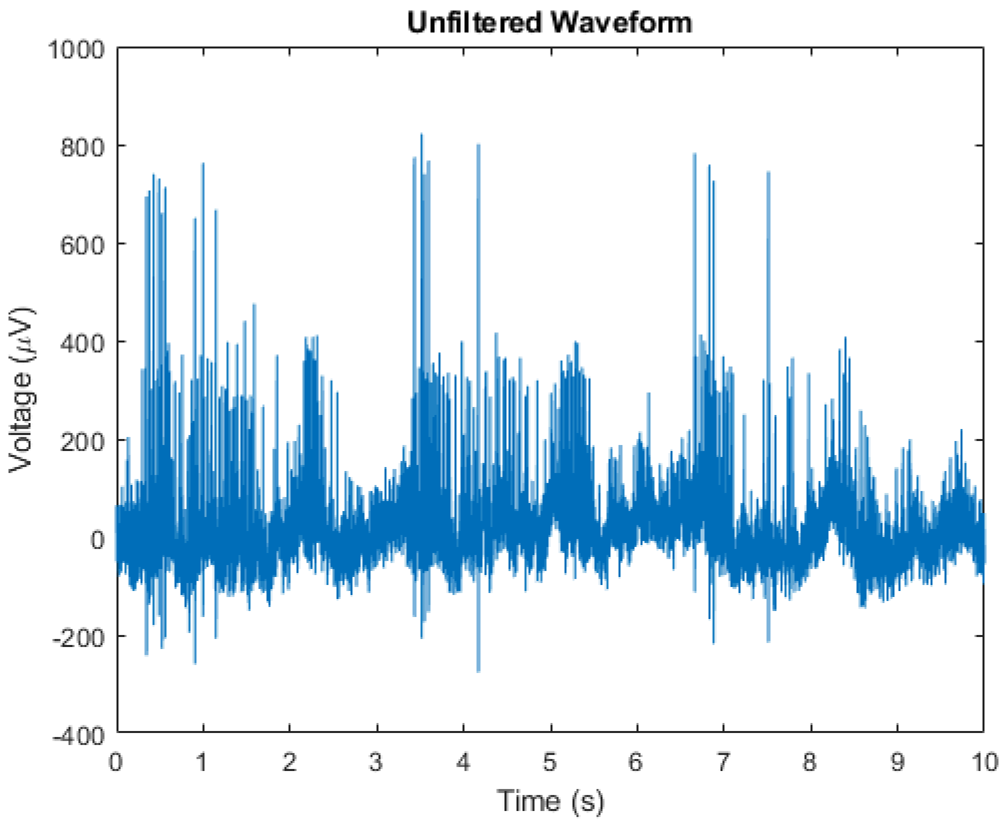
Table of Contents

Plot the entire waveform

Loads data and plots raw waveform

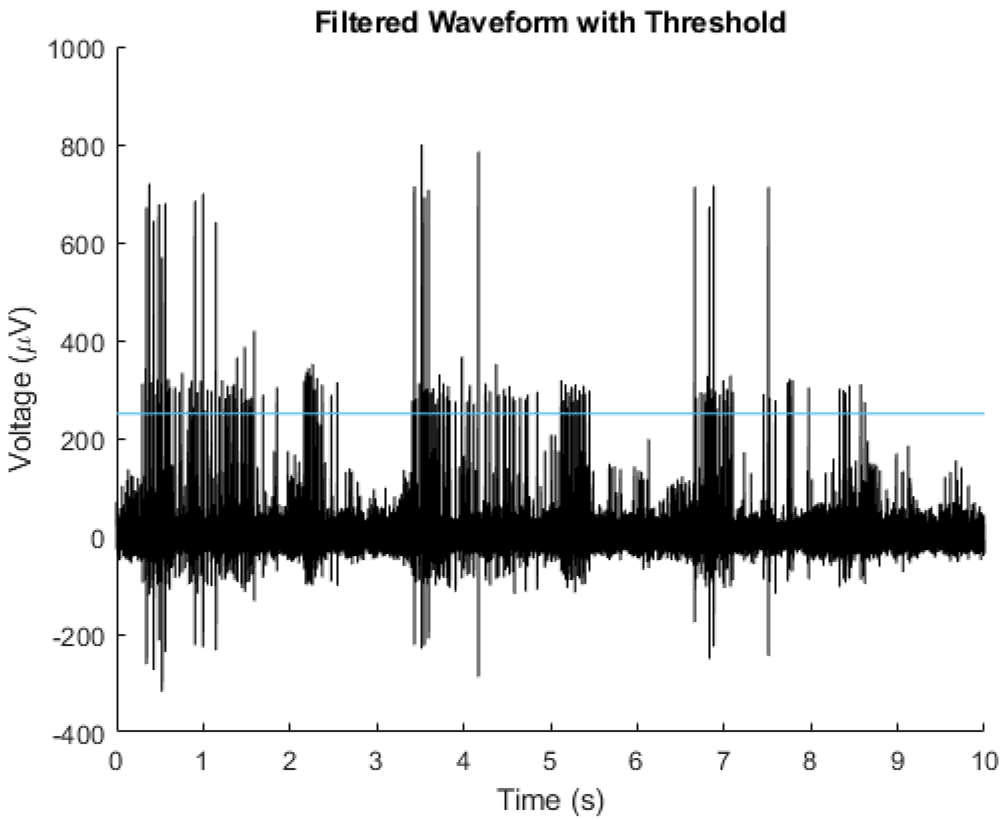
```
load('ps5_data.mat')
t = linspace(0,10,300000);
figure
plot(t, RealWaveform')
title('Unfiltered Waveform')
xlabel('Time (s)')
ylabel('Voltage (\muV)')
```



Plot filtered waveform

You will notice a low frequency component in the signal, known as the local field potential (LFP). We are not interested in analyzing the LFP during spike sorting. Remove the LFP using a high-pass filter, stopping frequencies below 250 Hz. Also, set a threshold, Vthresh = 250 μV, and plot the threshold as a line across plot

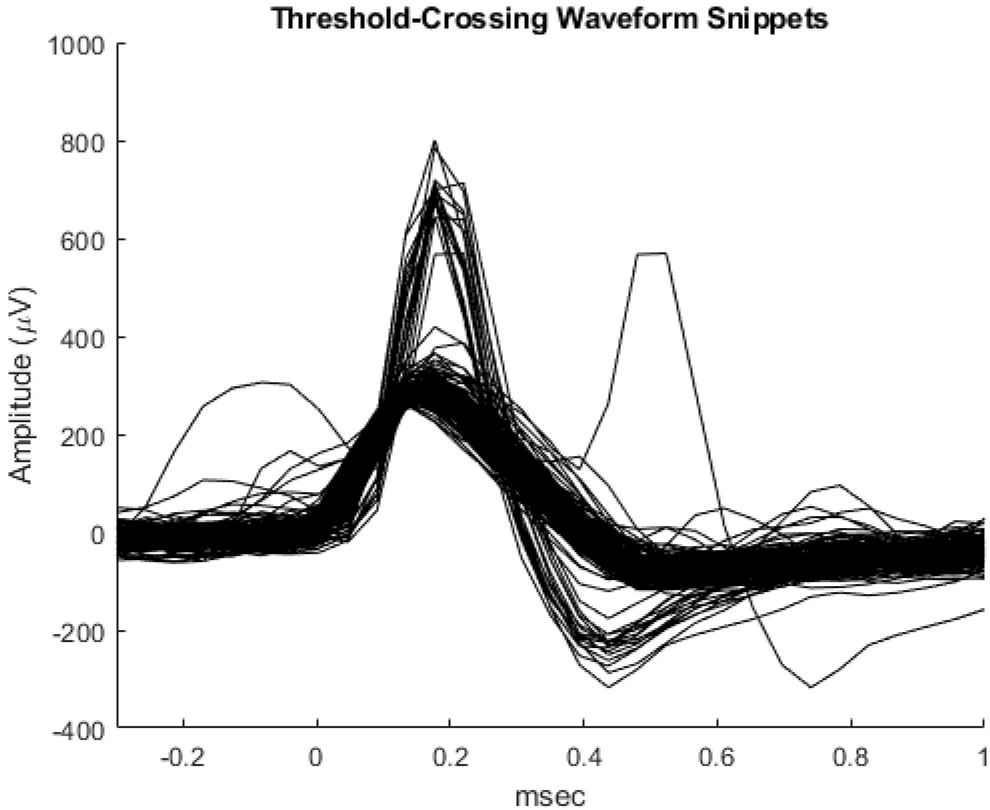
```
x = RealWaveform;
f_0 = 30000; % sampling rate of waveform (Hz)
f_stop = 250; % stop frequency (Hz)
f_Nyquist = f_0/2; % the Nyquist limit
n = length(x);
f_all = linspace(-f_Nyquist,f_Nyquist,n);
desired_response = ones(n,1);
desired_response(abs(f_all)<=f_stop) = 0;
x_filtered = real(ifft(fft(x).*fftshift(desired_response)));

figure
hold on
plot(t,x_filtered','Color', 'black')
line([0 10],[250 250],'Color',[0.3010 0.7450 0.9330])
title('Filtered Waveform with Threshold')
xlabel('Time (s)')
ylabel('Voltage (\muV)')
hold off
```



Spike Detection

Take 1 ms snippets of the waveform beginning 0.3 ms before each threshold crossing. Each snippet should have 31 samples; the tenth sample should be less than Vthresh, and the eleventh sample should be greater than Vthresh

```
threshold = find(x_filtered>=250);
cross = [];
group = threshold(2):threshold(2)+5;
for i = 1:length(threshold)
    index = threshold(i);
    if ismember(index,group)
        continue
    else
        cross = [cross index];
        group = threshold(i):threshold(i)+5;
    end
end

figure
hold on
t_snip = linspace(-0.3,1,31);
snippets = zeros(31,length(cross));
for j = 1:length(cross)
    index = cross(j);
    snip = x_filtered((index-10):(index+20),1);
    snippets(:,j) = snip;
    plot(t_snip,snip','Color','black')
end
xlim([-0.3 1])
title('Threshold-Crossing Waveform Snippets')
xlabel('msec')
ylabel('Amplitude (\muV)')
hold off
```



Clustering with the K-means algorithm

Implement the K-means algorithm and use it to determine the neuron responsible for each recorded spike. Treat each snippet as a point xn ? RD (n = 1, ..., N), where D = 31 is the number of samples in each snippet, and N is the number of detected spikes. We will assume that there are K = 2 neurons contributing spikes to the recorded waveform. Initialize the cluster centers using InitTwoClusters 1. Plot the objective fucntion (see notes), J, at each iteration and all waveform snippets assigned to each neuron

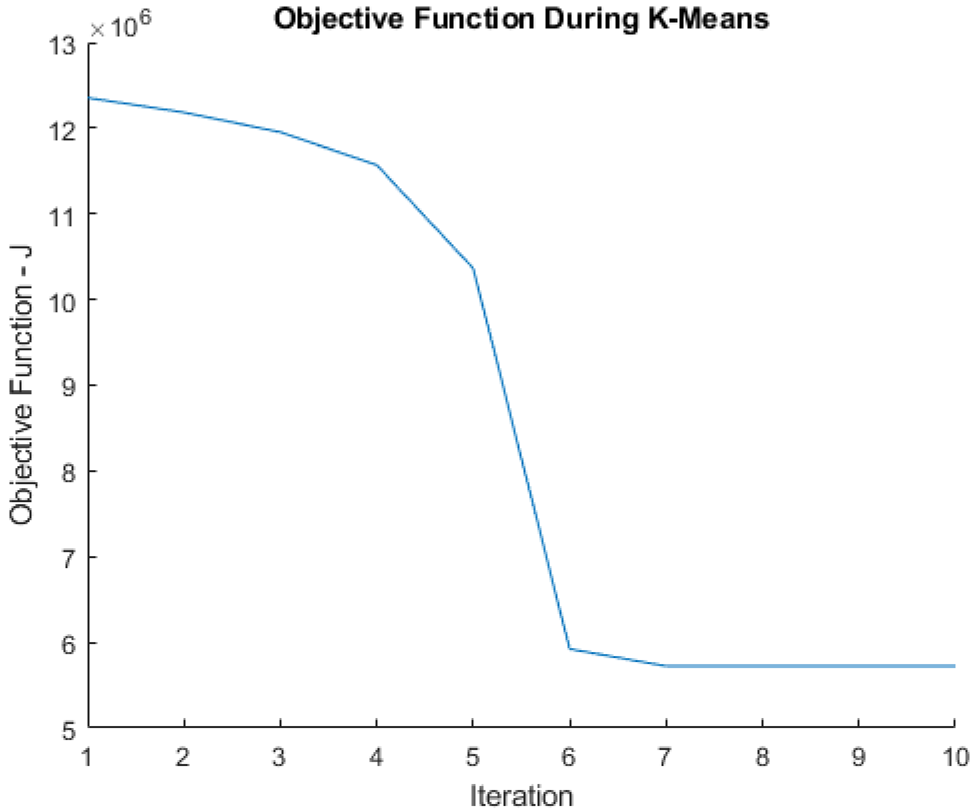
```
mu_1 = InitTwoClusters_1(:,1);
mu_2 = InitTwoClusters_1(:,2);
JM = zeros(1, 10);
for iteration = 1:10
    r_nk = zeros(length(cross),2);
    for n = 1:length(cross)
        spike = snippets(:,n);
        r = [sum((spike - mu_1).^2) sum((spike - mu_2).^2)];
        idx = find(r==min(r));
        r_nk(n,idx) = 1;
    end

    n_k1 = find(r_nk(:,1));
    n_k2 = find(r_nk(:,2));
    mu_1 = getMu(n_k1, snippets);
    mu_2 = getMu(n_k2, snippets);

    J = 0;
```

```
    for n1 = 1:length(n_k1)
        n = n_k1(n1);
        spike = snippets(:,n);
        J = J + sum((spike - mu_1).^2);
    end
    for n2 = 1:length(n_k2)
        n = n_k2(n2);
        spike = snippets(:,n);
        J = J + sum((spike - mu_2).^2);
    end
    JM(iteration) = J;
end
figure
hold on
plot(1:10, JM);
title('Objective Function During K-Means')
xlabel('Iteration')
ylabel('Objective Function - J')
hold off

plotWaveform(n_k1, mu_1, snippets)
title('Cluster 1 Waveform Snippets')
plotWaveform(n_k2, mu_2, snippets)
title('Cluster 2 Waveform Snippets')
```

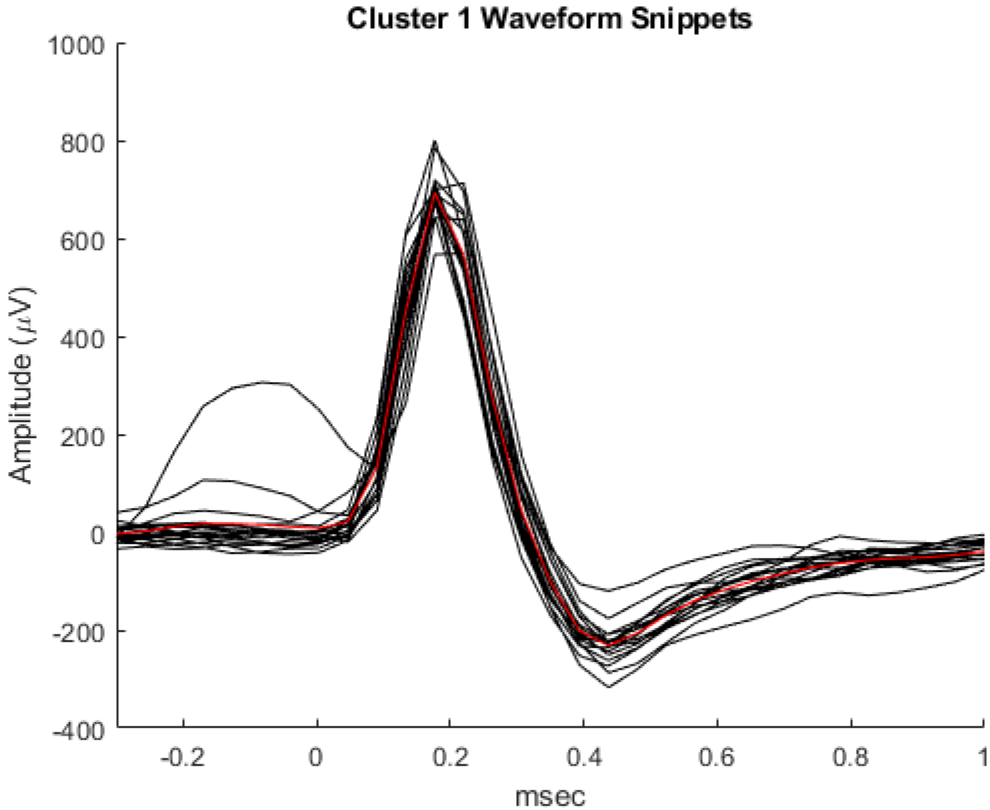
FUNCTIONS

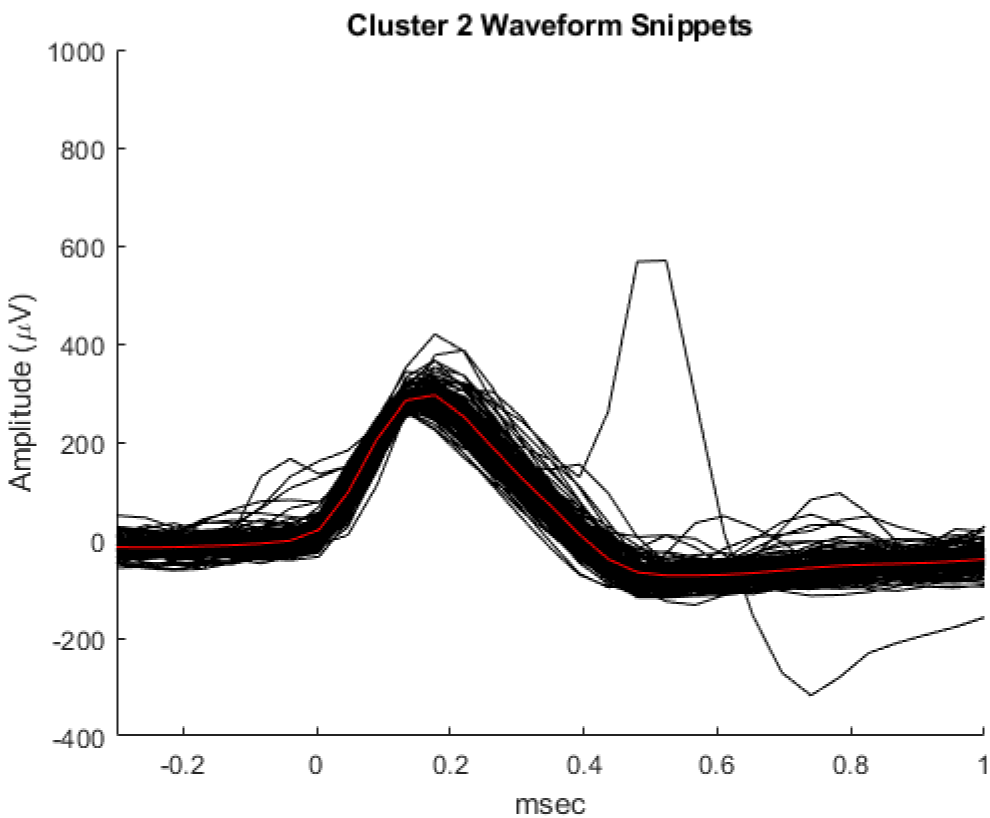
```
function [mu] = getMu(n_k, snippets)
% Calculates new cluster centers for each class
sum = zeros(31,1);
for i = 1:length(n_k)
    n = n_k(i);
    sum = sum + snippets(:,n);
end
mu = sum./length(n_k);
end

function [] = plotWaveform(n_k, mu, snippets)
% Plots the waveform snippets that belong to each class
% Red line represents cluster center (mu_k)
figure
hold on
t_snip = linspace(-0.3,1,31);
for i = 1:length(n_k)
    n = n_k(i);
    snip = snippets(:,n);
    plot(t_snip,snip','Color','black')
end
plot(t_snip,mu','Color','red')
xlim([-0.3 1])
ylim([-400 1000])
xlabel('msec')
ylabel('Amplitude (\muV)')
```

```
hold off
end
```



Objective Function During K-Means
×10^6
13
12
11
10
9
8
7
6
5
Objective Function - J
1
2
3
4
5
6
7
8
9
10
Iteration


Cluster 1 Waveform Snippets
Amplitude (μV)
msec
1000
800
600
400
200
0
-200
-400
-0.2
0
0.2
0.4
0.6
0.8
1


Cluster 2 Waveform Snippets
Amplitude (μV)
msec
1000
800
600
400
200
0
-200
-400
-0.2
0
0.2
0.4
0.6
0.8
1

Published with MATLAB® R2019b