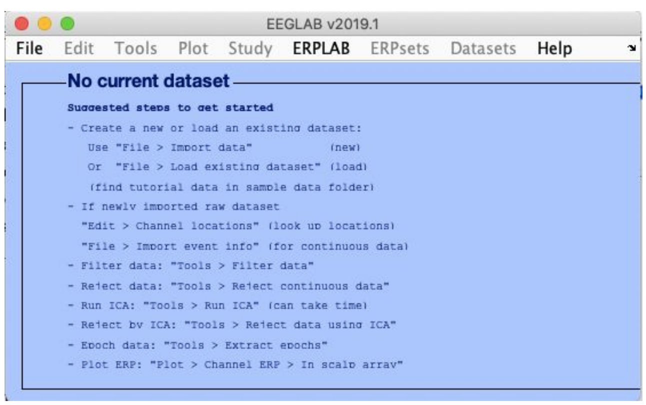
EEG & ERP Workflow

Prerequisites

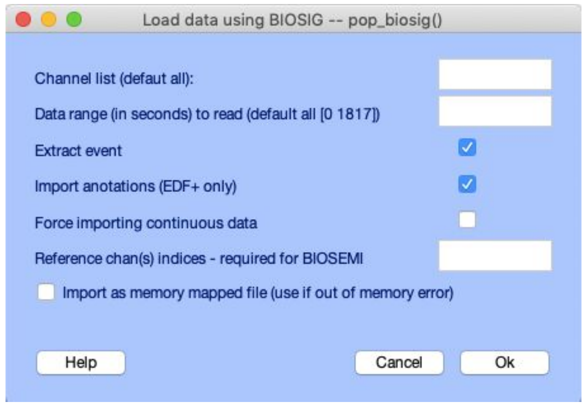
1. Make sure MatLab is installed
2. Download EEGlab: EEGLab is a toolbox that facilitates EEG data analysis within MatLab; more information is provided in this link: https://sccn.ucsd.edu/eeglab/index.php. To get access to the installer, you must request it here: https://sccn.ucsd.edu/eeglab/download.php
3. ERPlab plugin: ERP stands for event related potential, which basically means the squiggly brain waves that are produced in EEG data after an event triggers some response. Information can be found here: https://erpinfo.org/erplab. To download, the previous link will redirect you to the GitHub: https://github.com/lucklab/erplab/releases, where the zip for the installer can be found.
4. Files needed for data analysis include:
    a. A .bdf file that contains the raw EEG data
    b. A .mat file that contains raw data (most importantly a file that contains the events)
    c. It's helpful to have these two files in a folder of its own for sake of organization

Importing data

1. In the MatLab command window run `eeglab` (to distinguish between English and commands, I will be switching fonts). This should create a popup that looks like the window below:



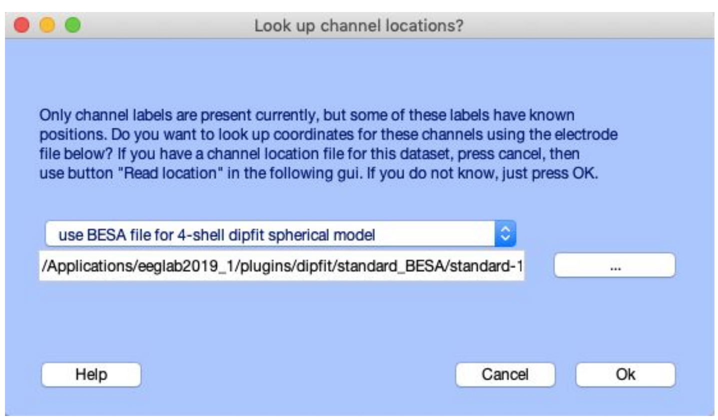
2. On the navigation bar, there are several dropdowns such as File, Edit, Tools, etc. Click > File > Import data > Using EEGLab functions and plugins > From BIOSEMI BDF file (BIOSIG Toolbox). Then, select the bdf file.
3. You will see the popup shown below. Under "Reference chan(s) indices - required for BIOSEMI" input `35 36`. These are the bilateral mastoids.



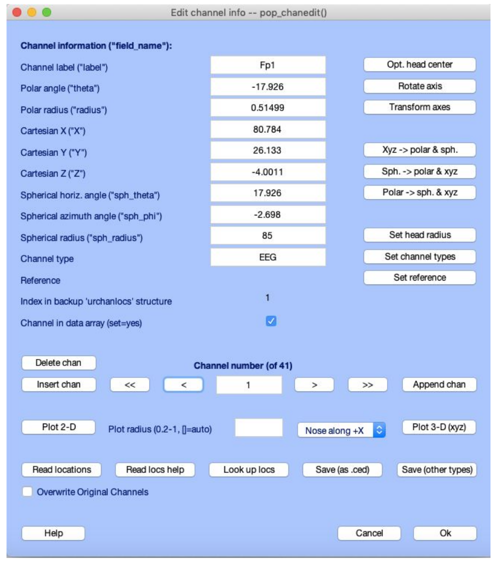
4. The GUI will then ask "What do you want to do with the new dataset." Give it a name, then on the EEGLab GUI click > File > Save current datasets.

Basic edits and previewing the data

1. Setting channels: we need to import the channels into the dataset. Click > Edit > Channel locations. A popup like the one shown below will show up. The default file should work, but if there isn't one, you may have to use the file at http://www.biosemi.com/download/Cap_coords_all.xls. After hitting "ok" another popup with each of the 41 channels will appear (shown below). We will only modify the field Channel type for a few of these channels. For the channel Sync, we set Channel type to `SYNC`. And for channels veog and heog, we set it to `EYE`. Finally hit "ok."

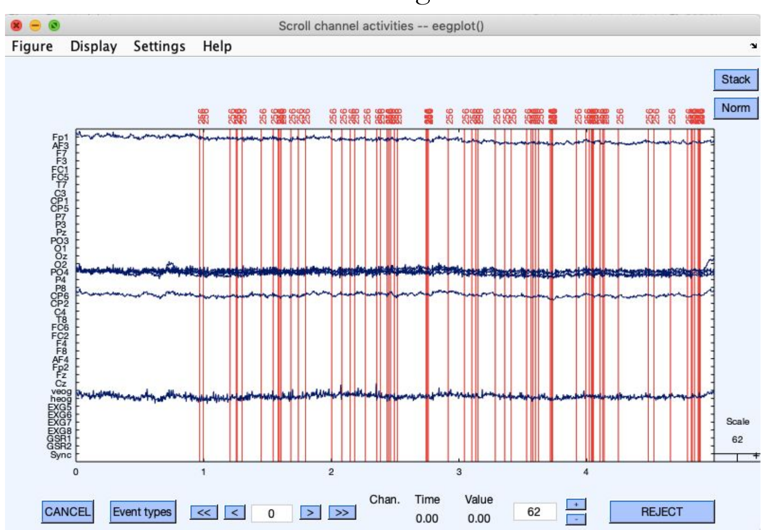


Popup after selecting Channel locations



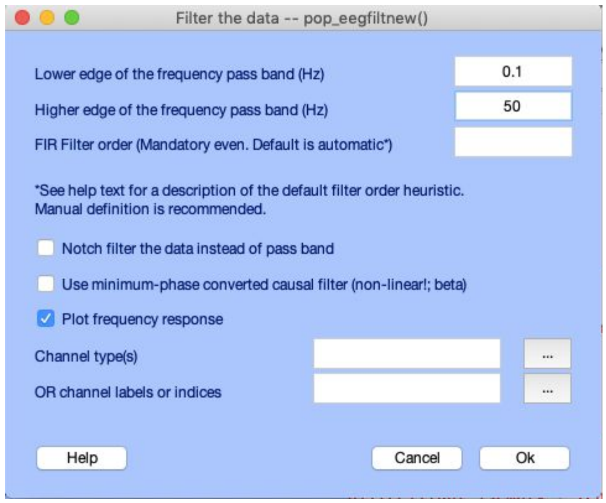
Popup after selecting Ok in the previous popup

2. Preview the data: On the menu bar, click > Plot > Channel data (scroll). This will give some unintelligible window like the one below. On the menu bar of this window, click > Display > Remove DC Offset; this makes the data more readable. To adjust how much data is displayed, use the controls under the > Settings tab.



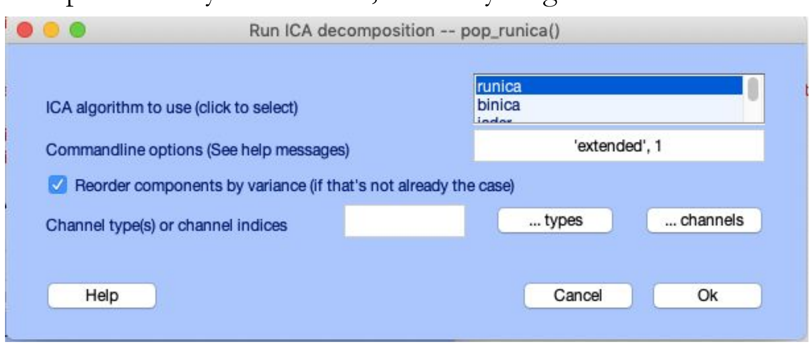
3. Setting events: You may notice colored vertical lines when you display the data. These correspond to events. Check if the events are incorrect. If they make no sense, events must be set manually. In the MatLab command window run `EEG.event` this will give the struct with the array of events. If this array is incorrect, set it using `EEG.event = event_file`, where `event_file` is the name of the file with the array of events (this is usually part of the .mat file which must be opened with `load your_data_file.mat`). When this step is complete, scroll through the data to see if it makes logical sense.
4. Filtering: To apply basic band filters on the EEGLab menu bar click through > Tools > Filter the data > Basic FIR filter (new, default). This gives the popup shown below.

Typically, we apply a lower edge frequency pass of around ~0.1 Hz and a Higher edge frequency pass of ~50 Hz. These numbers depend on what kind of noise you want to eliminate and the context of the experiment. Finally, preview the data and save this as a new dataset.

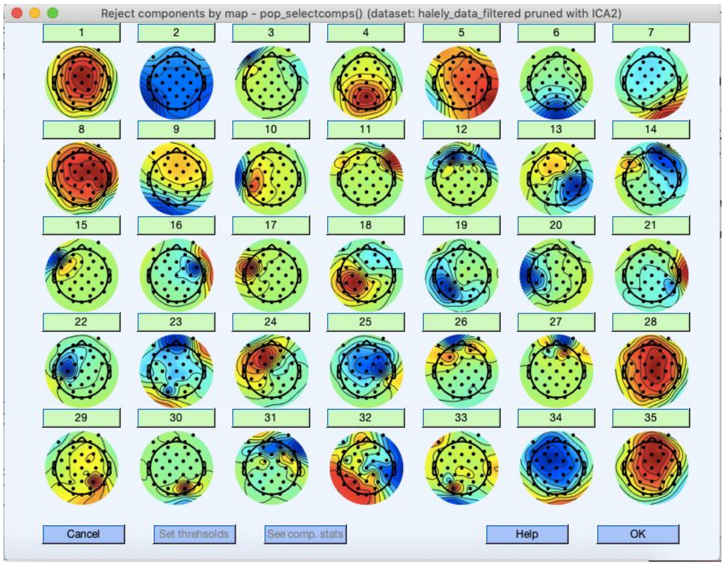


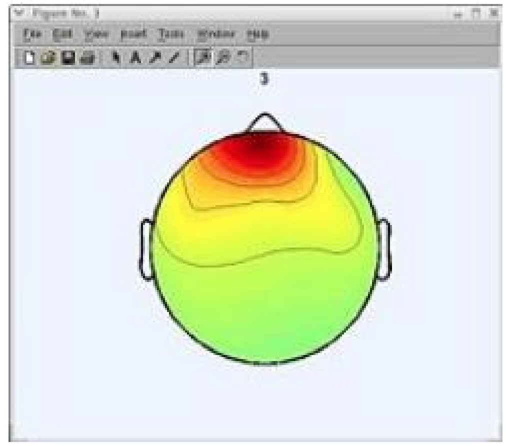
Removal of eye components

1. The purpose of removing eye components is to eliminate noise and red-herrings in the data. Sometimes a reaction may be coupled with a blink, which may be coupled with other channel's behaviors in the EEG. In these cases, it's important to remove the frequencies or components that correspond to eye movements. The basic idea is to run an ICA to find these components and delete those that correlate the most with the eye channels.
2. From the EEGLab menu bar, click through > Tools > Decompose data by ICA. In the popup that follows select all "runica" for ICA algorithm to use and select all channel types. Hit "ok." This process may take a while, so it may be good to let it run overnight.

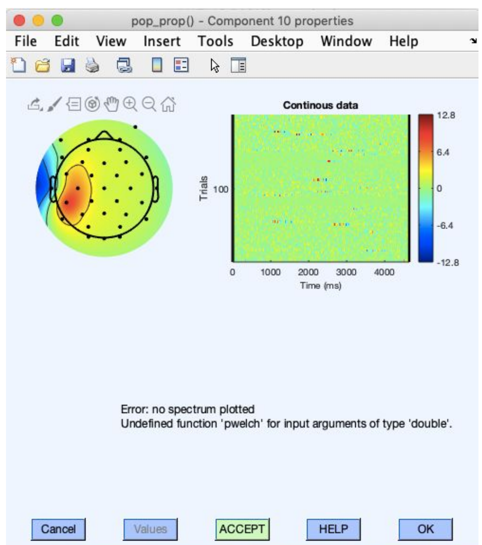


3. From the EEGLab menu bar click > Tools > Inspect/label components by map. This will give you a popup like the one below with a different number of channels. Select the channels that are particularly active in the eye regions. An example of these images is shown below (sccn.ucsd.edu).

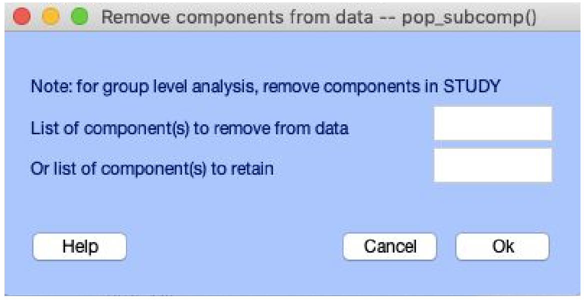




4. Once clicking on the green number corresponding to a component that you may want to reject, change the "accept" button at the bottom to "reject" by clicking on it. Then hit ok. See figure below:



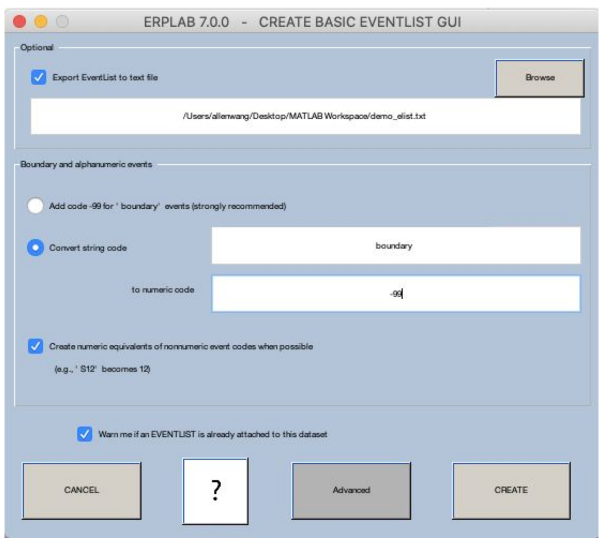
5. Finally on the EEGlab menu bar click > Tools > Remove components from data. This will bring up the list of components that you have selected to reject from the previous step. After hitting ok, the GUI will ask you to preview the data. Pay attention to the heog and veog channels to see if they are more uniform than before. At this step, if the data is not good, it's important to go back and adjust the components that you choose to reject.



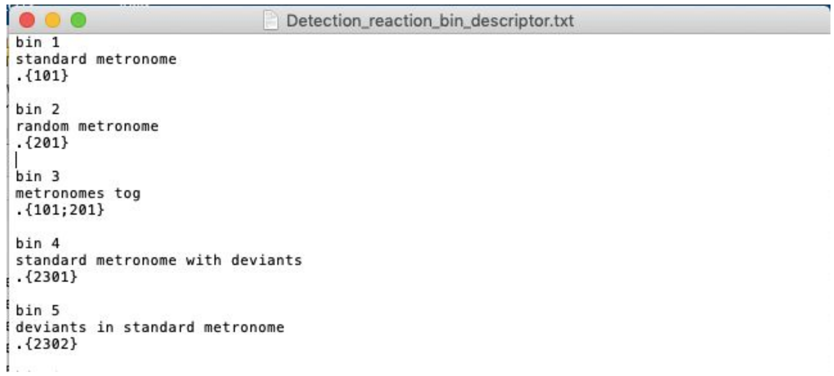
6. When the data is good to go, save this as a new data set!

Elist and binlister

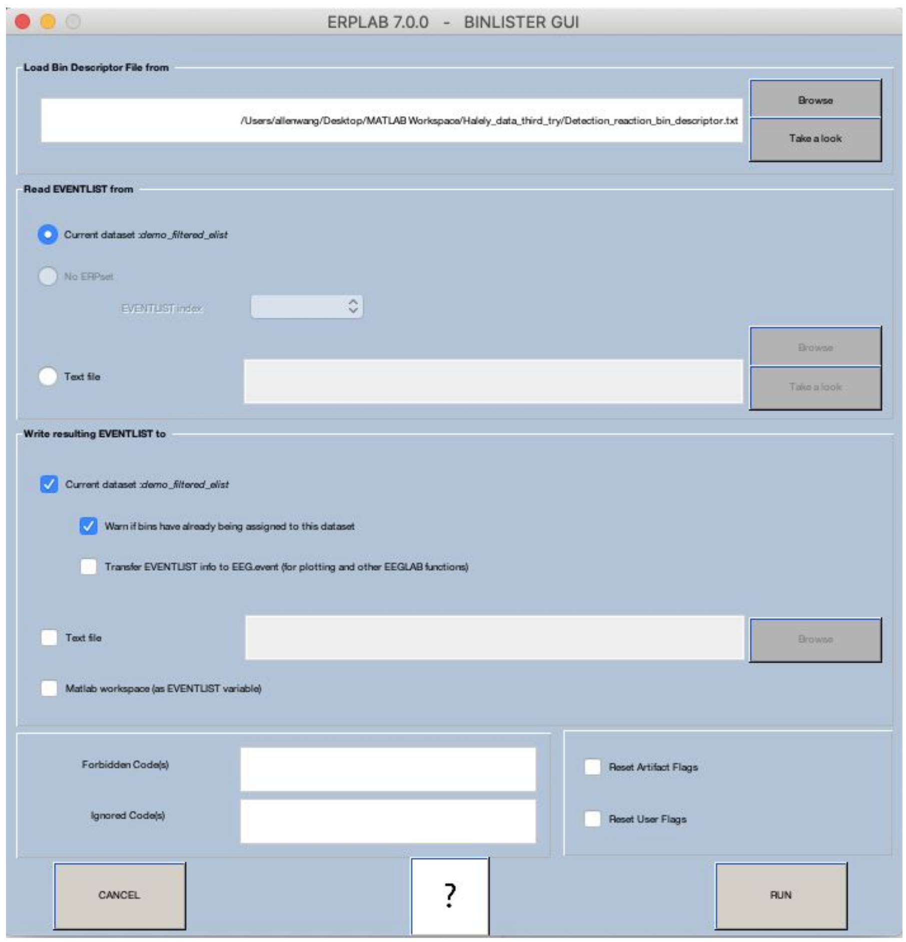
1. Before continuing check `EEG.event` to see if it is the correct set of events.
2. Through the EEGLab menu bar click > ERPLAB > EventList > Create EEG EVENTLIST. This will give you an ugly GUI like the one below. Save it as a separate text file if the event list has not been made yet and accept the default settings. This is an event list that is compatible with binlister that will allow you to group certain events together to be viewed on an ERP plot later.



3. Next you need to make sure you have a bin descriptor .txt file. An example of such a file is shown below. For information on how to specify the bin descriptors see this link: https://github.com/lucklab/erplab/wiki/Assigning-Events-to-Bins-with-BINLISTER.
Every group of four lines is specified as follows:
    a. "bin" followed by an integer
    b. A description that cannot contain the three letter sequence "bin" within any word
    c. A descriptor of all events that will be averaged in this bin
    d. An empty line



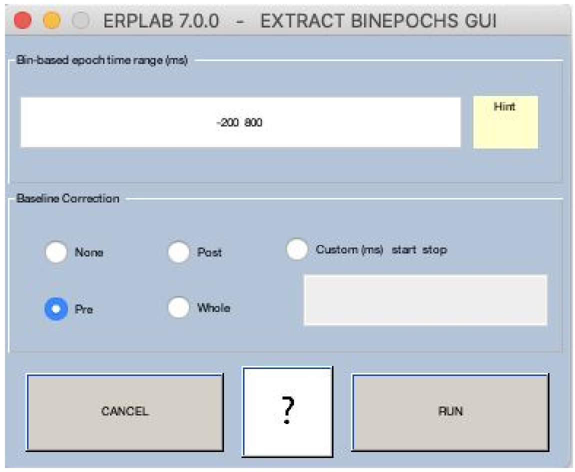
4. After your bin descriptor is made, from the EEGLab menu bar click > ERPLAB > Assign bins (BINLISTER). This gives the GUI below. Typically the default fields are fine; make sure that the elist used is the one made from step 1. Afterwards run and save the dataset.



Extracting epochs

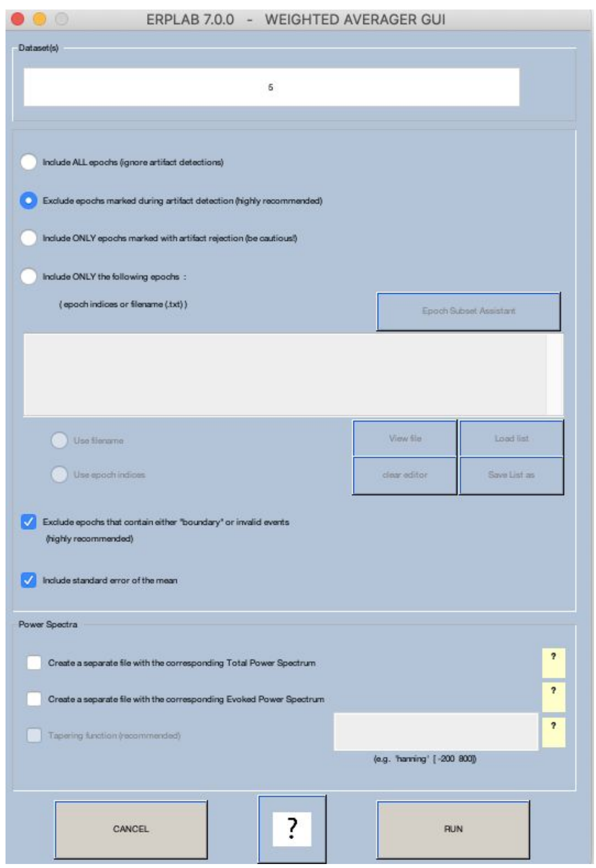
1. Now we have to extract the epochs for the events in each bin. To do this from the EEGLab menu bar click > ERPLAB > Extract bin-based epochs. This gives the following GUI. The

arguments specify how long before the event and after the event you would like to extract. Adjust to fit your needs and hit run. Finally, save the dataset.

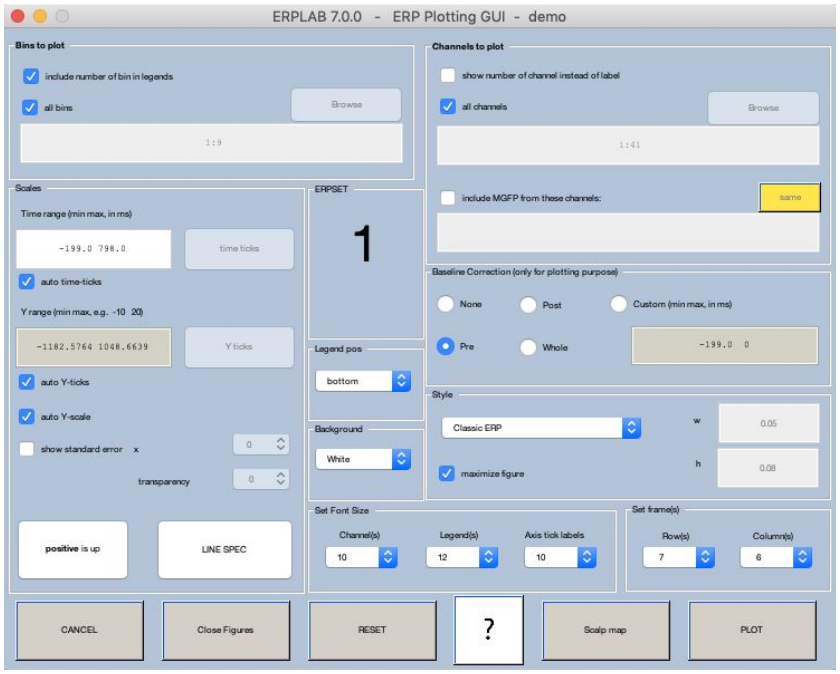


Compute and plot ERP

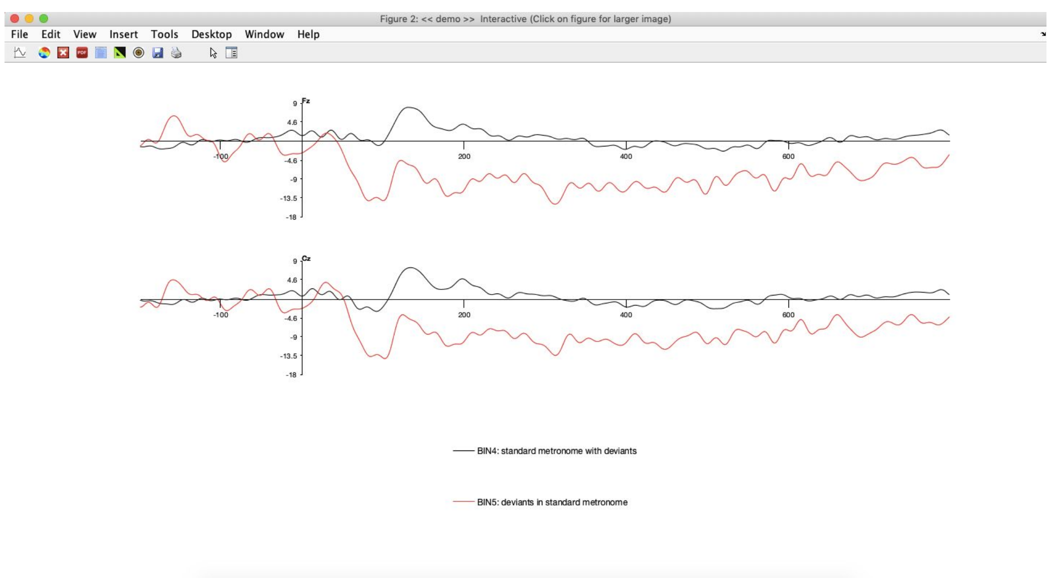
1. To compute the averaged ERPs, from the EEGLab menu bar click > ERPLAB > Compute averaged ERPs. The default fields in the following GUI suffice. Make sure that the correct dataset is selected and hit run. This will prompt a second GUI which will ask you to save the ERP set.



2. Finally, to plot the ERPs, from the EEGLab menu bar, click ERPLAB > Plot ERP > Plot ERP waveforms. This gives the following GUI, which allows you to select which bins to plot, which channels to plot, what error bars to plot, etc.



3. Final plot should look something like …



… this!