Component indices (default:all)

Rejection threshold RV (%) 100

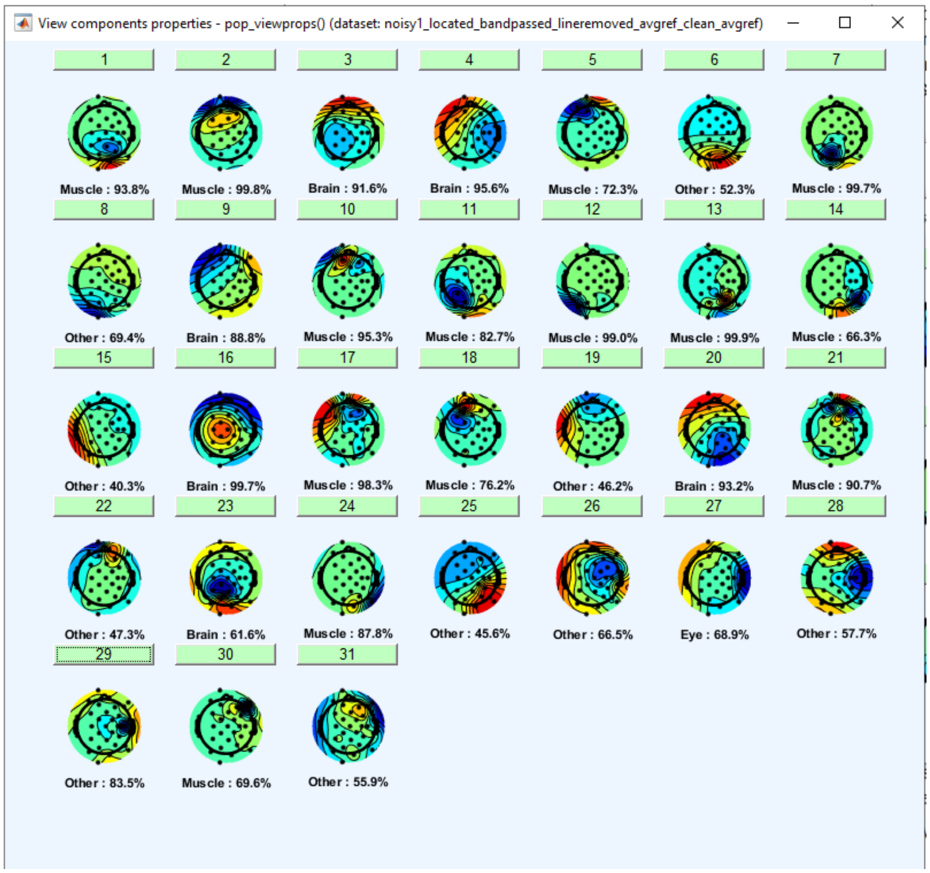
☐ Remove dipoles outside the head

☐ Fit bilateral dipoles (check)

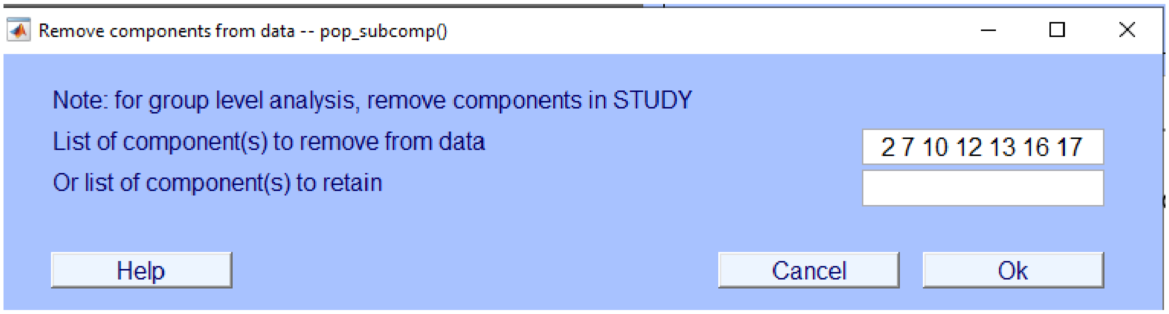
☐ Plot resulting dipoles using dipplot (check)

plotting options 'normlen' 'on'

Now that the sources are located and separated, we need to decide which ones are brain and which ones are not. The plugin iclabel does this for us.



Based on these results we remove the components which are not brain with a high confidence.



Remove components from data -- pop_subcomp()

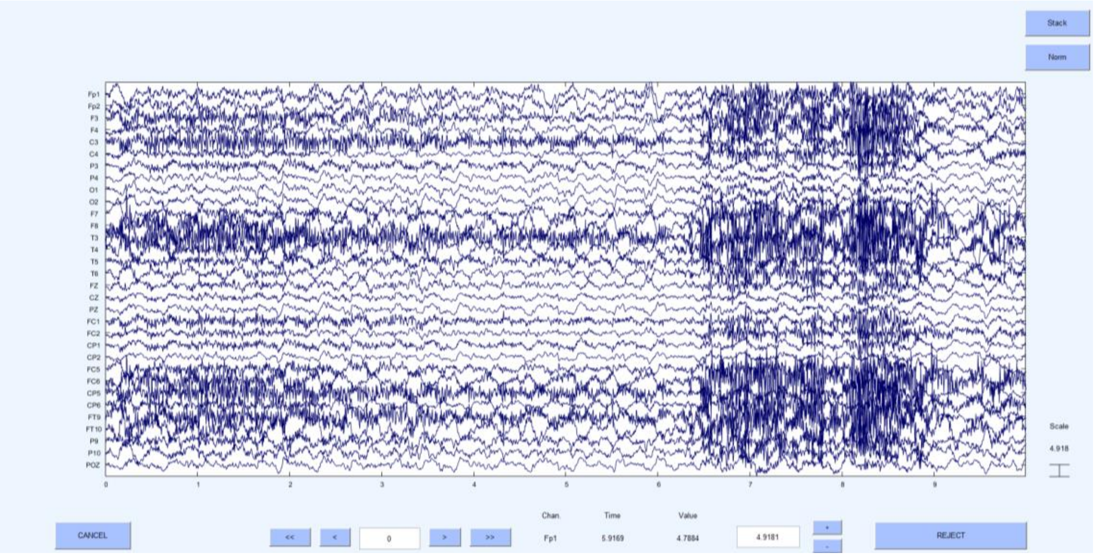
Note: for group level analysis, remove components in STUDY

List of component(s) to remove from data: 2 7 10 12 13 16 17

Or list of component(s) to retain:

Then we can epoch the results into different parts based on events and changes in signal. Here because our timeframe is too short we don't need epoching but in a large file its very useful.

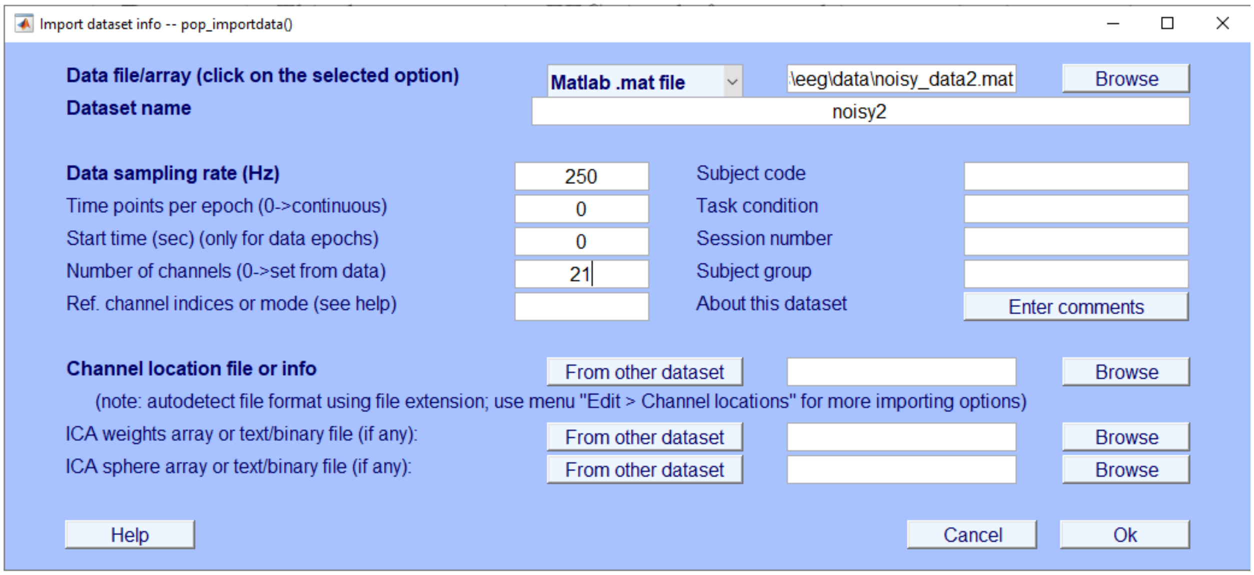
The final result is here:



The files of different steps of this report are attached to the report. We apply same methods to data 2 again but this time only images and steps are here and not everything is explained again.

Second dataset:

Importing :

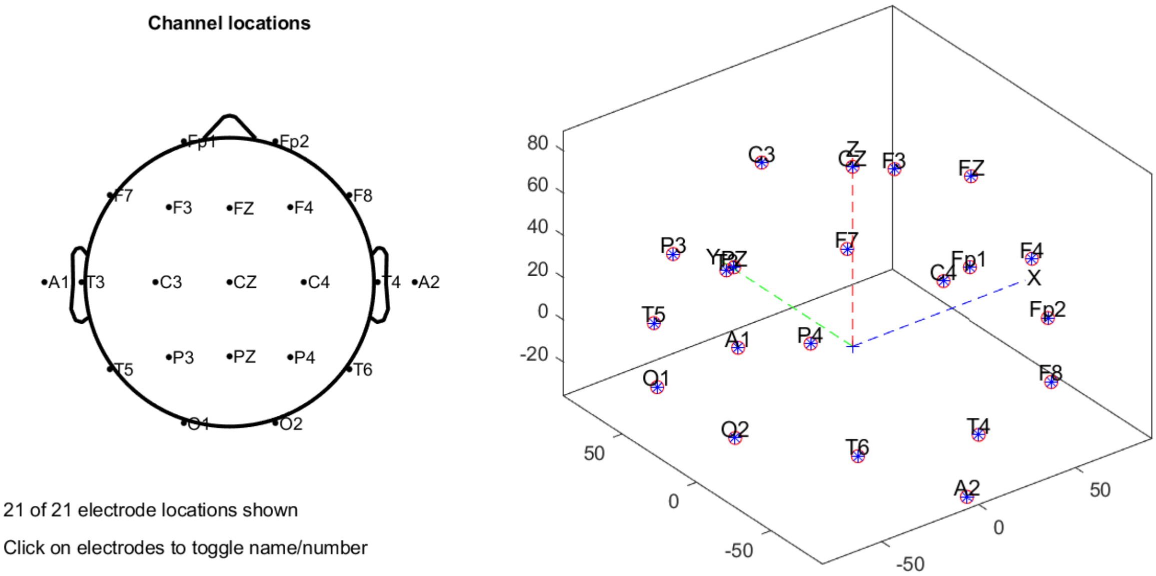


Import dataset info -- pop_importdata()

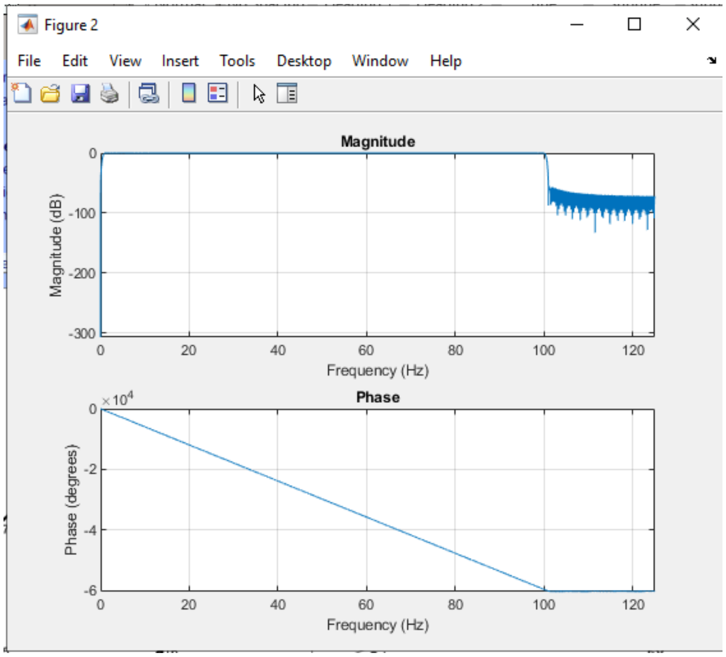
Data file/array (click on the selected option)	Matlab .mat file	\eeg\data\noisy_data2.mat	
Dataset name	noisy2		
Data sampling rate (Hz)	250	Subject code	
Time points per epoch (0->continuous)	0	Task condition	
Start time (sec) (only for data epochs)	0	Session number	
Number of channels (0->set from data)	21	Subject group	
Ref. channel indices or mode (see help)		About this dataset	
Channel location file or info			
ICA weights array or text/binary file (if any):			
ICA sphere array or text/binary file (if any):			

(note: autodetect file format using file extension; use menu "Edit > Channel locations" for more importing options)

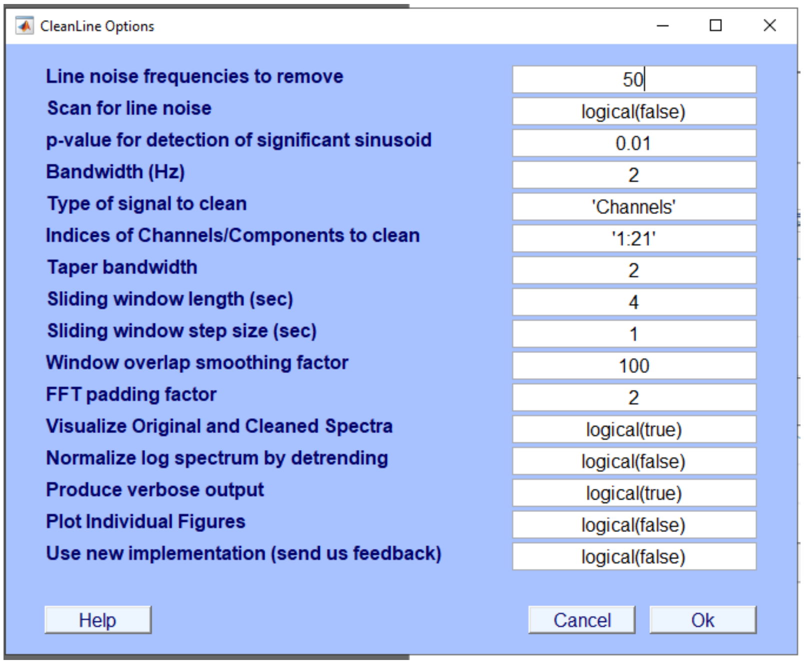
Channel locations:



Using bandpass filter:

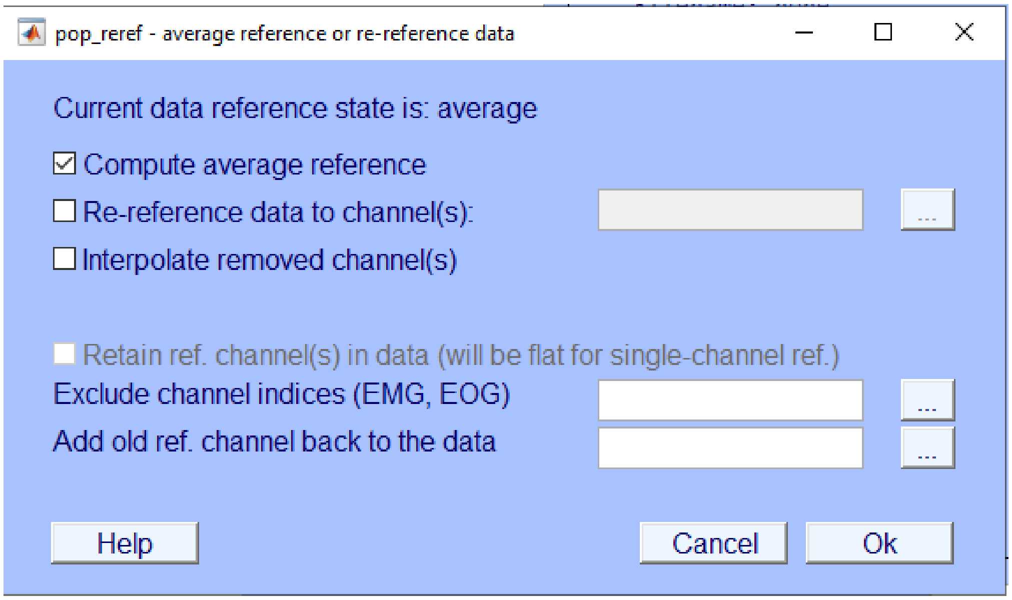


Using cleanline:



Line noise frequencies to remove	50
Scan for line noise	logical(false)
p-value for detection of significant sinusoid	0.01
Bandwidth (Hz)	2
Type of signal to clean	'Channels'
Indices of Channels/Components to clean	'1:21'
Taper bandwidth	2
Sliding window length (sec)	4
Sliding window step size (sec)	1
Window overlap smoothing factor	100
FFT padding factor	2
Visualize Original and Cleaned Spectra	logical(true)
Normalize log spectrum by detrending	logical(false)
Produce verbose output	logical(true)
Plot Individual Figures	logical(false)
Use new implementation (send us feedback)	logical(false)

Re-reference 1:



pop_reref - average reference or re-reference data

Current data reference state is: average

☑ Compute average reference

□ Re-reference data to channel(s):

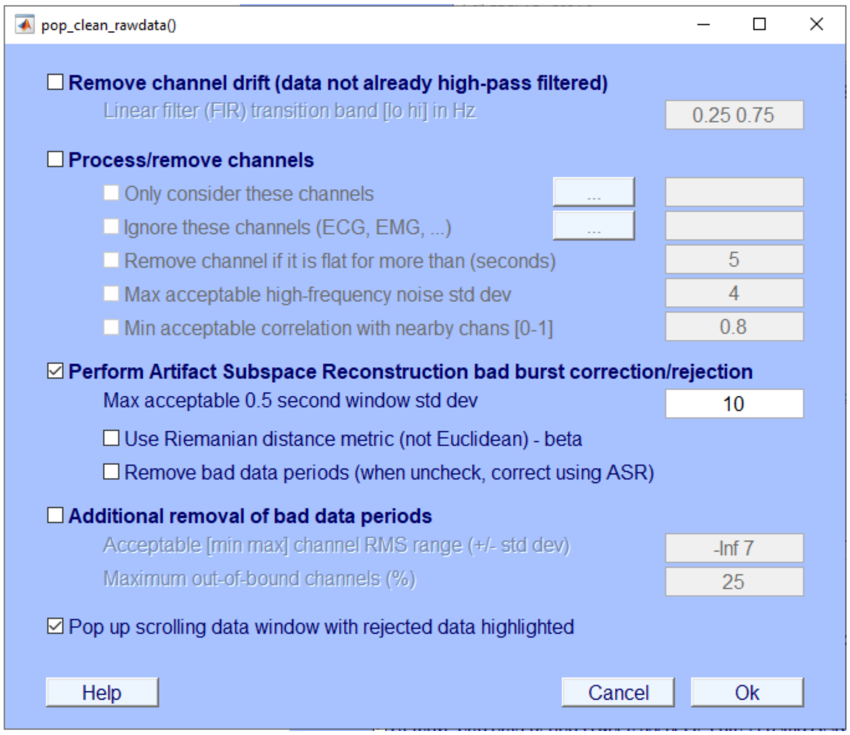
□ Interpolate removed channel(s)

□ Retain ref. channel(s) in data (will be flat for single-channel ref.)

Exclude channel indices (EMG, EOG)

Add old ref. channel back to the data

Removing artifacts:



pop_clean_rawdata()

□ **Remove channel drift (data not already high-pass filtered)**

Linear filter (FIR) transition band [lo hi] in Hz: 0.25 0.75

□ **Process/remove channels**

□ Only consider these channels

□ Ignore these channels (ECG, EMG, ...)

□ Remove channel if it is flat for more than (seconds): 5

□ Max acceptable high-frequency noise std dev: 4

□ Min acceptable correlation with nearby chans [0-1]: 0.8

☑ **Perform Artifact Subspace Reconstruction bad burst correction/rejection**

Max acceptable 0.5 second window std dev: 10

□ Use Riemanian distance metric (not Euclidean) - beta

□ Remove bad data periods (when uncheck, correct using ASR)

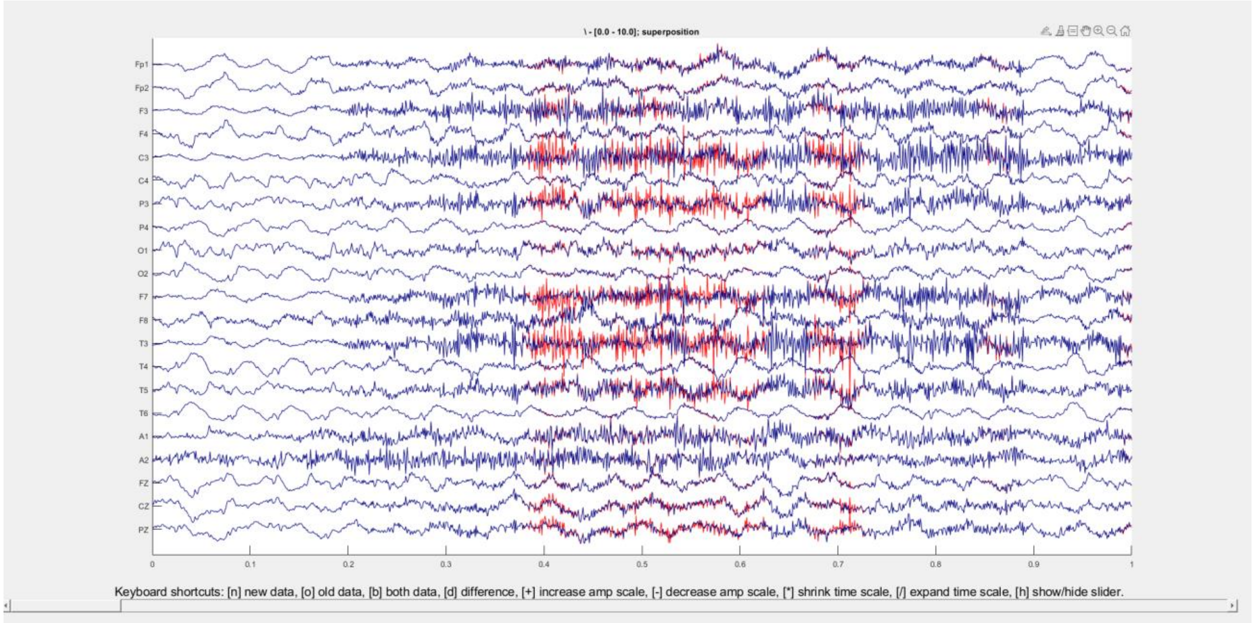
□ **Additional removal of bad data periods**

Acceptable [min max] channel RMS range (+/- std dev): -Inf 7

Maximum out-of-bound channels (%): 25

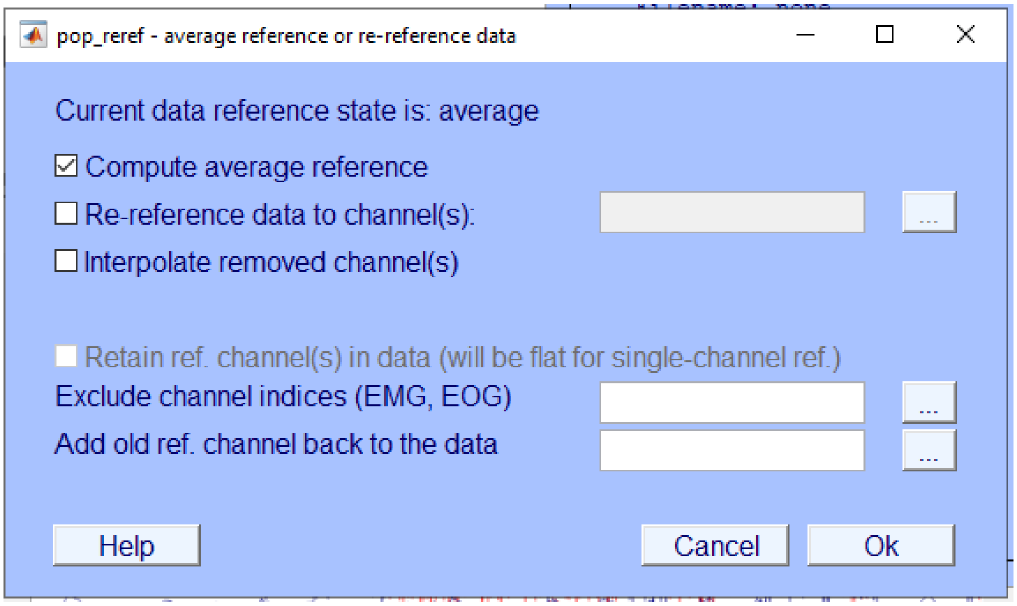
☑ Pop up scrolling data window with rejected data highlighted

As it was described in the tutorial videos, for shorter datasets we need to set STD dev lower to get the same level of results.



Keyboard shortcuts: [n] new data, [o] old data, [b] both data, [d] difference, [+] increase amp scale, [-] decrease amp scale, [*] shrink time scale, [/] expand time scale, [h] show/hide slider.

Re-reference 2:



Current data reference state is: average

☑ Compute average reference

☐ Re-reference data to channel(s):

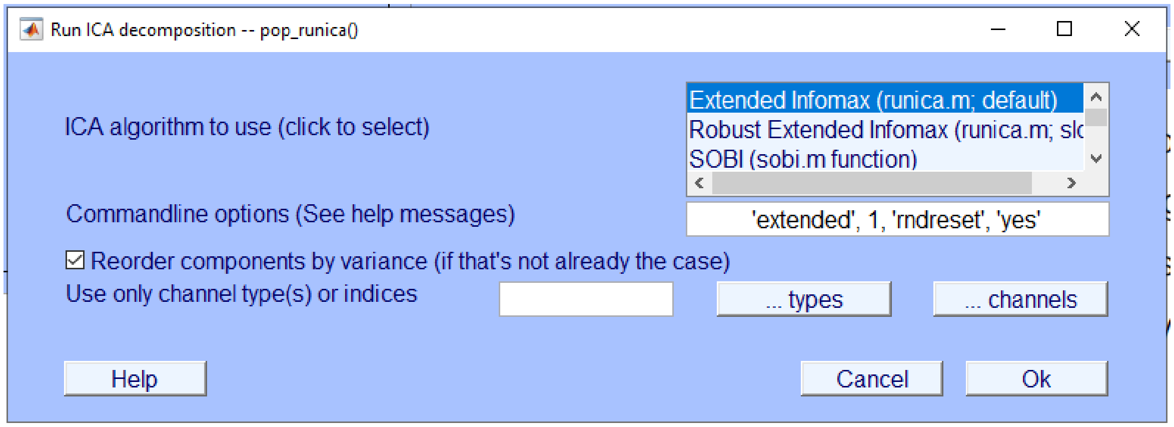
☐ Interpolate removed channel(s)

☐ Retain ref. channel(s) in data (will be flat for single-channel ref.)

Exclude channel indices (EMG, EOG)

Add old ref. channel back to the data

ICA decomposition:



Run ICA decomposition -- pop_runica()

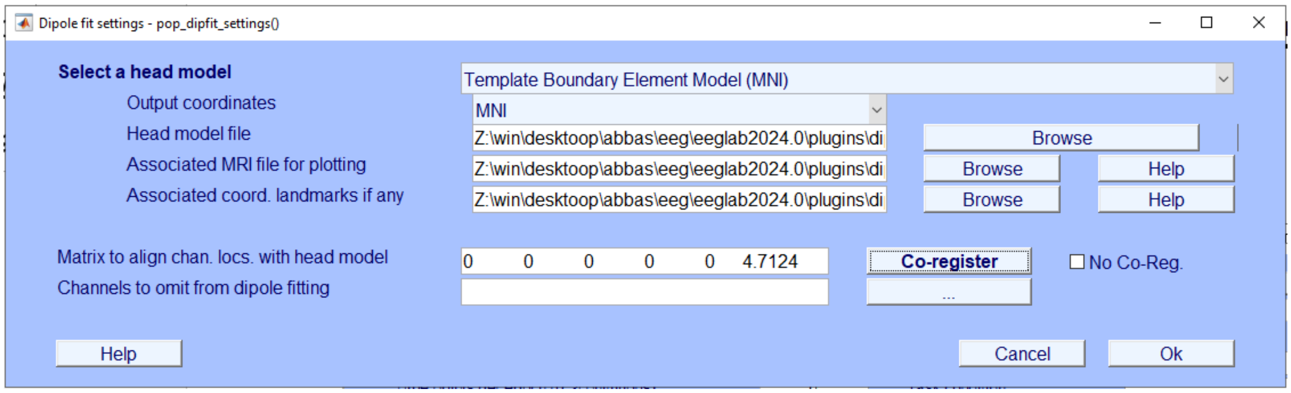
ICA algorithm to use (click to select): Extended Infomax (runica.m; default) | Robust Extended Infomax (runica.m; slc | SOBI (sobi.m function)

Commandline options (See help messages): 'extended', 1, 'rndreset', 'yes'

☑ Reorder components by variance (if that's not already the case)

Use only channel type(s) or indices:

Setting head model:



Dipole fit settings - pop_dipfit_settings()

Select a head model: Template Boundary Element Model (MNI)

Output coordinates: MNI

Head model file: Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\di

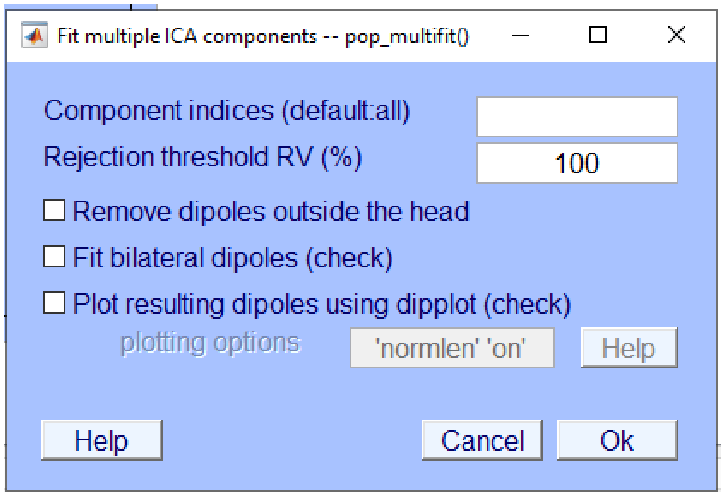
Associated MRI file for plotting: Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\di

Associated coord. landmarks if any: Z:\win\desktoop\abbas\eeg\eeglab2024.0\plugins\di

Matrix to align chan. locs. with head model: 0 0 0 0 0 4.7124

Channels to omit from dipole fitting:

□ No Co-Reg.



Fit multiple ICA components -- pop_multifit()

Component indices (default:all):

Rejection threshold RV (%): 100

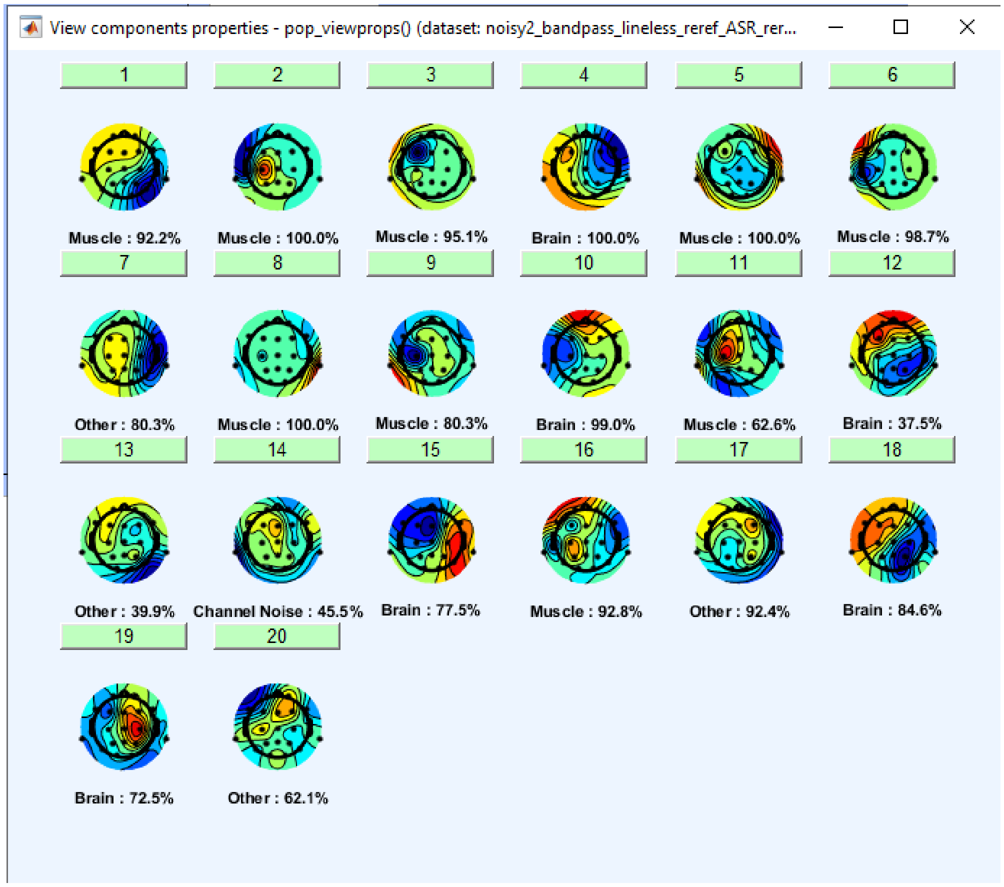
□ Remove dipoles outside the head

□ Fit bilateral dipoles (check)

□ Plot resulting dipoles using dipplot (check)

plotting options: 'normlen' 'on'

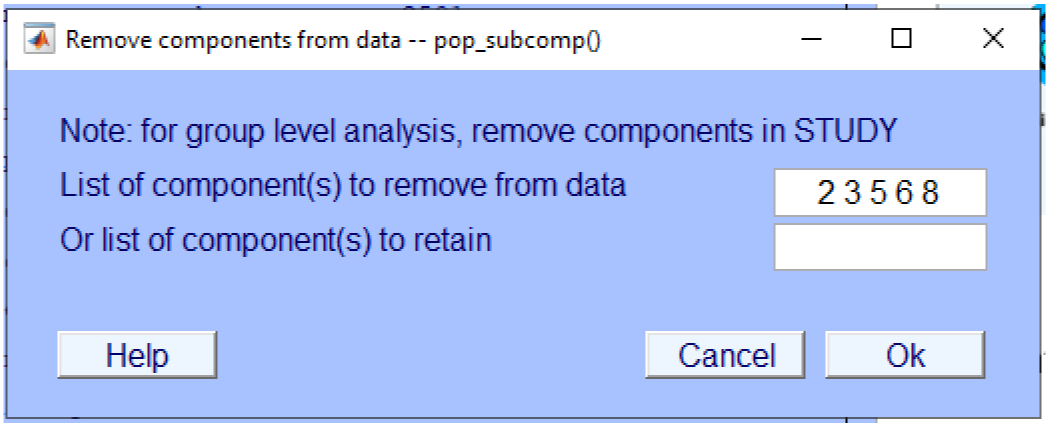
IClabel source identification:



View components properties - pop_viewprops() (dataset: noisy2_bandpass_lineless_reref_ASR_rer...

1 Muscle : 92.2%
2 Muscle : 100.0%
3 Muscle : 95.1%
4 Brain : 100.0%
5 Muscle : 100.0%
6 Muscle : 98.7%
7 Other : 80.3%
8 Muscle : 100.0%
9 Muscle : 80.3%
10 Brain : 99.0%
11 Muscle : 62.6%
12 Brain : 37.5%
13 Other : 39.9%
14 Channel Noise : 45.5%
15 Brain : 77.5%
16 Muscle : 92.8%
17 Other : 92.4%
18 Brain : 84.6%
19 Brain : 72.5%
20 Other : 62.1%

Removing identified non brain sources:



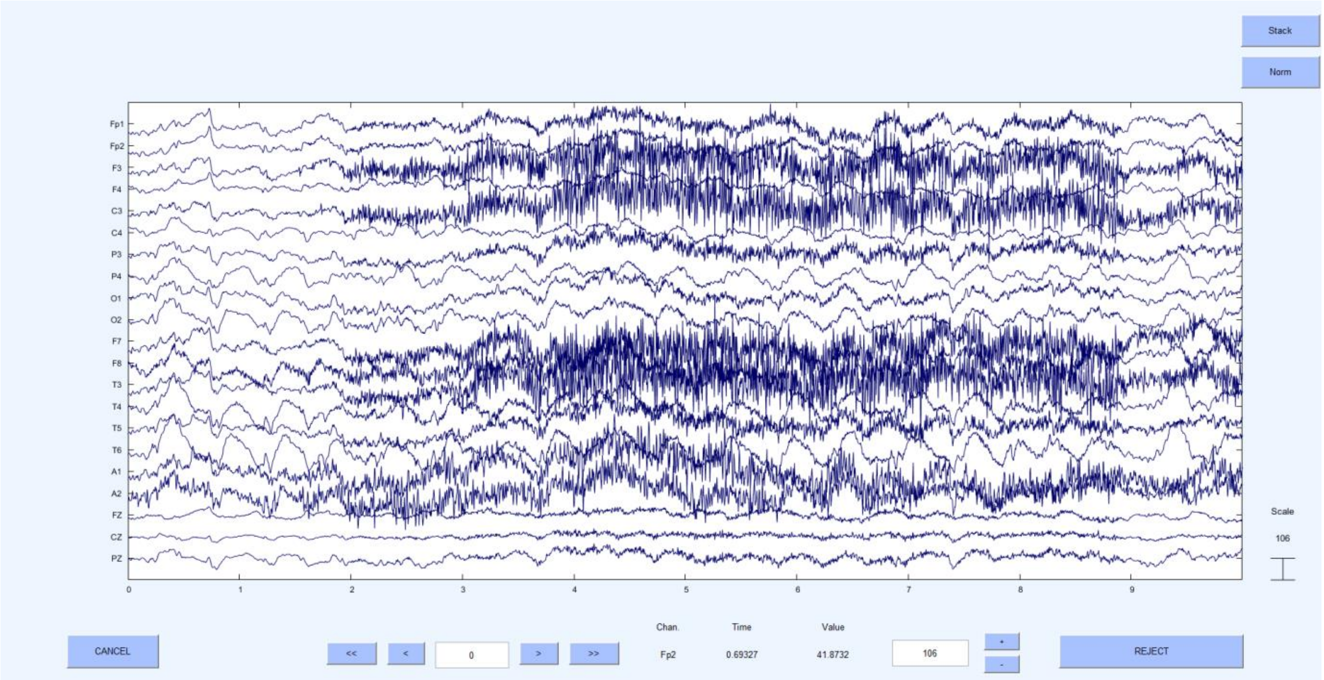
Remove components from data -- pop_subcomp()

Note: for group level analysis, remove components in STUDY

List of component(s) to remove from data: 2 3 5 6 8

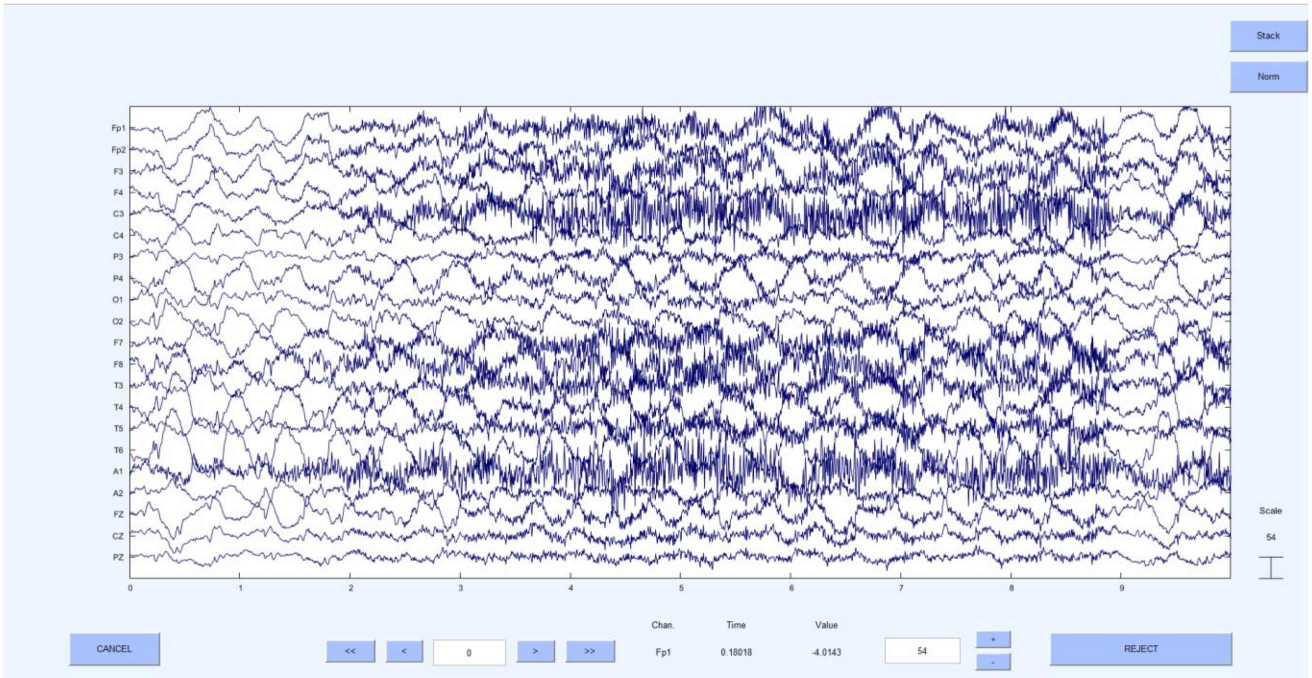
Or list of component(s) to retain:

Comparison:



Chan.	Time	Value
Fp2	0.69327	41.8732

original



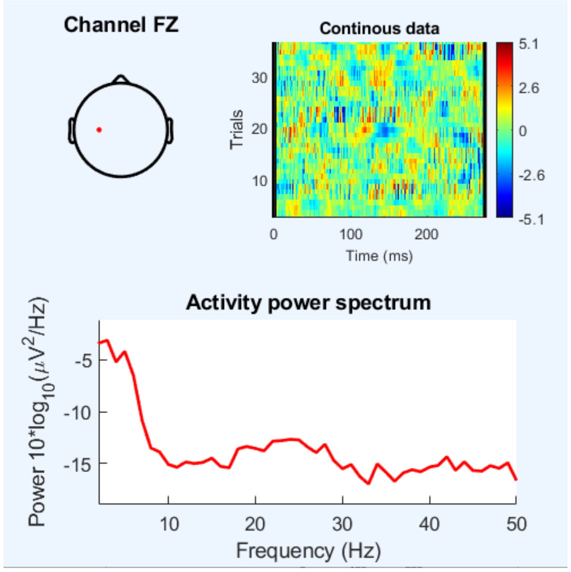
Chan.	Time	Value
Fp1	0.18018	-4.0143

Final

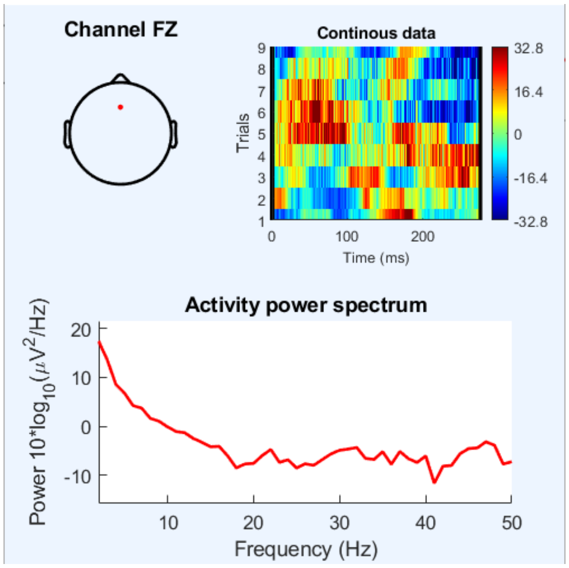
The files associated with second dataset are available with the report too.

5) Deliverables:

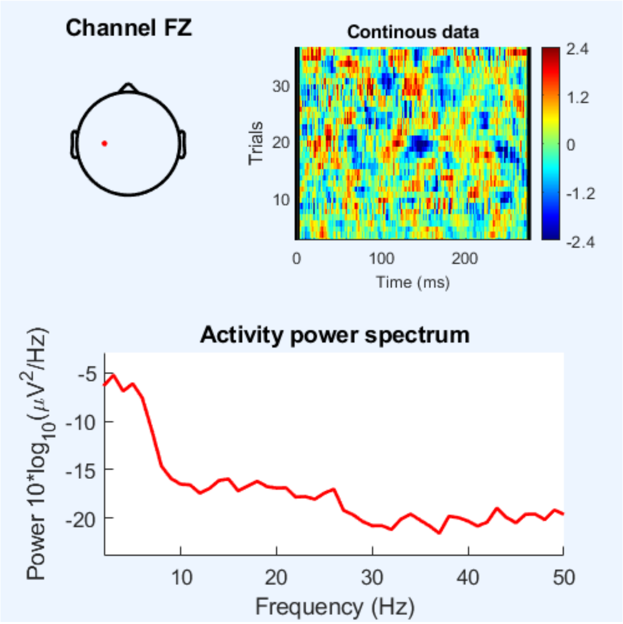
Frequency Spectrum: Present figures showing the frequency spectrum of the Fz channel data before and after filtering.



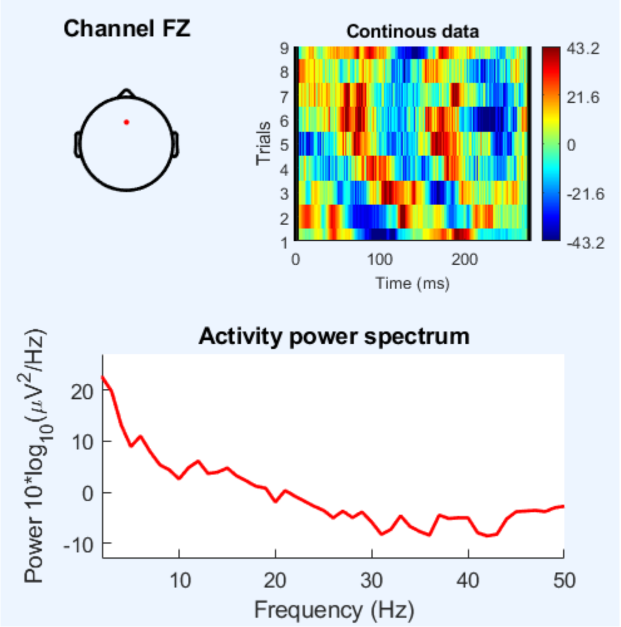
Data1, noisy, Fz channel



Data2, noisy, Fz channel



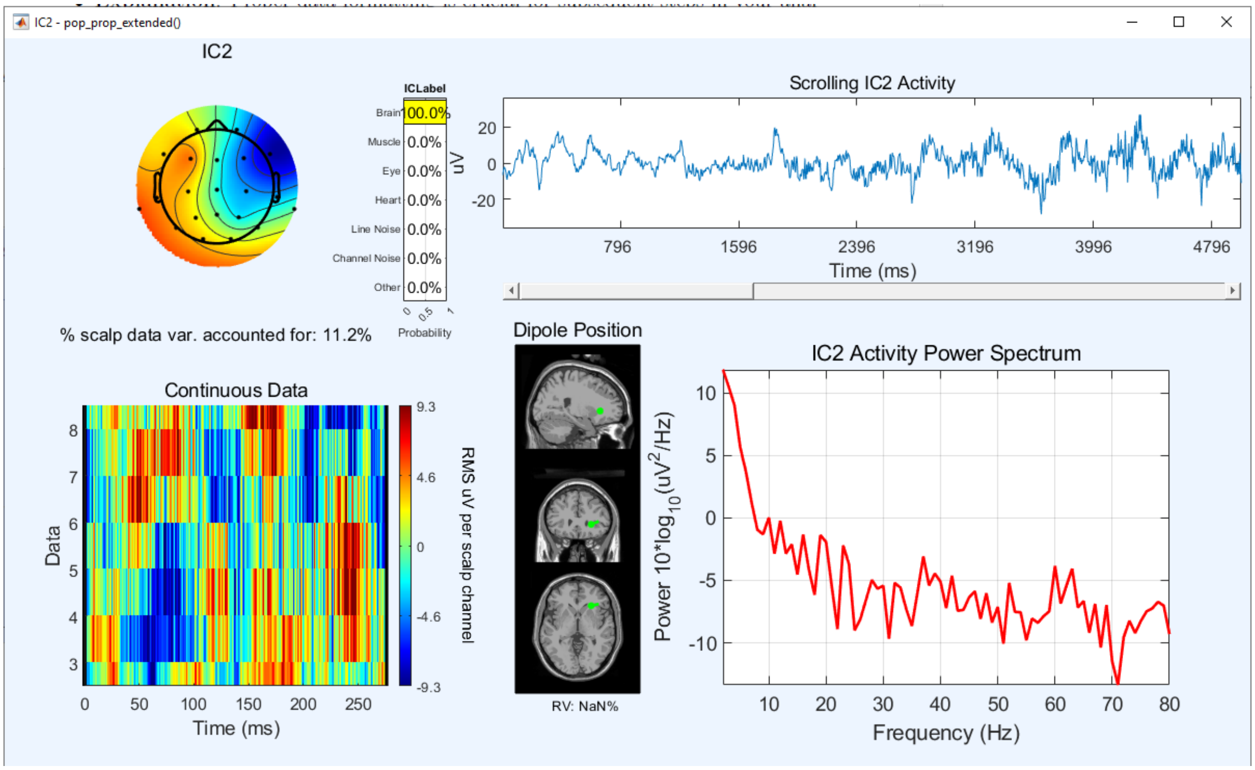
Data1, clean, Fz channel



Data2, clean, Fz channel

ICA Components: Provide a figure from one of the brain components identified by ICA, with details on the component.

Here we present the best brain signal we had, which had 100% brain tag. There are some notable characteristics about this component, for example it exactly matches the 1/f amplitude which we are expecting, or its location is in the brain which means its not eye or muscle.



Processed Data: Report on the final cleaned and processed data, including any removed noisy trials.

This comparison was done in the previous section. We plotted each step and observed the changes.