

Synaptech Suite User Guide

Winter 2024

Versioned Jan 31, 2024



Table of Contents

Preface	3
Pivot Off the Biology	4
Experimental Goals	6
Note on dF/F	7
Data Format	9
Navigating the Workspace	12
Analyzing Data	13
Step 1: Process Traces	15
Step 2: Timestamp Alignment	20
Step 3: Data Pooling	24

Preface

The goal of this manual is to provide you with the necessary tools to analyze and interpret fiber photometry data using Synaptech Suite. Synaptech Suite is a cloud-based comprehensive fiber photometry analysis pipeline that allows you to process raw signals, create interactive plots, align photometry data to timestamps, and store and share data—all without writing a single line of code.

Synaptech Suite is python-based and runs off a remote kernel, meaning that no software needs to be installed, and no minimum computer specifications are required. Data storage and processing can be accessed via web browser.

The Synaptech Suite pipeline has been optimized over hundreds of recordings from labs across the world. It consists of 3 primary processing scripts that are designed to be run consecutively;

1. Signal processing
2. Aligning timestamps to fiber photometry data
3. Parsing and plotting data around timestamps of interest

Details for each section are discussed below.

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 the activity of a cell population of interest. In fiber photometry, calcium concentration can be measured as a proxy for cell firing.

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-gated 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, resulting in a transient spike in intracellular calcium concentration that correlates with AP firing, albeit slightly delayed.

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 like GCaMP 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 mentioned above, 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.

Additionally, with fiber photometry, 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 group 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. The processing steps in the Synaptech Suite signal processing pipeline take this into account.

In sum, a fluorescent GCaMP 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.

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_{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.

Data Format

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
FrameCounter	SystemTimestamp	LedState	ComputerTimestamp	G0	G1
0	30019.40276	7	29921663.98	0.09706	0.178338
1	30034.47988	2	29921682.83	0.122309	0.229784
2	30034.50451	1	29921687.87	0.094739	0.178294
3	30034.52998	2	29921692.81	0.120843	0.229177
4	30034.55556	1	29921697.8	0.093952	0.178282
5	30034.57981	2	29921702.79	0.119794	0.228347
6	30034.6054	1	29921707.78	0.093888	0.179127
7	30034.62993	2	29921713.05	0.119899	0.228589
8	30034.65544	1	29921713.73	0.093842	0.179018
9	30034.68012	2	29921713.73	0.120262	0.228647
10	30034.70531	1	29921713.73	0.093778	0.179125
11	30034.73089	2	29921732.77	0.120085	0.226737
12	30034.75535	1	29921737.76	0.094007	0.178462
13	30034.78092	2	29921742.74	0.119821	0.226568
14	30034.80556	1	29921747.73	0.094274	0.178459
15	30034.83144	2	29921752.72	0.119846	0.227103

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

System Timestamp: Time when a frame was recorded using the internal clock in the FP3002 (<5 μ 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.

Computer Timestamp: Time when a frame was recorded using the clock of the computer running Bonsai. This column is formatted as time of day in total milliseconds.

G0, G1, etc: Mean pixel value within the user-drawn regions of interest (ROI), i.e. photometry data. Each ROI is numbered according to how they are drawn in the FP3002 node, and are labeled G or R depending on if they capture red or green 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. The number of data columns in each file will depend on the number of ROIs drawn.

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.

A	B	C	I	J	K	L	M	N
FrameCoun	Timestamp	LedState	ComputerTimes	R0	G1			
0	1479.081568	7	29921663.98	0.006221568627	0.002721568627			
1	1479.09264	2	29921682.83	0.09988002015	0.06965665374		470	
2	1479.103744	1	29921687.87	0.07649011083	0.08548503214		415	
3	1479.11488	4	29921692.81	0.08071877858	0.09323343459		560	
4	1479.125984	2	29921697.8	0.09939563667	0.06919211531			
5	1479.137088	1	29921702.79	0.07446248159	0.08321968014		red emission	
6	1479.148192	4	29921707.78	0.08184545455	0.09580023423		green emission	
7	1479.159296	2	29921713.05	0.09990908316	0.06944889953			
8	1479.1704	1	29921713.73	0.0768398357	0.08540608554			
9	1479.181536	4	29921713.73	0.08195783151	0.09309238429			
10	1479.19264	2	29921713.73	0.1001193056	0.06951205681			
11	1479.203744	1	29921732.77	0.0770684647	0.08540193045			
12	1479.214848	4	29921737.76	0.08234921336	0.09124822309			
13	1479.225952	2	29921742.74	0.100545563	0.06948130918			
14	1479.237056	1	29921747.73	0.07737459506	0.08530636351			
15	1479.248192	4	29921752.72	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.

Navigating the Workspace

Launching your cloud workspace from a browser will open the landing page for your lab with the following file directory. Default folders will appear, along with the three processing scripts (.ipynb files). Below are descriptions of each folder.

Archived: previous iterations of notebooks

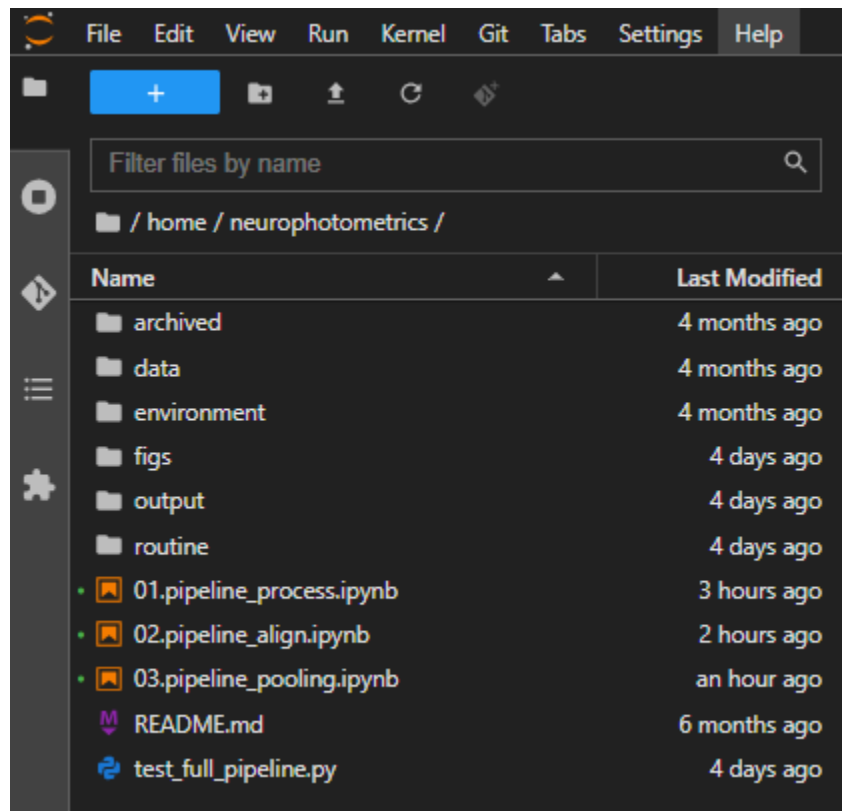
Data: where raw data files can be uploaded and stored

Environment: contains the default files necessary to run the notebooks

Figs: default folder where generated figures will save

Output: default folder where processed data will save

Routine: contains the back end code for functions referenced throughout the 3 notebooks.



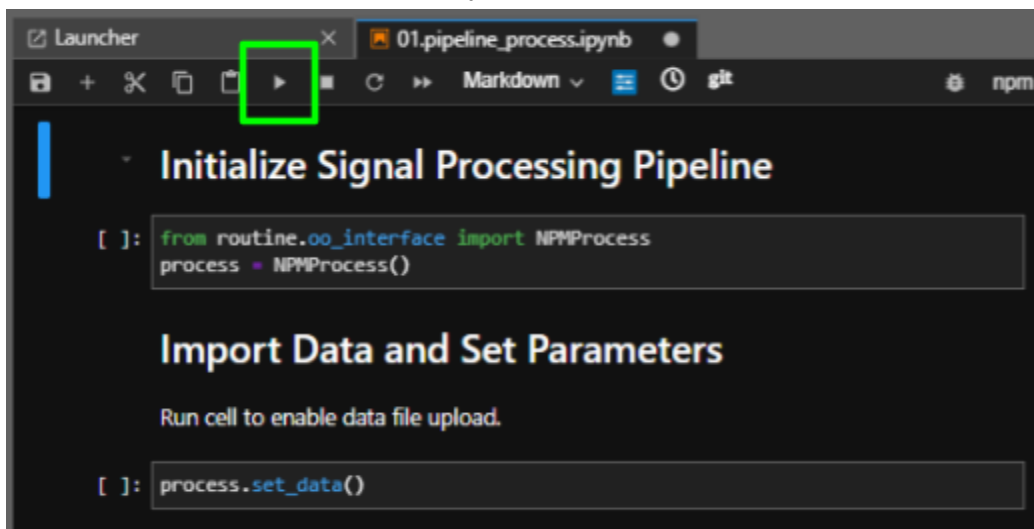
Analyzing Data

Currently, Synaptech Suite only supports data recorded with FP3002 systems and the latest version of Neurophotometrics Bonsai software (0.6.0 and later). To update to the latest NPM packages in Bonsai, please see instructions [here](#).

The analysis pipeline consists of 3 separate notebooks designed to be run in order

1. **01.pipeline_process**: this script imports raw data, crops a user-specified number of data points, de-interleaves and plots raw data, curve fits and normalizes data, plots normalized data, and writes out processed data in a universal format.
2. **02.pipeline_align**: this step imports timestamp data generated by keydown nodes, Arduinos, and behavior cameras and aligns timestamps to photometry data
3. **03.pipeline_pooling**: the final step of the pipeline segments data around timestamps of interest, creates peri-event plots, and exports segmented data

To open each notebook, double click on the file name in the directory to open a new tab. Jupyter notebooks consist of two types of “cells”, markdown and code. Markdown cells contain annotation, and code cells contain executable lines of code and are outlined in gray. To run a code cell, select it, then click the play button on the toolbar or hit shift+enter.



The screenshot shows a Jupyter Notebook interface with the following content:

- Toolbar:** Includes a play button (run cell) highlighted with a green box, and other icons for save, copy, paste, and refresh.
- Section 1:** **Initialize Signal Processing Pipeline** (Markdown cell).
 - Code Cell 1:**

```
[ ]: from routine.oo_interface import NPMProcess  
process = NPMProcess()
```
- Section 2:** **Import Data and Set Parameters** (Markdown cell).
 - Text:** Run cell to enable data file upload.
 - Code Cell 2:**

```
[ ]: process.set_data()
```

Vertical labels on the right side of the notebook indicate the cell types: Markdown, Code, Markdown, and Code.

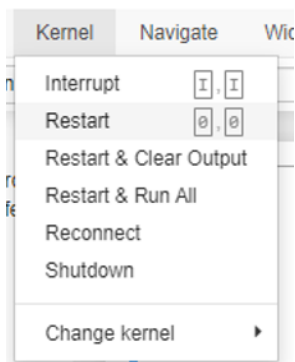
Cells that have not been run will have empty brackets next to them. Once a cell is run, a number will appear in these brackets, indicating that the functions have been executed.

```
Initialize Signal Processing Pipeline
```

```
[1]: from routine.oo_interface import NPMPProcess
      process = NPMPProcess()
```

```
Process initialized
```

To restart the kernel and clear outputs between files, click the Kernel button on the toolbar and select restart.



Step 1: Process Traces

Below are details on how to execute each step, and the scientific rationale behind them. The first few steps of this script involve entering analysis parameters. Next, raw signals will be de-interleaved, cropped, and normalized to yield a universally-formated processed data file for alignment with timestamps in later steps.

1. Initialize Signal Processing Pipeline

Run cell to initialize the pipeline, once finished "Process initialized" will appear.

```
Initialize Signal Processing Pipeline  
[1]: from routine.oo_interface import NPMPProcess  
      process = NPMPProcess()  
  
Process initialized
```

2. Import Data and Set Parameters

Run cell, then use the 'Select' button to choose the FP data file (.csv) to analyze, then hit select again.

3. Specify path to write out processed data and figures

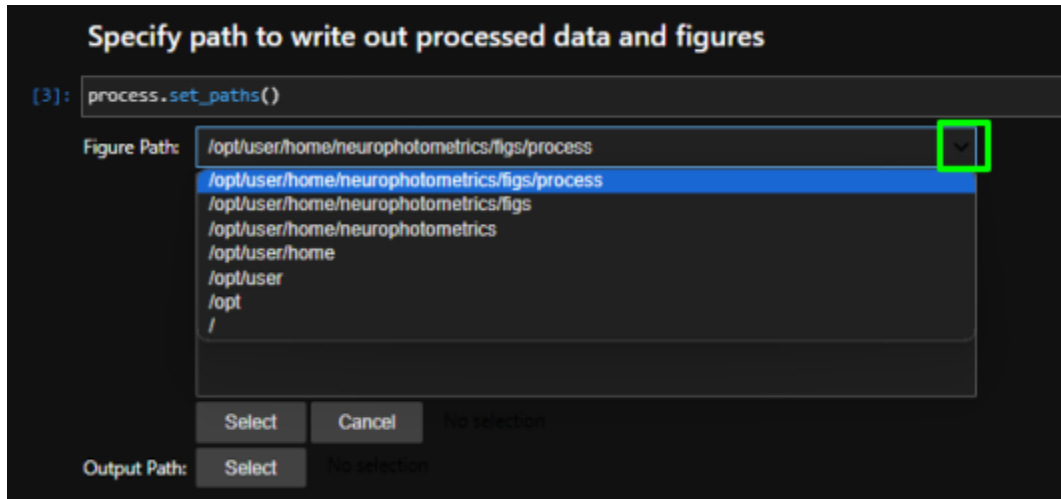
Select the folder where figures and processed FP data files generated in 01.pipeline.process will be saved. If no selection is made, files will save to the following default paths:

Data files: /opt/user/home/labname/output/process/signals

Figures: opt/user/home/labname/figs/process/

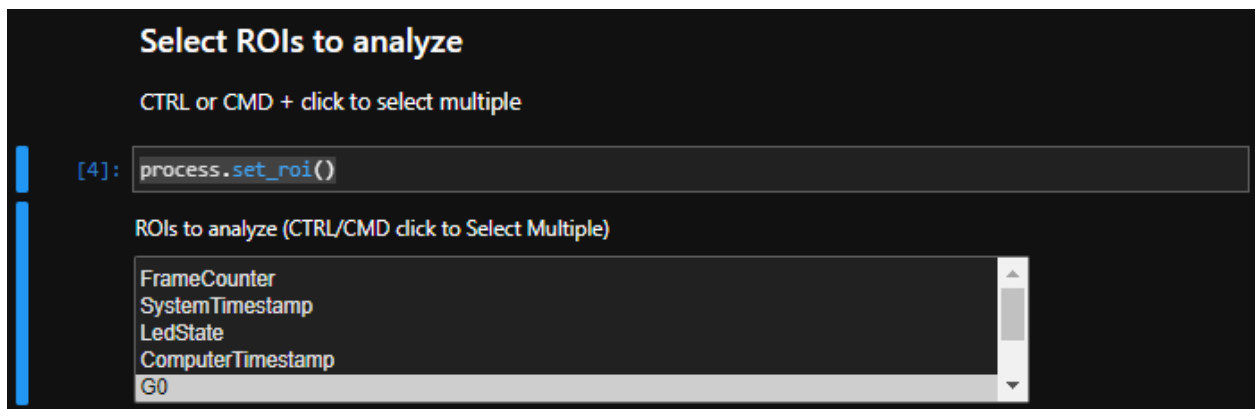
To navigate back to other folders, use the dropdown menu next to the path name. Note-processed data files will have a universal name for downstream processing steps, so if

you are analyzing multiple files, be sure to save them into different folders.



4. Select ROIs to analyze

Running this cell will populate a window with all of the column header names from your FP data file. Select the columns containing the data you would like to analyze. If you have extraneous data columns in your files, it is not necessary to include them in this step. Please only select columns containing data- labeled as G0, G1, R2, R3, etc. CTRL + click or CMD + click to select multiple.



5. Specify Region Names for ROIs

If you would like to rename data columns to specify the animal or brain region corresponding to each column, type preferred labels into the box(es) that appear after running the cell. These names will be used for figures generated in the workspace, and will not change raw data files.

6. Specify the number of frames to discard at the beginning of the file

The vast majority of fiber photometry traces follow an exponential decay-shaped curve, the steepest part of which occurs right at the beginning of the recording. Discarding frames from the start of the recording can improve curve fitting that occurs in later steps.

Use the slider or type in the number of frames you wish to discard. We recommend starting with 100 frames (at 60hz, this is less than 2 seconds of recording), unless your experiments necessitate analyzing the signal right at the start of the recording. If you do not wish to crop any frames, enter 0.

7. Select which channel to use as a reference signal

Fiber photometry signals are often reported as dF or dF/F , referring to the change in signal relative to some baseline signal, 'F'. In order to calculate dF or dF/F , we have to define what 'F', or the estimated baseline is. In theory, the true baseline signal is the resting fluorescence recorded when no cells are firing. In practice, without single-cell resolution, it is impossible to know if there are active cells or not, so we make an educated estimation.

In this step, there is an option to choose which signal to use as a reference signal for each excitation wavelength in each ROI using the dropdown menus. The raw signal itself will not be used as a baseline, but rather a biexponential fit of the signal. Note: all possible combinations of ROI and excitation wavelength are presented in this step regardless of the recording parameters used (i.e. 560 will always show up as an option even if only the 415 and 470 LEDs were used in the recording). Some of these can be ignored. For example, for recordings where only 415 and 470 LEDs were used, the reference for the 560 signal in each ROI can be left as 'No correction'. Below are our recommendations for what reference signals to choose.

For green ROIs where GCaMP was the sensor used, we recommend using the isosbestic channel (415 nm) to estimate the baseline. At the isosbestic point of GCaMP, the emission signal should be the same whether or not there is Ca^{2+} present, thus, any changes in fluorescence can be attributed to other sources like photobleaching of tissue autofluorescence, photobleaching of GCaMP, and motion. As these factors will also impact the emission signal when the 470 LED is on, the 415 nm channel is a convenient reference signal. Further, if your recording contains multiple green ROIs, we recommend using the 415 signal from the same ROI as the signal you are normalizing. For example, for a recording with ROI-A and ROI-B, we recommend using ROI-A 415 as the reference signal for ROI-A 415 and 470, and ROI-B 415 for ROI-B 415 and 470 signals.

For neurotransmitter sensors (dLight, iSeroSnFR, iGluSnFR etc), fewer conventions have been established. We recommend consulting the literature for the spectral properties of the sensor used. For sensors that are insensitive to 415 light, we recommend 'No correction' for 415 signals, and using the 470 signal itself as the reference for the 470 signal. If a static red indicator like tdTomato or mCherry was co-injected as a control signal, we recommend using the 560 signal from the corresponding red ROI as the reference signal for the 470 signal in green ROIs.

For red-shifted calcium indicators or neurotransmitters including RCaMP, jRGECO, rdLight etc, we recommend using the 560 signal (in a red ROI) itself as the reference.

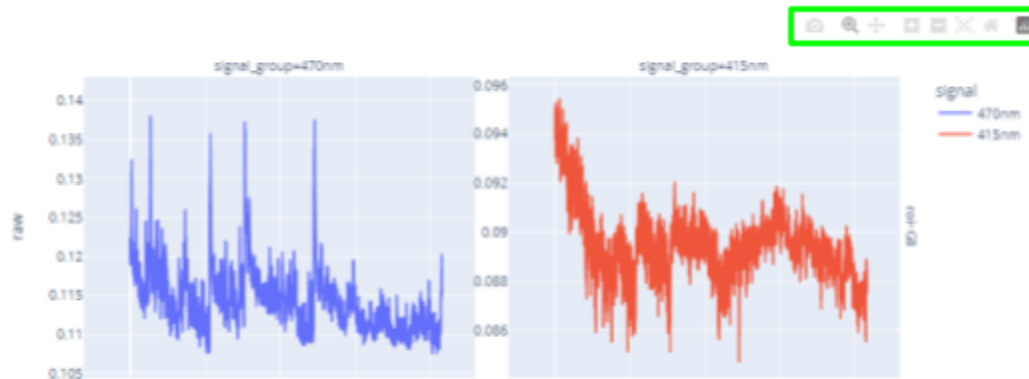
Below are recommended parameters for a recording with 2 green ROIs were GCaMP was the sensor used.

```
process.set_baseline()
```

Region A-560nm:	No correction	▼
Region A-415nm:	Region A-415nm	▼
Region A-470nm:	Region A-415nm	▼
Region B-560nm:	No correction	▼
Region B-415nm:	Region B-415nm	▼
Region B-470nm:	Region B-415nm	▼

8. Load and visualize raw data

Running this cell will plot the raw, deinterleaved data for selected ROIs in an interactive figure. The default view will show a zoomed in snippet of the data, use the buttons that appear when the mouse is hovered over the figure to zoom out or scroll through the data. Figures will save automatically in the path specified above, but to download the plot with the same zoom/scale parameters as is displayed, click the camera icon.

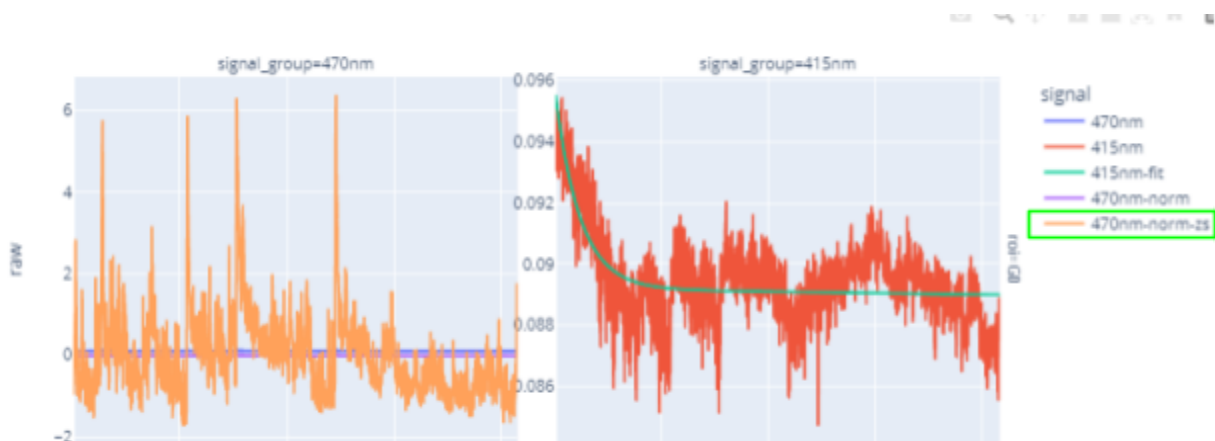


9. Correct for Photobleaching

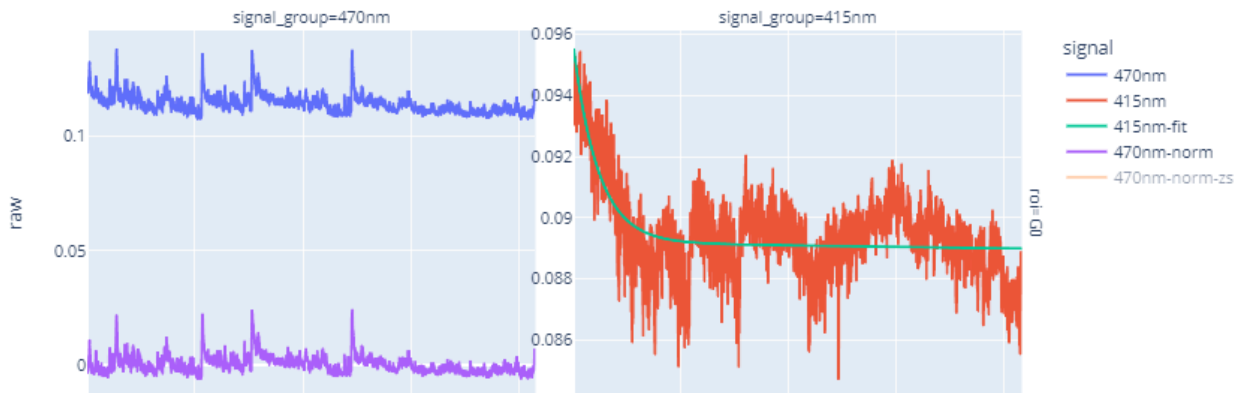
This step will correct for photobleaching in each signal using the reference signals specified above. To do this, the script will first deinterleave raw data to separate 415, 470, and 560 signals. Next, it will fit a biexponential curve (green) to the specified reference signal (red)

You may notice that in most recordings, the y-axis values for the reference signal/fitted curve are different from the signal of interest (blue).

To account for this, the script will scale the fitted reference signal to the signal of interest. It will then subtract the scaled and fitted reference from the raw signal of interest to yield normalized data (dF). Finally, it will z-score the normalized data and plot all traces.



You may notice that the raw 470 data (blue), and normalized dF 470 data (purple) look like flat lines- this is due to axes scaling. To hide the z-scored trace (yellow), click the trace on the figure legend. The axes will auto-scale.

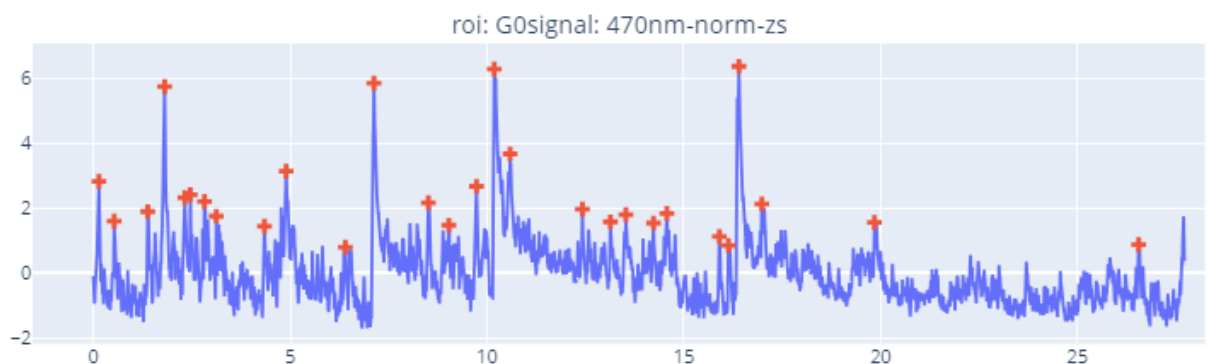


10. Peak Finding- Specify threshold for peak finding

The final step of this script is peak finding. It will detect peaks in z-scored data above a user-defined threshold. Running this cell will enable a slider widget where you can set the peak threshold to use (in z-scores). You can also type the number in.

11. Find Peaks and Plot

This will plot z-scored data, and label all peaks above threshold.



12. Save Processed Data

Running this cell will write out normalized data (dF) and z-scored data to the path specified above. Corrected data will save in a universal format for further processing in pipeline steps 2 and 3, and can be downloaded for plotting in other software.

Step 2: Timestamp Alignment

The purpose of this script is to align behavior timestamps of various formats with normalized photometry signals processed in step 1. This step will output a master file of timestamps aligned to FP data for plotting in step 3.

1. Initialize Timestamp Alignment

Run cell to initialize the alignment, once finished “Alignment initialized” will appear.

2. Select processed data file

Run cell, then use the select button to choose the processed data file you wish to use. Processed files are saved to the path specified in step 1. Normalized 470 and 560 data are saved as 470nm-norm.csv and 560nm-norm.csv

3. Specify where to save timestamps aligned with FP data

Select the folder where figures and timestamps aligned to FP data generated in 01.pipeline.process will be saved. If no selection is made, files will save to the following default paths:

Data files: /opt/user/home/labname/output/process/

Figures: opt/user/home/labname/figs/process/

To navigate back to other folders, use the dropdown menu next to the path name. Note-processed data files will have a universal name for downstream processing steps, so if you are analyzing multiple files, be sure to save them into different folders.

4. Select timestamp file(s) to align

Use the file browser to find the timestamp files you wish to align. The pipeline will automatically detect the format of selected files. The following timestamp formats are currently supported:

A. Timestamps generated using a KeyDown node in Bonsai

File Description	Column 1	Column 2	Column 3	Column 4	Column 5
Keydown	Key pressed	Computer TS			

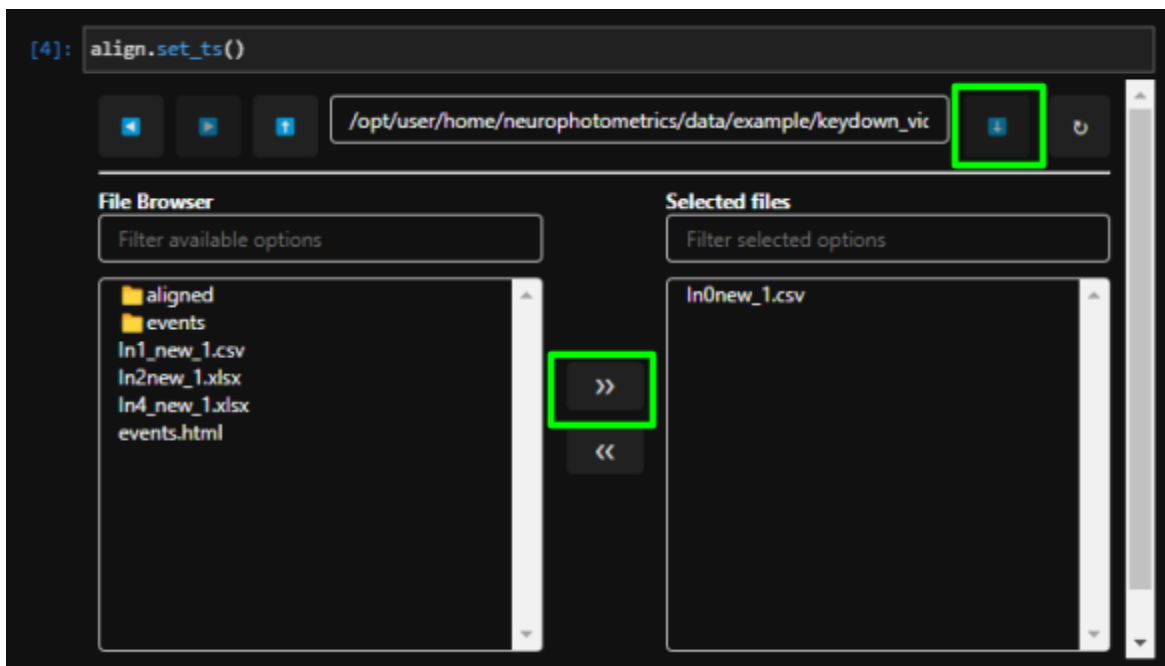
B. Hand scored behavior for videos recorded and timestamped through Bonsai. Note: files must be in this exact format and must be uploaded together

File Description	Column 1	Column 2	Column 3	Column 4	Column 5
Behavior Video frame: computer TS	Behavior camera frame #	Computer TS			
Hand scored behavior	Behavior camera frame #	Key press annotation or True/False			

C. Timestamps recorded with a Digital IO node in Bonsai

File Description	Column 1	Column 2	Column 3	Column 4	Column 5
Digital IO	Input that was triggered- human readable	Input that was triggered- computer readable	True/ False	FP system TS	Computer TS

Click the blue down arrow to open the selected folder. To select multiple files, use CTRL+click or CMD+ click. Then click the white double arrow button to add selected files.



5. Align timestamps with FP system clock

Run cell to align timestamps with FP data. Once finished, the pipeline will specify what timestamp format it has interpreted the selected files as.

Align timestamps with FP system clock

```
[5]: align.align_data()
```

```
Interpreting ExampleData_FC_timestamp.csv as ts_keydown  
aligned ExampleData_FC_timestamp.csv
```

6. Write out aligned timestamp file

Master file with aligned timestamps will write out to the path specified above and will be named 'master.csv'

Step 3: Data Pooling

This step will segment data processed in step 1 around timestamps aligned in step 2 to create peri-event plots.

1. Initialize Module

Run cell to initialize pooling

2. Select aligned timestamp file

Select the aligned timestamp file named 'master.csv'. Unless otherwise specified, aligned timestamp files are saved in the following path:

`/opt/user/home/labname/output/process/aligned/`

3. Select output path for figures and segmented photometry data

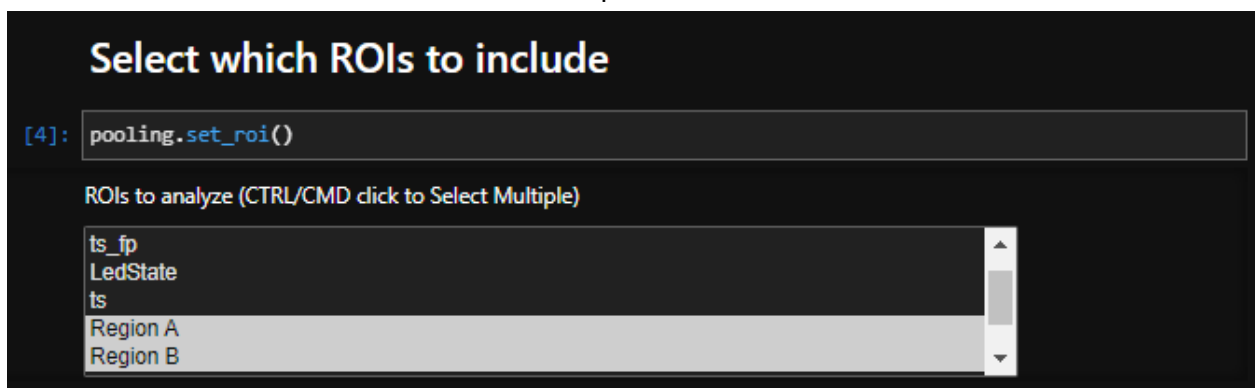
If no selection is made, figures and files will write out to the following paths:

Segmented FP data: `/opt/user/home/labname/output/process/`

Figures: `opt/user/home/labname/figs/process/`

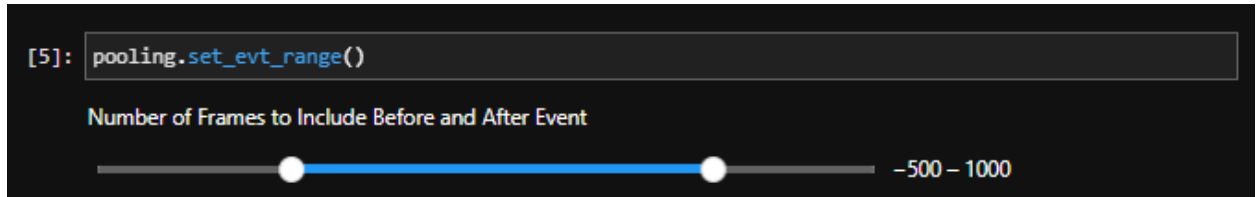
4. Select which ROIs to include in segmentation

Run cell to display column headers from master.csv, then select which ROIs you wish to include. The data columns are named according to the user-chosen ROI names in step 1. If names were not specified, the default name for data columns are G0, G1, R2, etc. Use CTRL+ click or CMD+ click to select multiple.



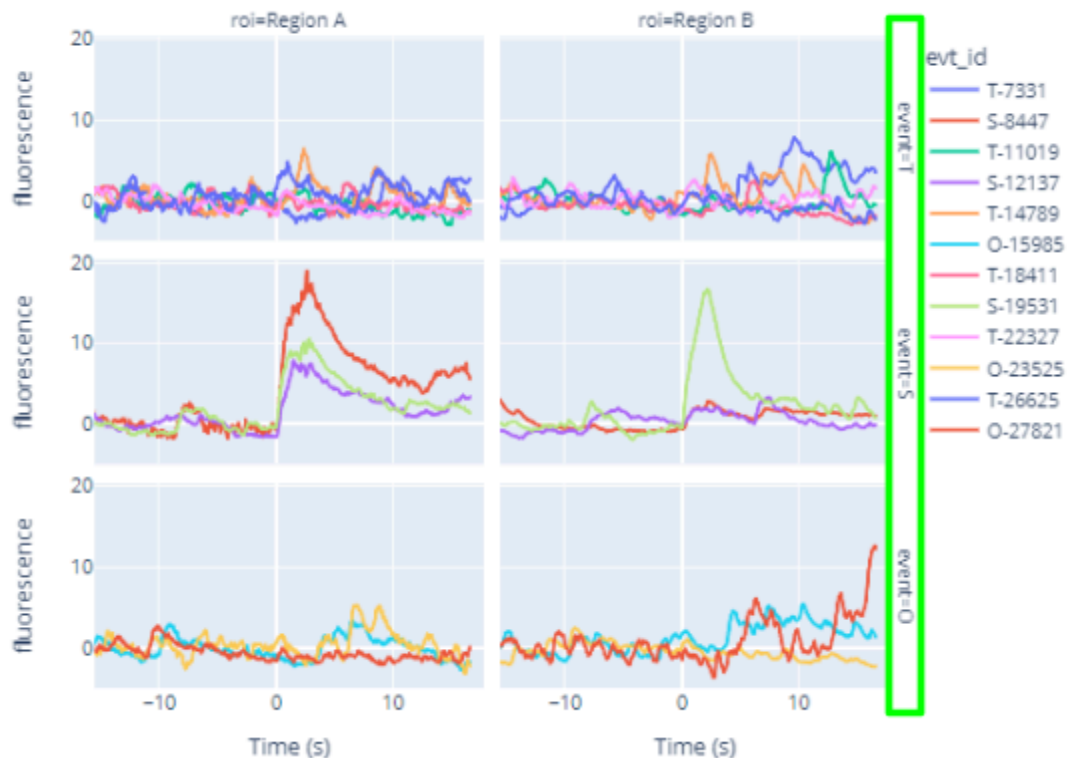
5. Choose the number of frames to include on either side of timestamp

Use the slider, or enter the number of frames that you wish to include before and after each timestamp. The first and second numbers refer to the number of frames before and after, respectively.



6. Create Peri-event plots

Run cell to segment data and create peri-event plots. The time on the x-axis is shown as time in seconds relative to the timestamp of interest. For keydown behaviors, different key presses will be plotted on different axes, and the event type will be specified on the right-hand y-axis.



7. Export and save data

Segmented data will save to the path specified above