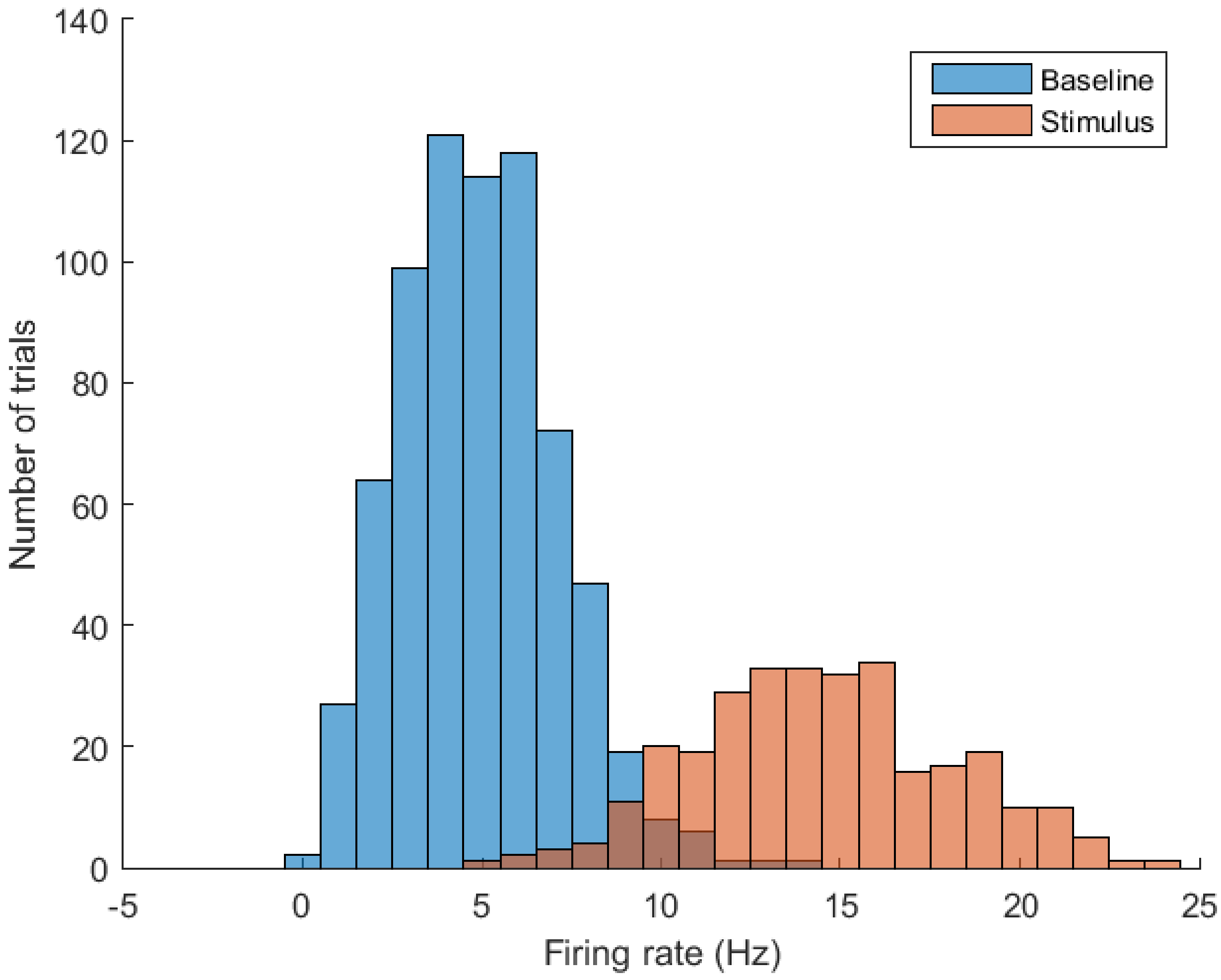
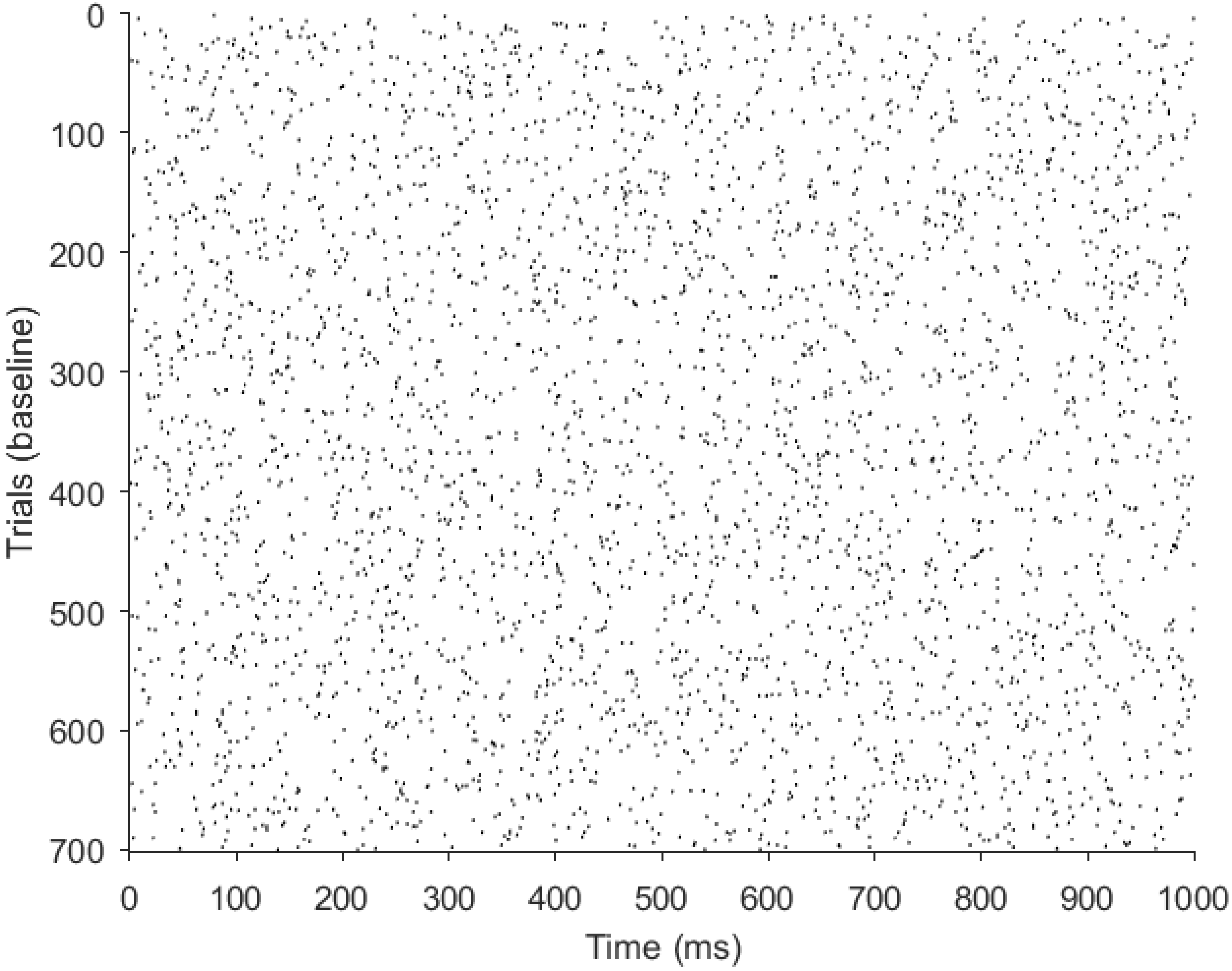
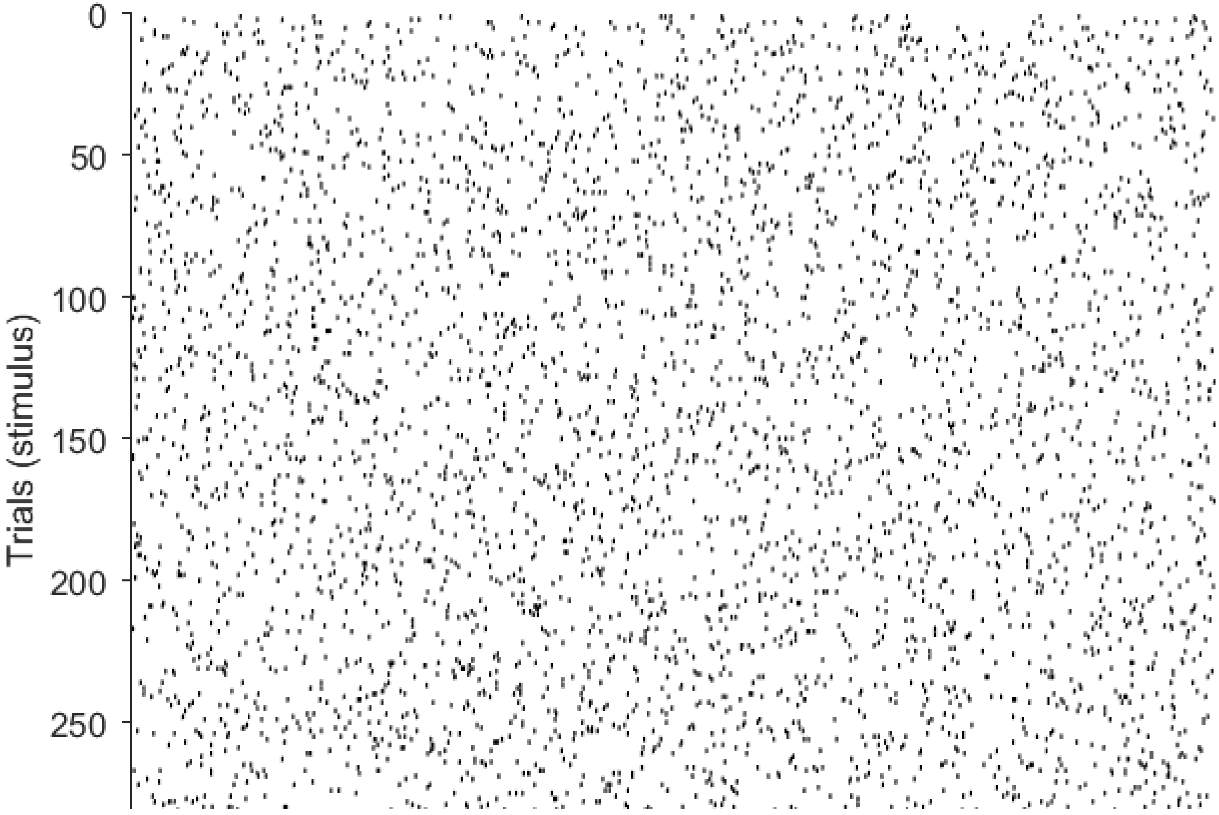
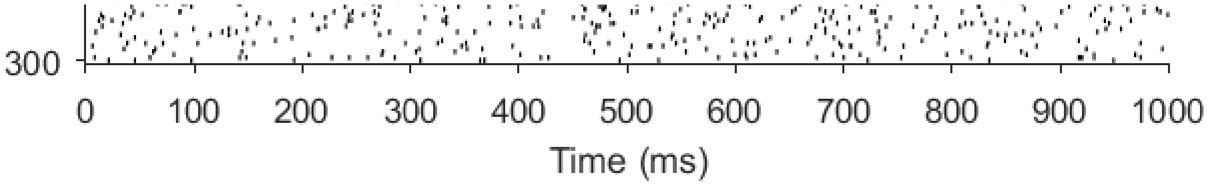
```matlab
%{
-----------------------------------------------------------
          Neural decoding for stimulus detection
                         Sam Qian
-----------------------------------------------------------

I simulate Poisson spike trains of one neuron, which has some baseline
firing rate and a higher firing rate in response to a stimulus. I
then use Baye's rule to determine the probability of a stimulus being
present using the probability density functions generated from simulated
data.

resources consulted:
https://praneethnamburi.wordpress.com/2015/02/05/simulating-neural-spike-trains/

Requires: plotSpikeRaster_v1.2

%}

clear all;
close all;
clc;


% Set neuron and trial parameters
tSim = 1; % simulation duration in s
nTrials = 1000; % number of trials to simulate
pS = 0.3; % probability of stimulus being present
fr_baseline = 5; % firing rate without stimulus, in Hz
fr_stimulus = 15; % firing rate with stimulus present, in Hz
nTrialsBaseline = round(nTrials*(1-pS)); % number of trials that are baseline
nTrialsStimulus = round(nTrials*pS); % number of trials that are stimulus


% Generate spikes
dt = 1/1000; % timestep (in s) in simulation (i.e. 1 ms)
nBins = floor(tSim/dt); % number of time bins for the given simulation duration
spikeTrains_baseline = rand(nTrialsBaseline, nBins) < fr_baseline*dt; % generate spike train
spikeTrains_stimulus = rand(nTrialsStimulus, nBins) < fr_stimulus*dt; % generate spike train


% Plot distribution of spike counts per sec (i.e. the firing rates of simulated
% trials)
for i=1:nTrialsBaseline;
    count_baseline(i,1) = (size(find(spikeTrains_baseline(i,:)==1),2))/tSim;
end

for i=1:nTrialsStimulus;
    count_stimulus(i,1) = (size(find(spikeTrains_stimulus(i,:)==1),2))/tSim;
end

figure;
hold on
h1=histogram(count_baseline);
h2=histogram(count_stimulus);
xlabel('Firing rate (Hz)');
ylabel('Number of trials');
legend('Baseline','Stimulus');
```



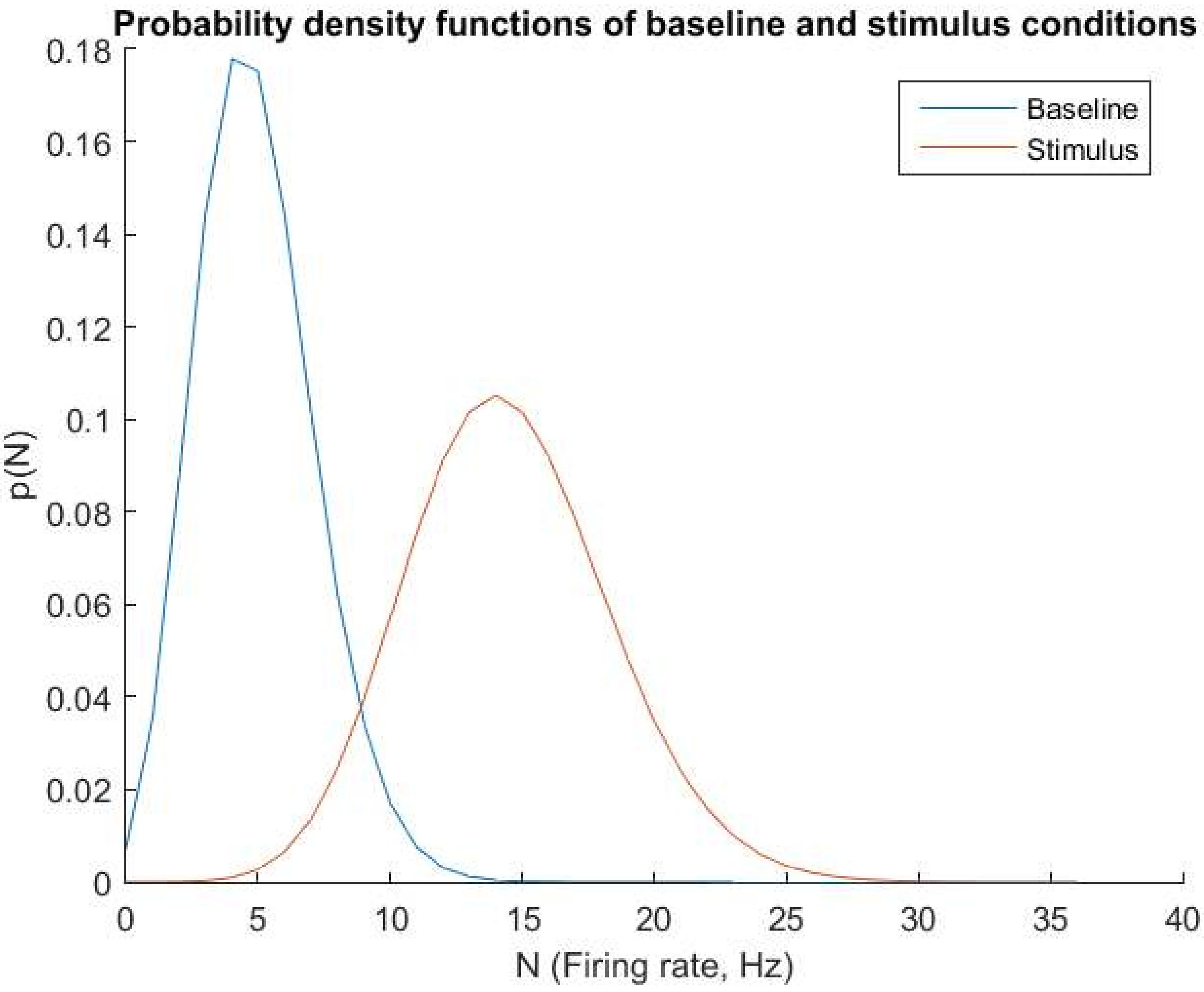
```
% Raster plots of stimulated spike trains
figure;
LineFormat.LineWidth = 0.7;
LineFormat.Color = [0 0 0];
plotSpikeRaster(spikeTrains_baseline,'PlotType','vertline','LineFormat',LineFormat);
xlabel('Time (ms)');
ylabel('Trials (baseline)');

figure;
LineFormat.LineWidth = 0.7;
LineFormat.Color = [0 0 0];
plotSpikeRaster(spikeTrains_stimulus,'PlotType','vertline','LineFormat',LineFormat);
xlabel('Time (ms)');
ylabel('Trials (stimulus)');
```


Trials (baseline)
0
100
200
300
400
500
600
700
0
100
200
300
400
500
600
700
800
900
1000
Time (ms)

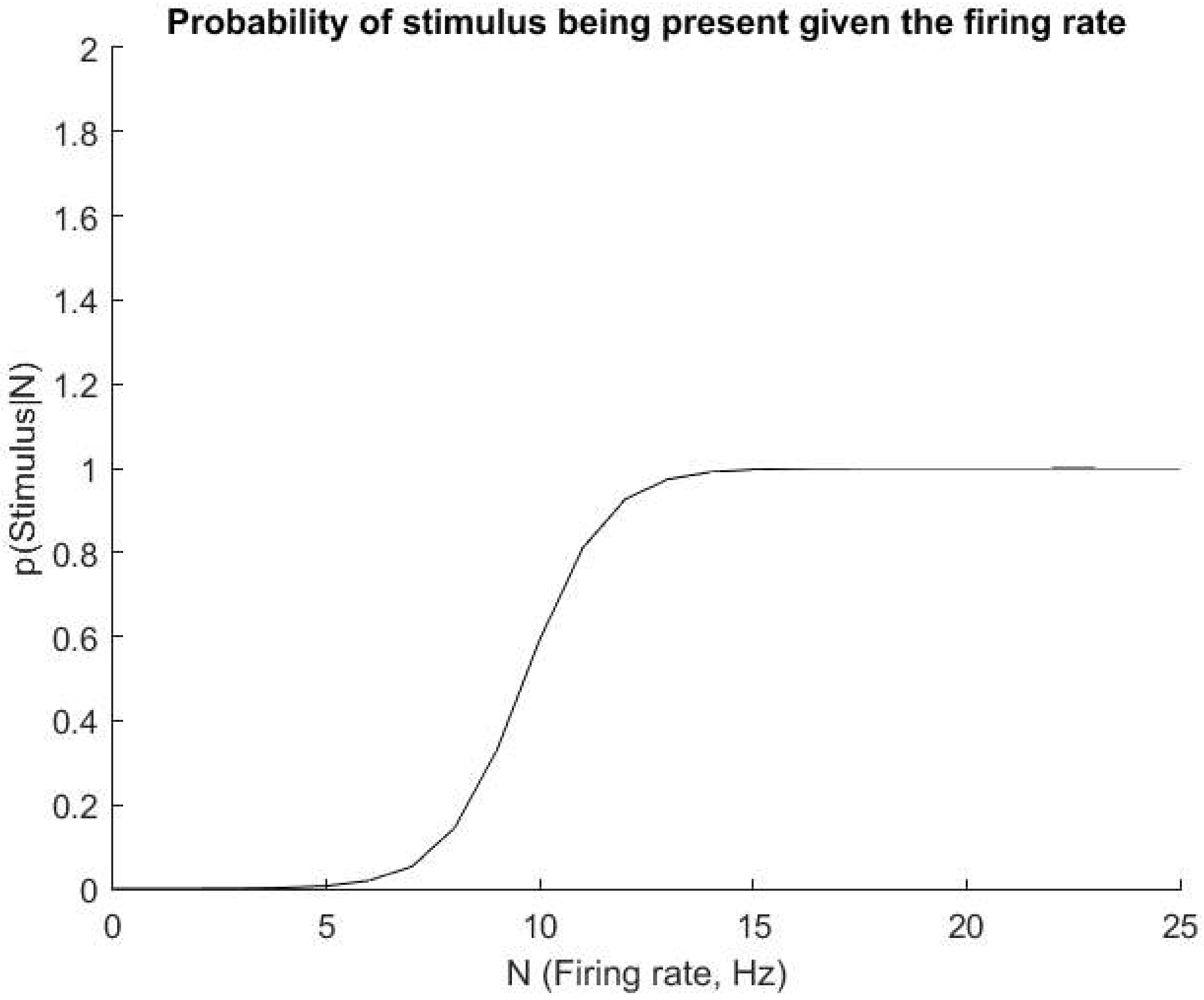

Trials (stimulus)
0
50
100
150
200
250



```
% Use simulated spike train to estimate lambda to re-derive probability
% density functions
lambdahat_baseline = poissfit(count_baseline);
lambdahat_stimulus = poissfit(count_stimulus);

pd_baseline = makedist('Poisson',lambdahat_baseline);
pd_stimulus = makedist('Poisson',lambdahat_stimulus);

x = [0:max(count_stimulus)*1.5];
y_baseline = pdf(pd_baseline, x);
y_stimulus = pdf(pd_stimulus, x);

% Plot pdf derived from stimulated spike trains
figure;
hold on;
plot(x,y_baseline);
plot(x,y_stimulus);
xlabel('N (Firing rate, Hz)');
ylabel('p(N)');
legend('Baseline', 'Stimulus');
title('Probability density functions of baseline and stimulus conditions');
```



```
% Bayesian probability of stimulus present given a firing rate
posterior = (y_stimulus*pS)./(y_stimulus*(pS)+y_baseline*(1-pS));

figure;
hold on;
plot(x, posterior,'k');
xlabel('N (Firing rate, Hz)');
ylabel('p(Stimulus|N)');
title('Probability of stimulus being present given the firing rate');
xlim([0 25])
ylim([0 2])
```



Published with MATLAB® R2015a