CNELab: A fast and user-friendly toolbox to review and process neurophysiological signal

Tianxiao Jiang, Nuri F. Ince* *Senior IEEE Member*

Abstract— **This paper presents a cross-platform Matlab graphical user interface (GUI) software named as CNELab to review and process neurophysiological signal such as electroencephalogram (EEG), electrocorticogram (ECoG), magnetoencephalography (MEG) and local field potential (LFP) recordings. Compared to the other data visualization toolboxes, this software aims to provide fast data visualization and online processing with support for long-term recordings with arbitrary size. Professional tools that are frequently used in commercial software are included such as data scrolling and navigating, data preprocessing (zooming, multithreaded filtering, masking), data annotating and event file importing/exporting, montage file importing, synchronized video playback. In addition, some frequently used data analysis methods such as power spectrum density (PSD) estimation and time-frequency map analysis are also embedded in the software. CNELab is freely available and redistributed under the GNU General Public License V3.0 (GPLv3) for non-commercial use and open source development (*http://incelab.bme.uh.edu/tools*).**

I. Introduction

Neurophysiologist's interpretation on visualized raw data is still critical for clinical diagnostics. Nowadays, it is more and more common to obtain long-term recordings from a large number of channels. However, it is not necessarily a trivial task for researchers to obtain a professional data view. Developing a visualizaton toolbox with common features such as events and montage from scratch can be tedious and error prone. Therefore, we propose our customized Matlab software-CNELab to facilitate and expedite this procedure.

Developed in the clinical neural engineering lab of University of Houston, CNELab has been proven to be useful to review the high-density ECoG and multi-track LFP recordings. It has also been used to generate raw data plots and automatize repetitive data analysis such as PSD or time-frequency maps for a number of publications [1]–[5]. After years of development, we have shaped CNELab into a general purpose GUI software that can be easily used to review any type of long-term recording. Compared to its predecessor–BioSigPlot [6], CNELab is more controllable and user-friendly with various extended features such as customized data visualization, event/montage file import/export, individual channel filtering and gain adjustment, etc. Data plotting and preprocessing in CNELab are redeveloped to be more efficient and cross-platform compatible. The new video feature in CNELab is also extended to be compatible with the new Matlab releases. As the internal structure of CNELab has been largely rewritten since the original modification from BioSigPlot, we decided to publish CNELab as a separate Matlab toolbox. Many useful features in BioSigPlot such as measurer and multi-datasets view mode have been retained in CNELab without detailed discussion in this paper.

In the next section, we listed several frequently used features of CNELab and briefly discussed the ideas behind them. After that, program specifications with prerequisites and installation procedures are provided. Finally, the conclusion and future development plan of CNELab are discussed in the last section.

II. Material and methods

A. Data structure

As CNELab is written in object oriented (OO) way, a separate class (*CommonDataStructure*) is dedicated to manage datasets importing and exporting. Third party Matlab toolboxes were used to support data formats such as edf and fif. In addition, *CommonDataStructure* accepts direct Matlab file (*.mat) importing with specific structures. The recommended format is the intrinsic data structure used within the class. However, a simple structure with only one field that represents the data matrix is also compatible.

Type *CommonDataStructure* in Matlab command line will generate an object with a full list of the fields including the data matrix (*Data*) and sampling rate (*DataInfo.SampleRate*). The sampling frequency can be modifed in CNELab via Settings » Sample Rate. The recommended way to import/export dataset with *CommonDataStructure* object is to use the predefined methods *load* and *save*. Currently only mat file saved in *CommonDataStructure (cds)* format can be used to concatenate multiple segments to review long-term recordings. In practice, similar to a list data structure, each *cds* file is designed to resemble a list node that contains fields which point to the previous and next *cds* files. By opening any of the *cds* file, CNELab is able to recognize all the others and dynamically load necessary ones into memory while navigating through the dataset. This implementation enables CNELab to review arbitrary large datasets that are stored in a list of files with fixed size. *multi_file_link_demo.m* provides an example to transform original mat files of long-term recordings into connected *cds* files to be explored in CNELab.

B. Preprocessing

Only a segment of data with limited size is preprocessed each time in CNELab. The maximum buffer size of the segment is 300 megabytes by default and can be further modified inside CNELab (Data » Buffer). While navigating

*Nuri F. Ince (corresponding author) is with the Department of Biomedical Engineering, University of Houston, Houston, TX, Tel.: 713-743-4461; Fax: 713-743-0226; Email: nfince@uh.edu through the dataset, data is dynamically loaded into memoery from the external files.

The most frequently used preprocessing method is digital filtering. CNELab offered two C++ mex file implementations of multithreaded filtering to be cross-platform compatible. (*WinMultiThreadedFilter* is written in Windows Threads for windows operating system while *UnixMultiThreadedFilter* used POSIX Threads to be compatible with Unix like operating system. Second-order Butterworth IIR filters are implemented with forward and backward filterings to achieve zero phase shift. A major advantage of CNELab's multithreaded filtering is to apply different filters on different channels at the same time. This is especially useful when the dataset includes multi-modality sensors such as EEG, EMG and EKG which need to be explored in separate subbands. To compromise the overhead of thread creation and the performance enhancement from parallel computing, the default thread number equals to the core number of the computer. Filtering controls are provided in the filter panel. Once enabled, one or more of the following filters can be combined:

- *Low*: cut-off frequency (Hz) for high-pass filter.
- *High*: cut-off frequency (Hz) for low-pass filter.
- *Notch*: interference frequency (Hz) for notch filter.
- *Custom*: customized filter designs loaded from external files in `CNELAB_ROOT▸db▸filters` .

Notch filter is primarily used to remove the power line noise. In addition, the harmonics can also be removed by enabling harmonic notch filters in Settings › Notch Filter › Harmonics. The stop band of each notch filter is set to be 2Hz.

CNELab also implements basic spatial filtering such as neighborhood interpolation and advanced spatial filterings such as principle component analysis (PCA, ⤯) and independent component analysis (ICA, ⊛). However, these features are under testing to be compatible with the newest CNELab release.

In CNELab, only selected channels (highlighted in magenta) are used for processing. Here is a list of the supported selection methods:

- Left click for single channel selection.
- ctrl or ⌘ left click to add selection. ⇧ left click to select all channels in between.
- ctrl+A to select all channels in the current dataset.
- No selection selects all channels in all displayed datasets.
- j and k navigates selection downwards and upwards respectively.
- ctrl or ⌘ left click on selected channel removes the selection.
- esc removes all selections.

C. Visualization

CNELab implements a number of optimizations to accelerate data plotting. While navigating or scrolling through the dataset, only the 'YData' properties of the line objects are updated. Based on the fact that the pixel number on the computer screen is usually insufficient to display every value of the signal especially when the window is large, the preprocessed data is subsampled before plotting. The display buffer size is limited to be 2M in default and can be further modified in Display › Buffer.

A rich set of tools are implemented in CNELab for customized data visualization and navigation controls (Fig. 1). Here is a list of commonly used ones:

- ∩ switches to the next dataset in single dataset mode. 1|2 displays multiple datasets horizontally. ½ displays multiple datasets vertically.
- *Chan* sets the displayed channel number in each window. *Win* sets the window length in seconds.
- *Sen* sets the scale of selected channels.
- 🔍 (ctrl or ⌘ + =) increases the gain on selected channels.
- 🔍 (ctrl or ⌘ + -) decreases the gain on selected channels.
- 🔍 adjusts the gain automatically on selected channels.
- 👁 ,👁 masks and unmasks selected channels from both processing and visualization.
- Settings › Color › Canvas sets the background color (ctrl or ⌘ + B).
- Settings › Color › Lines sets the selected lines color (ctrl or ⌘ + L).

There are generally two ways to navigate through data in CNELab. The traditional navigation controls are provided in the navigation panel at the bottom of the window. It includes jumping to a specific time point (s), jumping backward or forward (‹, ›), jumping backward or forward by window length («, ») and jumping to the start or the end of the dataset (⏮, ⏭). In addition, CNELab implements event navigation as follows:

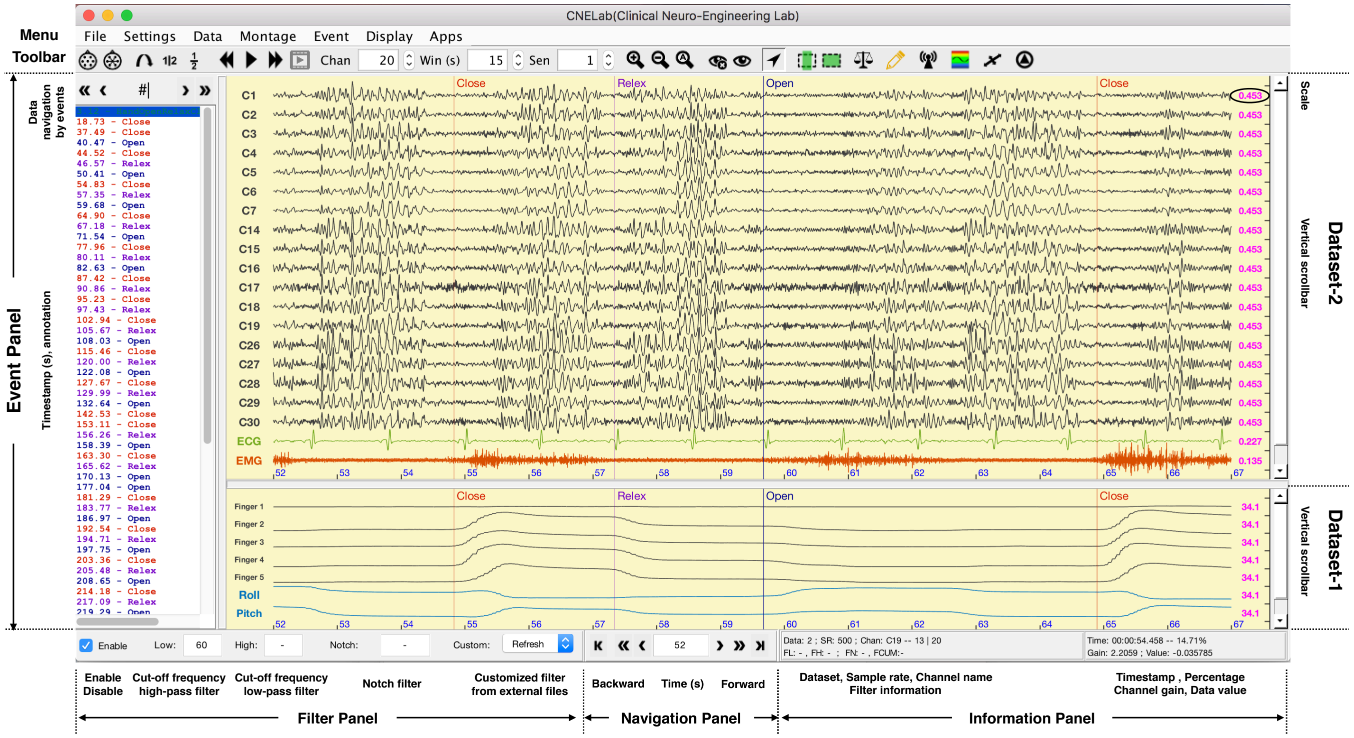
- Left click on desired event in the event panel triggers data navigation.
- < navigates to the previous event in the event list while > navigates to the next one.
- << navigates to the previous event of the same text while >> navigates to the next one.

Data plots in CNELab can be copied into a new Matlab figure for exporting or further modification via File › Save As › Figure (ctrl or ⌘ + P).

Two activex controls (Windows Media Player and VLC player) are implemented to review video with the data in Windows system. CNELab uses VLC player (if installed) in default to achieve better video format compatibility. Navigating through the dataset triggers synchronized video positioning and vice versa. Video starting time relative to the dataset can be modified (Settings › Video › StartEnd) to synchronize the video with the data.

D. Montage and events

Montage is a useful to select channels and derive bipolar or customized projections for visualization. CNELab recognizes several intuitive montage file formats (File › Load › Montage). The recommended format is comma-separated value (CSV)



Chan 20 Win (s) 15 Sen 1

```
3.12 - HandOpenRelaxCl
18.73 - Close
37.49 - Close
40.47 - Open
44.52 - Close
46.57 - Relex
50.41 - Open
54.83 - Close
57.35 - Relex
59.68 - Open
64.90 - Close
67.18 - Relex
71.54 - Open
77.96 - Close
80.11 - Relex
82.63 - Open
87.42 - Close
90.86 - Relex
95.23 - Close
97.43 - Relex
102.94 - Close
105.67 - Relex
108.03 - Open
115.46 - Close
120.00 - Relex
122.08 - Open
127.67 - Close
129.99 - Relex
132.64 - Open
142.53 - Close
153.11 - Close
156.26 - Relex
158.39 - Open
163.30 - Close
165.62 - Relex
170.13 - Open
177.04 - Open
181.29 - Close
183.77 - Relex
186.97 - Open
192.54 - Close
194.71 - Relex
197.75 - Open
203.36 - Close
205.48 - Relex
208.65 - Open
214.18 - Close
217.09 - Relex
219.29 - Open
```

☑ Enable Low: 60 High: - Notch: - Custom: Refresh

52

Data: 2 ; SR: 500 ; Chan: C19 -- 13 | 20
FL: - , FH: - ; FN: - , FCUM:-

Time: 00:00:54.458 -- 14.71%
Gain: 2.2059 ; Value: -0.035785

Fig. 1: A screenshot of CNELab's main interface. Two datasets are visualized together in the horizontal view mode. Dataset-2 includes ECoG channels (C1...C30) ,ECG and EMG which are filtered in different bands and displayed in separate scales. Event panel on the left side lists all annotations and related timestamps (s) in the datasets. The upper part of the main window includes menu bar and toolbar while the bottom part includes filter panel, navigation panel and information panel. Vertical scroll bar of each dataset is placed on the right side.

text file. Each row of montage CSV file should have channel name, montage transformation and optionally channel group name from left to right in sequence. Channel name is the new name assigned for the channel. Montage transformation should be written following a simple arithmetic format. Assume the original channels are named as C1, C2, ... Cn, a montage transformation should be written in the following format:

$$w_1 \times C1 \pm w_2 \times C2 \cdots \pm w_n \times Cn \tag{1}$$

Where, w_i is the number that represents the weight of each channel and defaults to 1 if not provided explicitly. Channels belong to the same group are assigned to the same color for visualization. If not the original channel names are not specified, channel indices can be used in the montage file. Lines prefixed by % in the file are treated as comments. An example of montage CSV file is provided in Table-I.

TABLE I: An example of montage CSV file

%Channel name,	Montage transformation,	Group[1]
Bipolar ECG,	ECG_L-ECG_R,	ECG
Mean Ref,	0.75*C1-0.25*C2-0.25*C3-0.25*C4,	EEG

Events are text messages annotated at specific time points of the dataset. In CNELab, events can be either imported from external files (File ⟩ Load ⟩ Events) or created while reviewing the dataset (✎). External CSV event file is comprised of at least two columns. The first column are the timestamps in seconds while the second column are the related text annotations. Optional columns are color and event code. An example of event CSV file is provided in Table-II.

In default, event code equals to zero and non-zero event code represents events detected automatically from the trigger channel. CNELab automatically assign color for events such that the events with the same annotations are displayed in the same color. Both events and montages can be switched from each other in Event and Montage once loaded into CNELab. Events can also be retrieved from a selected trigger channel. CNELab provides a EKG R-wave peak detection function as an example. To add customized trigger event detection methods, the following procedure needs to be executed in sequence:

TABLE II: An example of event CSV file

%Time (s),	Text,	RGB color[1],	Code[1]
0,	Task start,	0 1 0,	0
200,	Task end,	0 1 0,	0

1) Write customized detection function with the same input and output parameters as *get_ekg_qrs.m* and load into CNELab via Event ⟩ Trigger ⟩ Function ⟩ Load.

[1] Optional

2) Check the desired function listed in Event › Trigger › Function;
3) Select the trigger channel and execute the detect function (Event › Trigger › Detect).

E. PSD and time-frequency maps

CNELab uses Welch's methods to estimate power spectrum density on selected channels and time ranges (📡). Time range selection methods are listed as follows:

- ⬚ selects the data in between by left clicks. ctrl or ⌘ left click on a selected time range deselects it.
- ⬚ selects the current window.
- No selection is equivalent to select the entire range.
- ➚ (esc) deselects all tools and resets the focus to the main window.

PSD estimation parameters such as window length and overlap length can be modified in Apps › Power Spectrum Density. When multiple segments are selected, each power spectrum is weighted by the segment length and averaged to generate a unbiased estimation.

Time-frequency analysis in CNELab can also be performed on selected channels and segements (▦). Apps › Time Frequency Map provides customized parameter settings and visualization controls. Both PSD and time-frequency analysis can be averaged across trials that are annotated in the dataset.

III. Program specifications

A. Prerequisites

CNELab is currently developed in Matlab 2016b under MAC operating system. However backward compatibility is maintained with Matlab 2014a. Releases before 2014a might not be compatible with CNELab as the precompiled Java GUI components used depend on Java 1.7. In addition, Matlab 2014b and 2015a are not suggested because of noticeable GUI performance drops due to Matlab's updating its graphic system in these two releases. Although precompiled mex function of multithreaded filter for both Unix and Windows systems are provided, it is suggested to recompile them in the local running machine for maximum compatibility.

As the video playback in CNELab depends on the activex control, this feature is automatically disabled in Unix like operating system such as MAC. It is also suggested to install VLC player in Windows system so that CNELab can use for better video format support. Otherwise, windows multimedia player would be chosen instead.

B. Installation and execution

To install CNELab on a local machine is straightforward. First unzip the package and switch the Matlab directory to the root folder of CNELab. Then type *setup* in Matlab command line. This will automatically add necessary directories into Matlab path and attempt to recompile the mex functions if the compiler is accessible.

To launch the application, simply type *cnelab* in the command line. The package has provided a demo dataset recorded from a digital dataglove which can be directly opened from Start from demo.

IV. Conclusion

CNELab is designed as a user-friendly GUI software to facilitate the visualization of long-term recordings and expedite preprocessing and simple spectrum analysis. Many features in CNELab such as montage and events are designed to be comparable to professional data review software that are only commercially available. As CNELab is written mainly in Matlab and fully open source available under GPLv3 license, it is expected to be of great useful to the research community that focuses on the biomedical signal analysis.

Although after years of development, CNELab is still consistently maintained and updated. Apart from the video playback feature, CNELab should be fully platform independent. The future development plan includes adding a cross-platform video playback package. CNELab also includes a number of advanced analysis features within Apps. These features such as Spatial Spectral Map and Cross Correlation are designed for specific applications [2], [3] and thus not discussed here for simplicity. Other implemented features that are not discussed might be still under development or evaluation to be fully compatible.

In the near future, we plan to organize the source code of CNELab to make it more scalable for development and prepare a fully documented guideline. We hope this software package can encourage more research on the related fields and inspire researchers and developers to build customized toolboxes upon it.

V. System Specifications

A. Prerequisite

As CNELBehv is currently maintained in Visual Studio 2013, Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 needs to be installed. It also requires additional drivers and runtime libraries (dll) to interface the digital gloves and webcam which will be provided on our website.

Backward Matlab compatibility of CNELab is up to 2014a. Matlab releases before 2014a needs to recompile the Java GUI components. Matlab 2014b and 2015a are not suggested due to noticeable GUI performance decline due to its graphic engine updates. Although precompiled mex functions of multithreaded filter for both Unix and Windows systems are provided, it is suggested to install a C++ compiler in Matlab to recompile them in the local running machine for maximum compatibility. As the video playback in CNELab depends on the activex control, this feature is automatically disabled in Unix operating system such as MAC and Linux. In order for Matlab to recognize VLC activex control, VLC needs to be installed in the same bit version (32-bit/64-bit) with Matlab.

B. Installation and execution

CNELBehv uses a text file (*C:/Windows/win.ini*) for program initialization. The COM port numbers of connected sensors including Teensy USB/bluetooth and digital gloves need to be changed accordingly in the initialization file.

CNELab installation is straightforward. First unzip the package and switch the Matlab directory to the root folder of CNELab. Then type *setup* in command line. This will automatically add necessary directories into Matlab path and attempt to recompile the mex functions if the compiler is accessible. To launch CNELab, simply type *cnelab* in the command line. The package has provided a demo dataset recorded from a digital glove which can be directly accessed from the *Start from demo* entry.

References

[1] C. Cao, D. Li, T. Jiang, N. F. Ince, S. Zhan, J. Zhang, Z. Sha, and B. Sun, "Resting state cortical oscillations of PD patients without and with Subthalamic deep brain stimulation, a MEG study.," *Journal of clinical neurophysiology : official publication of the American Electroencephalographic Society*, sep 2014.

[2] T. Jiang, N. F. Ince, T. Jiang, T. Wang, S. Mei, Y. Li, X. Wang, S. Prabhu, and Z. Sha, "Investigation of the spatial and spectral patterns of hand extension/flexion using high-density ECoG," in *7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 589–592, IEEE, apr 2015.

[3] T. Jiang, N. F. Ince, T. Jiang, T. Wang, S. Mei, Y. Li, X. Wang, and Z. Sha, "Local spatial correlation analysis of hand flexion/extension using intraoperative high-density ECoG," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, (Milan), pp. 6190–6193, Institute of Electrical and Electronics Engineers Inc., 2015.

[4] I. Telkes, J. Jimenez-Shahed, A. Viswanathan, A. Abosch, and N. F. Ince, "Prediction of STN-DBS Electrode Implantation Track in Parkinson's Disease by Using Local Field Potentials," *Frontiers in Neuroscience*, vol. 10, no. May, pp. 1–16, 2016.

[5] T. Jiang, T. Jiang, T. Wang, S. Mei, Q. Liu, Y. Li, X. Wang, S. Prabhu, Z. Sha, and N. F. Ince, "Classification of Hand Flexion / Extension Using High-density ECoG," in *BRAININFO 2016 : The First International Conference on Neuroscience and Cognitive Brain Information Classification*, no. c, (Barcelona), pp. 1–4, 2016.

[6] S. Boudet, L. Peyrodie, P. Gallois, D. H. De l'Aulnoit, H. Cao, and G. Forzy, "BioSigPlot: An opensource tool for the visualization of multi-channel biomedical signals with Matlab," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, vol. 2013, pp. 1956–1959, IEEE, jul 2013.