**Boston University**

**Electrical & Computer Engineering**

**EC463 Senior Design Project**

Final Test Plan

**Neuron Spike Identification with Machine Learning**



Chen Yang, PhD

Yang Research Laboratory

cheyang@bu.edu

by

Team 2

Spike Sorters

Team Members

Victoria Carlsten [carlsten@bu.edu](mailto:email1@bu.edu)

Hao Chen [ha0chen@bu.edu](mailto:email1@bu.edu)

Claire Cropper [ccropper@bu.edu](mailto:email1@bu.edu)

Shi Gu [bengushi@bu.edu](mailto:email1@bu.edu)

**Required Materials/Equipment:**

Hardware:

* Personal Laptop

Software:

* Python, 1 scripts
* 2 data files
  + *spikesortingVTJason.py*
    - Load recording data with different type (30 min\_0001.abf, Data\_Subject\_05\_Session\_01.h5)
    - Detect spikes
    - Delete unnecessary data
    - Implement Regular K-means and PCA analysis
    - Implement Optimized K-TOPS algorithm and PCA visualization
    - Evaluate the Optimized template and K-means with L-ratio and Silhouette score

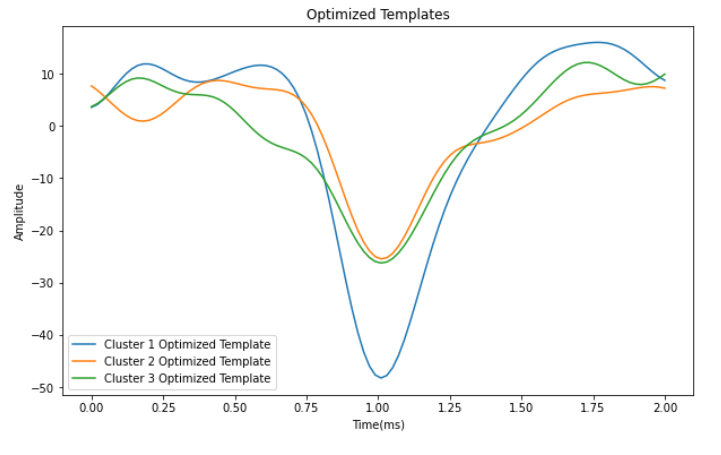
**Setup:**

The setup only consists of one part: using our personal laptops to generate graphs via one Python script. Firstly, running the spikesortingVTJason.py will implement K-means and PCA to generate clustering results. The user will need to input 2 cluster numbers to select the interested cluster. And a new clustering result with a different algorithm will be generated to compare with the former result.

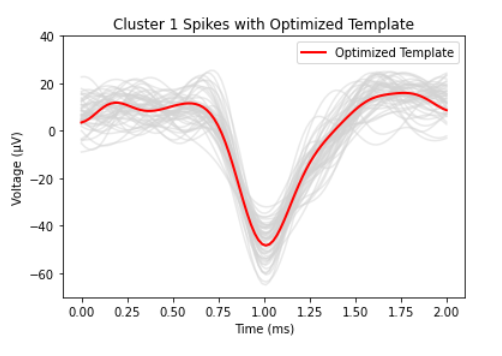
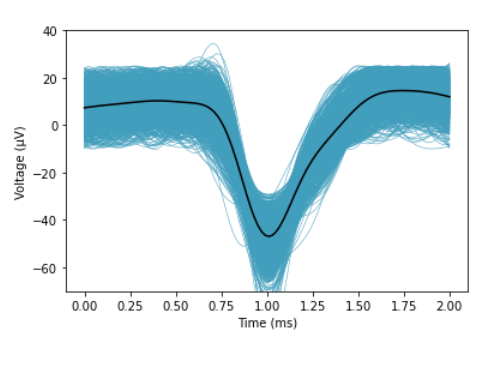
**Pre-testing Setup Procedure:**

spikesortingVTJason.py file:

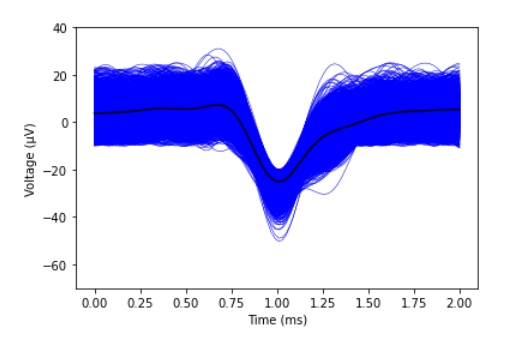
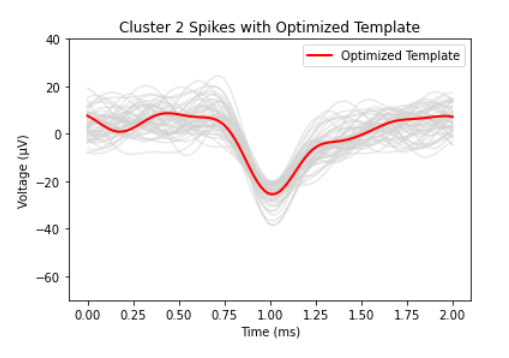
1. Make sure the user is in the correct working directory.
2. Make sure the electrophysiology datasets are imported and in the working directory.
3. Run all the lines of the python script, *spikesortingVTJason.py* and make sure no errors occur.
4. Observe the plots generated below the code.



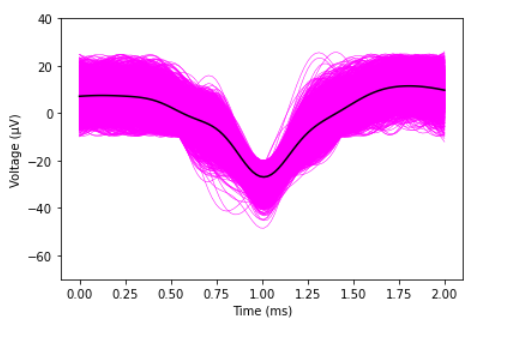
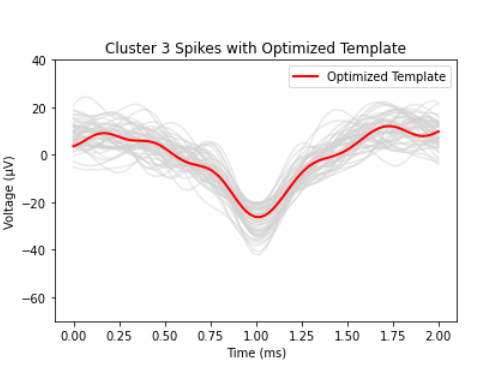
*Figure 1. Optimized Templates for 3 clusters generated by initial k-means*

******

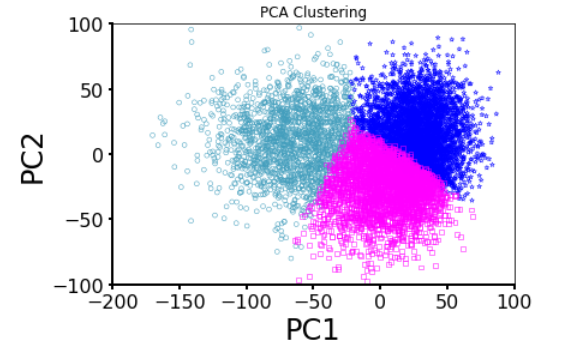
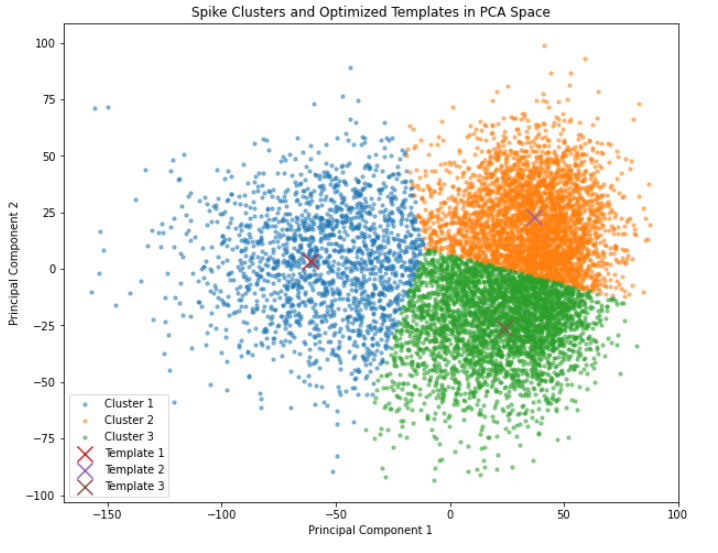
*Figure 2. Optimized Template (left) and mean of spikes (right) with all spikes in cluster 1*



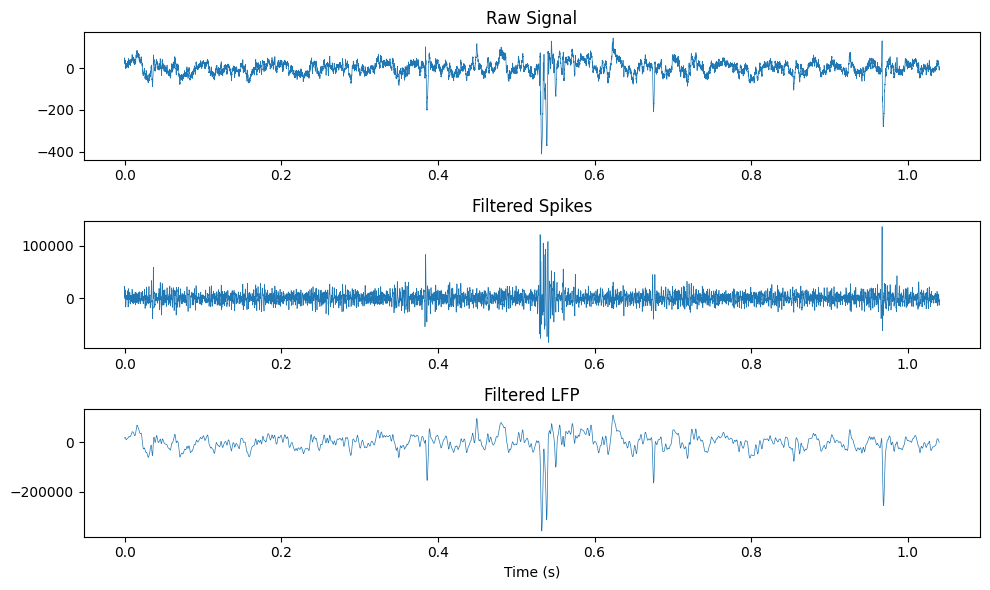
*Figure 3. Optimized Template (left) and mean of spikes (right) with all spikes in cluster 2*

**

*Figure 4. Optimized Template (left) and mean of spikes (right) with all spikes in cluster 3*



*Figure 5. Clustering using K-TOPS (left) and k-means (right) in PCA space*



*Figure 6. Raw Signal, Filtered Spikes, Filtered LFP Graph of .h5 File*

**Testing Procedure:**

1. Run spikesortingVTJason.py
2. The computer will automatically open the working directory and the users can choose the dataset they want to import.
3. Ensure dataset is compatible with the algorithm constraints
4. Input two clusters that users are interested in to see detailed graphs
5. The computer will then generate clustering results using a different clustering approach.
6. The program will generate the evaluation results for the clustering.

**Measurable Criteria:**

1. spikesortingVTJason.py should correctly load the recording data that users selected and give error messages if users choose unsupported datatype.
2. Spikes should be detected according to the threshold value and output a plot similar to Figure 2.
3. First step of PCA analysis of the spike array should output a plot similar to Figure 3.
4. There should be three(desired by us) clusters clearly classified in the PCA graph.
5. Two clusters can be selected in a separate graph by input.
6. Three clusters using the K-TOPS algorithm should be shown in a different graph.
7. The evaluation results (L-ration and Silhouette score) of two different clusterings should be presented.

**Score Sheet**

| **Object** | **Category** | **Correct? (Y/N)** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Result →** | | |