Tools for Quantitative Analysis of Calcium Signaling Data Using Jupyter-Lab Notebooks

John Rugis[a,c]
James Chaffer[b]
James Sneyd[a]
David Yule[b]

[a]Department of Mathematics, University of Auckland, New Zealand.
[b]Department of Pharmacology and Physiology, University of Rochester, Rochester, NY. USA.
[c]Corresponding author j.rugis@auckland.ac.nz

Supplemental Notes

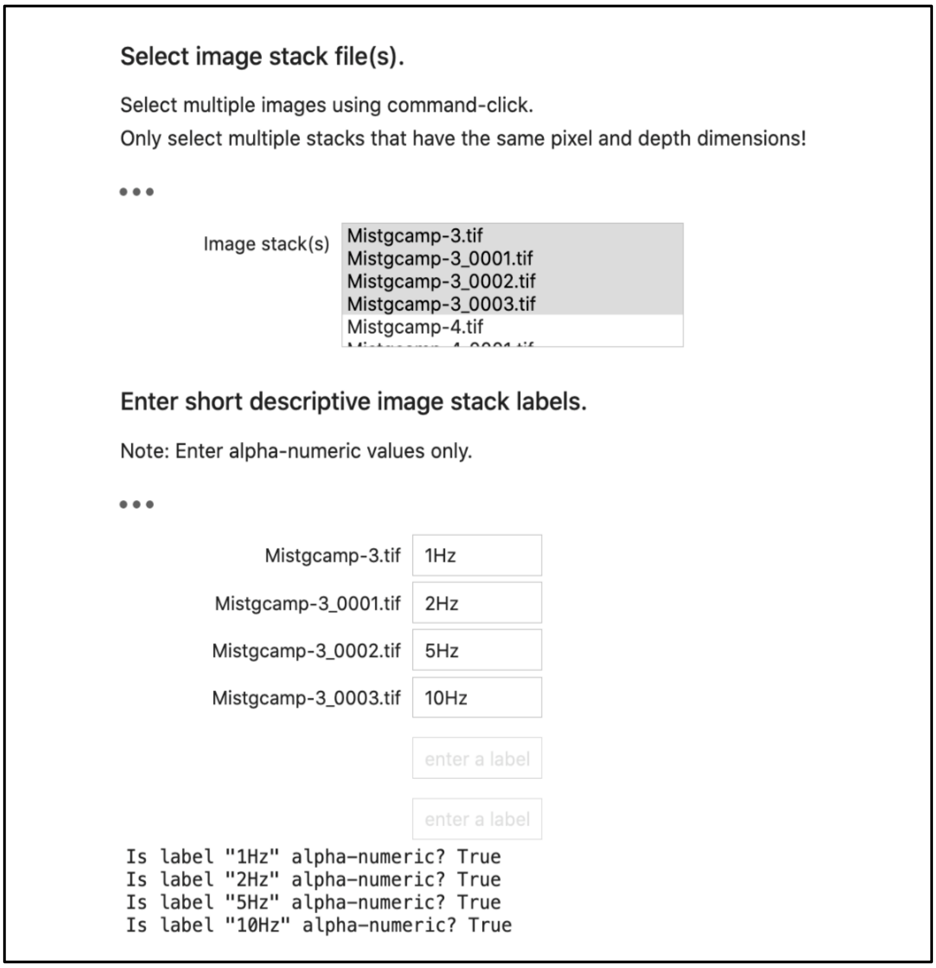
1. The Analysis Notebooks

The notebook collection is available for download from a repository on GitHub at the link found in reference [1]. For quick access to instructions on downloading the notebooks (as well as instructions for installing Python and Jupyter-Lab) click on the following link: https://github.com/jrugis/cell_tools/blob/master/Installation_Instructions.txt

Note that the notebooks are subject to ongoing update and revision. Please email the corresponding author of this paper to report any issues with notebook installation and functionality.

1.1 Notebook User Interface

The typical notebook usage process within Jupyter consists of sequentially executing interactive input blocks along with non-interactive analysis script code blocks. You can execute highlighted blocks in Jupyter-Lab by clicking on the triangular "play" button in the notebook menu bar. Note that you need to "click out" of interactive code blocks. For more details on the Jupyter-Lab user interface, refer to one of the many online tutorials.



Select image stack file(s).

Select multiple images using command-click.
Only select multiple stacks that have the same pixel and depth dimensions!

Image stack(s)
Mistgcamp-3.tif
Mistgcamp-3_0001.tif
Mistgcamp-3_0002.tif
Mistgcamp-3_0003.tif
Mistgcamp-4.tif

Enter short descriptive image stack labels.

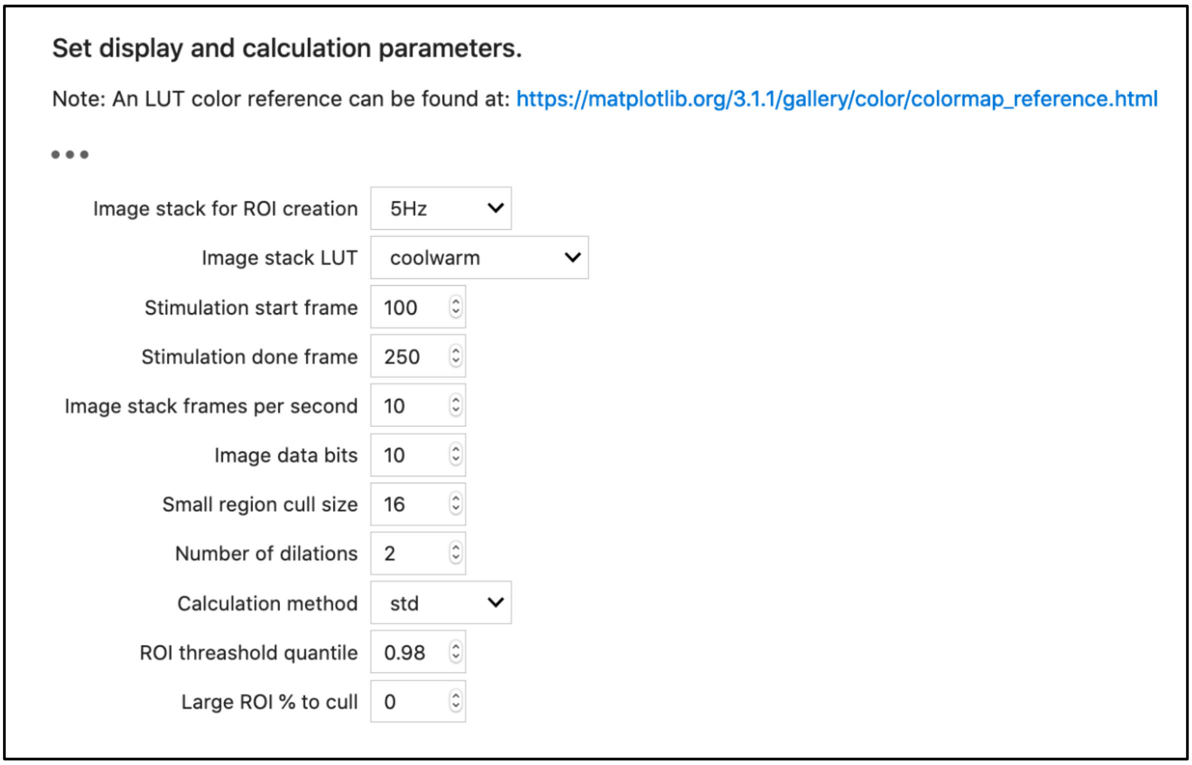
Note: Enter alpha-numeric values only.

Mistgcamp-3.tif	1Hz
Mistgcamp-3_0001.tif	2Hz
Mistgcamp-3_0002.tif	5Hz
Mistgcamp-3_0003.tif	10Hz
	enter a label
	enter a label

```
Is label "1Hz" alpha-numeric? True
Is label "2Hz" alpha-numeric? True
Is label "5Hz" alpha-numeric? True
Is label "10Hz" alpha-numeric? True
```

Figure 1 - Screenshot of interactive panel for selecting image stacks and entering descriptive labels. The labels are error checked to ensure that they are contain only alpha-numeric characters.

As an example of an interactive input block, Figure 1 is a screenshot from the ROI_Detection notebook showing the panel for selecting image stacks and creating descriptive stack labels. Note the error checking feedback displayed to the user in the bottom of this panel.



Set display and calculation parameters.

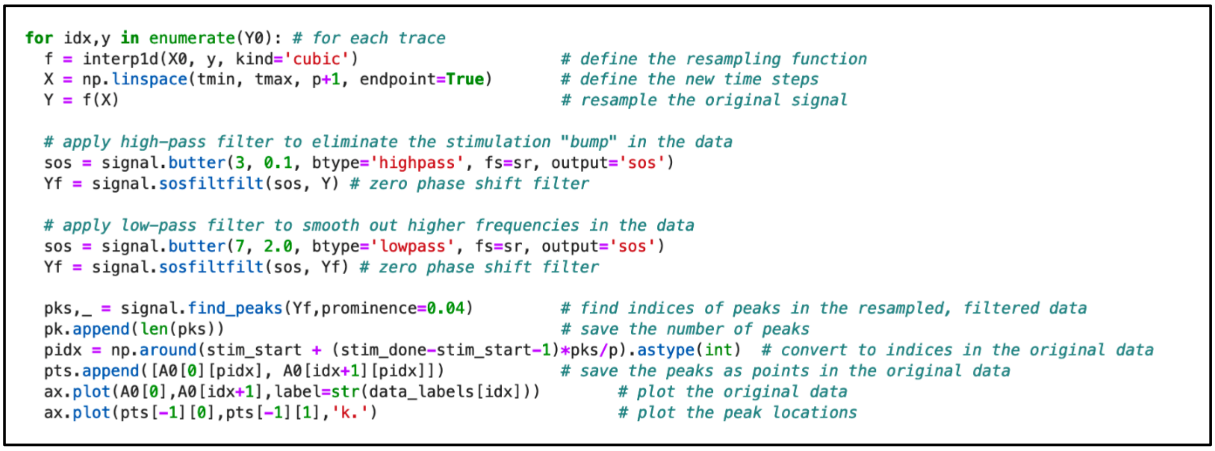
Note: An LUT color reference can be found at: https://matplotlib.org/3.1.1/gallery/color/colormap_reference.html

•••

Image stack for ROI creation	5Hz
Image stack LUT	coolwarm
Stimulation start frame	100
Stimulation done frame	250
Image stack frames per second	10
Image data bits	10
Small region cull size	16
Number of dilations	2
Calculation method	std
ROI threashold quantile	0.98
Large ROI % to cull	0

Figure 2 - Screenshot of interactive panel for ROI identification parameter settings. "Image stack LUT" refers to the color look up table that is used to color code displayed results. Stimulation start and done frames allow the user to specify the time range in which cell response to stimulation is expected. The frames per second and image data bits numbers are used to calibrate calculated results. "Small region cull size" is used to both remove spurious individual noise pixels as well remove regions whose pixel count is too low to be statistically meaningful. "Number of dilations" is used to remove small holes in, as well as smooth the edges of, regions of interest. The specified "Calculation method" is used to combine, per-pixel, the stimulated portion of the image stack. "RIO threshold quantile" sets the sensitivity for whether a (combined) pixel is part of a region of interest. "Large ROI % to cull" can be used to optionally eliminate overlapping regions of interest (which likely include multiple cells).

Figure 2 is a screenshot from the same notebook showing the interactive panel for parameter and option settings. The range of available options gives the user some flexibility in analysis and helps generalize the tool.



```
for idx,y in enumerate(Y0): # for each trace
  f = interp1d(X0, y, kind='cubic')                    # define the resampling function
  X = np.linspace(tmin, tmax, p+1, endpoint=True)      # define the new time steps
  Y = f(X)                                             # resample the original signal

  # apply high-pass filter to eliminate the stimulation "bump" in the data
  sos = signal.butter(3, 0.1, btype='highpass', fs=sr, output='sos')
  Yf = signal.sosfiltfilt(sos, Y) # zero phase shift filter

  # apply low-pass filter to smooth out higher frequencies in the data
  sos = signal.butter(7, 2.0, btype='lowpass', fs=sr, output='sos')
  Yf = signal.sosfiltfilt(sos, Yf) # zero phase shift filter

  pks,_ = signal.find_peaks(Yf,prominence=0.04)        # find indices of peaks in the resampled, filtered data
  pk.append(len(pks))                                  # save the number of peaks
  pidx = np.around(stim_start + (stim_done-stim_start-1)*pks/p).astype(int)  # convert to indices in the original data
  pts.append([A0[0][pidx], A0[idx+1][pidx]])           # save the peaks as points in the original data
  ax.plot(A0[0],A0[idx+1],label=str(data_labels[idx]))       # plot the original data
  ax.plot(pts[-1][0],pts[-1][1],'k.')                        # plot the peak locations
```

Figure 3 - Notebook peak detection script extract.

As an example of an analysis script code block, Figure 3 shows an extract from the ROI_Detection notebook. In most cases, users will simply execute, and not modify, code blocks. However, users can, if desired, modify the contents of code blocks to alter and fine tune notebook functionality.

1.2 Frequency Analysis

The frequency analysis notebook can be used to further characterize stimulation response curves.

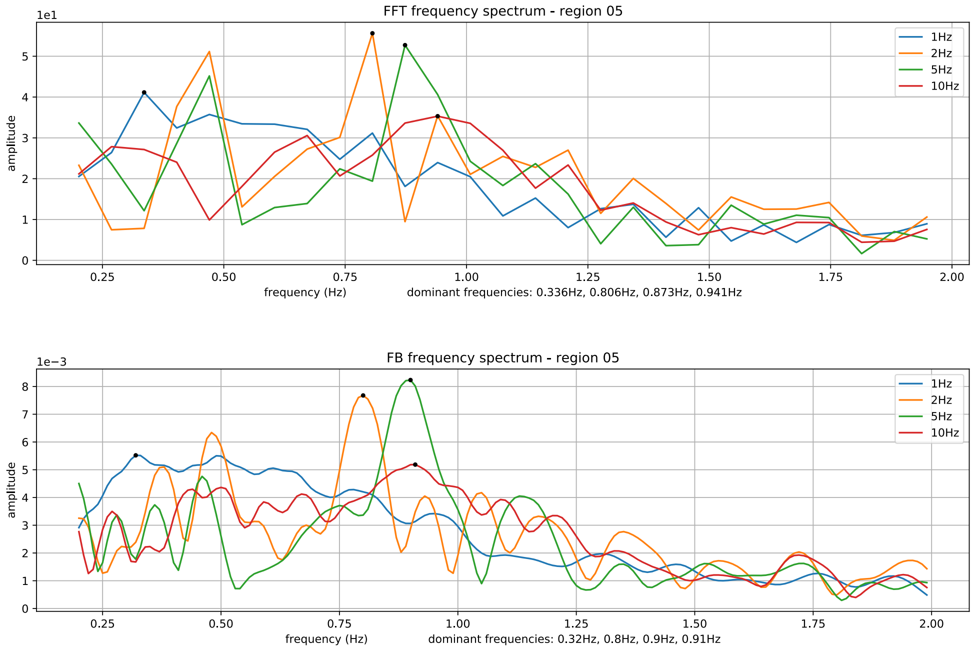


Figure 4 - Frequency response of a sample region the Mistgcamp-3 dataset using the FFT method (top) and the filter bank method (bottom).

Frequency analysis results for the same sample region in the Mistgcamp-3 data are shown in Figure 4, with the FFT method on top and the filter-bank method on the bottom. The dominant frequency in each trace is marked with a black dot. The FFT method gives frequencies at clearly separated discrete values. Alternatively, in this case, the filter-bank method gives a much more continuous response, having approximately ten times the resolution.

1.3 Movie Making

With any physiological response data that is analyzed in time, it can be insightful to display the results via animation. To that end, we designed a movie making notebook.

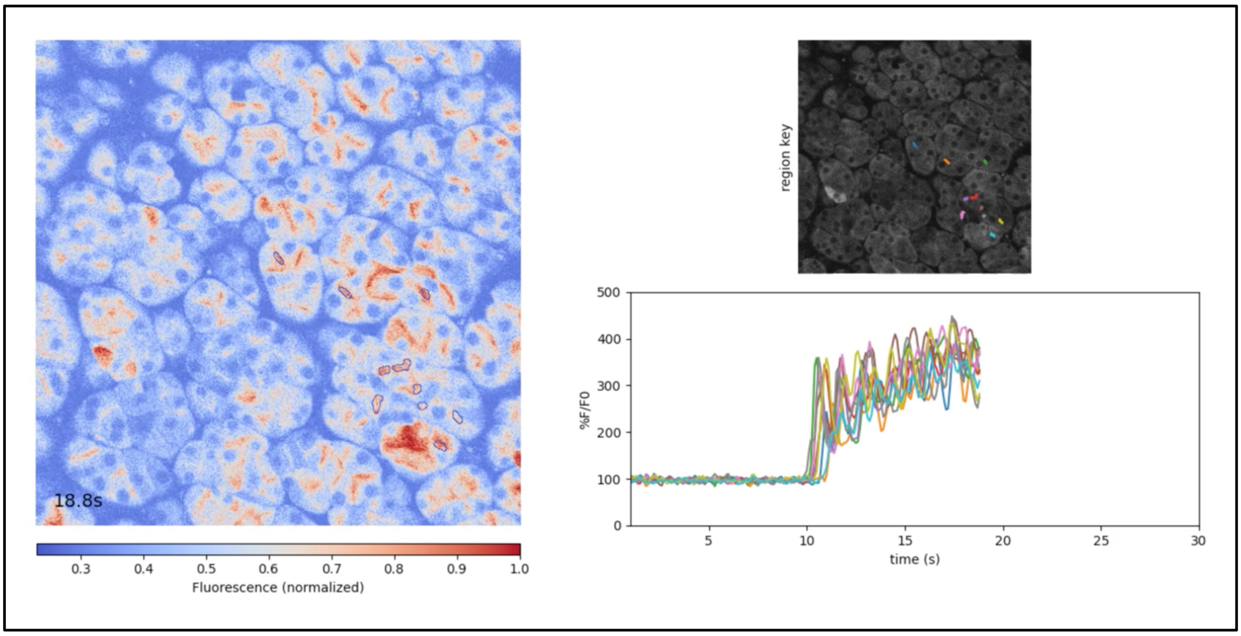


Figure 5 - Movie screenshot, part way through dynamic display of image sequence and its coupled plot.

A display screenshot of a movie of the Mistgcamp-3 dataset partway through playback is shown in Figure 5. Live playback of the cell image stack is coupled with a color-coded region-of-interest key and an associated color-coded real-time plot of multiple regions. Note that, during playback, the plot is dynamically drawn, left to right, in sync with progress through the image stack.

2. Experiments

Step-by-step details for notebook usage in each of the experiments follows. The notebook collection itself is available for download from a repository on GitHub[1]. For quick access to instructions on downloading the notebooks (as well as instructions for installing Python and Jupyter-Lab) click on the following link:
https://github.com/jrugis/cell_tools/blob/master/Installation_Instructions.txt

Note that the notebooks are subject to ongoing update and revision. Please email the corresponding author of this paper to report any issues with notebook installation and functionality.

Additionally, the example datasets will need to be downloaded as indicated in the experiment sections that follow.

2.1 CCh and Trypsin Experiments

For quick access to instructions on downloading the CCh and Trypsin example datasets, click on each of the following links respectively:
https://github.com/jrugis/cell_tools/blob/master/Experiment_Instructions_CCH.txt
https://github.com/jrugis/cell_tools/blob/master/Experiment_Instructions_Trypsin.txt
Follow the first part of the instructions to download the datasets. The remaining steps in the instructions give a brief outline of what is described in more detail below.

The process of analyzing these experiment types begins with placing either the provided trypsin experiment folders and/or the provided CCH experiment folders into the image_stacks folder within cell_tools base directory. These experiment folders each contain a 340 nm tiff image stack file and a 340/380 nm ratio tiff image stack file. The “Package Import” and “File Import” code blocks of Ratiometric_ROI_Detection can then be run, and an FPS (frames per second) value can be chosen appropriate for the image stacks (in this case it is 1 FPS). The “Create Result(s) Folder(s)” and “Threshold Sizing Baseline Image(s)” code blocks can then be run.

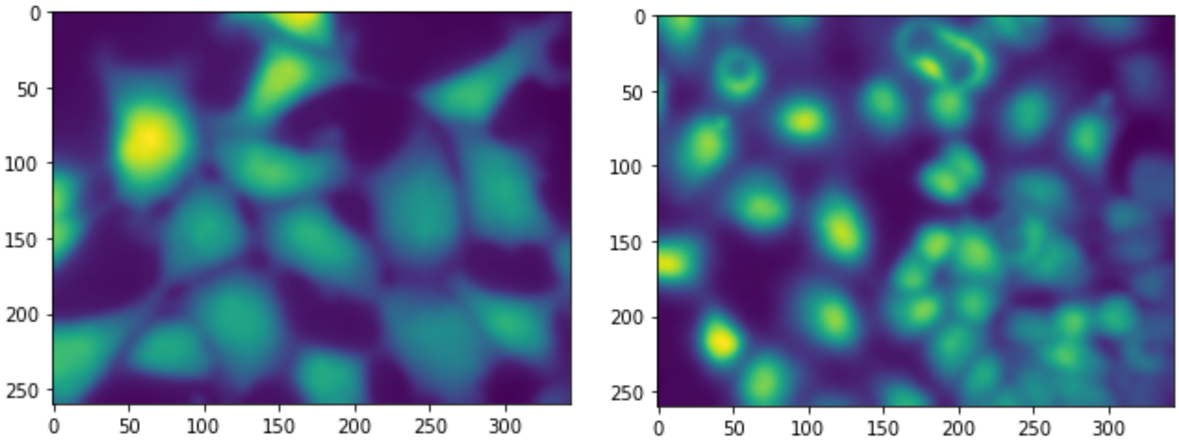


Figure 6 - Smoothed 340 nm image stack frame of Trypsin Experiment 4 (left) and CCH Experiment 5 (right)

This results in the output of a smoothed version of the first 340 nm image stack frame from each experiment being displayed (as in, for example Figure 6). Along with this display an option for threshold sizing is offered, which for these sample experiments should be left at “Small”.

The “Initial Mask Creation” code block can then be run, which results in a side-by-side of the smoothed 340 nm image stack frame and initial mask being displayed along with some options for erosions and ROI rejection by size or location. These options allow some flexibility in mask creation and should be left to their default values for the example experiments.

Adaptive thresholding is applied to the first frame of the 340 nm image stack in order to create a binary mask. Bounding boxes are then created for each object in the binary mask, with each object possibly being a cluster of cells. A negative normalized distance map is made for each binary mask object contained within a bounding box, and the same bounding box coordinates are used to obtain the normalized intensity values of the objects from the original 340 nm frame. These intensity values are then subtracted from the normalized distance map values, resulting in a distance transform map that accounts for intensity. The improved distance map for each object is then individually segmented using watershed segmentation. This method allows for accurate segmentation of heavily overlapped cells better than using a distance map alone [2]. These segmented objects are then combined using the bounding box coordinates into a single image frame.

The resulting segmented image is then further processed, with Canny edge detection and standard computer graphics flood fill being used to geometrically seal off the sides of ROIs that fall along the edge of the image frame (if the user chooses to keep edge ROIs). The first frame of the ratio image stack is converted to a binary mask using thresholding, and a bitwise AND operation is then used to eliminate any objects in the segmented image which do not have a corresponding presence in the ratio image mask.

The result is a segmented ratio image mask, which is labelled for ROI analysis. The ROI analysis is performed the exact same way as in the ROI_Detection notebook, with the average pixel response being plotted and values saved in csv files. These values contained within the csv files are then used in the csv analysis notebooks.

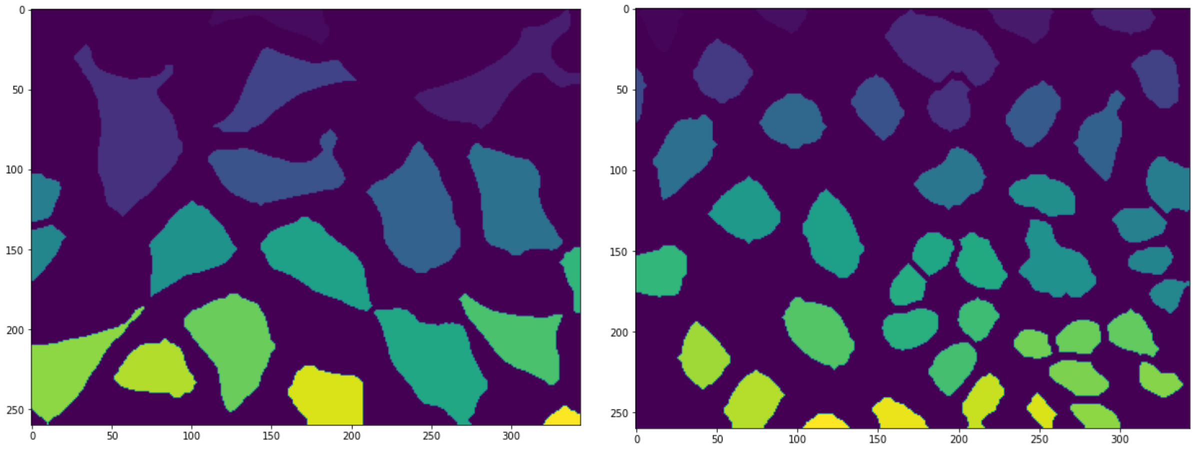


Figure 7 - Initial segmented masks displayed after running "Segmentation" code block of Trypsin Experiment 4 (Left) and CCH Experiment 5 (Right)

The "Segmentation" code block can then be run, which results in the display of the segmented masks for each experiment as can be seen, for example, in Figure 7.

The "Create Removal Mask(s)" and "Display Removal Mask(s)" code blocks can then be run, which further process the masks and display a labeled mask for each experiment for any final ROI removals by the user. No final ROI removals are required for the example experiments, so the "Submit Removal Selection(s)" and "Finalize Mask(s)" code blocks can be run.

Finally, in the same notebook, the "Plotting and Data Collection" code block can be run, which will begin filling the experiment specific "Results" folders with plots and csv file outputs. If desired, the "Optional plotting of individual pixel intensities of each ROI and average" code block can be run, but running it is not required for moving onto the next analysis.

Next, using the CSV_Peak_Detection notebook, the "Package Import" and "Select data files for analysis" code blocks can be run. Either the Trypsin Experiments or the CCH Experiments should be selected from the drop-down menu and the "Create a new analysis results directory" code block can be run. The "Choose which region columns to analyze" code block can then be run, which will display options for selecting which region columns to analyze and whether to remove “bad” traces automatically and link stimulation zones. For the example experiments, these options should be left unaltered.

The “Initial Data Plotting” code block can then be run, which will display a plot of the ROI traces along with options for number of anomalies/artifacts to remove for each file and number of stimulation zones for all files. For the Trypsin Experiment examples the number of stimulation zones should be “1”, and for the CCH Experiment examples it should be “3”.

Since no anomalies/artifacts are being removed, the anomalies/artifacts code blocks can be skipped and the next code block to run will be the “Set Stimulation Zones” code block. This will result in the display of the ROI traces of each file along with a slider for setting the time boundaries for each stimulation zone (one for Trypsin and three for CCH). The stimulation zone values for the CCH Experiment example should be set to 50-300, 350-600, and 650-900. The stimulation zone values for the Trypsin Experiment example should be set to 80-250.

Next the “Plot data and stimulation time zones” code block should be run, which displays the selected stimulation zone(s) on the plot(s) themselves. There are also options displayed for starting and automatic baselines, which should both be checked for running these example experiments. The “Peak counting and latency calculation settings” code block can then be run, which outputs another plot showing the stimulation zones and a series of options. For the example experiments “peak counting sensitivity”, “threshold standard deviations”, “response % standard deviations”, “length of starting baselines”, and “length of automatic baselines” should be left at their default values. “Slope calculation method” however, should be changed to “gaussian fit”. The “Plot peaks and calculate latency settings” code block can then be run. Analyzed plots will be displayed sequentially.

The plots are stored, along with other analysis files, in the analysis results folders found in the experiment specific results folders mentioned earlier. The analysis files consist of a csv file for every region and contain the time and height of each peak identified within that region’s trace. Note that the height values are raw and are not relative to the baseline of the trace. Also included is a full plot of all the region traces, in both pdf and png file format.

An Excel file is included which contains the time and average value over all the regions, along with the standard deviation at each of these time points. A pdf of the standard deviation plot is also included. Finally, included is a “total_data_by_zone” csv file, which contains the same values as displayed in the individual region plots. This data is formatted as to be convenient to do further analysis if the user so wishes, with stimulation zones being separated by empty rows.

The first of these “total_data_by_zone” values is the peak count, which is useful for measuring the number of oscillations in the trace. The next value is the first peak value, which is useful for quantifying the initial stimulation response of the trace. The next value is the area, this being the area under the curve with the baseline subtracted. This area value allows for quantification of the total response of the ROI to stimulation.

Another value included is latencies, which allows for measurement of the time between stimulation and response in the trace. The latency point, which is calculated by finding the point at which the traces rise above the threshold value, is used in calculating this value. Another time value included in the csv file is rise times, which allows for measurement of the time it took for the trace to go from the latency point to the following peak point. This allows the user to quantify the speed of the stimulation response once it has begun. The next value is the maximum slope, which is the maximum slope value obtained from whichever slope calculation method was chosen by the user. This allows for the user to identify the maximum rate of increase of the Ca2+ fluorescence.

The next two values are the unstimulated average and starting unstimulated average, which are the average within the user-chosen time length of the trace right before the stimulation zone and at the beginning of the trace respectively. These values are useful for identifying a change in the baseline of the trace prior to the stimulation zone. The final value contained in this csv file is the responding percentage, which measures the percentage of traces which went above the user-set response % threshold. This allows for quick determination of the extent to which there was a response in the experiment.

2.2 SOCE Experiment

For quick access to instructions on downloading the required example dataset, click on each of the following link:
https://github.com/jrugis/cell_tools/blob/master/Experiment_Instructions_CPA.txt
Follow the first part of the instructions to download the dataset. The remaining steps in the instructions give a brief outline of what is described in more detail below.

The process of analyzing this experiment type begins with placing the provided CPA experiment folders into the image_stacks folder within the cell_tools base directory. These experiment folders each contain a 340 nm tiff image stack file and a 340/380 nm ratio tiff image stack file.

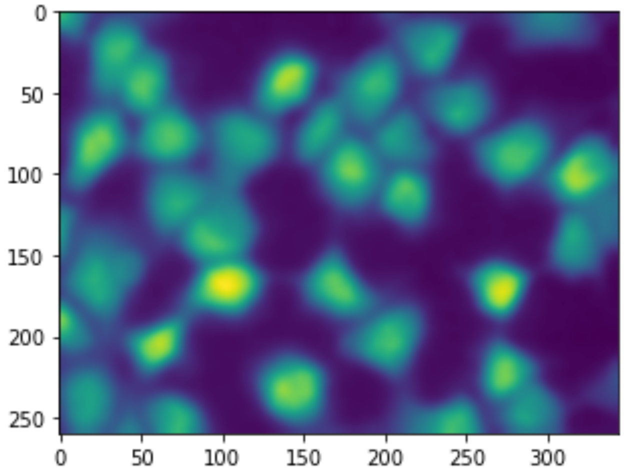


Figure 8 - Smoothed 340 nm image stack frame of CPA Experiment 1

Analysis begins with the Ratiometric_ROI_Detection notebook, using the same steps and settings as with the CCh and trypsin experiments. Refer back to the CCh and trypsin experiments for those steps. However, the plots and analysis output will now be those associated with the CPA experiment. The output of a smoothed version of each CPA experiment first 340 nm image stack frame will be displayed as in, for example, Figure 8.

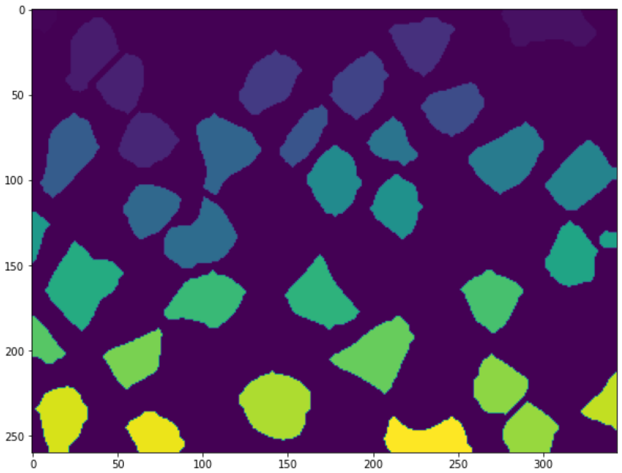


Figure 9 - Initial segmented mask displayed after running the “Segmentation” code block of CPA Experiment 1

The “Segmentation” code block will produce the display of the segmented masks for each experiment as can be seen, for example, in Figure 9. The “Create Removal Mask(s)” and “Display Removal Mask(s)” code blocks will display a labeled mask for each experiment.

Opening the CSV_Peak_Detection_SOCE notebook, the “Package Import”, and “Select data files for analysis”, code blocks can be run. The CPA experiments should be selected from the drop-down menu and the “Create a new analysis results directory” code block can be run. The “Choose which region columns to analyze” code block can then be run, which will display options for selecting which region columns to analyze and whether to remove “bad” traces automatically and link stimulation zones. For the example experiments, these options should be left unaltered.

The “Initial Data Plotting” code block can then be run, which will display a plot of the ROI traces along with options for number of anomalies/artifacts to remove for each file and number of stimulation zones for all files. For the CPA Experiment examples the number of stimulation zones should be set to “2”.

Since no anomalies/artifacts are being removed, these code blocks can be skipped and the next code block to run will be the “Set Stimulation Zones” code block. This will result in the display of the ROI traces of each file along with a slider for setting the time boundaries for each stimulation zone, two in this case. The stimulation zone values for the CPA experiment examples should be 200-800 s and 900-1074 s.

Next the “Plot data and stimulation time zones” code block should be run, which displays the selected stimulation zones on the plots themselves. There are also options displayed for starting and automatic baselines, which should both be checked for running the example experiments. The “Calculation settings” code block can then be run, which outputs another plot showing the stimulation zones and a series of options. For the example CPA experiments “threshold standard deviations”, “response % standard deviations”, “length of starting baselines", and "length of automatic baselines" should be left at their default values. "Slope calculation method" however, should be changed to "polynomial fit". The "Plot analysis by region" code block can then be run and analyzed plots will be displayed sequentially while also being sent, along with other analysis files, to the analysis results folders found in the experiment specific results folders mentioned earlier.

Generated analysis files include a pdf file of the same plot output displayed by the notebook for each region trace. An Excel file is included which contains the time and average value over all the regions, along with the standard deviation at each of these time points. A pdf of the standard deviation plot is also included. Finally, included is a "total_data_by_zone" csv file, which contains the same values as displayed in the individual region plots. This data is formatted as to be convenient to do further analysis, with stimulation zones being separated by empty rows.

The first of these "total_data_by_zone" values is the first peak value, which is useful for quantifying the initial stimulation response of the trace. It is important to note that in the case of the SOCE notebook the first peak value indicates the maximum point of the trace without the baseline subtracted.

The next value is the area, this being the area under the curve with the baseline subtracted. This area value allows for quantification of the total response of the ROI to stimulation. Another value included is the maximum value which is the same as the first peak value except with the baseline subtracted. This is useful for comparing the maximums between different experiments, as it eliminates the discrepancy of the baselines.

Another value included is latencies, which allows for measurement of the time between stimulation and response in the trace. The latency point, which is calculated by finding the point at which the traces rise above the threshold value, is used in calculating this value. Another time value included in the csv file is rise times, which allows for measurement of the time it took for the trace to go from the latency point to the following peak point. This allows the user to quantify the speed of the stimulation response once it has begun. The next value is the maximum slope, which is the maximum slope value obtained from whichever slope calculation method was chosen by the user. This allows for the user to identify the maximum rate of increase of the Ca2+ fluorescence.

The next two values are the unstimulated average and starting unstimulated average, which are the average within the user chosen time length of the trace right before the stimulation zone and at the beginning of the trace respectfully. These values are useful for identifying a change in the baseline of the trace prior to the stimulation zone. To make measuring this change in SOCE experiments more streamlined, a difference value is also included which is the unstimulated average subtracted from the starting unstimulated average.

The final value contained in this csv file is the responding percentage, which measures the percentage of traces which went above a user set threshold. This allows quick determination of the extent to which there was a response in the experiment. Each of the values contained in the "total_data_by_zone" csv file is useful for quantifying a specific aspect of the traces, and together allow for a detailed quantitative analysis of the experiment.

2.3 GSK Experiment

For quick access to instructions on downloading the required example dataset, click on the following link:
https://github.com/jrugis/cell_tools/blob/master/Experiment_Instructions_GSK.txt
Follow the first part of the instructions to download the dataset. The remaining steps in the instructions give a brief outline of what is described in more detail below.

The process of analyzing this experiment type begins with placing the provided GSK experiment folders into the image_stacks folder within the cell_tools base directory. These experiment folders each contain a 340 nm tiff image stack file and a 340/380 nm ratio tiff image stack file.

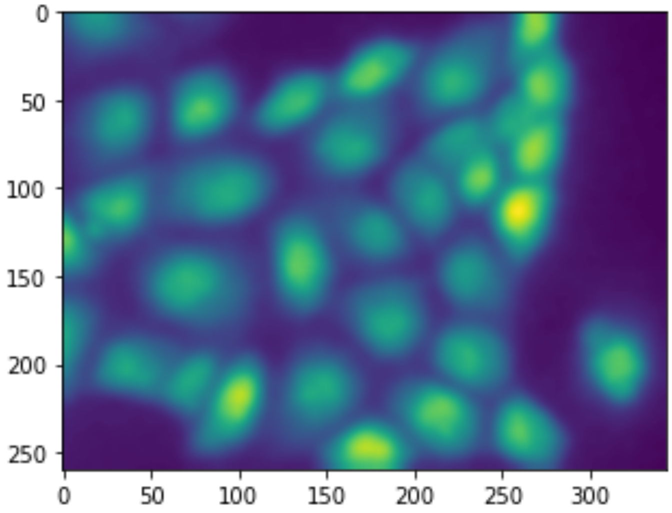


Figure 10 - Smoothed 340 nm image stack frame of GSK Experiment 3

Analysis begins with the Ratiometric_ROI_Detection notebook, using the same steps and settings as with the CCh and trypsin experiments. Refer back to the CCh and trypsin experiments for those steps. However, the plots and analysis output will now be those associated with the GSK experiment. The output of a smoothed version of each GSK experiment first 340 nm image stack frame will be displayed as in, for example, Figure 10.

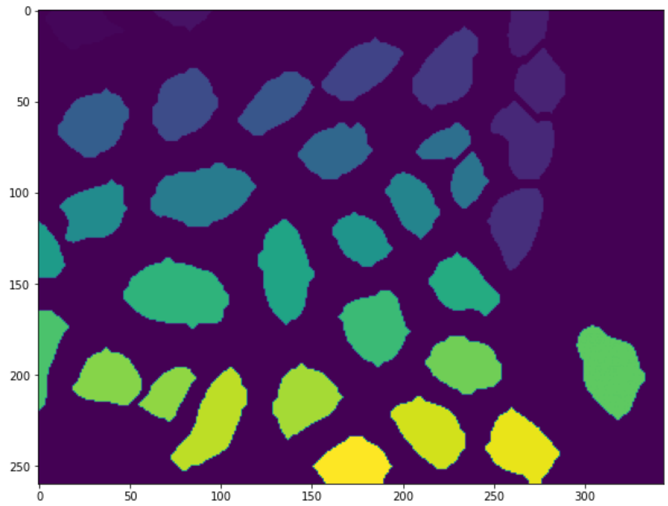


Figure 11 - Initial segmented mask displayed after running "Segmentation" code block of GSK Experiment 3

The "Segmentation" code block will produce the display of the segmented masks for each experiment as can be seen, for example, in Figure 11.

The "Create Removal Mask(s)" and "Display Removal Mask(s)" code blocks will display a labeled mask for each experiment.

Opening the GSK_Peak_Detection_Inhibitor notebook, the "Package Import", and "Select data files for analysis", code blocks can be run. The GSK experiments should be selected from the drop-down menu and the "Create a new analysis results directory" code block can be run. The "Choose which region columns to analyze" code block can then be run, which will display options for selecting which region columns to analyze and whether to remove "bad" traces automatically and link stimulation zones. For the example experiments, these options should be left unaltered.

The "Initial Data Plotting" code block can then be run, which will display a plot of the ROI traces along with options for number of anomalies/artifacts to remove for each file and number of stimulation zones for all files. For the GSK experiment examples the number of stimulation zones should be set to "1".

Since no anomalies/artifacts are being removed, these code blocks can be skipped and the next code block to run will be the "Set Stimulation Zones" code block. This will result in the display of the ROI traces of each file along with a slider for setting the time boundaries for each stimulation zone, one in this case. The stimulation zone values for the GSK Experiment examples should be 50-200s.

Next, the "Plot data and stimulation time zones" code block should be run, which displays the selected stimulation zone on the plots themselves. There are also options displayed for starting and automatic baselines, which should both be checked for running the example experiments. The "Calculation settings" code block can then be run, which outputs another plot showing the stimulation zones and a series of options. For the example GSK experiments “threshold standard deviations”, “response % standard deviations”, “length of starting baselines”, and “length of automatic baselines” should be left at their default values. “Slope calculation method” however, should be changed to “polynomial fit”. The “Plot analysis by region” code block can then be run and analyzed plots will be displayed sequentially while also being sent along with other analysis files to the analysis results folders found in the experiment specific results folders mentioned earlier.

Generated analysis files include a pdf file of the same plot output displayed by the notebook for each region trace. An Excel file is included which contains the time and average value over all the regions, along with the standard deviation at each of these time points. A pdf of the standard deviation plot is also included.

Finally, included is a “total_data_by_zone” csv file. The first of these “total_data_by_zone” values are the minimum, which is the minimum point of the trace within the stimulation zone without the baseline subtracted. The next value is the area, this being the area under the curve with the baseline subtracted. This area value allows for quantification of the total response of the ROI to stimulation. Another value included is the maximum value which is the maximum point of the trace within the stimulation zone, and in the case of the Inhibitor notebook output the baseline is not subtracted.

Another value included is latencies, which allows for measurement of the time between stimulation and response in the trace. The latency point, which is calculated by finding the point at which the traces fall below the threshold value, is used in calculating this value.

Another time value included in the csv file is fall times, which allows for measurement of the time it took for the trace to go from the latency point to the minimum point within the stimulation zone. This allows the user to quantify the speed of the stimulation response once it has begun. The next value is the minimum slope, which is the minimum slope value obtained from whichever slope calculation method was chosen by the user. This allows for the user to identify the maximum rate of decrease of the Ca2+ fluorescence.

The next two values are the unstimulated average and starting unstimulated average, which are the average within the user chosen time length of the trace right before the stimulation zone and at the beginning of the trace respectfully. These values are useful for identifying a change in the baseline of the trace prior to the stimulation zone. To make measuring this change in inhibitor experiments more streamlined, a difference value is also included which is the unstimulated average subtracted from the starting unstimulated average.

The final value contained in this csv file is the responding percentage, which measures the percentage of traces which went above the user set threshold. This allows for quick determination of the extent to which there was a response in the experiment. Each of these values contained within the “total_data_by_zone” csv file is useful for quantifying a specific aspect of the traces, and together allow for a detailed quantitative analysis of the experiment being analyzed.

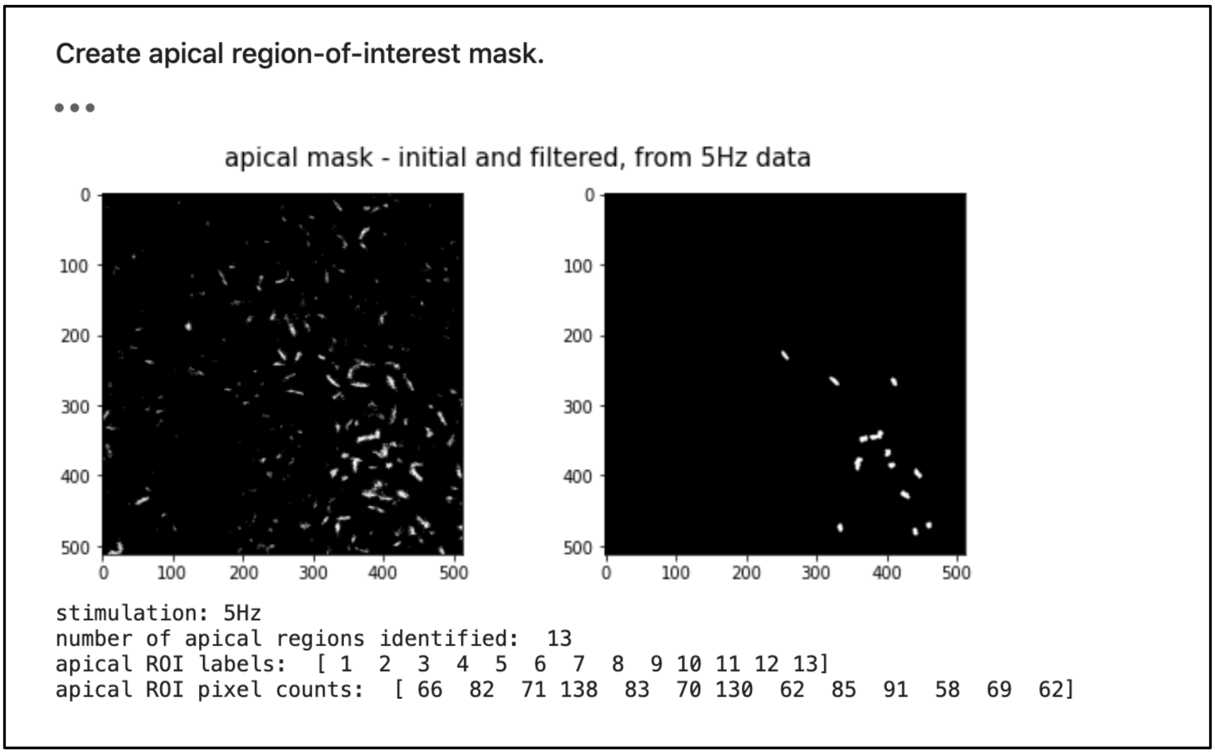
2.4 gCamp Experiment

For quick access to instructions on downloading the required example dataset, click on the following link:
https://github.com/jrugis/cell_tools/blob/master/Experiment_Instructions_GCAMP.txt
Follow the first part of the instructions to download the dataset. The remaining steps in the instructions give a brief outline of what is described in more detail below.

The process of analyzing this experiment type begins with placing the provided GCAMP data files into the image_stacks folder within the cell_tools base directory. Analysis begins with the ROI_Detection notebook. Start by running the "Initialize" code block and then the "Select image stack file(s)" code block, selecting all four of the "Mistgcamp-3" files.

Run the "Enter short descriptive image stack labels" code block and type in the four labels "1Hz", "2Hz", "5Hz", "10Hz". Run the "Set display and calculation parameters" code block, setting the "Image stack for ROI creation" to "5Hz" and the "Calculation method" to "std". The "Create a new time-stamped results directory" code block and the "Get the image stack for ROI creation" code blocks should then be run.

Next the "Unstimulated average (or standard deviation) over time", "Stimulated average (or standard deviation) over time" and "Stimulated minus unstimulated average (or standard deviation) over time" code blocks can be run. The respective plots will appear.



Create apical region-of-interest mask.

•••

```
stimulation: 5Hz
number of apical regions identified:  13
apical ROI labels:  [ 1  2  3  4  5  6  7  8  9 10 11 12 13]
apical ROI pixel counts:  [ 66  82  71 138  83  70 130  62  85  91  58  69  62]
```

Figure 12 - Region-of-interest mask for the Mistgcamp-3 dataset, initial on the left and filtered on the right. Note that this particular dataset was acquired at a pixel resolution that resulted in many initially identified regions with pixel counts that were too low to use for meaningful statistics. The user options shown in Figure 2 were used to filter these out.

Then run the "Create region-of-interest mask" code block. An initial apical mask is created by simple pixel intensity thresholding. Subsequent filtering removes undesired small regions and smooths the remaining regions as shown, for example, in Figure 12. Behind the scenes, in a script code block, the filtering process employs the image processing operations of binary dilation and binary erosion.

Optionally, for referencing the analysis plots that follow, the "FOR REFERENCE: Annotate the ROI mask regions by number" code block can be run.

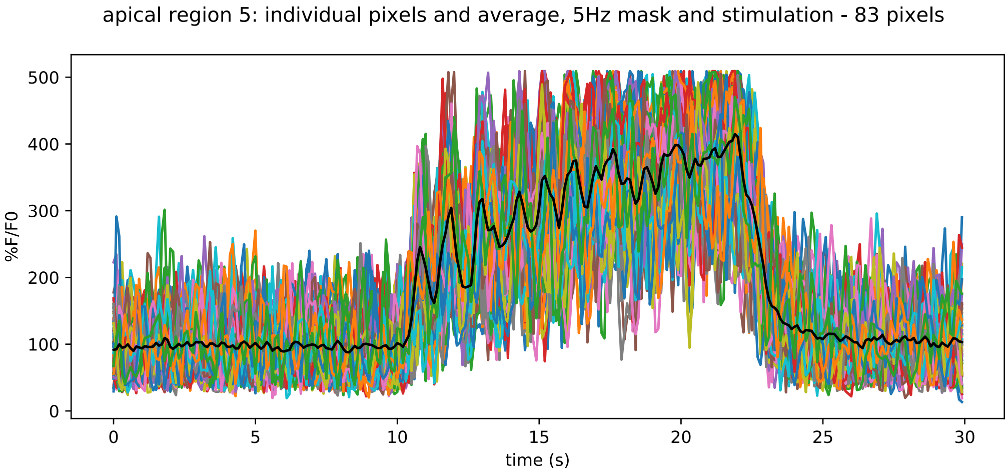


Figure 13 - Plot of ROI individual pixels from one region in the Mistgcamp-3 dataset.

The "DIAGNOSTIC: Plot ROI responses for the same stimulation frequency used to generate the mask" code block can be run. The region masks are used to selectively extract pixels from the original image stack. For example, all of the pixels (in color) as well as the pixel average (in black) over time, are shown for one region in Figure 13.

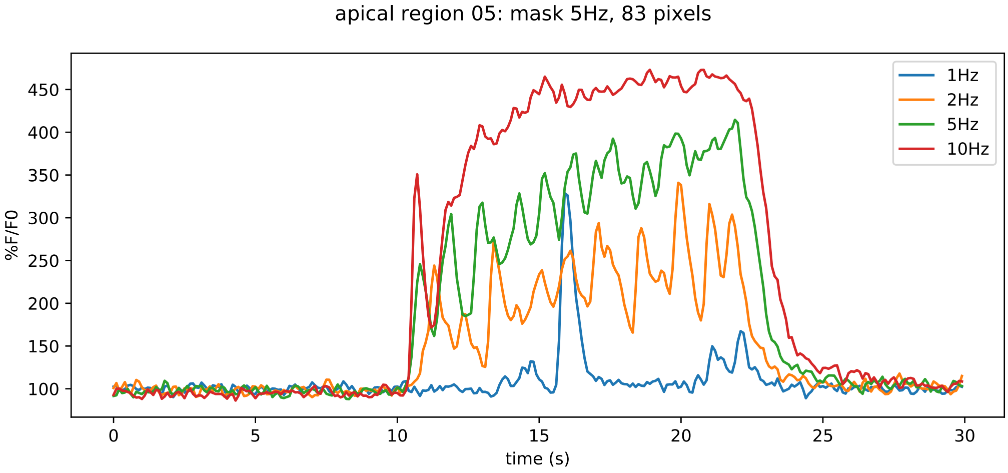


Figure 14 - ROI summary plot with four stimulation frequencies from one region in the Mistgcamp-3 dataset.

Optionally, the "RESULTS: Plot responses for all stimulation frequencies and regions using the mask from above" code block can be run. The notebook displays plots for all regions. For comparison with different stimulus frequencies, these same region masks are applied to the other included image stacks. The final result for one region and the four stimulation frequency regimens is shown in Figure 14.

Next, open the Peak_Detection notebook in JupyterLab and run the "Initialize" code block. Run the "Select a results directory and peak counting options" code block and select the most current GCAMP results directory. Then the "Peak counting over all regions" code block can be run.

The peak processing code block employs several signal processing operations from the Scientific Python[3] package. The code block employs a sequence of signal resampling, low-pass zero phase filtering and high-pass zero-phase filtering to remove the stimulation “bump” as well as high frequency noise from the data. Zero-phase filtering ensures that the filtered signals are not time shifted relative to the original signal[4]. This followed by a generic (but fully featured) peak detector and a mapping back into the original response data.

All the generated plots are saved as pdf files and summary data is saved in csv files. The “peaks_latencies_summary” csv file contains summary data for each region. Peak counts, area under the curve, first peak height, and maximum peak height as well as latency, risetime and maximum slope from stimulation onset to first peak, are all included.

References

[1] J. Rugis, Cell Tools - source code and instructions, https://github.com/jrugis/cell_tools.git, accessed: November 2022.

[2] M. Abdolhoseini, M.G. Kluge, F.R. Walker, S.J. Johnson, Segmentation of Heavily Clustered Nuclei from Histopathological Images, Scientific reports, 9 (2019) 4551-4551.

[3] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, C. SciPy, L.A.N.M. Los Alamos National Lab, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nature methods, 17 (2020) 261-272.

[4] F. Gustafsson, Determining the initial states in forward-backward filtering, IEEE transactions on signal processing, 44 (1996) 988-992.

Figure captions

Figure 1 - Screenshot of interactive panel for selecting image stacks and entering descriptive labels. The labels are error checked to ensure that they are contain only alpha-numeric characters.

Figure 2 - Screenshot of interactive panel for ROI identification parameter settings. “Image stack LUT” refers to the color look up table that is used to color code displayed results. Stimulation start and done frames allow the user to specify the time range in which cell response to stimulation is expected. The frames per second and image data bits numbers are used to calibrate calculated results. “Small region cull size” is used to both remove spurious individual noise pixels as well remove regions whose pixel count is too low to be

statistically meaningful. “Number of dilations” is used to remove small holes in, as well as smooth the edges of, regions of interest. The specified “Calculation method” is used to combine, per-pixel, the stimulated portion of the image stack. “RIO threshold quantile” sets the sensitivity for whether a (combined) pixel is part of a region of interest. “Large ROI % to cull” can be used to optionally eliminate overlapping regions of interest (which likely include multiple cells).

Figure 3 - Notebook peak detection script extract.

Figure 4 - Frequency response of a sample region the Mistgcamp-3 dataset using the FFT method (top) and the filter bank method (bottom).

Figure 5 - Movie screenshot, part way through dynamic display of image sequence and its coupled plot.

Figure 6 - Smoothed 340 nm image stack frame of Trypsin Experiment 4 (left) and CCH Experiment 5 (right)

Figure 7 - Initial segmented masks displayed after running “Segmentation” code block of Trypsin Experiment 4 (Left) and CCH Experiment 5 (Right)

Figure 8 - Smoothed 340 nm image stack frame of CPA Experiment 1

Figure 9 - Initial segmented mask displayed after running the “Segmentation” code block of CPA Experiment 1

Figure 10 - Smoothed 340 nm image stack frame of GSK Experiment 3

Figure 11 - Initial segmented mask displayed after running “Segmentation” code block of GSK Experiment 3

Figure 12 - Region-of-interest mask for the Mistgcamp-3 dataset, initial on the left and filtered on the right. Note that this particular dataset was acquired at a pixel resolution that resulted in many initially identified regions with pixel counts that were too low to use for meaningful statistics. The user options shown in Figure 2 were used to filter these out.

Figure 13 - Plot of ROI individual pixels from one region in the Mistgcamp-3 dataset.

Figure 14 - ROI summary plot with four stimulation frequencies from one region in the Mistgcamp-3 dataset.