**Short tutorial: EEG data pre-processing and analysis**

Janir Ramos da Cruz; e-mail: janir.ramos@epfl.ch

# Pre-processing: APP pipeline

1. Set-up
   1. Start EEGLAB
   2. Install BIOSIG toolbox to import data: File >> Using EEGLAB functions and plugins >> From Biosemi BDF file (BIOSIG)
   3. Install CleanLine toolbox for powerline artifact removal: File >> Manage EEGLAB extensions >> Data processing extensions >> CleanLine
2. Automatic pre-processing of EEG data
   1. Run the first section of App pipeline (App\_rs.m) to set up some parameters for EEG data pre-processing
   2. Select the location of the patients EEG data ( …\Sample Data\SZPatients)
   3. Select the location of the controls EEG data ( …\Sample Data\Controls)
   4. Run the second section of the pipeline (the results will be in the Folder Results of the directory of each group)\*

\* I would suggest to make several breakpoints and see the output of each of the steps of the pipeline, if you are interested. These are the main steps of the pipeline:

1) High-pass filtering, to eliminate signal low-frequency non-stationarity (for example, slow drifts in the mean);

2) Removal of power line noise, with minimum distortion;

3) Robust re-referencing, to a robust estimate of the mean of all channels;

4) Detection, removal and interpolation of bad channels;

5) Detection and removal of bad epochs;

6) Detection and removal of eye movement-, muscular-, and bad channel-related artifacts based on ICA;

7) Detection, removal and interpolation of bad channels in epochs.

# Microstate analysis

1. Set-up
   1. Start EEGLAB
   2. Install the Microstates Analysis plugin: File >> Manage EEGLAB extensions >> Data processing extensions >> MicrostateAnalysis
2. Run the microstate analysis
   1. Run the first section of MicrostateAnalysis.m to select the pre-processed data from patients (first), controls (second) and then specify the folder to save to save your results
   2. Run the second section MicrostateAnalysis.m to load each subject data for each group, re-reference to the mean of all channels (in case your data is not already re-referenced to this), bandpass filter between 2 to 20 Hz (like Thomas Koenig does, or you can select 1 to 40, as Christoph Michel does).
   3. Identify the individual microstates maps: Tools >> Microstates >> Identify microstate maps

Parameters: Use AAHC algorithm instead of k-means (more efficient and does not depend on initial conditions), min number of classes = 3, max number of classes = 6, number of restarts (important only for k-means, use 20, 50, or more to avoid local minima), Max number of maps to use = 500 (to save time, but for better estimate use inf), GFP peaks only (have larger SNR), no polarity, show maps when done (for quality control)

* 1. Run the third section MicrostateAnalysis.m to do the same to all subjects
  2. Combine microstates maps across subjects to have mean cluster for each Group: Tools >> Microstates >> Average microstate maps across dataset

Select the patients dataset and name the mean GrandMean SZPatients

Do the same for controls and name the mean GrandMean Controls

* 1. Combine microstates maps across groups to have the Grand mean cluster: Tools >> Microstates >> Compute grand mean microstate maps across means

Select the two group means and name the grand mean as GrandGrandMean

Select Show maps when done

Choose More to see more clusters of microstates

* 1. Edit the “relative” position of the microstates to best fit the prototypical A,B,C,D microstates of the literature: Plot >> Edit microstates maps >> More >> Man. sort
  2. Sort each group microstates according to the grand mean microstates (GrandGrandMean): Tools >> Microstates >> Sort mean microstate maps according to grand mean

Select the two group means (GrandMean SZPatients, GrandMean Controls) and choose GrandGrandMean as the sorting “source”.

* 1. Sort the individuals microstates based on their group means (do for each group individually): Tools >> Microstates >> Sort individual microstate maps according to mean

E.g.: select the patients data and choose GrandMean SZPatients as the sorting “source” (do the same for controls).

* 1. Compute the classical microstates features: Tools >> Microstates >> Quantify microstates in dataset (mean template maps)

Select all the patients and control data and choose GrandGrandMean as the template mean that we want to use.

Parameters: Number of classes = 4, Fitting only on GFP peaks (if not you need to specify the Label smoothing window and Non-smoothness penalty, but you need to be an expert on this), Remove potentially truncated microstates.

Save the data as .csv which can be read by most statistical softwares

* 1. Statistical group analysis:

1. Download and install JASP (jasp-stats.org/download)
2. Load your .csv data on JASP
3. If you want to do statistics on the contribution (% of time spending on each microstate for each subject) for the factor of Group (Patients and Control) and Microstate (1, 2, 3, 4), you need to do a repeat measures ANOVA.

Select ANOVA >> Repeated Measures

In levels you should have 4 (A,B,C,D)

Select the 4 contribution columns to the Repeated Measures Cells

Select Group as Between Sujects Factors

# Relative Amplitude of Frequency Bands

* 1. Run the file to compute the relative frequency bands (ComputeRelAmplitudes.m)
  2. Select the location of your APP pre-processed EEG data ( …\Sample Data\Controls\Results)
  3. In the matlab file lines 50 to 62 you can modify the lower and upper bound of your relative frequencies to match the filtering you did using APP. For example, if you used from 1 to 40 (bandpass), then you should use from 1 to 40, instead of 1 to 70 as in the defaults.