EEGLAB- Quick Tutorial

Assaf Peleg, Dor Zazon

November 22, 2020

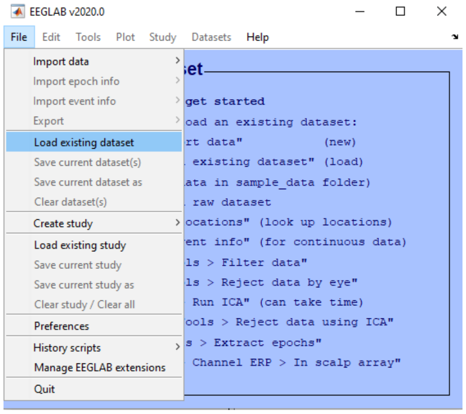
First steps

First, download **EEGLAB** and acquire a relatively new version of **MATLAB**. You can download **EEGLAB** using the following link: `https://sccn.ucsd.edu/eeglab/download.php`. To access **EEGLAB** wiki go to: `https://sccn.ucsd.edu/wiki/EEGLAB`.

Next, open **MATLAB**, set your workspace to the folder in which you installed/extracted **EEGLAB** and write in the **Command Window** ”eeglab” and press Enter.

Examining a dataset

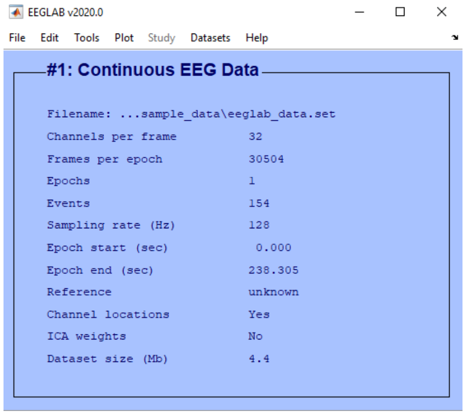
1) **Loading a dataset**: The first thing you might want to do is load a dataset, examine it’s properties and take a look at it. To do so, click on **File** and then choose **Load existing dataset**. See **Figure 1**.



```
get started
oad an existing dataset:
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"
```

Figure 1: Loading an existing dataset

2) Next, the dataset will be loaded and it's properties will be printed to the screen, e.g. the dataset that we have loaded has 32 channels(electrodes) and 154 events which are markers to align data with specific events that have occurred such as stimulus on the screen. See **Figure 2**.



#1: Continuous EEG Data

```
Filename: ...sample_data\eeglab_data.set
Channels per frame           32
Frames per epoch             30504
Epochs                       1
Events                       154
Sampling rate (Hz)           128
Epoch start (sec)             0.000
Epoch end (sec)              238.305
Reference                    unknown
Channel locations            Yes
ICA weights                  No
Dataset size (Mb)            4.4
```

Figure 2: Dataset properties

3) **Looking at the data**: The following thing you might want to do is to look at the data with your own eyes. To do so, click on **Plot** and then choose **Channel data (scroll)**. See **Figure 3**.

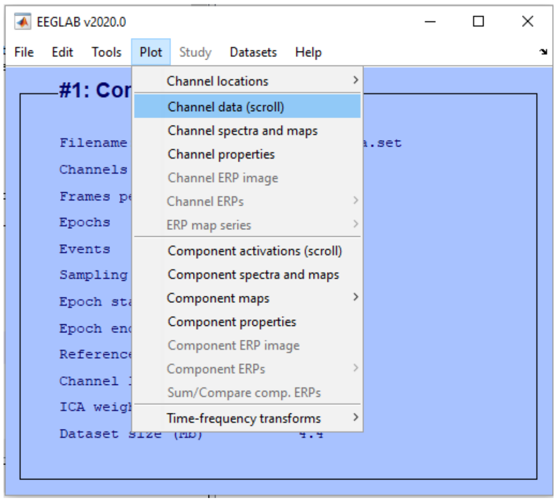


Figure 3: Looking at the data

4) A window will open. On the left you'll see the names of the channels/electrodes, in the center you'll see the actual data and on the bottom you'll see arrows allowing you to scroll it. You'll also notice colored lines- these are the events we mentioned earlier. See **Figure 4**.

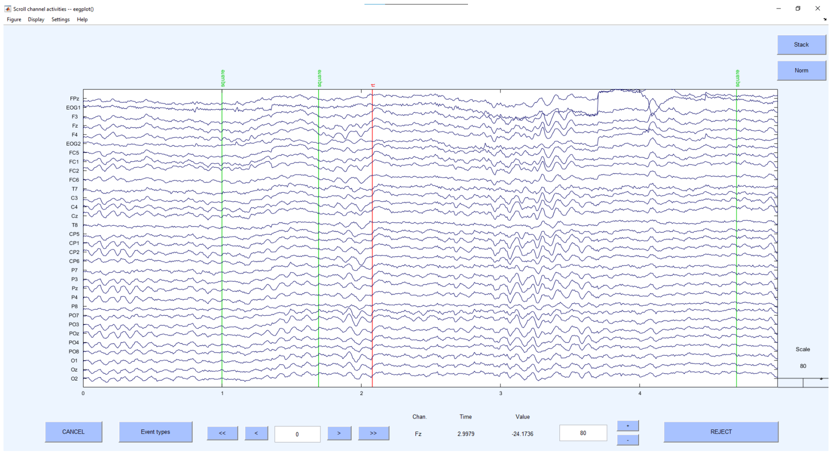


Figure 4: Scrolling the data

Cleaning the data

1) **Removing bad sections**: A look at the data might reveal some bad and noisy sections(in all the electrodes at the same time), and we might want to manually delete them(This will affect the phase of the waves because we cut the data but it will not be significant). This can be done by selecting them using the mouse(click on the start of the section, hold your mouse and drag it to the end of the section). Then, click on the **Reject** button.
See **Figure 5**.

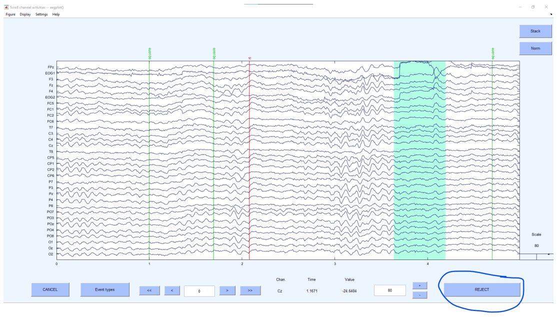
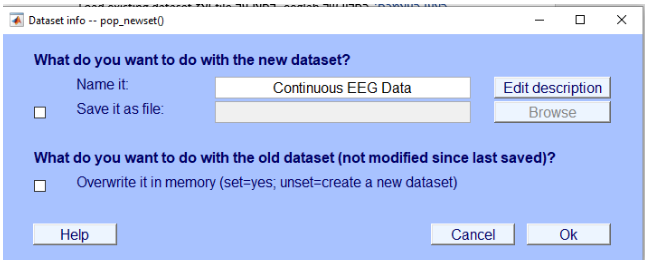


Figure 5: Rejecting bad sections

2) Next the scrolling window will close and a new window will open. This window will allow you to choose if to overwrite your current dataset with the changes by selecting overwrite and ok or create a new one by clicking ok without selecting anything(See **Figure 6**)- **Explanation**: When working in **EEGLAB**, datasets can be loaded and used in parallel. Each time you preform an action, it will be preformed on the current dataset. To choose a dataset go to **Datasets**. For example, we chose not to overwrite our datasets so now we can choose between two different datasets. See **Figure 7**.



Dataset info -- pop_newset()

What do you want to do with the new dataset?

Name it: Continuous EEG Data

□ Save it as file:

Edit description

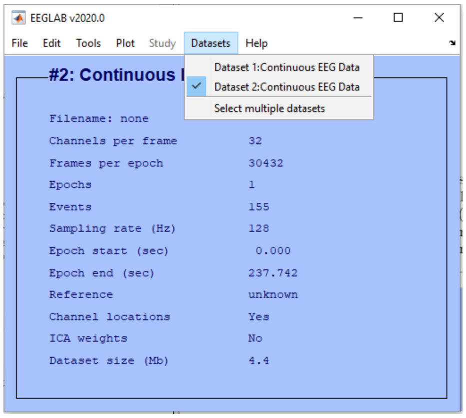
Browse

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)

Help Cancel Ok

Figure 6: Saving the changes

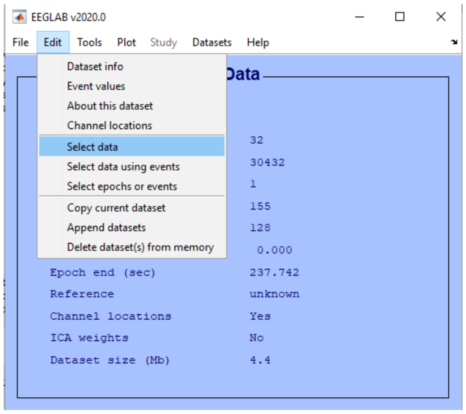


#2: Continuous I

```
Filename: none
Channels per frame          32
Frames per epoch            30432
Epochs                      1
Events                      155
Sampling rate (Hz)          128
Epoch start (sec)            0.000
Epoch end (sec)             237.742
Reference                   unknown
Channel locations           Yes
ICA weights                 No
Dataset size (Mb)           4.4
```

Figure 7: Choosing a dataset

3) **Removing bad electrodes**: Our next order of business is to look for bad electrodes as some of them might be noisy and full of garbage altogether. After we scroll the data once more, and decide that there is such an electrode, we would want to delete it. To do so, click on **Edit** and then choose **Select data**. See **Figure 8**.

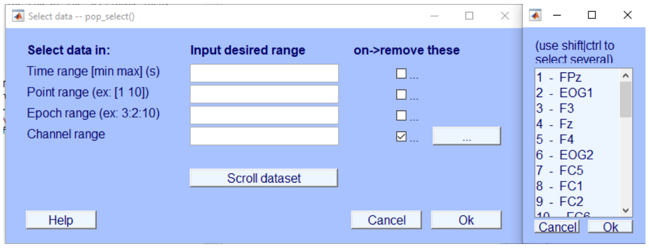


Data

```
                            32
                            30432
                            1
                            155
                            128
                             0.000
Epoch end (sec)             237.742
Reference                   unknown
Channel locations           Yes
ICA weights                 No
Dataset size (Mb)           4.4
```

Figure 8: Selecting data to reject

4) Next, check the box of **"on→remove these"** for **Channel range** and click on the button with the ellipses(...) to choose the electrodes you want to delete. Finally click on ok and choose if to overwrite the dataset or create a new one. See **Figure 9** and **Figure 10**.



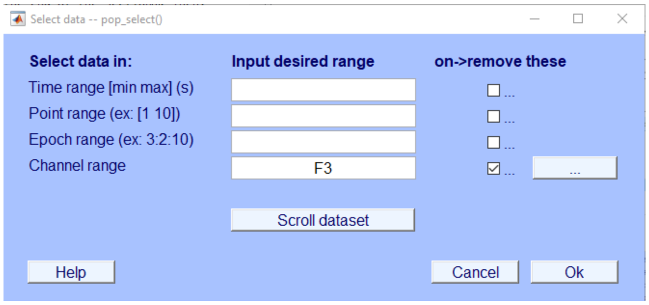
Select data -- pop_select()

Select data in:	Input desired range	on->remove these
Time range [min max] (s)		□ ...
Point range (ex: [1 10])		□ ...
Epoch range (ex: 3:2:10)		□ ...
Channel range		☑ ...

(use shift|ctrl to select several)

1 - FPz
2 - EOG1
3 - F3
4 - Fz
5 - F4
6 - EOG2
7 - FC5
8 - FC1
9 - FC2
10 - FC6

Figure 9: After checking the box and clicking on the ellipses



Select data -- pop_select()

Select data in:	Input desired range	on->remove these
Time range [min max] (s)		□ ...
Point range (ex: [1 10])		□ ...
Epoch range (ex: 3:2:10)		□ ...
Channel range	F3	☑ ...

Figure 10: After picking an electrode

5) If you scroll the data now, you wont see that electrode anymore. See **Figure 11**.

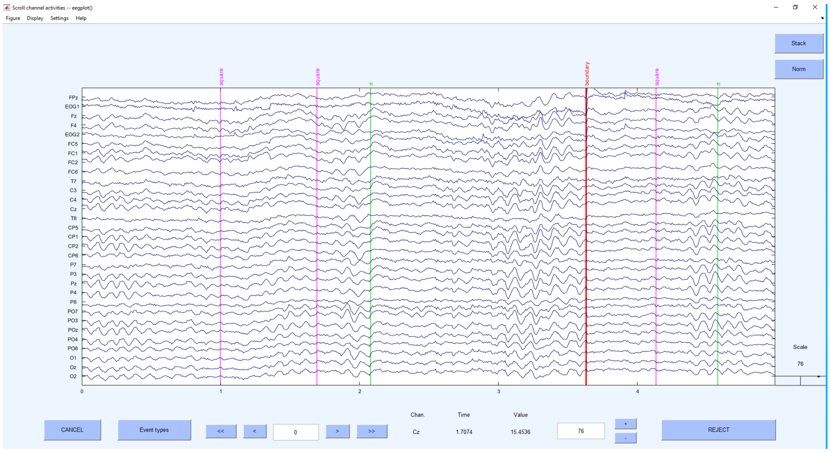


Figure 11: F3 electrode is now gone

Filtering the data

1) **HIGH PASS Filter**: Next, we'll want to filter the data by searching for which frequencies we have undesirable peaks and checking if they need to be filtered or not(We'll search for the frequencies which we would want to always remove). However, first thing first and we'll need to look at the data again. This time will use a PSD graph. To do so, click on **Plot** and then choose **Channel spectra and maps**. See **Figure 12**.

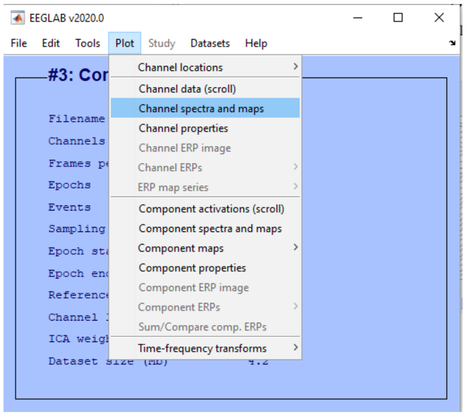
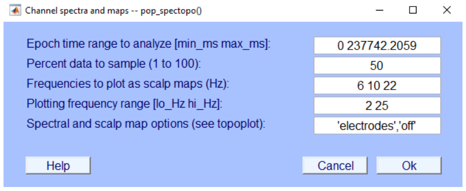


Figure 12: Channel spectra and maps

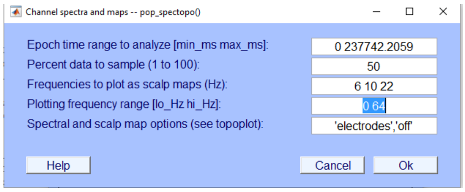
2) A window will open which will allow us to choose several options for the PSD graph. See **Figure 13**.



Epoch time range to analyze [min_ms max_ms]: 0 237742.2059
Percent data to sample (1 to 100): 50
Frequencies to plot as scalp maps (Hz): 6 10 22
Plotting frequency range [lo_Hz hi_Hz]: 2 25
Spectral and scalp map options (see topoplot): 'electrodes','off'

Figure 13: PSD graph options

3) For example, we would like to view the PSD graph for frequencies between 0hz and 64hz, and therefore we change the plotting range and click ok. See **Figure 14**.



Epoch time range to analyze [min_ms max_ms]: 0 237742.2059
Percent data to sample (1 to 100): 50
Frequencies to plot as scalp maps (Hz): 6 10 22
Plotting frequency range [lo_Hz hi_Hz]: 0 64
Spectral and scalp map options (see topoplot): 'electrodes','off'

Figure 14: PSD graph plotting range- 0hz to 64hz

4) Next, the PSD graph will open. First, we can see the alpha waves at 10hz and we see an interesting anomaly at 60hz. This is the American power supply frequency where in Israel it will be 50hz. See **Figure 15**. We would surely want to filter that. We can do this using a **HIGH PASS Filter**.

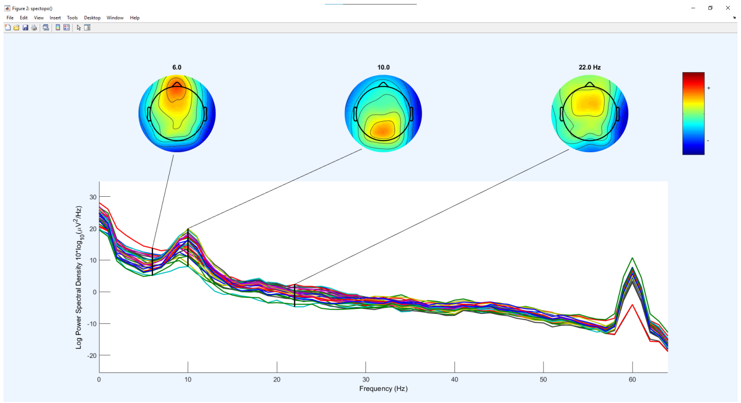


Figure 15: PSD

5) To do so, go to **Tools**, **Filter the data** and then **Basic FIR**. See **Figure 16**.

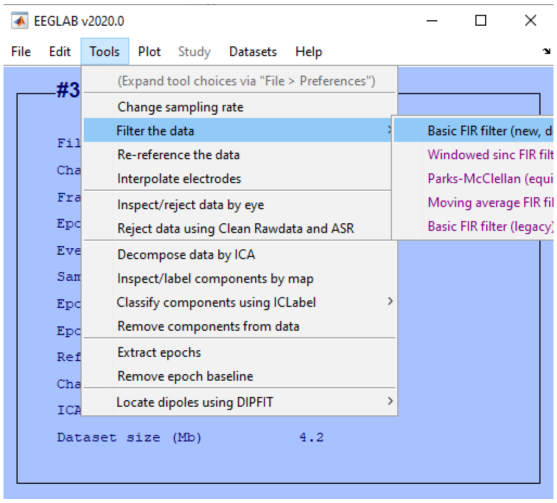
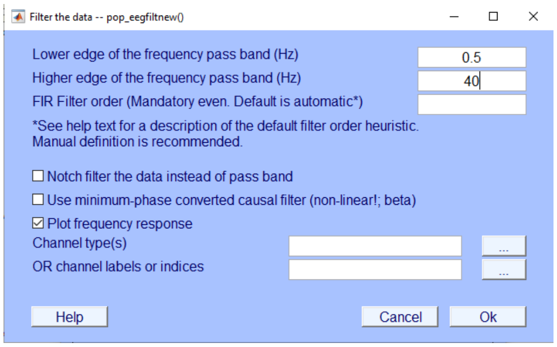


Figure 16: Basic FIR

6) The filter we'll want to do mostly is 0.5hz to 40hz as we are not interested in higher or lower frequencies. To do so, enter 0.5 to the low filter and 40 to the high filter and click ok. Click ok to create a new dataset as always when the saving window opens. See **Figure 17**.



Filter the data -- pop_eegfiltnew()

Lower edge of the frequency pass band (Hz) 0.5
Higher edge of the frequency pass band (Hz) 40
FIR Filter order (Mandatory even. Default is automatic*)

*See help text for a description of the default filter order heuristic.
Manual definition is recommended.

□ Notch filter the data instead of pass band
□ Use minimum-phase converted causal filter (non-linear!; beta)
☑ Plot frequency response
Channel type(s)
OR channel labels or indices

Figure 17: 0.5hz to 40hz HIGH PASS filter

7) Next, a window will open plotting the filter being used. See **Figure 18**. However, to see the actual affect on our data we'll need to plot another PSD graph(Don't forget to change the plotting range as before). See **Figure 19**. As can be seen from the PSD graph, the 60hz peak was removed.

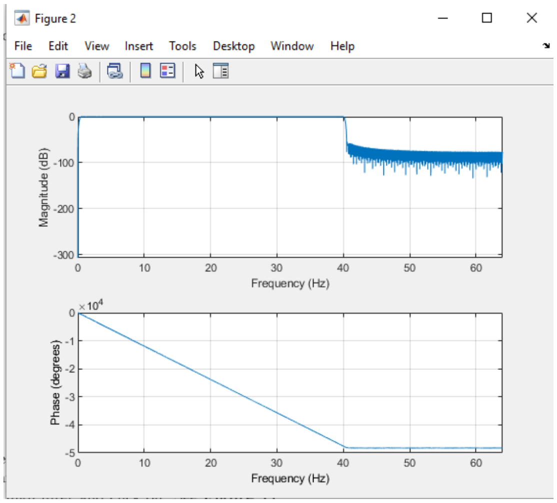


Figure 18: Filter plot

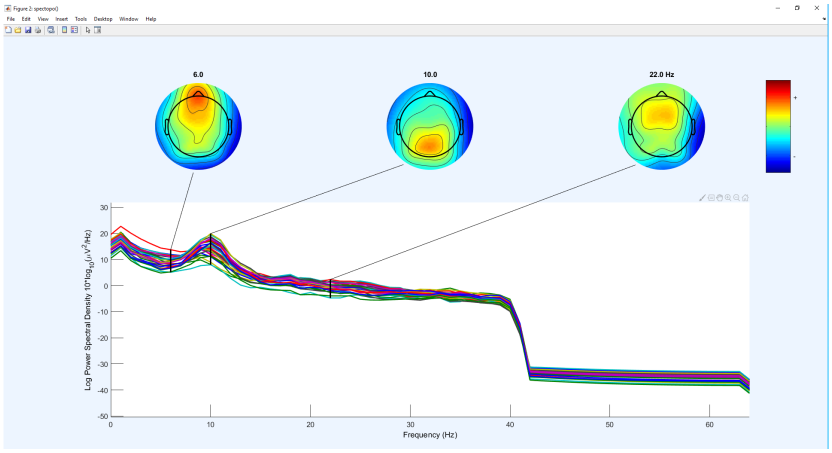
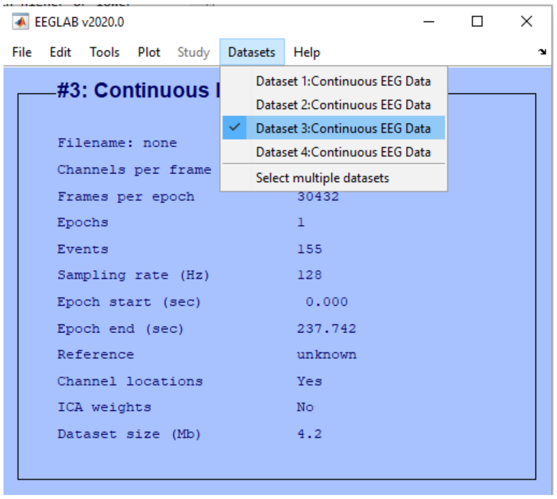


Figure 19: PSD graph

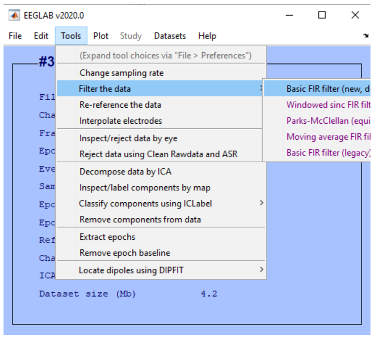
8) **Notch Filter**: Another way to filter is to use a notch filter. A notch filter will allow us to discard a range of frequencies. First, we'll return to the previous dataset before the affects of the high pass filter. In order to do so, go to **Datasets** and then choose dataset 3. See **Figure 20**.


EEGLAB v2020.0
File
Edit
Tools
Plot
Study
Datasets
Help
#3: Continuous I
Dataset 1:Continuous EEG Data
Dataset 2:Continuous EEG Data
✓ Dataset 3:Continuous EEG Data
Dataset 4:Continuous EEG Data
Select multiple datasets

```
Filename: none
Channels per frame
Frames per epoch               30432
Epochs                         1
Events                         155
Sampling rate (Hz)             128
Epoch start (sec)               0.000
Epoch end (sec)                237.742
Reference                      unknown
Channel locations              Yes
ICA weights                    No
Dataset size (Mb)              4.2
```

Figure 20: Choosing the previous dataset

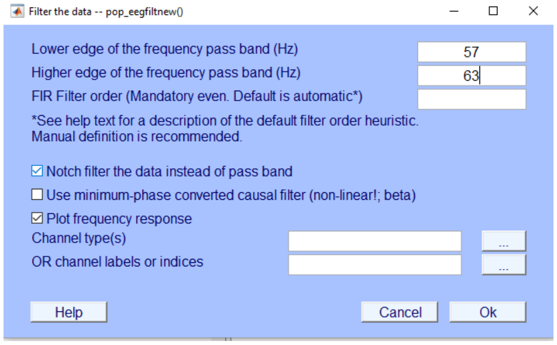
9) Next, to apply the filter go to **Tools**, **Filter the data** and then **Basic FIR**. See **Figure 21**.



```
Dataset size (Mb)          4.2
```

Figure 21: Opening the filter window again

10) Then, check the box near the notch filter and enter in the low filter input box 57 and in the high filter input box enter 63. See **Figure 22**.



Filter the data -- pop_eegfiltnew()

Lower edge of the frequency pass band (Hz) 57

Higher edge of the frequency pass band (Hz) 63

FIR Filter order (Mandatory even. Default is automatic*)

*See help text for a description of the default filter order heuristic.
Manual definition is recommended.

☑ Notch filter the data instead of pass band

□ Use minimum-phase converted causal filter (non-linear!; beta)

☑ Plot frequency response

Channel type(s)

OR channel labels or indices

Figure 22: Applying a notch filter

11) Next, a window will open plotting the filter being used. However, to see the actual affect on our data we'll need to plot another PSD graph(Don't forget to change the plotting range as before). See **Figure 23**. As can be seen from the PSD graph, the 60hz peak was removed, but this time the filter was narrower and replaced the 60hz peak with a negetive one.

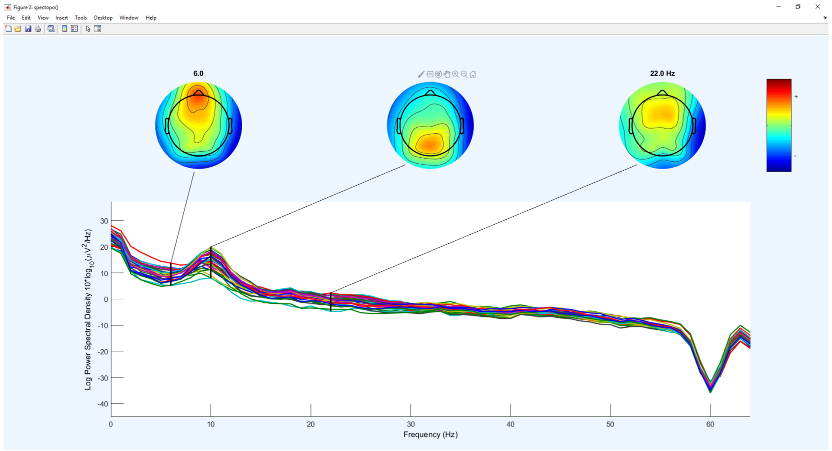
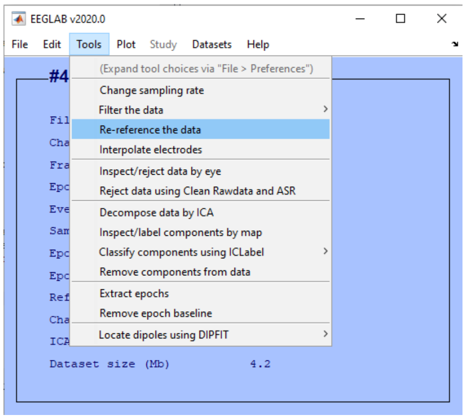


Figure 23: PSD graph

12) Before we continue, let's return to dataset 4 in which we preformed the high pass filter.

Final touches

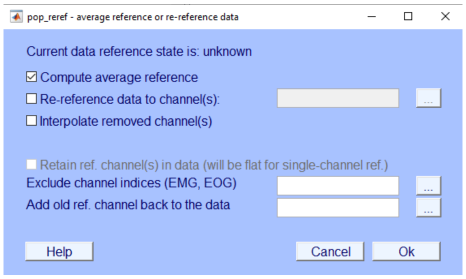
1) **Average reference**: The following thing we might want to do is removing relatively robustly bias and noise that affects all the electrodes altogether such as eye blinks. To do so, we can calculate the average reference and remove it. Click on **Tools** and choose **Re-reference the data**. See **Figure 24**.


EEGLAB v2020.0
File
Edit
Tools
Plot
Study
Datasets
Help
#4
(Expand tool choices via "File > Preferences")
Change sampling rate
Filter the data
Re-reference the data
Interpolate electrodes
Inspect/reject data by eye
Reject data using Clean Rawdata and ASR
Decompose data by ICA
Inspect/label components by map
Classify components using ICLabel
Remove components from data
Extract epochs
Remove epoch baseline
Locate dipoles using DIPFIT
Fil
Cha
Fra
Epc
Eve
Sam
Epc
Epc
Ref
Cha
ICA

Dataset size (Mb) 4.2

Figure 24: Re-referencing the data

2) Make sure **Compute the average reference** is selected, click ok and save the new dataset. See **Figure 25**.



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

Figure 25: Average reference

3) **Interpolation**: Next, after we cleaned the data with various methods, we need to handle our deleted electrode. Recall that we removed F3 electrode as we thought it was too noisy and full of garbage data. However, if we have many test subjects, and in each subject we delete a different electrode, that might cause problems. Therefore we cannot just remove noisy electrodes. To mitigate this we can use interpolation. First, to make this clear, choose the first dataset and then click on **Plot**, **Channel locations** and then choose **By name**. See **Figure 26**.

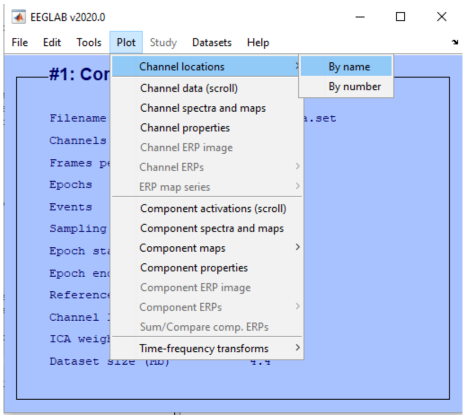


Figure 26: Plotting channel locations

4) The way we will mitigate our issue is to use nearby electrodes data from FC5, FC1, Fz, etc... to figure out what might the F3 electrode would have reported. See **Figure 27**.

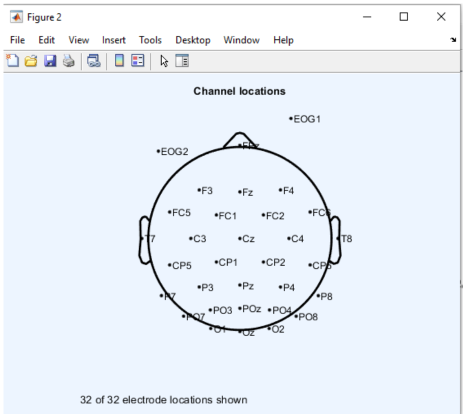
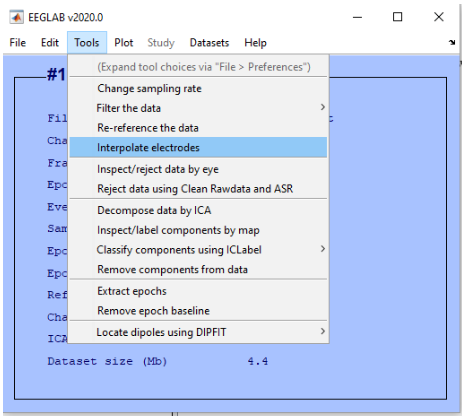


Figure 27: Channel location plot

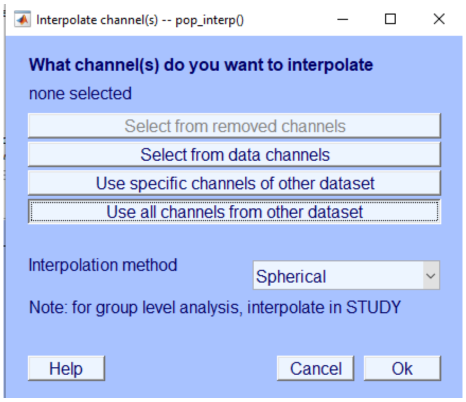
5) So, we figured out we need interpolation. First, choose our latest dataset number 6. Now, click on **Tools** and then choose **Interpolate electrodes**. See **Figure 28**.


EEGLAB v2020.0
File
Edit
Tools
Plot
Study
Datasets
Help
#1
(Expand tool choices via "File > Preferences")
Change sampling rate
Filter the data
Re-reference the data
Interpolate electrodes
Inspect/reject data by eye
Reject data using Clean Rawdata and ASR
Decompose data by ICA
Inspect/label components by map
Classify components using ICLabel
Remove components from data
Extract epochs
Remove epoch baseline
Locate dipoles using DIPFIT
Fil
Cha
Fra
Epo
Eve
Sam
Epo
Epo
Ref
Cha
ICA

Dataset size (Mb) 4.4

Figure 28: Interpolate electrodes

6) Next, the interpolation needs to know the location of the F3 electrode. However, as we deleted it, we don't know it's location. The solution to this is to use our other datasets. For example, our first dataset must contain F3 location. So, in the window that opens choose **Use all channels from other dataset**, in the window that opens enter 1 and click ok. See **Figure 29**.


Interpolate channel(s) -- pop_interp()
Select from removed channels
Select from data channels
Use specific channels of other dataset
Use all channels from other dataset
Help
Cancel
Ok

What channel(s) do you want to interpolate

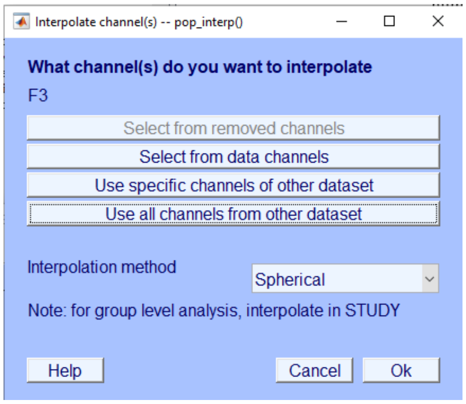
none selected

Interpolation method Spherical

Note: for group level analysis, interpolate in STUDY

Figure 29: Use all channels from other dataset

7) If all went well you should see at the top of the window that the F3 electrode was figured out automatically as the one to be interpolated. See **Figure 30**.


Interpolate channel(s) -- pop_interp()
Select from removed channels
Select from data channels
Use specific channels of other dataset
Use all channels from other dataset
Help
Cancel
Ok

What channel(s) do you want to interpolate

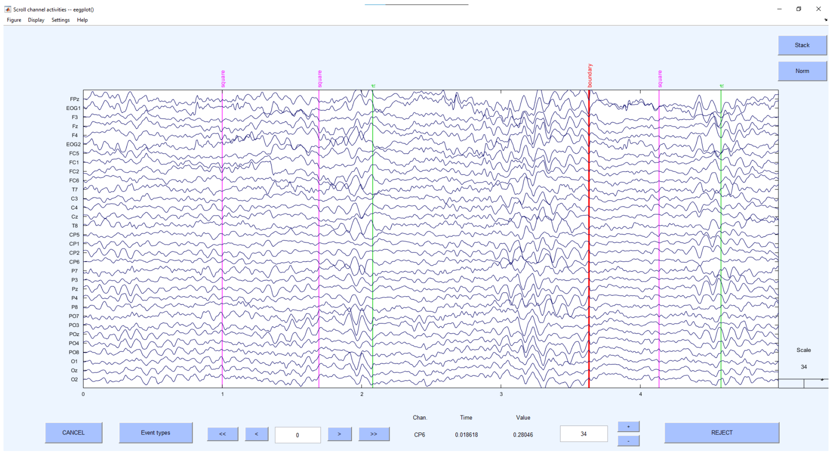
F3

Interpolation method: Spherical

Note: for group level analysis, interpolate in STUDY

Figure 30: Interpolation

8) Now click ok and choose not to overwrite the dataset. If you plot now the data you should see that the F3 electrode is back. See **Figure 31**.



Chan. CP6

Time 0.018618

Value 0.28046

Figure 31: F3 electrode is back

9) **Re-sampling**: A final touch might be to re-sample to data in order to save space and later in time complexity. You could do this by going to **Tools** and choosing **Change sampling rate**.

ICA, RANSAC and ASR

1) **ICA**: After we cleaned our data, we can use several algorithms to utilize it(ICA) or continue cleaning it(ASR, RANSAC). One such algorithm is ICA. ICA allows us seperate our data from all the electrodes to different independent components such as eye blinks. For more info: `https://sccn.ucsd.edu/wiki/Chapter_09:_Decomposing_Data_Using_ICA`

2) To preform ICA, go to **Tools** and then choose **Decompose data by ICA**. See **Figure 32**.

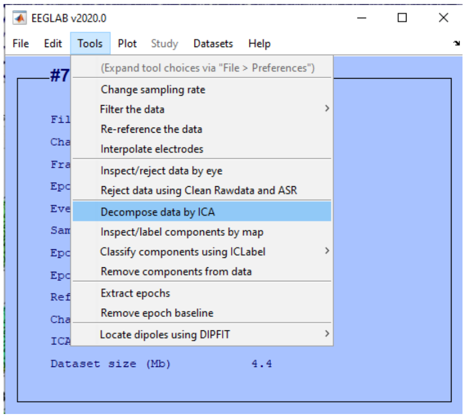
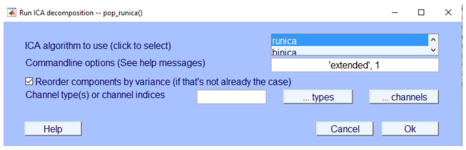


Figure 32: Decompose data by ICA

3) In the window that opens click ok(this might take some time so be patient, you can see the progress of the ICA training in the **MATLAB Command Window**). See **Figure 33**.



ICA algorithm to use (click to select): runica / binica

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

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

Channel type(s) or channel indices:

Figure 33: ICA

4) Next, we can plot the different ICA components using a scroll or components maps. See **Figure 34** and **Figure 35**.

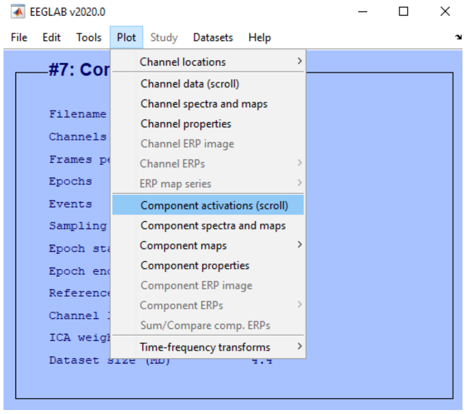


Figure 34: Scroll

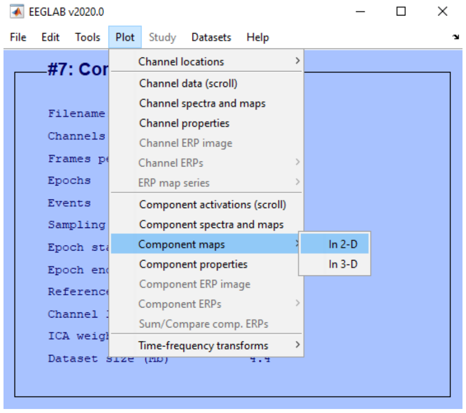
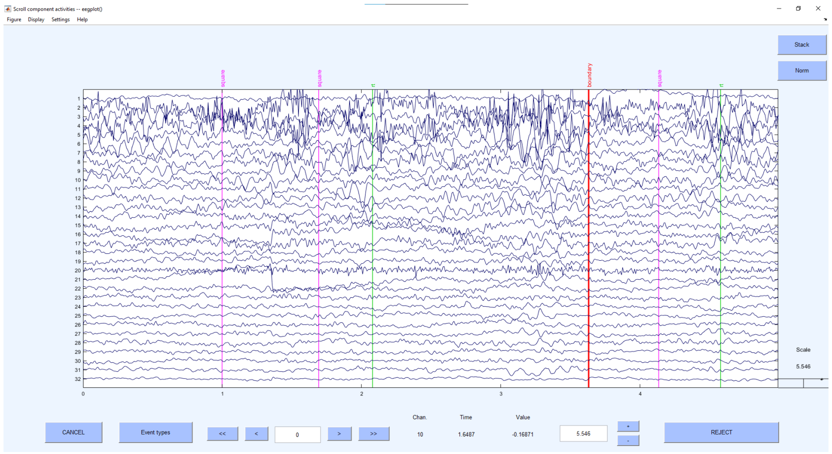


Figure 35: Components maps

5) In the scroll plot, we can see at the left that now instead of electrodes we have the ICA components by number. See **Figure 36**.



Chan. 10 Time 1.6487 Value -0.16871

0

5.546

Figure 36: Scroll plot

6) In the components maps plot we can see a map plot of each components. See **Figure 37**. For example, we can see the component 20 is hot in the front of the face, this might be from eye blinks or from other muscles of the face- therefore we might want to delete that component. For more info see the wiki.

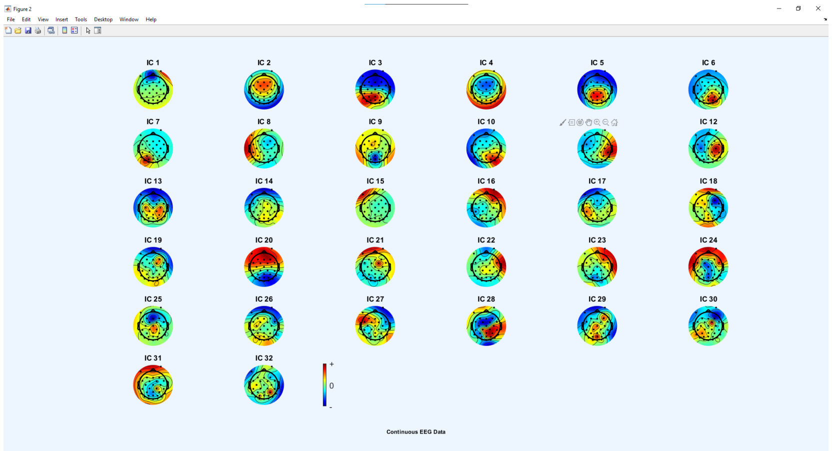


Figure 37: Components maps plot

7) **ASR & RANSAC**: The ASR and RANSAC algorithms might help you automatically clean noisy data. For more info, see the wiki.

Getting the code & Saving the dataset

1) **Exporting to code**: To export your actions into code, you can write ”eegh” in the **Command Window** of **MATLAB**. You might want to preform the actions without plotting as these actions will be exported as well.

2) **Saving the dataset**: After you are finished, you can save your dataset. To save your dataset, go to **File** and choose one of several save options. See **Figure 38**.

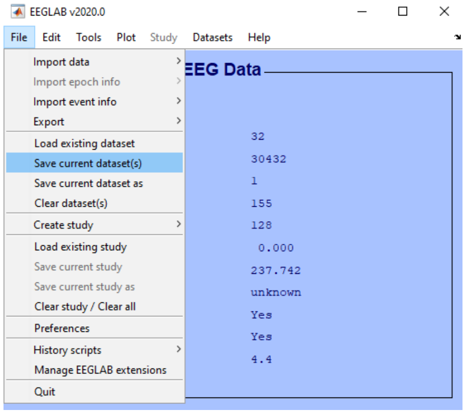


Figure 38: Saving the dataset

Good luck and enjoy from Assaf & Dor