Assignment #8: Analysis of Neuronal Imaging

Recent advances in optical sensing and stimulation are placing optical technologies as either complementary or potentially viable alternatives to electrode-based devices for similar purposes. In this homework, we will explore analysis of optical imaging data to monitor neuronal activity in mouse cortex. We will use a raw two-photon imaging data set where images were continuously acquired over time in a single location (known as time series imaging; 1200 images acquired at 5 frames-per-second). The goal of the homework is to pre-process the data and use clustering to identify differences (or similarities) in neuronal activity from the imaged location. The field-of-view covers 400um x 400um at a depth of 240um from the cortical surface (a small portion of the mouse brain).

Individual neurons encode activity in time and form inter-connected networks in space. To extract the information that reflects the activity of neurons, our analysis will focus on motion correction and clustering. For this exercise you will need the data file “tmp_tser8683_raw.mat” and the following Matlab functions: “imMotionCorrect”, “imshift2”, “im_smooth”. In the data set provided, all images are contained within the “data” variable. This variable has dimensions of (x, y, Ch, T), where x and y form the image acquired at time T. At each time T, two images were acquired with different sensitivities; images in channel 1 contain red fluorescence from blood vessels and images in channel 2 contain green GCaMP fluorescence. Recall from class that this sensor becomes brighter when neurons that have it increased their spiking activity.

Based on this information, put together a Matlab script that answers the following questions. Make sure you also include written answers for the questions that do not require Matlab code. A checklist is included at the end to verify all the necessary deliverables for this assignment.

1. Load the matlab file and show the image sequence for the green fluorescence images recorded in channel 2 using the pseudo-code below. Repeat it for channel 1 using the same code (but with chno=1). Answer the following questions (related to what you see): (1) Do you see neurons? (2) How many neurons are active? (3) Are all of them doing the same thing? (4) From observation, how many independent groups of neurons (clusters) are there? (5) Is there motion in this data set? (6) How much motion would you say there is in units of pixels? Place your answers in the caption of a figure where you show two panels, on the left panel the image of frame #30 from channel 2, and on the right panel the difference between frame #680 and #30 from channel 2. Repeat this for channel 1 images. Please add appropriate titles and color bars.

```
load tmp_tser8683.mat

figure(1), clf,
colormap gray,

imagesc( data(:,:,2,30) ), axis image, colormap gray, colorbar,
imagesc( im_smooth( data(:,:,2,30), 0.6) ), axis image, colormap gray, colorbar,

myminmax=[0 1];          % try [0.05 0.65]
fps=5.0;
chno=2;
t=[0:size(data,4)]/fps;
for mm=1:size(data,4),
   imagesc( squeeze(data(:,:,chno,mm) ), myminmax),
   % OR try this
   % imagesc( im_smooth(data(:,:,chno,mm), 0.6), myminmax),
   axis image, colorbar,
   xlabel([‘Time= ‘,num2str(t(mm)),’sec,  Frame #’, num2str(mm)])
   drawnow,
end

imagesc( data(:,:,2,30) - data(:,:,2,680) ), axis image, colormap gray, colorbar,

imagesc( data(:,:,1,30) - data(:,:,1,680) ), axis image, colormap gray, colorbar,
```

2. Calculate the average GCaMP image in channel 2 (green) and display it in a figure. You can use the “mean” command in Matlab for this keeping in mind that time is in dimension 4 of the data variable. Make sure the image is square, the color scale is gray and a colorbar is included. Please add an appropriate title. You can adjust the myminmax values to help you see how many cells are there (for example, try myminmax=[0 1]; or, myminmax=[0.05 0.55];).

3. Generate a motion corrected data set of the GCaMP images (channel 2 only) using the function imMotionCorrect.m; to see how it works, type “help imMotionCorrect”. This function uses the method discussed in class based on cross-correlation and sub-pixel interpolation. Make sure you select an image of your choosing as reference; however, select the first image for the final result. Make sure that you capture both outputs from this function (the estimated motion and the corrected data set). Plot the estimated motion for each image using the plot command. Is the result reasonable? Is it similar to the amount of motion you estimated by eye?

4. Re-run the movie in step #1 using the corrected images to see whether the motion was indeed reduced or eliminated, was it? (You can double check this step by repeating it using the images in channel 1). Calculate the average GCaMP image in channel 2 using the motion corrected data and display it in a figure. Make sure the image is square, the color scale is gray and a colorbar is included. Please add an appropriate title. Compare it with the image you generated in #2. Are they different in any way?

5. Now that the data set has been corrected for motion, let’s extract the time series of each cell in the data set. Each cell has been segmented for you and provided in the variable maskC included in the mat-file. Extract the change in intensity from each segmented cell in maskC. Then, generate an image of the cell mask and plot the time series from some of the cells. You can use the following pseudo-code to guide you.

```
figure,
imagesc(maskC),
axis image, colormap jet, colorbar,

for mm=1:size(data_corrected,3),
   tmpim=data_corrected(:,:,mm);
   for nn=1:max(maskC(:)),
      data_C(mm,nn)=mean(tmpim(find(maskC==nn)));
   end
end
```

6. Next, generate a mask for a region-of-interest that contains the “neuropil” (the area that likely contains axons and dendrites but no cell bodies). For this, we want to exclude the dark areas in the GCaMP image which are occupied by blood vessels. Calculate the average blood vessel image (channel 1 data). Generate a blood vessel mask by applying a threshold to this image (for example, vessel_mask = (vessel_image > thr);). Alternatively, you can use the dark areas as the vessel mask (for example, vessel_mask = gcamp_image < 0.2). Then, generate a cell body mask using the variable maskC (for example, cell_mask = (maskC > 0);). Generate the neuropil_mask using both the vessel_mask and cell_mask (note that you will need to invert these using (~cell_mask)&(~vessel_mask)). Generate a figure of your neuropil mask. Finally, extract the neuropil time series using the code below and plot it. Compare this plot with the plot from #5.

```
gcamp_image = mean( data(:,:,2,:), 4);
vessel_image = mean( data(:,:,1,:), 4);

for mm=1:size(data_corrected,3),
   tmpim=data_corrected(:,:,mm);
   neuropil(mm)=mean(tmpim(find(neuropil_mask)));
end
```

7. Now, let’s use k-means to explore potential clusters in the data. Use the pseudo-code below to test several different of clusters. Select the number of clusters you think best represents the data and justify your selection based on the image of the clusters and their average time series.

```
nk = 6;
kk = kmeans( data_C.’ , nk, ‘distance’, ‘correlation’ );

kim = zeros( size(data,1), size(data,2) );
for m=1:max( maskC(:) ),
        kim(find(maskC==m)) = kk(m);
end

figure, clf,
imagesc(kim), axis image, colormap jet, colorbar,

for m=1:max( kk(:) ),
        tck(:,m) = mean( data_C(:, find(kk==m)), 2);
end

figure, clf,
for m=1:nk, subplot(3,3,m), plot( t, tck(:,m) ), axis tight, grid on, drawnow, end

figure, clf,
imagesc( corrcoef( tck ) ), axis image, colormap jet, colorbar,
```

Extra-credit:

8. Did motion correction have an impact of the number of clusters selected? Can you justify your answer based on using the original data for cluster extraction instead of motion corrected data?

9. Next, let's use PCA to explore potential clusters in the data. Just like discussed in class, we can use PCA to segregate the changes in GCaMP intensity from different pixels or cells into principal (or orthogonal) components. To do this, we need to perform PCA where the imaging data has 2 dimensions (time along the rows and space along the columns). Recall that PCA assumes the data is zero-mean, so we will also remove the mean intensity from the data. Plot the mean intensity of all images over time, then plot the first 12 vectors of U.

```
datasz=size(data);
clear data

newdata = reshape(data_corrected,[datasz(1)*datasz(2) datasz(3)]);
avgdata = mean(newdata, 2);
newdata = newdata - repmat(avgdata, [1 datasz(3)]);

[U,V,W] = svd( newdata );
% if your computer states insufficient memory, use this command instead
% [U,V,W]=svd(newdata, ‘econ’);
```

10. Do you see any common relationships between the extracted components (U) and any of the plots you have generated so far?

11. Display the coefficient images that correspond to the first 12 components. You can use the following pseudo-code.

```
figure,
for mm=1:12,
  subplot(3,4,mm),
  imagesc( reshape(W(:,mm), [datasz(1) datasz(2)]) ),
  axis image, colormap jet, colorbar,
end
```

12. Given what you have done so far, how many clusters do you estimate there are in these data and why?

Extra-extra-credit:

13. How do these clusters help you understand the data that electrodes record?

14. Calculate the average signal from all blood vessels (images in channel 1) and plot the intensity over time.

15. Are the changes in blood vessel intensity similar to any of the time series you calculated above (or to the average intensity over time from Ch#2 images)? What would you expect? How is this similar or different to your expectation?

Submission Checklist

1. Answer the 6 questions referenced in #1 after observing the imaging data
2. Figure of the average GCaMP image in Ch#2
3. Plot the estimated motion parameters
4. Figure of the average GCaMP image in Ch#2 after motion correction
5. Figure of cell mask and plot cell time series
6. Figure of neuropil mask and plot neuropil time series
7. K-means cluster image and average time series for each cluster

Extra-credit Checklist

8. Impact of motion on cluster selection
9. PCA components
10. Component information, what do you see
11. Images of coefficients
12. Estimate the #clusters based on PCA result

13. Do data like these help you understand electrode recordings?
14. Average image in Ch#1
15. Average time series signal from Ch#1, how does it compare to anything you have done so far and/or to the average time series signal from Ch#2?