

# **FP3002 Manual - Data Analysis**

Summer 2023

Versioned 2023-05-01



## Table of Contents

---

|   |    |
|---|----|
| <b>Preface</b>                                  | 3  |
| Pivot Off the Biology                           | 4  |
| Experimental Goals                              | 6  |
| <b>Data Format</b>                              | 8  |
| Understanding Frame Flags                       | 11 |
| <b>Introduction to Analyzing Data in MATLAB</b> | 12 |
| Importing Data into MATLAB                      | 13 |
| Visualizing Data                                | 14 |
| <b>Strategies for Further Processing</b>        | 18 |
| Correcting for photobleaching                   | 19 |
| Create Peri-Event Plots                         | 22 |
| Spike Detection                                 | 25 |
| Note on dF/F                                    | 27 |

## Preface

The goal of this manual is to provide you with the necessary tools to interpret and analyze fiber photometry data. It should be relevant to novice and seasoned programmers.

Keep these things in mind as you read through each section:

1) Pivot off the biology:

What is the underlying biological construct you are looking to measure? In what ways is your measurement the same or different from said construct?

2) Pivot off of your experimental goal(s):

There are many levels of analysis / rigor with which you can approach your experiment. If your analysis is too cursory, this will leave you open to both false positive and false negative results. If your analysis is too rigorous, you may end up performing unnecessary controls or diving down a rabbit hole that is tangential to your end goal.

## Pivot Off the Biology

---

The central premise of photometry as a technique is to measure a fluorescence signal as a proxy for a biological event.

For example, the ability of GCaMP to fluoresce – that is, absorb a “blue” photon and emit a “green” photon – is heavily affected by whether or not it is bound to a calcium ion or not. In an ideal case, GCaMP behaves like GFP when it is bound to calcium and does not fluoresce at all when it is not bound to calcium.

So, if we keep the level of “blue” excitation light constant, the variation in our “green” emission signal should be correlated with the concentration of calcium wherever our indicator is expressed.

However, in systems neuroscience, calcium *concentration* is often not the biological construct researchers are interested in measuring. Instead, they are interested in firing rate or spiking.

*How is internal calcium concentration related to spiking?*

In general, the intracellular calcium concentration in neurons is maintained at the nanomolar level. As the resting membrane potential of neurons is negative, there is both a concentration gradient as well as an electrochemical gradient which leads to a strong driving force into the cell.

When a neuron fires an action potential (AP), the resting membrane potential transiently becomes positive. This opens voltage-sensitive calcium channels which results in an influx of calcium into the cell. This “extra” calcium is extremely tightly regulated – with free calcium being effluxed out of the cell or scooped up by the endoplasmic reticulum.

*What are the relevant timescales here?*

An AP occurs on the order of 1ms. Calcium concentration begins increasing *after* an AP and persists for ~10-100ms.

Fluorescent indicators have binding kinetics that slow our detection and extend our perceived duration of this change in calcium concentration to ~100ms to >1s.

Calcium concentration versus firing *rate*: As we discussed, calcium concentration is a lagging indicator of an action potential that persists for at least an order of magnitude longer than the electrical impulse. It then follows that as the firing rate of a neuron increases, the concomitant influxes of calcium begin to summate. For example, a neuron firing at 1Hz may appear to have discrete calcium events that fall back to “baseline” whereas a neuron firing at 10 Hz will have a raised intracellular calcium concentration with dynamic events laid on top.

The relationship between calcium concentration and firing rate is affected by which version of the sensor you are using. Sensors should be selected to cover, as linearly as possible, the range of relevant firing rates of the neurons you are recording from.

In thinking about our photometry signal, we aren't recording from a single cell – but rather a volume of tissue underneath our optical fiber. This signal is the sum of calcium-dependent changes in fluorescence across a plethora of neurons. A practical consideration: as these signals sum together, it becomes less and less likely that you will observe a “baseline” signal during which the mean firing rate is ~0. Furthermore, amplitude of individual events may be affected by variation in expression level as well as the location of a neuron relative to your fiber. However, this tends to average out.

**A photometry signal is a proxy for the mean firing rate of neurons in the volume underneath your fiber.**

## Experimental Goals

---

This might sound obvious – but the goal of the experiment dictates the analysis approach. It also affects the viability/necessity of different correction techniques.

For example, imagine your goal is to see if there is a response of a given brain region to a foot shock. This can be a trivial analysis: record from the animal, administer a foot shock, and then look at your signal before and after and compare.

This type of experiment won't be heavily affected by, for instance, how you correct for photobleaching or what you call "baseline." You will end up with a very clear event-related fluorescence trace that will satisfy your experimental goals.

Now imagine you wanted to see if a certain pharmacological treatment affected the response to a foot shock. Now, you need to carefully consider photobleaching and how you assign a baseline. Does the drug affect the activity of this region prior to administration of the shock?

Though this falls under the umbrella of "personal preference," we do have a few recommendations that can make the process easier.

The absolute value of your signal (in the FP3002's case, presented as a fraction of maximum mean pixel value, or F) is most affected by factors that are likely not relevant to your experiment. For example: small differences in expression level, fiber placement, or anatomy can have an outsized effect on the amplitude of your signal.  $dF/F$  is, theoretically, a way to correct for this variation. However, this analysis is hugely affected by what one calls "baseline" and there is not a consensus in the field on how to perform these corrections. **Our preference is to design experiments where the results are invariant to across-animal differences in the absolute value of signal.** To do this, it requires a paradigm shift in how one designs an experiment.

It is quite common to set up an experiment where an animal is presented with a standardized stimulus repeatedly. That is, there is next-to-no variance in the

independent variable. This is important for techniques with low signal to noise ratio (SNR), such as EEG or fMRI, where many trials need to be averaged together to detect a signal. Photometry, however, has a relatively high SNR with the majority of all experiments achieving single trial resolution. Under these conditions, we recommend a different approach.

Example: You have a cohort of identical animals, where the only source of variation is the fiber placement relative to your cells of interest. Some animals show a very robust SNR and others show a much lower SNR. However, all animals show a response to a standardized stimulus 100% of the time. If you average the results within-animal, then compare across-animal, you will observe a common effect – however, there will be quite a bit of variance in your combined results. If the variation is extreme enough, you are at risk of a false negative. Further, it may not be representing your data in an appropriate manner.

Remember, each animal showed a response to the stimulus 100% of the time. The likelihood that this occurred by chance should be infinitesimal.

Now, imagine we set up the experiment in a different way. Instead of presenting the same stimulus over and over, we now add variation to our independent variable. This could be as simple as varying the intensity or duration (for foot shocks or tone), taking into consideration the order of presentation (for events known to result in habituation), or diving into a behavioral analysis in more detail.

With this revised experimental design, for each animal, you will be able to assess the response AS A FUNCTION of the variation in the independent variable. **Of note, the shape of this dose response curve is, largely, invariant to differences in expression level/fiber placement (with some ceiling and floor effects).** It also allows you to make more nuanced inferences about the data.

## Data Format

### Data Format

---

There are a few ways to save your photometry data. We recommend saving the non-deinterleaved file, as this is the rawest form of the data available, and it makes analysis easier down the road. Data files are automatically saved as a .csv (comma separated value) file when a Bonsai workflow is stopped.

For every frame, we save a packet of information detailing the internal state of the system in addition to our photometry signals. The number of columns in your datafile will depend on the number of ROIs you record from, but every file will start with the same default columns.

| A            | B           | C        | D           | E       | F       | G      | H      | I              | J              |
|--------------|-------------|----------|-------------|---------|---------|--------|--------|----------------|----------------|
| FrameCounter | Timestamp   | LedState | Stimulation | Output0 | Output1 | Input0 | Input1 | Region0R       | Region1G       |
| 0            | 1479.081568 | 7        | 0           | 0       | 0       | 0      | 0      | 0.006221568627 | 0.002721568627 |
| 1            | 1479.09264  | 2        | 0           | 0       | 0       | 0      | 0      | 0.09988002015  | 0.06965665374  |
| 2            | 1479.103744 | 1        | 0           | 0       | 0       | 0      | 0      | 0.07649011083  | 0.08548503214  |
| 3            | 1479.11488  | 4        | 0           | 0       | 0       | 0      | 0      | 0.08071877858  | 0.09323343459  |
| 4            | 1479.125984 | 2        | 0           | 0       | 0       | 0      | 0      | 0.09939563667  | 0.06919211531  |
| 5            | 1479.137088 | 1        | 0           | 0       | 0       | 0      | 0      | 0.07446248159  | 0.08321968014  |
| 6            | 1479.148192 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08184545455  | 0.09580023423  |
| 7            | 1479.159296 | 2        | 0           | 0       | 0       | 0      | 0      | 0.09990908316  | 0.06944889953  |
| 8            | 1479.1704   | 1        | 0           | 0       | 0       | 0      | 0      | 0.0768398357   | 0.08540608554  |
| 9            | 1479.181536 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08195783151  | 0.09309238429  |
| 10           | 1479.19264  | 2        | 0           | 0       | 0       | 0      | 0      | 0.1001193056   | 0.06951205681  |
| 11           | 1479.203744 | 1        | 0           | 0       | 0       | 0      | 0      | 0.0770684647   | 0.08540193045  |
| 12           | 1479.214848 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08234921336  | 0.09124822309  |
| 13           | 1479.225952 | 2        | 0           | 0       | 0       | 0      | 0      | 0.100545563    | 0.06948130918  |
| 14           | 1479.237056 | 1        | 0           | 0       | 0       | 0      | 0      | 0.07737459506  | 0.08530636351  |
| 15           | 1479.248192 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08232208789  | 0.09323343459  |

**FrameCounter:** Sequentially counts the number of frames collected during an experiment.



**Timestamp:** Time when a frame was recorded using the internal clock in the FP3002 (<5  $\mu$ s precision). This column is formatted as seconds since the system was booted up.

**LED state:** The frame flag indicates which LED was on during each frame. For more info, see Frame Flags Section.

**Stimulation/Digital IOs:** (Column 4-8) Binary, 0 if off, 1 is on. On means a “high” voltage value was at the DIO port when the FP3002 frame is sent to bonsai (about 500 usec after the exposure ends). It is very easy to miss the logging of “high” values in this file. We recommend using the DigitalIO nodes in bonsai to accurately track that data, instead.

**RegionXR:** Mean pixel value within the confines of the user-selected region of interest (ROI) on the half of the image sensor that captures red emission. Note: These data are normalized as the fraction of the maximum value, so the range is 0-1. X is an integer starting at 0.

**RegionYG:** Mean pixel value within the confines of the user-selected region of interest (ROI) on the half of the image sensor that captures green emission. Note: These data are normalized as the fraction of the maximum value, so the range is 0-1. Y is an integer starting at 0.

The FP3002 system records a data point for each ROI for each frame, regardless of which LED is on. This means that if you are recording with all 3 LEDs, you will end up with 415, 470, and 560 readouts in both red and green ROIs, and these values will be interleaved. This means that in order to plot or process your data, you first need to deinterleave the data.

The frame flag tells you which row corresponds to which LED. The first frame will always be an initialization frame and have a frame flag of 7. Below we have color coded an example file for clarity.

| Frame | Timestamp   | LedState | Stimulation | Output0 | Output1 | Input0 | Input1 | Region0R       | Region1G       |  |                |
|-------|-------------|----------|-------------|---------|---------|--------|--------|----------------|----------------|--|----------------|
| 0     | 1479.081568 | 7        | 0           | 0       | 0       | 0      | 0      | 0.006221568627 | 0.002721568627 |  |                |
| 1     | 1479.09264  | 2        | 0           | 0       | 0       | 0      | 0      | 0.09988002015  | 0.06965665374  |  | 470            |
| 2     | 1479.103744 | 1        | 0           | 0       | 0       | 0      | 0      | 0.07649011083  | 0.08548503214  |  | 415            |
| 3     | 1479.11488  | 4        | 0           | 0       | 0       | 0      | 0      | 0.08071877858  | 0.09323343459  |  | 560            |
| 4     | 1479.125984 | 2        | 0           | 0       | 0       | 0      | 0      | 0.09939563667  | 0.06919211531  |  |                |
| 5     | 1479.137088 | 1        | 0           | 0       | 0       | 0      | 0      | 0.07446248159  | 0.08321968014  |  | red emission   |
| 6     | 1479.148192 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08184545455  | 0.09580023423  |  | green emission |
| 7     | 1479.159296 | 2        | 0           | 0       | 0       | 0      | 0      | 0.09990908316  | 0.06944889953  |  |                |
| 8     | 1479.1704   | 1        | 0           | 0       | 0       | 0      | 0      | 0.0768398357   | 0.08540608554  |  |                |
| 9     | 1479.181536 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08195783151  | 0.09309238429  |  |                |
| 10    | 1479.19264  | 2        | 0           | 0       | 0       | 0      | 0      | 0.1001193056   | 0.06951205681  |  |                |
| 11    | 1479.203744 | 1        | 0           | 0       | 0       | 0      | 0      | 0.0770684647   | 0.08540193045  |  |                |
| 12    | 1479.214848 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08234921336  | 0.09124822309  |  |                |
| 13    | 1479.225952 | 2        | 0           | 0       | 0       | 0      | 0      | 0.100545563    | 0.06948130918  |  |                |
| 14    | 1479.237056 | 1        | 0           | 0       | 0       | 0      | 0      | 0.07737459506  | 0.08530636351  |  |                |
| 15    | 1479.248192 | 4        | 0           | 0       | 0       | 0      | 0      | 0.08232208789  | 0.09323343459  |  |                |

Importantly, not all of these data points are meaningful. For example, we don't care what the emission in the red channel is when we excite a green sensor with the 415 or 470 LED, and we don't care about the green emission when we excite a red sensor with the 560 LED. These excitation/emission combinations can be ignored during analysis. Below are the relevant emission channels for each excitation wavelength.

| Excitation LED | Green Emission | Red Emission |
|----------------|----------------|--------------|
| 415            | ✓              |              |
| 470            | ✓              |              |
| 560            |                | ✓            |

If you are only using a single indicator, it is not necessary to use all excitation wavelengths, or record both emission channels. For example, if you are just recording GCaMP, we recommend just using the 470 and 415 LEDs and including green ROIs in your Bonsai file.

## Understanding Frame Flags

Frame Flags are a way to understand what events occurred during a recording. Frame Flags can be found in columns three through eight of your photometry csv file. Column three represents the LED state. Using frame flags can aid *post hoc* data processing by noting if a component was simultaneously on while data was recorded. For example, you can more efficiently compare fiber photometry data when the laser was ON versus OFF.

|                             | Column Position | State is LOW/OFF | L415 | L470 | L560 |
|-----------------------------|-----------------|------------------|------|------|------|
| <b>LED State</b>            | 3               | 0                | 1    | 2    | 4    |
| <b>Output 1 signal HIGH</b> | 4               | 0                | 1    | 1    | 1    |
| <b>Output 0 signal HIGH</b> | 5               | 0                | 1    | 1    | 1    |
| <b>Stimulation ON</b>       | 6               | 0                | 1    | 1    | 1    |
| <b>Input 1 HIGH</b>         | 7               | 0                | 1    | 1    | 1    |
| <b>Input 0 HIGH</b>         | 8               | 0                | 1    | 1    | 1    |

**Table 1:** Frame flags representing when a component is active. If the value is 0, the component was not active during a frame. If the value is 1, the component was active during a frame. Note that column 3: LED State, displays different values based upon what LED was active.

Columns four through eight are binary. If the component was active during a given frame, the value will be 1. If the component was not active during a given frame, the value will be 0. Column three: LED State specifies what LED was active during a given frame. The value will be 1 if the 415 nm LED was active. The value will be 2 if the 470 nm LED was active. The value will be 4 if the 560 nm LED was active. **These values are independent of the order of LEDs in your trigger sequence (i.e 2 will always be 470).** A value of 0 represents a “None” frame wherein no LED is active and is typically used in extended (>8 hr) recordings. A value of 7 represents the initiation of the recording.

## Introduction to Analyzing Data in MATLAB

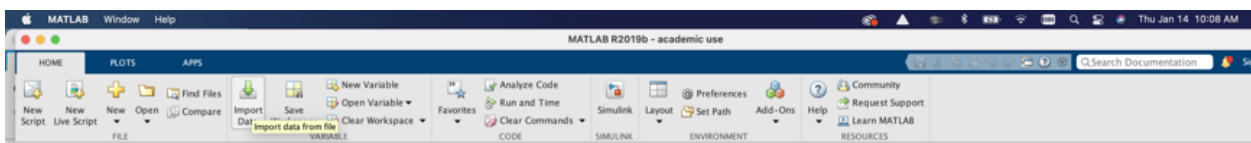
Any programming language can be used to analyze Fiber Photometry data, however, we do not recommend trying to analyze data in Excel. Below is example code and information about how to analyze your data in MATLAB. You will need MATLAB 2019b or later, as well as the following MATLAB packages to run all functions in the code below (generally free with academic license): Curve fitting Toolbox, Signal Processing Toolbox, Image Processing Toolbox, and Statistics and Machine Learning Toolbox.

All processing steps below are guidelines, we encourage you to tailor scripts to fit the needs of your experiments and signals.

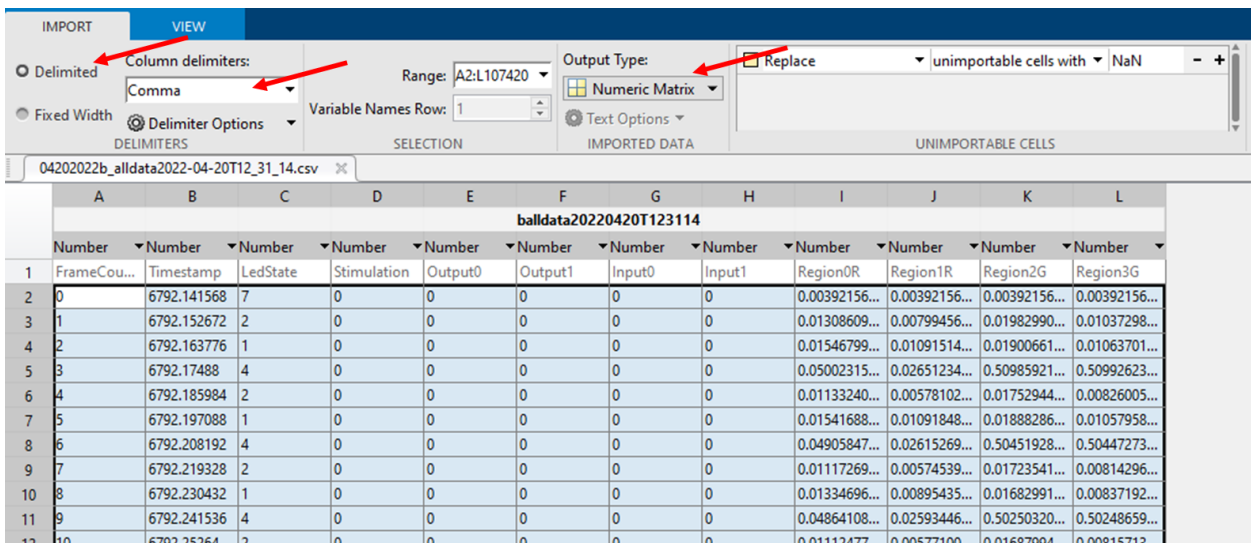
## Importing Data into MATLAB

There are many ways to input data into MATLAB – including methods for batch processing. Here, we will go over the simplest method first.

To import your data into MATLAB, click on “Import Data” in the toolbar.



The FP data are stored as a .csv file (comma separated value). Select “Delimited” and “Comma” at the top left corner of the import window to ensure the data are imported correctly. Lastly, under Output Type, select “Numeric Matrix.”



|    | A                              | B           | C        | D           | E       | F       | G      | H      | I             | J             | K             | L             |
|----|--------------------------------|-------------|----------|-------------|---------|---------|--------|--------|---------------|---------------|---------------|---------------|
|    | <b>balldata20220420T123114</b> |             |          |             |         |         |        |        |               |               |               |               |
|    | Number                         | Number      | Number   | Number      | Number  | Number  | Number | Number | Number        | Number        | Number        | Number        |
| 1  | FrameCou...                    | Timestamp   | LedState | Stimulation | Output0 | Output1 | Input0 | Input1 | Region0R      | Region1R      | Region2G      | Region3G      |
| 2  | 0                              | 6792.141568 | 7        | 0           | 0       | 0       | 0      | 0      | 0.00392156... | 0.00392156... | 0.00392156... | 0.00392156... |
| 3  | 1                              | 6792.152672 | 2        | 0           | 0       | 0       | 0      | 0      | 0.01308609... | 0.00799456... | 0.01982990... | 0.01037298... |
| 4  | 2                              | 6792.163776 | 1        | 0           | 0       | 0       | 0      | 0      | 0.01546799... | 0.01091514... | 0.01900661... | 0.01063701... |
| 5  | 3                              | 6792.17488  | 4        | 0           | 0       | 0       | 0      | 0      | 0.05002315... | 0.02651234... | 0.50985921... | 0.50992623... |
| 6  | 4                              | 6792.185984 | 2        | 0           | 0       | 0       | 0      | 0      | 0.01133240... | 0.00578102... | 0.01752944... | 0.00826005... |
| 7  | 5                              | 6792.197088 | 1        | 0           | 0       | 0       | 0      | 0      | 0.01541688... | 0.01091848... | 0.01888286... | 0.01057958... |
| 8  | 6                              | 6792.208192 | 4        | 0           | 0       | 0       | 0      | 0      | 0.04905847... | 0.02615269... | 0.50451928... | 0.50447273... |
| 9  | 7                              | 6792.219328 | 2        | 0           | 0       | 0       | 0      | 0      | 0.01117269... | 0.00574539... | 0.01723541... | 0.00814296... |
| 10 | 8                              | 6792.230432 | 1        | 0           | 0       | 0       | 0      | 0      | 0.01334696... | 0.00895435... | 0.01682991... | 0.00837192... |
| 11 | 9                              | 6792.241536 | 4        | 0           | 0       | 0       | 0      | 0      | 0.04864108... | 0.02593446... | 0.50250320... | 0.50248659... |
| 12 | 10                             | 6792.25264  | 2        | 0           | 0       | 0       | 0      | 0      | 0.01112477    | 0.00577100    | 0.01687994    | 0.00815713    |

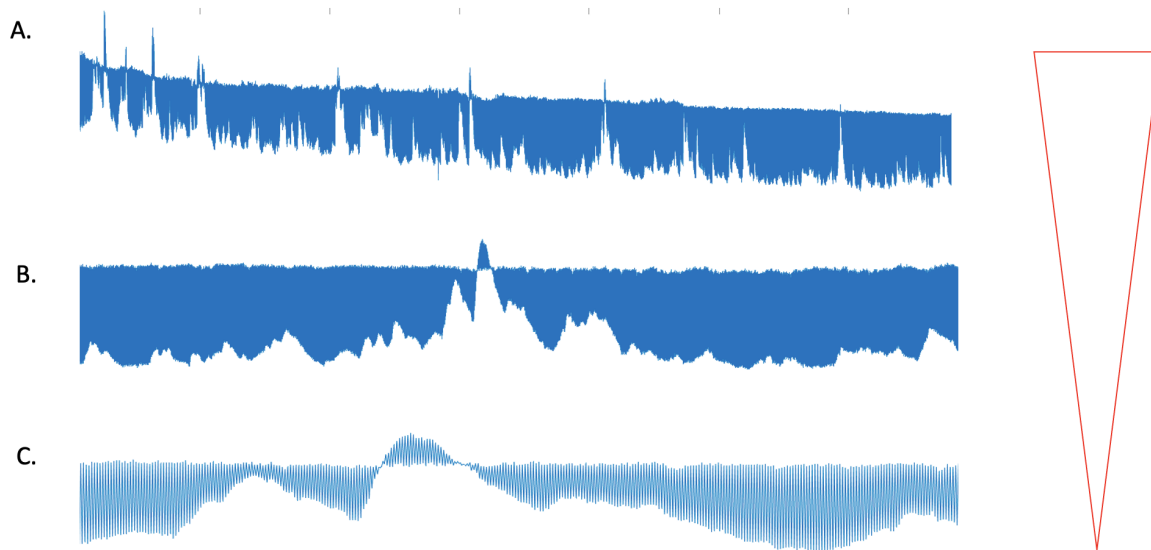
## Visualizing Data

---

It is important to familiarize yourself with what “raw” photometry data look like. To do this, plot the data from one of your ROIs. No need to worry about an X value.

figure

```
plot(data(:,9)) %plots all data points from column 9 with number of datapoints on x axis
```



This is an example of a dataset with a high signal-to-noise ratio (SNR) in a brain region with quite a bit of spontaneous (and evoked) activity. Subplots A, B, and C are zoomed in versions of the same data.

Looking at A, we see there are two parts of the data -- the lower part where we see a lot of activity and the upper part that is relatively flat. Zooming in further, with B, it becomes even clearer and we can observe the telltale asymmetrical kinetics of GCaMP (in this case, 6s). Faster rise, slower decay. Zooming in even further, with C, we can see that every-other-datapoint is either from the group with signal (lower group, in this case) or without (higher group, in this case).

With this dataset, researchers were oscillating between 415nm stimulation and 470nm stimulation resulting in a calcium-independent (415nm, upper group) and calcium-dependent (470nm, lower group) data streams.

Next, we will deinterleave these data using the “LED State” column (column 3). For example:

figure

```
plot(data(data(:,3)==2,9)) %plots all data from column 9 where the flag value (in column 3) was 2 indicating that the 470nm LED was on; data plotted in A below
```

```
plot(data(data(:,3)==1,9)) %plots all data from column 9 where the flag value (in column 3) was 1 indicating that the 415nm LED was on; data plotted in B below
```

LED States

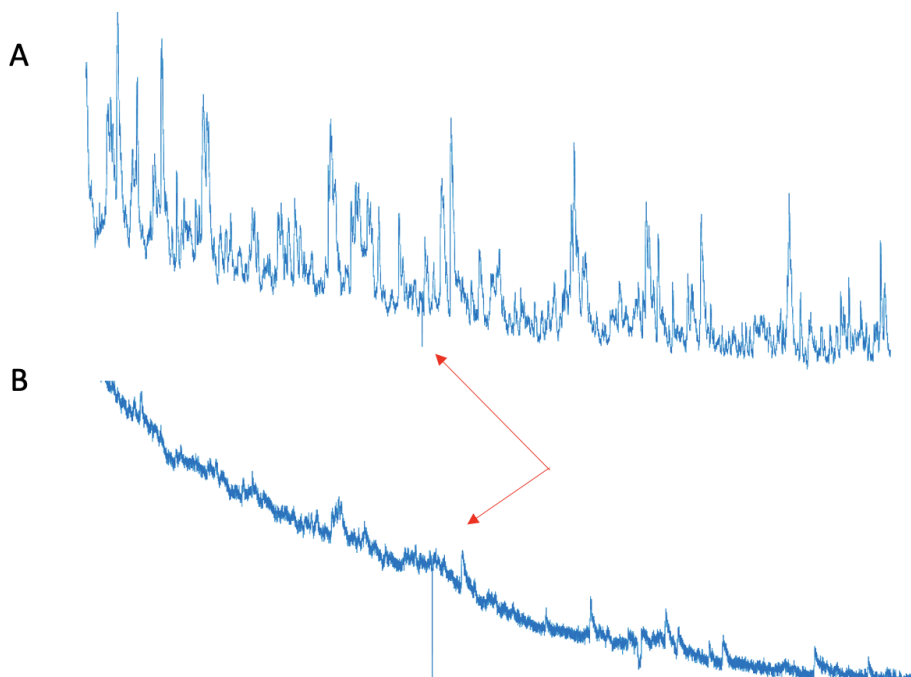
0: None Frame (no LED)

1: 415 LED

2: 470 LED

4: 560 LED

7: Initialization frame



Note: Red arrows indicate a period where the patch cord was gently bent past the point it can maintain total internal reflection. Note the artifact in both channels.

Just by looking through the deinterleaved data, we can glean a tremendous amount of useful information. Notice that the traces have both a “fast” and “slow” component, where the slow component refers to the overall sloped shape of the trace, and the fast component refers to the spikes and artifacts on top of it.

First, let’s consider the fast component. In this example, the 470 signal looks “busier” than the 415, which is an indicator of a good signal. Events in the 470-signal have a fast rise and slower decay, which is typical for GCaMP. There is some overlap between the two channels, but the events in the 470 channel are much larger. Overall, we can see at this point that this is a quality signal.

Now let’s consider the slow component of the signal. It is very typical with fiber photometry recordings to have an overall decay in signal due to photobleaching of the patch cord, fluorescent indicator, and autofluorescence of the tissue. This shape does not provide a tremendous amount of biologically-relevant information and can be corrected to flatten traces.

*Note: You can have bonsai save a chart of your deinterleaved data. This allows you to quickly scan through files (without opening MATLAB or any analysis software) to see what your signal looks like.*

**ALWAYS SCAN THROUGH DE-INTERLEAVED DATA FROM ANY PHOTOMETRY RECORDING.**

Note that it is not guaranteed that you will have the same number of datapoints for each LED.

`FP.raw = imported matrix; %change imported matrix to be whatever name your data file imported as`



```
FP.n_channels = length(unique(FP.raw(2:end,3))); %extracts number of channels (ie  
415, 470, 560 ignoring the initialization frame);
```

```
%ensure equal number of frames for each channel
```

```
while mod(length(FP.raw),FP.n_channels) ~= 0  
    FP.raw(end,:) = [];  
end
```

```
FP.data(:,1) = FP.raw(FP.raw(:,3)==2,2); %timestamp when 470 LED is on  
FP.data(:,2) = FP.raw(FP.raw(:,3)==2,9); %F value from first ROI when 470 LED is on  
FP.data(:,3) = FP.raw(FP.raw(:,3)==1,9); %F value from first ROI when 415 LED is on
```

## Strategies for Further Processing

---

There are various strategies one can take once the data is scanned through, depending on the purpose of the experiment and the sensor. We outline the most common ones below.

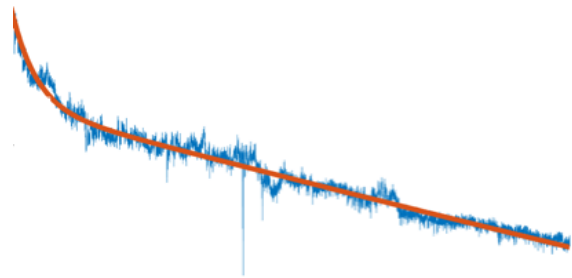
1. Correct data for photobleaching
2. Align data to some event
3. Segment data around said event
4. Find peaks

## Correcting for Photobleaching

---

After deinterleaving data, you may want to remove the slow component of your signal to correct for photobleaching. There are a couple of approaches out there. This is the one we recommend for 90% of datasets.

- 1) Fit isosbestic with biexponential decay



figure

```
 tiledlayout(4,1)
```

```
 ax1 = nexttile;
```

```
 plot(FP.data(:,3)); %plot 415 data
```

```
 hold on
```

```
 temp_x = FP.raw(FP.raw(:,3)==2,1); %use frame number as temporary x variable for fitting to avoid sig figs issue with MATLAB
```

```
 FP.fit1 = fit(temp_x,FP.data(:,3),'exp2'); %fit raw isosbestic with biexponential
```

```
 plot(FP.data(:,3)) %plot raw 415
```

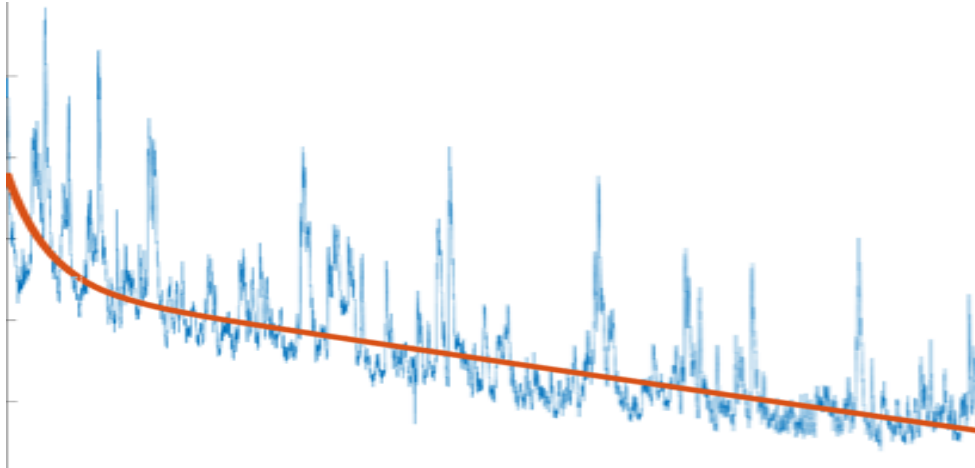
```
 plot(FP.fit1(temp_x)); %overlay fit onto plot
```

- 2) Linearly scale that fit to the calcium dependent data (using Robust Fit)

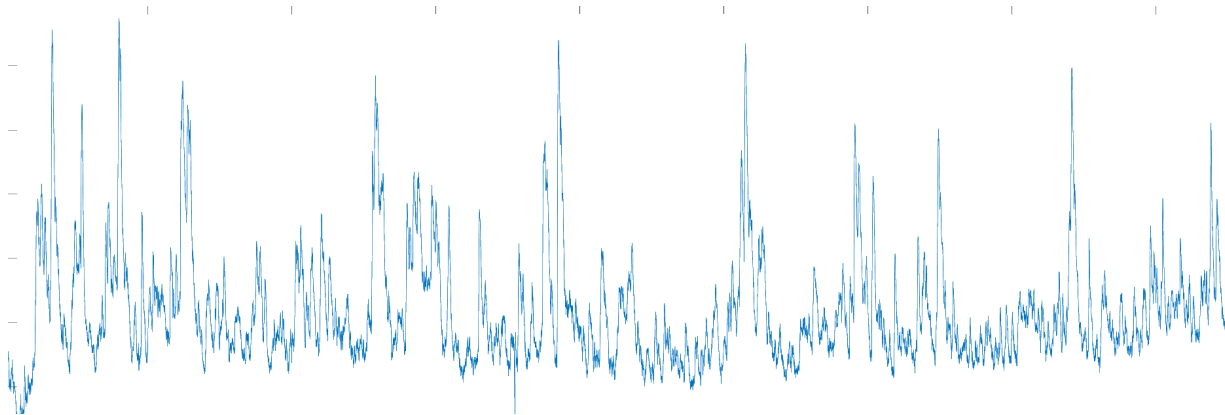
```
 FP.fit2 = robustfit(FP.fit1(temp_x),FP.data(:,2), 'bisquare'); %scale exp2 fit using robust fit
```

```
 FP.lin_fit = FP.fit1(temp_x)*FP.fit2(2)+FP.fit2(1); %save the output
```

```
ax2=nexttile  
plot(FP.data(:,2)); %plot raw 470  
hold on  
plot(FP.lin_fit); %plot scaled fit
```



3) Divide your raw 470 data by the scaled fit (dF)



```
FP.data(:,4) = FP.data(:,2) ./ FP.lin_fit; %save normalized F value
```

```
ax3=nexttile;  
plot(FP.data(:,4)); %plot normalized F value
```

4) Calculate  $dF/F$  (%)

`FP.data(:,5) = 100*(FP.data(:,2)-(FP.lin_fit))./FP.lin_fit;` %this is ONE way to get a "dF/F" signal. it isn't perfect without a solid baseline measurement, but it has been published. Note that it only rescales the "normalized F" value.

`ax4 = nexttile;`  
`plot(FP.data(:,5))`

For an additional resource, please see [this paper](#).

## Create Peri-Event Plots

---

There are a number of ways to synchronize an external signal to your photometry data. For the purposes of this example, we'll assume that you generated a .csv file that contains a timestamp (using the FP system as the timer) for each event.

First, decide on how many data points, before and after, the event you are interested in looking at.

```
beh= 'your behavior timestamp file' %fill in the name of this variable with your filename
```

```
FP.bl = 100; %number of datapoints before immobility bout
```

```
FP.seg_dur = 200; %number of datapoints after start of immobility bout
```

```
clear temp %clear temporary variable
```

```
counter = 1; %initialize counter
```

```
for i = 1:length(beh)
```

```
    temp = find(FP.data(:,1)>=beh(i)); %finds photometry datapoint nearest to start of  
    immobility bout
```

```
    if temp >= FP.bl && temp+FP.seg_dur <= length(FP.data) %ignores immobility bouts  
        that are too close to the beginning or end of the recording
```

```
        FP.ERF(:,counter) = FP.data(temp-FP.bl:temp+FP.seg_dur,4); %takes a segment of  
        data including the specified number of datapoints before and after the timestamp
```

```
        FP.ERF(:,counter) = FP.ERF(:,counter) - mean(FP.ERF(1:FP.bl,counter)); %baseline  
        correction – set baseline period to 0
```

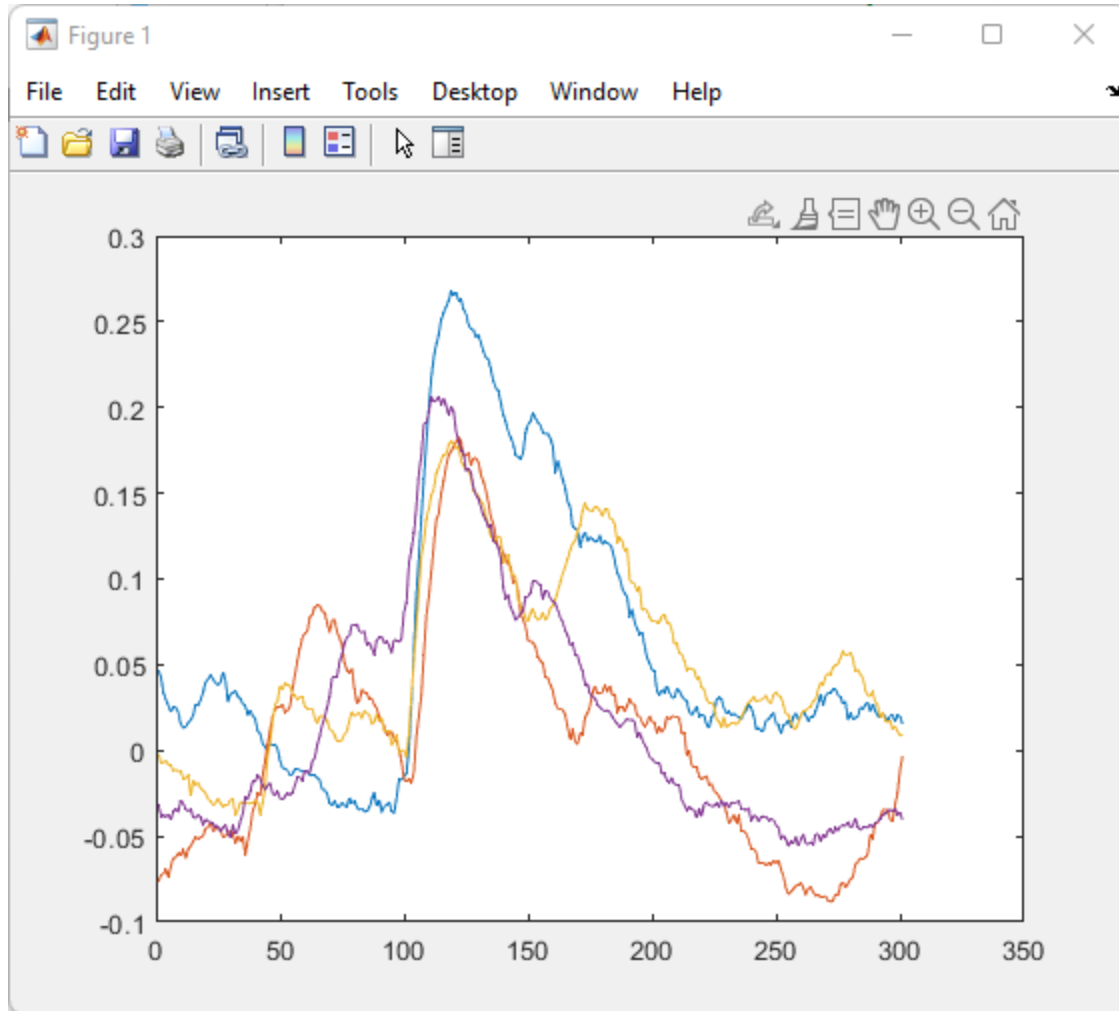
```
        counter = counter + 1; %increment counter
```

```
    else
```

```
    end
```

```
end
```

`plot(FP.ERF)`

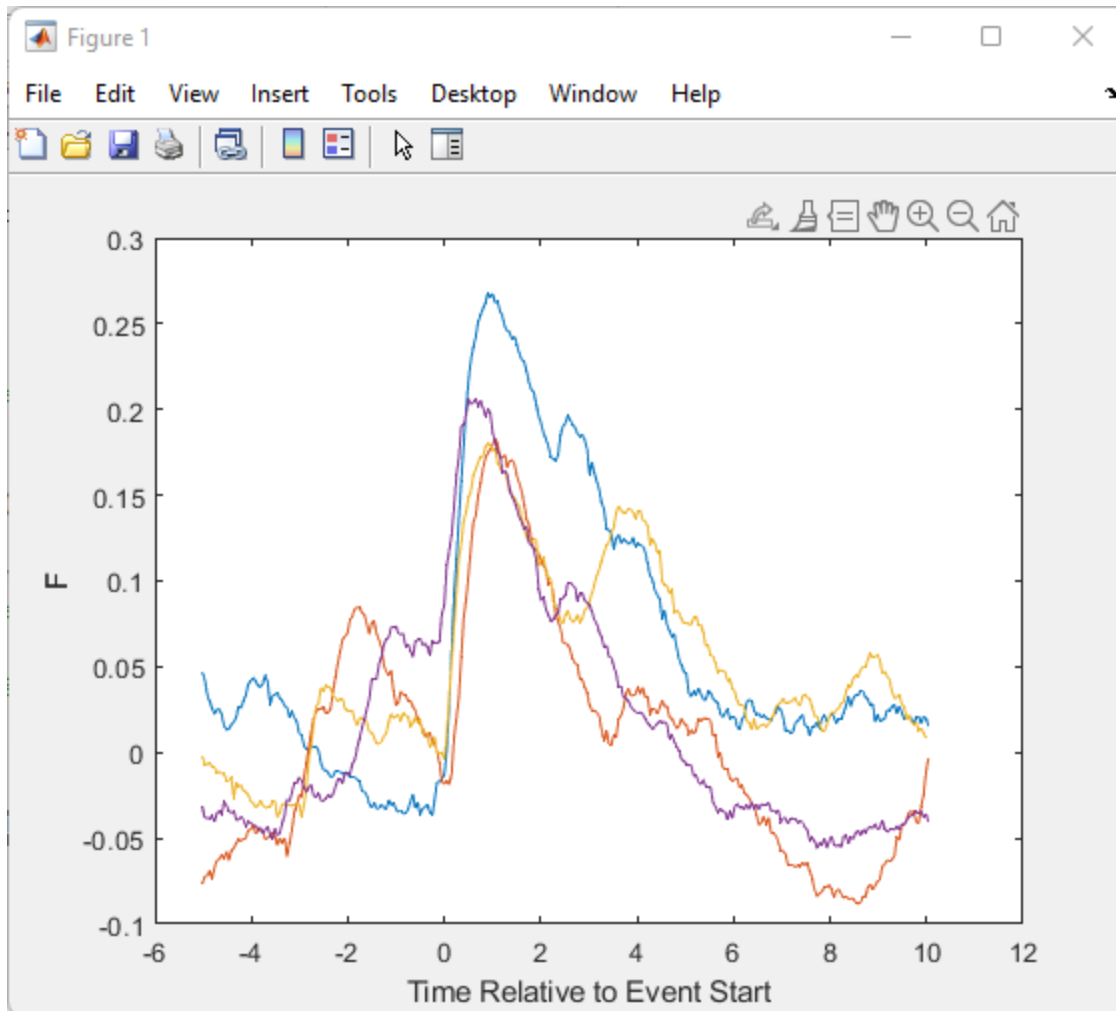


Note that the X value here is not particularly useful, it is the number of datapoints in the trace. Let's extract the effective frame rate (that is, the frame rate after de-interleaving) from the data.

```
FP.fps = 1/(mean(diff(FP.data(:,1)))); %take the average of the first derivative of the time variable to get the mean period between frames. 1/period = frequency.
```

```
FP.ERF_time = linspace(-FP.bl/FP.fps,FP.seg_dur/FP.fps,length(FP.ERF(1).data));
```

```
plot(FP.ERF_time,FP.ERF);  
xlabel('Time Relative to Event Start')  
ylabel('F')
```



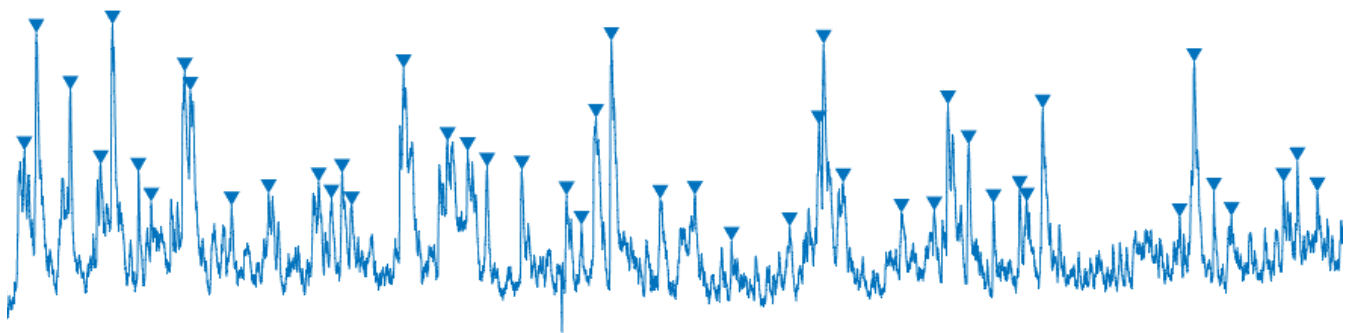


## Spike Detection

---

Another common metric you may want to look at are spikes. Matlab has built in peak finding capabilities that make it very simple to identify spikes in your data

```
%detect peaks with a peak prominence above .06  
[~, FP.peaks] = findpeaks(FP.data(:,5),'MinPeakProminence',0.06);  
  
%plot trace with labeled peaks  
figure  
findpeaks(FP.corrected_data(:,3),'MinPeakProminence',0.06)
```



You can also estimate instantaneous spike frequency to see how firing rates change over time or in response to a stimulus.

```
FP.peak_freq = zeros(size(FP.data(:,5))); %initialize an array of zeros equal to the length  
of your deinterleaved data
```

```
FP.peak_freq(FP.peaks) = 1; %change time point where peaks were localized to "1",  
this makes it easier to calculate frequency
```

figure

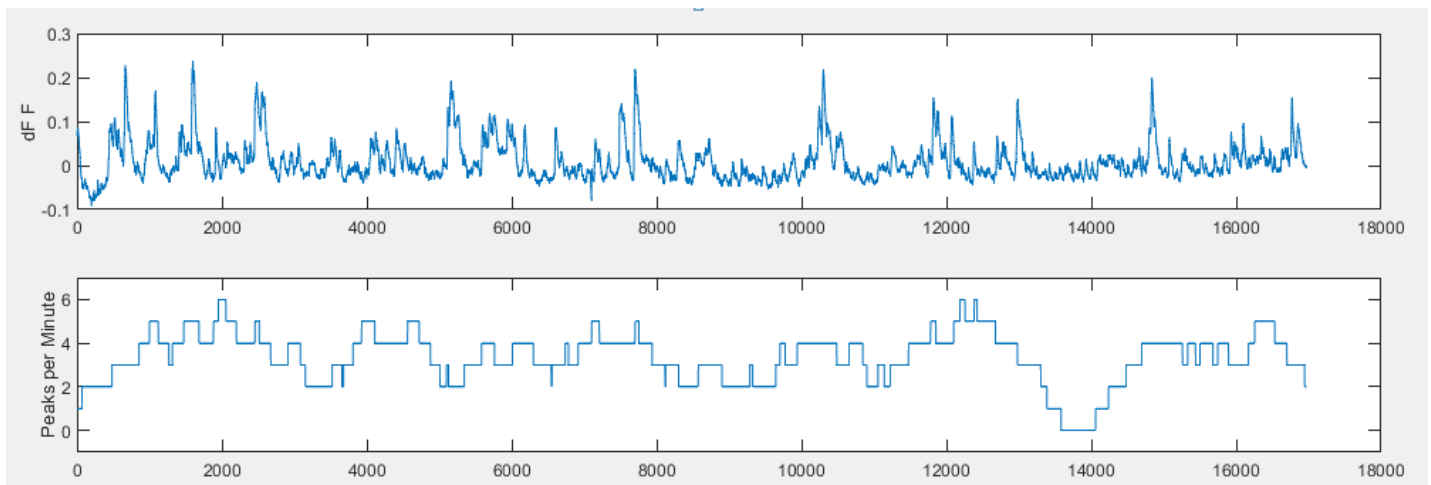
```
 tiledlayout(4,1)
```

```
 ax1 = nexttile;
```

```
 plot(FP.corrected_data(:,3)) %plot normalized data
```

```
 ax2 = nexttile;
```

```
 plot((movsum(FP.peak_freq,round(FP.fps*60)))) %plot estimate of instantaneous  
 frequency -- note: you can change the sliding window size here. if you do so, you will  
 need to adjust the output. for example, if you double the window size, you will need to  
 divide the output by 2 to normalize to "peaks per minute"
```



## Note on dF/F

---

As noted earlier, the absolute value of your signal is heavily affected by variables that are often not important to your biological question. These can range from varying levels of expression to fiber placement to local anatomy. Additionally, the concentration of the available indicator is not static due to photobleaching. In theory, dF/F is supposed to normalize measurements both within (i.e. photobleaching) and across animals. In practice, however, the story is a bit more complex and there is a lot of variability in how different labs approach this topic.

$$(F_{\text{observed}} - F_{\text{baseline}}) / F_{\text{baseline}}$$

There are only two terms in this calculation.  $F_{\text{observed}}$  is simply the “F” value that the photometry system spits out. In our case, technically, it is the normalized mean pixel value – but the units aren’t important here. What one calls “baseline” can hugely skew this normalization technique – especially given that the baseline is dynamic.

In an ideal world, the baseline would be the F value observed by the system when your biological event is not occurring. If you are using GCaMP, this would be the F value you pick up when the mean firing rate is equal to zero. Unfortunately, this can be a difficult measurement to make.

Tip: Record from your region of interest, then put the animal under anesthesia. MOST brain regions decrease in activity under anesthesia and this will give you a biologically relevant, within-animal baseline measurement.

In our example code, we make the following assumptions:

- 1) The slow decay of the 415-related signal is mostly explained by a decrease in the available concentration of our indicator.
- 2) The 470-signal is of sufficient duration such that this slow decay is represented in the data.

In our experience, this approach is appropriate for >90% of all recordings.