NEAR: User Manual

November 5, 2021

This document contains the technical details of each of the user-defined parameters to be defined for NEAR processing.

1 List of Parameters

1.1 Basic Parameters

The basic ones are just to enable and disable different steps in processing pipelines. For example, **isLPF**. See table 1 for the list of basic parameters.

1.2 Advanced Parameters

The advanced ones are detailed parameters for each of the enabled preprocessing steps. For example, if the user has enabled lowpass filtering by setting **isLPF** to 1, then it is important to define the relevant parameter **lpc** with a value (e.g., 40 Hz). See table 2 for the list of basic parameters.

2 Tutorial Script

To get yourself familiarized with different parameters, which is crucial to extend NEAR pipeline to your applications, we have created a tutorial script named *NEAR_Pipeline_Tutorial_v1_0.m*. The file can be found in this repository and it can be executed block-by-block for a sample subject.

3 Single Subject Processing

For performing NEAR preprocessing for a single subject, the users can open the file *NEAR_singlesubject_processing.m*. The parameters should be set, and the function run_NEAR.m will process the given data using the parameters. For the downstream analysis, such as ERP analysis, the users can write their own code at the end of the script.

4 Batch Processing

If the users would like to apply NEAR preprocessing for a batch of EEG files, it can be done so by using the file *NEAR_batch_processing.m*. Just like the single subject processing, the parameters can be set quickly, and for each file if a downstream analysis is preferred, appropriate scripting can be done within the for loop.

5 Report and Saving Functionalities of NEAR

By settng the parameters isReport and isSave to 1, the users can have their processed datasets along with a comprehensive summary of the preprocessing steps performed. The files can be found in the folders "NEAR_Processed" and "NEAR_Reports" for processed .set files and report files respectively.

ID	Parameter Name	Parameter Description	Possible Values
1	isLPF	is Low Pass Filter (to be applied)?	0 if you do not want to apply a Low Pass Filter; 1 otherwise.
2	isHPF	is High Pass Filter (to be applied)?	0 if you do not want to apply a High Pass Filter; 1 otherwise.
3	isSegt	is Segmentation (to be applied)?	0 if you wish to segment the data; 1 otherwise.
4	isERP	is your data an ERP?	1 if you want to epoch the data; 0 otherwise.
5	isBadCh	is Bad Channel (to be detected)?	0 if you do not want to detect bad channels; 1 otherwise.
6	isVisIns	is Visual Inspection? (for bad channels)	0 if you do not want to see channel-wise statistics; 1 otherwise.
7	isBadSeg	is Bad Segmentation (to be detected)?	0 if you do not want to detect bad segments; 1 otherwise.
8	isInterp	is Interpolation (to be performed)?	0 if you do not want to perform interpolation; 1 otherwise.
9	isAvg	is average re-referencing (to be performed)?	0 if you do not want to perform re-referencing; 1 otherwise.
10	isReport	is Report function enabled?	1 if you wish to save a report file; 0 otherwise.
11	isSave	is Save function enabled?	1 if you wish to save the processed data; 0 otherwise.

Table 1: List of Basic Parameters

6 Hyperparameters Tuning

There are 2 main hyperparameters to be calibrated on the user's data (by conventional 70-30 training-testing split). They are

1. LOF Threshold (to be used in NEAR Bad Channel Detection)
2. ASR Parameter (to be used in NEAR Bad Segments Correction)

In addition to NEAR custom scripts, we provide basic scripts found in "TuneLOF" and "TuneASR" folders in this repository. While TuneLOF uses the most common metric F1 Score, TuneASR requires users to define a measure of interest namely ERP-SNR, or FTR described in our manuscript.

Note: For TuneLOF, we also provide a sample EEG file along with labelled bad channels found in the openNeuro website: https://openneuro.org/datasets/ds002034/versions/1.0.1

ID	Parameter Name	Parameter Description	Possible Example Values
1	lpc	Low Pass Cut-off	40 Hz
2a	hptf	High Pass Transition Frequency	[0.25 0.75] Hz
2b	hpc	High Pass Cut-off Frequency	0.1 Hz
3a	sname	Segmentation File Name (in .xlsx)	segt_visual_attention.xlsx
3b	sloc	Segmentation File Location	Absolute Path of .xlsx file
3c	look_thr	Looking Times Thresold	5000 ms
4a	erp_event_markers	Event Markers	'eyes open', 'eyes close'
4b	erp_epoch_duration	Epoch Duration in seconds	[0 1.2] s
4c	erp_remove_baseline	Remove baseline	1 if you want to; 0 otherwise
4d	baseline_window	Baseline Window in ms	[0 200] ms
5a	isFlat	is Flat-line channels (to be detected)?	0 or 1
5b	flatWin	Tolerance window length	5 s
5c	isLOF	is LOF algorithm (to be applied)?	0 or 1
5d	dist_metric	Distance Metric for LOF algorithm	'euclidean' or 'seuclidean'
5e	thresh_lof	Threshold for LOF algorithm	1.5
5e	isAdapt	thresh_lof is incremented by 1 if the total % of bad channels exceed this limit	10%
5f	isPeriodogram	is Periodogram analysis applied?	0 or 1
5g	frange	Frequency Range to be considered	[1 10] Hz (motion artifacts)
5h	winsize	window length for periodogram	1 s
5i	winov	window overlap length	0.66 s
5j	pthresh	Threshold for Periodogram analysis (in terms of SD)	2
6a	rej_cutoff	ASR Parameter (k)	20
6b	rej_mode	ASR Mode: Correction or Removal	'off' for correction or 'on' for removal
6c	add_reject	Additional Rejection by ASR?	'off' or 'on'
7	reref	Re-reference electrode (in case isAvg = 1)	[10] or 'E129'

Table 2: List of Advanced Parameters