EEG data analysis pipeline

Edited by M.G.

Date: Mar. 2021

Purpose: summarize analysis methods for general use @ Zhejiang University

Copyright © 2021, Mengyuan Gong

All rights reserved.

Overview: We used BioSemi as the EEG recording system at Zhejiang University. For conventional EEG data analysis, we preferred using ERPLAB as a plugin in EEGLAB.

**Important Materials**

A full documentation for ERPLAB is posted here: <https://github.com/lucklab/erplab/wiki/Manual>

Read this paper before you start to play with ERPLAB:

Lopez-Calderon, J. & Luck S.J. (2014). ERPLAB: an open-source toolbox for the analysis of event-related potentials. *Front Hum Neurosci*. 8(213):1-12

For advanced user, the ERP Methodology Blog could be particularly useful:

<https://erpinfo.org/blog?offset=1588392367635>

For time-frequency analysis of EEG data, FIELDTRIP toolbox could be better, a full documentation is posted here: <https://www.fieldtriptoolbox.org/documentation/>

To install plugins in EEGLAB, see the link below. In general, we installed BIOSIG, ERPLAB, CORRMAP, CleanRawData, Filedtrip-lite

<https://eeglab.org/others/EEGLAB_Extensions.html>

Below listed the EXG electrodes that are assigned to particular location (keep the naming consistent to avoid confusions)

|  |  |
| --- | --- |
| EXG1: Left mastoid (M1) | EXG3: left-upper eye (SO1) |
| EXG2: Right mastoid (M2) | EXG4: left-lower eye (IO1) |
| EXG7: backup electrodes for bad channels | EXG5: left-lateral eye (LO1) |
| EXG8: backup electrodes for bad channels | EXG6: right-lateral eye (LO2) |

For data recording: save each run/block of data and concatenate across runs/blocks after completing the experiment for each subject. This could be helpful when data recording is interrupted by accident.

ERPLAB: basic data processing pipeline

1. Loading data – Select reference electrodes (Pz is preferred, code: 31)

* Make sure you have BIOSIG Toolbox installed

A single channel should be used as reference when importing raw data to BioSemi system. We choose Pz (code: 31) because it is anatomically close to CMS-DRL location. All signals should be re-referenced to an average mastoid.

1. Append datasets

Edit -> Append datasets

Because we collected EEG data by run/block. We need to concatenate the datasets across runs for each subject before preprocessing.

1. Filter the data

ERPLAB -> Filter and frequency tools -> Filters for EEG data

* Make sure you have MATLAB Signal Processing Toolbox installed
* Filter type: IIR Butterworth
* Click “Apply filter to segments defined by boundary events”
* Cutoff frequencies -> Remove mean value (DC bias) before filtering

Note that BDF files have a large DC offset, you should either use BioSemi supplied "Converter" to BDF->EDF (I use the default setting of 0.16 hz highpass filter) or filter the data once in EEGLAB*.*

* Cutoff frequencies: 0.1 (high-pass) – 30 (low-pass)
* Roll-off & Filter order -> Order (2nd or 4th order filer, 12 or 24 dB/octave roll-off)

1. Adding channel locations

Edit -> Channel locations

Accept the default and define the EXG electrodes in the pop-out window

1. Downsample the data (optional: we do not do this in general)
2. Re-reference the data

5.1 Tools -> Re-reference the data

* Select M1 and M2 as the reference channels
* Exclude EOG channel

5.2 Alternative approach: ERPLAB -> EEG Channel operations

* Reference assistant -> Ch\_REF = (ch65 + ch66)/2 (Make sure “All channels” button is checked)
* <- Make sure “Create new dataset” is checked

This will re-reference the original data to the average of the left and right mastoids. However, after this step is done, the reference parameter shown in the GUI is still unknown??? (we need to confirm it).

1. Create the event list

ERPLAB -> EventList -> Create EEG Eventlist -> Advanced

EVENTLIST provides information for ach event, including a numeric code, a text label, a time of occurrence, enable/disable flag (to mark events that should be excluded because of errors during data collection)

* Enter Event Info and Bin Info, then click “update line”
* Equation List -> Save list as a text file
* Write resulting EVENTLIST to a text file (not sure why this step is necessary) -> Apply
* A pop-out window asked for Info to be used as Event Type, select “Numeric Codes”

1. Assign Bins

ERPLAB -> Assign bins (BINLISTER)

How to assign bins? See the “Cheat Sheet” below for detailed description

<https://github.com/lucklab/erplab/wiki/BDF-Library>

<https://github.com/lucklab/erplab/wiki/Assigning-Events-to-Bins-with-BINLISTER:-Tutorial>

1. Epoch data

ERPLAB -> Extract bin-based epochs

The choice of time window depends on your experiments. In general, we used a pre-stimulus period (-200 ms to 0 ms) as the baseline activity.

1. Artifact detection and rejection of bad epochs

9.1 How to deal with eye movement?

Rejection approach (this approach eliminates certain amount of trials)

ERPLAB -> Artifact detection in epoched data -> Moving window peak-to-peak threshold

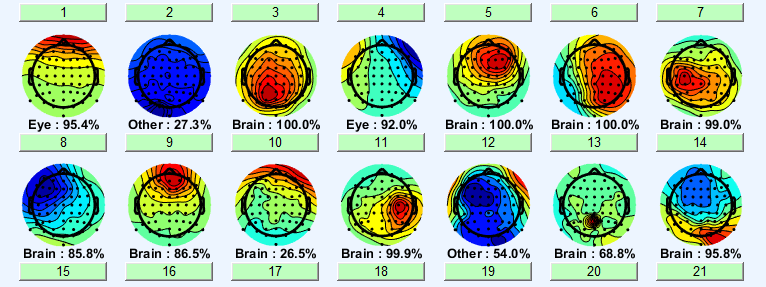
* Suggested voltage threshold: 40 µV. Run this on eye channels only with a window width of 200 ms and a stepsize of 100 ms.

ICA Correction (this correct, but not exclude, trials with eye movement)

Tools -> Decompose data by ICA

* Accept the default (ICA algorithm: “runica”) and select “EEG” as the channel type (this step took a long time).

Tools -> Classify components using ICLabel -> Label component



Tools -> Remove component from data

* Compare the data before versus after IC removal to ensure that obvious eye movements are corrected.

1. Having difficulty deciding which ICs should be removed? Here is a great overall that describes how to use the combination of scalp distribution, time course and frequency to distinguish among artifacts: <https://labeling.ucsd.edu/tutorial/labels>.
2. To understand the issues with ICA-based correction, see this link: <https://erpinfo.org/blog/2018/6/18/hints-for-using-ica-for-artifact-correction>.

9.2 How to deal with other general artifacts?

* Round 2: ERPLAB -> Artifact detection in epoched data -> Simple voltage threshold (suggested voltage threshold: ±75 µV). Run this on all scalp channels to exclude large artifacts.
* CORRMAP was recommended to perform a semi-automatic identification of common EEG artifacts (<http://www.debener.de/corrmap/corrmapplugin1.html>)
  1. Final visual inspection of artifacts

1. Average the ERPs

ERPLAB -> Compute averaged ERPs

* Keep default setting: exclude epochs marked during artifacts detection
* Keep track of the summary per bin (the output in command window tells you the proportion of rejected trials for each bin)

1. Compute any difference waves (e.g., contralateral vs. ipsilateral activity)

ERPLAB -> ERP operations -> ERP Bin Operations

<https://github.com/lucklab/erplab/wiki/ERP-Bin-Operations>

* Select channels on the left side (odd code, 1,3,5 …) and right side (even code, 2,4,6 …)
* Select the conditions (bin number)
* Define the condition names that correspond to the bins
* Check if the code in Bin Operation GUI looks right

1. Compute any regional activity:

ERPLAB -> ERP operations -> ERP Channel Operations

Channels can be created and modified as the Bin operation.

1. Scripting processing in EEGLAB/ERPLAB

* Make a script in EEGLAB is quite easy and useful for multi-subject processing. It does not even require much programming experience. You can do processing steps in the GUI manually and then reproduce the script command using *EEG.history or ERP.history* in Matlab command window.

See the example code below:

<https://github.com/lucklab/erplab/wiki/Example-3:-A-Simple-Script-That-Actually-Does-Something-Useful>