Python Motor Unit Tracking Scripts Guideline

Yuxi Zheng
November 20, 2024

This project is designed for motor unit tracking and includes the workflow for motor unit tracking, comprising pre-processing, tracking-processing, and post-analysis. Users need to complete motor unit decomposition before proceeding with this workflow. This program has been created and tested in Python 3.10.5; please ensure this version or a later one is installed on user's device. Note, these scripts are only for use with EMG data collected using the TMSi 4_8_L HD-EMG grid in trapezoidal force trajectory contraction experiments!

1. Pre_processing_gui.py

1.1 This script is used to generate a GUI for pre-processing before conducting motor unit tracking. Before running this script, the user needs to modify the code on lines 5, 7, 8, and 10 to match their experimental setup, as shown in Figure 1. In this script:

- Line 5 'mvc' represents the force level of MVC used during the experiment, where '15' means 15% of MVC.
- Line 7 'increase_rate' indicates the percentage of MVC that increases or decreases per second during the experiment's rising ramp and falling ramp phases, here '5' represents a 5% increase or decrease per second.
- Line 8 'steady_time' represents the duration of the constant force phase in the experiment, here '15' indicates that the constant force phase lasts for 15 seconds.
- Line 10 'resting_time' represents the duration for which the force trajectory remains at zero before the subject officially starts contracting, here '10' means that the subject maintained a non-contracted state for 10 seconds before starting.
- Line 11 'sampling_frequency' denotes the sampling frequency used to collect EMG signals with TMSi devices."

```
mvc=15 # mvc level

increase_rate=5 # Force increasing rate per second
steady_time=15 # constant force period length (Steady period length)

resting_time=10 # resting time length. Here we assume the resting time is 10 second
sampling_frequency=2000 #sampling frequency
```

Figure 1

1.2 When running this script, the user needs to first choose to load a JSON format file of an already decomposed EMG recording. Subsequently, a GUI as shown in Figure 2 will be displayed.

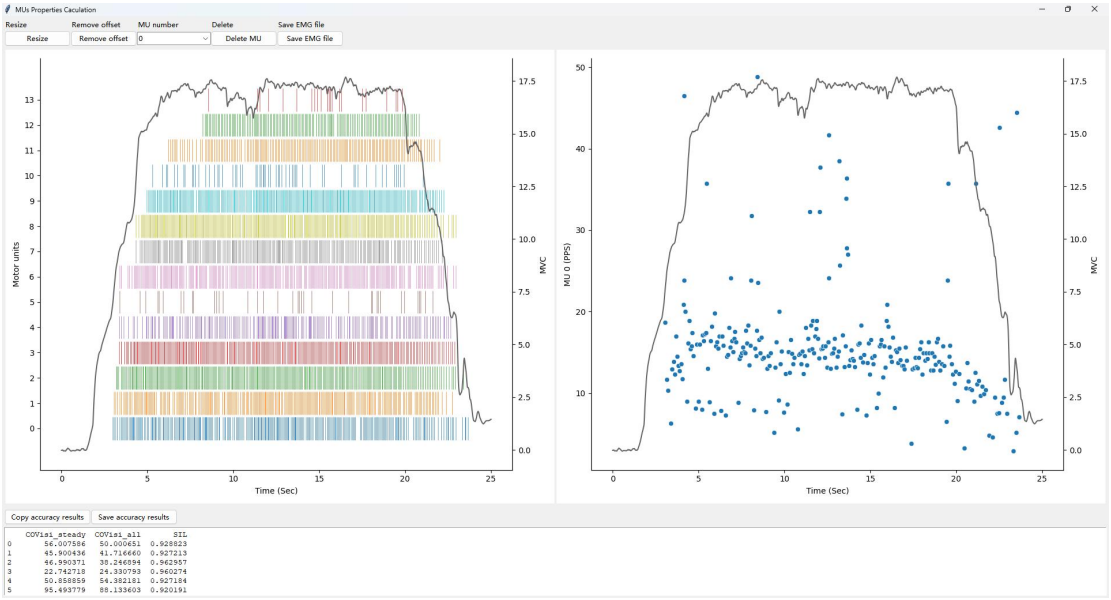

Figure 2

The middle-left side of the GUI displays the motor unit pulses plot of the EMG file, which includes the binary representation of the firing times for each motor unit. Users can select which motor unit they want to observe in the MU number bar above, and the Instantaneous Discharge Rate plot (IDR plot) of that motor unit will be displayed on the middle-right side. Users can click the 'Resize' button at the top of the window and enter a specific time point. As shown in the Figure 3, entering '8' will resize the EMG file to the time interval from the 8th second at the start to the 8th second before the end.

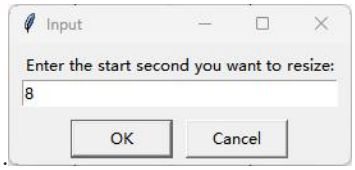

Figure 3

Users can also click the 'Remove offset' button and enter a specific number of sample points, for example as shown in the Figure 4 by entering '3000' (if the sampling rate is 2000, this would select the first 1.5 seconds of the file). The script then refers to the reference signal (force signal) values of the first 3000 sample points and eliminates their offset (i.e., moves the force signal at the resting time to 0% of MVC), effectively shifting this segment of the reference signal to the 0 position on the y-axis.

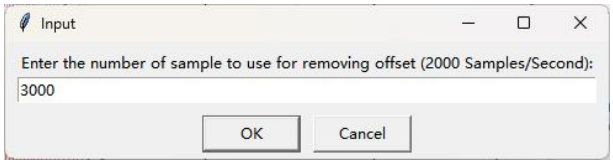

Figure 4

Users can also refer to the accuracy information of the motor unit decomposition and the IDR plot information displayed in the textbox below to choose to delete any unreasonable motor units. After editing

the EMG file, users can then choose to save the file.

2. MU_tracking.py

This script has already sorted each channel of the TMSi 4_8_L HD-EMG grid spatially, as detailed in lines 17 to 22 of the script, as shown in Figure 5.

```
custom_sorting_order_TMSi_4_8_L = [
    [31, 30, 29, 28, 27, 26, 25, 24],
    [23, 22, 21, 20, 19, 18, 17, 16],
    [15, 14, 13, 12, 11, 10,  9,  8],
    [ 7,  6,  5,  4,  3,  2,  1,  0],
]

'''
custom_sorting_order_TMSi_4_8_L = [
    [31[R4C8], 30[R4C7], 29[R4C6], 28[R4C5], 27[R4C4], 26[R4C3], 25[R4C2], 24[R4C1]],
    [23[R3C8], 22[R3C7], 21[R3C6], 20[R3C5], 19[R3C4], 18[R3C3], 17[R3C2], 16[R3C1]],
    [ 15[R2C8],  14[R2C7], 13[R2C6], 12[R2C5], 11[R2C4], 10[R2C3], 9[R2C2], 8[R2C1]],
    [  7[R1C8],  6[R1C7],  5[R1C6],  4[R1C5],  3[R1C4],  2[R1C3],  1[R1C2],  0[R1C1]],
]
'''
```

Figure 5

The motor unit tracking GUI generated by running this script is shown in Figure 6, with the left side showing matching plots of the single differential derivative plot of two EMG files, and the right side displaying the IDR plots of the two files. After completing motor unit tracking, users need to select a file directory to save the tracking results.

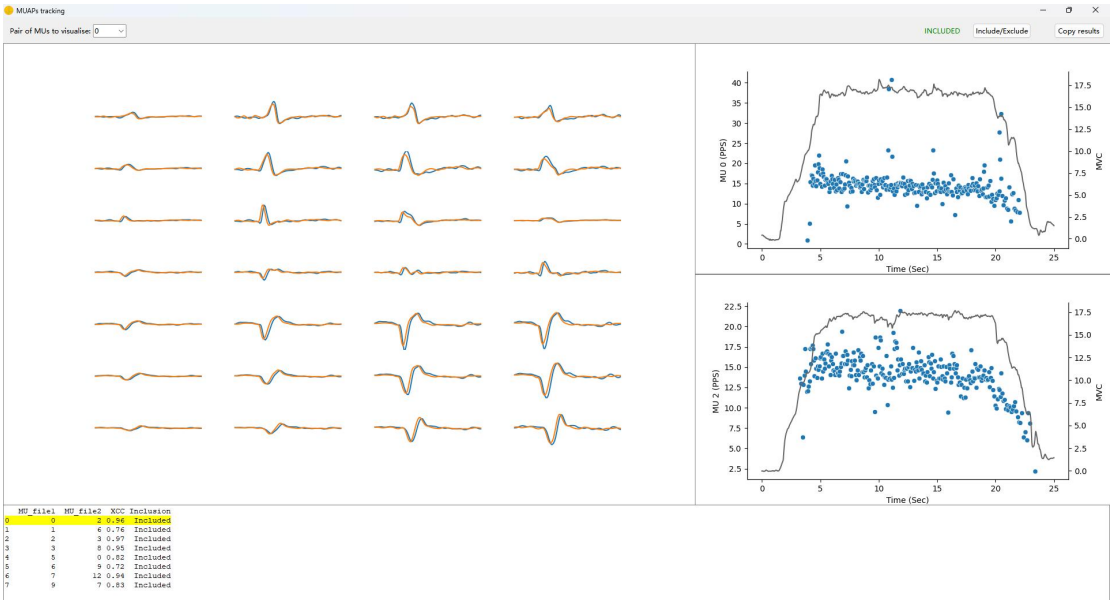

Figure 6

In this GUI, the arrangement of channels is as shown in Figure 6. Users can refer to the TMSi Grid Datasheet to locate the corresponding position of each channel on the grid in the GUI.

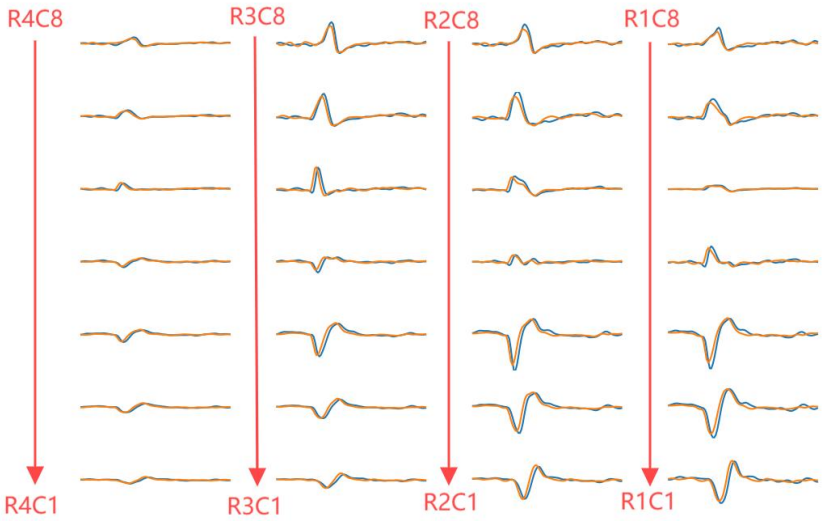

Figure 6

3. Post-analysis_gui.py

This script is the same as Pre_processing_gui.py, and users need to modify the corresponding parameters in the code according to their experimental settings. When running this script, users need to load three files. The first two files are EMG files used for motor unit tracking in MU_tracking.py, and the third file is a result file generated using MU_tracking.py. The GUI generated by this script is shown in Figure 7.

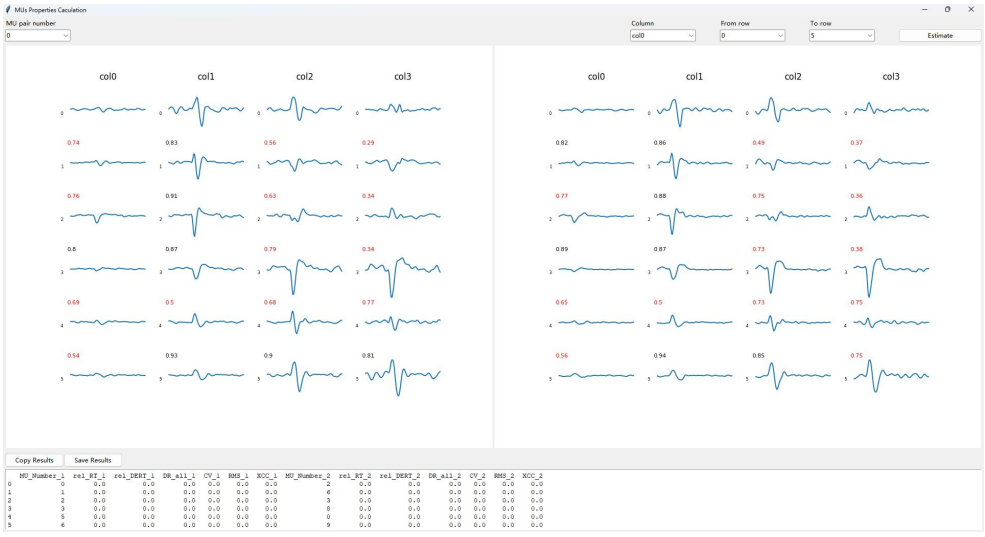

Figure 7

Users can select the matched motor unit pair number from the 'MU pair number' dropdown menu at the top of the GUI, and the middle part will display the double differential derivative plot of the corresponding motor units from the two loaded EMG files. Based on the appearance of these two plots, users can select the appropriate columns and rows interval in the GUI to estimate the properties of that motor unit in two different recordings. These values will be displayed in the textbox below. Finally, users can specify a directory to save these properties results as a file. Note, the order of each channel in the double differential derivative plot should refer to Figure 6.