WeBrain Tool Instruction

Li Dong: Lidong@uestc.edu.cn

Version：1.4
Date：5/8/2023

The Key Laboratory for NeuroInformation of Ministry of Education,
School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, 610054, China

Content

1. What is WeBrain

WeBrain is a web-based computing platform that enables large-scale EEG and EEG-fMRI multimodal data storing, exploring and analyzing using cloud High-Performance Computing (HPC) facilities across UESTC and China. WeBrain connects researchers of different fields to EEG and multimodal tools and processing power required to handle the large datasets that have become the norm in the field. It also aims to construct an International Virtual Community of Brainformatics (IVCB) to set the scene for more ambitious multi-national initiatives and cooperation in brain research. It does at the same time reduce the technical expertise required to use these resources. It provides an easy-to-use for novice users (even no computer programming skills) and flexibility for experienced researchers. It is not necessary to install any software or system for users, all need is a modern web browser of any kind. A range of resources including neuroimaging analysis tools are available, as well as documents related to WeBrain. Below are tools integrated in the WeBrain as yet.

2. Data input of Tools

Currently, in the WeBrain, EEG data of each subject should be zipped as a separated zip file ONLY. For example, a zip file of EEG data (one subject) can be a 'Sub_01.zip', which contains all EEG files generated by EEG system (e.g. files of BrainProduct EEG system: sub_01.dat, sub_01.vhdr and sub_01.vmrk) or a folder consisting of the EEG files. Here, 3 types of EEG data structures are supported in the WeBrain.

1) Associated with the WeBrain platform, a new and more flexible data structure, named the Standard EEG Data Structure (SEDS) is suggested. It is proposed to meet the needs of both small-scale EEG data batch processing in single-site studies and large-scale EEG data sharing and analysis in single-/multisite studies (especially on cloud platforms). Two versions (MATLAB and Docker versions) of the EEG Datafile Restructuring Toolbox (DRT) have been developed to restructure EEG data files according to the SEDS. The DRT GUI (MATLAB version) dramatically reduces the time required for novice researchers, while the DRT (Docker version) is more efficient for experienced researchers. All materials including SEDS documents, tools, example datasets, etc., are available on the WeBrain website (https://webrain.uestc.edu.cn/) and Wiki (https://github.com/WeCloudHub/DRT). More details about the SEDS can be seen in the paper: *Li Dong et al., 2021, DRT: A New Toolbox for the Standard EEG Data Structure in Large-scale EEG Applications, submitted.*

2) As an extension to the Brain Imaging Data Structure (BIDS) Specification for EEG, BIDS-EEG has been supported in the WeBrain. EEG data files could be reorganized as BIDS-EEG using BIDS-MATLAB-TOOLS (https://sccn.ucsd.edu/eeglab/plugin_uploader/plugin_list_all.php) first, and then

zipped as a zip files and uploaded to the WeBrain. More details about BIDS-EEG can be seen in the paper *Pernet, C. R., et al. (2019). "EEG-BIDS, an extension to the brain imaging data structure for electroencephalography." Sci Data 6(1): 103.*

3) For small-scale EEG data, data files could be directly zipped as a zip file by hand. The zip file should contain all EEG files generated by EEG system or a folder consisting of the EEG files.

Warning: DO NOT enter any blank spaces in the input file names!!! If there are any blank spaces in the file names, the WeBrain system will rename EEG data file names (it may lead to unexpected errors).

It is strongly suggested to reorganize EEG data files using offline tools before uploading these files to the WeBrain platform.

2.1 Load EEG data

EEGLAB functions (based on eeglab14_1_0b) are used to load various EEG data. If only channel labels are present currently, but some of these labels have known positions. It will try to look up coordinates for these channels using the electrode file Standard-10-5-Cap385_witheog.elp (except ANT). Currently, supporting EEG data format are:

EEGLAB .set File (recommended)
The function is 'pop_importdata()'. It supports reading the EEGLAB dataset files ({.set, .fdt}).
ASCII/Float .txt File or MATLAB .mat/.dat File
The function is 'pop_importdata()'. When reading a MATLAB .mat file, please confirm that it must contain only one MATLAB variable (channels × time points). When reading a ASCII .txt file, please confirm that is must contain only EEG data with channels × time points. If your data are .txt or .mat files, sampling rate must be filled in the parameter box when using some EEG tools. It is suggested to use DRT tool (https://webrain.uestc.edu.cn/) to convert EEG data to EEGLAB .set files, by writing sampling rate and channel locations.
Curry 6/7 .dat File
The function is 'pop_loadcurry()'. It supports reading NeuroScan Curry6/7 continuous EEG data files ({.dat, .dap and .rs3}).
Curry 8/9 .cdt File
The function is 'pop_loadcurry()'. It supports reading NeuroScan Curry8 continuous EEG data files ({.cdt, .cdt.ceo and .cdt.dpa}).
NeuroScan .cnt File
The function is 'pop_loadcnt()'. It supports reading NeuroScan EEG data files (.cnt). Data format 16bit or 32bit will be auto-detected.

NeuroScan .EEG File

The function is ‘pop_loadeeg()’. It supports reading the NeuroScan EEG data files (.EEG). If the .EEG file is epoched data (channels × time points × trials), some tools may be not available.

Biosemi .bdf/.edf File

The function is ‘pop_readbdf()’. Based on Biosemi functions, it supports reading 16-bit European standard "EDF" (European Data Format) and 24-bit BDF (Biosemi Data Format).

Brain Vis. Rec. .vhdr File

The function is ‘pop_loadbv()’. It supports reading Brain Vision Analyzer data files ({.dat, .vhdr, .vmrk} or {.eeg, .vhdr, .vmrk}).

EGI MFF File

The function is ‘pop_readegimff ()’. It supports reading EGI MFF data files (.mff).

博瑞康 Neuracle EEG Recorder

The function is ‘pop_importNeuracle ()’. It supports reading 博瑞康 Neuracle EEG Recorder data files ({data.bdf, evt.bdf}). Noting that the filenames of EEG file collected by Neuracle EEG Recorder are fixed as “data”, “evt”.

ANT .cnt File

The function is ‘pop_loadeep_v4 ()’. It supports reading ANT cnt data files (.cnt). If only channel labels are present currently, but some of these labels have known positions. WeBrain will try to look up coordinates for these channels using the electrode file ‘file ANT_WG_standard_346.ced’ from plugin “ANTeepimport1.13”.

Table 1: Data formats supported by the WeBrain

	Supported file formats
EEG file formats (Manufacturer/file format)	ASCII/Float file (*.txt) MATLAB (*.mat or *.dat) EEGLAB ({*.set, *.fdt} or *.fdt) Curry 6/7 ({*.dat, *.dap, *.rs3}) Curry 8/9 ({*.cdt, *.cdt.ceo, *.cdt.dpa}) Brain Products/Brain Vision ({*.vhdr, *.vmrk, *.dat} or {*.vhdr, *.vmrk, *.eeg}) NeuroScan (*.cnt or *.EEG) Biosemi/European Data Format (*.bdf or *.edf) BIOSIG (*.edf, *.edf+, *.gdf or *.bdf) EGI MFF file (.mff) 博瑞康 Neuracle EEG Recorder ({data.bdf, evt.bdf}) ANT (*.cnt)

3. EEG Tools

Warning: DO NOT enter any blank spaces in the input parameters!!!

3.1 WB_EEG_REST

WB_EEG_REST is a tool of Reference Electrode Standardization Technique (REST) in WeBrain. REST is a re-reference technique, a software method for translating multichannel spontaneous EEG or event-related potentials with reference at any a physical point on brain/body surface or the post-processed data referenced at average or linked ears etc. to a new dataset with reference at Infinity where the potential is zero/constant (Yao, 2001; Yao et al., 2005). Currently, REST is increasingly acknowledged by EEG/ERPs community around the world (to our knowledge, at least 12 countries/areas), and more than 50 studies have actually adopted REST to get zero reference as the foundation of their novel findings. Meanwhile, the REST has been regarded as the Rosetta Stone for scalp EEG (Kayser and Tenke, 2010) and listed in the new guidelines of International Federation of Clinical Neurophysiology (IFCN) for EEG analysis. More details about REST toolbox can also be seen in the paper (Dong et al., 2017).

Use REST please cite:

Yao, D., 2001. A method to standardize a reference of scalp EEG recordings to a point at infinity. Physiol Meas. 22, 693-711.

Dong, L., et al., 2017. MATLAB Toolboxes for Reference Electrode Standardization Technique (REST) of Scalp EEG. Frontiers in Neuroscience. 11.

Parameters

lfFile: Leadfield file.The leadfield can be a matrix (*.dat, output of leadfield.exe, sources X channels) which is calculated by using the forward theory, based on the electrode montage, head model and equivalent source model. It can also be the output of ft_prepare_leadfield.m (e.g. lf.leadfield) based on real head modal using FieldTrip.

rechanns: string with indices of the re-referenced channels (channels re-referencing to REST, e.g. '[1:4,7:30]'), or 'all'.

Outputs

For each subject, a zip file which contain the re-referenced EEG data will be generated (saved as EEG .set file which contains the EEG potentials with zero reference, channels × time points).

Leadfield Calculation

The leadfield matrix is required to be calculated for a new electrode system. It can be

the output of ft_prepare_leadfield.m (e.g. lf.leadfield) based on real head modal using FieldTrip. Users can also download the 'Leadfield.exe' to calculate a leadfield matrix of a concentric-three-spheres head model. It calculates the leadfield matrix from the 3000 cortical dipoles (spherical equivalent dipoles) and the newly given electrode array for the canonical concentric-three-spheres head model. The array of real electrode coordinates (coordinates of fiducial points are not required) is suggested to be saved in a '*.txt' ASCII file with their Cartesian x (the left ear is defined as -x axis), y (the nasion is the +y axis), z coordinates in three columns, while the coordinates will be auto-normalized and -matched to the upper spherical cap of head model inside the program. In addition, noting that the executable file 'Leadfield.exe' is compiled using C language on Windows system to calculate the leadfield matrix; if you want to run it on Linux system (Ubuntu), a simple solution is to install the 'Wine' software first (i.e. enter the command 'sudo apt-get install wine' in a terminal). The leadfield calculation consists of the following 2 steps.

1. File → Load Electrode File: '*.txt' ASCII file; x, y, z positions in three columns only;

2. File → Calculate Lead Field. It may take a few minutes that depends on the size of the matrix and the computer. When the calculation is completed, the leadfield matrix is saved as 'Lead_Field.dat' (sources × channels) in the directory of electrode file.

Note

The reference of input EEG data should be a scalp point (e.g. the tip of the nose, vertex, linked mastoids or linked ears) or average reference.

Links

REST:
http://www.neuro.uestc.edu.cn/rest/

REST software
http://www.neuro.uestc.edu.cn/rest/Down.html

EEGLAB
http://sccn.ucsd.edu/eeglab/index.html

FieldTrip
http://www.fieldtriptoolbox.org/start
http://www.fieldtriptoolbox.org/reference/ft_prepare_leadfield
http://www.fieldtriptoolbox.org/development/project/example_fem
http://www.fieldtriptoolbox.org/tutorial/headmodel_eeg_fem?s[]=fem&s[]=headmodel

3.2 WB_EEG_Mark

WB_EEG_Mark is a tool to automatically mark bad block/good quality EEG data based on thresholding z-scores/global field power (Fig. 1). It is recommended before calculating EEG indices (e.g. power, networks) or ERPs. Steps of marking EEG data consist of:

[1] Filtering all EEG data (Passband filtering and Notch filter)

[2] Z-transforming the EEG data/calculating global field power. Per channel/electrode every time point is z-normalized (mean subtracted and divided by standard deviation). Or the z-scored standard deviation (global field power, GFP) of the signal at all selected electrodes is calculated.

[3] Averaging z-values/using global field power over channels/electrodes allows evidence for an artifact to accumulate and averaging it over channels.

[4] Threshold the accumulated z-score/global field power for each epoch/window. Bad blocks are labeled by '9999' (EEG.event.type is '9999', percentage (absolute value) above threshold > 1% for each small epoch). Good quality data are labeled by '2001' (EEG.event.type is '2001', percentage (absolute value) above threshold < 5% for each small epoch). If bad blocks with label '9999' already existed, the bad block data will be NOT marked as good quality data.

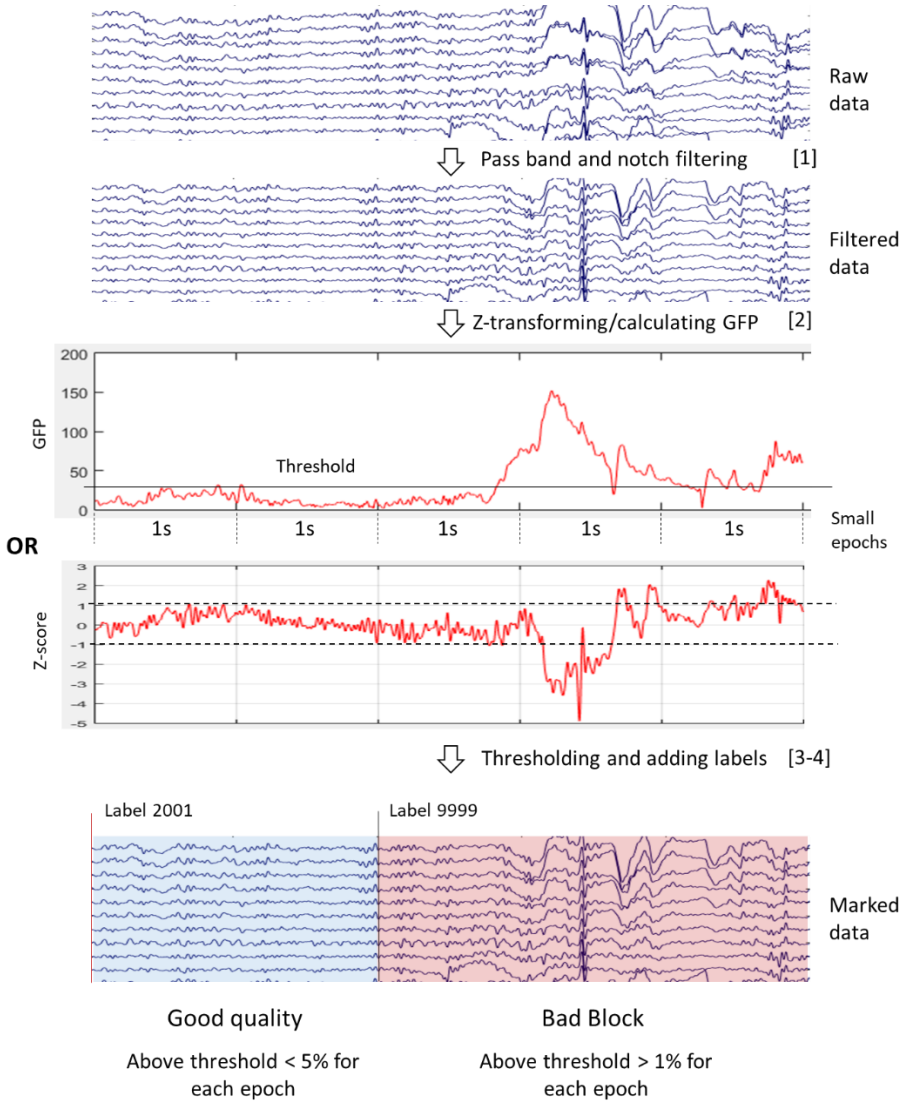

Fig. 1: Pipeline of automatically marking bad block/good quality EEG data.

Parameters

passband: Passband of filtering. Default is '[1,60]'. If passband is empty ('[]'), bandpass filtering will be skipped.

NotchBand: Band of notch filter. Default is '[45,55]'. In China, power frequency is 50Hz, while it is 60 Hz in USA. If max(passband)<min(NotchBand) or NotchBand is empty, notch filtering will be skipped.

flag1: flag1 = 0: mark bad blocks (Default); flag1 = 1: mark good quality data.

flag2: flag2 = 0: global field power (Default); flag2 = 1: z-tranforming.

Thre: Threshold of z-score/global field power. Default of global field power (z-scored standard deviation) is 3. For various EEG data, the threshold may be changed by user flexibly.

WinLenth: Length of the window (small epoch). Unit is second. Default is 1 sec.

seleChanns: String with indices of the selected channels (e.g. '[1:4,7:30]'), or 'all'. Default is 'all'.

srate: Sampling rate of EEG data. It can be automatically detected from EEG data, if it is '[]'. But for ASCII/Float .txt File and MATLAB .mat File, users should fill the sampling rate by hand. Default is '[]'.

Outputs

For each subject, a zip file which contains the marked EEG data (saved as EEG .set file which contains the mark events) will be generated. Bad blocks are labeled by '9999' (EEG.event.type is '9999'). Good quality data were labeled by '2001' (EEG.event.type is '2001').

EEG.MarkPercent: Percentage of marked event (duration).

EEG.dataZ: Z-score or z-scored global field power of data.

Links

Automatic artifact rejection in FieldTrip

http://www.fieldtriptoolbox.org/tutorial/automatic_artifact_rejection

3.3 WB_EEG_runICA

WB_EEG_runICA is a tool to run ICA on EEG data based on EEGLAB function - runica(). It perform Independent Component Analysis (ICA) decomposition of input data using the logistic infomax ICA algorithm of Bell & Sejnowski (1995) with the natural gradient feature of Amari, Cichocki & Yang, or optionally the extended-ICA algorithm of Lee, Girolami & Sejnowski, with optional PCA dimension reduction. Annealing based on weight changes is used to automate the separation process. ICA is usually used to remove artifact (e.g. eye blink) or extract features (e.g. ERP) from EEG data.

Parameters

selechanns: number with indices of the selected channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all'.

ICs: Number of ICA components. Default is number of EEG channels or number of retained PCs. Noting that we usually run ICA using many more trials that the sample decomposition presented here. As a general rule, finding N stable components (from N-channel data) typically requires more than kN^2 data sample points (at each channel), where N^2 is the number of weights in the unmixing matrix that ICA is trying to learn and k is a multiplier. The value of k will increase as the number of channels increases. When data are insufficient, then using the 'pca' option (Set the number of PCs to retain) to find fewer than N components may be the only good option. This parameter may return strange results. This is because the weight matrix is rectangular instead of being square.

Ntrain : Perform tanh() "extended-ICA" with sign estimation N training blocks. If N > 0, automatically estimate the number of sub-Gaussian sources. If N < 0, fix number of sub-Gaussian comps to -N [faster than N>0] (default is 0 -> off). The "runica" Infomax algorithm can only select for components with a supergaussian activity distribution (i.e., more highly peaked than a Gaussian, something like an inverted T). If there is strong line noise in the data, it is preferable to set the training blocks as 1, so the algorithm can also detect subgaussian sources of activity, such as line current and/or slow activity.

PCs: decompose a principal component (default = [] -> off, i.e. default is NOT performing PCA analysis.) subspace of the data. Value is the number of PCs to retain.

stop: stop training when weight-change < this (default is 1e-6, if less than 33 channel 1e-7 is recommended).

maxsteps: max number of ICA training steps. Default is 512.

sphering: ['on'/'off'] flag sphering of data. Default is 'on'.

Note: EEG data will be imported as EEG structure using EEGLAB. EEG.data should be channels × time points OR channels × time points × epochs.

Outputs

For each subject, a zip file which contain the EEG dataset with new fields icaweights, icasphere and icachansind (channel indices). (saved as EEG .set file which contains the ICA results). The EEG dataset can also be imported and used in EEGLAB.

EEG.icasphere: ICA sphere;

EEG.icaweights: ICA weights;

EEG.icachansind: selcet channels;

EEG.activations: ICA components. If EEG data is epoched, the activations (corresponding to ICA time courses) will be generated;

EEG.icawinv = pinv(icaweights * icasphere); % a priori same result as inv.

EEG.ICAPara.ICs: number of ICA components;

EEG.ICAPara.Ntrain: perform tanh() "extended-ICA" with sign estimation N training blocks;

EEG.ICAPara.PCs: the number of PCs to retain.

EEG.ICAPara.stop: stop training when weight-change < this.

EEG.ICAPara.MaxSteps: max number of ICA training steps;

EEG.ICAPara.sphering: flag sphering of data;

3.4 WB_EEG_QA

WB_EEG_QA is a stable tool to realize quality assessment (QA) of a continuous EEG raw data (e.g, resting-state EEG data). The bad data in small windows of each channel could be detected by kinds of 4 methods, and a number of indices related to the data quality will be calculated. Meanwhile, the overall data quality rating will be also provided, including levels of A, B, C, D (corresponding to perfect, good, poor, bad). The QA consists of (Fig. 2):

[1] A continuous EEG data of each channel will be high pass filtered (>1Hz) and then segmented as small windows;

[2] Detecting constant or NaN/Inf signals in each window (Method 1). The windows containing any NaN/Inf or with tiny SD/median values ($<10^{-10}$) are considered as bad windows.

[3] Detecting unusually high or low amplitude using robust standard deviation across time points in each window (Method 2). If the z-score of robust time deviation falls below a threshold or the absolute amplitude exceeds a value of microvolts (150μV), the window is considered to be bad.

[4] Detecting high or power frequency noises in each window by calculating the noise-to-signal ratio (NSR) based on Christian Kothe's method (Method 3). If the z-score of estimate of signal above 40/50Hz (power frequency minus 10 Hz) to that below 40/50 Hz above a threshold or absolute NSR exceeds 0.5, the small window is considered to be bad.

[5] Detecting low correlations with other channels in each window using Pearson correlation (default) or RANSAC correlation (Method 4). If the maximum correlation (absolute correlation coefficients) of the window of a channel to the other channels falls below a threshold, the window is considered bad.

[6] Calculating a number of indices relative to the data quality and rating the EEG raw data.

More details about the QA tool can be seen in the paper: *Zhao et al., 2021, Quantitative signal quality assessment for large-scale continuous scalp EEG with big data perspective, submitted.*

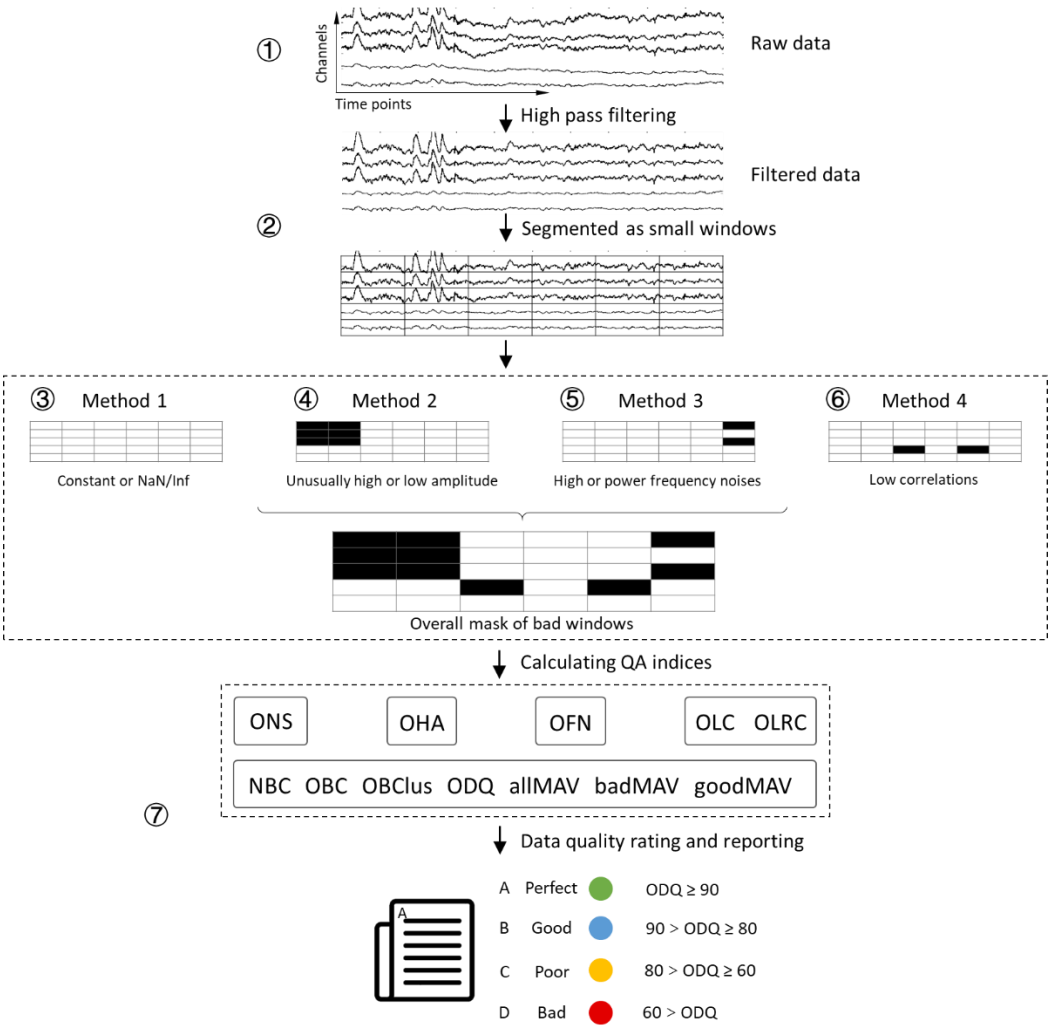

Fig. 2: Pipeline of quality assessment of continuous EEG raw data. (1) Raw EEG data with artifacts such as eye blink, eye movement etc. (2) The continuous EEG data of each channel will be high pass filtered and then segmented as small windows. Here 'WindowSeconds' is the window size (e.g. 1 sec.) over which the following methods are conducted. (3) Detecting constant or NaN/Inf signals in each window (Method 1). (4) Detecting unusually high or low amplitude using robust standard deviation across time points in each window (Method 2). If the z score of robust time deviation falls below 'robustDeviationThreshold' or the absolute amplitude exceeds 150 microvolts (μV), the small window is considered to be bad. (5) Detecting high or power frequency noises in each window by calculating the noise-to-signal ratio based on Christian Kothe's method (Method 3) (clean_rawdata0.32 https://sccn.ucsd.edu/wiki/Artifact_Subspace_Reconstruction_(ASR)). If the z score of estimate of signal above 40 Hz (power frequency - 10Hz) to that below 40 Hz above 'highFrequencyNoiseThreshold' or absolute NSR exceeds 0.5, the small window is considered to be bad. Noting that if the sampling rate is below 2×power frequency, this step will be skipped. (6) Detecting low correlations with other channels in each window using Pearson correlation (default) or RANSAC correlation (Method 4). For Pearson correlation, if the maximum correlation of the window of a

channel to the other channels falls below 'correlationThreshold', the window is considered bad. For RANSAC correlation (Bigdely-Shamlo et al., 2015), each window of a channel is predicted using RANSAC interpolation based on a RANSAC fraction of the channels. If the correlation of the prediction to the actual behavior falls below 'ransacCorrelationThreshold' or calculation is too long, the window is marked as bad. The time cost of this method is high, and the channel locations are required. The RANSAC correlation is optional and default is not performed. (7) Calculating a number of indices relative to the data quality and rating the EEG raw data.

Parameters

WindowSeconds: the window size (in seconds, default = 1 sec.) over which the above methods are conducted.

HighPassband: lower edge of the frequency for high pass filtering. Default is 1 Hz.

seleChanns: number with indices of the selected EEG channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all'.

badWindowThreshold: cutoff fraction of bad windows (default = 0.4) for detecting bad channels.

robustDeviationThreshold: Z-score cutoff for robust time deviation in each window (default = 5).

PowerFrequency: power frequency. Default is 50 Hz (in Chinese). Noting that in USA, power frequency is 60Hz.

FrequencyNoiseThreshold: Z-score cutoff for NSR (signal above power frequency - 10Hz). Default is 3. If the z score of estimate of signal above 40 Hz (power frequency - 10Hz) to that below 40 Hz above 'highFrequencyNoiseThreshold' or absolute NSR exceeds 0.5, the small window is considered to be bad.

flagNotchFilter : flagNotchFilter = 1: remove 0.5×power frequency noise using notch filtering. Default is off (flagNotchFilter = 0).

correlationThreshold: maximal correlation below which window is bad (range is (0,1), default = 0.6). If the maximum correlation of the window of a channel to the other channels falls below 'correlationThreshold', the window is considered bad.

ransacCorrelationThreshold: cutoff correlation for abnormal wrt neighbors(default = [] | --> not performed).

ransacChannelFraction: fraction of channels for robust reconstruction (default = 0.3).

ransacSampleSize: samples for computing RANSAC (default = 50).

srate: sampling rate of EEG data. It can be automatically detected in EEG data. But for ASCII/Float .txt File or MATLAB .mat File, user should fill the sampling rate by hand. Default is '[]'.

Note:

- Assumptions of QA tool:
  - The signal is a structure of continuous data with data and sampling rate at least.

- No segments of the EEG data have been removed.

- Noting that quality assessing EEG raw data would NOT change the raw data.
- If channel locations are not contained in EEG data or selected channels do not contain channel locations, the RANSAC correlation is invalid.
- Noting that if the sampling rate is below 2×power frequency, the step of detecting high or power frequency noises will be skipped.

Outputs

For each subject, a mat file which contains a structure array of QA results will be generated (saved as results_QA_*.mat file which contains the QA results and parameters of each step). Meanwhile, a table file named QA_table.mat which lists all indices of all subjects (including calculated and skipped subjects) in a table will be generated at same time.

results_QA.ONS: Overall ratio of No Signal windows. The ONS ranges from 0 to 1. The ONS = 0 if and only if there is no NaN or constant signals in the data. In contrast, the ONS = 1 for all NaN or constant signals;

results_QA.OHA: Overall ratio of High Amplitude windows. The OHA ranges from 0 to 1. The OHA = 0 if and only if there is no high amplitude window in the data, and the OHA = 1 for all are high amplitude bad windows in the data;

results_QA.OFN: Overall ratio of high Frequency Noise windows. The OFN ranges from 0 to 1. The OFN = 0 if and only if there is no high frequency noise window in the data, and the OFN = 1 for all are high amplitude bad windows in the data;

results_QA.OLC: Overall ratio of Low Correlation windows. The OLC ranges from 0 to 1. The OLC = 0 if and only if there is no low correlation windows in the data, and the OLC = 1 for all windows are low correlation bad windows in the data;

results_QA.OLRC: Overall ratio of windows of Low RANSAC Correlation (optional). The OLRC = 0 if and only if there is no low correlation windows in the data, and the OLRC = 1 for all windows are low correlation bad windows in the data;

results_QA.badChannels: The index of bad channels of which the ratio of the bad quality windows exceed a certain threshold (0.4 by default);

results_QA.NBC: No. of Bad Channels;

results_QA.OBC: Overall ratio of Bad Channels. The OBC tends to 0 for no bad channels and to 1 for all bad channels;

results_QA.OBClus: Overall ratio of Bad Clusters. The number of the connected components of the bad quality windows. This measure can describe the situations of the bad quality windows in the data. OBClus tends to 1 for a wide noises in the data, to 0 for no bad clusters. The lower of OBClus is, the less part of EEG signals is contaminated. If ODQs of two EEG data are same, the quality of the data with lower OBClus is better than another;

results_QA.ODQ: Overall Data Quality: the overall ratio of good data windows. The ODQ ranges from 0 to 100. The ODQ = 0 if and only if there is no good window

in the data, and the ODQ = 100 for all are good windows in the data;

results_QA.DataQualityRating: Overall Data Quality Rating

Level A: ODQ >= 90;

Level B: ODQ >= 80 && ODQ < 90;

Level C: ODQ >= 60 && ODQ < 80;

Level D: ODQ < 60;

results_QA.allMAV: mean absolute value of all windows;

results_QA.badMAV: mean absolute value of bad windows;

results_QA.goodMAV: mean absolute value of good windows;

results_QA.NoSignalMask: a mask of windows with no signals (with dimension channels × windows);

results_QA.AmpliChannelMask: a mask of windows with high amplitudes (with dimension channels × windows);

results_QA.FrequencyNoiseMask: a mask of windows with high frequency (and power frequency, if applicable) noise (with dimension channels × windows);

results_QA.LowCorrelationMask: a mask of windows with low correlations (with dimension channels × windows);

results_QA.RansacBadWindowMask: a mask of windows with RANSAC low correlations (with dimension channels × windows);

results_QA.OverallBadMask: a mask of windows with overall bad signals (with dimension channels × windows);

results_QA.fractionBadWindows: fractions of bad windows for each channel (with dimension channels × 1);

results_QA.badChannelsFromAll: logical value of bad channels from all methods (with dimension channels × 1).

Parameter details:

results_QA.parameters.srate: sampling rate;

results_QA.parameters.WindowSeconds: window size in seconds (default = 1 sec);

results_QA.parameters.HighPassband: lower edge of the frequency for high pass filtering, Hz;

results_QA.parameters.selechanns: number with indices of the selected channels (e.g. [1:4,7:30] or 'all').Default is 'all';

results_QA.parameters.badWindowThreshold: cutoff fraction of bad windows;

results_QA.parameters.PowerFrequency: power frequency. Default is 50 Hz (in Chinese). Noting that in USA, power frequency is 60Hz;

results_QA.parameters.robustDeviationThreshold: Z-score cutoff for robust channel deviation;

results_QA.parameters.FrequencyNoiseThreshold: Z-score cutoff for nosie-to-signal ratio (signal above 40 Hz);

results_QA.parameters.correlationThreshold: maximal correlation below which

window is bad (range is (0,1));

results_QA.parameters.chanlocsflag: flag of channel locations. if chanlocsflag = 1: have channel locations;

results_QA.parameters.chanlocsXYZ: xyz coordinates of selected channels;

results_QA.parameters.chanlocs: channel locations of selected channels;

results_QA.parameters.ransacSampleSize: samples for computing RANSAC (default = 50);

results_QA.parameters.ransacChannelFraction: fraction of channels for robust reconstruction (default = 0.3);

results_QA.parameters.ransacCorrelationThreshold: cutoff correlation for abnormal wrt neighbors(default = [] | --> not performed).

3.5 WB_EEG_prepro

WB_EEG_prepro is a specific and stable tool to perform standardized preprocessing of continuous EEG raw data to remove a kind of artifacts (e.g. resting state EEG data), and obtain clean EEG data with REST reference. It is supported to preprocess EEG raw data with single point, average or linked LM reference. Preprocessing EEG raw data consists of (Fig. 3):

[1] Quality assessment of EEG raw data first. Noting that quality assessment (QA) do NOT change the EEG raw data. If the overall data quality (ODQ) exceed a threshold (default is 80), then the preprocessing could be continue;

[2] Passband and notch filtering, if applicable;

[3] Artifact removal: EOG regression;

[4] Artifact removal: residual artifact removal;

[5] Bad channel interpolation and re-referencing to REST;

[6] Quality assessment of preprocessed EEG data after artifact removal;

[7] Marking residual bad block with unusually high or low amplitude (>6) using z-scored STD across channels, and then clean EEG data with REST reference are obtained finally.

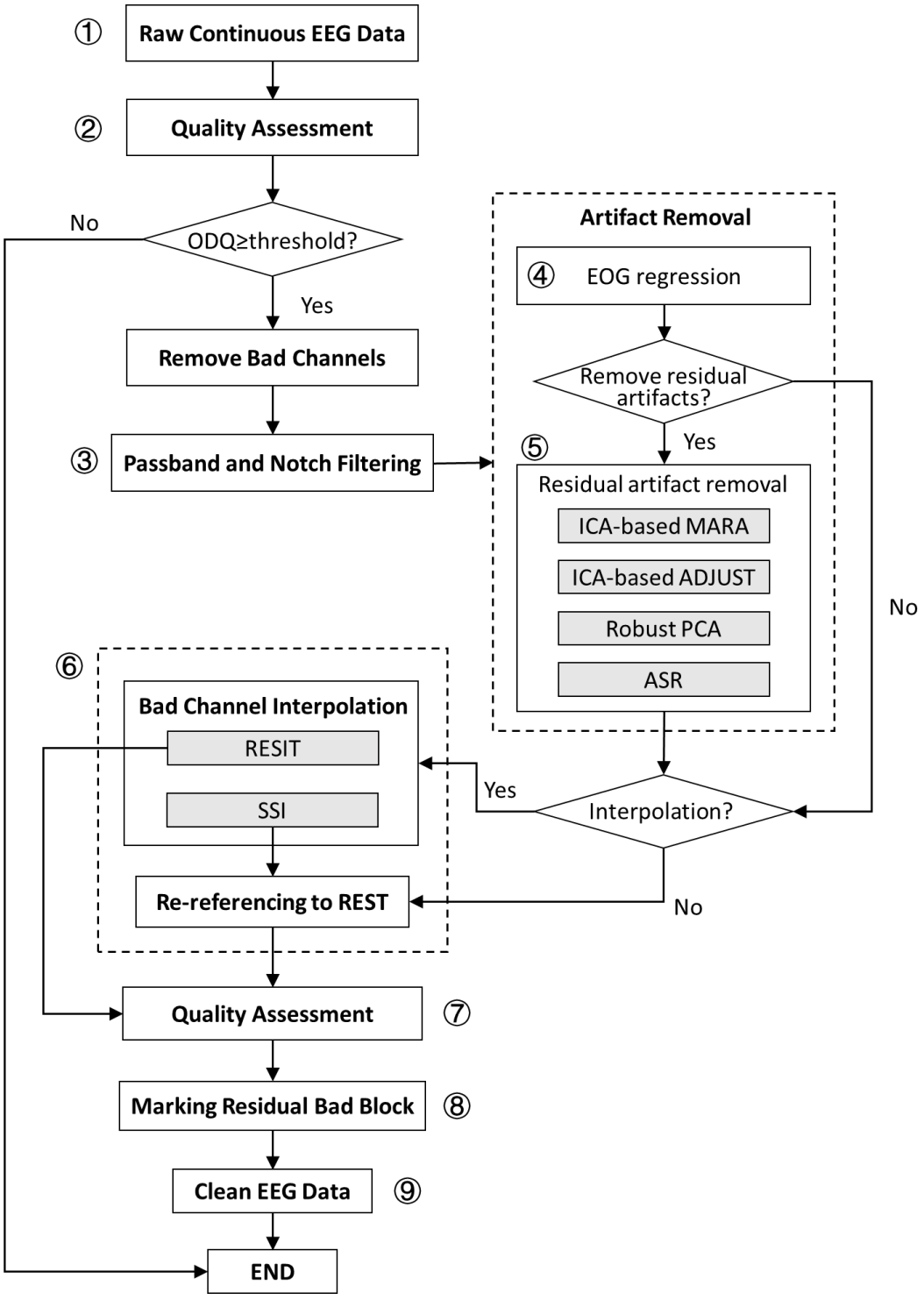

Fig. 3: Pipeline of standardized preprocessing of continuous EEG raw data. (1) Raw EEG data with artifacts such as eye blink, eye movement etc. (2) Quality assessment of EEG raw data, automatically. If the overall data quality (ODQ) exceed a threshold (ranging from 0-100, default is 80), then the preprocessing could be continue. The EEG raw data will be assessed by kinds of methods, and bad channels will be identified at same time. More details can be seen in WB_EEG_QA. (3) Passband and notch filtering (if applicable). The data can be filtered first using Hamming windowed

sinc FIR filter. (4) Artifact removal: EOG regression. A linear regression model will be utilized to remove EOG artifacts using z-scored EOG channels (5) Artifact removal: residual artifact removal. The residual artifacts will be removed using common used methods. Currently, a kind of 4 common used methods are provided in the WeBrain, including ICA based Multiple Artifact Rejection Algorithm (MARA), ICA based ADJUST, robust PCA and artifact subspace reconstruction (ASR) methods. (6) Bad channel interpolation and re-referencing to REST. For interpolation, reference electrode standardization interpolation technique (RESIT) and spherical spline interpolation (SSI) are provided in the WeBrain. (7) Quality assessment of preprocessed EEG data after artifact removal. (8) Marking residual bad block with unusually high or low amplitude using z-scored standard deviation (STD>6) across channels. Bad blocks will be marked as label 9999 using WB_EEG_Mark. (9) Clean EEG data with REST reference are obtained finally.

Parameters

seleChanns: number with indices of the selected channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all'.

EOGchanns: number with indices of the EOG channels. Default is '[]'.

thre_ODQ: threshold of overall data quality (ODQ). If ODQ ≥ a threshold, then the preprocessing could be continue. Default is 80. Noting that the quality assessment (QA) would NOT change the EEG raw data, and some default QA parameters are fixed in the preprocessing tool (window size is 1 sec, lower edge of the frequency for high pass filtering is 1 Hz, cutoff fraction of bad windows is 0.4, Z-score cutoff for robust channel deviation is 5, Z-score cutoff for noise-to-signal rate is 3 and correlation threshold is 0.6).

passband: passband of filtering. Default is '[1,40]'.

PowerFrequency: power frequency. Default is 50 Hz (in Chinese). Noting that in USA, power frequency is 60Hz.

keepUnselectChannsFlag:

keepUnselectChannsFlag = 0: do not keep unselected channels (default); keepUnselectChannsFlag = 1: keep all channels.

badChannelInterploateFlag:

badChannelInterploateFlag = 0: do NOT interpolate, and if have channel locations in EEG.chanlocs, then re-referencing to REST (Dong et al., 2017; Yao, 2001);

badChannelInterploateFlag = 1 (default): interpolate the bad channels rows of EEG.data using reference electrode standardization interpolation technique (RESIT); default is using RESIT (The bad channels will be interpolated with REST reference (Dong et al., 2017; Dong et al., 2021; Yao, 2001));

badChannelInterploateFlag = 2: interpolate the bad channels rows of EEG.data using spherical spline interpolation (SSI) (Perrin et al., 1989), and

re-referencing to REST.

residualArtifactRemovalFlag:

residualArtifactRemovalFlag = 0: no removal;

residualArtifactRemovalFlag = 1: ICA based MARA (Multiple Artifact Rejection Algorithm) (Winkler et al., 2011);

residualArtifactRemovalFlag = 2: ICA based ADJUST (Mognon et al., 2011);

residualArtifactRemovalFlag = 3: rPCA method (Lin et al., 2010);

residualArtifactRemovalFlag = 4: ASR method (Mullen et al., 2013).

MARA_thre: cuttoff posterior probability for each IC of being an artefact while using MARA method. Default is 0.7.

badWindowThreshold: cutoff fraction of bad windows (default = 0.6) for detecting bad channels.

srate: sampling rate of EEG data. It can be automatically detected in EEG data. But for ASCII/Float .txt File or MATLAB .mat File, user should fill the sampling rate by hand. Default is '[]'.

Note:

- Noting that quality assessing EEG raw data would NOT change the raw data. If the overall data quality (ODQ) exceed a threshold (default is 80), then the preprocessing could be continue.
- EEG data will be imported as EEG structure of EEGLAB. Dimension of EEG.data must be channels × time points.
- If channel locations are not contained in EEG data or selected channels do not contain locations, methods required EEG channel coordinates are invalid (e.g. interpolation, ICA-based MARA, ICA-based ADJUST, and REST re-referencing etc.).
- It is suggested to use this tool to preprocess EEG raw data with single point/AVG/linked LM reference. It is not supported for the EEG raw data with a specific non-unipolar recording montage, such as the ipsilateral mastoid (IM) or the contralateral mastoid (CM).

Outputs

For each subject, a zip file which contains the preprocessed EEG data will be generated (saved as *_prepro.set file which contains the clean EEG data (EEG.data with dimension channels × time points) and preprocessing info (parameters and results of each preprocessing step). The file can be further analyzed by WeBrain online or EEGLAB offline. Following fields will be further added in the *.set file.

EEG.preprocessed.PassbandFilter.check = 'yes' or 'no' for pass band filtering;

EEG.preprocessed.PassbandFilter.passband: pass band;

EEG.preprocessed.PassbandFilter.comments = 'Hamming windowed sinc FIR filter';

EEG.preprocessed.NotchFilter.check = 'yes' or 'no' for notch filtering;

EEG.preprocessed.NotchFilter.notchband: notch filtering band;
EEG.preprocessed.EOGregression.check = ‘yes’ or ‘no’ for EOG regression
EEG.preprocessed.EOGregression.EOGchanns = EOG channels;

EEG.preprocessed.residualArtifactRemoval.check = ‘no’: skip residual artifact removal;
EEG.preprocessed.residualArtifactRemoval.MARA.check = ‘yes’ : use ICA-based MARA method to remove residual artifacts; more details can be seen in I. Winkler, S. Haufe, and M. Tangermann, Automatic classification of artifactual ICA-components for artifact removal in EEG signals, Behavioral and Brain Functions, 7, 2011.
EEG.preprocessed.residualArtifactRemoval.MARA.ICs: number of ICA components to compute (default is 'pca' arg);
EEG.preprocessed.residualArtifactRemoval.MARA.ICANtrain: perform tanh() "extended-ICA" with sign estimation N training blocks; default is 0;
EEG.preprocessed.residualArtifactRemoval.MARA.ICAstop: ICA stop training when weight-change < this;
EEG.preprocessed.residualArtifactRemoval.MARA.ICAMaxSteps: max number of ICA training steps;
EEG.preprocessed.residualArtifactRemoval.MARA.ICAsphering: [‘on’/‘off’] flag sphering of data; default is ‘on’;
EEG.preprocessed.residualArtifactRemoval.MARA.artcomps: list of artifacted ICs detected by MARA;
EEG.preprocessed.residualArtifactRemoval.MARA.MARAinfo: structure containing more information about MARA classification;
MARAinfo.posterior_artefactprob: posterior probability for each IC of being an artefact.
MARAinfo.normfeats: <6 x nIC > features computed by MARA for each IC, normalized by the training data. The features are: (1) Current Density Norm, (2) Range in Pattern, (3) Local Skewness of the Time Series, (4) Lambda, (5) 8-13 Hz, (6) FitError.
EEG.preprocessed.residualArtifactRemoval.MARA.MARA_thre: cuttoff posterior probability for each IC of being an artefact while using MARA method;

EEG.preprocessed.residualArtifactRemoval.ADJUST.check = ‘yes’: use ICA-based ADJUST method to remove residual artifacts; More details about ADJUST can be seen in Mognon A, Jovicich J, Bruzzone L, Buiatti M, ADJUST: An Automatic EEG artifact Detector based on the Joint Use of Spatial and Temporal features. Psychophysiology 48 (2), 229-240 (2011).
EEG.preprocessed.residualArtifactRemoval.ADJUST.ICs: number of ICA components to compute (default is 'pca' arg);
EEG.preprocessed.residualArtifactRemoval.ADJUST.ICANtrain: perform tanh()

"extended-ICA" with sign estimation N training blocks; default is 0;
EEG.preprocessed.residualArtifactRemoval.ADJUST.ICAstop: ICA stop training when weight-change < this;
EEG.preprocessed.residualArtifactRemoval.ADJUST.ICAMaxSteps: max number of ICA training steps;
EEG.preprocessed.residualArtifactRemoval.ADJUST.ICAsphering: ['on'/'off'] flag sphering of data; default is 'on';
EEG.preprocessed.residualArtifactRemoval.ADJUST.artcomps: list of artifacted ICs;
EEG.preprocessed.residualArtifactRemoval.ADJUST.horizcomps: list of horizontal eye movement (HEM) ICs;
EEG.preprocessed.residualArtifactRemoval.ADJUST.vertcomps: list of vertical eye movement (VEM) ICs;
EEG.preprocessed.residualArtifactRemoval.ADJUST.blinkcomps: list of eye blink (EB) ICs;
EEG.preprocessed.residualArtifactRemoval.ADJUST.disccomps: list of GD ICs;
EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_DV: SVD threshold;
EEG.preprocessed.residualArtifactRemoval.ADJUST.diff_var: SVD feature values;
EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_K: TK threshold;
EEG.preprocessed.residualArtifactRemoval.ADJUST.meanK: TK feature values;
EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_SED: SED threshold;
EEG.preprocessed.residualArtifactRemoval.ADJUST.SED: SED feature values;
EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_SAD: SAD threshold;
EEG.preprocessed.residualArtifactRemoval.ADJUST.SAD: SAD feature values;
EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_GDSF: GDSF threshold;
EEG.preprocessed.residualArtifactRemoval.ADJUST.GDSF: GDSF feature values;
EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_V: MEV threshold;
EEG.preprocessed.residualArtifactRemoval.ADJUST.nuovaV: MEV feature values;

EEG.preprocessed.residualArtifactRemoval.rPCA.check = 'yes': use robust PCA to remove residual artifacts;
EEG.preprocessed.residualArtifactRemoval.rPCA.lambda: weight on sparse error term in the cost function;
EEG.preprocessed.residualArtifactRemoval.rPCA.tol: tolerance for stopping criterion;
EEG.preprocessed.residualArtifactRemoval.rPCA.maxIter: maximum number of iterations;

EEG.preprocessed.residualArtifactRemoval.ASR.check = 'yes': use ASR method to remove residual artifacts; more details about ASR can be seen in the tool 'clean_rawdata';
EEG.preprocessed.residualArtifactRemoval.ASR.burst_crit: standard deviation cutoff for removal of bursts (via ASR).A quite conservative value is 5;
EEG.preprocessed.residualArtifactRemoval.ASR.burst_crit_refmaxbadchns: this

number is the maximum tolerated (0.05-0.3) fraction of "bad" channels within a given time window of the recording that is considered acceptable for use as calibration data;

EEG.preprocessed.residualArtifactRemoval.ASR.burst_crit_reftolerances: these are the power tolerances outside of which a channel in a given time window is considered "bad", in standard deviations relative to a robust EEG power distribution (lower and upper bound). Together with the previous parameter this determines how ASR calibration data is be extracted from a recording. Can also be specified as 'off' to achieve the same effect as in the previous parameter. Default is [-3.5,5.5];

EEG.preprocessed.Interpolation.check = ‘no’: skip bad channel interpolation and re-referencing to REST only (if have channel locations);

EEG.preprocessed.Interpolation.comments = ‘re-referencing to REST based on 3-concentric spheres head model’;

EEG.preprocessed.RESITinterpolation.check = ‘yes’: use RESIT method to reconstruct bad channels;

EEG.preprocessed.RESITinterpolation.badchanns: list of bad channels;

EEG.preprocessed.RESITinterpolation.comments = ‘REST based on 3-concentric spheres head model’;

EEG.preprocessed.SphericalSplinesInterpolation.comments = ‘re-referencing to REST based on 3-concentric spheres head model’;

EEG.preprocessed.SphericalSplinesInterpolation.check = ‘yes’: use SSI method to reconstruct bad channels;

EEG.preprocessed.SphericalSplinesInterpolation.badchanns: list of bad channels;

EEG.preprocessed.QA.check = ‘yes’: QA after artifact removal;

EEG.preprocessed.QA.comments = ‘QA after artifact removal’;

EEG.preprocessed.QA.results: results of QA for artifact removed data;

EEG.preprocessed.MarkBadBlock.check = ‘yes’ or ‘no’: marking or no marking residual bad block with unusually high or low amplitude using zscored standard deviation;

EEG.preprocessed.MarkBadBlock.comments = ‘Marking residual bad block after artifact removal’;

EEG.preprocessed.MarkBadBlock.zscoredGFP: global field power of z-scored standard deviation across channels;

EEG.preprocessed.MarkBadBlock.STDthreshold: standard deviation threshold; it is equal to the Z-score cutoff for robust channel deviation in QA.

Links:
Some EEG preprocessing tools:
https://sccn.ucsd.edu/wiki/Artifact_Subspace_Reconstruction_(ASR)
https://www.nitrc.org/projects/adjust/
https://github.com/methlabUZH/automagic
https://github.com/germangh/eeglab_plugin_aar
https://github.com/VisLab/EEG-Clean-Tools
https://github.com/bwrc/ctap

3.6 WB_EEG_prepro_cm

WB_EEG_prepro_cm is a specific and stable tool to perform standardized preprocessing of continuous EEG raw data to remove a kind of artifacts (e.g. resting state EEG data), and obtain clean EEG data with REST reference. It is supported for the EEG raw data with a specific non-unipolar recording montage, such as the ipsilateral mastoid (IM) or the contralateral mastoid (CM). Preprocessing EEG raw data consists of (Fig. 4):

[1] Quality assessment of EEG raw data first. Noting that quality assessment (QA) do NOT change the EEG raw data. If the overall data quality (ODQ) exceed a threshold (default is 80), then the preprocessing could be continue;
[2] Passband and notch filtering, if applicable;
[3] Artifact removal: EOG regression;
[4] Artifact removal: residual artifact removal;
[5] Bad channel interpolation and re-referencing to REST;
[6] Quality assessment of preprocessed EEG data after artifact removal;
[7] Marking residual bad block with unusually high or low amplitude (>6) using z-scored STD across channels, and then clean EEG data with REST reference are obtained finally.

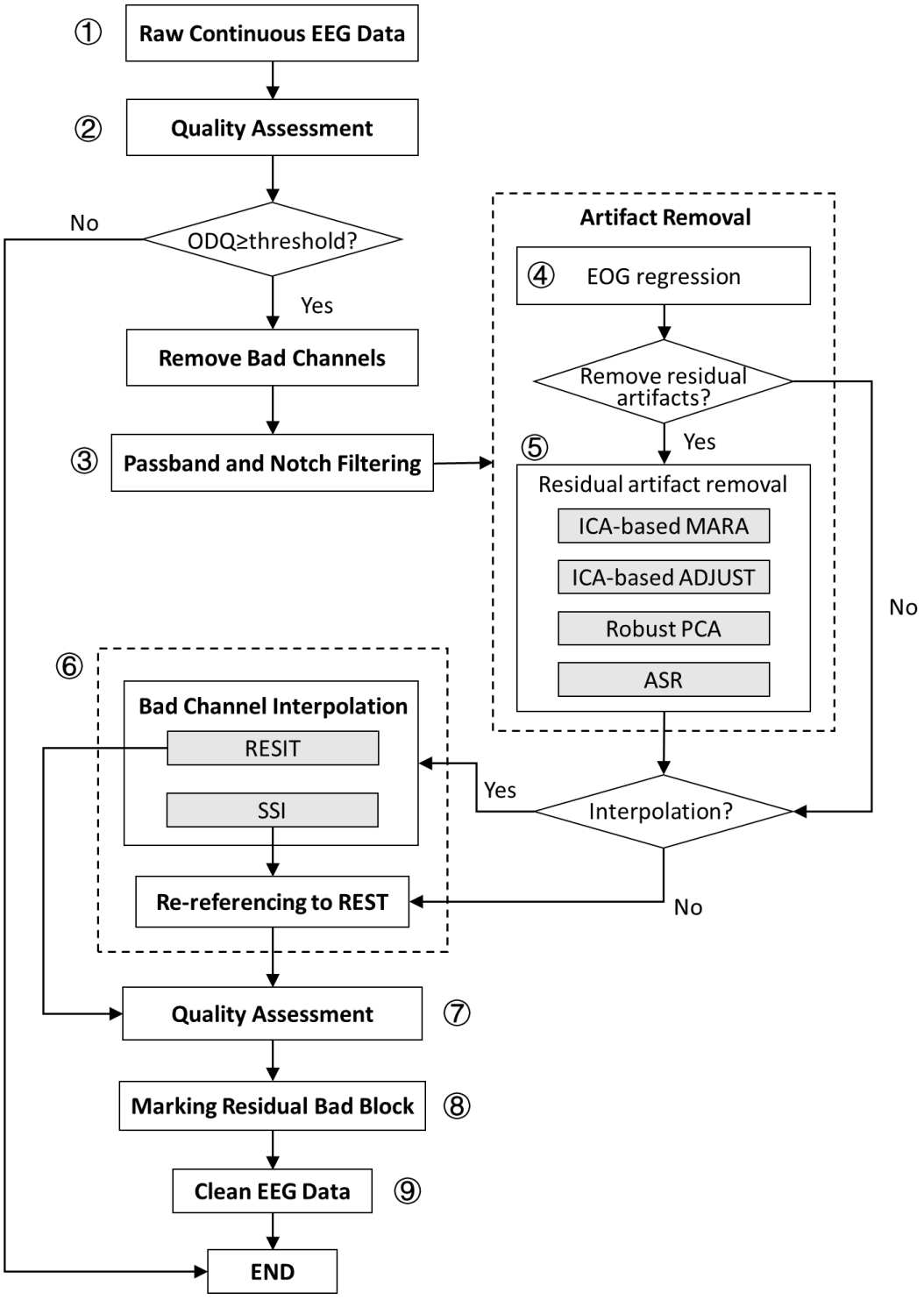

Fig. 4: Pipeline of standardized preprocessing of continuous EEG raw data. (1) Raw EEG data with artifacts such as eye blink, eye movement etc. (2) Quality assessment of EEG raw data, automatically. If the overall data quality (ODQ) exceed a threshold (ranging from 0-100, default is 80), then the preprocessing could be continue. The EEG raw data will be assessed by kinds of methods, and bad channels will be identified at same time. More details can be seen in WB_EEG_QA. (3) Passband and notch filtering (if applicable). The data can be filtered first using Hamming windowed

sinc FIR filter. (4) Artifact removal: EOG regression. A linear regression model will be utilized to remove EOG artifacts using z-scored EOG channels (5) Artifact removal: residual artifact removal. The residual artifacts will be removed using common used methods. Currently, a kind of 4 common used methods are provided in the WeBrain, including ICA based Multiple Artifact Rejection Algorithm (MARA), ICA based ADJUST, robust PCA and artifact subspace reconstruction (ASR) methods. (6) Bad channel interpolation and re-referencing to REST. For interpolation, reference electrode standardization interpolation technique (RESIT) and spherical spline interpolation (SSI) are provided in the WeBrain. (7) Quality assessment of preprocessed EEG data after artifact removal. (8) Marking residual bad block with unusually high or low amplitude using z-scored standard deviation (STD>6) across channels. Bad blocks will be marked as label 9999 using WB_EEG_Mark. (9) Clean EEG data with REST reference are obtained finally.

Parameters

leftchanns: index of left channels excluding left Reference (e.g. left mastoid). e.g. '[1:2:16]';

rightchanns: index of right channels excluding right Reference (e.g. right mastoid). e.g. '[2:2:16]';

xyz_leftRef: coordinates of reference for left channels, e.g. '[-0.309,0.9511,0]'. The row is xyz coordinates of reference for left channels.

xyz_rightRef: coordinates of reference for right channels, e.g. '[-0.309,-0.9511,0]'. The row is xyz coordinates of reference for right channels.

EOGchanns: number with indices of the EOG channels. Default is '[]'.

thre_ODQ: threshold of overall data quality (ODQ). If ODQ ≥ a threshold, then the preprocessing could be continue. Default is 80. Noting that the quality assessment (QA) would NOT change the EEG raw data, and some default QA parameters are fixed in the preprocessing tool (window size is 1 sec, lower edge of the frequency for high pass filtering is 1 Hz, cutoff fraction of bad windows is 0.4, Z-score cutoff for robust channel deviation is 5, Z-score cutoff for noise-to-signal rate is 3 and correlation threshold is 0.6).

passband: passband of filtering. Default is '[1,40]'.

PowerFrequency: power frequency. Default is 50 Hz (in Chinese). Noting that in USA, power frequency is 60Hz.

keepUnselectChannsFlag:

keepUnselectChannsFlag = 0: do not keep unselected channels (default); keepUnselectChannsFlag = 1: keep all channels.

badChannelInterploateFlag:

badChannelInterploateFlag = 0: do NOT interpolate, and if have channel locations in EEG.chanlocs, then re-referencing to REST (Dong et al., 2017; Yao, 2001);

badChannelInterploateFlag = 1 (default): interpolate the bad channels rows of

EEG.data using reference electrode standardization interpolation technique (RESIT); default is using RESIT (The bad channels will be interpolated with REST reference (Dong et al., 2017; Dong et al., 2021; Yao, 2001));

badChannelInterploateFlag = 2: interpolate the bad channels rows of EEG.data using spherical spline interpolation (SSI) (Perrin et al., 1989), and re-referencing to REST.

residualArtifactRemovalFlag:

residualArtifactRemovalFlag = 0: no removal;

residualArtifactRemovalFlag = 1: ICA based MARA (Multiple Artifact Rejection Algorithm) (Winkler et al., 2011);

residualArtifactRemovalFlag = 2: ICA based ADJUST (Mognon et al., 2011);

residualArtifactRemovalFlag = 3: rPCA method (Lin et al., 2010);

residualArtifactRemovalFlag = 4: ASR method (Mullen et al., 2013).

MARA_thre: cuttoff posterior probability for each IC of being an artefact while using MARA method. Default is 0.7.

badWindowThreshold: cutoff fraction of bad windows (default = 0.6) for detecting bad channels.

srate: sampling rate of EEG data. It can be automatically detected in EEG data. But for ASCII/Float .txt File or MATLAB .mat File, user should fill the sampling rate by hand. Default is '[]'.

Note:

- Noting that quality assessing EEG raw data would NOT change the raw data. If the overall data quality (ODQ) exceed a threshold (default is 80), then the preprocessing could be continue.
- EEG data will be imported as EEG structure of EEGLAB. Dimension of EEG.data must be channels × time points.
- If channel locations are not contained in EEG data or selected channels do not contain locations, methods required EEG channel coordinates are invalid (e.g. interpolation, ICA-based MARA, ICA-based ADJUST, and REST re-referencing etc.).
- It is supported for the EEG raw data with a specific non-unipolar recording montage, such as the ipsilateral mastoid (IM) or the contralateral mastoid (CM) ONLY.

Outputs

For each subject, a zip file which contains the preprocessed EEG data will be generated (saved as *_prepro.set file which contains the clean EEG data (EEG.data with dimension channels × time points) and preprocessing info (parameters and results of each preprocessing step). The file can be further analyzed by WeBrain online or EEGLAB offline. Following fields will be further added in the *.set file.

EEG.preprocessed.PassbandFilter.check = ‘yes’ or ‘no’ for pass band filtering;
EEG.preprocessed.PassbandFilter.passband: pass band;
EEG.preprocessed.PassbandFilter.comments = ‘Hamming windowed sinc FIR filter’;
EEG.preprocessed.NotchFilter.check = ‘yes’ or ‘no’ for notch filtering;
EEG.preprocessed.NotchFilter.notchband: notch filtering band;
EEG.preprocessed.EOGregression.check = ‘yes’ or ‘no’ for EOG regression
EEG.preprocessed.EOGregression.EOGchanns = EOG channels;

EEG.preprocessed.residualArtifactRemoval.check = ‘no’: skip residual artifact removal;
EEG.preprocessed.residualArtifactRemoval.MARA.check = ‘yes’ : use ICA-based MARA method to remove residual artifacts; more details can be seen in I. Winkler, S. Haufe, and M. Tangermann, Automatic classification of artifactual ICA-components for artifact removal in EEG signals, Behavioral and Brain Functions, 7, 2011.
EEG.preprocessed.residualArtifactRemoval.MARA.ICs: number of ICA components to compute (default is 'pca' arg);
EEG.preprocessed.residualArtifactRemoval.MARA.ICANtrain: perform tanh() "extended-ICA" with sign estimation N training blocks; default is 0;
EEG.preprocessed.residualArtifactRemoval.MARA.ICAstop: ICA stop training when weight-change < this;
EEG.preprocessed.residualArtifactRemoval.MARA.ICAMaxSteps: max number of ICA training steps;
EEG.preprocessed.residualArtifactRemoval.MARA.ICAsphering: [‘on’/‘off’] flag sphering of data; default is ‘on’;
EEG.preprocessed.residualArtifactRemoval.MARA.artcomps: list of artifacted ICs detected by MARA;
EEG.preprocessed.residualArtifactRemoval.MARA.MARAinfo: structure containing more information about MARA classification;
MARAinfo.posterior_artefactprob: posterior probability for each IC of being an artefact.
MARAinfo.normfeats: <6 x nIC > features computed by MARA for each IC, normalized by the training data. The features are: (1) Current Density Norm, (2) Range in Pattern, (3) Local Skewness of the Time Series, (4) Lambda, (5) 8-13 Hz, (6) FitError.
EEG.preprocessed.residualArtifactRemoval.MARA.MARA_thre: cuttoff posterior probability for each IC of being an artefact while using MARA method;

EEG.preprocessed.residualArtifactRemoval.ADJUST.check = ‘yes’: use ICA-based ADJUST method to remove residual artifacts; More details about ADJUST can be seen in Mognon A, Jovicich J, Bruzzone L, Buiatti M, ADJUST: An

Automatic EEG artifact Detector based on the Joint Use of Spatial and Temporal features. Psychophysiology 48 (2), 229-240 (2011).

EEG.preprocessed.residualArtifactRemoval.ADJUST.ICs: number of ICA components to compute (default is 'pca' arg);

EEG.preprocessed.residualArtifactRemoval.ADJUST.ICANtrain: perform tanh() "extended-ICA" with sign estimation N training blocks; default is 0;

EEG.preprocessed.residualArtifactRemoval.ADJUST.ICAstop: ICA stop training when weight-change < this;

EEG.preprocessed.residualArtifactRemoval.ADJUST.ICAMaxSteps: max number of ICA training steps;

EEG.preprocessed.residualArtifactRemoval.ADJUST.ICAsphering: ['on'/'off'] flag sphering of data; default is ‘on’;

EEG.preprocessed.residualArtifactRemoval.ADJUST.artcomps: list of artifacted ICs;

EEG.preprocessed.residualArtifactRemoval.ADJUST.horizcomps: list of horizontal eye movement (HEM) ICs;

EEG.preprocessed.residualArtifactRemoval.ADJUST.vertcomps: list of vertical eye movement (VEM) ICs;

EEG.preprocessed.residualArtifactRemoval.ADJUST.blinkcomps: list of eye blink (EB) ICs;

EEG.preprocessed.residualArtifactRemoval.ADJUST.disccomps: list of GD ICs;

EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_DV: SVD threshold;

EEG.preprocessed.residualArtifactRemoval.ADJUST.diff_var: SVD feature values;

EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_K: TK threshold;

EEG.preprocessed.residualArtifactRemoval.ADJUST.meanK: TK feature values;

EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_SED: SED threshold;

EEG.preprocessed.residualArtifactRemoval.ADJUST.SED: SED feature values;

EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_SAD: SAD threshold;

EEG.preprocessed.residualArtifactRemoval.ADJUST.SAD: SAD feature values;

EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_GDSF: GDSF threshold;

EEG.preprocessed.residualArtifactRemoval.ADJUST.GDSF: GDSF feature values;

EEG.preprocessed.residualArtifactRemoval.ADJUST.soglia_V: MEV threshold;

EEG.preprocessed.residualArtifactRemoval.ADJUST.nuovaV: MEV feature values;

EEG.preprocessed.residualArtifactRemoval.rPCA.check = ‘yes’: use robust PCA to remove residual artifacts;

EEG.preprocessed.residualArtifactRemoval.rPCA.lambda: weight on sparse error term in the cost function;

EEG.preprocessed.residualArtifactRemoval.rPCA.tol: tolerance for stopping criterion;

EEG.preprocessed.residualArtifactRemoval.rPCA.maxIter: maximum number of iterations;

EEG.preprocessed.residualArtifactRemoval.ASR.check = ‘yes’: use ASR method to

remove residual artifacts; more details about ASR can be seen in the tool ‘clean_rawdata’;

EEG.preprocessed.residualArtifactRemoval.ASR.burst_crit: standard deviation cutoff for removal of bursts (via ASR).A quite conservative value is 5;

EEG.preprocessed.residualArtifactRemoval.ASR.burst_crit_refmaxbadchns: this number is the maximum tolerated (0.05-0.3) fraction of "bad" channels within a given time window of the recording that is considered acceptable for use as calibration data;

EEG.preprocessed.residualArtifactRemoval.ASR.burst_crit_reftolerances: these are the power tolerances outside of which a channel in a given time window is considered "bad", in standard deviations relative to a robust EEG power distribution (lower and upper bound). Together with the previous parameter this determines how ASR calibration data is be extracted from a recording. Can also be specified as 'off' to achieve the same effect as in the previous parameter. Default is [-3.5,5.5];

EEG.preprocessed.Interpolation.check = ‘no’: skip bad channel interpolation and re-referencing to REST only (if have channel locations);

EEG.preprocessed.Interpolation.comments = ‘re-referencing to REST based on 3-concentric spheres head model’;

EEG.preprocessed.RESITinterpolation.check = ‘yes’: use RESIT method to reconstruct bad channels;

EEG.preprocessed.RESITinterpolation.badchanns: list of bad channels;

EEG.preprocessed.RESITinterpolation.comments = ‘REST based on 3-concentric spheres head model’;

EEG.preprocessed.SphericalSplinesInterpolation.comments = ‘re-referencing to REST based on 3-concentric spheres head model’;

EEG.preprocessed.SphericalSplinesInterpolation.check = ‘yes’: use SSI method to reconstruct bad channels;

EEG.preprocessed.SphericalSplinesInterpolation.badchanns: list of bad channels;

EEG.preprocessed.QA.check = ‘yes’: QA after artifact removal;

EEG.preprocessed.QA.comments = ‘QA after artifact removal’;

EEG.preprocessed.QA.results: results of QA for artifact removed data;

EEG.preprocessed.MarkBadBlock.check = ‘yes’ or ‘no’: marking or no marking residual bad block with unusually high or low amplitude using zscored standard deviation;

EEG.preprocessed.MarkBadBlock.comments = ‘Marking residual bad block after artifact removal’;

EEG.preprocessed.MarkBadBlock.zscoredGFP: global field power of z-scored standard deviation across channels;

EEG.preprocessed.MarkBadBlock.STDthreshold: standard deviation threshold; it is equal to the Z-score cutoff for robust channel deviation in QA.

Links:

Some EEG preprocessing tools:

https://sccn.ucsd.edu/wiki/Artifact_Subspace_Reconstruction_(ASR)

https://www.nitrc.org/projects/adjust/

https://github.com/methlabUZH/automagic

https://github.com/germangh/eeglab_plugin_aar

https://github.com/VisLab/EEG-Clean-Tools

https://github.com/bwrc/ctap

3.7 WB_EEG_CalcPower

WB_EEG_CalcPower is a tool to calculate power indices using time-frequency analysis of EEGALB (using function timefreq()). Calculating power indices consists of (Fig. 5):

[1] Specific event data can be extracted according to the input 'eventlabel'. If the input 'eventlabel' is empty, all data will be used. If applicable, EEG segments in bad block (label 9999, marked by the tool WB_EEG_Mark) will also be rejected automatically, and NOT used to calculate power indices.

[2] Specific event EEG signals will be divided into small epochs.

[3] EEG data of each epoch (default is 5s epoch) was subjected to time-frequency analysis with Fast-Fourier Transform (FFT) to obtain the absolute EEG band power at each electrode in the specific bands. Each data epoch will be linearly detrended before time-frequency analysis. The power value is calculated by:

$$Y_{power} = 10 * log10(\frac{||\frac{2}{0.375} * Y||^2}{window\ size})$$

where Y is complex number calculated by FFT, $\|.\|$ is complex modulus operations (using 'abs' function of MATLAB), the unit is $10 \cdot \log10\mu V^2/Hz$, the window size is calculated by $\max(\text{pow2}(\text{nextpow2}(\text{length}(\text{epoch})) - 3),4)$. Noting that, multiply by 2 account for negative frequencies, and counteract the reduction by a factor 0.375 that occurs as a result of cosine (Hann) tapering.

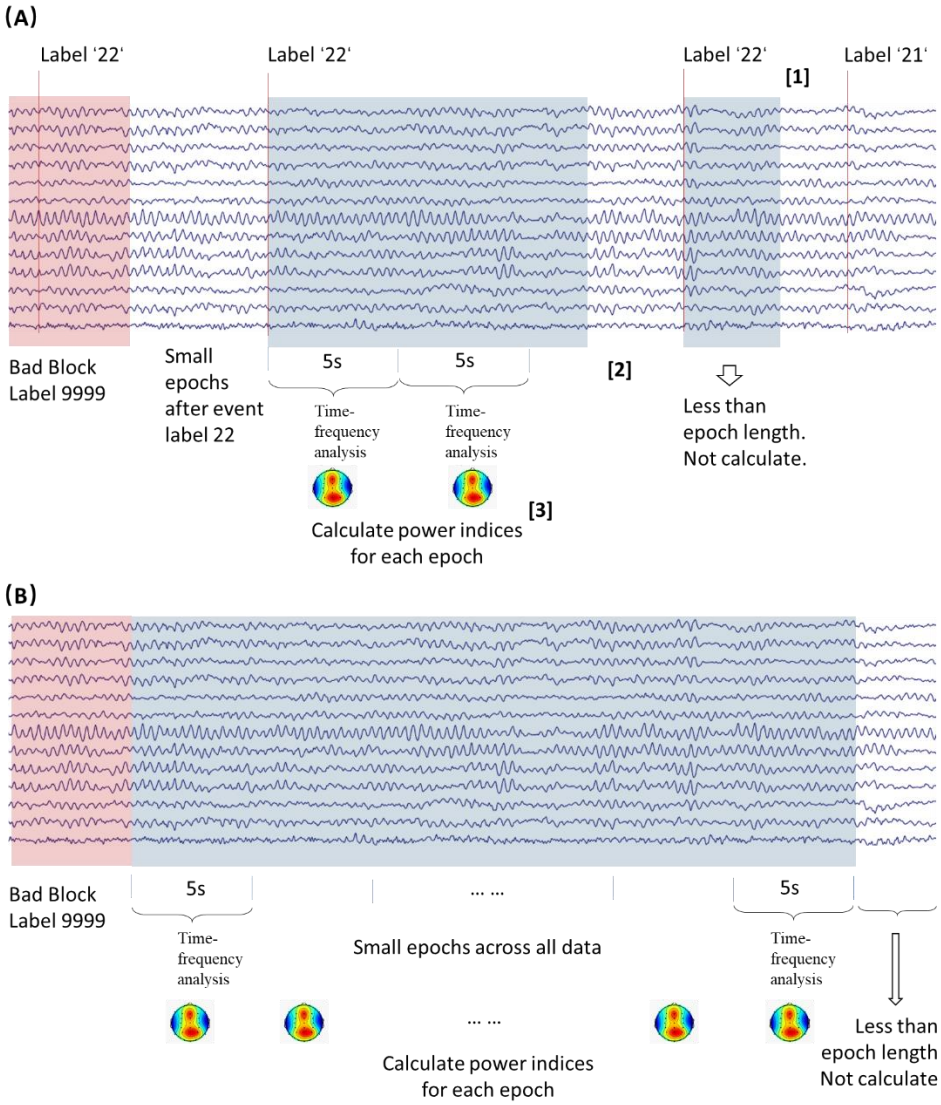

Fig. 5: Pipelines of calculating power indices of EEG data. (A) Calculating power indices of a specific event data. Step1: Specific event data are first extracted according to the input 'eventlabel' (e.g. label '22'). If applicable, EEG segments in bad block (label 9999, marked by WB_EEG_Mark) will also be rejected automatically, and NOT used to calculate power indices. Step2: Specific event EEG signals will be divided into small epochs. Step3: EEG data of each small epoch (default is 5s epoch) will be subjected to time-frequency analysis with Fast-Fourier Transform (FFT) to obtain the absolute EEG band power at each electrode in the specific bands. (B) Calculating power indices of all data. All data will be divided into small epochs first, and then data of each epoch (excluding bad blocks) will be used to calculate power indices.

Default frequency bands are delta (Nuwer et al., 1994), theta (Nuwer et al., 1994), alpha1 (Malver et al., 2014), alpha2 (Malver et al., 2014), beta1 (Jobert et al., 2013; Malver et al., 2014), beta2 (Jobert et al., 2013; Malver et al., 2014), beta3 (Jobert et al., 2013; Malver et al., 2014), gamma1 (Jobert et al., 2013; Nuwer et al., 1994) and

gamma2 (Nuwer et al., 1994). Indices of power ratio (Kashefpoor et al., 2016; Snaedal et al., 2010; Thatcher et al., 2005) are r1, r2, r3, r4, r5 and r6, as well as peak of alpha frequency (PAF).

Default list of 27 indices:

delta: mean power across 1 - 4 Hz

theta: mean power across 4 - 8 Hz

alpha1: mean power across 8 - 10.5 Hz

alpha2: mean power across 10.5 - 12.5 Hz

beta1: mean power across 12.5 - 18.5 Hz

beta2: mean power across 18.5 - 21 Hz

beta3: mean power across 21 - 30 Hz

gamma1: mean power across 30 - 40 Hz

gamma2: mean power across 40 - 60 Hz

fullband: mean power across 1-60 Hz

relative power in specific band = power of specific band/total power across fullband.

r1 = theta / (alpha1 + alpha2 + beta1);

r2 = (delta + theta) / (alpha1 + alpha2 + beta1 + beta2);

r3 = theta/alpha = theta / (alpha1+alpha2);

r4 = theta/beta = theta / (beta1 + beta2 + beta3);

r5 = delta/theta;

r6 = alpha/beta = (alpha1 + alpha2) / (beta1 + beta2 + beta3);

PAF (peak of alpha frequency) = max power in alpha (alpha1+alpha2) band.

More details about power indices of EEG data can be seen in relative papers (Chen et al., 2008; Kashefpoor et al., 2016; Malver et al., 2014; Nuwer et al., 1994; Snaedal et al., 2010; Thatcher et al., 2005).

Parameters

epochLenth: Length of small epochs to calculate power. Unit is second. Default is 5s. Epochs less than epochLenth will be not used to calculate indices. If epochLenth is negative (Fig. 6), it means that if possible, data before event labels (eventlabel) will be used (no overlapped).

eventlabel: Event label which means specific event data. If it is empty, all data will be used to calculate indices. If eventlabel is not found in events, NO data will be epoched and calculated. If structure event (eventlabel) doesn't include duration, the duration will be equal to epochLenth. DO NOT contain blank spaces in the event label (e.g. S 22). Strings of events will be compared ignoring space characters. Default is empty ('[]'). You can also input multiple event labels at once, split by comma (e.g. 'S22,S23'); and all these event data will be calculated.

bandLimit: A string array with specific frequency bands. Default is '[1,4],[4,8],[8,10.5],[10.5,12.5],[12.5,18.5],[18.5,21],[21,30],[30,40],[40,60],[

1,60]'. If it is '[]', default bands will be used. Bands should be separated by comma.

bandName: A string array with names of frequency bands (separated by comma). bandName must be corresponding to bandLimit (e.g. bandName 1 is corresponding to bandLimit 1). Names of frequency bands are required and included in ('delta';'theta';'alpha1';'alpha2';'beta1';'beta2';'beta3') for r1-r6 and PAF, or included in ('delta';'theta';'alpha';'beta') for r3-r6 and PAF. For relative power indices, 'fullband' must be included in bandName ('fullband'). Default is 'delta,theta,alpha1,alpha2,beta1,beta2,beta3,gamma1,gamma2,fullband'. If it is '[]', default band names will be used.

seleChanns: String with indices of the selected channels (e.g. '[1:4,7:30]'), or 'all'. Default is 'all'.

proportion: overlapped proportion for each segments/sliding windows. It should be [0,1). Default is 0 (no overlapped).

srate: Sampling rate of EEG data. It can be automatically detected in EEG data, if it is '[]'. But for ASCII/Float .txt File and MATLAB .mat File, users should fill the sampling rate by hand. Default is '[]'.

grouplabel: Group label of subject. It may be used to statistical analysis in the future. Default is '[]'.

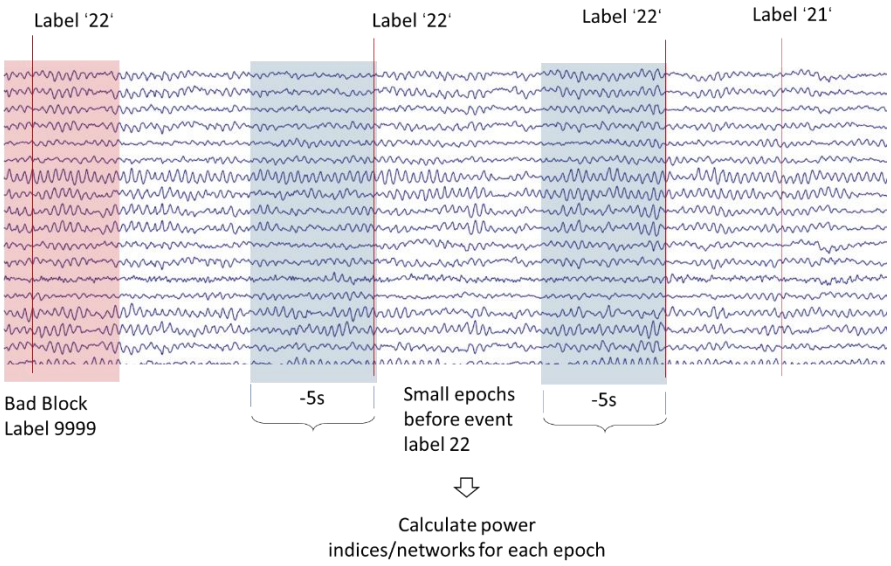

Fig. 6: If epochLenth is negative, it means that if applicable, data before event labels (eventlabel) will be used (no overlapped).

Note: EEG data will be imported as EEG structure using EEGLAB. EEG.data should be channels × time points OR channels × time points × epochs. If size of EEG.data is channels × time points × epochs (has been epoched), all data will be used to calculate power indices. If applicable, epochs in bad block

(label 9999, marked by wb_EEG_Mark) will also be rejected automatically, and NOT used to calculate power indices.

Outputs

For each subject, output is a MATLAB .mat file (named power_*) in which is a structure EEG_results including power indices, mean indices across epochs and parameters.

EEG_results.type: type of results. i.e. 'power'

EEG_results.filename: filename of EEG data;

EEG_results.Power: power across frequency bands (channels×bands×epochs); the power value is transformed by natural logarithm, i.e. $Y_{power} = \ln(Y_{power})$.

EEG_results.Power_relative: relative power across frequency bands (channels×bands×epochs);

EEG_results.PAF : max power in alpha(alpha1+alpha2) band (channels×epochs);

EEG_results.R1 : r1 (channels×epochs);

EEG_results.R2 : r2 (channels×epochs);

EEG_results.R3 : r3 (channels×epochs);

EEG_results.R4 : r4 (channels×epochs);

EEG_results.R5 : r5 (channels×epochs);

EEG_results.R6 : r6 (channels×epochs);

EEG_results.Block_percentage: percentage of EEG data used to calculate indices

EEG_results.Power_mean: mean power across epochs (channels×bands); the power value is transformed by natural logarithm, i.e. $Y_{power} = \ln(Y_{power})$.

EEG_results.Power_relative_mean : mean relative power across epochs (channels×bands);

EEG_results.R1_mean : mean r1 across epochs (channels×1);

EEG_results.R2_mean : mean r2 across epochs (channels×1);

EEG_results.R3_mean : mean r3 across epochs (channels×1);

EEG_results.R4_mean : mean r4 across epochs (channels×1);

EEG_results.R5_mean : mean r5 across epochs (channels×1);

EEG_results.R6_mean : mean r6 across epochs (channels×1);

EEG_results.PAF_mean : mean PAF across epochs (channels×1);

EEG_results.spectrum: frequency spectrum across each epochs (channels×freqs×epochs); the value is transformed by natural logarithm, i.e. $Y_{power} = \ln(Y_{power})$.

EEG_results.freqs: Frequencies of time-frequency analysis (1×freqs);

EEG_results.spectrum_mean: mean frequency spectrum across each epochs (channels×freqs); the power value is transformed by natural logarithm, i.e. $Y_{power} = \ln(Y_{power})$.

EEG_results.grouplabel: group label of subject. It may be used to statistical analysis in the future.

EEG_results.parameter.WaveletCycles: The number of cycles for the time-frequency decomposition. Default is using FFTs and Hanning window tapering;

EEG_results.parameter.WaveletMethod: Wavelet method/program to use. Default is 'dftfilt3': Morlet wavelet or hanning DFT (exact Tallon baudry);

EEG_results.parameter.TaperingFunction: FFT Tapering function is 'hanning' function;

EEG_results.parameter.DetrendStr: 'On': Linearly detrend each data epoch before time-frequency analysis.

EEG_results.parameter.bandLimit: An array with specific frequency bands;

EEG_results.parameter.bandName: A cell array with band names;

EEG_results.parameter.eventlabel: An event label which means specific data;

EEG_results.parameter.selechanns: An array with selected channels;

EEG_results.parameter.epochLenth: Length of small epochs. Unit is time point.

EEG_results.parameter.srate: Sampling rate of EEG data.

EEG_results.parameter.proportion: Overlapped proportion for each segments/sliding windows;

EEG_results.parameter.chanlocs: channel locations.

EEG_results.parameter.ref: reference of EEG data.

Links

NIT
http://www.neuro.uestc.edu.cn/NIT.html

EEGLAB
http://sccn.ucsd.edu/eeglab/index.html

3.8 WB_EEG_CalcERP

WB_EEG_CalcERP is a tool to create averaged event related potential (ERP) for each EEG channel at scalp level. Calculating ERP consists of (Fig. 8):

[1] Filter data using Hamming windowed sinc FIR filter. This step is optional, and default is no filtered (i.e. set passband as []).

[2] Extract epochs (default is [-0.2 0.8] sec) and baseline correction ([-0.2, 0] sec): A continuous EEG dataset will be converted to epoched data by extracting data epochs time locked to specified event types or event indices. If applicable, time locked events corresponding to correct-reaction marker will be extracted (i.e. marker1). In addition, events in bad block (label 9999, marked by WB_EEG_Mark) will also be rejected automatically.

[3] Artifact rejection in epoched data using simple voltage threshold. Three criterions including amplitude, gradient and max-min criterions were used to reject artifact

trials.

[4] ERP will be obtained from averaged clean epochs (default is [-0.2 0.8] sec).

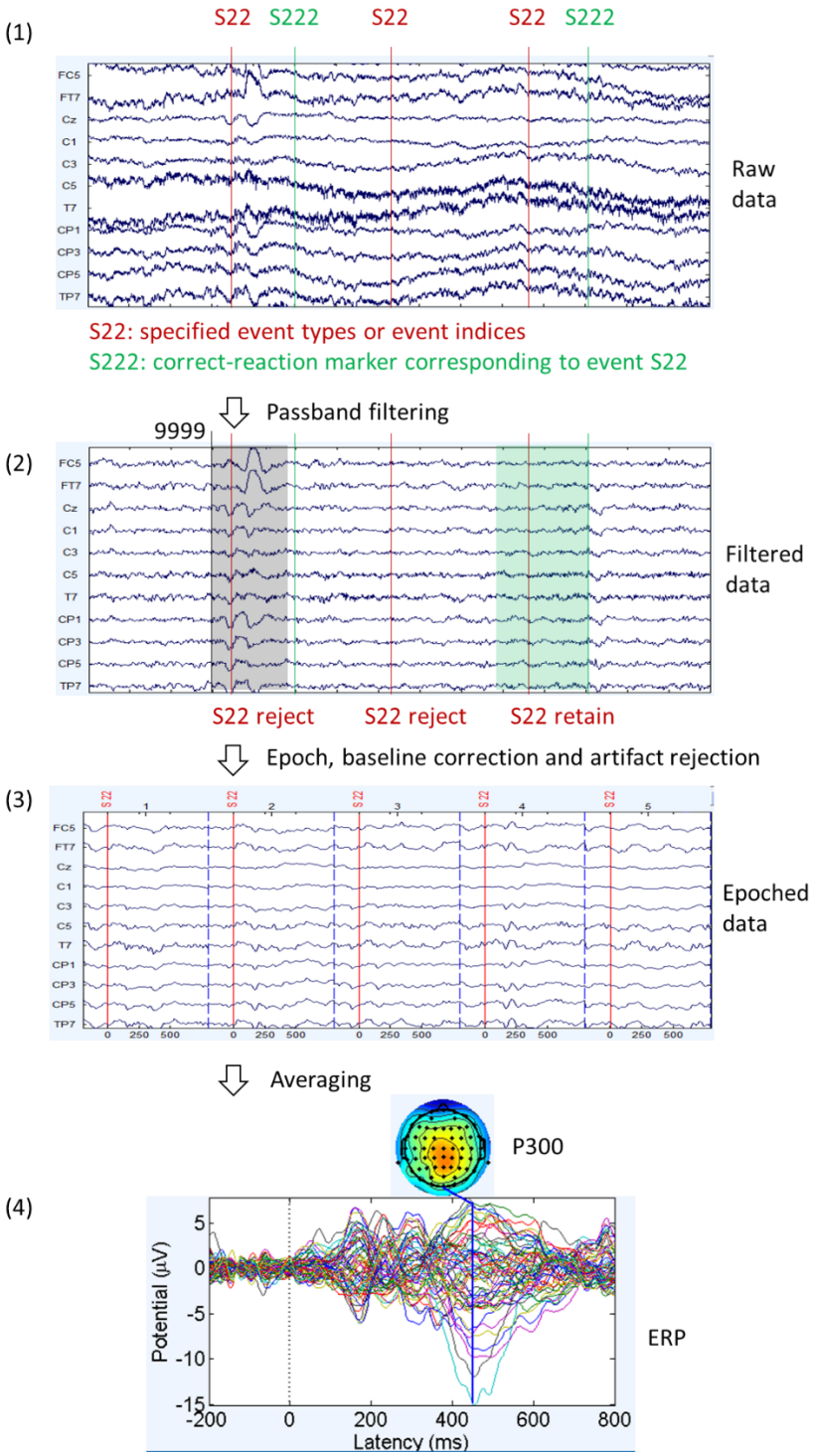

Fig. 8: Pipeline of ERP analysis. (1) Raw EEG data with events. As an example, S22 is a specified event type (e.g. target stimulus), and S222 is correct-reaction marker corresponding to S22 if applicable). (2) Filtering EEG data, if applicable. The data can be filtered first using Hamming windowed sinc FIR filter. (3) Extract epochs (e.g. [-0.2, 0] sec), baseline correction and artifact rejection. A continuous EEG dataset will

be converted to epoched data by extracting data epochs time locked to specified event types or event indices. Then, baseline correction (e.g. [-0.2, 0] sec) and artifact rejection will be conducted on epoched data. Three criterions of simple voltage threshold, including amplitude, gradient and max-min criterions, will be used to reject artifact trials. If applicable, time locked events corresponding to correct-reaction marker will be extracted only (i.e. marker1). In addition, events in bad block (label 9999, marked by WB_EEG_Mark) will also be rejected automatically. (4) Averaging. ERP will be obtained from clean epoched data.

Parameters

event1: specified event types or event indices (e.g. event label). If event label is not found, NO data will be epoched and calculated. DO NOT contain blank spaces in the event label (e.g. S 22). Strings of events will be compared ignoring space characters. Default is '[]'. The letters are case-sensitive. You can also input multiple event labels at once, split by comma (e.g. 'S22,S23'); and all these event data will be calculated.

epochlimits: epoch latency range [start, end] in seconds relative to the time-locking events. Default is '[-0.2,0.8]' sec.

valuelim_1: threshold of amplitude criterion to reject artifact trials: Lower and upper bound latencies for trial data. If one positive value is given, the opposite value is used for lower bound. For example, use [-100,100] microvolts (μV) to remove artificial epoch. Default is [-100,100] (μV).

valuelim_2: threshold of gradient criterion to reject artifact trials: maximum allowed voltage step/sampling point. Default is 50 microvolts (μV).

valuelim_3: threshold of max-min criterion to reject artifact trials: maximum allowed absolute difference in the segment/epoch. Default is 150 microvolts (μV).

selechanns: number with indices of the selected channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all'.

marker1: correct-reaction marker corresponding to the specified event (e.g. event1). DO NOT contain blank spaces in the marker1 (e.g. S 222). Default is '[]'.

t1: duration (in seconds) before correct-reaction marker. Default is 2s.

passband: passband of filtering (e.g. '[1,30]'). Default is NO filtered (i.e. '[]').

srate: sampling rate of EEG data. It can be automatically detected in EEG data. But for ASCII/Float .txt File or MATLAB .mat File, user should fill the sampling rate by hand. Default is '[]'.

Note:

- EEG data will be imported as EEG structure using EEGLAB. EEG.data should be channels × time points OR channels × time points × epochs. If size of EEG.data is channels × time points × epochs (has been epoched), all data will be used to create ERP.
- Clean trials are satisfying 3 criterion of artifact rejection for each channel at same

time.

Outputs

For each subject, a zip file which contain the ERP data will be generated (saved as ERP .set file which contains the ERP potentials, EEG.data with dimension channels × time points × trials). The ERP .set file can be imported and used in EEGLAB. Following fields will be further added in the ERP .set file.

EEG.eventlist: list of accepted events;
EEG.erp: averaged event-related potentials for each channel;
EEG.trials: No. of trials;
EEG.xmin: Epoch latency limits [start] in seconds;
EEG.xmax: Epoch latency limits [end] in seconds;
EEG.epoch: filling with values of other events in the same epochs.

3.9 WB_EEG_CalcNetwork

WB_EEG_CalcNetwork is a basic tool to calculate EEG network between EEG channels at scalp level or source level. Calculating EEG network consists of (Fig. 7):

[1] Specific event data can be extracted according to the input 'eventlabel'. If the input 'eventlabel' is empty, all data will be used. If applicable, EEG segments in bad block (label 9999, marked by WB_EEG_Mark) will be rejected automatically, and NOT used to calculate network.

[2] Specific EEG signals will be divided into small epochs.

[3] EEG data of each epoch (default is 5-s epoch) was subjected to calculate correlation/coherence/PSI/PLV to obtain the EEG network across electrodes in the specific bands

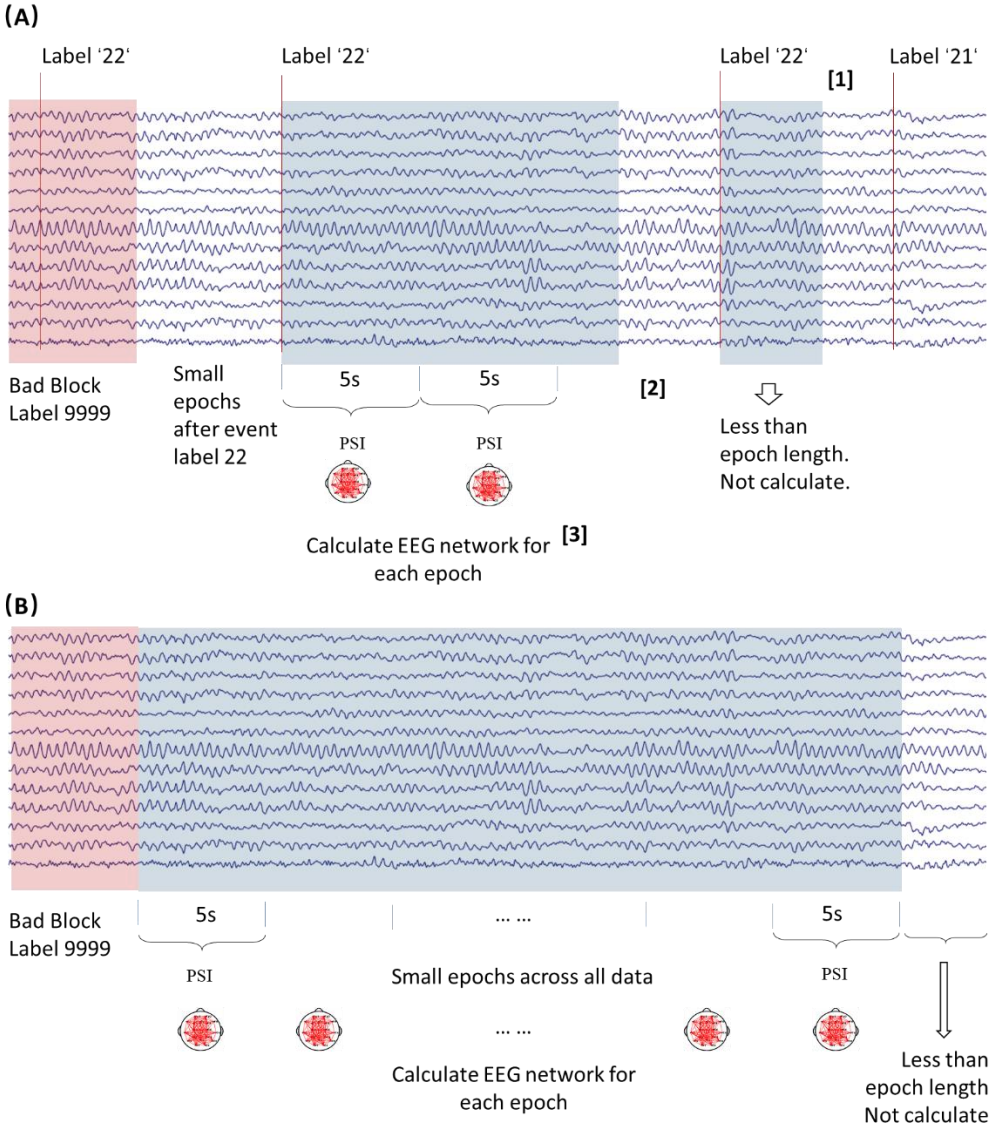

Fig. 7: Pipelines of calculating EEG networks of EEG data. (A) Calculating EEG networks of a specific event data. Step1: Specific event data are first extracted according to the input 'eventlabel' (e.g. label '22'). If applicable, EEG segments in bad block (label 9999, marked by WB_EEG_Mark) will also be rejected automatically, and NOT used to calculate coefficients. Step2: Specific event EEG signals will be divided into small epochs. Step3: EEG data of each small epoch (default is 5s epoch) will be subjected to calculate correlation/coherence/PSI/PLV to obtain the EEG network across electrodes in the specific bands. (B) Calculating EEG networks of all data. All data will be divided into small epochs first, and then data of each epoch (excluding bad blocks) will be used to calculate EEG networks.

Connections can be defined as (Bob et al., 2008; Edagawa and Kawasaki, 2017; Lee and Hsieh, 2014; Xu et al., 2014):

1. Pearson's correlation:

$$Corr = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}}$$; where C is the covariance matrix.

2. Magnitude squared coherence. The magnitude squared coherence estimate is a function of frequency with values between 0 and 1 that indicates how well x corresponds to y at each frequency. The magnitude squared coherence is a function of the power spectral densities, $P_{xx}(f)$ and $P_{yy}(f)$, of x and y, and the cross power spectral density, $P_{xy}(f)$, of x and y:

$$Cohere = \frac{|P_{xy}(f)|^2}{P_{xx}(f)P_{yy}(f)}$$

3. Phase Synchronization Index (PSI):

$$PSI = \sqrt{<\cos(\Delta\phi(t))>^2 + <\sin(\Delta\phi(t))>^2}$$

Where $\Delta\phi(t) = \phi_1(t) - \phi_2(t)$ is the instantaneous phase difference between two EEG signals for a particular frequency; $<\cdot>$ means the temporal average. PSI tends to 0 for asynchronous processes and to 1 for phase locked systems. Considering real noisy data, neither 0 nor 1 is reached, but we can expect that the PSI for synchronized oscillations will be significantly larger than for unsynchronized processes.

4. Phase Locking Value (PLV). For two EEG signals with data length L, the PLV is defined as:

$$PLV = \left| \frac{1}{L} \sum_{t=0}^{L} e^{i\phi(t)} \right|$$

The phase locking index is sensitive to phase change and its value ranges from 0 to 1. The PLV =1 if and only if the condition of strict phase locking is obeyed. In contrast, the PLV = 0 for uniformly distributed phases.

Default frequency bands are (Chen et al., 2008; Malver et al., 2014; Nuwer et al., 1994; Thatcher et al., 2005):

delta: 1 - 4 Hz
theta: 4 - 8 Hz
alpha: 8 - 12.5 Hz
beta: 12.5 - 25 Hz
high beta: 25 - 30 Hz
gamma1: mean power across 30 - 40 Hz
gamma2: mean power across 40 - 60 Hz
fullband: 1 - 60 Hz

Parameters

epochLenth: Length of small epochs to calculate power. Unit is second. Default is 5s. Epochs less than epochLenth will be not used to calculate indices. If epochLenth is negative (Fig. 2), it means that if possible, data before event labels (eventlabel) will be used (no overlapped).

eventlabel: Event label which means specific event data. If it is empty, all data will be used to calculate indices. If eventlabel is not found in events, NO data will be epoched and calculated. If structure event (eventlabel) doesn't include duration, the duration will be equal to epochLenth. DO NOT contain blank spaces in the event label (e.g. S 22). Strings of events will be compared ignoring space characters. Default is empty ('[]'). You can also input multiple event labels at once, split by comma (e.g. 'S22,S23'); and all these event data will be calculated.

bandLimit: A string array with specific frequency bands. Default is '[1,4],[4,8],[8,12.5],[12.5,25],[25,30],[30,40],[40,60],[1,60]'. If it is '[]', full band will be used. Bands should be separated by comma.

seleChanns: String with indices of the selected channels (e.g. '[1:4,7:30]'), or 'all'. Default is 'all'.

method: Method used to calculate EEG network. Default is 'psi'.

'corr': Pearson's correlation

'cohere': Magnitude squared coherence

'psi': Phase Synchronization Index

'plv': Phase Locking Value

proportion: overlapped proportion for each segments/sliding windows. It should be [0,1). Default is 0 (no overlapped).

srate: Sampling rate of EEG data. It can be automatically detected in EEG data, if it is '[]'. But for ASCII/Float .txt File and MATLAB .mat File, users should fill the sampling rate by hand. Default is '[]'.

grouplabel: Group label of subject. It will be used to statistical analysis in the future. Default is '[]'.

Note: EEG data will be imported as EEG structure using EEGLAB. EEG.data should be channels × time points OR channels × time points × epochs. If size of EEG.data is channels × time points × epochs (has been epoched), all data will be used to calculate EEG network. If applicable, epochs in bad block (label 9999, marked by WB_EEG_Mark) will also be rejected automatically, and NOT used to calculate network.

Outputs

For each subject, output is a MATLAB .mat file (network_*.mat) in which is a structure EEG_results including connection matrices, mean connection matrices across epochs, z-score connection matrices and parameters. The results can be used to calculate network measures based on graph theory using EEG tool

WB_EEG_calcBasicNetIndices.

EEG_results.type: type of results, i.e. ‘network’;
EEG_results.nettype: type of network;
‘BU’: binary undirected network;
‘BD’: binary directed network;
‘WU’: weighted undirected network;
‘WD’: weighted directed network.
EEG_results.M: connection matrix (symmetric) across frequency bands. Size of M is channels × channels × epochs × frequency bands;
EEG_results.M_zscore: Fisher's z-score connection matrix across frequency bands. Size of M_zscore is channels × channels × epochs × frequency bands;
EEG_results.M_mean: mean connection matrix across epochs. Size of M_mean is channels × channels × frequency bands;
EEG_results.M_zscore_mean: mean Fisher's z-score connection matrix across epochs. Size of M_zscore_mean is channels × channels × frequency bands;
EEG_results.Block_percentage: percentage of EEG data used to calculate network.
EEG_results.filename: filename of EEG data;

EEG_results.parameter.bandLimit: an array with specific frequency bands;
EEG_results.parameter.bandName: a cell array with band names;
EEG_results.parameter.eventlabel: an eventlabel which means specific data;
EEG_results.parameter.selechanns: an array with selected channels;
EEG_results.parameter.epochLenth: length of small epochs. Unit is time point;
EEG_results.parameter.srate: sampling rate of EEG data;
EEG_results.parameter.method: method used to calculate EEG network;
EEG_results.parameter.proportion: overlapped proportion for each segments/sliding windows;
EEG_results.parameter.chanlocs: channel locations;
EEG_results.parameter.ref: reference of EEG data;
EEG_results.grouplabel: group label of subject. It will be used to statistical analysis in the future.

3.10 WB_EEG_CalcNetMeasures

WB_EEG_CalcNetMeasures is a tool to calculate network measures based on graph theory using BCT toolbox. MATLAB .mat file will be imported as EEG_results structure using WeBrain tool ‘WB_EEG_CalcNetwork’. EEG_reuslts should contain connection matrix M (EEG_results.M) and EEG result type (EEG_results.type) at least. First two dimensions of EEG_results.M should be channels/nodes × channels/nodes, and EEG_results.type should be ‘network’. More details of network measures can be seen in relative paper (Rubinov and Sporns, 2010) and BCT toolbox:

https://sites.google.com/site/bctnet/Home.

Parameters

nettype: the type of compatible associated network. Default is using nettype saved in EEG_results (‘[]’）. Nettype can be :
‘BU’: binary undirected network;
‘BD’: binary directed network;
‘WU’: weighted undirected network;
‘WD’: weighted directed network.

proportion: proportions of weights to preserve (proportional thresholding). ONLY used for nettype ‘BD’ or ‘BU’. Default is ‘[0.1:0.1:0.9]’. Range: proportion = 1 (all weights preserved) to proportion = 0 (no weights preserved).

flag: flag = 1: calculate network measures for networks (EEG_results.M) of all epochs
flag = 0: calculate network measures for mean networks (EEG_results.M_mean) across epochs (default).

Links:
BCT toolbox
https://sites.google.com/site/bctnet/Home

Outputs
For each subject, output is a MATLAB .mat file (netmeasure_*.mat) in which is a cell NetMeasure including network measures and parameters.

NetMeasure: a cell array which contains network measures (1 × frequencies, proportions × frequencies or proportions × epochs × frequencies)

NetMeasure.nettype: the type of compatible associated network;
NetMeasure.degree: node degree;
NetMeasure.indegree: node indegree;
NetMeasure.outdegree: node outdegree;
NetMeasure.Kn: mean degree of network;
NetMeasure.Kcost: cost of network;
NetMeasure.Kn_in: mean indegree of network;
NetMeasure.Kn_out: mean outdegree of network;

NetMeasure.strength: node strength;
NetMeasure.strength_m: mean node strength of network;
NetMeasure.instrength: node instrength;
NetMeasure.outstrength: node outstrength;

NetMeasure.instrength_m: mean node instrength of network;
NetMeasure.outstrength_m: mean node outstrength of network;

NetMeasure.Cn: node clustering coefficient. The clustering coefficient is the fraction of triangles around a node and is equivalent to the fraction of node's neighbors that are neighbors of each other;
NetMeasure.Cn_m: mean clustering coefficient of network;
NetMeasure.Ln: characteristic path length of network. The reachability matrix describes whether pairs of nodes are connected by paths (reachable). The distance matrix contains lengths of shortest paths between all pairs of nodes. The characteristic path length is the average shortest path length in the network;
NetMeasure.Eglob: global efficiency of network. The global efficiency is the average inverse shortest path length in the network, and is inversely related to the characteristic path length;
NetMeasure.Eloc: node local efficiency. The local efficiency is the global efficiency computed on the neighborhood of the node, and is related to the clustering coefficient.
NetMeasure.Eloc_m: mean local efficiency of network;

NetMeasure.BC: node betweenness centrality. Node betweenness centrality is the fraction of all shortest paths in the network that contain a given node. Nodes with high values of betweenness centrality participate in a large number of shortest paths;
NetMeasure.BC_m: mean node betweenness centrality of network;

NetMeasure.assort_coef: assortativity coefficient (undirected graph: strength/strength correlation). The assortativity coefficient is a correlation coefficient between the degrees of all nodes on two opposite ends of a link. A positive assortativity coefficient indicates that nodes tend to link to other nodes with the same or similar degree;
NetMeasure.assort_coef1: assortativity coefficient (directed graph: out-strength/in-strength correlation);
NetMeasure.assort_coef2: assortativity coefficient (directed graph: in-strength/out-strength correlation);
NetMeasure.assort_coef3: assortativity coefficient (directed graph: out-strength/out-strength correlation);
NetMeasure.assort_coef4: assortativity coefficient (directed graph: in-strength/in-strength correlation);

NetMeasure.rich_club.rich_coef: rich-club coefficients at level (degree) k, 1×levels. The rich club coefficient at level k is the fraction of edges that connect nodes of degree k or higher out of the maximum number of edges that such nodes might

share;

NetMeasure.rich_club.proportion: proportions (0<p<1) of the strongest weights; 1× proportions

NetMeasure.rich_club.rich_coef_proportion: rich-club coefficients of each proportion; proportions × levels

NetMeasure.rich_club.Nk: number of nodes with degree > k;

NetMeasure.rich_club.Ek: number of edges remaining in subgraph with degree > k.

3.11 WB_EEG_CalcLeadfield_standardBEM

WB_EEG_CalcLeadfield_standardBEM is a tool to generate conduction model of the head based on boundary element method (BEM) using standard MRI T1 image, and computes the forward model for many dipole locations on a 2D brain mesh or regular 3D grid and stores it for efficient inverse modelling using FieldTrip for EEG. The coordinates of head model is standard MNI space, and the electrodes will be aligned later to the existing standard head model. Some codes obtained from FieldTrip 20181025 and EEGLAB were integrated. More details of leadfield calculation can be seen in the FieldTrip toolbox (http://www.fieldtriptoolbox.org/).

Parameters

seleChanns: number with indices of the selected EEG channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all';

elecDirecFlag:

0: XYZ coordinates is the electorde array with their Cartesian x (the left ear is defined as -x axis), y (the nasion is the +y axis), z coordinates in three columns.

1: XYZ coordinates is the electorde array with their Cartesian x (the nasion is the +x axis), y (the left ear is the +y axis), z coordinates in three columns. Default is 1.

gridresolution: The grid resolution of dipoles (sources) inside the brain. If it is empty or <=0, the default dipoles are vertices which are little smaller than brain, and the orientations of dipoles are their normal vector directions, i.e. the normals of the brain mesh. If it >0 (unit is mm), the dipoles are distributed on regular 3D grid inside brain mesh. The orientations of dipoles are X, Y and Z orientations, i.e. there are X, Y and Z oriented dipoles. Default is '[]'.

Note:

- A standard headmodel using 'dipoli' method based on BEM were used in this tool (Fig. 9). The headmodel contains a standard Boundary Element Method volume conduction model of the head that can be used for EEG forward and inverse

computations. The geometry is based on the “colin27” template that is described further down. The BEM model is expressed in MNI coordinates in mm. A very similar BEM volume conduction model (based on the same template data) is described and validated by Fuchs et al. in Clin Neurophysiol. 2002 May; 113(5):702-12. More details can see: http://www.fieldtriptoolbox.org/template/headmodel/.

vol: headmodel used in the function;
vol.bnd: mesh of scalp, skull and brain;
vol.cond: conductivity of tissues, order is [scalp, skull and brain];
vol.type: BEM method used (default is ‘dipoli’);
vol.unit:unit of head model coordinates.

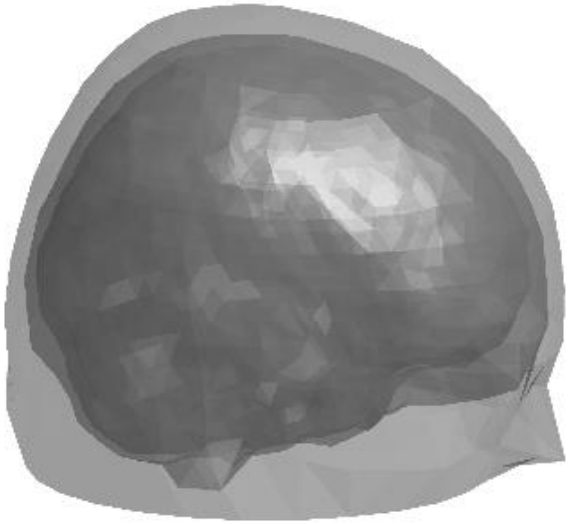

Fig. 9: A standard headmodel used in the tool. The conductivity of tissues are: brain = 0.33, skull = 0.0041 and scalp = 0.33; and the number of vertices of tissues are: brain = 1500 points, skull = 1000 points and scalp = 500 points.

- The “colin27” anatomical MRI and its relation to the TT and MNI template atlas is described in detail on http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach The original construction of the averaged MRI is detailed in [http://www.ncbi.nlm.nih.gov/pubmed/9530404| Holmes CJ, Hoge R, Collins L, Woods R, Toga AW, Evans AC. Enhancement of MR images using registration for signal averaging. J Comput Assist Tomogr. 1998 Mar-Apr;22(2):324-33.]
- Most of electrode location files are supported (More details can be seen in readlocs() in EEGLAB):
  '.loc' or '.locs' or '.eloc': polar coordinates;
  '.sph': Matlab spherical coordinates;
  '.elc': Cartesian 3-D electrode coordinates scanned using the EETrak software;
  '.elp': Polhemus-.'elp' Cartesian coordinates;

'.elp': BESA-'.elp' spherical coordinates: Need to specify 'filetype','besa';
'.xyz': Matlab/EEGLAB Cartesian coordinates;
'.asc', '.dat': Neuroscan-.'asc' or '.dat' Cartesian polar coordinates text file;
'.sfp': BESA/EGI-xyz Cartesian coordinates;
'.ced': ASCII file saved by pop_chanedit() in EEGLAB.

Links:

The standard headmodel:
http://www.fieldtriptoolbox.org/template/headmodel/

FieldTrip toolbox and Forward Problem:
http://www.fieldtriptoolbox.org/
http://www.fieldtriptoolbox.org/workshop/baci2017/forwardproblem/
http://www.fieldtriptoolbox.org/tutorial/headmodel_eeg_bem/?s[]=headmodel&s[]=eeg&s[]=bem

EEGLAB:
http://sccn.ucsd.edu/eeglab/index.html

Outputs

For each subject/electrode file, output is a MATLAB .mat file (lf_*.mat) in which contains a structure lf including leadfield and parameter settings and a structure elec_aligned including aligned electrodes and parameter settings (Fig. 11). Meanwhile, a *.png figure file of alignment will be provided to check electrode alignment (Fig. 10).

lf: leadfield results;
lf.leadfieldMatrix: leadfield matrix saved as (channels × sources);
lf.label: channel labels;
lf.dim: dimension of dipole (source) grid;
lf.unit: unit of head model coordinates;
lf.pos: positions of dipoles;
lf.mom: nomrals of dipoles (3 × sources);
lf.normals: nomrals of dipoles (sources × 3);
lf.inside: Boolean value of whether the lf.pos inside the brain;
lf.cfg: configuration of leadfield calculation;
lf.leadfield: leadfield saved as cell;
elec_aligned: aligned electrode coordinates;
elec_aligned.elecpos: aligned electrode positions;
elec_aligned.label: channel labels;
elec_aligned.cfg: configuration of alignment.

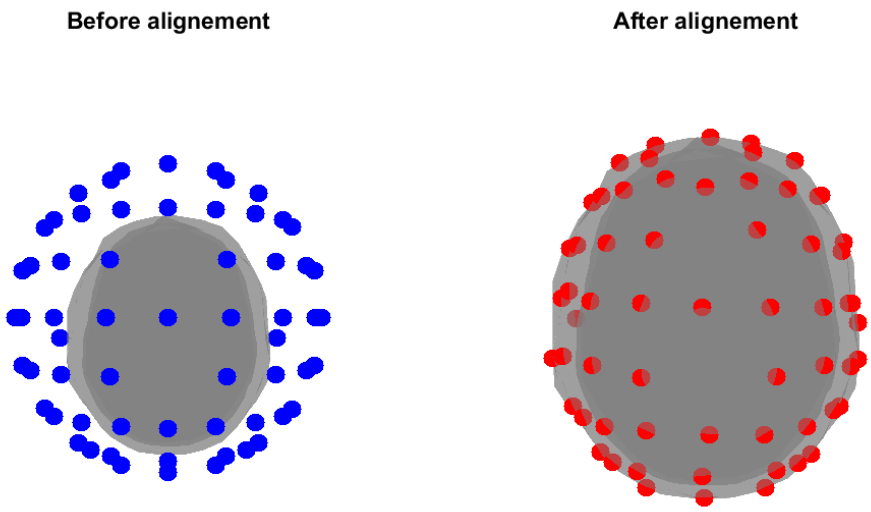

Fig. 10: An example result of alignment. The left figure shows the electrodes are not on the scalp, and the right figure shows the electrodes are well aligned.

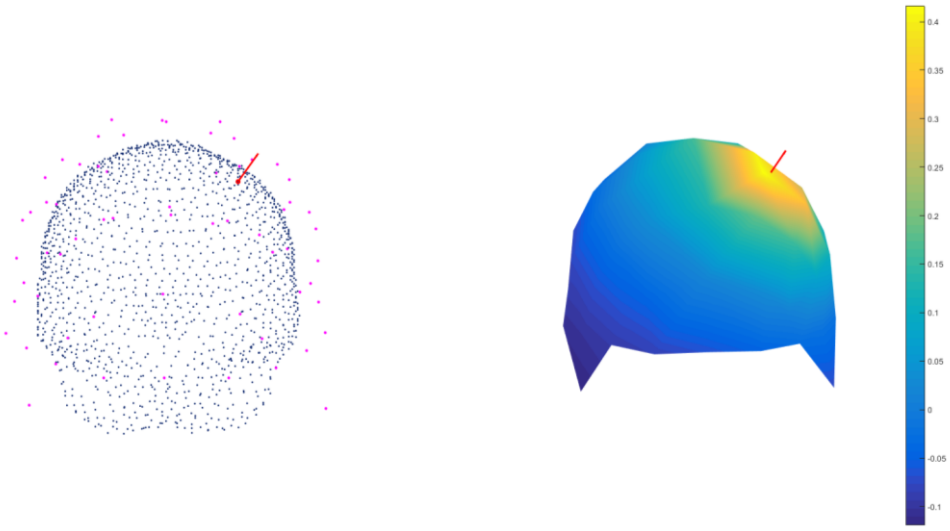

Fig. 11: An example result of leadfield calculated by the tool. The left figure shows 1500 sources/dipoles (black points) and electrodes (purple points).The red arrow shows a source/dipole with orientation (the dipole orientation is its normal vector direction). The right figure shows the leadfield distribution of the example dipole.

3.12 WB_EEG_sourceimage

WB_EEG_sourceimage is a tool to estimate source signals of scalp EEG/ERP data based on a forward model and inverse method (e.g. sLORETA). Source imaging estimation consists of (Fig.12):

[1] Loading EEG data and check the items including data, channel locations, and

sampling rate.
[2] Calculating the leadfield matrix by solving forward problem based on selected head model and channel locations. Or obtaining a user defined leadfield matrix.
[3] If needed, passband filtering the EEG data. Default is no filtering.
[4] Specific event data can be extracted according to the input 'eventlabel'. If the input 'eventlabel' is empty, all data will be used. And, specific EEG signals will be divided into small epochs. If applicable, EEG segments in bad block (label 9999, marked by WB_EEG_Mark) will be rejected automatically, and NOT used to estimate sources.
[5] EEG data of each epoch (default is 5-s epoch) is subjected to estimate sources to obtain the EEG signals in the source space using an inverse method such as 'sLORETA' (Dale et al., 2000; Pascual-Marqui, 2002).
[6] If needed, matching source signals to a brain template (e.g. AAL template) to obtain the averaged source signals of brain regions.
[7] Saving the results of EEG source signals and parameters as a .set file.

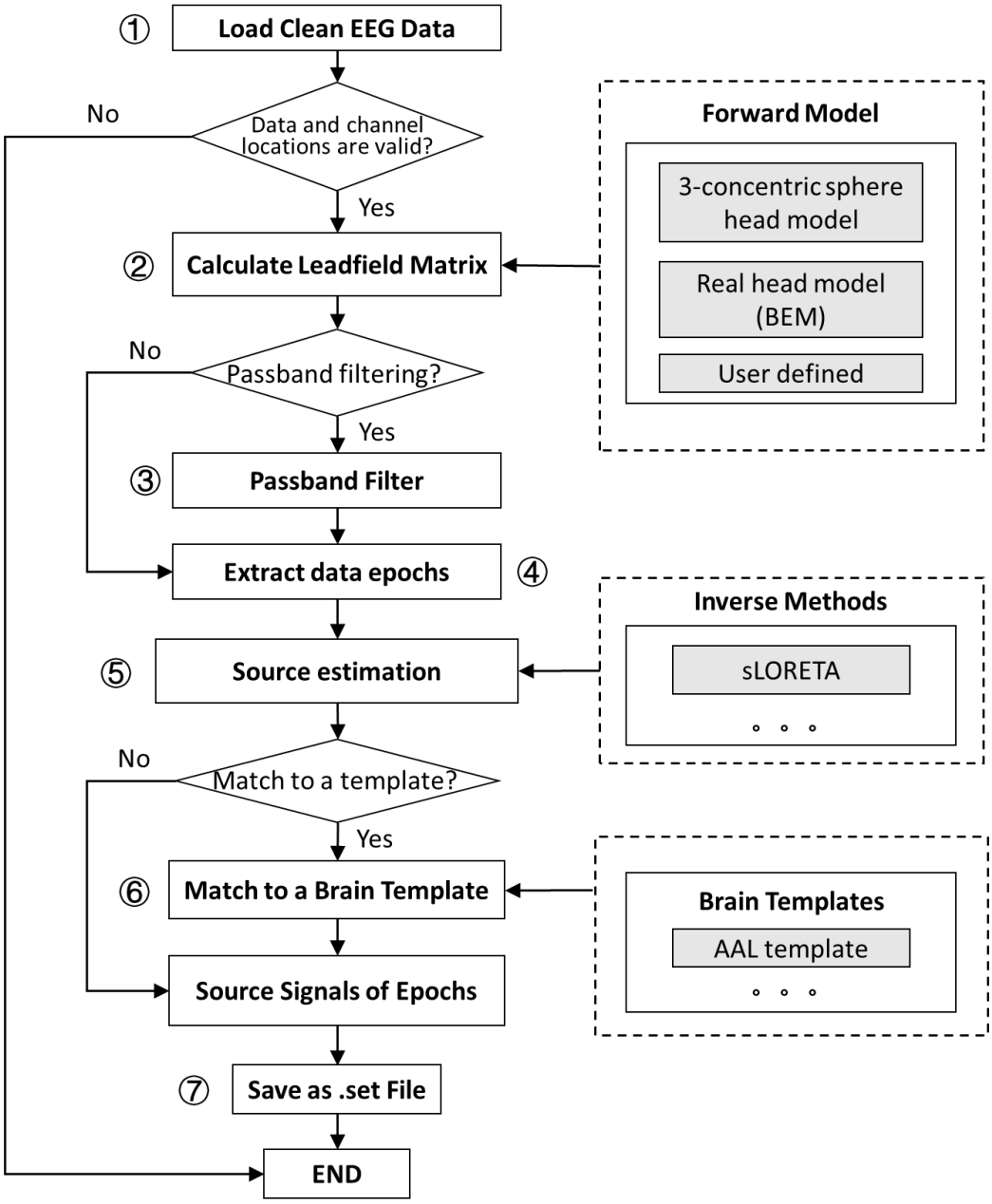

Fig. 12: The pipeline of EEG source imaging. (1) Loading EEG data and check the items including data, channel locations, and sampling rate. (2) Calculating the leadfield matrix by solving forward problem based on selected head model and channel locations. Or obtaining a user defined leadfield matrix. The channel locations will be matched to the head model automatically. (3) If needed, passband filtering the EEG data. Default is no filtering. (4) Specific event data can be extracted according to the input 'eventlabel'. If the input 'eventlabel' is empty, all data will be used. And, specific EEG signals will be divided into small epochs. If applicable, EEG segments in bad block (label 9999, marked by wb_EEG_Mark) will be rejected automatically, and NOT used to estimate sources. (5) EEG data of each epoch (default is 5-s epoch) is subjected to estimate sources to obtain the EEG signals in the source space using an inverse method such as 'sLORETA'. (6) If needed, matching source signals to a brain

template (e.g. AAL template) to obtain the averaged source signals of brain regions. (7) Saving the results of EEG source signals and parameters as a .set file.

Parameters

epochLenth: Length of small epochs to calculate EEG source signals. Unit is second. Default is 5s. If epochLenth is negative, it means that if possible, data before event labels (eventlabel) will be used (no overlapped).

eventlabel: Event label which means specific event data. Default is empty. If it is empty, all data will be used. If eventlabel is not found, NO data will be epoched and calculated. If structure event (eventlabel) doesn't include duration, the duration will be equal to epochLenth. You can also input multiple event labels at once, split by comma (e.g. 'S22,S23'); and all these event data will be calculated.

seleChanns: a string number with indices of the selected channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all'.

passband: Pass band of filtering. Default is no filtering (i.e. '[]').

ForwardMethod: Forward method used to calculate the leadfield matrix.

ForwardMethod = 0: use a user defined leadfield matrix. While ForwardMethod = 0, the leadfield file must be inputed.

ForwardMethod = 1: if EEG contains correct channel locations, it will automatically calculate leadfield matrix based on 3-concentric sphere head model.

ForwardMethod = 2: if EEG contains correct channel locations, it will automatically calculate leadfield matrix based on based on real head modal (BEM modal) using FieldTrip.

proportion: overlapped percentage for each segments/sliding windows. It should be [0,1). Default is 0 (no overlapped).

SourceMethod: Inverse method used to calcualte EEG source signals. Default is 'sloreta'.

'sloreta': EEG source imaging using sLORETA method (based on Dale et al. standardization) (Dale et al., 2000; Pascual-Marqui, 2002). The sLORETA is a tomographic method for electric neuronal activity, where localization inference is based on images of standardized current density. The method is denoted as standardized low resolution brain electromagnetic tomography (sLORETA). Noting that the Dale et al. standardization is used in the standardized estimation of sLORETA in the WeBrain.

alpha: regularization parameter of inverse method.

alpha >= 0: use user-defined value;

alpha = 'mean': use averaged alpha with tikh regularization (default);

alpha = 'timevarying': use time-varying alpha with tikh regularization. The time cost of 'time-varying' alpha is extremely high for EEG time courses. It is used for the situation of less topographies.

elecDirecFlag: It is valid while calcuating leadfield. It is optional.

elecDirecFlag = 0: XYZ coordinates is the electorde array with their Cartesian x (the left ear is defined as -x axis),y (the nasion is the +y axis), z coordinates in three columns.

elecDirecFlag = 1: XYZ coordinates is the electorde array with their Cartesian x (the nasion is the +x axis),y (the left ear is the +y axis), z coordinates in three columns. Default is 1.

matchflag: Matching the source signals to a brain template (e.g. AAL template) to obtain the averaged source signals of brain regions.

matchflag = [] (default): no matching;

matchflag = 'aal': matching to AAL template and obtaining averaged source signals of brain regions;

erpflag: Estimating the source signals for averaged ERP waves? It is useful for the segmented ERP epoch data (i.e. the demension of EEG.data is channels × time points × epochs).

erpflag = 0 (default): estimating the source signals of each epoch;

erpflag = 1: averaging trials and then estimating the source signals of averaged ERP waves. It is valid, while the data is epoched.

leadfieldfile: a user defined lead field file (optional). It is valid, while the ForwardMethod = 0 only. The file could be a MATLAB .mat file, containing a matrix named 'leadfield' (channels × sources/dipoles, e.g. 60 channels × 6144 sources/dipoles) or a MATLAB structure containing leadfield of x,y,z-orientations (e.g. leadfield.X with dimension channels × dipoles (x-orientation);leadfield.Y with dimension channels × dipoles (y-orientation); leadfield.Z with dimension channels × dipoles (z-orientation)) which is calculated by using the forward theory, based on the electrode montage, head model and equivalent source model. It can also be the output of ft_prepare_leadfield.m (e.g. lf.leadfield, dipoles contain x,y,z-orientations, 60 channels × 6144*3 dipoles) based on real head modal (BEM modal) using FieldTrip.

gridresolution：The grid resolution of dipoles (sources) inside the brain (optional). It is valid, while ForwardMethod = 2. If it is empty or <=0, the default dipoles are vertices which are little smaller than brain, and the orientations of dipoles are their normal vector directions, i.e. the normals of the brain mesh. If it >0 (unit is mm), the dipoles are distributed on regular 3D grid inside brain mesh. The orientations of dipoles are X, Y and Z orientations, i.e. there are X, Y and Z oriented dipoles. Default is empty.

srate: Sampling rate of EEG data (optional). Default is obtained from EEG data ('[]').

intermfile: A mat file contains a number of intermediate data which may be required by some methods in the WeBrain. It is no need to input.

vol_bem1: a standard headmodel using 'dipoli' method based on BEM. It is optional, and is valid while ForwardMethod = 2.

v0_aal: header information for AAL template image.

V.fname - the filename of the image.

V.dim - the x, y and z dimensions of the volume

V.dt - A 1×2 array. First element is datatype (see spm_type). The second is 1 or 0 depending on the endian-ness.

V.mat - a 4×4 affine transformation matrix mapping from voxel coordinates to real world coordinates.

V.pinfo - plane info for each plane of the volume.

V.pinfo(1,:) - scale for each plane

V.pinfo(2,:) - offset for each plane

The true voxel intensities of the jth image are given by: val*V.pinfo(1,j) + V.pinfo(2,j) V.pinfo(3,:) - offset into image (in bytes). If the size of pinfo is 3x1, then the volume is assumed to be contiguous and each plane has the same scale factor and offset.

AAL: a matrix with brain region number of AAL template. It is optional, and is valid while matchflag = 'aal'.

Output:

EEG: a structure of EEG containing results of EEG source imaging.

EEG.data: EEG sources with dimension No. of dipoles × No. of time points.

EEG.parameter.eventlabel: An eventlabel which means good quality data;

EEG.parameter.selechanns: An array with selected channels;

EEG.parameter.epochLenth: Length of small epochs, and unit is time point;

EEG.parameter.srate: Sampling rate of EEG data;

EEG.parameter.sourcemethod: Inverse method used to estimate EEG source signals;

EEG.parameter.proportion: overlapped percentage for each segments/sliding windows;

EEG.parameter.chanlocs: EEG channel locations on the scalp;

EEG.parameter.ref: original EEG reference;

EEG.parameter.leadfield: leadfield matrix;

EEG.parameter.alpha: regularization parameter;

EEG.parameter.elecDirecFlag: flag of channel directions;

EEG.parameter.chaninfo: channel information.

EEG.parameter.erpflag: flag of average ERP.

EEG.parameter.headmodel: default headmodel, and details see below;

EEG.parameter.ForwardMethod: forward method used in the tool.

EEG.parameter.passband: pass band of filtering.

EEG.parameter.gridresolution: the grid resolution of dipoles (sources) inside the brain.

EEG.parameter.elec_aligned: aligned channel locations of electrodes.

EEG.parameter.BrainRegionInd: brain region indices according to a template such as AAL.

EEG.parameter.template: the template used to obtain the averaged source signals of brain regions.

EEG.parameter.matchflag: flag of matching source signals to a brain template.

Important Issues in Source Imaging

Forward model

(1) 3-concentric sphere head model

Default head model are based on 3-concentric spheres, in which the radii of spheres are normalized by the largest sphere. i.e. [0.87, 0.92, 1], corresponding to brain, skull and scalp. Conductivities of 3 concentric spheres are normalized by the largest resistivity, and default is [1, 0.0125, 1] (Fig. 13). A high-density canonical cortical mesh was used to define the dipoles. These meshes were obtained by warping a template mesh to the T1-weighted structural anatomy of an individual subject, as described in Mattout et al. (2007) (Mattout et al., 2007). This warping is the inverse of the transformation derived for the spatial normalization of the subject's structural MRI image. The template mesh was generated by Fieldtrip (http://fieldtrip.fcdonders.nl/download.php), and was extracted from a structural MRI of a neurotypical male. The wrapping procedure provided a high-density mesh with 33,001 vertexes, which was uniformly distributed on the gray-white matter interface. The mesh was further down-sampled to 6,144 vertexes (SPM MNI space) to reduce the computational load. More details see: *Lei, X., et al. (2012). "Incorporating fMRI Functional Networks in EEG Source Imaging: A Bayesian Model Comparison Approach." Brain Topogr* (Lei et al., 2012). Finally, the leadfield matrix was calculated analytically based on spherical harmonic spectra theory (Yao, 2000; Yao et al., 2004).

headmodel details: default head model are based on 3-concentric spheres.

headmodel.r: the radii of spheres which are normalized by the largest sphere. E.g. [0.87,0.92,1].

headmodel.type: head model type. It is concentric spheres

headmodel.cond (optional): the conductivities (Brain, Skull and Scalp) which are normalized by the largest resistivity. E.g. For 3 concentric spheres, default is [1, 0.0125, 1];

headmodel.tissue (optional): tissues. E.g. ['brain' 'skull' 'scalp'];

headmodel.o: center of the spheres (optional if origin).e.g. [0,0,0].

headmodel.terms: the constants for the Legendre expansion for EEG leadfields. The returning value is the constant for the third layer on the outer surface, e.g. K(3, r=R3).

headmodel.sourcemesh.bnd: the source mesh of dipoles.

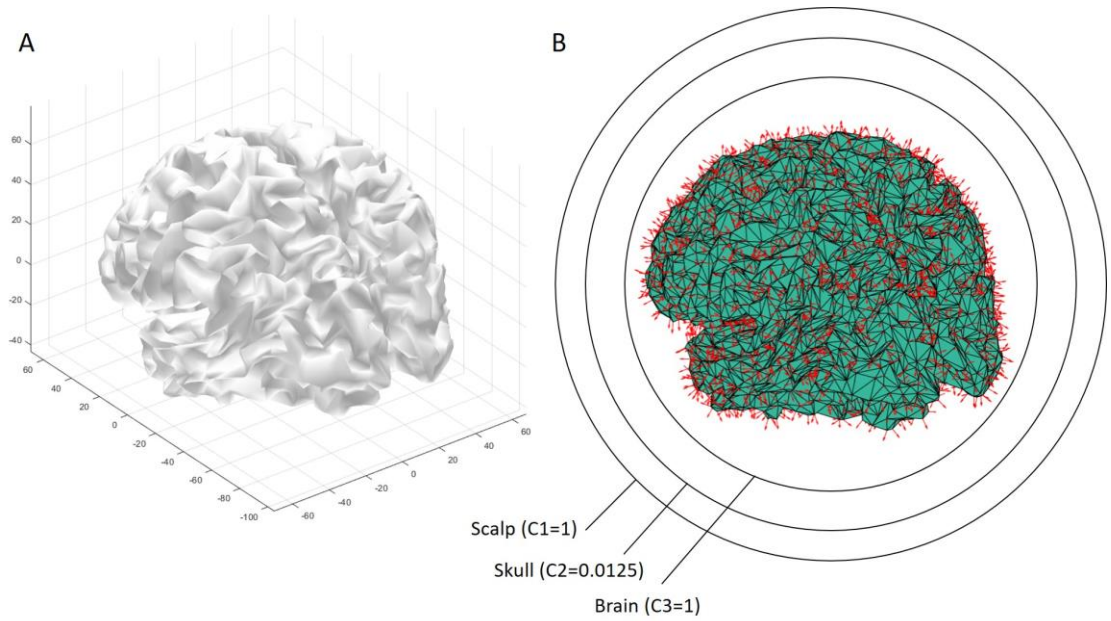

Fig.13: The default 3-concentric sphere head model in the WeBrain. A: the default geometrically triangular grid in the WeBrain which is based on the standard brain (MNI space) and contains 6144 dipoles. B: 3-concentric spheres and their conductivities. Nodes and their normal vector directions are showed.

(2) Real head model based on Boundary Element Method

The details about BEM and real head model used in the WeBrain are as follows:

A standard headmodel using 'dipoli' method based on Boundary Element Method (BEM) were used in the WeBrain. The headmodel contains a standard BEM volume conduction model of the head that can be used for EEG forward and inverse computations. The geometry is based on the "colin27" template that is described further down. The BEM model is expressed in MNI coordinates in mm. A very similar BEM volume conduction model (based on the same template data) is described and validated by Fuchs et al. in Clin Neurophysiol. 2002 May;113(5):702-12. More details see : http://www.fieldtriptoolbox.org/template/headmodel/. More details about the real head model and BEM can also be seen in the 3.11 WB_EEG_CalcLeadfield_standardBEM.

The "colin27" anatomical MRI and its relation to the TT and MNI template atlas is described in detail on http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach. The original construction of the averaged MRI is detailed in [http://www.ncbi.nlm.nih.gov/pubmed/9530404| Holmes CJ, Hoge R, Collins L, Woods R, Toga AW, Evans AC. Enhancement of MR images using registration for signal averaging. J Comput Assist Tomogr. 1998 Mar-Apr;22(2):324-33.]

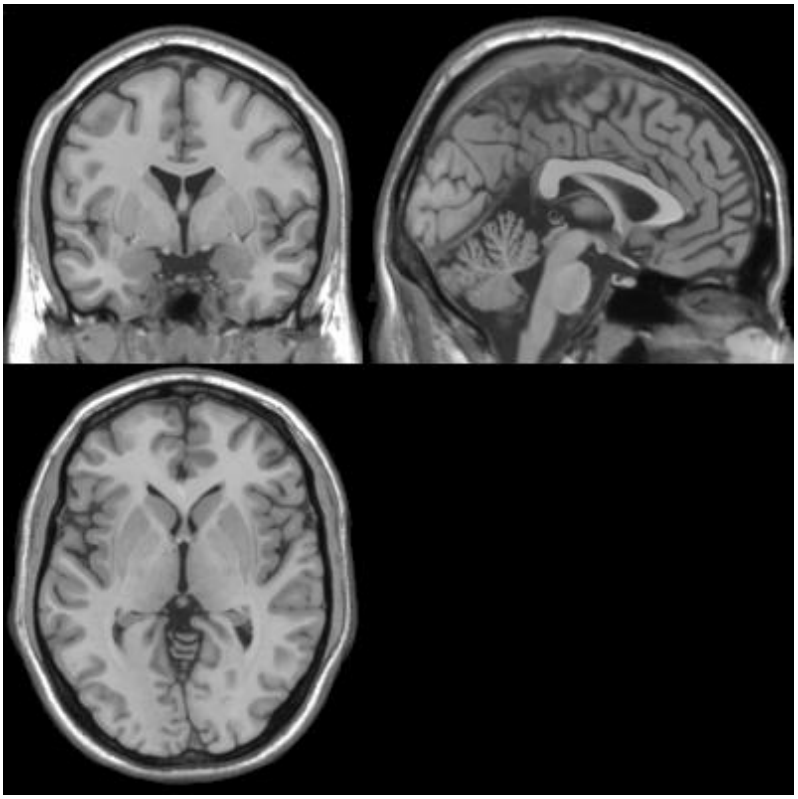

Fig.14: The “colin27” anatomical MRI.

headmodel: headmodel info

headmodel.bnd: mesh of scalp, skull and brain.

headmodel.cond: conductivity of tissues, order is [scalp, skull and brain].

headmodel.type: BEM method used (i.e. ‘dipoli’).

headmodel.unit:unit of head model coordinates.

Inverse Method

(1) sLORETA

The details of sLORETA can be seen in the paper: *Pascual-Marqui, R. D. (2002). "Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details." Methods Find Exp Clin Pharmacol 24 Suppl D: 5-12.*

The sLORETA tool is also available on http://www.uzh.ch/keyinst/loreta.htm.

Brain Template

(1) AAL template

An automated anatomical parcellation of the spatially normalized single-subject high-resolution T1 volume provided by the Montreal Neurological Institute (MNI), named Automated Anatomical Labeling (AAL, Fig.15) template (Tzourio-Mazoyer et al., 2002), is used to get average source signals in the brain regions in the WeBrain. The list of brain regions can be seen in the Table 1. More details about the AAL template are available on https://www.gin.cnrs.fr/en/tools/aal/.

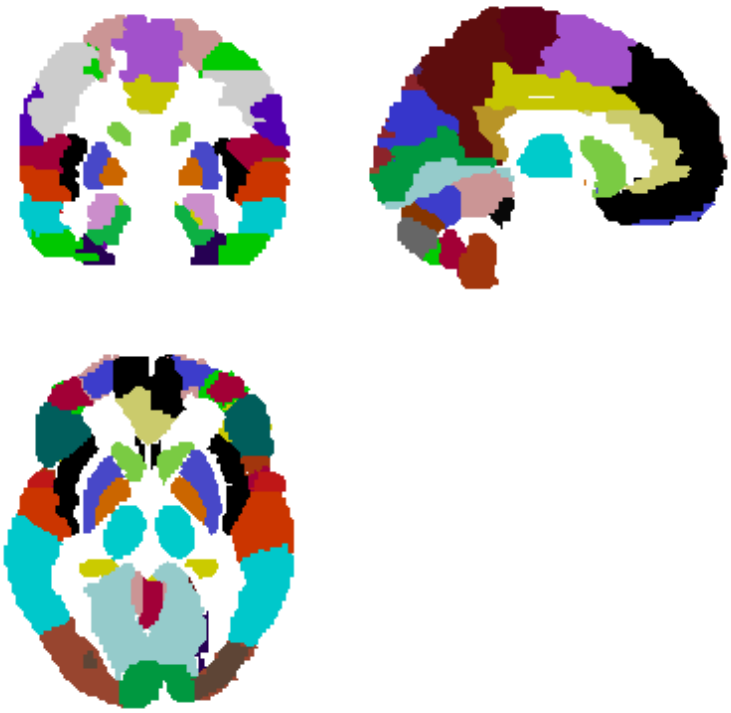

Fig. 15: AAL template used in the WeBrain.

Table 1: The list of brain regions for AAL template.

No.	Brain Region (abbreviation) L/R: left/right	Full Names	Chinese
1	Precentral_L	Precental gyrus	中央前回
2	Precentral_R	Precental gyrus	中央前回
3	Frontal_Sup_L	Superior frontal gyrus, dorsolateral	背外侧额上回
4	Frontal_Sup_R	Superior frontal gyrus, dorsolateral	背外侧额上回
5	Frontal_Sup_Orb_L	Superior frontal gyrus, orbital part	眶部额上回
6	Frontal_Sup_Orb_R	Superior frontal gyrus, orbital part	眶部额上回
7	Frontal_Mid_L	Middle frontal gyrus	额中回
8	Frontal_Mid_R	Middle frontal gyrus	额中回
9	Frontal_Mid_Orb_L	Middle frontal gyrus, orbital part	眶部额中回
10	Frontal_Mid_Orb_R	Middle frontal gyrus, orbital part	眶部额中回
11	Frontal_Inf_Oper_L	Inferior frontal gyrus, opercular part	岛盖部额下回
12	Frontal_Inf_Oper_R	Inferior frontal gyrus, opercular part	岛盖部额下回
13	Frontal_Inf_Tri_L	Inferior frontal gyrus, triangular part	三角部额下回
14	Frontal_Inf_Tri_R	Inferior frontal gyrus, triangular part	三角部额下回
15	Frontal_Inf_Orb_L	Inferior frontal gyrus, orbital part	眶部额下回
16	Frontal_Inf_Orb_R	Inferior frontal gyrus, orbital part	眶部额下回
17	Rolandic_Oper_L	Rolandic operculum	中央沟盖
18	Rolandic_Oper_R	Rolandic operculum	中央沟盖

19	Supp_Motor_Area_L	Supplementary motor area	补充运动区
20	Supp_Motor_Area_R	Supplementary motor area	补充运动区
21	Olfactory_L	Olfactory cortex	嗅皮质
22	Olfactory_R	Olfactory cortex	嗅皮质
23	Frontal_Sup_Medial_L	Superior frontal gyrus, medial	内侧额上回
24	Frontal_Sup_Medial_R	Superior frontal gyrus, medial	内侧额上回
25	Frontal_Med_Orb_L	Superior frontal gyrus, medial orbital	眶内额上回
26	Frontal_Med_Orb_R	Superior frontal gyrus, medial orbital	眶内额上回
27	Rectus_L	Gyrus rectus	回直肌
28	Rectus_R	Gyrus rectus	回直肌
29	Insula_L	Insula	脑岛
30	Insula_R	Insula	脑岛
31	Cingulum_Ant_L	Anterior cingulate and paracingulate gyri	前扣带和旁扣带脑回
32	Cingulum_Ant_R	Anterior cingulate and paracingulate gyri	前扣带和旁扣带脑回
33	Cingulum_Mid_L	Median cingulate and paracingulate gyri	内侧和旁扣带脑回
34	Cingulum_Mid_R	Median cingulate and paracingulate gyri	内侧和旁扣带脑回
35	Cingulum_Post_L	Posterior cingulate gyrus	后扣带回
36	Cingulum_Post_R	Posterior cingulate gyrus	后扣带回
37	Hippocampus_L	Hippocampus	海马
38	Hippocampus_R	Hippocampus	海马
39	ParaHippocampal_L	Parahippocampal gyrus	海马旁回
40	ParaHippocampal_R	Parahippocampal gyrus	海马旁回
41	Amygdala_L	Amygdala	杏仁核
42	Amygdala_R	Amygdala	杏仁核
43	Calcarine_L	Calcarine fissure and surrounding cortex	距状裂周围皮层
44	Calcarine_R	Calcarine fissure and surrounding cortex	距状裂周围皮层
45	Cuneus_L	Cuneus	楔叶
46	Cuneus_R	Cuneus	楔叶
47	Lingual_L	Lingual gyrus	舌回
48	Lingual_R	Lingual gyrus	舌回
49	Occipital_Sup_L	Superior occipital gyrus	枕上回
50	Occipital_Sup_R	Superior occipital gyrus	枕上回
51	Occipital_Mid_L	Middle occipital gyrus	枕中回
52	Occipital_Mid_R	Middle occipital gyrus	枕中回
53	Occipital_Inf_L	Inferior occipital gyrus	枕下回
54	Occipital_Inf_R	Inferior occipital gyrus	枕下回
55	Fusiform_L	Fusiform gyrus	梭状回
56	Fusiform_R	Fusiform gyrus	梭状回
57	Postcentral_L	Postcentral gyrus	中央后回
58	Postcentral_R	Postcentral gyrus	中央后回
59	Parietal_Sup_L	Superior parietal gyrus	顶上回
60	Parietal_Sup_R	Superior parietal gyrus	顶上回

61	Parietal_Inf_L	Inferior parietal, but supramarginal and angular	顶下缘角回
62	Parietal_Inf_R	Inferior parietal, but supramarginal and angular	顶下缘角回
63	SupraMarginal_L	Supramarginal gyrus	缘上回
64	SupraMarginal_R	Supramarginal gyrus	缘上回
65	Angular_L	Angular gyrus	角回
66	Angular_R	Angular gyrus	角回
67	Precuneus_L	Precuneus	楔前叶
68	Precuneus_R	Precuneus	楔前叶
69	Paracentral_Lobule_L	Paracentral lobule	中央旁小叶
70	Paracentral_Lobule_R	Paracentral lobule	中央旁小叶
71	Caudate_L	Caudate nucleus	尾状核
72	Caudate_R	Caudate nucleus	尾状核
73	Putamen_L	Lenticular nucleus, putamen	豆状壳核
74	Putamen_R	Lenticular nucleus, putamen	豆状壳核
75	Pallidum_L	Lenticular nucleus, pallidum	豆状苍白球
76	Pallidum_R	Lenticular nucleus, pallidum	豆状苍白球
77	Thalamus_L	Thalamus	丘脑
78	Thalamus_R	Thalamus	丘脑
79	Heschl_L	Heschl gyrus	颞横回
80	Heschl_R	Heschl gyrus	颞横回
81	Temporal_Sup_L	Superior temporal gyrus	颞上回
82	Temporal_Sup_R	Superior temporal gyrus	颞上回
83	Temporal_Pole_Sup_L	Temporal pole: superior temporal gyrus	颞极：颞上回
84	Temporal_Pole_Sup_R	Temporal pole: superior temporal gyrus	颞极：颞上回
85	Temporal_Mid_L	Middle temporal gyrus	颞中回
86	Temporal_Mid_R	Middle temporal gyrus	颞中回
87	Temporal_Pole_Mid_L	Temporal pole: middle temporal gyrus	颞极：颞中回
88	Temporal_Pole_Mid_R	Temporal pole: middle temporal gyrus	颞极：颞中回
89	Temporal_Inf_L	Inferior temporal gyrus	颞下回
90	Temporal_Inf_R	Inferior temporal gyrus	颞下回

3.13 WB_EEG_calcLZC

WB_EEG_calcLZC is a tool to calculate Lempel-Ziv Complexity indices for each EEG channel. “Complexity” is a widely used nonlinear dynamic concept in the EEG analysis. Because the complexity analysis mainly represents the degree of randomness in time series, the complexity of EEG data measures the capacity of information in the EEG signal fragment and then may reflect the underlying activeness of the neurons (Bai et al., 2015; Hu and Zhang, 2019). Lempel-Ziv complexity was proposed by Lempel and Ziv (Lempel and Ziv, 1976) and along with its derivatives has found numerous applications in characterizing the randomness of biological signals,

especially in EEG analysis.

Calculating LZC consists of:
[1] Specific event data can be extracted according to the input ‘eventlabel’. If the input ‘eventlabel’ is empty, all data will be used. If applicable, EEG segments in bad block (label 9999, marked by wb_pipeline_EEG_Mark) will be rejected automatically, and NOT used to calculate Lempel-Ziv Complexity.
[2] Specific EEG signals will be divided into small epochs.
[3] EEG data of each epoch (default is 1-s epoch) was subjected to calculate the Lempel-Ziv Complexity across electrodes in the specific bands (default is fullband).

Noting that data in bad blocks (labeled as 9999) will not be used to calculate complexity indices !!!

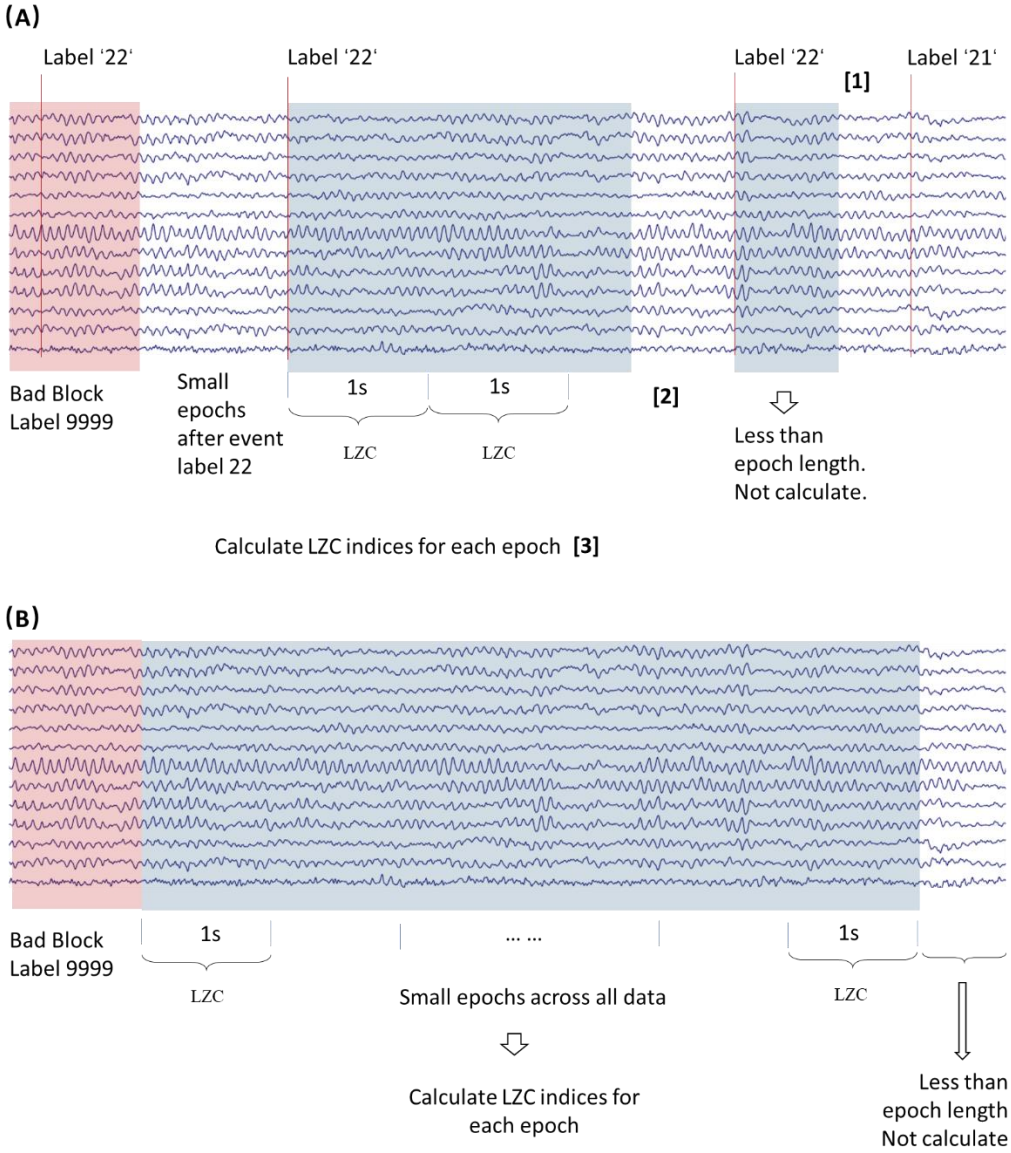

Fig. 16: Pipelines of calculating LZC indices of EEG data. (A) Calculating LZC

indices of a specific event data. Step1: Specific event data are first extracted according to the input 'eventlabel' (e.g. label '22'). If applicable, EEG segments in bad block (label 9999, marked by WB_EEG_Mark) will also be rejected automatically, and NOT used to calculate indices. Step2: Specific event EEG signals will be divided into small epochs. Step3: EEG data of each small epoch (default is 1s epoch) will be subjected to calculate LZC indices across electrodes in the specific bands. (B) Calculating LZC indices across continuous data. All data will be divided into small epochs first, and then data of each epoch (excluding bad blocks) will be used to calculate LZC indices.

Lempel-Ziv Complexity (Bai et al., 2015; Hu and Zhang, 2019) including:

PLZC: Permutation Lempel-Ziv Complexity

LZCmean: Lempel-Ziv Complexity using coarse-graining method based mean value (In this coarse-graining approach, the mean value of the amplitude values will be selected as Td).

LZCmedian: Lempel-Ziv Complexity using coarse-graining method based median value (In this coarse-graining approach, the median value of the amplitude values will be selected as Td).

LZCmid: Lempel-Ziv Complexity using coarse-graining method based midpoint value (In this coarse-graining approach, the midpoint value of the amplitude values will be selected as Td).

LZCkmeans: Lempel-Ziv Complexity using coarse-graining method based k-means value (The LZCkmeans estimates the grouping of data points around centroids corresponding to points around which most of the data is agglomerated). In the initial iteration of the method, we set the two initial centroids as z1(1)= mean + 0.05*mean and z2(1) = mean - 0.05*mean. The vaule of 0.05 is according to previous studies (Linde et al. 1980; Zhou et al. 2011).

References:

Hu, L. and Z. Zhang (2019). EEG Signal Processing and Feature Extraction, Springer. Chapter 11 Nonlinear Neural Dynamics

Yang Bai, Zhenhu Liang, Xiaoli Li, A permutation Lempel-Ziv complexity measure for EEG analysis, Biomedical Signal Processing and Control, Volume 19, 2015.

Parameters

Input: zip paths of each subject (separate by commas). e.g. '*\sub_01.zip,*\sub_02.zip'. EEG data will be loaded as EEG structure imported by EEGLAB. EEG.data should be channels × time points OR channels × time points × epochs.

Output: output path.

combs_project_id: project ID (only print for WeBrain).

epochLenth: length of small epochs to calculate EEG LZC. unit is second. Default is 5s. If epochLenth is negative, it means that if possible, data before event labels (eventlabel) will be used (no overlapped).

eventlabel: Event label which means specific event data. Default is empty. If it is empty, all data will be used. If eventlabel is not found, NO data will be epoched and calculated. If structure event (eventlabel) doesn't include duration, the duration will be equal to epochLenth. You can also input multiple event labels at once, split by comma (e.g. 'S22,S23'); and all these event data will be calculated.

bandLimit: A cell array with specific frequency bands. Default is full band (empty).

seleChanns: number with indices of the selected channels (e.g. '[1:4,7:30]' or 'all'). Default is 'all';

m: the number of sampling points required for the pattern calculation, default is 4; it is used to calculate PLZC.

tau: the interval between the extracted sampling points, if it is 1, then the two extracted sampling points are separated by one signal sampling point, default is 9; it is used to calculate PLZC.

proportion: overlapped percentage for each segments/sliding windows. It should be [0,1). Default is 0 (no overlapped).

Output:

EEG_results: results including Lempel-Ziv Complexity matrices and parameters.

EEG_results.type: type of results, i.e. 'complexity'.

EEG_results.PLZC: Permutation Lempel-Ziv Complexity, channels × epochs × frequency bands

EEG_results.LZCmean: Lempel-Ziv Complexity using coarse-graining method based mean value, channels × epochs × frequency bands

EEG_results.LZCmedian: Lempel-Ziv Complexity using coarse-graining method based median value, channels × epochs × frequency bands

EEG_results.LZCmid: Lempel-Ziv Complexity using coarse-graining method based midpoint value, channels × epochs × frequency bands

EEG_results.LZCkmeans: Lempel-Ziv Complexity using coarse-graining method based k-means value. In the initial iteration of the method, we set the two initial centroids as z1(1) = mean + 0.05*mean and z2(1) = mean - 0.05*mean. The vaule of 0.05 is according to previous studies (Linde et al. 1980; Zhou et al. 2011). channels × epochs × frequency bands

EEG_results.mPLZC: mean PLZC across epochs, size is channels × frequency bands;

EEG_results.mLZCmean: mean LZCmean across epochs, size is channels ×

frequency bands;

EEG_results.mLZCmedian: mean LZCmedian across epochs, size is channels × frequency bands;

EEG_results.mLZCmid: mean LZCmid across epochs, size is channels × frequency bands;

EEG_results.mLZCkmeans: mean LZCkmeans across epochs, size is channels × frequency bands;

EEG_results.Block_percentage: percentage of EEG data used to calculate indices.

EEG_results.filename: filename of EEG data.

EEG_results.parameter.bandLimit: An array with specific frequency bands;

EEG_results.parameter.bandName: A cell array with band names;

EEG_results.parameter.eventlabel: An eventlabel which means good quality data;

EEG_results.parameter.selechanns: An array with selected channels;

EEG_results.parameter.epochLenth: Length of small epochs. Unit is time point.

EEG_results.parameter.srate: Sampling rate of EEG data.

EEG_results.parameter.m: the number of sampling points required for the pattern calculation.

EEG_results.parameter.tau: the interval between the extracted sampling points.

EEG_results.parameter.chanlocs: selected channel locations;

EEG_results.parameter.ref: EEG reference.

3.14 WB_EEG_timefreq

WB_EEG_timefreq is a tool to conduct time-frequency analysis to reveal time-varying spectrum of non-stationary EEG signals. It will calculate time-frequency spectrum and inter-trial coherence (ITC) events across event-related trials (epochs) of each channel time series. More details can be seen in EEGLAB function, newtimef() or timefreq().

Calculating time-frequency spectrum and ITC indices consists of

[1] Specific event data are first extracted according to the input 'eventlabel'. If the input 'eventlabel' is empty and the data is 2D (channels X time points, i.e. not epoched), all data will be used. If applicable, EEG segments in bad block (label 9999, marked by wb_pipeline_EEG_Mark) will also be rejected automatically, and NOT used to calculate power indices.

[2] Specific event EEG signals will be divided into small epochs.

[3] EEG data of epochs (default is 1s epoch) were subjected to time-frequency

analysis to obtain the ERSP and ITC indices at each electrode in the full bands (Default maximal cut-off frequecy band is 50Hz).

Noting that data in bad blocks (labeled as 9999) will not be used to calculate indices !!!

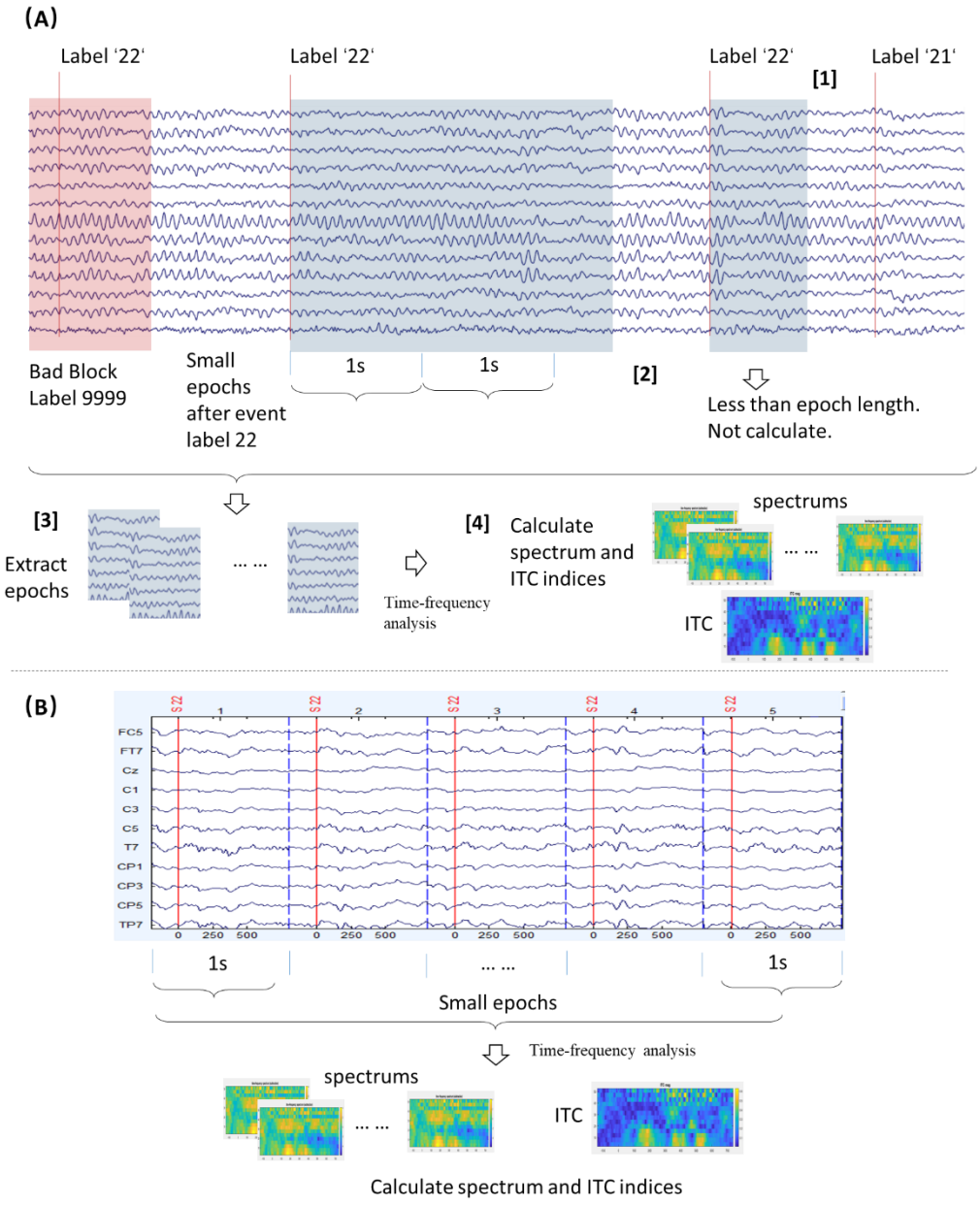

Fig. 17: Pipelines of calculating time-frequency spectrum and ITC indices of EEG data.

The spectrum value is calculated by:

$$Y_{spectrum} = 10 * log10(\frac{||\frac{2}{0.375} * Y||^2}{window\ size})$$

where Y is complex number calculated by time-frequency decomposition, $\|.\|$ is complex modulus operations (using 'abs' function of MATLAB), the unit is $10 \cdot \log10\mu V^2/Hz$, the window size is calculated by $\max(\text{pow2}(\text{nextpow2}(\text{length}(\text{epoch})) - 3),4)$. Noting that, multiply by 2 account for negative frequencies, and counteract the reduction by a factor 0.375 that occurs as a result of cosine (Hann) tapering.

References:

Hu, L. and Z. Zhang (2019). EEG Signal Processing and Feature Extraction, Springer. Chapter 6 Spectral and Time-Frequency Analysis

Parameters

Input: zip paths of each subject (separate by commas). e.g. '*\sub_01.zip,*\sub_02.zip'. EEG data will be loaded as EEG structure imported by EEGLAB. EEG.data should be channels×time points OR channels×time points×epochs.

Output: output path.

combs_project_id: project ID (only print for WeBrain).

epochLenth: length of small epochs to calculate power (no overlapped). Unit is second. Default is 1s. If epochLenth is negative, it means that if possible, data before event labels (eventlabel) will be used (no overlapped).

eventlabel: Event label which means specific event data. Default is empty. If it is empty, all data will be used to calculate indices. If eventlabel is not found in events, NO data will be epoched and calculated. If structure event (eventlabel) doesn't include duration, the duration will be equal to epochLenth.

seleChanns: number with indices of the selected channels (e.g. '[1:4,7:30]' or 'all').Default is 'all';

WaveletCycles: [real] indicates the number of cycles for the time-frequency decomposition {default: 0} if 0, use FFTs and Hanning window tapering. or [real positive scalar] Number of cycles in each Morlet wavelet, constant across frequencies. or [cycles cycles(2)] wavelet cycles increase with frequency starting at cycles(1) and, if cycles(2) > 1, increasing to cycles(2) at the upper frequency, or if cycles(2) = 0, same window size at all frequencies (similar to FFT if cycles(1) = 1) or if cycles(2) = 1, not increasing (same as giving only one value for 'cycles'). This corresponds to pure wavelet with the same number of cycles at each frequencies if 0 < cycles(2) < 1, linear variation in between pure wavelets (1) and FFT (0). The exact number of cycles at the highest frequency is indicated on the command line.

WaveletMethod: ['dftfilt2'|'dftfilt3'] Wavelet method/program to use. (default is: 'dftfilt3')

'dftfilt2' Morlet-variant or Hanning DFT (calls dftfilt2() to generate wavelets).

'dftfilt3' Morlet wavelet or Hanning DFT (exact Tallon Baudry). Calls dftfilt3().

ITCtype: ['coher'|'phasecoher'] Compute either linear coherence ('coher') or phase coherence ('phasecoher') also known as phase coupling factor' {default: 'phasecoher'}c.

ntimesout: Number of output times (int<frames-winsize). Enter a negative value [-S] to subsample original time by S. Default is 200. winsize = max(pow2(nextpow2(length(epoch))-3),4).

srate: sampling rate of EEG data. It can be automatically detected in EEG data. But for ASCII/Float .txt File or MATLAB .mat File, user should fill the sampling rate by hand. Default is '[]'.

Output:

EEG_results: results including power indices, mean indices across epochs and parameters.

EEG_results.type: type of results;

EEG_results.spectrum: (channels × nfreqs × timesout × epochs) matrix of frequency spectrum for each epochs. log10 logarithm transformed

EEG_results.ITC: (channels × nfreqs × timesout) matrix of complex inter-trial coherencies. ITC is complex.

EEG_results.ITCphase: (channels × nfreqs × timesout) matrix of ITC phase in deg phase(itc)*180/pi;

EEG_results.ITCmag: (channels × nfreqs × timesout) matrix of ITC magnitude, abs(itc);

EEG_results.spectrum_mean: (channels × nfreqs × timesout) matrix of mean time-frequency spectrum across epochs;

EEG_results.spectrum_sub_mean: (channels × nfreqs × timesout) mean time-frequency spectrum across epochs (baseline correction by subtraction);

EEG_results.spectrum_relative_mean: (channels × nfreqs × timesout) mean time-frequency spectrum across epochs (baseline correction by relative change %);

EEG_results.spectrum_zscore_mean: (channels × nfreqs × timesout) mean time-frequency spectrum across epochs (baseline correction by z-score);

EEG_results.spectrum_ratio_mean: (channels × nfreqs × timesout) mean time-frequency spectrum across epochs (baseline correction by power ratio);

EEG_results.parameter.eventlabel: an eventlabel which means good quality data;

EEG_results.parameter.selechanns: an array with selected channels;

EEG_results.parameter.epochLenth: length of small epochs. Unit is time point.

EEG_results.parameter.srate: sampling rate of EEG data.

EEG_results.parameter.chanlocs: locations of selected channels;

EEG_results.parameter.ref: EEG reference;

EEG_results.parameter.WaveletCycles: Wavelet Cycles;

EEG_results.parameter.WaveletMethod: Wavelet Method;

EEG_results.parameter.ITCtype: ITC type;
EEG_results.parameter.TaperingFunction: Tapering function;
EEG_results.parameter.DetrendStr: ['on'|'off'], Linearly detrend each data epoch;
EEG_results.parameter.spectrum_unit: spectrum unit is '10*log10(μ V^2/Hz)';
EEG_results.parameter.ntimesout: No. of output time points.

3.15 WB_EEG_calcMicrostate

WB_EEG_calcMicrostate is a tool to conduct microstate analysis for resting-state EEG data or ERP data. It is used to offer a sparse characterisation of the spatio-temporal features of large-scale brain network activity. It will calculate microstate indices across event-related trials (epochs) of time series. More details can be seen in the relative articles (Poulsen et al., 2018; Tait and Zhang, 2022; Ville, 2010).

Microstate analysis for resting-state EEG data or ERP data. Calculating microstate indices consists of

[1] Specific event data can be extracted according to the input ‘eventlabel’. If the input ‘eventlabel’ is empty, all data will be used. If applicable, EEG segments in bad block (label 9999, marked by wb_pipeline_EEG_Mark) will also be rejected automatically, and NOT used to calculate power indices.

[2] Specific event EEG signals will be divided into small epochs.

[3] EEG data of each epoch (default is 5s epoch) was subjected to microstate analysis to obtain the quantitative indices of the presence of microstates (1 × classes) in each EEG data.

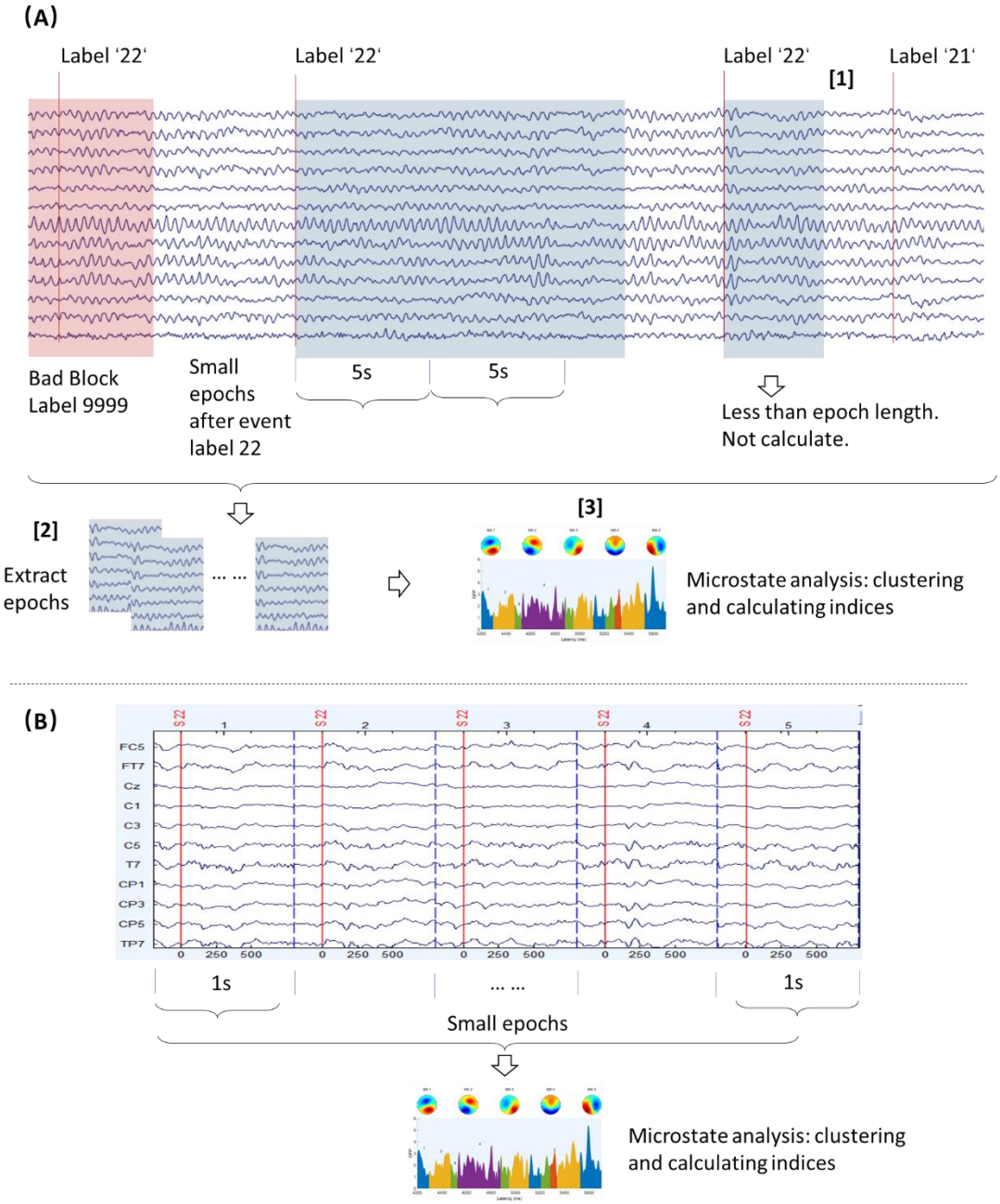

Fig. 18: Pipelines of microstate analysis.

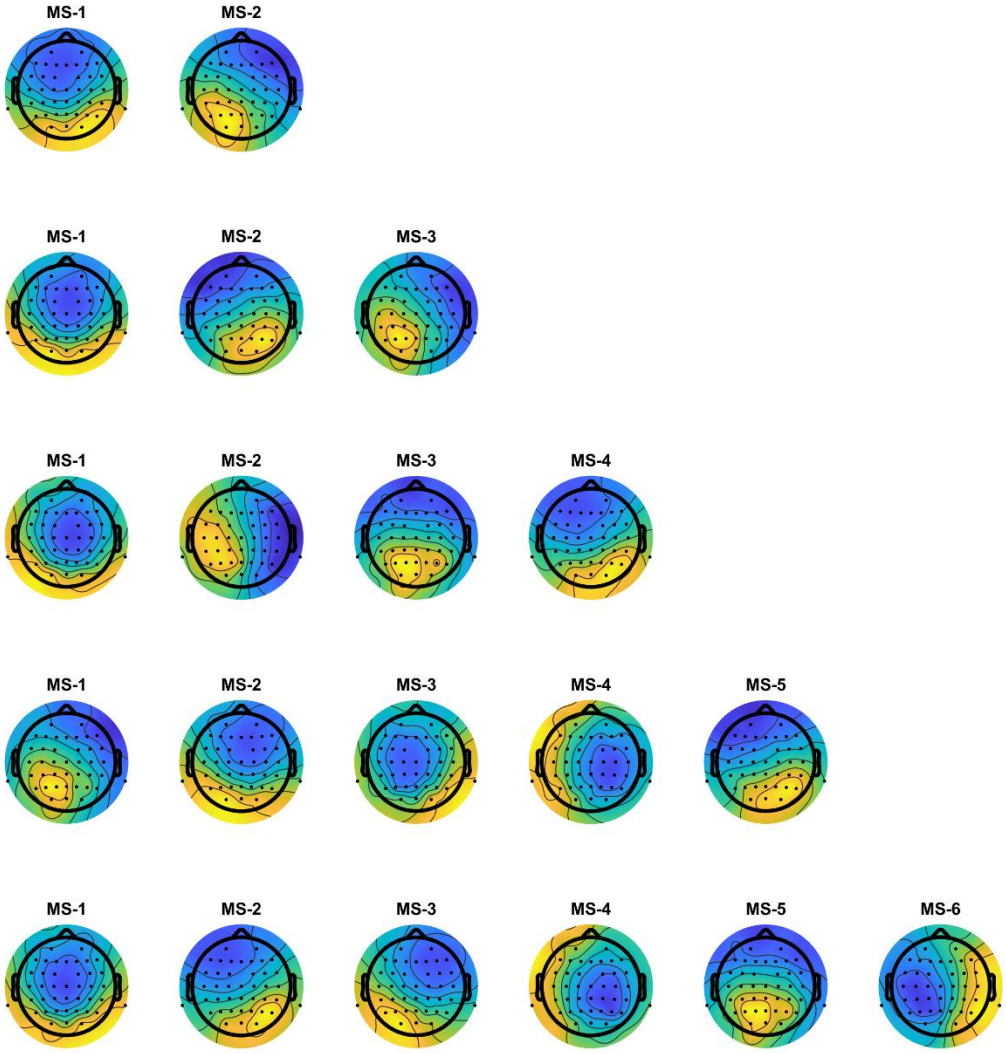

Fig. 19: Example results of microstates (rows: from 2 microstates to 6 microstates) for microstate analysis. Resting state eye-closed EEGs from 3 subjects were used.

References:

Poulsen A T, Pedroni A, Langer N, et al. Microstate EEGlab toolbox: An introductory guide. bioRxiv 289850[J]. 2018.

Tait, L. and J. Zhang (2022). "+microstate: A MATLAB toolbox for brain microstate analysis in sensor and cortical EEG/MEG." Neuroimage 258: 119346.

Ville (2010). "EEG microstate sequences in healthy humans at rest reveal scale-free dynamics." PNAS 107: 18179-18184.

Parameters

Input: zip paths of each subject (separate by commas). e.g. '*\sub_01.zip,*\sub_02.zip'. EEG data will be loaded as EEG structure imported using EEGLAB. EEG.data should be channels × time points OR channels × time points × epoches. EEG channel locations should be in EEG.chanlocs.

Output: output path.

combs_project_id: project ID (only print for WeBrain).

epochLenth: length of small epochs to calculate power (no overlapped). Unit is second. Default is 5s.If epochLenth is negative, it means that if possible, data before event labels (eventlabel) will be used (no overlapped).

eventlabel: Event label which means specific event data. Default is empty ('[]'). If it is empty, all data will be used to calculate indices. If eventlabel is not found in events, NO data will be epoched and calculated. If structure event (eventlabel) doesn't include duration, the duration will be equal to epochLenth.

seleChanns: string with indices of the selected channels (e.g. '[1:4,7:30]'), or 'all'.

flag_ref: 1: re-referencing to AVG to calculate Microstate; 0: use original reference.

GFPPeaks: Limit the selection of maps used for cluster to moments of global field power (GFP) peaks (1: yes; 0: no).

IgnorePolarity: Assign maps with inverted polarity to the same class (standard for resting EEG). resting-state: 1; ERP: 0.

Restarts: Number of times the k-means is restarted with a new random configuration. Default is 100.

MaxMaps: Max number of maps to use (default is 1000) -> Use a random subsample of the data to identify the clusters.

UseAAHC: 1: AAHC, 0: K-means. Use the AAHC algorithm instead of the k-means?

MinClasses: Minimal number of clusters to search for. Default is 2.

MaxClasses: Maximum number of clusters to search for. Default is 6.

Normalize: 1: Normalize EEG before clustering; 0: no. If use AAHC, normalization should be false, i.e. 0.

lambda: Penalty function for non-smoothness. Default is 1.

b: Window size for label smoothing (0 for none).Default is 30.

PeakFit: Whether to fit only the GFP peaks and interpolate in between (true, 1), or fit to the entire data (false, 0). Default is 1.

UseMeanTmpl: UseMeanTmpl -> 0 if the template from the data itself is to be used (default) -> 1 if a mean template is to be used.

srate: sampling rate of EEG data. It can be automatically detected in EEG data. But for ASCII/Float .txt File or MATLAB .mat File, user should fill the sampling rate by hand. Default is '[]'.

Output:

EEG_results: results including microstate stats and parameters.

EEG_results.type: result type, i.e. results of microstate.

EEG_results.MSStats: structure with quantitative indices of the presence of microstates (1 × classes) in each EEG data . The indices including:

MeanGFP: The strength of the average global activation during a given microstate k.

Duration: the Duration(k) is defined as the average duration of a given microstate (in second).

Occurrence: Occurrence(k) reflects the average number of times per second a microstate is dominant.

ExpVar: Relative variance explained by the model. It is a measure of how similar each EEG sample is to the microstate prototype it as been assigned to. The higher the GEV the better [0,1].

TotalTime: Total time in second for all microstates.

MeanDuation: Mean duration of all microstates.

MeanOccurrence: Mean occurrence of all microstates.

OrgTM: The original transition probabilities between microstates can be derived to quantify how frequently microstates of a certain class are followed by microstates of other classes. It not corrected for different number of occurrences of microstates (i.e. base rates of microstates).

ExpTM: The expected transition probabilities between microstates can be derived to quantify how frequently microstates of a certain class are followed by microstates of other classes. It adjusts for the base rate probabilities.

DeltaTM: Difference between the OrgTM and ExpTM, i.e. OrgTM - ExpTM.

DataSet: data set name.

Subject: subject name or ID.

Group: group label.

Condition: condition label.

filename: data filename.

Template: template used for clustering

SortInfo: microstates were sorted by "grand mean".

EEG_results.MSClass: cell structure with the presence of microstate classes in each EEG data. The values in the matrix are class label for each timepoint in each EEG data. The demension in "MSClass{1, nClasses}{1, subject k}" is timepoints × epochs.

EEG_results.results_bootstrap: Results of reordering microstate maps based on a mean template. It used to replicate the computation to report the optimal communality with cluster maps. Default number of bootstrap samples (repeats) is 10.

MeanComm: mean communality for each microstate class across all repeats (first nclass rows).

GrandMeanComm: grand mean communality across all MSclasses repeats.

idxmax: Optimal(maximal) communality with n maps/classes.

IgnorePolarity: Ignore the polarity of the maps to be sorted.1: yes; 0: no.

LearningsetSize:Percent samples used for the learning set. Default is 50.

MinClasses: min nClasses.

MaxClasses: max nClasses.

EEG_results.EpochData: quantitative indices of the presence of microstates (1 × classes) in each EEG data epochs.

Duration: the Duration(k, epoch) is defined as the average duration of a given microstate (in second) in each epoch.

Occurrence: Occurrence(k, epoch) reflects the average number of times per second a microstate is dominant in each epoch.

Contribution: Contribution(k, epoch) reflects the percentage of a microstate in each epoch.

EEG_results.parameters: parameters used to calculate microstate indices.

EEG_results.parameters.epochLenth: epoch length;

EEG_results.parameters.eventlabel: event label;

EEG_results.parameters.selechanns: selected channels;

EEG_results.parameters.ClustPars: parameters for clustering

EEG_results.parameters.FitPars: parameters for fitting;

EEG_results.parameters.chanlocs: channel locations of selected channels;

EEG_results.parameters.ref: EEG reference.

4. Copyright:

5. Acknowledgement

1. We thanks the REST, NIT, SPM, BCT, ADJUST, PREP, Clean_rawdata, Automagic, sLORETA, Fieldtrip, Microstates1.2 and EEGLAB toolboxes.
2. We thanks MATLAB and Docker.
3. Original author of this user guide is Li Dong.

Tool contributors:

WeBrain EEG tools	Supported methods	Integrated toolkits and their authors	Developers
WB_EEG_REST (v1.0)	REST Re-referencing	-	Li Dong
WB_EEG_Mark (v1.0)	Bad Block Marking	-	Li Dong
WB_EEG_runICA (v1.0)	ICA Running	EEGLAB (v14_1_0b, Arnaud Delorme and Scott Makeig)	Li Dong
WB_EEG_QA (v1.0)	Quality Assessment	-	Li Dong
WB_EEG_prepro (v1.0)	Preprocessing of Continuous EEG Raw Data (with the Reference of Single Point, Average or Linked Mastoids)	ADJUST (v1.1.1, Marco Buiatti); MARA (Irene Winkler and Eric Waldburger); clean_rawdata(0.32, Christian Kothe); inexact_alm_rpca (Minming Chen); EEGLAB (v14_1_0b, Arnaud Delorme and Scott Makeig)	Li Dong
WB_EEG_prepro_cm (v1.0)	Preprocessing of Continuous EEG Raw Data (with the Reference of Ipsilateral Mastoid or Contralateral Mastoid)	ADJUST (v1.1.1, Marco Buiatti); MARA (Irene Winkler and Eric Waldburger); clean_rawdata(0.32, Christian Kothe); inexact_alm_rpca (Minming Chen); EEGLAB (v14_1_0b, Arnaud Delorme and Scott Makeig)	Li Dong
WB_EEG_CalcPower (v1.0)	Power Spectrum Analysis	EEGLAB (v14_1_0b, Arnaud Delorme and Scott Makeig)	Li Dong
WB_EEG_CalcERP (v1.0)	Event-Related Potential Analysis	-	Li Dong
WB_EEG_CalcNetwork (v1.0)	EEG Network Calculation	-	Li Dong

WB_EEG_CalcNetMeasures (v1.0)	Network Topology Analysis	Brain Connectivity Toolbox (v2017_01_15 Olaf Sporns)	Li Dong
WB_EEG_CalcLeadfield_standardBEM (v1.0)	Generation of Conduction Model of Real Head	Fieldtrip (v20181025, Robert Oostenveld and Jan-Mathijs Schoffelen)	Li Dong
WB_EEG_sourceimage (v1.0)	EEG Source Imaging	Regularization functions (Per Christian Hansen and Michael Jacobsen) Fieldtrip (v20181025, Robert Oostenveld and Jan-Mathijs Schoffelen)	Li Dong
WB_EEG_calcLZC (v1.0)	Calculate Lempel-Ziv Complexity	-	Li Dong
WB_EEG_timefreq (v1.0)	Time-frequency Analysis	EEGLAB (v14_1_0b, Arnaud Delorme and Scott Makeig)	Li Dong
WB_EEG_calcMicrostate (v1.0)	Microstate Analysis	Microstates1.2 (Thomas Koenig)	Li Dong

6. References

Bai, Y., Liang, Z., Li, X., 2015. A permutation Lempel-Ziv complexity measure for EEG analysis. Biomedical Signal Processing and Control. 19**,** 102-114.

Bigdely-Shamlo, N., et al., 2015. The PREP pipeline: standardized preprocessing for large-scale EEG analysis. Frontiers in Neuroinformatics. 9.

Bob, P., et al., 2008. EEG phase synchronization in patients with paranoid schizophrenia. Neurosci Lett. 447**,** 73-7.

Chen, A.C., et al., 2008. EEG default mode network in the human brain: spectral regional field powers. Neuroimage. 41**,** 561-74.

Dale, A.M., et al., 2000. Dynamic statistical parametric mapping: combining fMRI and MEG for high-resolution imaging of cortical activity. Neuron. 26**,** 55-67.

Dong, L., et al., 2017. MATLAB Toolboxes for Reference Electrode Standardization Technique (REST) of Scalp EEG. Front Neurosci. 11.

Dong, L., et al., 2021. Reference Electrode Standardization Interpolation Technique (RESIT): A Novel Interpolation Method for Scalp EEG. Brain Topogr.

Edagawa, K., Kawasaki, M., 2017. Beta phase synchronization in the frontal-temporal-cerebellar network during auditory-to-motor rhythm learning. Scientific Reports. 7.

Hu, L., Zhang, Z., 2019. EEG Signal Processing and Feature Extraction, Vol., Springer.

Jobert, M., et al., 2013. Guidelines for the recording and evaluation of pharmaco-sleep studies in man: the International Pharmaco-EEG Society (IPEG). Neuropsychobiology. 67**,** 127-67.

Kashefpoor, M., Rabbani, H., Barekatain, M., 2016. Automatic Diagnosis of Mild Cognitive Impairment Using Electroencephalogram Spectral Features. J Med Signals Sens. 6**,** 25-32.

Kayser, J., Tenke, C.E., 2010. In search of the Rosetta Stone for scalp EEG: converging on reference-free techniques. Clin Neurophysiol. 121**,** 1973-5.

Lee, Y.Y., Hsieh, S., 2014. Classifying different emotional states by means of EEG-based functional connectivity patterns. PLoS One. 9**,** e95415.

Lei, X., Hu, J., Yao, D., 2012. Incorporating FMRI functional networks in EEG source imaging: a Bayesian model comparison approach. Brain Topogr. 25**,** 27-38.

Lempel, A., Ziv, J.J.I.T.o.i.t., 1976. On the complexity of finite sequences. 22**,** 75-81.

Lin, Z., Chen, M., Ma, Y., 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055.

Malver, L.P., et al., 2014. Electroencephalography and analgesics. Br J Clin Pharmacol. 77**,** 72-95.

Mattout, J., Henson, R.N., Friston, K.J., 2007. Canonical source reconstruction for MEG. Comput Intell Neurosci. 67613.

Mognon, A., et al., 2011. ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features. Psychophysiology. 48**,** 229-240.

Mullen, T., et al., 2013. Real-time modeling and 3D visualization of source dynamics and connectivity using wearable EEG. In: 2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC). Vol., ed.^eds. IEEE, pp. 2184-2187.

Nuwer, M.R., et al., 1994. IFCN Guidelines for Topographic and Frequency-Analysis of EEGs and EPs - Report of an IFCN Committee. Electroencephalography and Clinical Neurophysiology. 91**,** 1-5.

Pascual-Marqui, R.D., 2002. Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details. Methods Find Exp Clin Pharmacol. 24 Suppl D**,** 5-12.

Perrin, F., et al., 1989. Spherical splines for scalp potential and current density mapping. Electroencephalogr Clin Neurophysiol. 72**,** 184-7.

Poulsen, A.T., et al., 2018. Microstate EEGlab toolbox: An introductory guide. bioRxiv. 289850.

Rubinov, M., Sporns, O., 2010. Complex network measures of brain connectivity: uses and interpretations. Neuroimage. 52**,** 1059-69.

Snaedal, J., et al., 2010. The use of EEG in Alzheimer's disease, with and without scopolamine - A pilot study. Clinical Neurophysiology. 121**,** 836-841.

Tait, L., Zhang, J., 2022. +microstate: A MATLAB toolbox for brain microstate analysis in sensor and cortical EEG/MEG. Neuroimage. 258**,** 119346.

Thatcher, R.W., North, D., Biver, C., 2005. EEG and intelligence: Relations between EEG coherence, EEG phase delay and power. Clinical Neurophysiology. 116**,** 2129-2141.

Tzourio-Mazoyer, N., et al., 2002. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage. 15**,** 273-89.

Ville, 2010. EEG microstate sequences in healthy humans at rest reveal scale-free dynamics. PNAS. 107**,** 18179-18184.

Winkler, I., Haufe, S., Tangermann, M., 2011. Automatic classification of artifactual ICA-components for artifact removal in EEG signals. Behav Brain Funct. 7**,** 30.

Xu, P., et al., 2014. Differentiating between psychogenic nonepileptic seizures and epilepsy based on common spatial pattern of weighted EEG resting networks. IEEE Trans Biomed Eng. 61**,** 1747-55.

Yao, D., 2000. High-resolution EEG mappings: a spherical harmonic spectra theory and simulation results. Clin Neurophysiol. 111**,** 81-92.

Yao, D., 2001. A method to standardize a reference of scalp EEG recordings to a point at infinity. Physiol Meas. 22**,** 693-711.

Yao, D., et al., 2004. High-resolution electroencephalogram (EEG) mapping: scalp charge layer. Phys Med Biol. 49**,** 5073-86.

Yao, D., et al., 2005. A comparative study of different references for EEG spectral mapping: the issue of the neutral reference and the use of the infinity reference.

Physiol Meas. 26**,** 173-84.