SPIKE
COMPUTATIONAL
NEUROSCIENCE
COMMUNITY

EEG Processing:

An Entry to the World of Brain Waves

Topic:

Preprocessing EEG Signals Using EEGLAB

Author: Mohammadreza Shahsavari

Contact: mohamadrezashahsavary@gmail.com

What you are going to learn in this session?

Welcome to the session on EEGLAB! In this session, you will learn how to use EEGLAB, a powerful tool for processing and analyzing EEG data within MATLAB. We will start by introducing EEGLAB, its importance, and its key features, including a user-friendly interface and advanced analysis tools. You will learn how to start and navigate EEGLAB, import EEG data from various formats, and manage datasets. We will cover how to incorporate event markers and channel locations, and the importance of visualizing channel locations for accurate data mapping. You will understand EEG referencing, common reference types, and the process of re-referencing data to improve quality. We will discuss resampling and filtering techniques, including removing DC shifts and applying bandpass filters. You'll learn to identify and remove bad data and channels, ensuring data integrity. Lastly, we will explore spectral analysis tools to inspect the spectral properties of EEG channels, helping you identify and address abnormal activities in your data. By the end of this session, you will be well-equipped to preprocess and analyze EEG data using EEGLAB, laying a strong foundation for advanced EEG research.

- *Mohammadreza Shahsavari*

Table of Contents

1. What is EEGLAB and Why it is Important?

EEGLAB is a widely used open-source software environment for processing and analyzing electroencephalographic (EEG) data. Developed as a MATLAB toolbox, EEGLAB provides a comprehensive suite of tools for the visualization, preprocessing, and analysis of EEG data.

Key Features:

1. **User-Friendly Interface**: EEGLAB offers a graphical user interface (GUI) that simplifies the management and analysis of complex EEG datasets, making it accessible even to those with limited programming experience.
2. **Extensive Preprocessing Capabilities**: The software includes robust functions for data filtering, artifact rejection, and data segmentation, which are essential for preparing raw EEG data for analysis.
3. **Advanced Analysis Tools**: EEGLAB supports various advanced techniques such as Independent Component Analysis (ICA) for source separation, time-frequency analysis, and statistical measures. These tools enable researchers to extract meaningful patterns and insights from EEG data.
4. **Visualization**: The software provides multiple visualization options, including scalp maps, time-series plots, and 3D brain activity maps, helping researchers to interpret their findings effectively.
5. **Community and Extensions**: EEGLAB has a large and active user community that contributes to its continuous development and expansion. Numerous plugins are available, extending its functionality to cover a wide range of EEG analysis methods.

2. Overview of EEG Data Processing Using EEGLAB

Processing EEG data involves a series of methodical steps to ensure the data is clean and suitable for analysis. The following guide outlines these essential steps, providing a clear pathway for researchers to follow using EEGLAB, a powerful tool for EEG data analysis.

Steps Explained

1. **Data Collection**
    - The process begins with the collection of EEG data. This involves recording the brain's electrical activity using an EEG device. It is crucial to perform the recording in a controlled environment to minimize noise and artifacts.
2. **Importing Data into EEGLAB**
    - Once the EEG data is collected, the next step is to import it into EEGLAB. This software supports various file formats, allowing researchers to load their data seamlessly. Importing the data into EEGLAB is a foundational step that sets the stage for subsequent processing.
3. **Incorporating Event Markers and Channel Locations**
    - Event markers, which denote specific events or stimuli during the recording, and channel locations, which map the electrodes' spatial configuration, are essential components that need to be imported. This information is crucial for accurate analysis and interpretation of the EEG data.
4. **Re-referencing and Down-sampling**
    - Depending on the specific requirements of the analysis, re-referencing the data to a different electrode or down-sampling to a lower rate may be necessary. These adjustments help in managing the data more effectively and can enhance the clarity of the signals.
5. **Applying a High Pass Filter**

- A high pass filter, typically in the range of 0.5 to 1 Hz, is applied to remove slow drifts and trends in the data. This step is essential for focusing on the frequencies of interest and ensuring that the data is free from low-frequency noise.

6. **Examining Raw Data**
   - Visual inspection of the raw EEG data is a critical step. By examining the data, researchers can identify any obvious artifacts or anomalies that need to be addressed. This initial inspection helps in planning the subsequent cleaning steps.
7. **Identifying and Rejecting Bad Channels**
   - Channels that exhibit excessive noise or artifacts need to be identified and rejected. These bad channels can distort the overall data and negatively impact the results of the analysis. Identifying them early helps in maintaining data integrity.
8. **Rejecting Large Artifact Time Points**
   - Removing segments of data that contain large artifacts, such as those caused by eye blinks or muscle movements, is crucial. This step ensures that the remaining

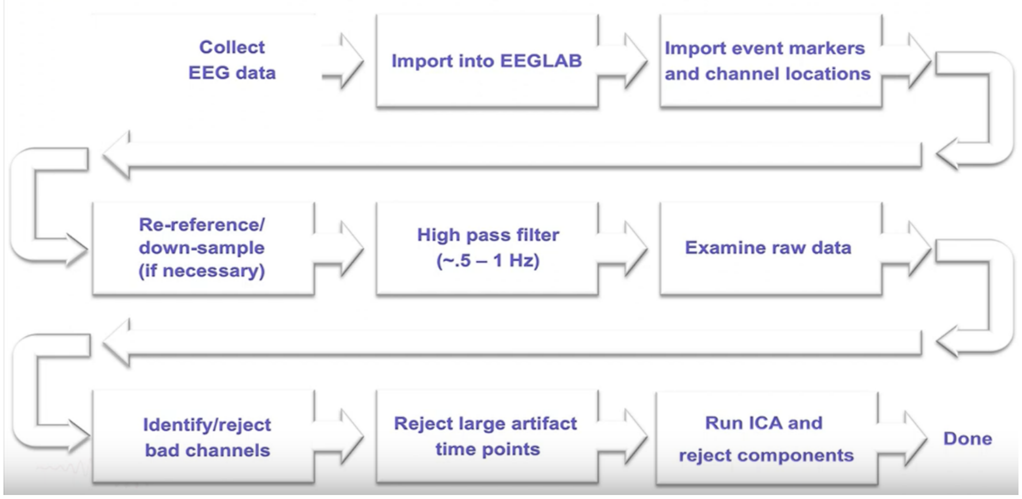

Collect EEG data
Import into EEGLAB
Import event markers and channel locations
Re-reference/ down-sample (if necessary)
High pass filter (~.5 – 1 Hz)
Examine raw data
Identify/reject bad channels
Reject large artifact time points
Run ICA and reject components
Done

Fig 1. Flowchart depicting the sequential steps for processing EEG data using EEGLAB. Each step is crucial for ensuring clean and reliable data for analysis.

3. Starting EEGLAB

Starting EEGLAB within the MATLAB environment involves a few straightforward steps. These steps ensure that EEGLAB is properly launched and ready for EEG data processing.

Steps to Launch EEGLAB

1. **Navigate to the EEGLAB Directory**
   - ○ Begin by opening MATLAB. In the command window, navigate to the directory where EEGLAB is installed. This can be done by typing the path to the EEGLAB folder in the command line or by using the graphical interface to browse to the appropriate directory.

```
>> # In the MATLAB command line
>> cd 'path_to_eeglab_directory'
```

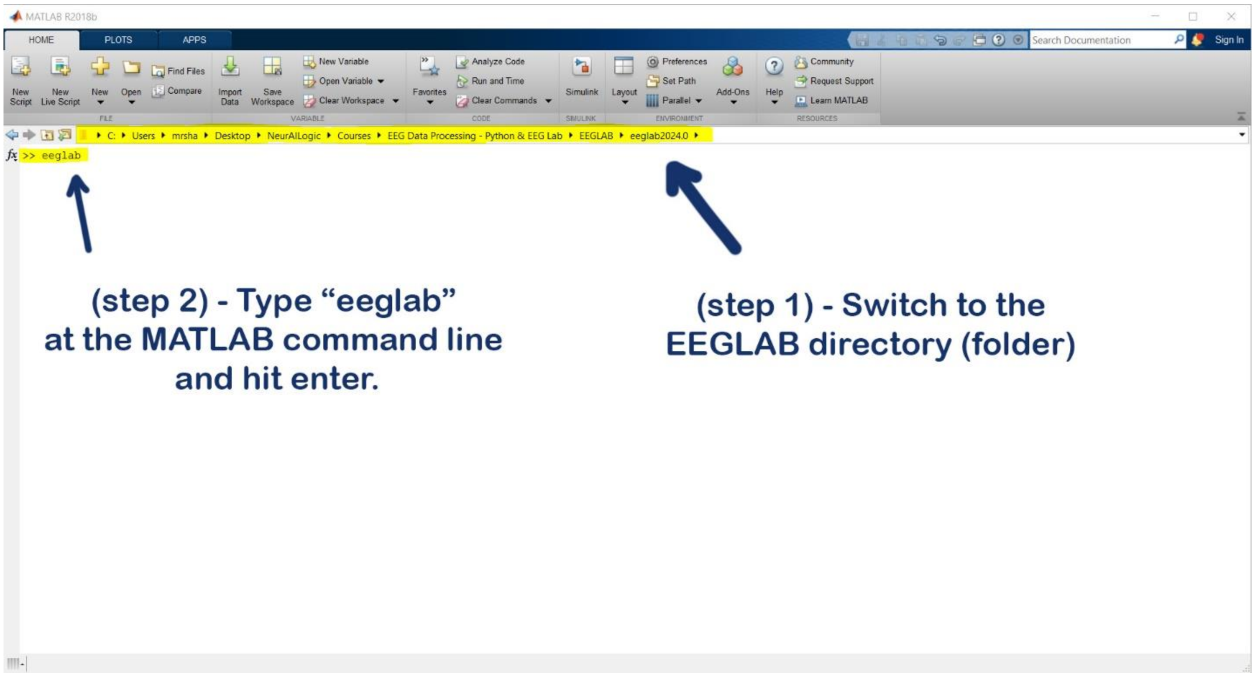
2. **Run EEGLAB**

- Once you are in the correct directory, initialize EEGLAB by typing eeglab into the MATLAB command line and pressing Enter. This command launches the EEGLAB graphical user interface (GUI), from which you can perform various EEG data processing tasks.

```
>> eeglab
```

By following these steps, users can easily set up and launch EEGLAB within MATLAB, enabling them to proceed with their EEG data analysis tasks.

Fig 2. Depicts how the EEGLAB software should be lunched.



```
C: ▸ Users ▸ mrsha ▸ Desktop ▸ NeurAILogic ▸ Courses ▸ EEG Data Processing - Python & EEG Lab ▸ EEGLAB ▸ eeglab2024.0 ▸
```

```
>> eeglab
```

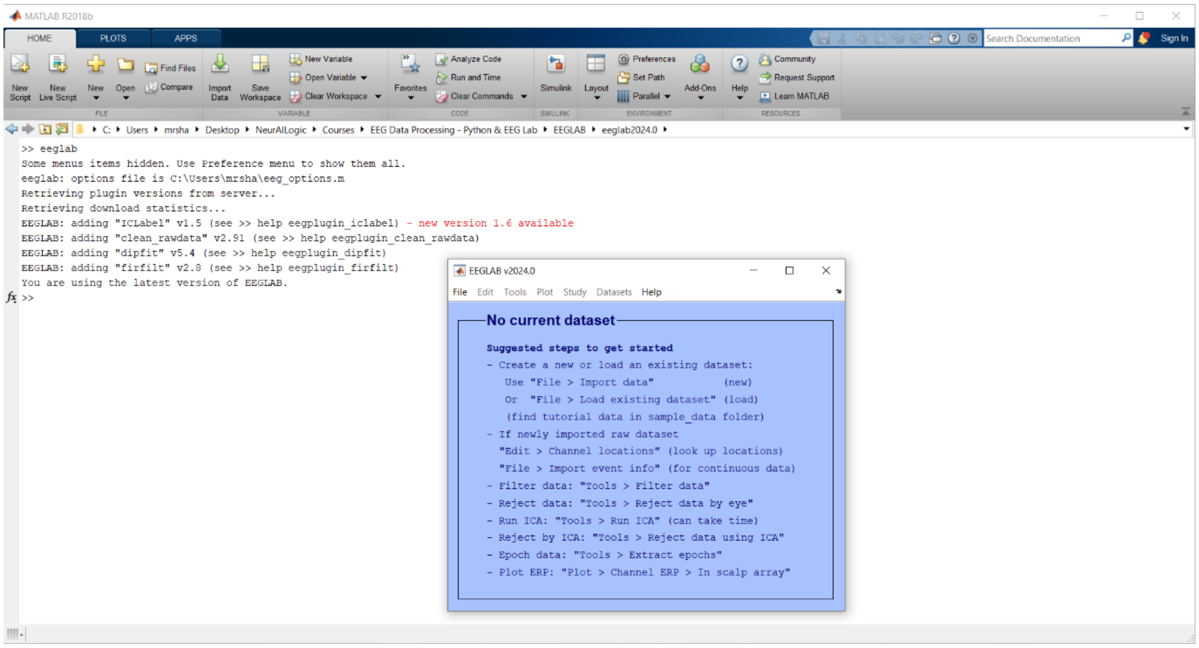
(step 2) - Type “eeglab” at the MATLAB command line and hit enter.

(step 1) - Switch to the EEGLAB directory (folder)

Fig 2. Steps to launch EEGLAB in MATLAB. First, navigate to the EEGLAB directory, then type ‘eeglab’ in the command line to initialize the software.

After launching EEGLAB, a new window pops up displaying the main EEGLAB interface with the message "No current dataset" and a list of suggested steps to get started.

Fig 3. shows this window that appears after launching EEGLAB.



```
>> eeglab
Some menus items hidden. Use Preference menu to show them all.
eeglab: options file is C:\Users\mrsha\eeg_options.m
Retrieving plugin versions from server...
Retrieving download statistics...
EEGLAB: adding "ICLabel" v1.5 (see >> help eegplugin_iclabel) - new version 1.6 available
EEGLAB: adding "clean_rawdata" v2.91 (see >> help eegplugin_clean_rawdata)
EEGLAB: adding "dipfit" v5.4 (see >> help eegplugin_dipfit)
EEGLAB: adding "firfilt" v2.8 (see >> help eegplugin_firfilt)
You are using the latest version of EEGLAB.
fx >>
```

```
No current dataset

Suggested steps to get started
- Create a new or load an existing dataset:
   Use "File > Import data"           (new)
   Or  "File > Load existing dataset" (load)
   (find tutorial data in sample_data folder)
- If newly imported raw dataset
  "Edit > Channel locations" (look up locations)
  "File > Import event info" (for continuous data)
- Filter data: "Tools > Filter data"
- Reject data: "Tools > Reject data by eye"
- Run ICA: "Tools > Run ICA" (can take time)
- Reject by ICA: "Tools > Reject data using ICA"
- Epoch data: "Tools > Extract epochs"
- Plot ERP: "Plot > Channel ERP > In scalp array"
```

Fig 3. Initial EEGLAB interface in MATLAB, providing a list of suggested steps to help users get started with their EEG data processing.

4. Importing Data into EEGLAB

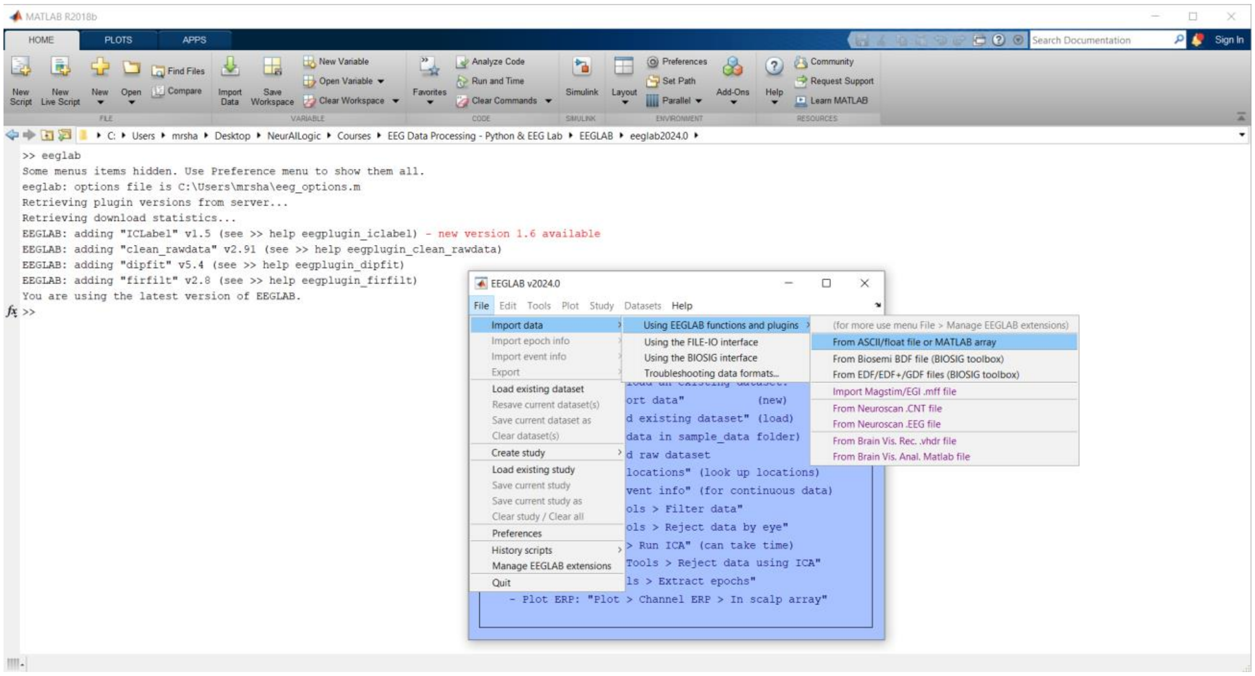
EEGLAB provides several ways to import EEG data. Here's how you can get started:

One easy method is to use the built-in EEGLAB functions and plugins. These tools help you import data from various EEG recording systems. You just need to select the right plugin for your data.

If your data is stored in ASCII or MATLAB formats, you can also import it into EEGLAB. To do this, go to the **File** menu, select **Import data**, and choose either ASCII or MATLAB format. This will load your data into EEGLAB without any trouble.

Another useful feature in EEGLAB is the BIOSIG toolbox. This toolbox can handle many different biomedical signal file formats. By selecting **Using the BIOSIG interface** from the **File** menu, you can import data from a wide range of EEG devices.

EEGLAB also has special options for data collected with Neuroscan or EGI systems. If you have data from Neuroscan, choose **From Neuroscan .CNT file**. For EGI data, select **From EGI .RAW file**. These options make it easy to load these specific types of data directly into EEGLAB.



```
>> eeglab
Some menus items hidden. Use Preference menu to show them all.
eeglab: options file is C:\Users\mrsha\eeg_options.m
Retrieving plugin versions from server...
Retrieving download statistics...
EEGLAB: adding "ICLabel" v1.5 (see >> help eegplugin_iclabel) - new version 1.6 available
EEGLAB: adding "clean_rawdata" v2.91 (see >> help eegplugin_clean_rawdata)
EEGLAB: adding "dipfit" v5.4 (see >> help eegplugin_dipfit)
EEGLAB: adding "firfilt" v2.8 (see >> help eegplugin_firfilt)
You are using the latest version of EEGLAB.
>>
```

Fig 4. Different methods for importing data into EEGLAB, providing flexibility for various file formats and sources.

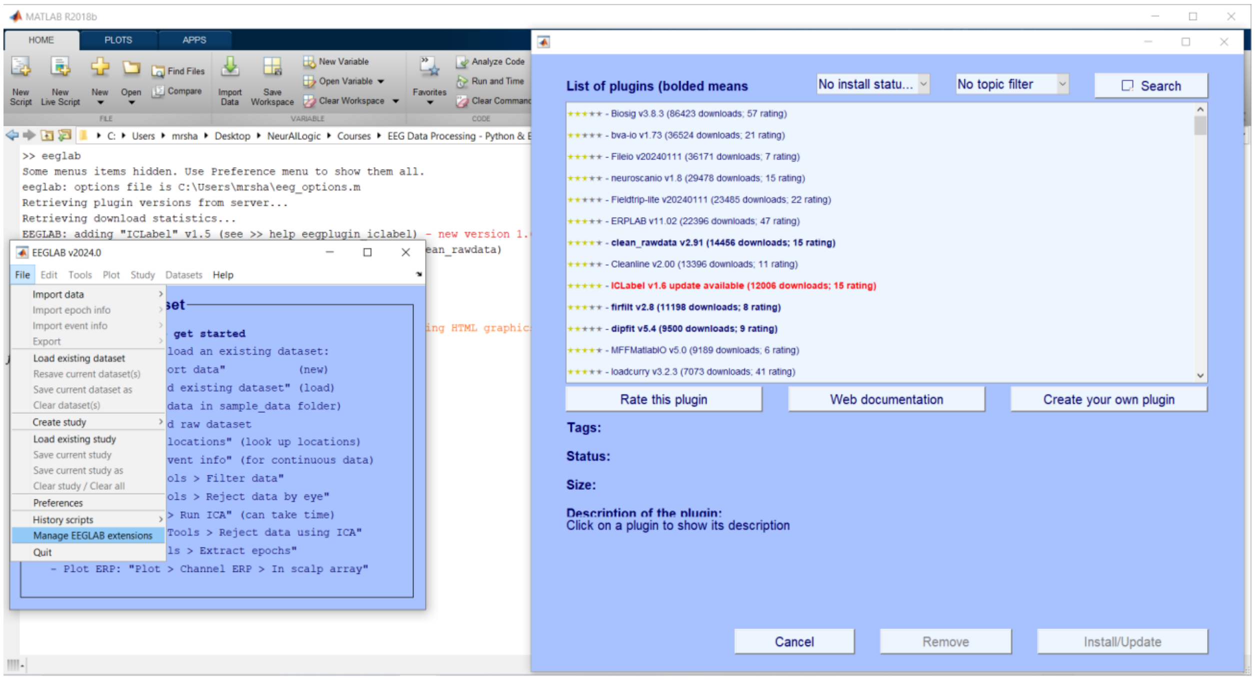
If you cannot import your EEG data using the built-in methods, you can add extra extensions to EEGLAB. These extensions will help you load data from other formats.

To install extensions, go to the **File** menu and click on **Manage EEGLAB extensions**. This will open a window showing a list of available plugins.

In this window, you will see many plugins that you can install to expand what EEGLAB can do. Each plugin has a description explaining what it is for and which file formats it supports.

To add a new plugin, click on it in the list and then click **Install/Update**. This will install the plugin and allow EEGLAB to import additional types of data.

Each plugin entry also has details and links to online documentation. This information helps you understand what each plugin does and how to use it.



```
>> eeglab
Some menus items hidden. Use Preference menu to show them all.
eeglab: options file is C:\Users\mrsha\eeg_options.m
Retrieving plugin versions from server...
Retrieving download statistics...
EEGLAB: adding "ICLabel" v1.5 (see >> help eegplugin_iclabel) - new version 1.
                                                              ean_rawdata)

                                                              ing HTML graphic
```

```
set
 get started
load an existing dataset:
ort data"              (new)
d existing dataset" (load)
data in sample_data folder)
d raw dataset
locations" (look up locations)
vent info" (for continuous data)
ols > Filter data"
ols > Reject data by eye"
> Run ICA" (can take time)
Tools > Reject data using ICA"
ls > Extract epochs"
  - Plot ERP: "Plot > Channel ERP > In scalp array"
```

List of plugins (bolded means

★★★★★ - Biosig v3.8.3 (86423 downloads; 57 rating)
★★★★★ - bva-io v1.73 (36524 downloads; 21 rating)
★★★★★ - Fileio v20240111 (36171 downloads; 7 rating)
★★★★★ - neuroscanio v1.8 (29478 downloads; 15 rating)
★★★★★ - Fieldtrip-lite v20240111 (23485 downloads; 22 rating)
★★★★★ - ERPLAB v11.02 (22396 downloads; 47 rating)
★★★★★ - **clean_rawdata v2.91 (14456 downloads; 15 rating)**
★★★★★ - Cleanline v2.00 (13396 downloads; 11 rating)
★★★★★ - **ICLabel v1.6 update available (12006 downloads; 15 rating)**
★★★★★ - **firfilt v2.8 (11198 downloads; 8 rating)**
★★★★★ - **dipfit v5.4 (9500 downloads; 9 rating)**
★★★★★ - MFFMatlabIO v5.0 (9189 downloads; 6 rating)
★★★★★ - loadcurry v3.2.3 (7073 downloads; 41 rating)

Tags:

Status:

Size:

Description of the plugin:
Click on a plugin to show its description

Fig 5. Installing extra extensions in EEGLAB to support additional data formats and enhance import capabilities.

If you have already processed your data using EEGLAB and saved it, you can easily continue working on the same data and pick up where you left, or share it with others. Here’s how to import a previously saved EEGLAB dataset:

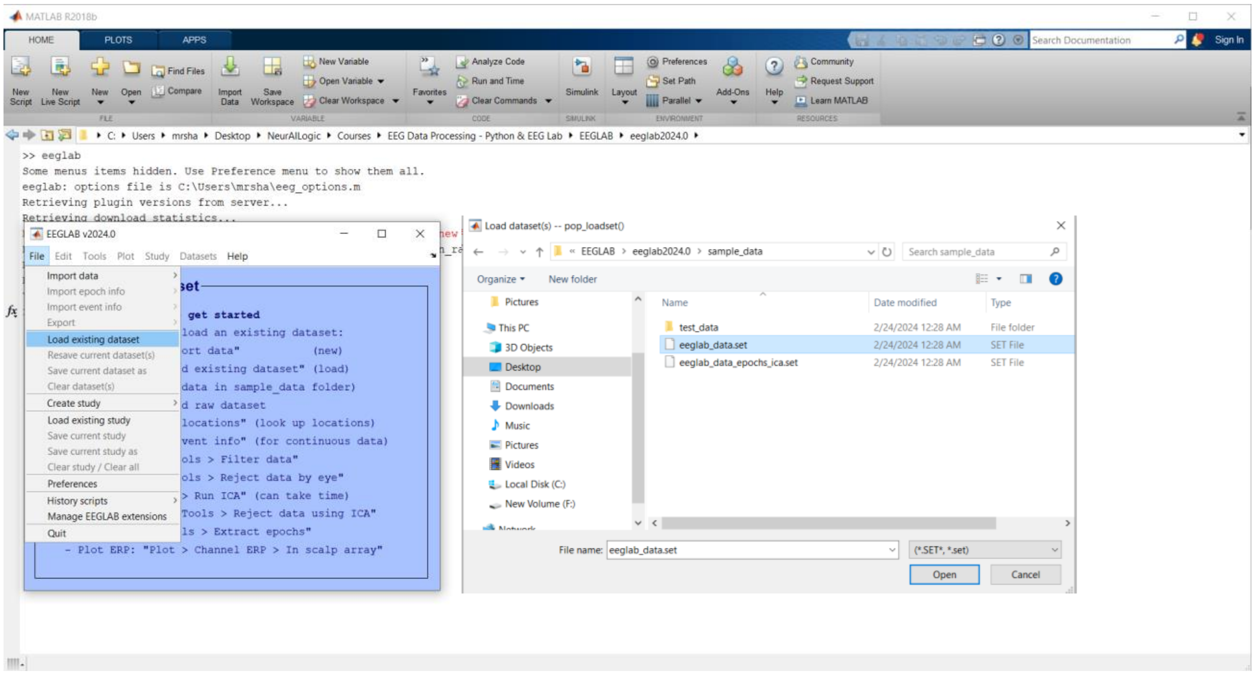
- To import a saved dataset, start by going to the **File** menu in the main EEGLAB window.
- Select **Load existing dataset** from the menu. This option allows you to bring in a dataset that you have previously saved.
- A file dialog will open. Navigate to the location where your EEGLAB dataset (**.set file**) is stored. Select the file and click Open.

EEGLAB datasets are saved with the .set file extension. These files include the data and information about preprocessing steps, event markers, and channel locations.

Using a saved dataset saves time because it allows you to continue from where you left off without repeating data import and preprocessing steps. This is especially useful for large datasets or complex preprocessing workflows.

EEGLAB also supports working with multiple datasets in a single session. This feature is helpful for comparing different conditions or subjects without needing to reload data each time.

By following these steps, you can efficiently import a saved EEGLAB dataset and continue your EEG data analysis smoothly.



```
>> eeglab
Some menus items hidden. Use Preference menu to show them all.
eeglab: options file is C:\Users\mrsha\eeg_options.m
Retrieving plugin versions from server...
Retrieving download statistics...
```

```
set
 get started
load an existing dataset:
ort data"           (new)
d existing dataset" (load)
data in sample_data folder)
d raw dataset
locations" (look up locations)
vent info" (for continuous data)
ols > Filter data"
ols > Reject data by eye"
> Run ICA" (can take time)
Tools > Reject data using ICA"
ls > Extract epochs"
  - Plot ERP: "Plot > Channel ERP > In scalp array"
```

File name: eeglab_data.set (*.SET*, *.set)

Fig 6. Importing a saved EEGLAB dataset by selecting 'Load existing dataset' from the File menu and the dataset's .set file.

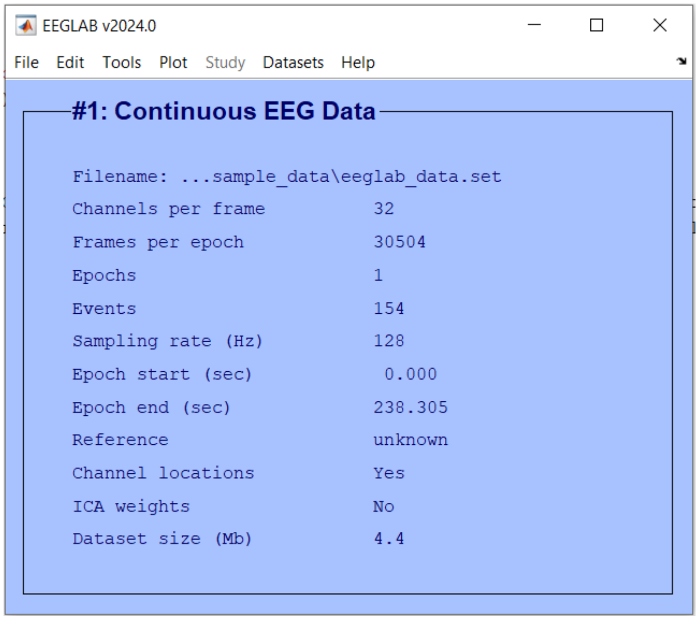
After loading your EEG data into EEGLAB, the software displays detailed information about the dataset. This summary helps you understand the key properties of your data (see Figure 7).

Once the dataset is loaded, a window appears showing important details about the data. Here are some of the key pieces of information provided:

- **Filename:** This shows the path and name of the dataset file that you have loaded, helping you confirm that the correct file is being used.
- **Channels per Frame:** This indicates the number of EEG channels recorded. In this example, there are 32 channels.
- **Frames per Epoch:** This number shows how many data points (frames) are in each epoch. For continuous data, this can be a large number since it includes all data points recorded over time.

- **Epochs:** This represents the number of epochs in the dataset. In this case, there is only 1 epoch since the data is continuous.
- **Events:** The total number of events or markers in the dataset is shown here. Events mark specific points of interest, such as stimuli or responses.
- **Sampling Rate (Hz):** This is the frequency at which the data was recorded. In this example, the data was sampled at 128 Hz.
- **Epoch Start and End (sec):** These values indicate the time range of each epoch. Here, it starts at 0 seconds and ends at approximately 238.305 seconds.
- **Reference:** This field shows the reference used for the recording. It is marked as "unknown" in this example, indicating that the reference was not specified.
- **Channel Locations:** This indicates whether channel location information is included in the dataset. In this example, channel locations are available.
- **ICA Weights:** This shows whether Independent Component Analysis (ICA) weights are included in the dataset. In this example, ICA weights are not present.
- **Dataset Size (MB):** The size of the dataset file is shown here. In this example, the file size is 4.4 MB.

Understanding these details is crucial for ensuring that the dataset is correctly loaded and ready for analysis. It provides a quick overview of the dataset's structure and properties.



#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
```

Fig 7. Dataset information displayed in EEGLAB after loading the data, showing details such as the number of channels, sampling rate, and epoch duration.

After loading your EEG data into EEGLAB, the first thing you might want to do is visualize the data. This helps you get an initial look at the EEG recordings and identify any obvious artifacts or issues.

Steps to Visualize EEG Data

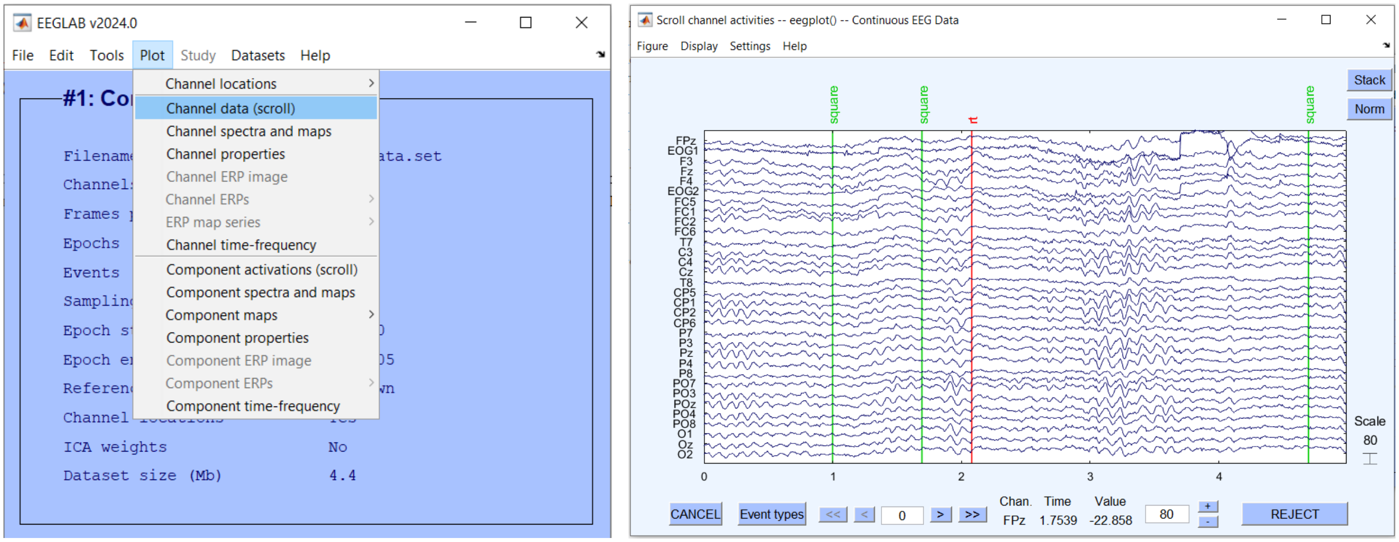
1. **Open the Plot Menu**
   - From the main EEGLAB window, go to the Plot menu at the top (see Figure 8).
2. **Select Channel Data (Scroll)**
   - In the Plot menu, select Channel data (scroll). This option opens a new window that allows you to scroll through the EEG data and examine the signals from each channel.

Viewing the Data

Once you select Channel data (scroll), a window will appear displaying the EEG data. Here are some key features of this view:

- **Channel Signals:** The window shows the signals from each EEG channel. The channels are listed on the left, and the signals are plotted over time.
- **Event Markers:** Event markers, such as stimuli or responses, are indicated with vertical lines. These markers help you see when events occurred during the recording.
- **Scrolling and Scaling:** You can scroll through the data to view different time segments. You can also adjust the scale to zoom in or out on the data, making it easier to see details.
- **Rejecting Artifacts:** If you notice any segments with significant artifacts, you can mark them for rejection. This helps in cleaning the data for further analysis.

Visualizing the EEG data in this way is a crucial step in the preprocessing workflow. It allows you to inspect the data quality and make initial decisions about data cleaning and artifact rejection (see Figure 8).



(a) (b)

Fig 8. (a) Accessing the Channel Data (scroll) option from the Plot menu in EEGLAB to visualize EEG data. (b) Scrolling through EEG data in EEGLAB, showing signals from multiple channels and event markers.

5. Incorporating Event Markers and Channel Locations

Importing Event Information

After visualizing your EEG data, the next important step is to import event information. Events mark specific points of interest, such as stimuli or responses, within your EEG recordings. EEGLAB offers several ways to import events, including importing events from a specific channel in the data.

There are several ways to import event information into EEGLAB. These options are available under the **File** menu by selecting **Import event info**. Here, we demonstrate how events can be imported from a data channel.

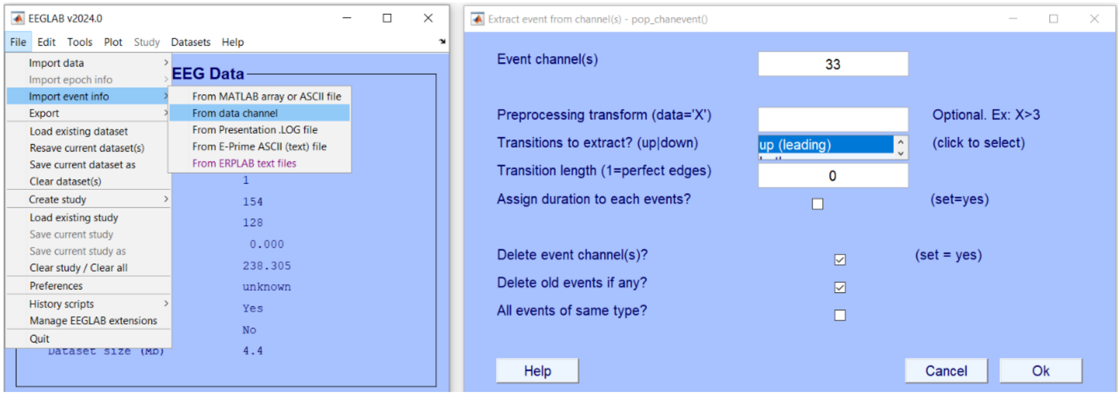
Steps to Import Events from a Channel

6. **Open the Import Event Menu**
   - From the main EEGLAB window, go to the **File** menu and select **Import event info** (see Figure 9).
7. **Select Event Source**
   - Choose **From data channel** from the list of options. This method allows you to extract events from a designated channel within your EEG data.
8. **Configure Event Extraction**
   - A new window will appear where you can configure the event extraction settings (see Figure 9).
   - **Event Channels:** Enter the channel number from which you want to extract events. This is typically a channel specifically used to record event markers during the EEG session.
   - **Preprocessing Transform:** If necessary, apply a transformation to the data before extracting events. This can be useful if the event markers are not in a straightforward format.
   - **Transitions to Extract:** Specify whether to extract upward (leading) or downward (trailing) transitions. This setting helps EEGLAB identify the points where events occur.
   - **Transition Length:** Set the transition length to define how precise the event detection should be.
   - **Assign Duration to Each Event:** If your events have a duration, you can specify this here.

- **Additional Settings:** You can choose to delete the event channels after extraction, remove any old events, and confirm if all events are of the same type.

9. **Execute Event Extraction**

- After configuring the settings, click **OK** to extract the events. EEGLAB will process the specified channel and generate events based on your configurations.



Event channel(s): 33

Preprocessing transform (data='X'): Optional. Ex: X>3

Transitions to extract? (up|down): up (leading) (click to select)

Transition length (1=perfect edges): 0

Assign duration to each events? □ (set=yes)

Delete event channel(s)? ☑ (set = yes)

Delete old events if any? ☑

All events of same type? □

Fig 9. Importing event information into EEGLAB from a specific data channel, including configuring event extraction settings and managing events.

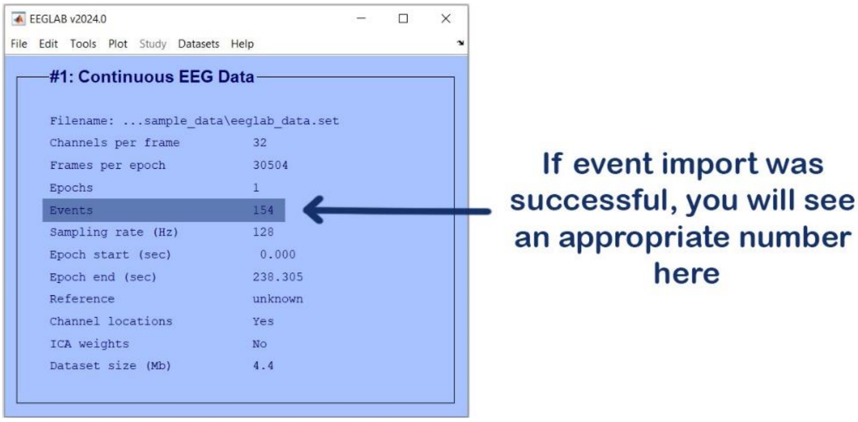
After importing event information into EEGLAB, it is important to verify that the events were successfully imported. This can be easily checked within the dataset information display.

Checking Event Import

Once the events are imported, look at the main EEGLAB window where the dataset information is displayed (see Figure 10).

In the dataset information window, find the line labeled Events. This line shows the total number of events detected in your EEG data.

If the event import was successful, you will see an appropriate number next to Events. This number should match the total events you expected to import from the specified channel.



#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
```

If event import was successful, you will see an appropriate number here

Fig 10. Verifying the successful import of events in EEGLAB by checking the event count in the dataset information display.

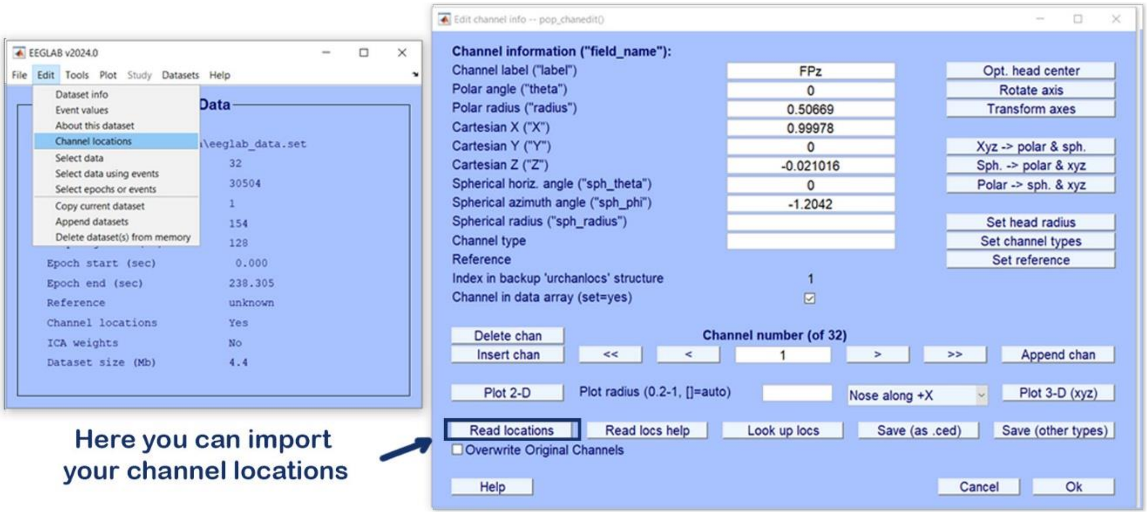
Importing Channel Locations in EEGLAB

After verifying the event import, the next step is to import channel locations. Importing channel locations is crucial because it maps the electrodes to their precise spatial positions on the scalp. This step is essential for accurate data analysis, including source localization, topographic mapping, and interpretation of the EEG signals relative to the brain regions they originate from.

Steps to Import Channel Locations

1. **Open the Edit Menu**
   - From the main EEGLAB window, go to the **Edit** menu and select Channel locations (see Figure 11). This option allows you to specify the exact locations of the EEG electrodes.

2. **Read Locations**
   - In the channel locations window, click on **Read locations**. This will open a dialog where you can select the file that contains your channel location information. The file usually includes the 3D coordinates of each electrode, which is necessary for visualizing and analyzing EEG data accurately.
3. **Select Channel Location File**
   - Navigate to the location of your channel location file. This file is typically in **.ced** or another supported format. Select the file and click **Open**. The channel location file maps each channel to a specific position on the scalp, ensuring that the data is correctly interpreted spatially.
4. **Verify and Apply Locations**
   - After selecting the file, the channel information will be displayed in the window. Verify the information to ensure it is correct. Look at the labels and positions to confirm they match your setup. Then, click OK to apply the channel locations to your dataset. This step integrates the spatial information with your EEG data, enabling precise topographic analysis and visualization.



```
\eeglab_data.set
                          32
                          30504
                          1
                          154
                          128
Epoch start (sec)          0.000
Epoch end (sec)           238.305
Reference                 unknown
Channel locations         Yes
ICA weights               No
Dataset size (Mb)         4.4
```

Channel information ("field_name"):

Channel label ("label")	FPz
Polar angle ("theta")	0
Polar radius ("radius")	0.50669
Cartesian X ("X")	0.99978
Cartesian Y ("Y")	0
Cartesian Z ("Z")	-0.021016
Spherical horiz. angle ("sph_theta")	0
Spherical azimuth angle ("sph_phi")	-1.2042
Spherical radius ("sph_radius")	
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

Here you can import your channel locations

Fig 11. Importing channel locations in EEGLAB by selecting 'Channel locations' from the Edit menu and loading the channel location file.

Visualizing Channel Locations in EEGLAB

Importing channel locations is crucial for accurate EEG data analysis. Once the channel locations are imported, it is important to visualize them to ensure that they are correctly mapped. Visualization helps confirm the spatial arrangement of electrodes and ensures that the data will be accurately interpreted.

Steps to Visualize Channel Locations

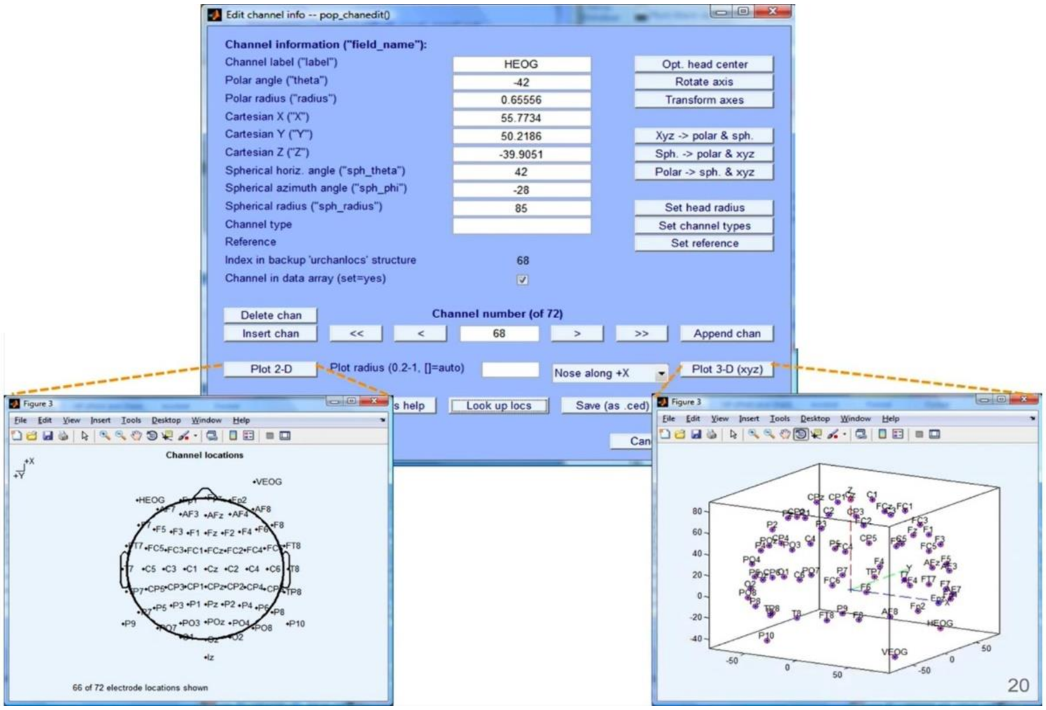
1. **Open the Channel Locations Window**
   - After importing the channel locations, the channel information window displays detailed coordinates and labels for each electrode (see Figure 12).
2. **Plot 2D Locations**
   - In the channel information window, click Plot 2-D. This option generates a 2D topographic map of the electrode positions. The 2D view is useful for verifying the relative positions of electrodes on a flat plane, often resembling a scalp map used in EEG recordings.
3. **Plot 3D Locations**
   - To get a more comprehensive view, click Plot 3-D. This option generates a 3D plot of the electrode positions. The 3D view provides a spatial representation of electrode locations, which is essential for understanding the geometry of the electrode array in three dimensions.

Importance of Visualization

- **Verification:** Visualizing the channel locations helps ensure that the electrodes are correctly positioned according to the standard montage. Any discrepancies can be quickly identified and corrected.
- **Analysis Accuracy:** Accurate channel location mapping is essential for advanced analyses such as source localization and connectivity mapping. Correct spatial

positioning allows for precise interpretation of the EEG signals relative to brain regions.

- **Data Quality:** By confirming the electrode positions visually, you can ensure the quality and reliability of your data. This step is vital before proceeding with further data processing and analysis.



Edit channel info -- pop_chanedit()

Channel information ("field_name"):
Channel label ("label"): HEOG
Polar angle ("theta"): -42
Polar radius ("radius"): 0.65556
Cartesian X ("X"): 55.7734
Cartesian Y ("Y"): 50.2186
Cartesian Z ("Z"): -39.9051
Spherical horiz. angle ("sph_theta"): 42
Spherical azimuth angle ("sph_phi"): -28
Spherical radius ("sph_radius"): 85
Channel type:
Reference
Index in backup 'urchanlocs' structure: 68
Channel in data array (set=yes): ☑
Channel number (of 72): 68
Plot radius (0.2-1, []=auto):
Nose along +X

Fig 12. Visualizing channel locations in EEGLAB with 2D and 3D plots to confirm accurate electrode mapping.

6. EEG Reference and Re-referencing

In EEG recordings, the reference is the electrode against which the voltage at each of the recording electrodes is measured. The choice of reference electrode can significantly affect the data and its interpretation. Common types of references include:

Single Electrode Reference: A single electrode, often placed on the mastoid bone (the bony area behind the ear) or earlobe, is used as a reference. This method is straightforward and easy to implement. However, it might pick up noise or artifacts from the location where it is placed, which can affect the overall EEG data quality.

Average Reference: The average reference uses the average of all electrodes as the reference point. This method tends to provide a balanced view of brain activity by distributing the reference evenly across the entire scalp. It reduces the bias that might be introduced by using a single electrode reference and helps in getting a more global perspective of the brain's electrical activity.

Linked Mastoid Reference: In this method, the average of the signals from electrodes placed on both mastoids is used as the reference. This approach helps to reduce the noise that might be picked up if only one mastoid was used. It is commonly used because it provides a relatively neutral reference point that does not strongly favor any one area of the scalp.

Cz Reference: The Cz reference uses the electrode placed at the vertex of the head (Cz) as the reference. This central location is often chosen because it is equidistant from many recording sites, which can help in balancing the recorded potentials across the scalp. However, like any single electrode reference, it can still be subject to location-specific noise or artifacts.

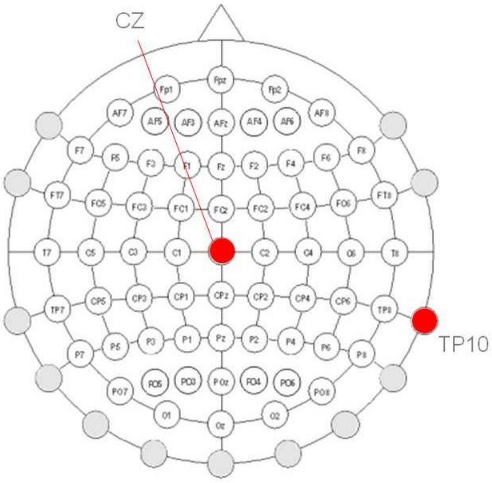


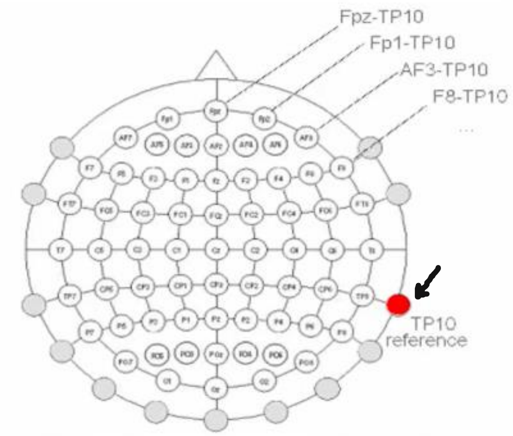
Fig 13. Illustration of common reference points in EEG recording on the scalp. The diagram highlights the locations of reference electrodes CZ and TP10, which are the most frequently used for EEG recoding.

There is no ‘best’ common reference site. Some researchers claim that non-scalp references (earlobes, nose) introduce more noise than a scalp channel reference though this has not been proven to our knowledge. If the data have been recorded with a given reference, they can usually be re-referenced (inside or outside EEGLAB) to any other reference channel or channel combination.

The concept of average referencing in EEG involves recalculating the reference point based on the average of all electrode activities. This ensures a balanced and neutral reference point across the entire scalp, minimizing bias from any single electrode. The average reference assumes that the sum of the potentials at all electrodes equals zero (see Fig. 14). The advantage of the average reference is based on Ohm’s law, which states that the sum of outward positive and negative currents across an isolated sphere should be zero. This is because the positive and negative currents across the scalp should balance each other out (see Fig. 15).

By using the average reference, we minimize the bias from any single electrode. This leads to a more balanced and accurate representation of the brain's electrical activity.

It is worth mentioning that calculating the average reference of the data is recommended prior to using ICA or performing source localization.



1. Average Reference assumption

$$Fpz + Fp1 + AF3 + F8 + FT8 + \dots + TP10 = 0$$

2. Recalculate the activity at reference TP10

If SE is the sum of all electrode activity

$$SE = Fpz + Fp1 + AF3 + F8 + \dots - 64TP10$$

$$SE = \underbrace{Fpz + Fp1 + AF3 + F8 + \dots + TP10}_{=0} - 65TP10$$

$$TP10 = - SE/65$$

Fig 13. Illustration of the average reference calculation process, showing how the activity at reference TP10 is recalculated based on the sum of all electrode activities.

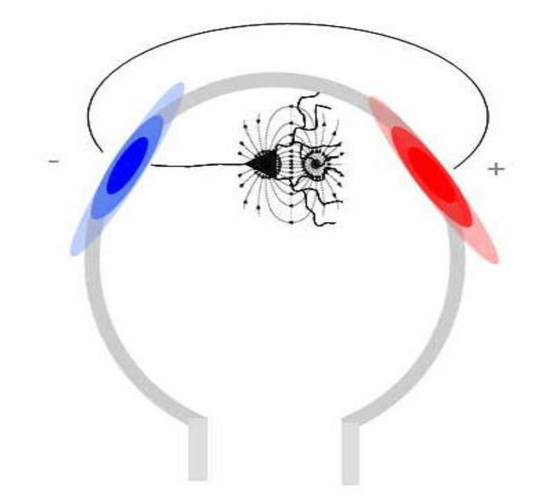


Fig 14. Illustration of average reference method in EEG. A tangential electrical source generates opposing currents: positive inward (left, blue) and negative outward (right, red). Summing currents across the scalp electrodes results in zero, supporting the average reference assumption.

The problem with this assumption is that true average reference data would require the distribution of electrodes to be even over the head. This is not usually the case, as researchers typically place more electrodes over certain scalp areas and fewer (if any) on the lower half of the head surface. As a consequence, an average reference result using one montage may not be directly comparable to an average reference result obtained using another montage.

Re-referencing

Re-referencing is the process of changing the reference electrode after the recording is finished. This technique can improve the quality and interpretability of EEG data. Converting data, before analysis, from fixed or common reference (for example, from a common earlobe or other channel reference) to 'average reference' is advocated by some researchers, particularly when the electrode montage covers nearly the whole head (as for some high-density recording systems). Re-referencing is often needed for several reasons:

Noise Reduction: The initial reference may contain noise or artifacts that can contaminate the EEG signal. For example, if the reference electrode is placed near a muscle, it might pick up muscle activity, which can interfere with the brain signals. Re-

referencing to a different electrode or to the average of all electrodes can help reduce this noise, leading to cleaner data.

Better Signal Representation: Some references might not be ideal for all types of analysis. For instance, using a single electrode as a reference might not provide the best representation of the EEG signal across the scalp. This can lead to an unbalanced view of brain activity, where some regions appear more active than they are. Re-referencing to the average of all electrodes can provide a more balanced and accurate representation of the brain's electrical activity.

Comparison Across Studies: Different studies might use different reference points, which can make it challenging to compare results directly. Re-referencing allows researchers to standardize their data to a common reference, facilitating comparison across different studies. This standardization is crucial for meta-analyses and for drawing broader conclusions from multiple studies.

Artifact Removal: Some references might pick up specific types of artifacts (e.g., muscle activity near the reference electrode). For example, an electrode placed near the forehead might pick up signals from eye movements or facial muscles. Re-referencing can help to minimize these artifacts by choosing a reference that is less affected by such activity, improving the overall quality of the EEG data.

Re-referencing in EEGLAB

To re-reference your data you should take these steps:

Open the Tools Menu

- From the main EEGLAB window, go to the ***Tools*** menu and select ***Re-reference the data***. This option allows you to change the reference electrode used in your EEG recordings.

Re-referencing Options

- When the re-referencing window pops up, you have two main options for re-referencing (see Fig. 15):
    1. **Re-reference the data by selecting specific channels (select <u>re-reference data to channel(s)</u> option)**
    2. **Compute average reference (select <u>compute average re-reference</u> option)**

Re-reference to Specific Channels

- **Select Re-reference Data to Channel(s):** If you select the *<u>re-reference data to channel(s)</u>* option, you can enter the specific channel or channels names you want to use as the new reference in the box in front of it.

  If you click on the three-dot box in front of this option, a secondary window will appear (the right window in the Fig. 15) where you can select the desired channels. Use the shift key to select multiple channels if needed.

Compute Average Reference

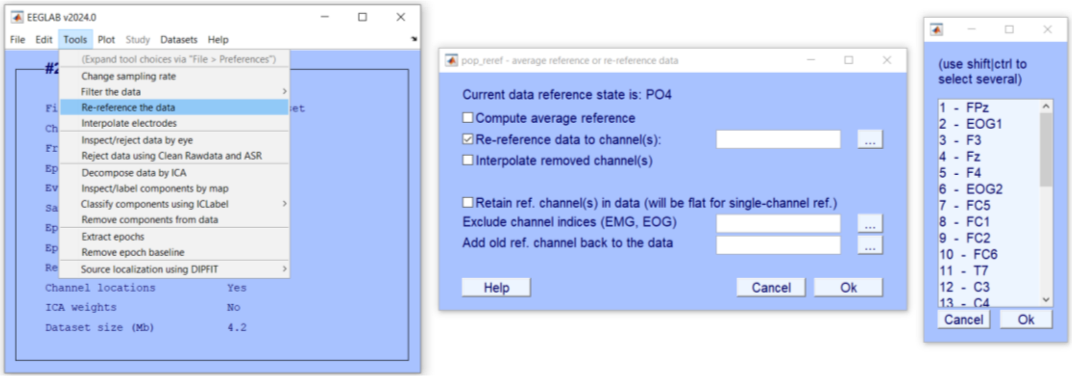
- **Select Compute Average Reference:** This option uses the average of all electrodes as the new reference. This method is highly beneficial as it balances the EEG signal across the scalp, providing a more global and unbiased view of the brain's electrical activity. By selecting the "Compute Average Re-reference" option, you can calculate the average reference for your data. After enabling this option, ensure you achieve the best possible reference by carefully considering the additional settings in the re-referencing window, which are explained below.

- **Additional Options:**
    - **Interpolate Removed Channels:** If any channels have been removed or contain bad data, this option allows for interpolation to include them in the

new reference calculation. This is particularly important when computing the average reference, as symmetric electrode positions on the scalp are necessary for accurate results. Interpolating missing or faulty channels before calculating the average reference ensures that the reference is more precise and reliable.

- **Retain Old Reference Channel in Data:** Keep the original reference channel in the dataset if needed. If the original reference channel is one of the main EEG channels, such as Cz, you should consider retaining it after re-referencing the data. On the other hand, if the original reference is not a main EEG channel, like a mastoid reference, you do not need to retain it and can remove it.
- **Exclude Channel Indices (EMG, EOG):** Exclude channels used for recording eye movements (EOG), muscle activity (EMG), or heart activity (ECG) from the re-referencing process. When calculating the average reference, it is beneficial to base it only on the electrodes placed on the scalp. Therefore, you must exclude electrodes used for recording eye movements, muscle activity, and heart activity.
- **Add Old Reference Channel Back to the Data:** Add the original reference channel back into the dataset after re-referencing.

After selecting the desired options, click **<u>OK</u>** to apply the new reference settings. EEGLAB will adjust the data accordingly, subtracting the new reference signal from each channel.

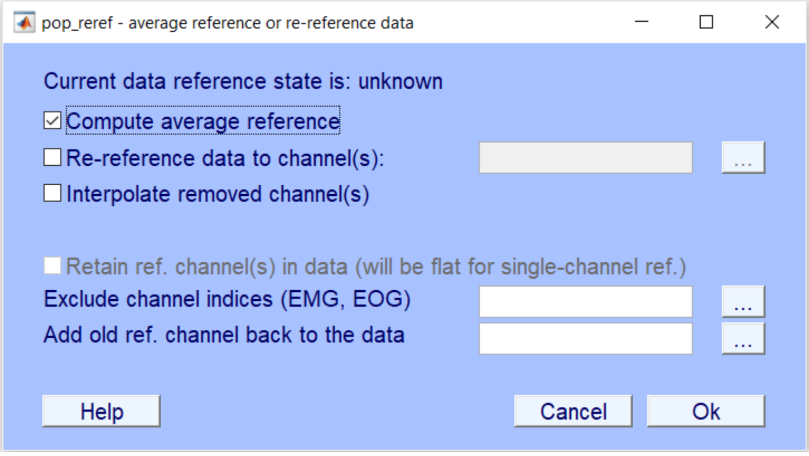

EEGLAB v2024.0
File
Edit
Tools
Plot
Study
Datasets
Help
(Expand tool choices via "File > Preferences")
Change sampling rate
Filter the data
Re-reference the data
Interpolate electrodes
Inspect/reject data by eye
Reject data using Clean Rawdata and ASR
Decompose data by ICA
Inspect/label components by map
Classify components using ICLabel
Remove components from data
Extract epochs
Remove epoch baseline
Source localization using DIPFIT
pop_reref - average reference or re-reference data
Help
Cancel
Ok
Cancel
Ok

Channel locations Yes
ICA weights No
Dataset size (Mb) 4.2

Current data reference state is: PO4
☐ 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

(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
11 - T7
12 - C3
13 - C4

Fig 15. Re-referencing window in EEGLAB, showing options for re-referencing the data by selecting specific channels or by computing an average reference. Here we selected re-referencing our data with specific channel(s) instead of computing the average reference.


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

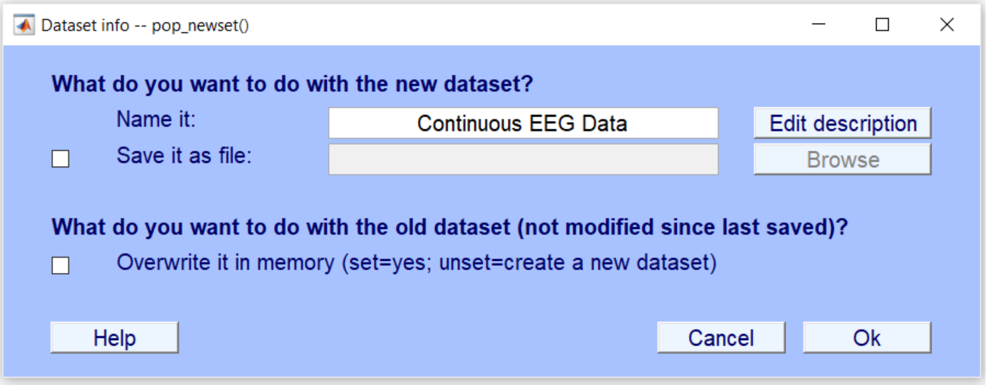
Fig 16. Re-referencing window in EEGLAB showing the option to compute an average reference, which balances the EEG signal across the scalp.

After re-referencing your EEG data, the next step is to save the new dataset. When you click OK to apply the re-referencing changes, a saving window pops up. This window which is shown in Fig. 16 allows you to decide what to do with the new re-referenced dataset and the original dataset.

Options for Saving the Data

This saving window provides two main decisions:

1. **Saving the New Dataset**
   - In the upper section of the window, you can decide what to do with the new re-referenced dataset.
   - **Name it:** You can provide a name for the new re-referenced dataset in the Name it: field. This helps in identifying the dataset easily later.
   - **Save it as file:** If you want to save the new dataset, you can do so by clicking the Save it as file checkbox and specifying the file path. Click Browse to select the desired location on your computer.
2. **Handling the Old Dataset**
   - The lower section of the window allows you to decide whether to overwrite the old dataset with the previous referencing or keep both datasets.
   - **Overwrite in Memory:** If you choose to overwrite the dataset, select Overwrite it in memory. This means the new re-referenced dataset will replace the old one in EEGLAB's memory, effectively deleting the original dataset.
   - **Keep Both Datasets:** If you do not select the overwrite option, a new dataset will be created alongside the previous one. This option allows you to retain both versions of your data for comparison or backup purposes.


Dataset info -- pop_newset()

What do you want to do with the new dataset?

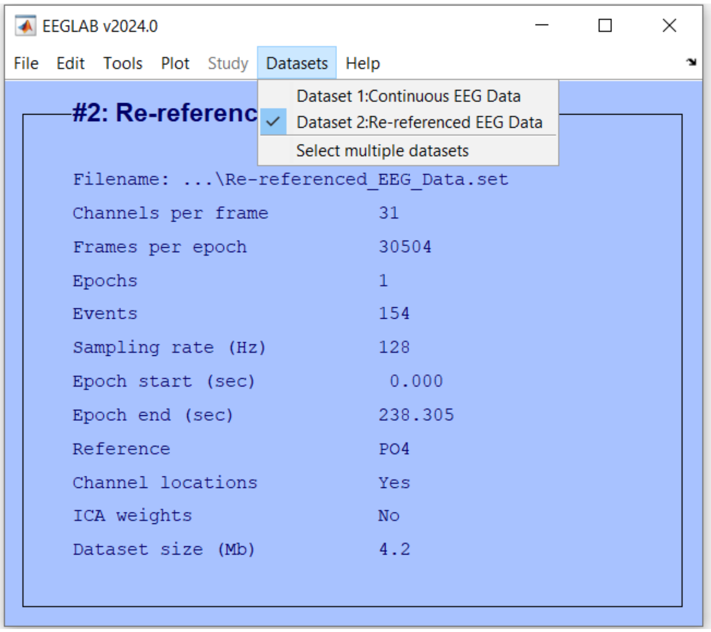
Name it: Continuous EEG Data

□ Save it as file:

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)

Fig 17. Saving window in EEGLAB that appears after re-referencing, allowing you to name and save the new dataset while choosing how to handle the original dataset.



#2: Re-referenc

```
Filename: ...\Re-referenced_EEG_Data.set
Channels per frame          31
Frames per epoch            30504
Epochs                      1
Events                      154
Sampling rate (Hz)          128
Epoch start (sec)            0.000
Epoch end (sec)             238.305
Reference                   PO4
Channel locations           Yes
ICA weights                 No
Dataset size (Mb)           4.2
```

Fig 18. Datasets menu in EEGLAB showing all current datasets in memory. If you choose not to overwrite the old dataset, a new dataset will appear here, preserving both versions in EEGLAB memory. In this example, the first dataset is the original with previous references, and the second dataset is the re-referenced dataset with the new reference.

Apart from re-referencing, whenever any changes are made to the dataset in EEGLAB, such as filtering, removing bad channels, or excluding data segments, you will encounter the window shown in Figure 16. This window allows you to handle dataset saving options, deciding how to save the new dataset and whether to overwrite the old dataset or keep both versions.

Resampling

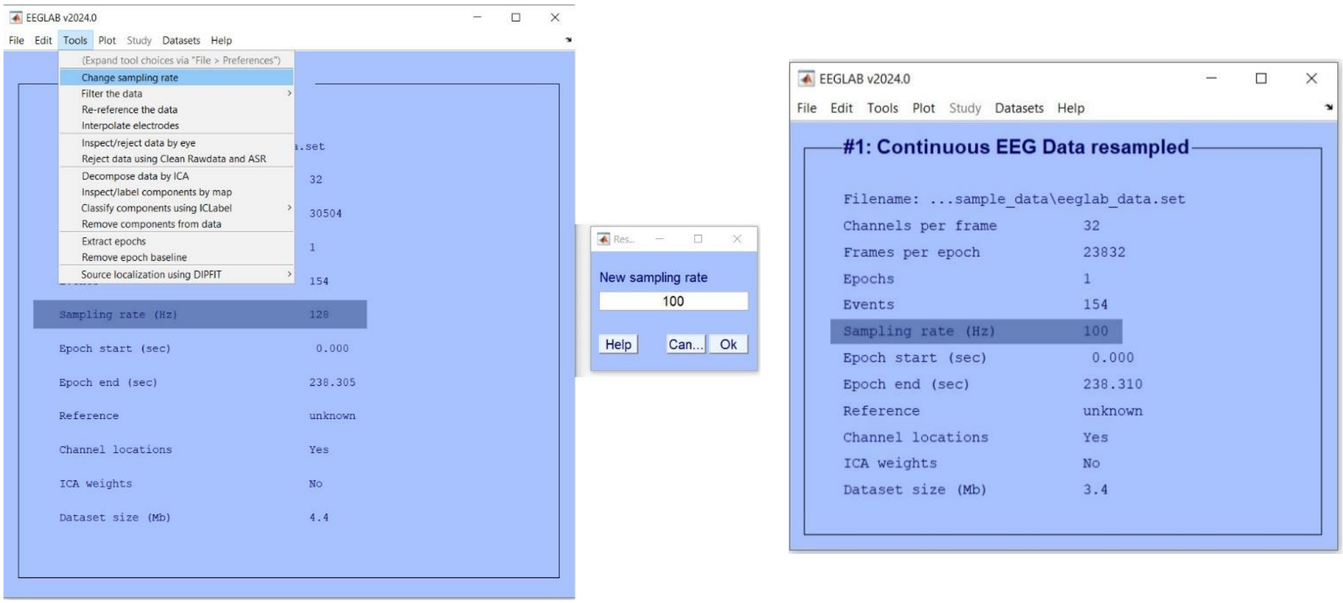
Resampling is an important step in EEG data processing that involves changing the sampling rate of your data. Resampling can be crucial for various reasons:

- **Data Reduction:** Lowering the sampling rate reduces the number of data points, which can significantly decrease the file size and make data handling more efficient.
- **Processing Speed:** With fewer data points, computational processes run faster, making data analysis more efficient.
- **Compatibility:** Some analysis methods and tools require data at specific sampling rates.

Steps to Resample EEG Data

1. **Open the Tools Menu**
    - From the main EEGLAB window, go to the **Tools** menu and select **Change sampling rate** (see Fig. 20). This option allows you to specify a new sampling rate for your EEG data.
2. **Enter New Sampling Rate**
    - A new window will pop up where you can enter the desired sampling rate. For example, if you want to change the sampling rate to 100 Hz, enter 100 in the field and click **OK**.
3. **View Resampled Data**
    - After resampling, the dataset information will update to reflect the new sampling rate. You can see the updated sampling rate and other details in the dataset information window (see Figure 20).

By clicking OK, a new save window will appear, similar to the one used for re-referencing data. This window will prompt you to choose what you want to do with the new and old datasets. Select your preferred options and proceed.


EEGLAB v2024.0
File Edit Tools Plot Study Datasets Help
(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
Source localization using DIPFIT

```
                                        ı.set
                                        32
                                        30504
                                        1
                                        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
```

New sampling rate: 100

EEGLAB v2024.0
File Edit Tools Plot Study Datasets Help

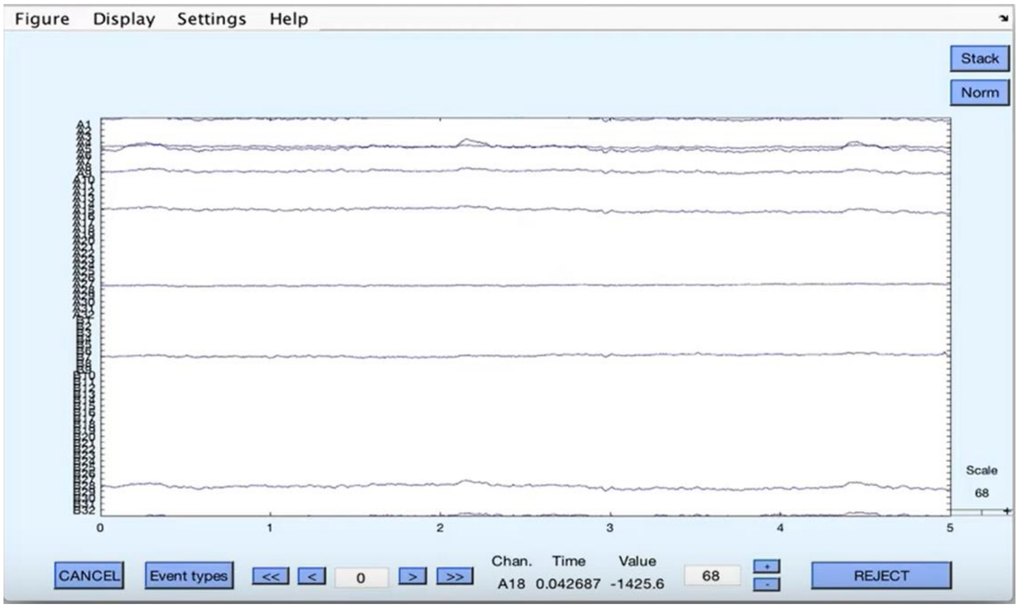
#1: Continuous EEG Data resampled

```
Filename: ...sample_data\eeglab_data.set
Channels per frame            32
Frames per epoch              23832
Epochs                        1
Events                        154
Sampling rate (Hz)            100
Epoch start (sec)              0.000
Epoch end (sec)               238.310
Reference                     unknown
Channel locations             Yes
ICA weights                   No
Dataset size (Mb)             3.4
```

Fig 19. The resampling process in EEGLAB, showing how to change the sampling rate from the Tools menu and the updated dataset information after resampling.

Filtering

Filtering is a crucial step in EEG data processing that enhances the quality of the recorded signals by removing unwanted noise and artifacts. One common issue in EEG recordings is DC shifts, which are slow drifts in the baseline voltage of the EEG signal. These shifts can be caused by various factors, such as changes in electrode impedance or slow physiological processes. DC shifts are particularly problematic because they make data hard to visualize and understand (see Fig. 21). Additionally, DC shifts introduce significant filter artifacts at the beginning and end of the signal. Therefore, it is best to remove them prior to filtering. These low frequency shifts can also affect the results of Independent Component Analysis (ICA), making it difficult to accurately decompose the EEG signals into their underlying components.



Chan. Time Value
A18 0.042687 -1425.6

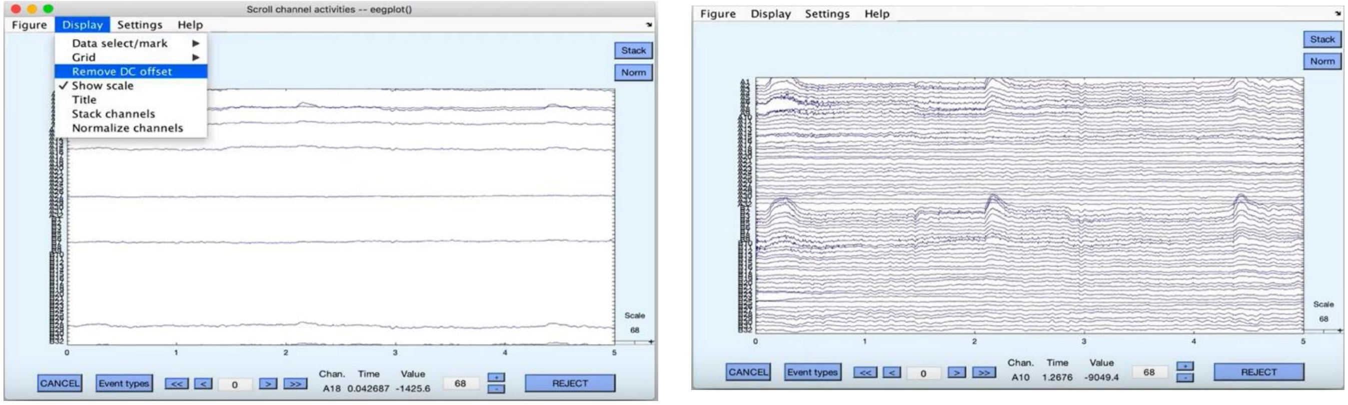
Fig 20. Example of EEG data with noticeable DC shifts, which appear as slow drifts in the baseline voltage. These shifts can complicate data visualization and interpretation, highlighting the need for effective filtering.

Besides DC shifts, EEG data can also be contaminated by various types of noise, such as:

- **Power Line Noise:** Typically at 50 or 60 Hz, depending on the local electrical system, this noise can introduce significant artifacts into the EEG data.
- **Muscle Activity (EMG):** High-frequency noise from muscle movements, especially from facial muscles, can obscure the brain signals.
- **Electrode Noise:** Poor electrode contact or movement can introduce spurious signals into the EEG data.
- **Environmental Noise:** External electrical devices and ambient electromagnetic fields can also introduce noise into the recordings.

Filtering helps to remove these unwanted signals, thereby improving the signal-to-noise ratio. For example, a high-pass filter can be used to eliminate DC shifts and other low-frequency noise, while a notch filter can effectively remove power line noise. By applying appropriate filters, researchers can ensure that the remaining EEG data is cleaner and more representative of the actual brain activity, leading to more accurate and reliable analysis results.

In Figure 22, it is demonstrated how you can remove DC shifts in the display options by choosing **Remove DC offset** under the **Display** menu. This option temporarily removes the DC shift for better visualization. However, to permanently remove this offset from the data, you must either apply a high-pass filter or remove the mean from each data channel.



Chan. Time Value
A18 0.042687 -1425.6

Chan. Time Value
A10 1.2676 -9049.4

Fig 21. Temporarily removing DC shifts by selecting "Remove DC offset" under the Display menu for better visualization.

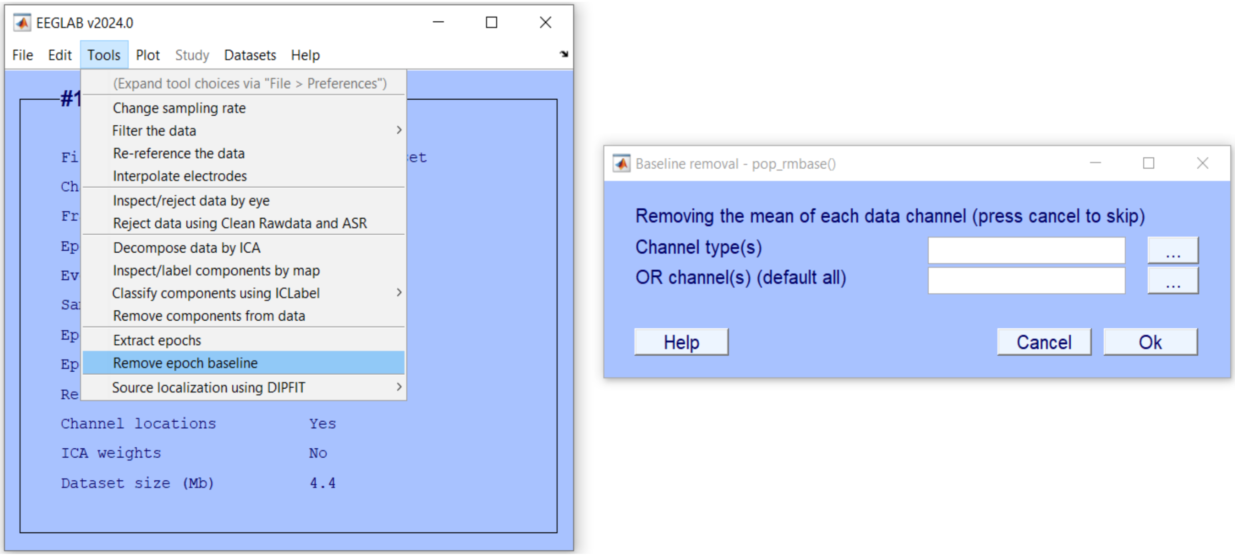
Removing DC Shift

To remove the DC shift in EEG channels, you can use the **Remove epoch baseline** function found under the **Tools** menu in EEGLAB. This selection will be highlighted, as depicted in the left image of Fig. 23.

Upon selecting Remove epoch baseline, a new window titled "Baseline removal - pop_rmbase()" will appear, as shown in the right image of Fig. 23.

Setting Parameters:

- **Channel Type(s):** In the first input field, you can specify the types of channels you wish to apply the filter to. Clicking the **...** button next to this field allows you to select the channel types from a list.
- **OR Channel(s) (default all):** Alternatively, you can specify individual channels by entering their names in this field. If this field is left blank, the filter will be applied to all channels by default.


EEGLAB v2024.0
File
Edit
Tools
Plot
Study
Datasets
Help
(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
Source localization using DIPFIT
#1
Baseline removal - pop_rmbase()
...
...
Help
Cancel
Ok

```
Channel locations          Yes
ICA weights                No
Dataset size (Mb)          4.4
```

Removing the mean of each data channel (press cancel to skip)

Channel type(s)

OR channel(s) (default all)

Fig 22. Removing DC Shift in EEG Channels Using EEGLAB

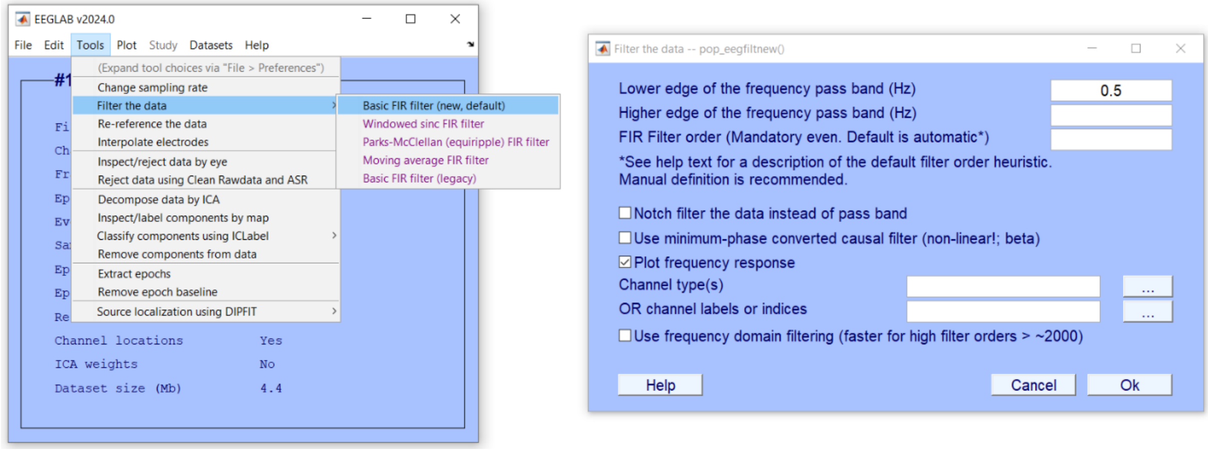
Filtering Data

To filter data in EEG channels, you can use the **Filter the data** function found under the **Tools** menu in EEGLAB. This selection will be highlighted, as depicted in the left image of Fig. 24.

Upon selecting **Filter the data** and choosing **Basic FIR filter (new, default)**, a new window titled "Filter the data - pop_eegfiltnew()" will appear, as shown in the right image of Fig. 24.

Setting Parameters:

- **Lower Edge of the Frequency Pass Band (Hz):** Enter the desired frequency that you desire all the frequencies below it want to be removed. In this example, the value is set to 0.5 Hz, which means frequencies below 0.5 Hz will be filtered out. This field must be specified if you are performing a high-pass or band-pass filter.
- **Higher Edge of the Frequency Pass Band (Hz):** Enter the desired frequency that you desire all the frequencies above it want to be removed. Leave this field blank if you are only performing a high-pass filter. In case of performing low pass or bandpass filter this field has to be specified.
- **FIR Filter Order:** The default setting is automatic, but you can manually define this if needed. It is recommended to refer to the help section for guidance on setting this parameter.
- **Channel Type(s):** In the first input field, you can specify the types of channels you wish to apply the filter to. Clicking the ... button next to this field allows you to select the channel types from a list.
- **OR Channel Labels or Indices:** Alternatively, specify individual channels by entering their labels or indices. If this field is left blank, the filter will be applied to all channels by default.


EEGLAB v2024.0
File Edit Tools Plot Study Datasets Help
#1
Fi
Ch
Fr
Ep
Ev
Sa
Ep
Ep
Re
(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
Source localization using DIPFIT
Basic FIR filter (new, default)
Windowed sinc FIR filter
Parks-McClellan (equiripple) FIR filter
Moving average FIR filter
Basic FIR filter (legacy)
Filter the data -- pop_eegfiltnew()
Help
Cancel
Ok

```
Channel locations        Yes
ICA weights              No
Dataset size (Mb)        4.4
```

Lower edge of the frequency pass band (Hz) 0.5

Higher edge of the frequency pass band (Hz)

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

☐ Use frequency domain filtering (faster for high filter orders > ~2000)

Fig 23. The figure illustrates the process of high-pass filtering EEG data in EEGLAB. The left panel shows the selection of "Filter the data" from the "Tools" menu, followed by "Basic FIR filter (new, default)." The right panel displays the filter configuration window, where the lower edge of the frequency pass band is set to 0.5 Hz to filter out frequencies below this threshold.

After configuring and applying the filter in EEGLAB, the filter response will be plotted. Fig. 25 displays two sample filter responses: the left panel shows the magnitude and phase response of a high-pass filter set at 0.5 Hz, where frequencies below 0.5 Hz are attenuated; the right panel shows the magnitude and phase response of a low-pass filter set at 50 Hz, where frequencies above 50 Hz are attenuated. These plots help visualize the effectiveness of the applied filters in attenuating unwanted frequencies.

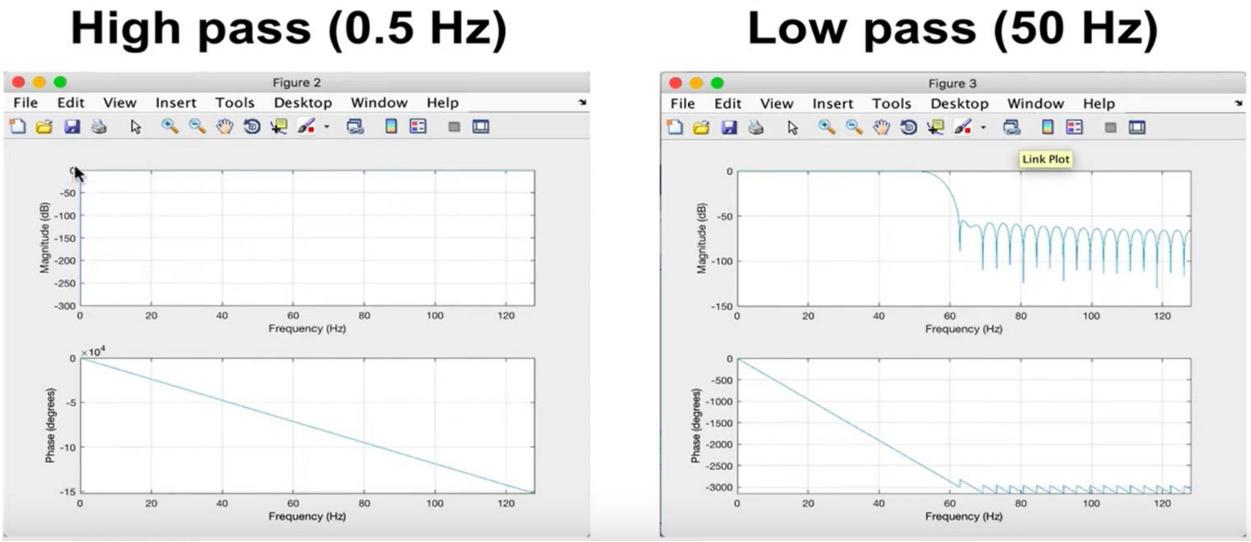


Fig 24. Filter Response Plots for High-Pass and Low-Pass Filters

he amplitude of artifacts (such as eye movements) is often larger than the amplitude of brain data which potentially decrease signal/noise ratio, bias data analysis and potential results

Visualizing Data and Looking for Artifacts

Artifacts in EEG data are non-neural signals that can obscure or distort the true brain activity being recorded. Unlike noise, which is random and lacks a detectable pattern, artifacts are not entirely random and are often caused by specific actions, such as eye blinking, which creates a known pattern on the EEG. These artifacts can originate from various sources, including muscle movements, eye blinks, and environmental interference. Identifying and removing these artifacts, which often have larger amplitudes than the brain signals, is crucial for ensuring the accuracy of EEG analysis.

Figure 26 gives examples of common artifacts in EEG.

1. Transient High-Frequency Event (Muscle)

- **Description**: This artifact is typically caused by muscle activity, such as clenching the jaw or moving the face.
- **Appearance in Signal Plot**: It manifests as brief, high-frequency spikes in the EEG signal. These are indicated in the figure within the black circle, showing sharp and fast oscillations.

2. Low-Frequency Event (Eye Movements)

- **Description**: Eye movements, including saccades and blinks, create low-frequency artifacts in the EEG data.
- **Appearance in Signal Plot**: These are represented by slow, wave-like patterns. The blue circle in the figure highlights this type of artifact, characterized by gradual, sweeping changes in the signal amplitude.

3. Discontinuity

- **Description**: Discontinuities can occur due to abrupt changes in the signal, possibly from electrode disconnections or sudden movements.
- **Appearance in Signal Plot**: They are identified by sudden jumps or breaks in the signal, as shown in the green circle in the figure.

4. High Noise

- **Description**: Environmental or electrical noise can introduce high levels of interference into the EEG data.
- **Appearance in Signal Plot**: This is depicted as erratic and chaotic patterns in the EEG signal, highlighted by the area marked with a green oval in the figure.

5. Linear Trend

- **Description**: A linear trend may appear due to electrode drift or gradual changes in the electrode-skin interface.
- **Appearance in Signal Plot**: It appears as a steady, sloping increase or decrease in the signal, marked by a blue arrow in the figure.

6. Blinks

- **Description**: Blinks are a common source of artifacts, seen when the eyes close and open.
- **Appearance in Signal Plot**: They appear as large, slow waves, often with a distinct pattern. The bottom panel of Figure 26 shows these artifacts with red arrows pointing to the blink events.

Identifying and Removing Artifacts

These artifacts are typically detected manually after EEG recording. Researchers visually inspect the data to identify and mark these non-neural signals. Automated algorithms can assist but often require manual confirmation to ensure accuracy. Once identified, these artifacts can be removed or corrected using various preprocessing techniques, such as filtering, independent component analysis (ICA), or regression methods.

Note that noise and artifacts are handled and removed differently. Noise is commonly removed using filters, while artifacts are usually tackled manually or with automatic tools like ICA (Independent Component Analysis). Therefore, do not confuse these two distinct concepts affecting EEG quality and their respective handling methods.

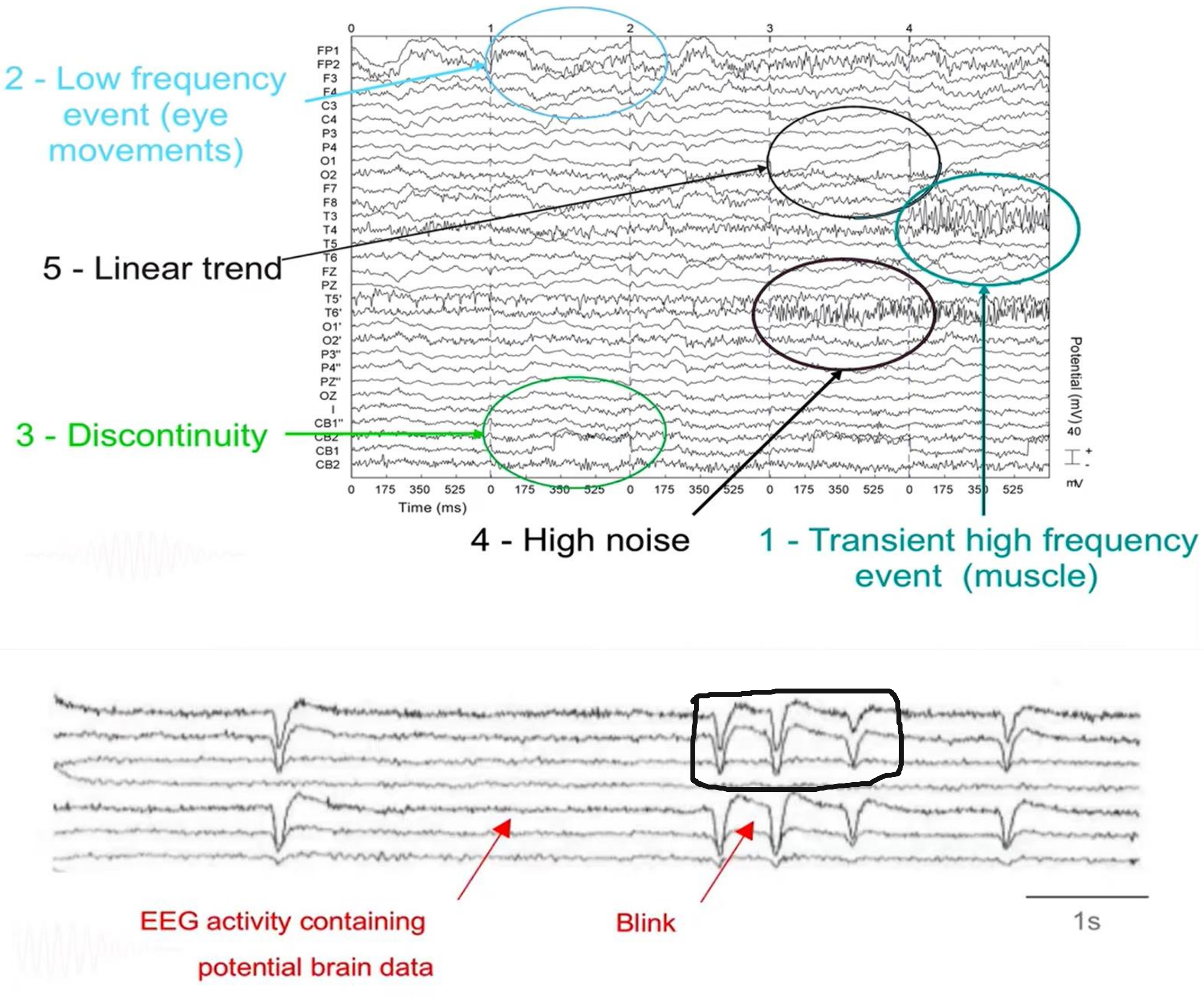


Fig 25. The figure illustrates various types of artifacts found in EEG recordings. These include transient high-frequency events (muscle activity), low-frequency events (eye movements), discontinuities, high noise, and linear trends. The bottom panel specifically highlights blink artifacts, which are common in EEG data.

Removing Bad Data and Channels

In EEG recordings, certain channels can be classified as "bad channels" when they exhibit significant deviations from typical brain activity patterns. These deviations can be due to various factors, including poor electrode contact, excessive noise, or artifacts such as muscle activity, eye movements, or external electrical interference. Bad channels can disrupt the overall quality of the EEG data, leading to inaccuracies in analysis and interpretation.

The identification of bad channels can be performed by visual inspection by an expert, who looks for abnormal amplitude fluctuations, noise, or patterns that do not correlate with other channels. In some cases, automated algorithms may also be used to flag potential bad channels for further review.

In Fig. 27, Channel 16 is highlighted in red, indicating it is contaminated with artifacts that significantly differ from the other channels. This contamination makes Channel 16 a bad channel, and it should be manually detected and removed from the data to maintain the accuracy of the EEG analysis. However, before certain operations, such as re-referencing, it is advisable to interpolate the removed channel to preserve the spatial information and ensure the consistency of the EEG dataset.

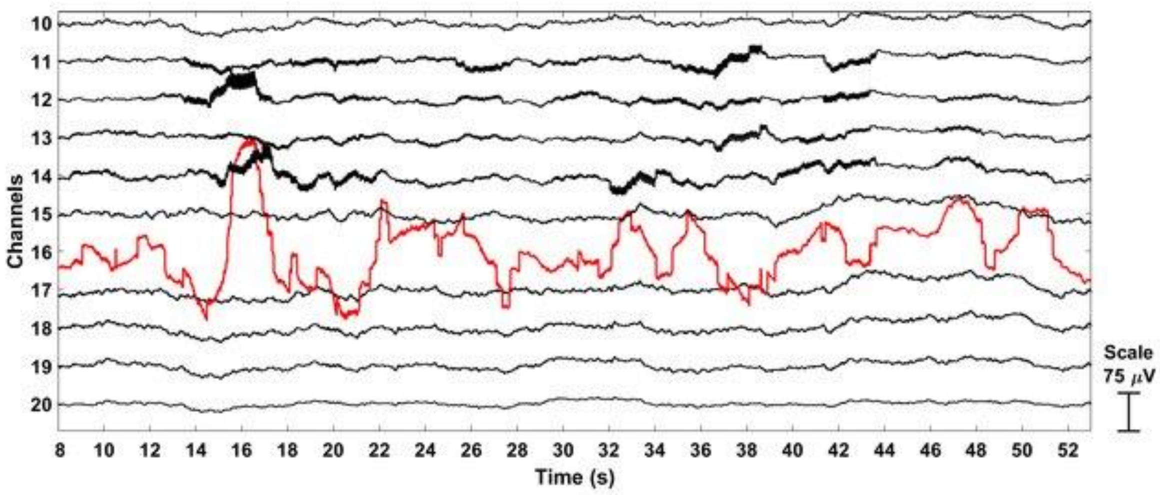


Fig 26. Bad channel (Channel 16) in red is contaminated with artifacts, illustrating the need for manual detection and removal.

After certain channels are identified as "bad channels" they usually should be removed. EEGLAB provides a straightforward tool to remove these channels from your dataset.

Steps to Remove Specific Channels:

1. **Access the Channel Selection Tool:**
   - Navigate to the "**Edit**" menu in the EEGLAB main interface.
   - Select "**Select data**" from the dropdown menu, which opens the data selection window.
2. **Select Channels for Removal:**
   - In the "Select data" window, you can input the desired range for time, points, epochs, and channels that you want to keep in your dataset.
   - To remove specific channels, click on the box next to "Channel range," as indicated by the orange circle in the figure. Checking this box reverses the channel selection process. Normally, selected channels are retained while all others are removed. If you want to specifically choose the channels to remove, check this box.
   - This will open a list of all available channels.

3. **Identify and Remove Bad Channels:**
   - Use the shift or control (ctrl) key to select multiple channels that have been identified as bad.
   - Once the channels are selected, click "Ok" to confirm their removal from the dataset.
4. **Finalize Removal:**
   - After confirming, the selected channels will be removed from your dataset.
   - This step is crucial to ensure the remaining data is clean and more representative of the true neural activity.

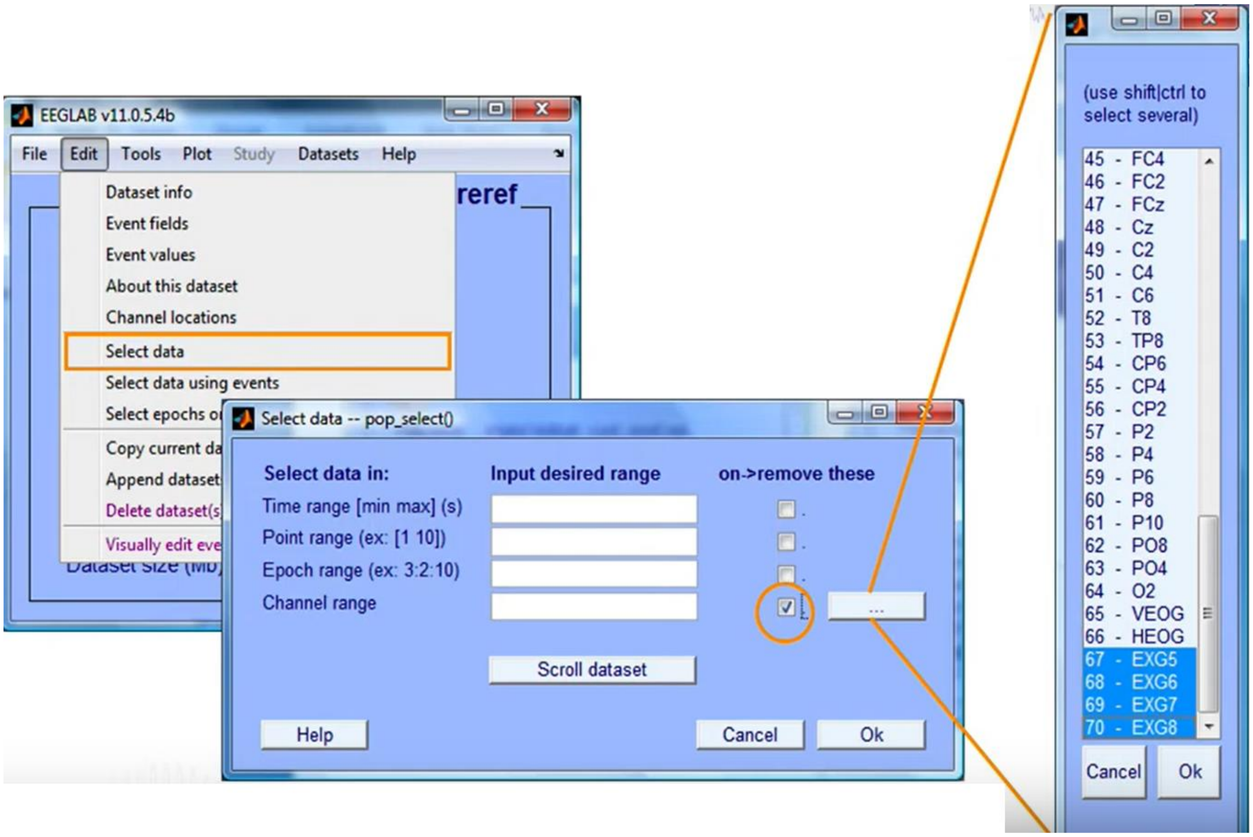


Fig 27. This figure demonstrates the tool within EEGLAB used to remove specific channels identified as bad.

Inspecting Channel Spectral Information in EEGLAB

One important tool in identifying bad channels in EEG data is inspecting their spectral information. This process helps determine whether a channel exhibits abnormal activity that might interfere with the overall data quality. EEGLAB provides tools to analyze the spectral properties of individual channels, allowing users to make informed decisions about whether to retain or remove specific channels.

Figure 29 indicates how to access EELAB tool for analyzing Spectral Information of EEG two EEG channels, and gives examples of two spectral Channel information’s. One of them corresponding to a normal EEG spectral, the other corresponding to an abnormal EEG spectral.

Steps to Inspect Channel Spectral Information:

1. **Access the Channel Properties Tool:**
   - Navigate to the "Edit" menu in the EEGLAB main interface.
   - Select "Channel properties" from the dropdown menu, which opens the channel properties window.
2. **Select Channels to Inspect:**
   - In the "Component properties" window, enter the channel indices you want to plot in the "Channel index(es) to plot" field.
   - Set the spectral options by specifying the frequency range in the "Spectral options" field. For instance, a common setting is to analyze frequencies between 2 and 50 Hz.
   - Click "Ok" to generate the spectral plots for the selected channels.
3. **Analyze Spectral Information:**
   - The resulting plots display several key pieces of information:
     - **Topographical Map**: Shows the location of the selected electrod on the scalp.

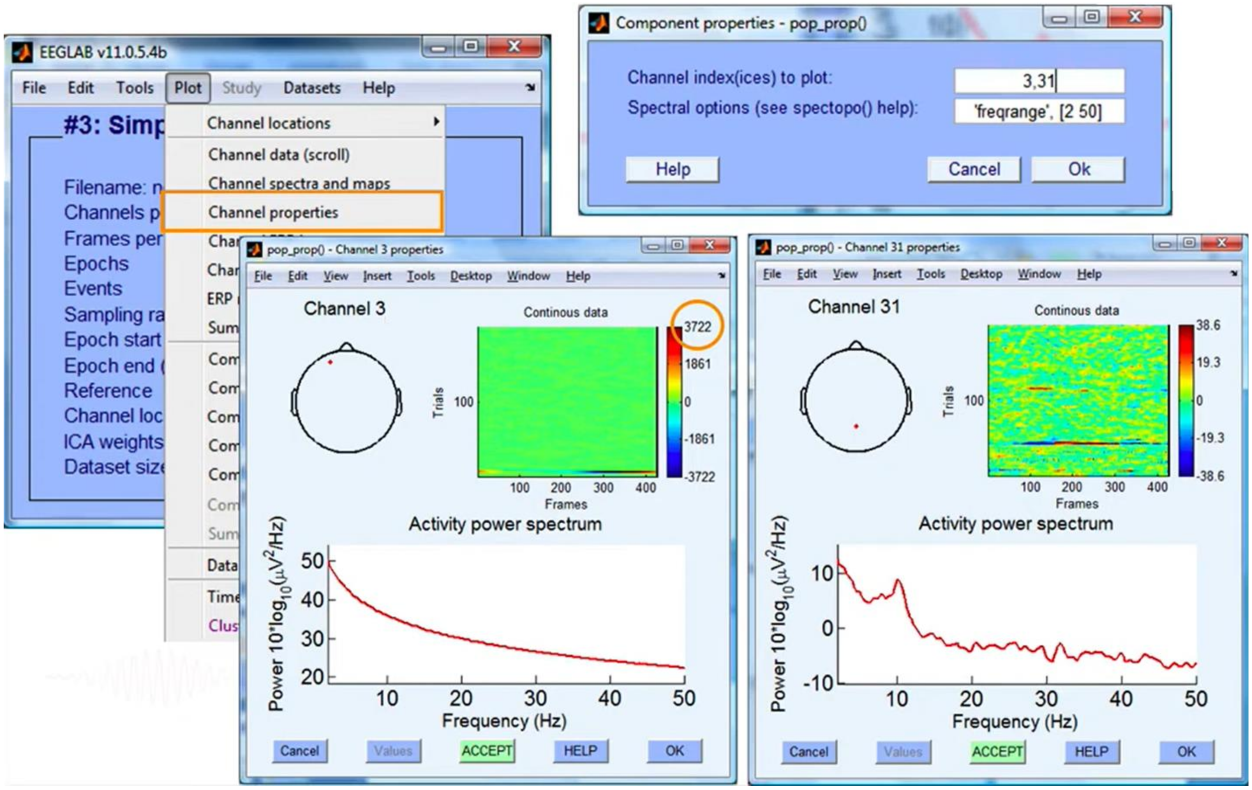
        - **Activity Power Spectrum**: Displays the power of the EEG signal across different frequencies, usually on a logarithmic scale.
        - **Continuous Data Plot**: Provides a time-domain visualization of the signal for continuous inspection.
    - Compare the spectral information of different channels to identify abnormalities.

Example Analysis:

- The left panel in Figure 28 shows the spectral information for Channel 3, which exhibits abnormal activity. The activity power spectrum shows an unusual power distribution, with significant power at higher frequencies.
- The right panel displays Channel 31, which shows a normal power spectrum with a typical distribution of power across the frequency range, indicating it is a good channel.

Importance of Spectral Analysis

Spectral analysis is vital for identifying bad channels because it reveals the frequency characteristics of the EEG signals, which are often more informative than time-domain signals alone. Channels with abnormal spectral properties, such as excessive power in high-frequency bands or unusual patterns, can introduce significant noise and artifacts into the data. Identifying and removing these channels ensures cleaner and more reliable EEG data for subsequent analysis.

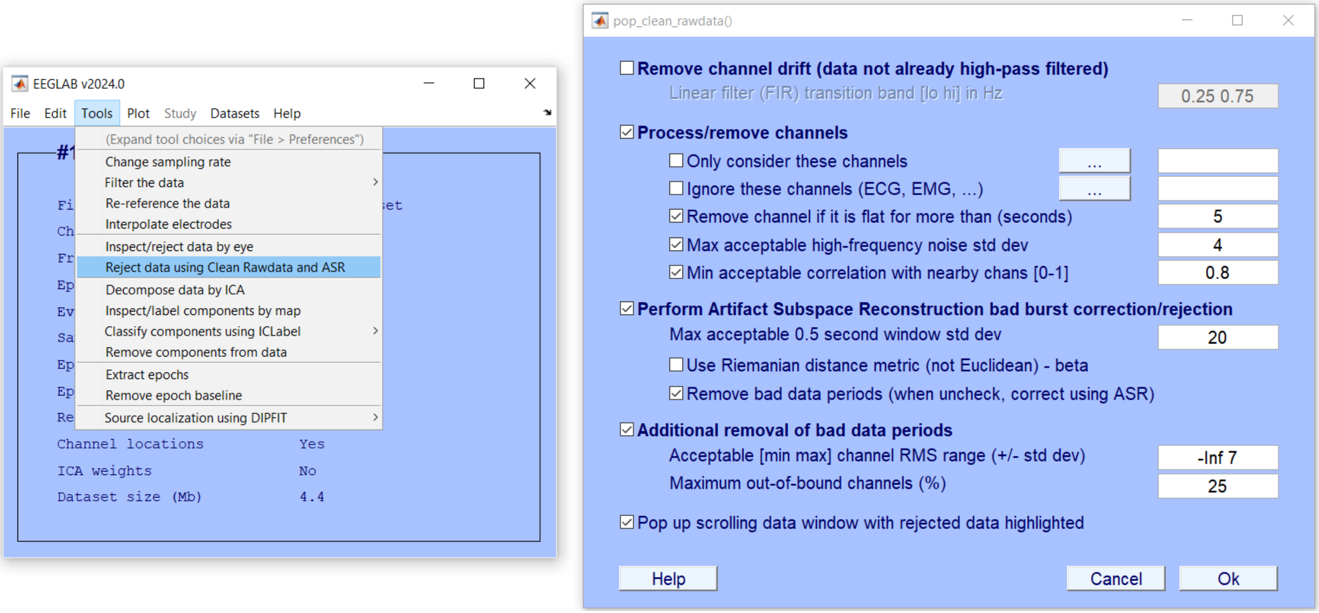


Component properties - pop_prop()

Channel index(ices) to plot: 3,31

Spectral options (see spectopo() help): 'freqrange', [2 50]

Fig 28. Example of spectral analysis in EEGLAB showing an abnormal spectrum for Channel 3 (left) and a normal spectrum for Channel 31 (right).


EEGLAB v2024.0
File Edit Tools Plot Study Datasets Help
(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
Source localization using DIPFIT
#1
Fi
Ch
Fr
Ep
Ev
Sa
Ep
Ep
Re
et

```
Channel locations          Yes
ICA weights                No
Dataset size (Mb)          4.4
```

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

Fig 29.

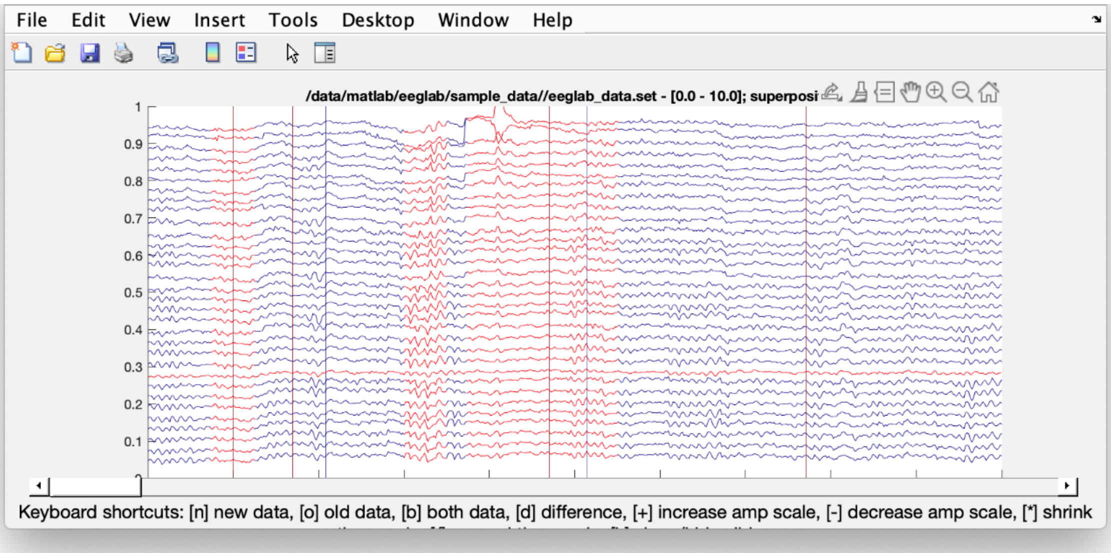
There are several sections to this menu indicating different sequential processes:

- The top section is about high-pass filtering your data. By default, this option is not selected since EEGLAB assumes that you might have already filtered your data. However, if this is not the case, you may select that option. The frequency limits indicate the transition bandwidth for the high pass filter (so 0.25 to 0.75 Hz indicate a high-pass filter at 0.5 Hz).
- The second option deals with removing bad channels. There are three methods to remove bad channels. Flat channels may be removed. Channels with a large amount of noise may be removed based on their standard deviation, and channels, which are poorly correlated with other channels, may be removed. The rejection threshold for channel correlation is set to 0.8. Note that channel rejection based on their correlation is performed differently

if you have imported channel locations (a different heuristic that takes into account channel location is used in case you have them - and we strongly advise importing channel locations before automated artifact rejection).

- The third section deals with rejecting bad portions of data using the Artifact Subspace Reconstruction (ASR) algorithm. The full description of this algorithm is outside the scope of this tutorial. For more information, we refer to this Appendix. ASR may be used to correct bad portions of data or to remove them. For offline EEG processing, we advise to remove them, which corresponds to the default options. First, ASR finds clean portions of data (calibration data) and calculates PCA-extracted components' standard deviation (ignoring physiological EEG alpha and theta waves by filtering them out). It rejects data regions if they exceed 20 times (by default) the standard deviation of the calibration data. The lower this threshold, the more aggressive the rejection is. The Riemannian distance is an experimental metric published in this article that claims superior performance – ASR's author C. Kothe disputes its claims.
- The fourth option deals with the additional rejection of bad portions of data based on a set number of channels passing a standard deviation threshold in a given time window. The time window size can be fine-tuned when running the function from the command line. This allows rejecting bad portions of data that ASR might have missed.
- The last option allows plotting results of the rejection with rejected data highlighted.

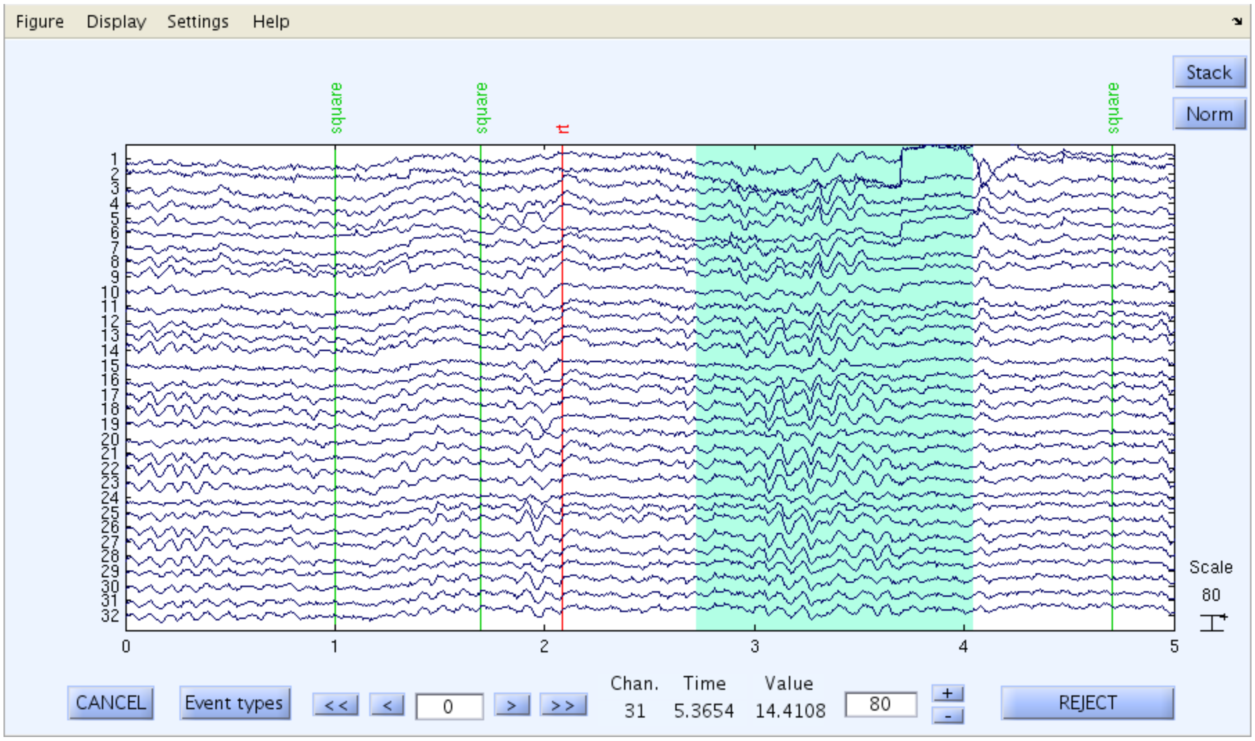
The result of rejecting data on the tutorial dataset are shown below. We can see the rejected channels in red and the rejected data portion when all channels are marked in red in a given time segment. If correcting data using the ASR algorithm (not shown here), the old and new (corrected) data will be overlaid. We do not recommend using ASR correction (as opposed to rejection) on pre-recorded data because this functionality was primarily developed for real-time applications. The consequences of using ASR on EEG post-processing are not clearly understood – although an increasing number of articles are being published on this subject.



/data/matlab/eeglab/sample_data//eeglab_data.set - [0.0 - 10.0]; superposi

Keyboard shortcuts: [n] new data, [o] old data, [b] both data, [d] difference, [+] increase amp scale, [-] decrease amp scale, [*] shrink

Fig 30.



Chan. 31
Time 5.3654
Value 14.4108
80

Fig 31.