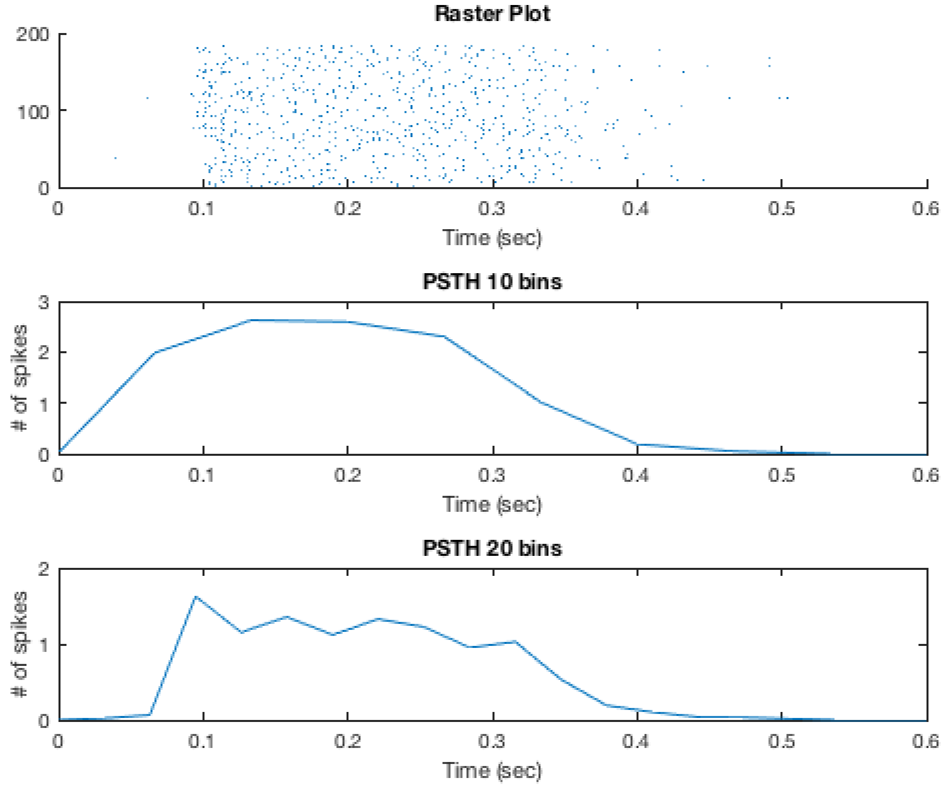
Table of Contents


```
load('mtSpikeTimes.mat');
```

1.1 & 1.2

```
figure; hold on
subplot(8,1,[1 2])
for i = 1:length(mtSpikeTimes)
    trial = mtSpikeTimes{i};

    for iid = 1:length(trial)
        hold on
        spkx=[trial(iid) trial(iid)];
        spky = [0 1] + i;
        line(spkx,spky,'LineWidth',1);
    end
end
xlabel('Time (sec)')
title('Raster Plot')

subplot(8,1,[4 5])
PlotPSTH(mtSpikeTimes, 10)
title('PSTH 10 bins')

subplot(8,1,[7 8])
PlotPSTH(mtSpikeTimes, 20)
title('PSTH 20 bins')

fprintf('As the number of bins increases, the "resolution" on the
 graph')
fprintf(' gets finer, which may mean the curve gets smoother if you
 start \n')
fprintf(' with too few bins, but may mean it gets more jagged if you
 go too')
fprintf(' high.')
```

As the number of bins increases, the "resolution" on the graph gets finer, which may mean the curve gets smoother if you start with too few bins, but may mean it gets more jagged if you go too high.



2.1

```
[mean_val, std_val, coef_var, fano, d] = GetStats(mtSpikeTimes, .03);
fprintf('For 30ms: \n')
fprintf('The mean firing rate is %4.5f \n', mean_val)
fprintf('The standard deviation of the firing rate is %4.5f \n',
 std_val)
fprintf('The coefficient of variation is %4.5f \n', coef_var)
fprintf('The Fano Factor of the counts is: %4.5f \n', fano)

clear mean_val std_val coef_var fano d

[mean_val, std_val, coef_var, fano, d] = GetStats(mtSpikeTimes, .3);
fprintf('For 300ms: \n')
fprintf('The mean firing rate is %4.5f \n', mean_val)
fprintf('The standard deviation of the firing rate is %4.5f \n',
 std_val)
fprintf('The coefficient of variation is %4.5f \n', coef_var)
fprintf('The Fano Factor of the counts is: %4.5f \n', fano)
```

For 30ms:

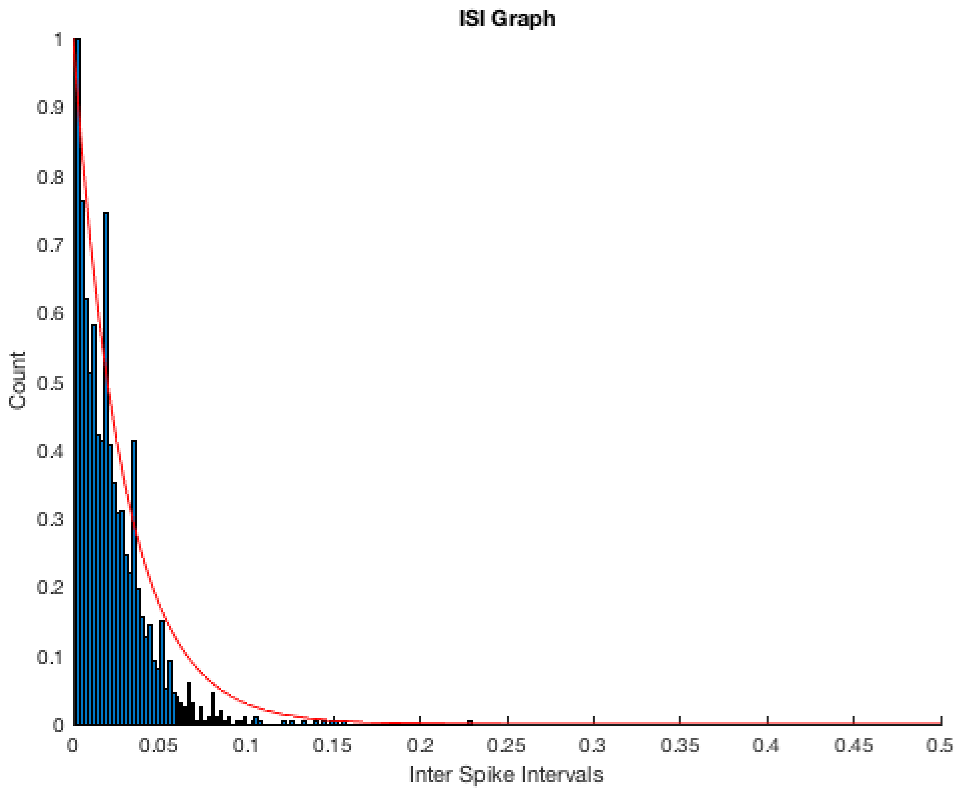
```
The mean firing rate is 81.52174
The standard deviation of the firing rate is 40.29766
The coefficient of variation is 0.49432
The Fano Factor of the counts is: 0.59760
For 300ms:
The mean firing rate is 35.19928
The standard deviation of the firing rate is 15.49300
The coefficient of variation is 0.44015
The Fano Factor of the counts is: 2.04578
```

2.2

ISI histogram

```
figure; hold on

%normalize histogram
[nums, cents] = hist(d, 100);
nums = nums / nums(1);
bar(cents, nums);
xlabel('Inter Spike Intervals')
ylabel('Count')
title('ISI Graph')


% fit exponential curve
x = linspace(0, .5, 100);
y = exp((-x)*mean_val);
plot(x,y, 'color', 'red');

fprintf('The curves match fairly well, the histogram appears to be')
fprintf(' following the exponential curve I added to the graph to ')
fprintf(' illustrate a Poisson distribution.')
```

```
The curves match fairly well, the histogram appears to be following
 the exponential curve I added to the graph to  illustrate a Poisson
 distribution.
```



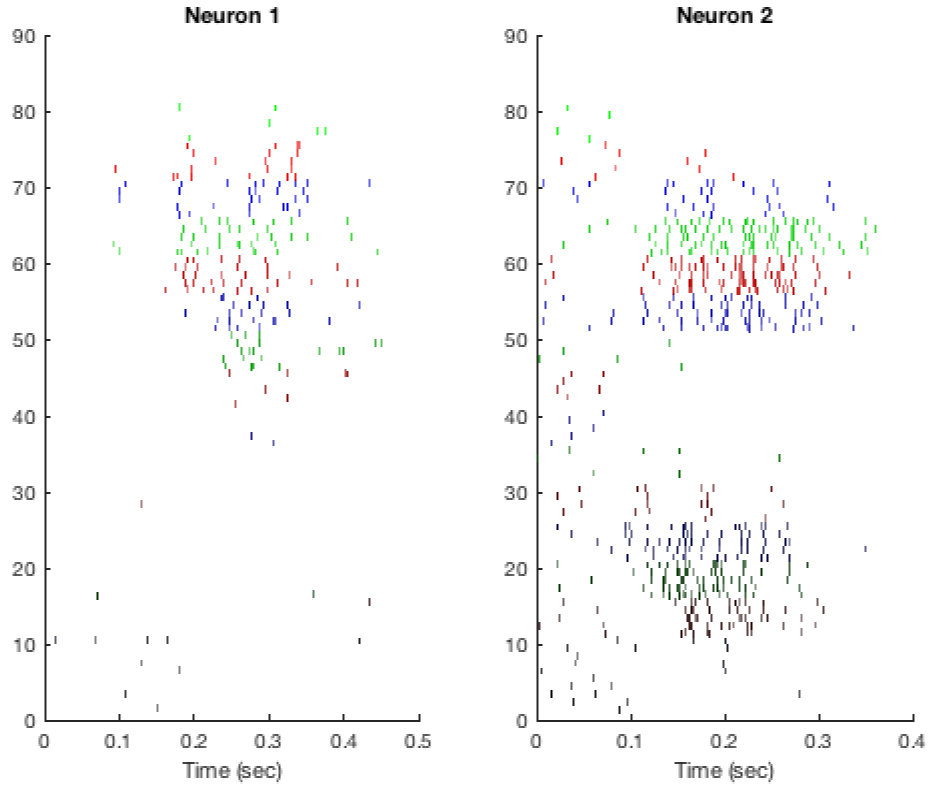

```
load('S1_Ideal_Observer_Analysis.mat');
```

3.1a

```
% preparing the data
amplitudes = Data.stimuli{1}(:,2,:);
directions = Data.stimuli{1}(:,3,:);

new_directions = [];
neuron_1 = {};
neuron_2 = {};
for i = 1:length(Data.spikes{1})
    if amplitudes(i) == 700
        neuron_1{length(neuron_1) + 1} = Data.spikes{1}{i};
        neuron_2{length(neuron_2) + 1} = Data.spikes{2}{i};
        new_directions = [new_directions directions(i)];
    end
end

% make a colormap
colormap = GetColormap(directions);
```

```
figure; hold on
```

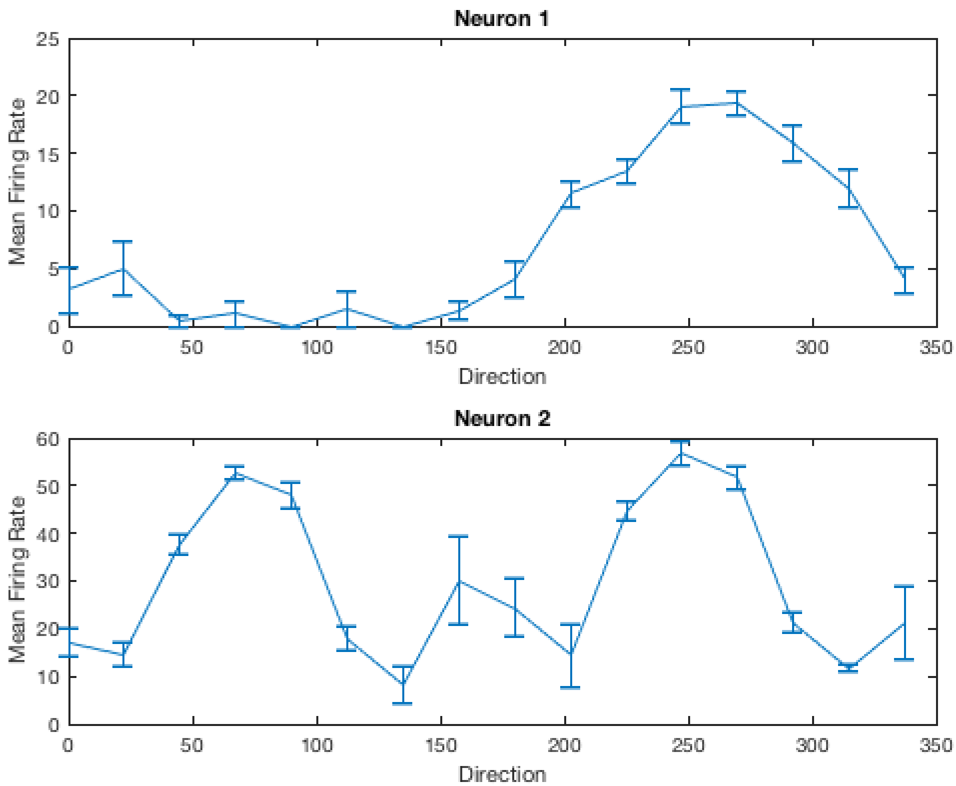
```
subplot(1,2,1)
RasterByDirection(new_directions, neuron_1, colormap)
title('Neuron 1');


subplot(1,2,2)
RasterByDirection(new_directions, neuron_2, colormap)
title('Neuron 2');
```



3.1b

```
figure; hold on

% sort trials by direction

trial_by_direction_1 = SortTrialsByDirection(new_directions,
 neuron_1);
trial_by_direction_2 = SortTrialsByDirection(new_directions,
 neuron_2);


% get the avg firing rates for each direction
% and plot. Get the preferred direction and sort firing rates
% by direction
subplot(2,1,1)
```

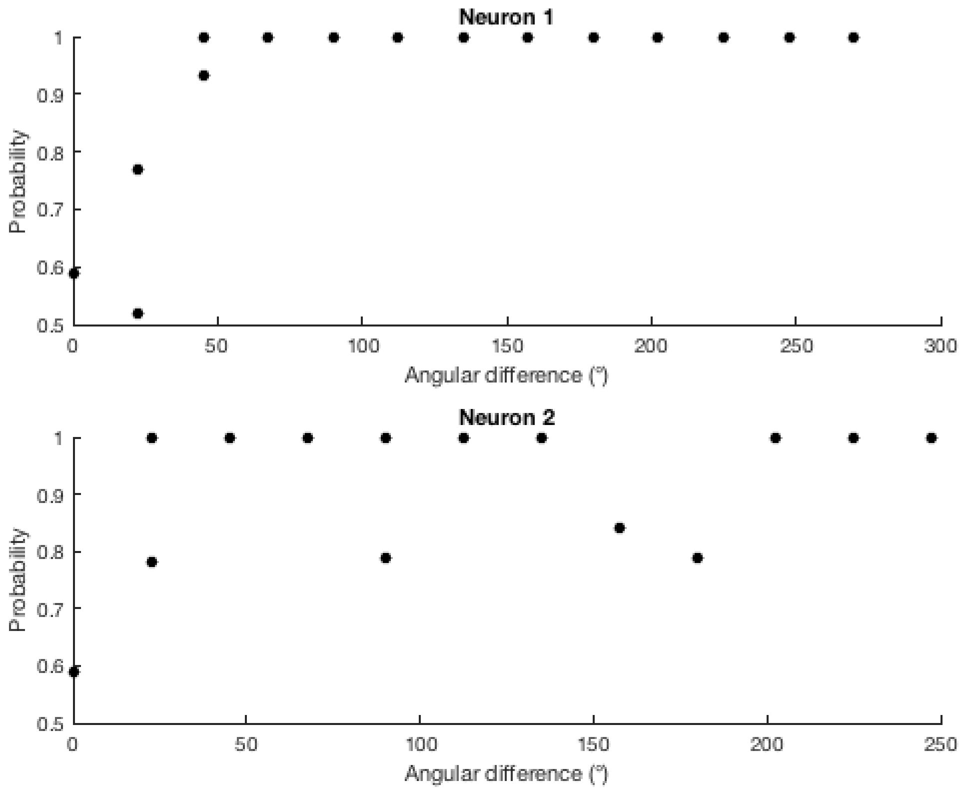
```
[preferred_1, rates_by_direction_1] = ...
    GetFiringRatesAndAvg(trial_by_direction_1);
title('Neuron 1')

subplot(2,1,2)
[preferred_2, rates_by_direction_2] = ...
    GetFiringRatesAndAvg(trial_by_direction_2);
title('Neuron 2')
```



3.1c

```
figure; hold on

subplot(2,1,1)
threshold_1 = PlotNeurometric(rates_by_direction_1, preferred_1, 1000)
title('Neuron 1');

subplot(2,1,2)
threshold_2 = PlotNeurometric(rates_by_direction_2, preferred_2, 1000)
title('Neuron 2')


fprintf('The threshold for both is around 22.5 \n')
fprintf('The second neuron is abnormal because it dips significantly')
fprintf(' below the threshold after it reaches it')
```

```
threshold_1 =

   22.5000


threshold_2 =

   22.5000

The threshold for both is around 22.5
The second neuron is abnormal because it dips significantly below the
 threshold after it reaches it
```

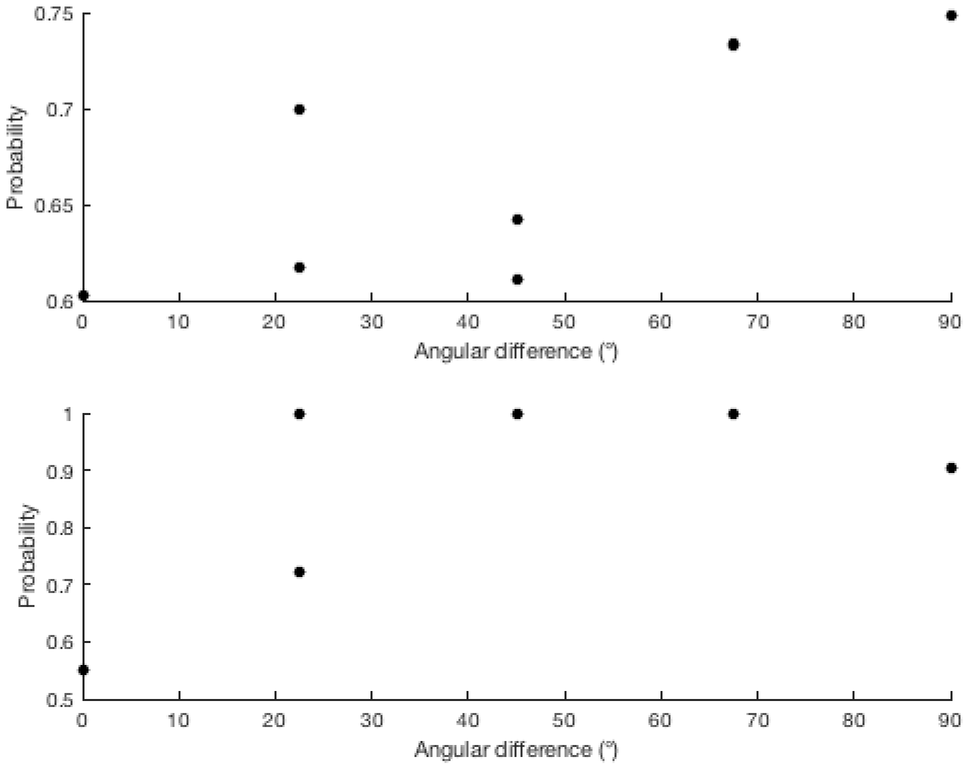


3.1d

```
[preferred_1, new_rates_1] = ResortByDirection(rates_by_direction_1);
[preferred_2, new_rates_2] = ResortByDirection(rates_by_direction_2);

figure; hold on

subplot(2,1,1)
threshold_1 = PlotNeurometric(new_rates_1, preferred_1, 10000)

subplot(2,1,2)
threshold_2 = PlotNeurometric(new_rates_2, preferred_2, 10000)
```

```
fprintf('The thresholds are: \n')
fprintf('Neuron 1: %2.1f \n', threshold_1)
fprintf('Neuron 2: %2.4f \n', threshold_2)
```

```
threshold_1 =

    90


threshold_2 =

   22.5000

The thresholds are:
Neuron 1: 90.0
Neuron 2: 22.5000
```



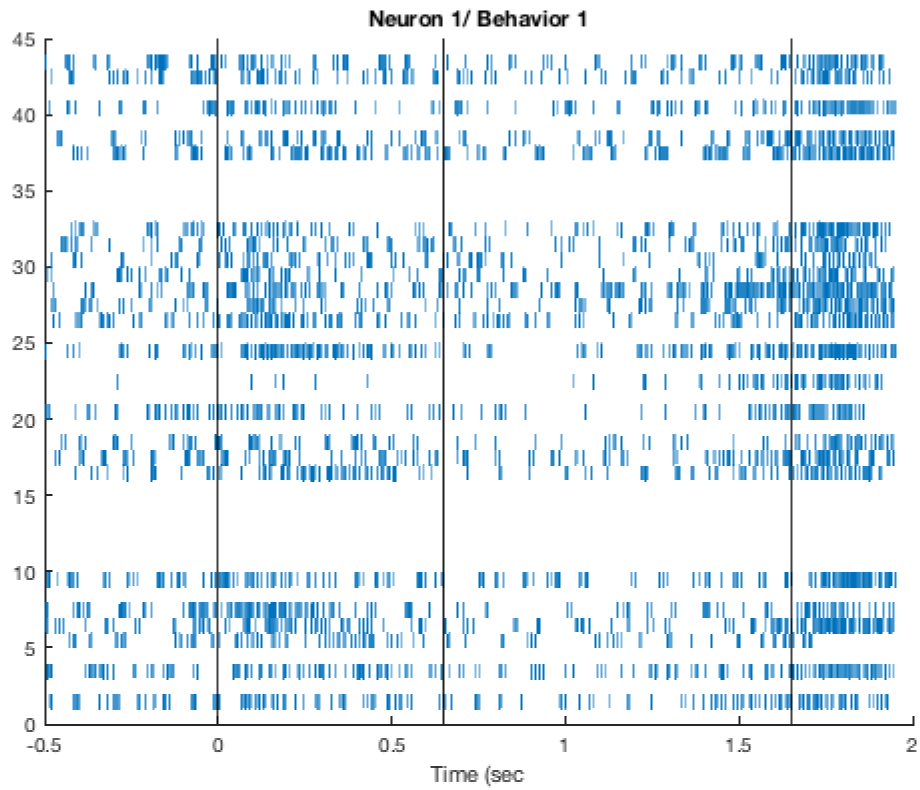
3.2

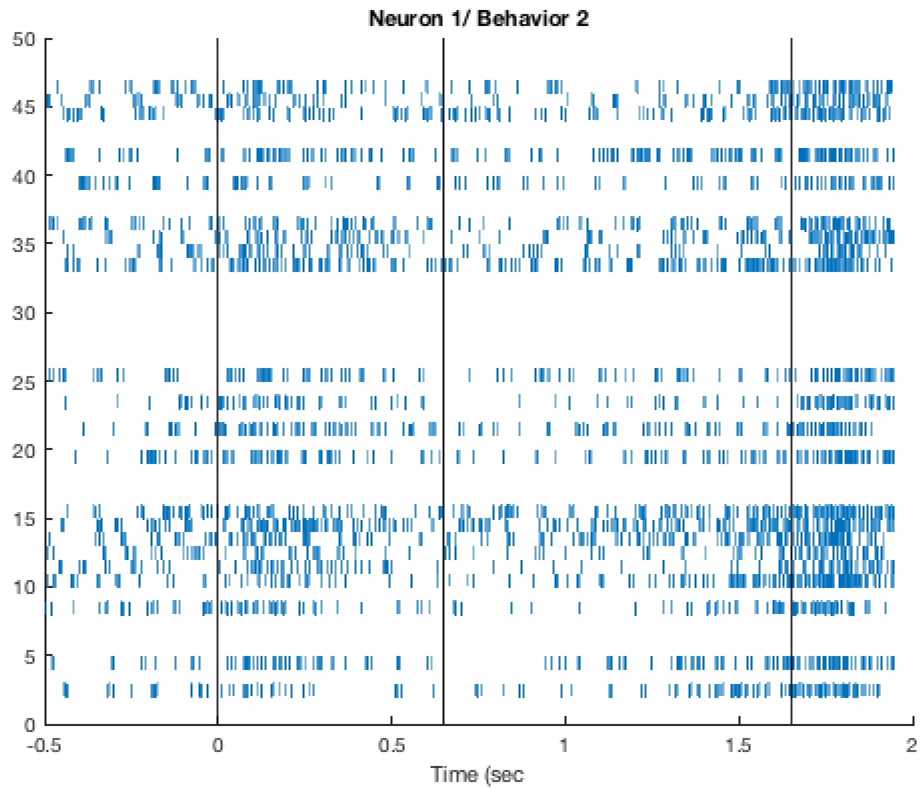
```
load('choiceData.mat')
```

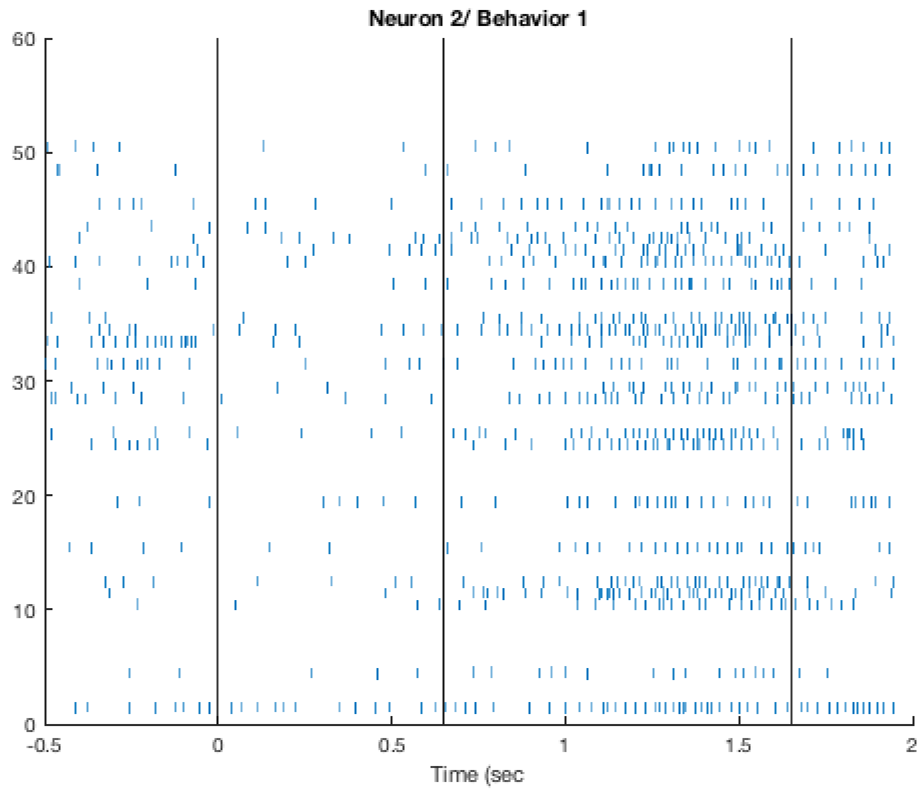
3.2a

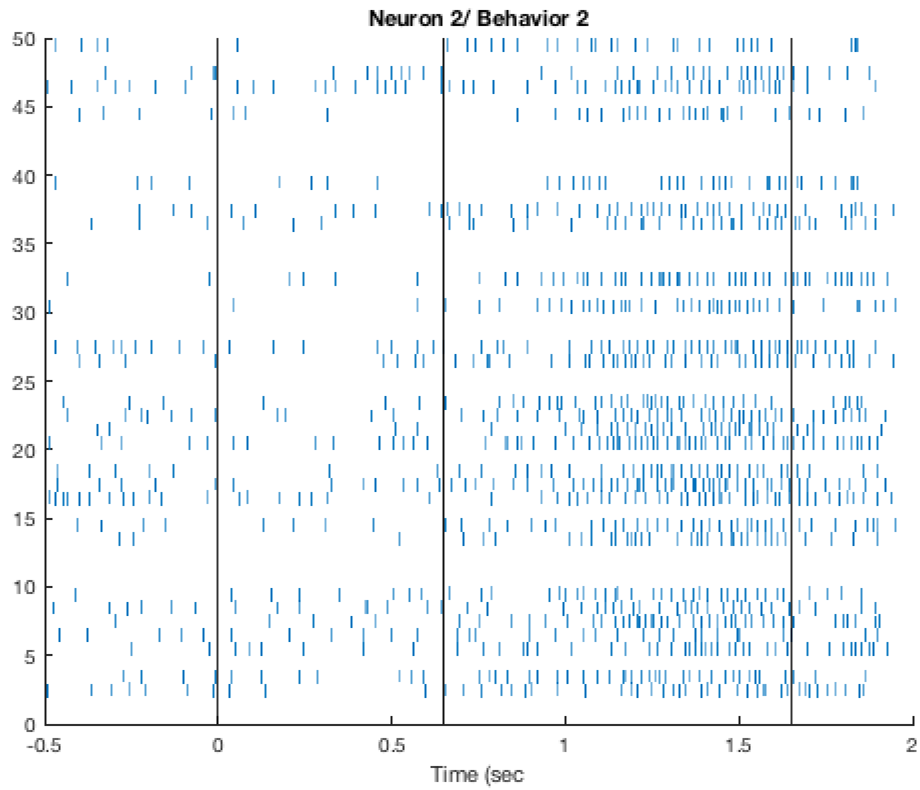
```
behaviors_1 = choiceData.behavioralReport{1};
```

```
behaviors_2 = choiceData.behavioralReport{2};
neuron_1 = choiceData.spikes{1};
neuron_2 = choiceData.spikes{2};

figure;hold on

RasterByBehavior(neuron_1, behaviors_1, 1)
title('Neuron 1/ Behavior 1')

figure; hold on
RasterByBehavior(neuron_1, behaviors_1, 2)
title('Neuron 1/ Behavior 2')

figure; hold on
RasterByBehavior(neuron_2, behaviors_2, 1)
title('Neuron 2/ Behavior 1')

figure; hold on
RasterByBehavior(neuron_2, behaviors_2, 2)
title('Neuron 2/ Behavior 2')
```




Neuron 1/ Behavior 2
50
45
40
35
30
25
20
15
10
5
0
-0.5
0
0.5
1
1.5
2
Time (sec


Neuron 2/ Behavior 1
60
50
40
30
20
10
0
-0.5
0
0.5
1
1.5
2
Time (sec



3.2b

```
p_1 = ChoiceProbabilities(neuron_1, behaviors_1, 1000);
p_2 = ChoiceProbabilities(neuron_2, behaviors_2, 1000);
```

3.2b

```
fprintf('The choice probabilities are: \n')
fprintf('Neuron 1 \n')
fprintf('Pretrial: %2.4f \n', p_1(1))
fprintf('Sample: %2.4f \n', p_1(2))
fprintf('Delay: %2.4f \n', p_1(3))
fprintf('Test: %2.4f', p_1(4))
fprintf('\n \n')
fprintf('Neuron 2 \n')
fprintf('Pretrial: %2.4f \n', p_2(1))
fprintf('Sample: %2.4f \n', p_2(2))
fprintf('Delay: %2.4f \n', p_2(3))
fprintf('Test: %2.4f \n', p_2(4))

fprintf('The choice probabilities in the pre-trial intervals are')
fprintf(' %2.4f and %2.4f \n', [p_1(1) p_2(1)])
fprintf('This means that even among the ambiguous trials, the ones
 which')
```

```
fprintf(' were coded with a 1 had a different distribution than those
\n')
fprintf('that were coded with a 2\n')
fprintf('In other words, the codings for ambiguous data were not
 randomly')
fprintf(' assigned')
```

```
The choice probabilities are:
Neuron 1
Pretrial: 0.3950
Sample: 0.4180
Delay: 0.5630
Test: 0.5200

Neuron 2
Pretrial: 0.4390
Sample: 0.4640
Delay: 0.6650
Test: 0.6120
The choice probabilities in the pre-trial intervals are 0.3950 and
 0.4390
This means that even among the ambiguous trials, the ones which were
 coded with a 1 had a different distribution than those
that were coded with a 2
In other words, the codings for ambiguous data were not randomly
 assigned
```

Published with MATLAB® R2017b