High-frequency Oscillations Identification and Imaging

About the project

This document describes software codes and sample data (clinical and simulation), which have been developed under support from the National Institutes of Health (NIH) via grants NS096761, EB021027, MH114233, AT009263, and EB029354 awarded to Dr. Bin He. The sample human data, which is deidentified, were collected as part of the NIH funded research at Mayo Clinic, Rochester, Minnesota, overseen by Dr. Gregory A. Worrell.

The source codes and testing data are provided as service to the scientific community and may be used for any non-commercial purposes under a CC-BY-NC-SA-4.0 license. Users should use the codes or data at their own risks. A copy of the CC-BY-NC-SA-4.0 license is provided along with this program. If not, see https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.

If this project or part of the codes or the data, provided here, helps your research or project, please cite the following paper in your publications or presentations.

Z. Cai et al., "Noninvasive High-frequency Oscillations Riding Spikes Delineates Epileptogenic Sources." Proc. Natl. Acad. Sci., PAGE NUMBER, 2021 (DOI).

Codes and data

This folder contains the codes and sample data used to analyze EEG recordings for high-frequency oscillations (HFOs) identification and imaging. The main framework is distributed into two workflows, the identification part and the imaging part. The included data consists of one de-identified data of a patient, containing inter-ictal spikes recoded in high-density (75 channels) EEG and brain geometric model generated from presurgical MRI, as well as two sets of simulation data, containing inter-ictal spikes embedded in different noise background.

The two workflows are implemented in two Matlab script files, namely, "main_hfoDetection.m" and "main_hfoImaging.m". The first script is to run identification of HFOs in the scalp EEG recordings (clinical or simulation), and the second script is to estimate the cortical source distribution given the identification results and the geometric head model. It is also good to note that the intermediate results have also been provided, which were generated by running the same codes here. Thus, it is possible to perform the imaging procedure alone without running the identification process first.

To run the codes, Matlab (The MathWorks, Inc., MA, USA) has to be installed on the computer. The current codes have been developed with Matlab 2018a and tested also with Matlab 2017b, 2019b, and 2020b. In addition, some of the codes require functions from third-party toolboxes for signal processing and visualization purposes, as listed in the following.

(i) EEGLab is used for visualization of the topo plots, which can be downloaded and deployed from https://sccn.ucsd.edu/eeglab/index.php. The version tested is v13.6.5b and v14.1.1b (recommended) and probably compatible with versions above.

(ii) jLab is used for time-frequency visualization, which can be downloaded and deployed from https://github.com/jonathanlilly/jLab. The version of jLab tested is v1.6.2.

(iii) regtools is used for regularization of the source imaging process, which can be downloaded and deployed from http://www2.compute.dtu.dk/~pcha/Regutools/. The version of Regularization Tools tested is v4.1.

(iv) SSD, Spatio-Spectral Decomposition, is used for extraction and denoising of HFO activities, which can be downloaded from https://github.com/svendaehne/matlab_SSD/.

(v) Optimal kmeans, kmeansElbow, is used for selecting optimal kmeans clusters for HFOs identification/extraction. This function was implemented by Sebastien De Landtsheer, https://it.mathworks.com/matlabcentral/fileexchange/65823-kmeans_opt.

(vi) findpeaks, built-in function in Matlab, with minor alterations for extra outputs, is used for locating the peaks in a signal.

(vii) tightSubplot, is used for setting axes with adjustable margins and gaps. This function was implemented by Pekka Kumpulainen and is available from the following link: https://www.mathworks.com/matlabcentral/fileexchange/27991-tight_subplot-nh-nw-gap-marg_h-marg_w.

For the codes to correctly locate the third-party toolboxes and functions, the corresponding toolbox codes could be deployed into the "toolbox" folder and separated into the subfolders, namely, "eeglab", "jLab", "regtools", and "thirdParty". The functions kmeansElbow, findpeaks, and tightSubplot with minor modifications has been provided in the "thirdParty" folder already. These toolboxes will be loaded automatically when running the main workflow codes. As an alternative, the user can also put the toolboxes or codes at other places, as long as the codes could be searched and called by Matlab. It should also be noted that if there already exist other versions of the same toolboxes in the Matlab search path, it is recommended to remove the duplicated ones before running the codes to avoid potential conflicts due to overwritten directories and compatibility.

Next, we will quickly go through the folders and codes and how to run the provided examples.

Folders

1. **data**
   This folder contains the sample data for running the examples and the corresponding results.
   1.1. imaging: This subfolder contains the imaging results generated by the source imaging procedure "main_hfoImaging.m". Example results are also provided in this folder.
   1.2. process: This subfolder contains the HFOs identification results generated by the HFOs identification procedure "main_hfoDetection.m". Example results are also provided in this folder.

1.3. raw: This subfolder contains the raw clinical and simulation dataset. The clinical data, which is deidentified, consists of clinically recorded inter-ictal spikes in high-density (75 channels) EEG and an individual brain geometric model in the format of leadfiled matrix. The simulation data contains two sets of inter-ictal spikes, including simulated true HFOs and spurious activities, such as sharp artifacts, in high-density EEG. Besides, the simulation ground truth, including the temporal and spatial (channel) locations where the true and false HFOs were added, is also provided in the simulation data file. In addition, a generic electrode location file (digitizer) is included for visualization purposes.

2. **hfo_common**
   This folder contains supporting functions that are commonly used by the identification and imaging procedures.

3. **hfo_identify**
   This folder contains functions for performing HFOs identification procedure, called by the script "main_hfoDetection.m".

4. **hfo_imaging**
   This folder contains functions for performing HFOs imaging procedure, called by the script "main_hfoImaging.m".

5. **hfo_visual**
   This folder contains functions that are used for visualization of the results.

6. **toolbox**
   This folder contains the main third-party toolboxes that are employed by the codes. The folder includes four subfolders for deploying the required toolboxes and functions, as listed previously (last section of **Codes and data**). Note that if there already exist other versions of the same toolboxes or functions in the Matlab path, it is recommended to remove the duplicated ones before running the codes to avoid potential conflicts due to overwritten directories and compatibility.

Procedures and examples

1. **Identification procedure**
   Run "main_hfoDetection.m" to quickly go through the whole identification workflow. As an alternative, this script can also be run by calling the individual sections step by step. The general process is briefly described below.
   1.1. The workspace is prepared and the data, either clinical or simulation, is loaded.
   1.2. Initial detection is performed by "main_hfoDetection_detection.m".
       1.2.1. A pool of HFO candidates is detected by a baseline detector.
       1.2.2. Several criteria are adopted to screen the candidate events.
       1.2.3. The remaining events and the corresponding information (e.g., temporal and spatial locations of each event) are saved in "3_HFOEvents_compact.mat".
   1.3. Features are extracted for each detected event using "main_hfoDetection_feature.m".

1.4. The events are clustered into several groups by "main_hfoDetection_clustering.m".

1.5.a. Upon the inspection of clustered activities, the putative HFOs riding on spikes could be identified and saved using "main_hfoDetection_saveHFO.m". The results are updated to the file "main_hfoDetection_feature.m".

1.5.b. For the simulation dataset, the identified HFOs are evaluated against the simulation ground truth using "main_hfoDetection_simEval.m". The results are updated to the file "main_hfoDetection_feature.m".

1.6. All corresponding results are saved. A summary of the detected events is saved in the file "3_HFOEvents_compact.mat". Besides, for the clinical dataset, the HFOs activity is extracted and saved in "HFOResults.mat" in the "./data/process" folder for later source imaging process.

2. Source imaging procedure

Run "main_hfoImaging.m" to quickly go through the whole source imaging workflow. As an alternative, this script can also be run by calling the individual sections step by step. The general process is briefly described below.

2.1. The workspace is prepared and the extracted signals, HFOs and spikes, are loaded.

2.2. The supporting information, including the leadfield, the electrode digitizer, and the imaging parameters, is setup in "setupImagingData.m".

2.3. The general source imaging process is implemented in an object-oriented paradigm, which assembles the basic process of source imaging in the class "jc_esi.m".

2.4. The object "jc_esi" is first initialized by setting up the scalp EEG measurements and the individual head model, contained in the leadfiled file.

2.5. The source imaging estimation is solved by sLORETA, standardized low-resolution brain electromagnetic tomography, which is implemented in the file "jc_sl_sLORETA.m".

2.6. The source imaging results are visualized and saved in the "./data/imaging" folder.

2.7. These steps could be run through for both extracted HFOs and averaged spikes.

This is the approach of High-frequency Oscillations Identification and Imaging in a nutshell.

Zhengxiang Cai

04/03/2021