MATLAB Spike Sorter

Joaquin Mansilla Yulan, Dr. Ing. Sergio Lew

Abstract

Nowadays, there are many offline spike sorters available in the market. However, most of them are priced closed-code softwares. A common issue that comes while doing spike sorting is the need to add new features or algorithms specifically for the signal being processed. The problem of closed-source software lies in user's inability to make changes or add features to the program. This involves from the interface of the software to the code as well. In this work, we develop a free open-source spike sorter on Matlab. This sorter provides most of the main tools used by offline sorters available and brings an easy way to add features and new code segments to the program using Matlab simple code. Our work incorporates the most common and useful tools for manual and semi-automatic spike sorting as a default configuration of the program.

1 Introduction

There are many techniques for making spike sorting. Most of them are manual and requires an expert in the field to achieve good results. This work provides a customizable tool that helps the user to reduce the amount of errors during the sorting process. To describe well how to use the developed tool, firstly it's necessary to explain the spike sorting technique chosen and steps to follow.

This process has a pipeline structure and starts with the acquisition of the signal, generally measured by arrays electrodes placed in somewhere on the brain cortex. The electrical signal obtained is called Raw Data. This signal is filtered digitally by a high-pass filter. A threshold decision is used to detect the spikes of the signal. If threshold is negative, detected peaks will be the ones going under the threshold. If it is positive, the peaks that surpass the threshold will be detected. The result of this process is called Waveforms. Waveforms which do not seem to any neural spike are eliminated manually. The next step is to get some features of these waveforms in order to discriminate them using only a few main characteristics of the signal. These features allows the user to group graphically waveforms with similar particular characteristics. It's possible to show two or three features at the same time by displaying those creating 2D or 3D points plots and then discriminate different "clouds" of feature points. The cloud detecting process is called Clustering and it can be made manually or semi-automatically using an automatic clustering algorithm. Finally, waveforms are separated in Units corresponding to each cluster and the work is saved. The following pipeline represents the steps of the spike sorting process:

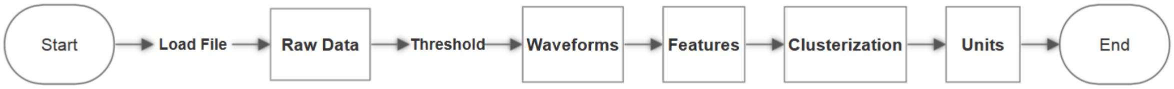


Figure 1: Spike sorting pipeline process

2 Development

Matlab Spike Sorter is a new free stand-alone tool for making offline spike sorting. The program has a default configuration that includes the most popular features used by other actual sorters and also has a few new ones. The greatest advantage of this tool is the user capability to set all the parameters of the program and also, modify his code, add new utilities and understand the signal processing made. Also, the friendly interface brings users a clear tracing of the sorting process and a fast way to learn to use this tool. In addition, Matlab is a massive tool well-known by most of the scientists and researchers of biomedical and neuroscience field. Therefore, Matlab Spike Sorter provides a convenient and powerful offline sorter tool which is also adjustable to any sorting project. Default features of this sorter are:

- Dynamic parameters
- Multiple channels data input
- High-pass filtering
- Threshold spike detection
- Waveforms manual selection
- Graphical feature space representation
- Three dimension features representation using color scale and points density
- Typical methods to feature extraction (PCA, Slice)
- Basic template-matching algorithm and K-Means (semi-automatic clustering)
- Waveform alignment
- Raster displays of spikes and continuous data
- Cluster shoot-frequency plot
- Results recording
- Adjustable interface

Our sorter has typical steps to follow in order to make spike sorting and according to the process explained before. It starts loading the Raw Data file and obtaining Waveforms. It's possible to select many features to generate the feature space conveniently and then to separate waveforms in Units according to each cluster. Finally, the result is saved and the process can be repeated. The next block diagram shows the sorting process and all features and tools that our program has by default:

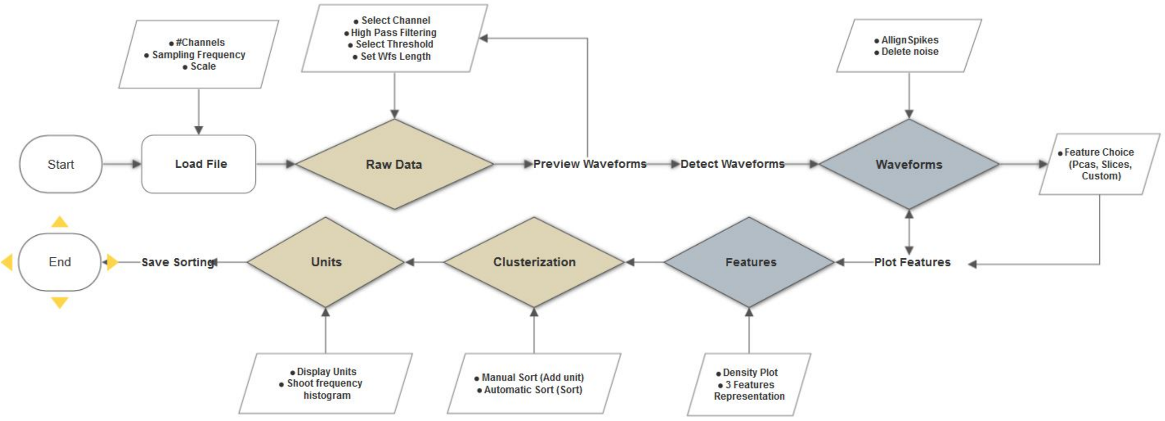
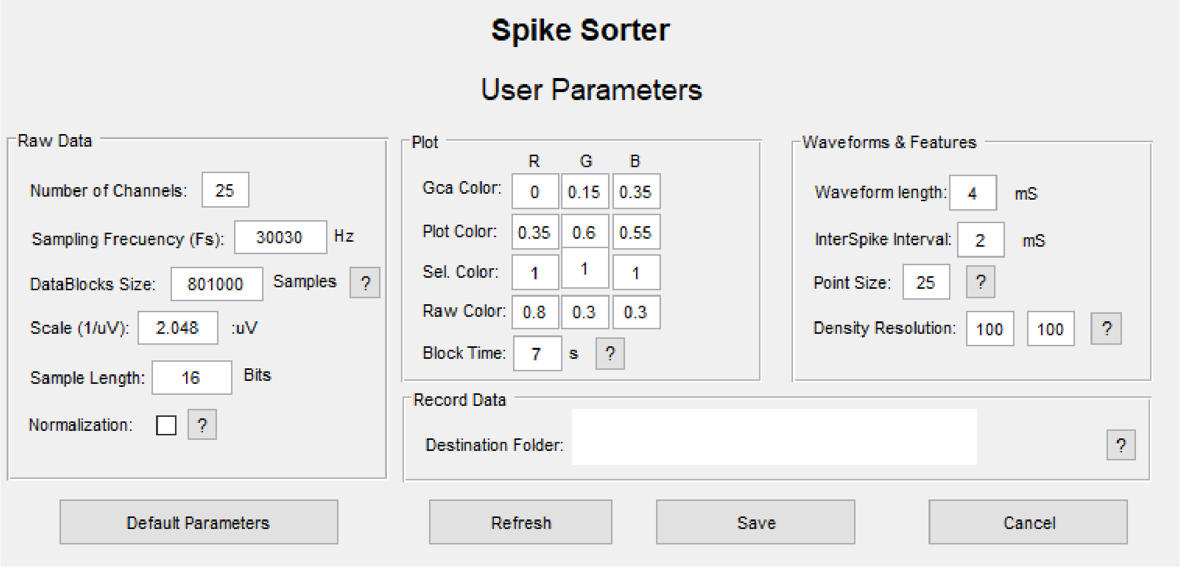


Figure 2: Matlab Spike Sorter diagram

In order to understand how to use this tool, each default feature of the program is described at the following section:

2.1 Start - Setting Parameters

Parameter Configuration The first step is to set the correct and desirable parameters of the program considering the characteristics of the input raw data, the kind of sorting wanted and the desired display of results and data. A good and careful selection of each parameter is recommended to obtain proper results and a good use of this tool. ”Parameter Configuration” opens a panel where is possible to set each parameter. Default parameters are the following:



Spike Sorter

User Parameters

Raw Data

Number of Channels: 25

Sampling Frecuency (Fs): 30030 Hz

DataBlocks Size: 801000 Samples

Scale (1/uV): 2.048 :uV

Sample Length: 16 Bits

Normalization: □

Plot

	R	G	B
Gca Color:	0	0.15	0.35
Plot Color:	0.35	0.6	0.55
Sel. Color:	1	1	1
Raw Color:	0.8	0.3	0.3

Block Time: 7 s

Waveforms & Features

Waveform length: 4 mS

InterSpike Interval: 2 mS

Point Size: 25

Density Resolution: 100 100

Record Data

Destination Folder:

Figure 3: Parameter configuration Panel

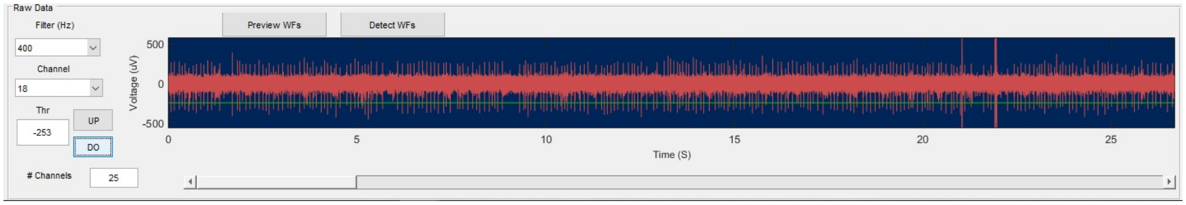
2.2 Load file

Raw Data It is possible to load two different types of raw data: a binary file of any type specified by the corresponding parameters or a one channel .mat file containing only one variable with the signal information of an unique channel. Default parameters expects a 16 bits binary file of 25 channels. Also, it is possible to scale the amplitude level of raw data by setting the Scale parameter or make a normalization of data by selecting Normalization parameter to 'on'.

Recorded Units Load a previously saved file containing the result of a clustering done earlier.

2.3 Raw Data

Once raw data is loaded, it will be plotted on the "Raw Data Plot" at the bottom of the screen (Voltage vs Time plot). The channel menu allows to display each channel on the plot. Before waveforms detection, it is necessary to determine the cutoff frequency of the digital High-Pass filter ("Filter (Hz)" box) and the threshold level using the "up" and "do" (down) buttons. It is possible to acquire a preview of waveforms in order to observe the shape of spikes under a selected threshold using "Preview Wfs". It will detect waveforms on just the displayed portion of the channel instead of all. This allows to test different levels of threshold fast, avoiding to waste time processing the entire signal. After applying the filter and selecting the convenient threshold, the next step is the spike detection on the entire channel using "Detect Wfs".



Raw Data

Filter (Hz): 400

Channel: 18

Thr: -253

Channels: 25

Figure 4: Raw Data

2.4 Waveforms

Once spikes are detected, they are extracted from the signal as waveforms and displayed on the Waveforms Plot at the left top corner of the screen. Default configuration creates Waveforms of 4 ms of length with the peak on the middle of the waveform. This length is adjustable (before loading the file) by changing the value in "Wfs L". Before getting features from these waveforms, it is necessary to fit them out.First step is to align the waveforms to compare them better and extract correctly their features. Align to maximum or minimum is convenient depending on shape of waveforms. Then, it is optimal to eliminate those waveforms who does not correspond to any neural spike (artifact spikes) by selecting them using any of both selectors (square and free hand) and "Delete selection" in red. After this important process, it is possible to extract features from the waveforms using "Plot Features".

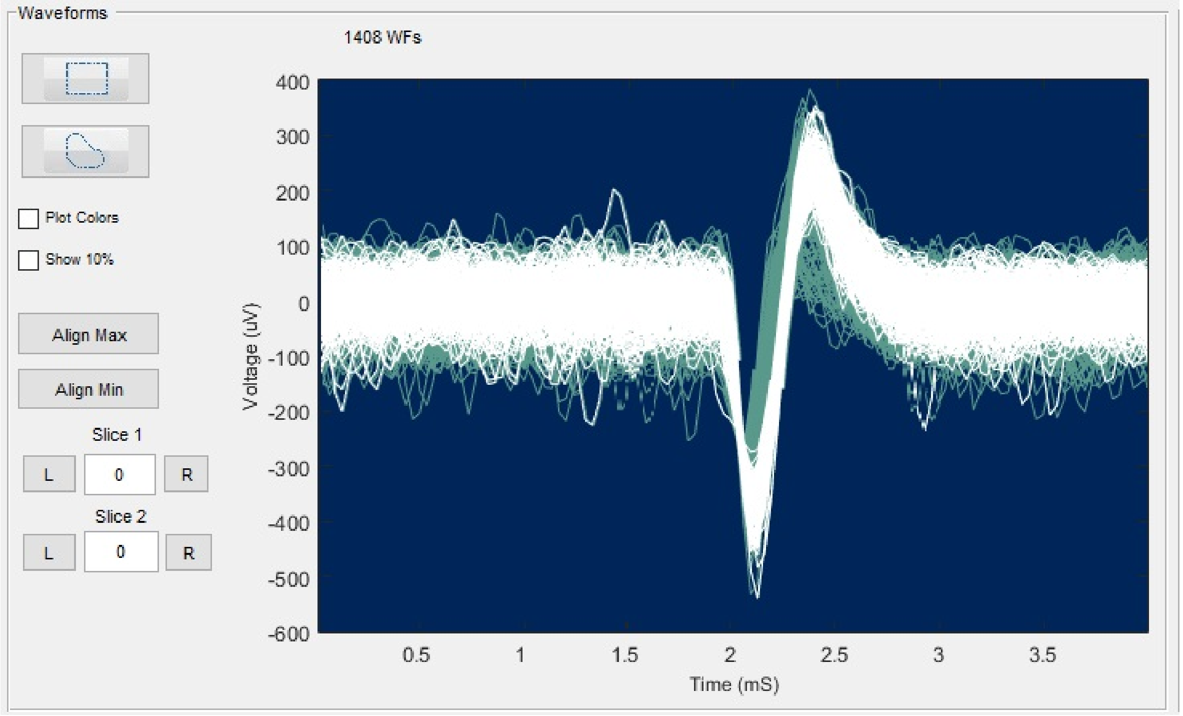


Figure 5: Waveforms

2.5 Features

Feature space is a representation of the two or three most relevant features of waveforms on a 2D plane plotted at the top-middle side of the screen. The selection of these features is extremely important in order to achieve a good clusterization and sorting. This tool comes with most used features for spike sorting: PCAs up to the fourth component (Principal Component Analysis) and 2 Slices (Amplitude of each signal at a preset instant of time). Slices can be located at the waveforms plot using the menus with the same name. The user could also add many others feature finders by appending functions to the Matlab script ”Features” located at the root of the program. It is possible to represent 3 features at the same time using the 2 spatial dimensions (x and y) and a third ”axis” as a color scale. The third feature could be also the points density also applying a color scale with ”Show Density”. Selection, deselection and deletion of feature points works as the same way as waveforms. Over this features plot, it is possible to make a correlation between each feature point and his waveform by clicking inside the plot axes and positioning the cursor over the desired point.

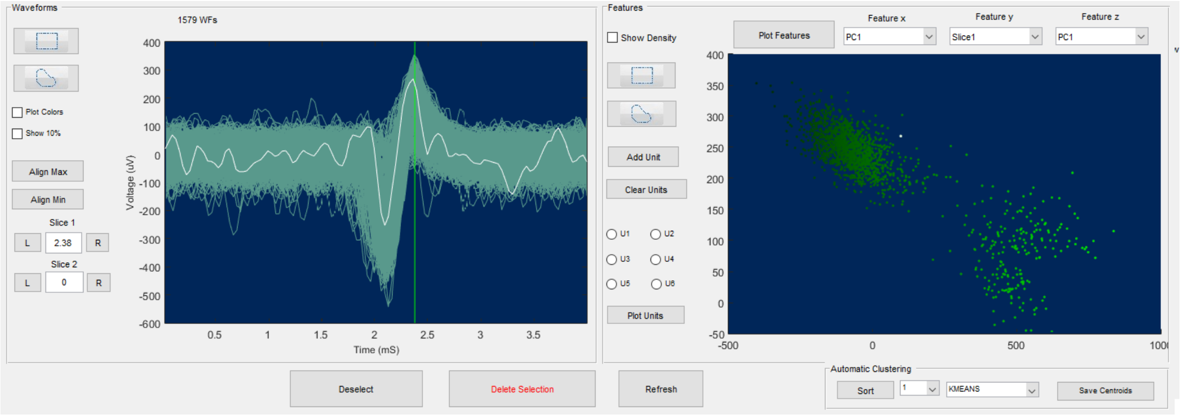

Feature x: PC1
Feature y: Slice1
Feature z: PC1
Slice 1: 2.38
Slice 2: 0
Automatic Clustering: 1 KMEANS

Figure 6: Features

2.6 Units

Once obtained a good characterization of waveforms, clusterization could be made manually by selecting a cloud of points on the features plot and using the ”Add Unit” function for each new cluster. Also, it is possible to do an automatic clusterization by using ”Sort” specifying the number of clusters to sort. Automatic algorithms are two by default: K-Means and Template Matching. The last one needs a template data base with a previous classification. In order to add new clusters as templates, it is necessary to use ”Save Centroids” at the end of each sorting. This process adds a new template taking the mean of the waveforms within the cluster as template. Also, it is possible to add new automatic clustering algorithms to the program adding new functions to the Matlab Script ”clustering_algorithm” located at the root of the program.

In order to reset the sorting, use the ”Clear Units” function to erase any classification made before. For displaying certain units, there is ”Plot Units” which shows only the selected units ”U#”.

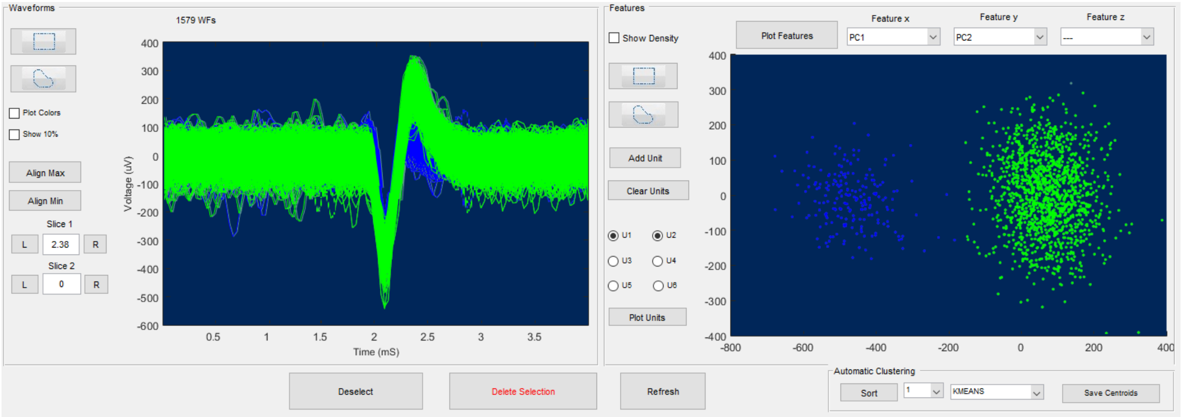

Feature x: PC1
Feature y: PC2
Feature z: ---
Slice 1: 2.38
Slice 2: 0
● U1
● U2
Automatic Clustering: 1 KMEANS

Figure 7: Units

2.7 Results and record

Then, it is important to observe the waveforms of each cluster separately and their shoot frequency, in order to determine if they could be in the same cluster or not and also, to get more valuable information about the behavior of each one. Function ”Show Units-Spikes” displays all waveforms of each cluster and an their histograms showing the number of occurrences of spikes for different time intervals between two consecutive shots in that cluster. Thus, if an error were found, the user could go back to any previous step and continue the process in the same way.

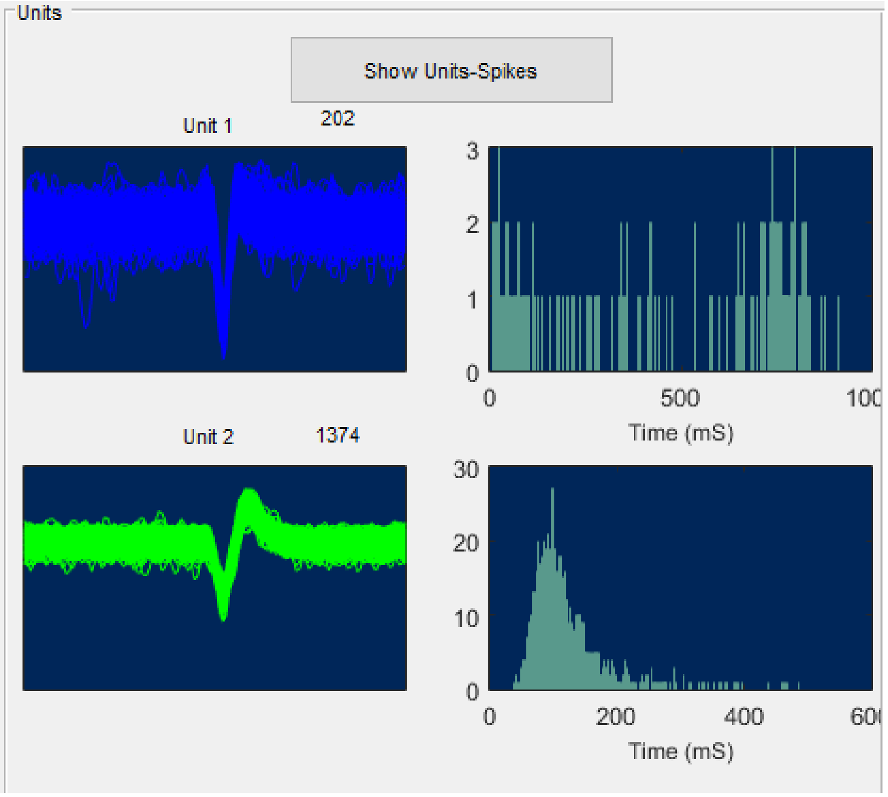


Figure 8: Results

Finally, the results are recorded by ”Save Spikes” which saves by default: the waveforms, clustering and time of occurrence of each spike. Furthermore, if desired, it is possible to save clusters as templates for the template matching algorithm as it was explained before.

3 Conclusion

Our development allows users to obtain an intuitive and customizable tool for making a fast and correct spike sorting. The greatest advantage of this work is the possibility of modify, add and edit features of the program. Due to the chosen platform is Matlab, this will produce a major interest in customize the tool and an easier way to do it. By default, the program has all the features necessaries to achieve a good sorting when user is enough expertly on spike sorting. Although, there is a need to get better automatic algorithms or semi-automatic algorithms in order to decrease sorting errors because of human errors . It would be very interesting to create and train models using machine-learning algorithms, may be supervised, for this purpose.

Finally, it is important to mention that this work pretends to be a start point in the creation a new full functionally Matlab tool for spike sorting.