UNIVERSITÄT BASEL, MRSIC-FLOGEL LAB

Ca^{++} Imaging Analysis in MATLAB

April 28, 2016

In this practical you will analyse 2-photon imaging data using MATLAB. This is the same approach neuroscientists use to analyse data in their studies. You should complete all steps by modifying the MATLAB functions you have been provided.

You will be supplied with with raw Ca^{++} imaging data from mouse V1. The mouse was anesthetized and was stimulated with drifting black and white grating stimuli, drifting in multiple different directions. From these data we can obtain a direction and orientation tuning curve for each cell in the field of view. Your task will be to to extract the raw fluorescence intensity trace from one cell, convert this to dF/F, and then determine the dF/F for each stimulus in order to produce a polar plot that shows response magnitude for different drift directions. You will then be able to select different cells and quickly pass the extracted response timecourses through the functions you have written to get tuning curves for several different cells.

While you work through the exercise, write your responses to the questions in a word or text document. You will need to save several figures, as well as the MATLAB functions you write to solve the exercise.

Remember that typing `help function_name` in MATLAB will give you a lot of information about how to use the function.

1 START MATLAB AND DOWNLOAD THE DATA

- Download the data needed for this practical from *http://mouse.vision/ca.zip* and unpack the zip archive.
- Start MATLAB (menu Applications/Education/Matlab).

check menu

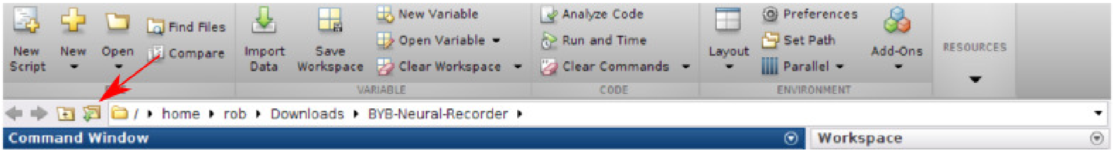


Figure 1.1

- Use the change directory button at the top of the screen to navigate to the directory containing the files you just unzipped (cf. figure 1.1).
- Run `addToPath` function in the command line to get a working environment.

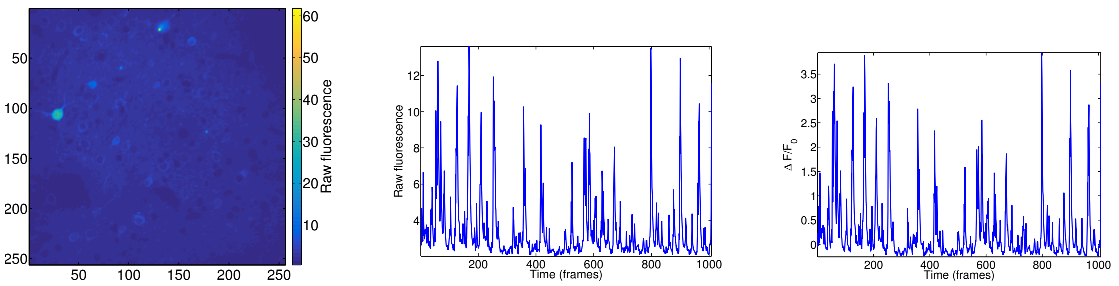
2 Extract response time-course from a cell

First, you will load images, select a cell and display its raw time series, as follows:

- Load the image stack called `Calcium_imaging_data_int8.tif` using the provided `load_stack` function. Do not forget to use `help load_stack` to see how to use this function.
- What is the size of the loaded stack (image width and height in pixels, number of frames)? **Write** down your answer.
- Calculate the average image using the `mean` function and assign it to a variable called `meanIM`.
- Plot the image using the `imagesc` function (see Figure 2.1a). **Save** your image.
- Look at the image. What does this projection tell you about the activity of different neurons? **Write** down your answer.
- Start ROI selection GUI using the `get_roi_trace` function. On the new figure, draw an ellipse around a cell to highlight it. Double-click on the ellipse, and the average time-trace of the pixels within your ROI will be returned.
- Plot the resulting time course using `plot` function (see Figure 2.1b). **Save** your image.

Now you will compute the *change* in fluorescence over time (dF/F).

- Open the provided file `calc_dF_F.m`. This function gets as input the raw trace from one cell and returns as output the dF/F. **Edit** the function to calculate and return the



(a) Mean projection

(b) Raw fluorescence trace

(c) $\Delta F/F_0$ time trace

Figure 2.1

dF/F, as follows:

1. Calculate $dF/F = (F - F_0)/F_0$. F_0 is the median of the fluorescence (F) distribution. Calculate this using the `median` function.

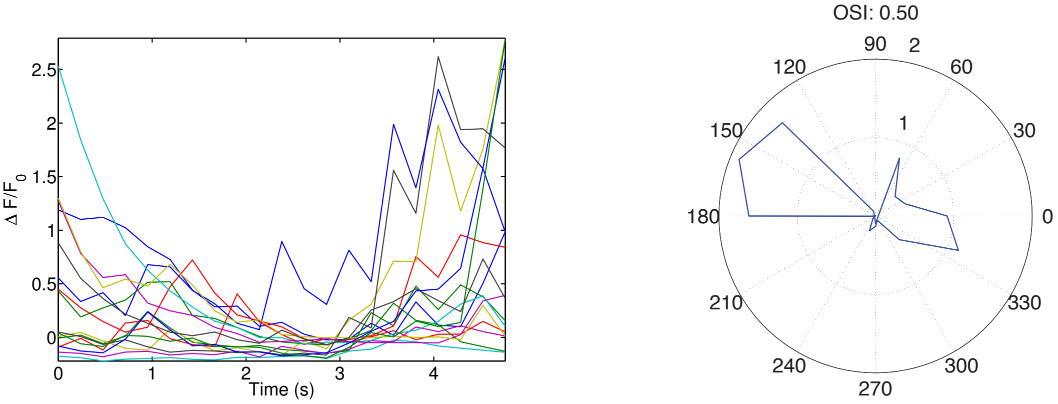
2. Subtract F_0 value from each fluorescence (F) value, and then divide the resulting value by F_0.

- Run your `calc_dF_F.m` function on the raw trace and plot the dF/F using `plot` (see Figure 2.1c). **Save** your image.

3 Understanding the stimulus presentation paradigm

In this dataset, each of the 16 stimulus drift directions was presented 3 times (The drifting grating was presented for 1.5 seconds preceded by a gray screen presented for 3.5 seconds). In the following, you will compute the average response timecourse over the multiple repetitions of the same stimulus.

- **Draw** below what this stimulation paradigm looks like over time, by indicating the time in seconds for baseline and for stimulus presentation, for a single presentation of a drafting grating.



(a) Average response, all orientations

(b) Polar plot of average response

Figure 3.1

- How many frames are in one stimulus presentation (blank+stim)? What is the imaging frame rate? **Write** down your answers and **add** the number of frames on your drawing above.

- The drift direction in degrees presented for each stimulus frame is saved in the file `ori_stimuli.mat`. Load these data using the `load` function. **Write** down the name of the loaded variable containing the orientation information:

- Open the provided file `meanTraces.m`. This function will be used to split up the response into presentations of the same directions, over the three trials per direction, and average over trials. **Complete** the file `meanTraces.m`. Pay attention to the comments in the file.

- Average the three individual trial traces for each direction using your `meanTraces` function. Plot these average traces for each of the 16 stimulus directions on the same plot (see Figure 3.1a). **Save** your image.

- Plot a single average trace over all stimuli. Where does the peak fall with respect to the stimulus start time? Refer to your diagram, above.

Should they write something on the diagram?

4 Calculate responses to different stimuli

You will now find the mean response to each of the 16 stimulus orientations.

- Open the file `makePolarPlot.m`. This function will be used to display the mean response for each orientation. **Edit** the function as follows:

1. Choose a window of 5 time points that should contain the peak response to a stimulus, during the stimulus period.

2. Average the 5 time points within this window, from the average timecourse of each stimulus orientation.

3. Plot the mean responses on a polar plot, which will reveal the orientation tuning of the cell, using the `polar` function (see Figure 3.1b).

- Why is a polar plot a better choice than a conventional x/y line or scatter plot? **Write** down your answer.

- Use the `makePolarPlot` function to plot the preferred orientation(s) of your cell. **Save** your image.

- Is your cell tuned to the drift direction of the stimulus? Is your cell tuned to the drift direction of two stimuli of the same orientation? **Write** down your answer.

To assess the selectivity of your cell, you will compute its *orientation selectivity index* (OSI).

- **Edit** the file `calc_osi.m` as follows:

simplify `mod` use in the script?

1. Find the stimulus direction that caused the biggest response ('preferred' stimulus).

2. Find the mean response of the preferred stimulus, and the stimulus of opposite drift direction (180 degrees away), and average the two values.

3. Find the mean response of the two stimuli that are 90 degree away from the preferred stimulus ('orthogonal' stimuli), and average the two values.

4. Orientation selectivity index (OSI) is calculated as
$OSI = (preferred - orthogonal)/(preferred + orthogonal)$

- Add the OSI to your polar plot, using the `title` function.

re-save the image?

Question about negative $OSI \in [0;1]$? Problems if dF/F is negative?

5 Repeating for other cells

You should now have a series of files that you can call in sequence to get a polar plot: `calc_dF_F.m` then `meanTraces.m` then `makePolarPlot.m` then `calc_osi.m`. If you select a new cell, you can pass it through these functions to quickly generate a polar plot.

- **Create** a new function called `cellbatch`, in a file called `cellbatch.m`. Provide as an input argument the response time course. In the function body call the functions you have made in order. Passing the correct inputs and outputs to each so that you can feed `cellbatch` function a response time course and get back a polar plot.
- Run `cellbatch` for 5 different cells and **save** the resulting plots.