

LBCN_preprocessing_pipeline_Su

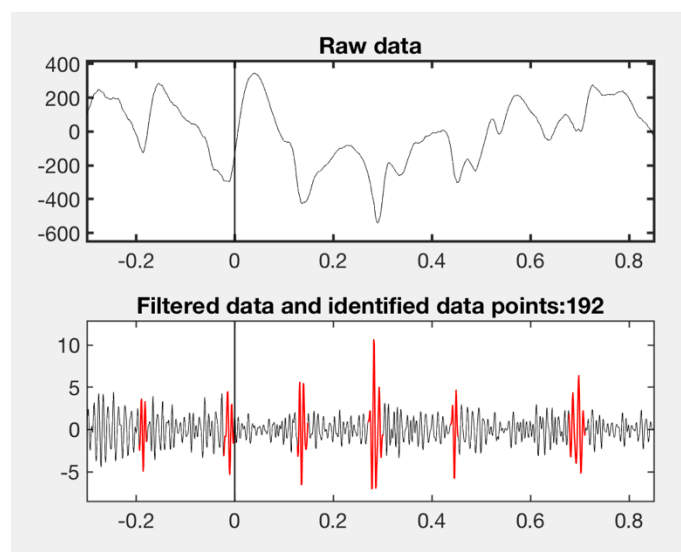
Modified from original SPM based preprocessing pipeline.

Overall, the modified pipeline reads DC trigger channels and pulses to skip before a block automatically. It should be able to identify what kind of task the current input is and load the corresponding pre-defined configurations. Currently it works for race, VTC and emotional face tasks (means that no additional customizations are needed). The modified pipeline also has higher tolerance to duplicated channel labels.

New features:

For the standard pipeline, two steps of artifact/pathological event detection were integrated:

1. The LBCN_findPChan function finds the spiky and HFO active channels based on HFO detection algorithm. The pathological channels will be excluded when calculating the common average. The indices of HFO/spikes will be stored and the user can choose to set those time points to NAN.
2. The LBCN_filt_bad_trial function does addition epoch-by-epoch data quality inspection. The function identifies epochs with very large/small variance (bad) and time stamps with spiky activities (spkind) in each epoch. The “bad” epochs will be excluded entirely, and the “spiky” datapoints will be set to NAN and be interpolated while plotting the final results. If half of the trials in a channel were assigned as “bad”, a window will pop out and the user could quickly review the rejected trials. After viewer the trials, the user can choose to keep them as good signal, abandon as bad signal, manually put the bad trials based on visual inspection, or quit and adjust the threshold. **In the HFB plot/subfunctions/Atf Detect folder there is a demo script and sample data for illustration purpose.**



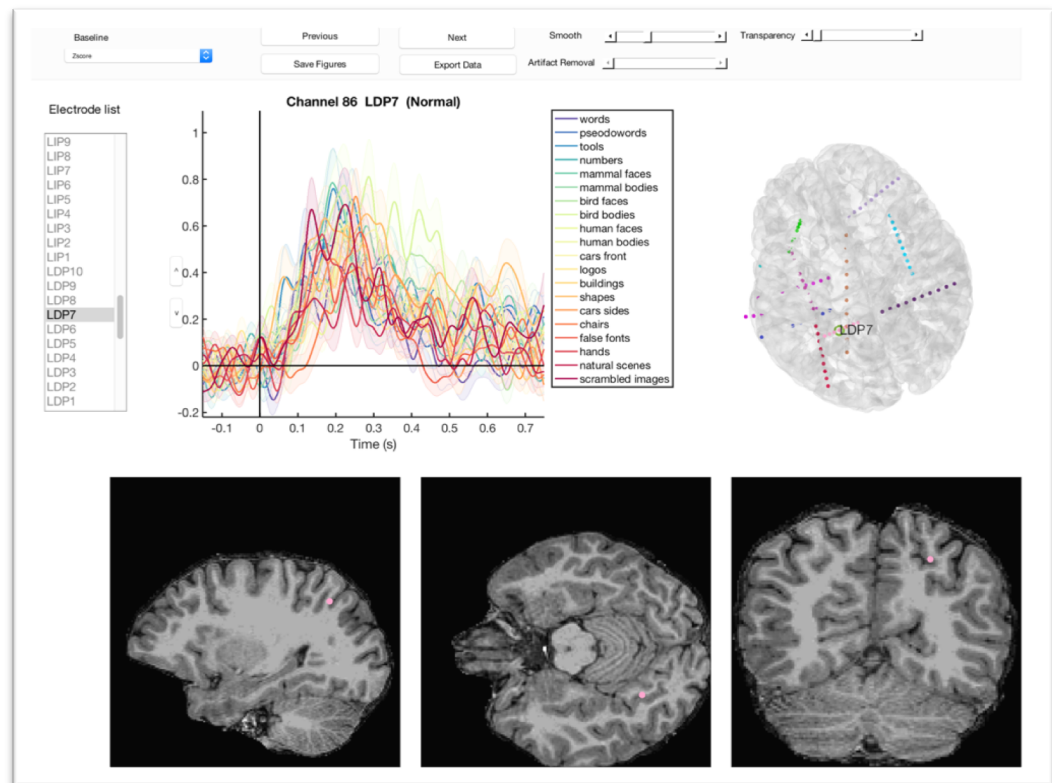
Other changes including the smoothing, legend fonts, for VTC task it plots the combined conditions instead of each condition (optional). After the calculation is finished, if the user has chosen not to save, the results will be plotted on fly with pause.

For the new pipeline (which also integrated the artifact removal steps), the user can choose whether to use FFT or wavelet decomposition for HFB computation. The two main functions (LBCN_plot_HFB.m and compute_HFB.m) read both SPM format files and simple data matrix (epoched raw data segments).

In particular, after epoching, the user can choose to run “ LBCN_plot_HFB “ and select eventSODATA files generated by SPM and review the result. Alternatively, the user can choose “Epoched_data_TASKNAME.m” file (generated before the execution this function if this pipeline is chosen) which stores the epoched data and some of the parameters. Or, the user can choose “Epoched_HFB.m” which stores the HFB signals (generated by this function before final plotting). Note that the function does not rename new files so if the use re-analyzed the data, the result files will overwrite the old results.

Finally at the plotting stage, the user can choose whether to review the plots using a viewer (GUI). The viewer also automatically loads the corresponding freesurfer image files and plots the electrode locations when navigating between channels -- if it is able to locate the folder in the subjects' directory. If it failed to find the freesurfer files, the user can manually select the freesurfer folder. If the folder is on server, the program will first copy the necessary files to the local directory where the subject's EEG data is stored. If no image folder is chosen, it will plot

the signals only. Currently the viewer reads the new pipeline's result data structure, but it should be compatible with other pipelines in future.



To run the pipeline

1. Standard pipeline:

```
LBCN_preprocessing_new([],[],[],task,1)
```

Inputs are the same (edf and sodata).

If save, find the last line

```
LBCN_plot_averaged_signal_epochs2(final,[],plot_cond,[],0,task,[],[],EXCLUDE,  
BCH,REMOVEID);
```

And put 1.

2. New pipeline:

`LBCN_preprocessing_new`

Or to specify the task,

`LBCN_preprocessing_new([],[],[],task).`

Inputs are the same (edf and sodata).

If save, go to line 176:

`save_plot = 0;`

Change it to 1. This is only necessary when you choose not to use the GUI for final plotting, with the GUI there is an option to save. To run without a GUI, open the `LBCN_plot_HFB.m` and uncomment line 272, and comment the rest.

You can also run:

`LBCN_plot_HFB`

And choose the eventSODATA files (if the data has been epoched only), the `Epoched_data_TASKNAME` file (if you want to re-compute the HFB using epoched data, and you've run this function for once), or the `Epoched_HFB` (if you want to review the results only).

Things are still under testing, everything you do with this pipeline will help me with the debugging so thanks in advance!

Su Liu @ LBCN, 2018