**Boston University**

**Electrical & Computer Engineering**

**EC464 Senior Design Project**

Prototype Testing 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:**

Hardware:

* Personal Laptop

Software:

* Python, 1 scripts
* 1 data files
  + *spikesortingVTJason.py*
    - Load recording data
    - Detect spikes
    - Delete unnecessary data
    - Implement Regular K-means and PCA analysis
    - Implement Optimized K-TOPS algorithm and PCA visualization
* Python, 1 script
* 2 data files
  + *spikesortingVTJason\_Data.py*
    - Load recording data files
    - Detect spikes
    - Delete unnecessary data
    - Implement Regular K-means and PCA analysis

**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.

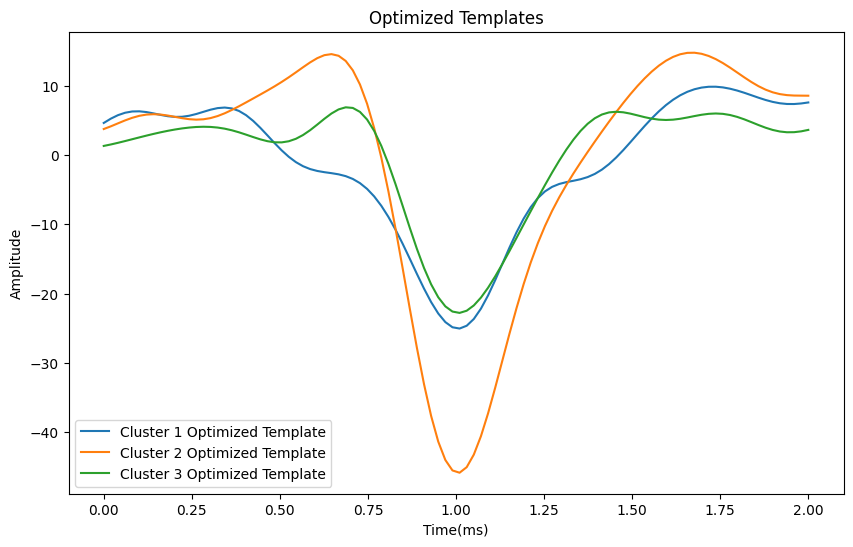
**Pre-testing Setup Procedure:**

spikesortingVTJason.py file:

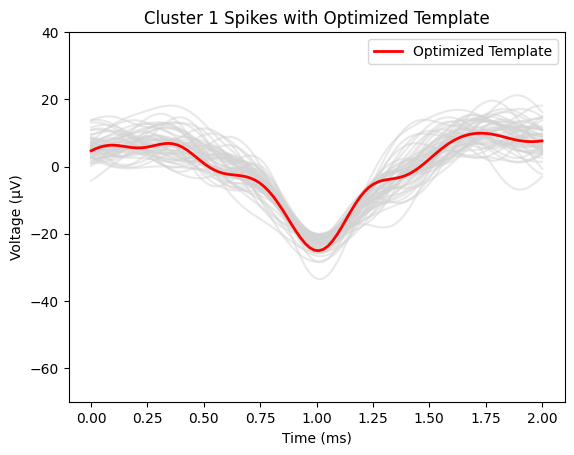
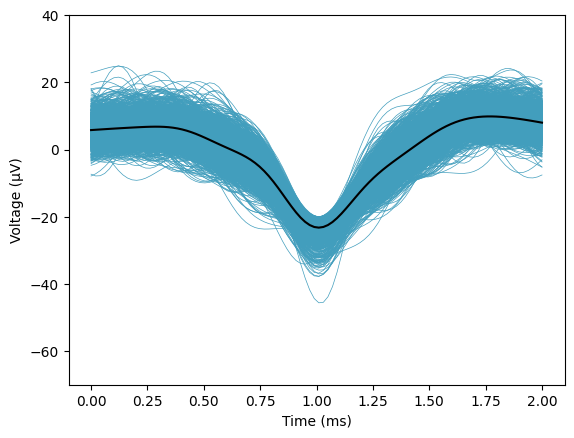
1. Make sure user is in the correct working directory
2. Make sure ‘10 min recording1.mat’ is in the working directory
3. Run all the lines of the python script, *spikesortingVTJason.py* and make sure no errors happen

spikesortingVTJason\_Data.py file:

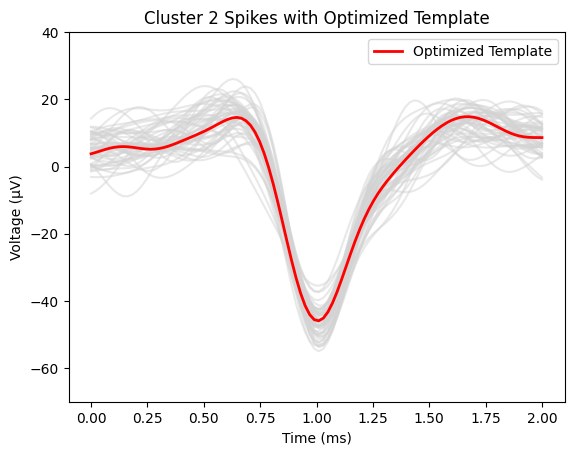
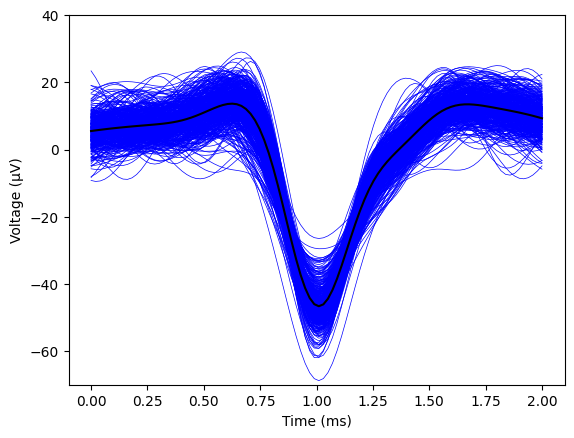
1. Make sure user is in the correct working directory
2. Make sure ‘Data\_Subject\_08\_Session\_01.h5’ is in the working directory
3. Run modified lines of the python script, *spikesortingVTJason.py*



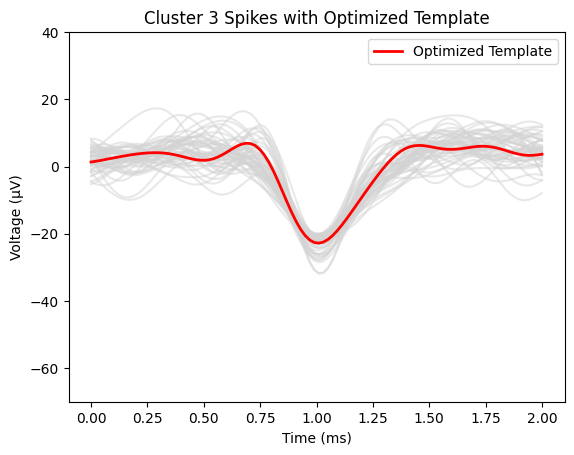
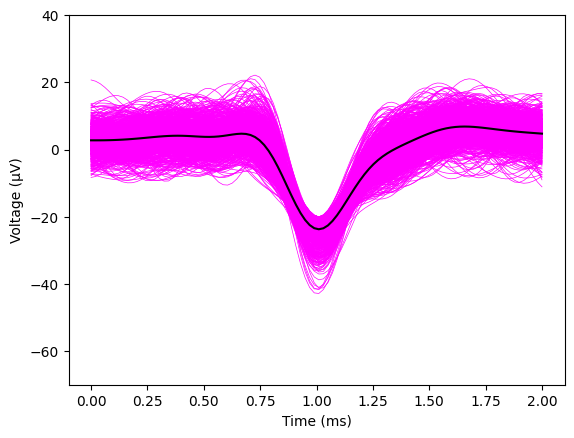
*Figure 1. Optimized Templates for 3 clusters generated by initial k-means*

******

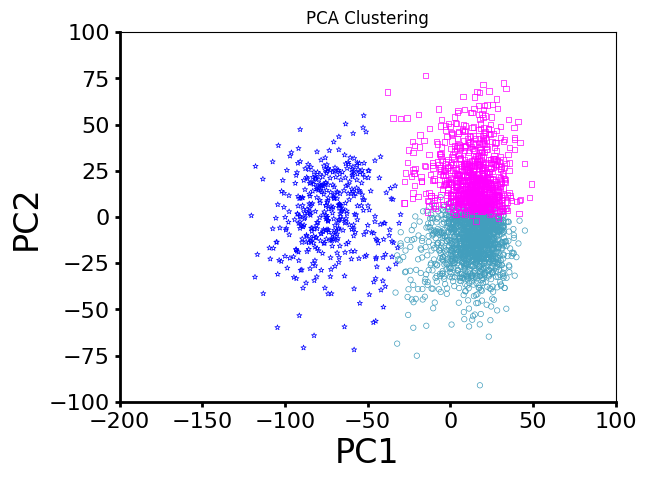
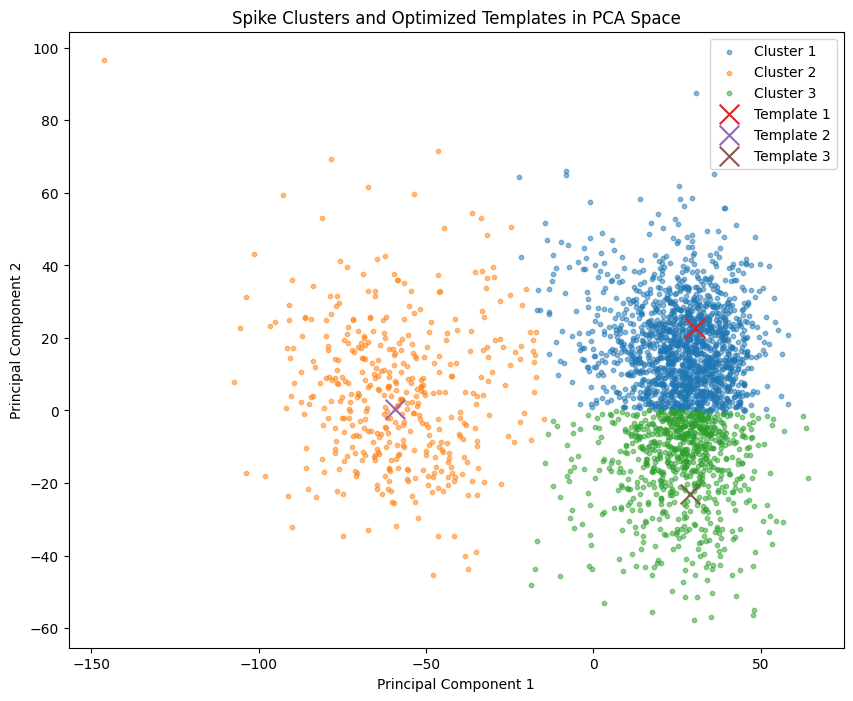
*Figure 2. Optimized Template(left) and mean of spikes(right) with all spikes in cluster 1*

**

*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-means(right) and K-TOPS(left) in PCA space*

**Testing Procedure:**

1. Run spikesortingVTJason.py
2. The computer will load the recording data from the ‘10 min recording1.mat’ and identify spikes
3. Input two clusters that users are interested in to see detailed graphs
4. The computer will then generate clustering results using a different clustering approach
5. Run spikesortingVTJason.py
6. The computer will load the recording data from the ‘Data\_Subject\_08\_Session\_01.h5’ and identify spikes

**Measurable Criteria:**

1. spikesortingVTJason.py (lines 1 - 118) should correctly load the recording data from the ‘10 min recording1.mat’, which is a mat file of pre-recorded 10 minutes of Electrophysiological Data, and then select the part of interest.
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. L-ratio should be around or lower than 0.02
8. spikesortingVTJason\_Data.py should correctly load the recording data from the ‘Data\_Subject\_08\_Session\_01.h5’, which is a h5 file of pre-recorded 29 seconds of Electrophysiological Data, and then select the part of interest.
9. Spikes should be detected according to the threshold value and output a plot similar to Figure 2.

**Score Sheet**

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