**EEGLab and You: Estimating Alpha Power!**

1. **Setup**

**Download MATLAB:**

<https://www.mathworks.com/help/install/ug/install-products-with-internet-connection.html>

**Download EEGLab:**

<https://sccn.ucsd.edu/eeglab/download.php>

**Opening EEGLab:**

Unzip EEGLab folder to desktop (or wherever convenient) and select the following icon to open EEGLab syntax within MATLAB:



To launch EEGLab, click “Run” at the top of MATLAB to execute the script:



You should now a new window appear which looks like Windows ‘95!



*Note.* If you close this window, you will lose everything (i.e., no autosaving)

1. **Troubleshooting (recommended)**

**Restore legacy functions for artifact rejection:**

1. In EEGLab, click File 🡪 Preferences
2. Click the box next to “If set, show all menu items from previous EEGLAB versions…” and click OK. *Restart* EEGLAB for these options to appear!



**Install EDF Importer:**

1. In EEGLab, click File 🡪 Import Data 🡪 Using EEGLAB functions and plugins 🡪 From EDF/EDF+/GDF (BIOSIG Toolbox)
2. Click “Yes” when asked to download Biosig extension



1. Now you can open EDF files! It will prompt you to open a file once installed.

**Install Signal Processing Toolbox (MATLAB):**

1. In MATLAB, click the APPS tab on the top of the window and then click on “Get More Apps” to open the Add-on explorer:

****

1. In the search box (top right), search for “Signal Processing Toolbox”

****

1. Click onto the extension and click “Install” and follow the instructions.



*Note.* This extension is needed to compute alpha power with EEGLab!

1. **Importing EDF File**
2. Click File 🡪 Import Data 🡪 Using EEGLAB functions and plugins 🡪 From EDF/EDF+/GDF (BIOSIG Toolbox)
3. Navigate to where you have saved the EDF files and select file
	1. EEGLab will always default to the EEGLab folder
	2. For quick and easy access:
		1. Pin EDF folder to Quick Access (Windows)
		2. Move EDF folder to EEGLab folder
4. Once selected, you’ll be given options to load the EDF file
	1. Keep settings default (no changes)
	2. Click OK to proceed



1. Once imported, you need to name the file
	1. Naming is *very* important in EEGLab, helps keep track of changes
	2. Use ID provided by file when in doubt, which can be linked to Qualtrics
	3. Click OK



1. You should now see the EDF file loaded into EEGLab with provided name:



1. **Fixing and Checking**

**Fix channel locations:**

1. Click Edit 🡪 Channel Locations
2. Click OK on the pop-up window, should default to location of location data in your EEGLab plugins included with your original download:



1. In the next pop-up, if you see values in the X, Y, Z fields, you have correctly imported channel locations to your dataset. Click OK and go to next step.



1. If fields are blank, double check the first field: *Channel label (“label”)*
	1. If these do not follow typical 10-20 System, then change manually
	2. Once changed, click OK and repeat steps above to generate data for the remaining fields

**Check out your data:**

1. Click Plot 🡪 Channel data (scroll) to generate scrollable plot
2. Scroll using < or > buttons listed below data
3. Exit out of plot after viewing, do NOT exit out of the main EEGLab window…



**Check out your events:**

1. Click Edit 🡪 Event values
2. You should (hopefully) see the first even in your dataset with its label
3. Click “>” to go through all events to check all markers are present in the file (see study’s protocol to determine correct number of events)
4. Click OK when you have confirmed all events presented



*Note.* If an event is missing, you can “Insert event” using estimate of when it may have occurred. Save these files for last until you are confident of event timings. You will need to make sure the “Type” is identical to other files and adjust Edftype for all other events (i.e., if Trigger#1 is missing [Edftype = 1], you need to adjust remaining events (i.e., change Trigger#2 to Edftype = 2, Trigger#3 to Edftype = 3, etc.).

1. **Filtering your data**
2. Click Tools 🡪 Filter the data 🡪 Basic FIR filter (new, default)
3. Put lower frequency for your bandpass filter in “Lower edge…” and the higher frequency in the “Higher edge…” field
	1. To determine values, check previous publications (e.g., “…data were filtered using a .1–30hz bandpass filter.”)
	2. In this case, .1 would be the lower edge and 30 the higher edge
	3. Click OK to proceed



1. Rename the dataset and click OK
	1. Add FIR (or suffix of choice) to end of ID number
	2. This helps keep track of where you are in the process in case something goes wrong later, and you need to backtrack



1. View scrollable plot to check if filter was applied (i.e., smoother lines)
	1. Navigate to older datasets using the “Datasets” dropdown menu
	2. Importance of naming datasets!
2. **Extracting Events**
3. Choose event of interest (see study’s protocol)
	1. You may need to repeat this for several events
	2. All events are of interest, but baseline and task are most important
	3. Check with supervisor on where to focus analysis if time is limited
4. Click Tools 🡪 Extract epochs
5. Click the box with “…” next to Time-locking event type(s) ([]=all)
	1. Select event using labels in pop-up box (e.g., Trigger#1 = baseline)
	2. Click OK in “…” box

 🡺

1. In the “Epoch limits” field, you need to specify length of event (in seconds) with the start/end times separated by a single space
	1. Start time should be 0 (zero)
	2. End time is length of event in seconds (see study’s protocol)
2. Rename dataset in the “Name for the new dataset” field using the relevant field following the previous suffix (e.g., 134344 FIR BASELINE)
3. Click OK



1. Confirm event was segmented correctly by looking the “Epoch start (sec)” and “Epoch end (sec) in the details of your new dataset:



1. **Mini-epochs and Artifact Rejection**

**Creating 2-second epochs:**

1. With EEGLab open on your new dataset, navigate to the original MATLAB window used to launch EEGLab
2. Click on the line to the right of *f*x >> on the command window
	1. Flashing cursor should appear…
	2. You are now ready to use SYNTAX
3. Copy/past the following code into the command window and press ENTER:

EEG = eeg\_regepochs(EEG, 'recurrence', 2, 'limits', [0 2]);

eeglab redraw;

1. You should be asked to name the new dataset with an added suffix (i.e., -2-s epochs). Click OK.
	1. If not, paste text above in notepad and copy again
	2. This removes any weird formatting Word may have applied
2. Check to make sure the code worked properly by looking at number of epochs/events and epoch end (sec) information of your dataset:



**Rejecting bad epochs:**

1. Click Tools 🡪 Reject data epochs 🡪 Reject data (all methods)
	1. Requires legacy dialogs to be accessible
	2. See troubleshooting above
2. In new pop-up window, focus on:
	1. Find abnormal values
	2. Find abnormal trends
3. Find abnormal values *(absolute maximum threshold)*
	1. Upper and lower limits should be same
	2. By default, use +75 (upper limit) and -75 (lower limit)
	3. Check published articles for further guidance
4. Find abnormal trends *(maximum allowed voltage gradient)*
	1. By default, use 50 for Max slope (uV/epoch)
	2. Check published articles for further guidance
5. For each section above, you need to click “Calc/Plot” to determine what 2-second segments do not meet specified criteria. A scrollable window should pop up, which highlights bad segments in yellow
	1. Use > and >> to jump around the dataset
	2. Many segments will be yellow! Do not panic
	3. Once you are satisfied with the visual inspect, click “UPDATE MARKS”
	4. Do this for BOTH abnormal values and trends



1. Check “Currently marked trials” to check bad epochs have been flagged
	1. It is likely there is overlap between marked trials from different methods
	2. If no trials are marked and you expected them… try Calc/Plot again



1. On the bottom of the window, select “Reject marked trials”
	1. You will be asked to name new dataset, add REJ suffix
	2. If you cannot access bottom of window, press Windows Key + up arrow to maximize the window which should let you see the bottom buttons!



**Troubleshooting (all epochs rejected)**

1. Change electrodes target for artifact rejection
	1. If only interested in F3/F4, find these location numbers in the Channel locations window from earlier (use > to sort through)
	2. For Enobio 8 files, F3 = 2 and F4 = 3
	3. In EEGLab, this can be written as “2 3” or “2:3”
		1. Space = and
		2. Colon = all numbers between



1. If problems persist, you can:
	1. Try tweaking the values used to identify bad epochs
	2. Use more strict filters (e.g., 1-15hz)
		1. This may need to be applied to ALL participants…
		2. May be worth just excluding participant altogether
2. **Extracting Alpha Power**
3. Click on the line to the right of *f*x >> on the command window
	1. Flashing cursor should appear…
	2. You are now ready to use SYNTAX (again)
4. Copy/past the following code into the command window and press ENTER:

% Select channels 1-8

channel\_indices = 1:8;

EEG = pop\_select(EEG, 'channel', channel\_indices);

% Define the frequency range of interest

freq\_range = [8 13]; % Alpha frequency range (8-13 Hz)

% Set the FFT window type

window\_type = 'hamming';

% Compute the FFT for each epoch and channel

num\_epochs = size(EEG.data, 3); % Number of epochs

num\_channels = length(channel\_indices); % Number of channels

alpha\_power = zeros(num\_channels, 1); % Initialize the alpha power matrix

for channel = 1:num\_channels

 for epoch = 1:num\_epochs

 data = EEG.data(channel,:,epoch); % Extract data for a specific epoch and channel

 % Apply the Hamming window

 data = data .\* hamming(length(data));

 % Compute the FFT

 [fft\_data, freqs] = pwelch(data, [], [], [], EEG.srate);

 % Find the indices corresponding to the alpha frequency range

 alpha\_indices = find(freqs >= freq\_range(1) & freqs <= freq\_range(2));

 % Compute the alpha power for the current epoch and channel

 epoch\_alpha\_power = sum(fft\_data(alpha\_indices));

 % Accumulate the alpha power across epochs

 alpha\_power(channel) = alpha\_power(channel) + epoch\_alpha\_power;

 end

 % Compute the average alpha power across epochs for the current channel

 alpha\_power(channel) = alpha\_power(channel) / num\_epochs;

end

% Save the average alpha power data as "alphaplz.txt"

save('alphaplz.txt', 'alpha\_power', '-ascii');

1. This saves a text file to the EEGLab folder
	1. This may take a minute or two…
	2. The file is named “alphaplz.txt”
	3. The rows correspond with channel locations (e.g., F3 = 2nd row)
	4. The values are very small, will be log transformed later
2. Move this file to a new location and rename with the ID number of the participant and the event it captures (e.g., baseline).
	1. Files will be merged together later for data analysis
3. Repeat steps for EACH event… going back to the FIR dataset
	1. Accessible via Datasets dropdown menu
	2. You can clear datasets via File 🡪 Clear datasets if your computer starts to experience slowdown or you feel overwhelmed
4. You can SAVE your datasets by clicking File 🡪 Save current dataset as
	1. .SET file is EEGLab specific
	2. Easy to pickup where you left off
	3. Save at LEAST the FIR dataset as jumping off point for other events