**Overview**

This document describes a procedure for registration, deconvolution, and analysis of two-photon data in the Goldberg Lab. The pipeline was adopted heavily from Mike Vaiana in Sarah Muldoon’s lab and tailored to fit CHOP infrastructure and be more flexible for our needs.

The entire process involves the following programs:

* ImageJ (with TurboReg plugin) – for registration and ROI selection
* MATLAB – for extracting traces
* Python – for deconvolution/data analysis

Briefly, from raw .tif frames captured by the 2p camera, we take the following steps:

1. Save raw output from Image Block Ripping Utility into the “Raw” folder of a predetermined file structure (described later).
2. Register using batchRegister.txt using a predefined batch size (usually 2500 or 5000), which is an ImageJ macro. This can be done on the 2p computer for maximum efficiency.
3. Save resulting registered files, and temperature trace (if there is one), on a network drive, in a “Movies” subdirectory using a predetermined file structure (described later).
4. Using ImageJ, manually select ROIs and save these to an “ROIs” subdirectory in a defined file structure (described later).
5. Using ImageJ, take an average frame of each batch and save this, as well as a .csv describing the offset of this average frame from the first average frame, in an “Images+Offsets” subdirectory using a defined file structure (described later). (For more information, see “Background” section below)
6. Using MATLAB, extract traces from each movie using the movie, ROIs, and average images+offsets
7. Using Python scripts, deconvolute and analyze data to extract key features.

All files needed to process the pipeline are split into the following high-level functionalities:

1. Directories containing binaries for actual programs already installed (e.g., MATLAB or ImageJ). ImageJ macros are stored with the ImageJ installation for easier running. A copy of ImageJ will be included in the “Repos” folder.
2. A “data” folder containing all movies, average images, ROIs, and other images (described in more detail below). As of 9/24/2018 this is Z:\Raymond\TrpM2-Cre
3. A “scripts” folder, containing all MATLAB scripts, Python scripts, as well as supporting packages and final raw traces. I will upload this in final form to Github. As of 9/24/2018 this is Z:\Raymond\Calcium Imaging Analysis\Vaiana

**Background**

The Goldberg Lab’s primary interest is in analyzing 2p data to determine the effect of interneuron subtype or other variables on firing dynamics. We split this goal into the following subproblems:

* identifying separate subpopulations of neurons in a single image through differential fluorescence expression
* identifying individual cells as regions of interest
* extracting meaningful traces from movie
* aligning extremely long movies by correcting for xy drift
* batchwise analysis capability (our machines have limited RAM)

Our first step is to register our raw movie of n frames. Keeping in mind subproblem 5, we register this movie in b batches with a batch size of n/b. Unfortunately, we cannot register each movie to the first frame in the movie; this would result in increasingly inaccurate registrations as the movie progressed. Instead, we register each frame to the first frame in each batch, such that no frame is more than n/b frames away from the frame it’s being registered to. Empirically this has shown to have good results.

Next, we need to manually select ROIs in ImageJ. We do so using an average image of the first batch. ROIs are saved as coordinates loadable by MATLAB. If we have multiple populations of neurons, each population of neurons is saved as a separate ROI file.

Our next problem arises in that separate batches, because they were registered to separate images, will have a fixed offset from each other in the xy plane. The best way to correct this is to accordingly offset our ROI coordinates for each batch. We will do this in during the trace extraction step, but for now we need to register the average image of each batch (each n/b frames) to the average image of the first batch, then save the average images and resultant offsets of those registrations in a .csv file for later use.

Finally, we will extract traces in MATLAB. For each movie, the script will process each batch in turn, extracting traces using the ROIs with batch-appropriate offsets. If there are multiple sets of ROIs for the movie (multiple neuronal populations), traces for each ROI set will be collected. Finally traces from each batch will be stitched together to form traces for the entire movie.

Following this, we run a set of Python scripts that uses OASIS to clean traces, create spike trains, and extract features from our raw .csv files.

**File Structure**

Many of our scripts assumes a certain file structure, especially for the data folder. It is described in detail here. In general the folder structure has the following format: (top level)\(data category)\(MouseFolder)\(batchFolder)\(files: images, stacks, rois, etc.)

* Top level: this contains exactly 5 subdirectories: Images+Offsets, Movies, ROIs, Raw, and Supplementary. **As of 9/24/18 this is Z:\Raymond\TrpM2-Cre-Data. This path will heretofore be referred to as “toplevel”**
  + Images+Offsets: subdirectories of this level have a mouseFolder naming scheme (described below), with one subfolder per imaging session.
    - MouseFolder: naming scheme is described below
      * BatchFolder: naming scheme is described below
        + Files: usually, one .tif image containing the batch-averaged image, and a .csv file that is the output of TurboReg’s batchwise average-image registration
  + Movies: this is the bulk of the data, contains motion corrected movies.\
    - MouseFolder: naming scheme is described below
      * BatchFolder: naming scheme is described below
        + Files: usually, a single tiff stack with the entire batch of motion-corrected images
  + ROIs: ImageJ output from ImageJ-based manual ROI selection.
    - MouseFolder: naming scheme is described below
      * Files: (no BatchFolder since ROIs are the same across batches) .zip files, directly outputted from ImageJ. If there are multiple distinct ROI sets, each will have its own zip file and be named accordingly.
  + Raw: a copy of non-motion corrected movies before TurboReg registration
    - Files: just all the files in the native structure produced by the image block ripping utility. Will have subfolders for channel, as well as singleimage, Tseries, etc. Also contains the temperature log for the imaging session.
  + Supplementary: anything session-specific of note not mentioned above
    - MouseFolder: naming scheme described below
      * Files: anything good, e.g. I saved average TrpM2+ images here

**MouseFolder naming scheme:** The naming scheme for MouseFolder-level subdirectories is as follows: [NAME]–[MM.DD.YYYY]–MOUSE-[MOUSENUMBER]. An example would be RAYMOND–05.07.2018–MOUSE-1193. It is important to adhere to this convention or risk breaking the code. **Make sure to use periods and not hyphens or backslashes in the date. Do not use spaces in the name as this will be parsed as a separate argument in the command line input to ImageJ.**

**BatchFolder naming scheme:** You should never have to name BatchFolder-level subdirectories on your own; they should always be created for you. However, they should follow a convention like “blahblahblah\_registered[indexIn]\_to\_[indexOut]”. All indices should have the same number of digits, with leading zeros filled in if necessary. This will also break the code if not followed.

* e.g. “registered00001\_to\_10000” and not “registered1\_to\_10000”.

**Step 1: Initial Data Collection and Saving**

**Input:** n/a – this is the initial data collection step.

**Involves:** PrairieView microscope, Image Block-Ripping Utility

**Output:** .tif files named something like “blahblahblahCh1000001.ome.tif”

After data collection with the 2p, a folder containing frames separated by session should be saved in **toplevel/Raw**. The folder should be named according to the MouseFolder naming convention described above. After data collection, use the PraireView’s accompanied block ripping utility to turn these files into .tifs interactable with ImageJ.

**Step 2: Registration**

**Input:** output of step 1

**Involves:** macro “batchRegister.ijm” inside /macros folder of Fiji installation – located in toplevel\..\Calcium Imaging Analysis\Vaiana\Fiji.app\macros

**Output:** output images named “registered[index].tif” located by batch inside folders named “registered[in]\_[out].tif.

Registration involves aligning images to cancel out translational motion from headbar wobble.

Within ImageJ, open and run the batchRegister.ijm macro (first edit it to the appropriate batchSize and WriteDirectory). It may fail on the first run, I can’t figure out why. Running it again will make it work.

You need to choose the batch size for registration for memory purposes – I usually use 5000, which works well on the microscope PC. You should set the readDirectory as the folder that was the output of step 1. Note that if the number of frames is not an integer multiple of batchSize, the remainder will not be registered.

**Step 3: Averaging + Offsetting**

**Input:** output of step 2

**Involves:** macros calculateAverages.ijm and calculateOffset.ijm in macros folder of Fiji installation – located in toplevel\..\Calcium Imaging Analysis\Vaiana\Fiji.app\macros (12.5.2018 – may choose combine these two into one macro calculateAverages.ijm, just as a note in case I forget to update this! –RL)

**Output:** .tif files arranged in a MouseFolder-level subdirectory in toplevel\Images+Offsets, containing batch folders corresponding to batches from Step 2’s output, with each batch folder containing an average .tif file named “AVG\_Ch4\_registered00001\_to\_50000.tif” (for example), as well as a file named offset.csv containing offset of batch average from the first batch’s average.

Step 2 registered each image in batches, with each batch’s images being registered to the first image in each batch. This creates systematic average offsets between batches that determine here. Briefly, we calculate the average image for each batch, then run TurboReg to output the offset required to align the average image of each batch with the average image for the first batch. This offset is saved as a csv file and the image average is saved as a .tif. The macros do this systematically for every batch in a MouseFolder.