

---

# Table of Contents

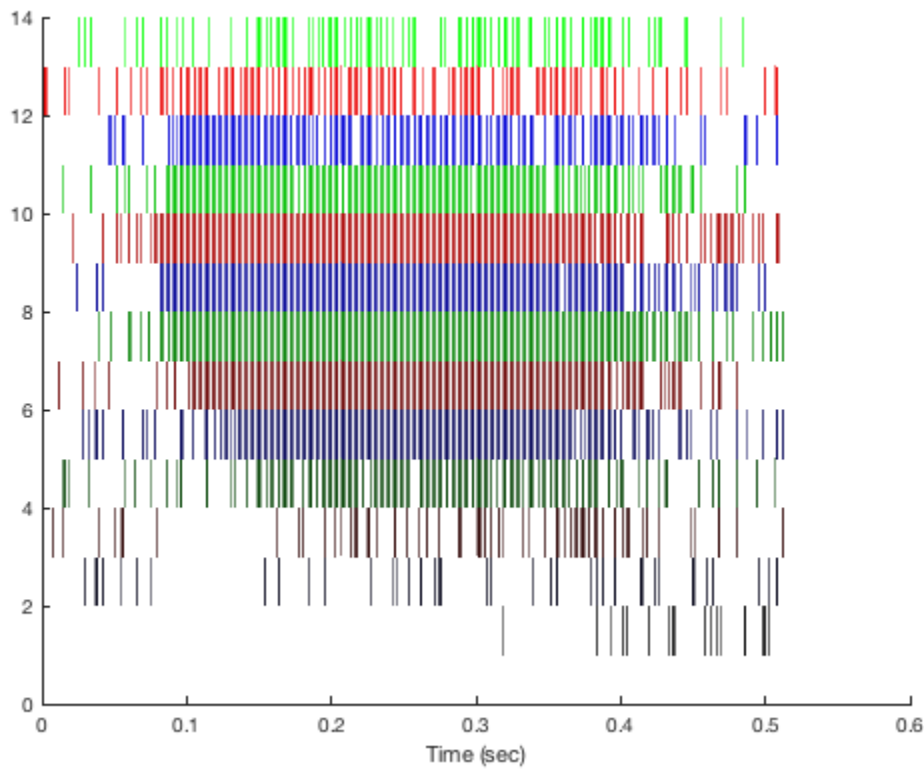
.....	1
1 .....	1
2 .....	2
Convert to array .....	2
3 .....	3
Bonus: Error bars .....	4

```
load('mtNeuron.mat')
dirs = mtNeuron.dirs;
times = mtNeuron.time;
data = mtNeuron.data;
```

# 1

```
cmap = GetColormap(mtNeuron.dirs);

figure; hold on
RasterByDirection(dirs, data, cmap, times);
```



---

## 2

```
counts = cumsum(data, 1);

dirProbs = {};

for i=1:length(dirs)
    dirProbs{i} = GetProbs(counts, i, dirs);
end
```

## Convert to array

```
probArray = ConvertToArray(dirProbs, counts);

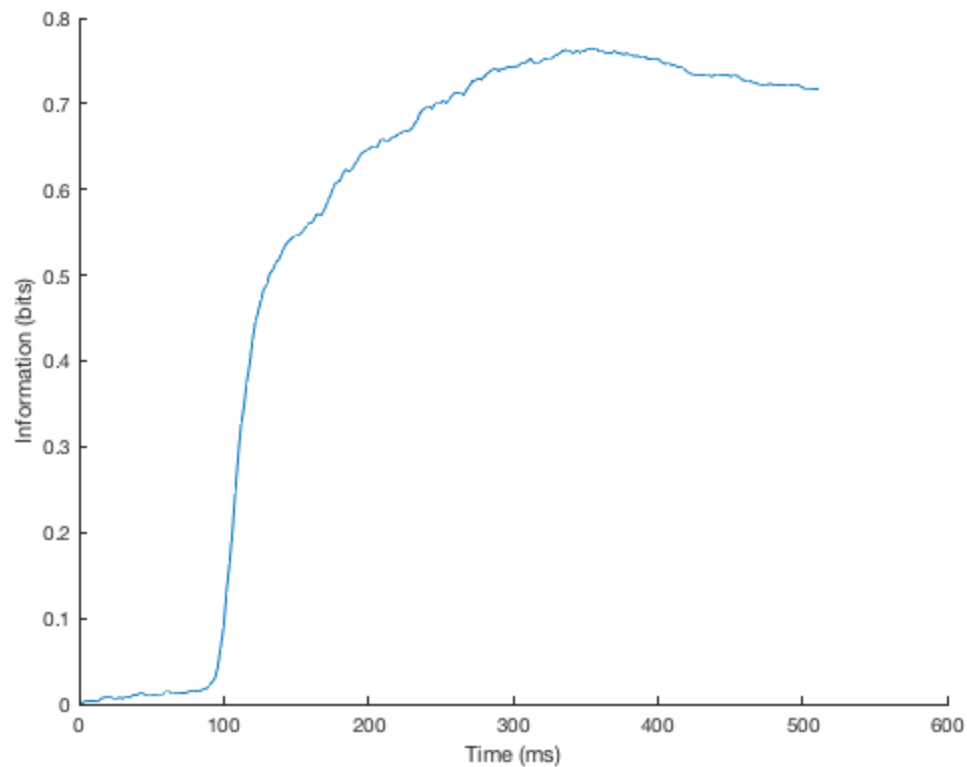
% Get PTn

P_T_n = getCountProbs(probArray);

% get mutual information and plot

mutual = MutualInf(probArray, P_T_n);

figure; hold on
t = times * 1000;
plot(t, mutual)
xlabel("Time (ms)")
ylabel("Information (bits)")
```



### 3

```
% to determine latency, we calculate the amount of time before the
maximum
% amount of information was available

[v max_ind] = max(mutual);

t_max = times(max_ind);

fprintf("The latency is %1.5f seconds\n", t_max)

fprintf("The proportion of information available 50ms after neural ")
fprintf("response was %1.5f\n", getPercentBefore(mutual, times, .05))

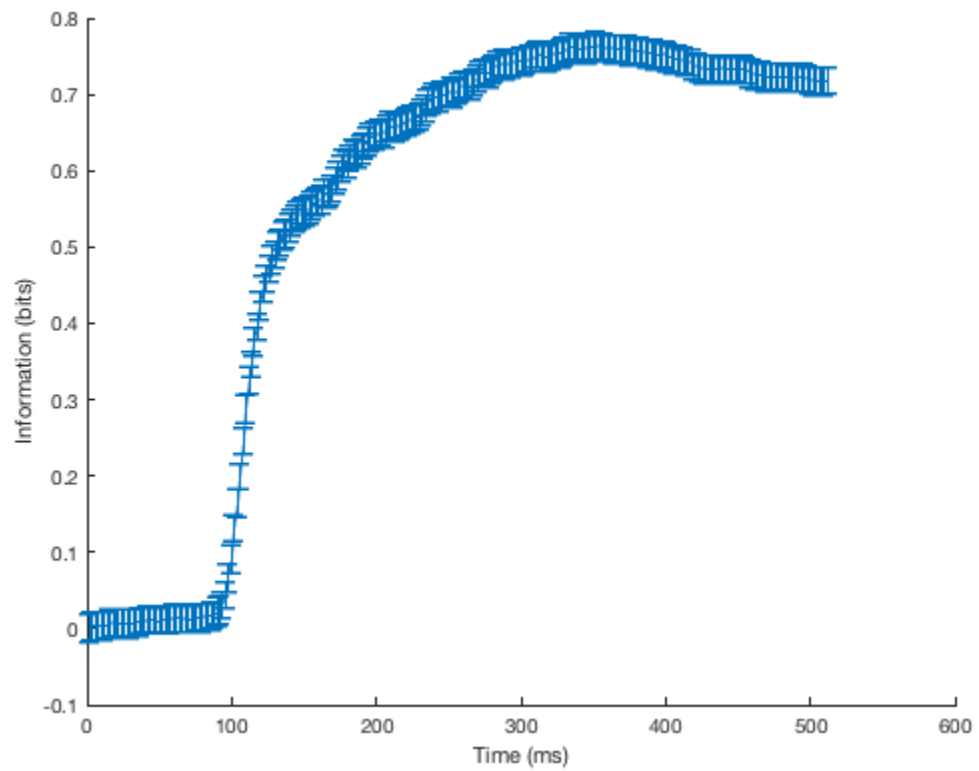
fprintf("The proportion of information available 100ms after neural ")
fprintf("response was %1.5f\n", getPercentBefore(mutual, times, .1))

The latency is 0.35200 seconds
The proportion of information available 50ms after neural response was
0.77189
The proportion of information available 100ms after neural response
was 0.90417
```

---

## Bonus: Error bars

```
% get standard error  
  
err = std(mutual) / sqrt(256) * ones(1,256);  
  
figure; hold on  
% plot with standard error  
errorbar(t, mutual, err)  
xlabel("Time (ms)")  
ylabel("Information (bits)")
```



*Published with MATLAB® R2017b*