Table of Contents

USER MANUAL

Contact: Niraj S. Desai (niraj.desai3@nih.gov)

Last modified: May 25, 2024

INSTALLATION

The *detect* software, though written in MATLAB, can be used by researchers whether or not they have access to MATLAB itself. Users who do have MATLAB installed on their computers ("MATLAB users") should follow the first set of directions to get the *detect* software. Users who do not have MATLAB but do have a Windows computer ("Other users") should follow the second set.

(1) MATLAB users

Download all files and folders from the project's Github site: https://github.com/nsdesai/detect

This download includes not only code specific to this project but code written by others which the project exploits. This code includes:

- CaImAn (CNMF algorithm) from the Flatiron Institute: https://github.com/flatironinstitute/CaImAn.
- PatchWarp (motion correction and image registration) from Hattori and Komiyama: https://github.com/ryhattori/PatchWarp/tree/main
- MLSpike (spike estimation) from Deneux and colleagues: https://github.com/MLspike/spikes
- Brick (toolbox of general functions to speed up MATLAB) from Thomas Deneux: https://github.com/MLspike/brick
- EMD (an implementation of the Earth Mover's Distance) from Ulaş Yilmaz: https://github.com/ulasyilmaz/earth-movers-distance

Put all downloaded files and folders in a single folder somewhere on your hard drive. For example, the folder could be located within the MATLAB folder in Documents, but this is not necessary.

Add the folder to the MATLAB path. On my computer, the folder is located at this address: 'C:\Users\niraj\Documents\MATLAB\detect_software'. To add it to the path, I would enter these lines at the MATLAB command line:

```
folderName = 'C:\Users\niraj\Documents\MATLAB\detect_software';
addpath folderName
savepath
```

You can do the same, remembering of course to change folderName to the location of your folder.

(2) Other users

Windows users who do not have a MATLAB license should go to this OneDrive site maintained by the NIH. The site includes three subfolders: for_redistribution, for_redistribution_files_only, and for_testing. Download these three folders and place them in a folder on your computer (it doesn't matter where or what the folder is named).

Look inside the subfolder for_redistribution for an executable called MyAppInstaller_web.exe. Running this program will install both the project's software and MATLAB Runtime. The second of these will be downloaded from the web, meaning that your computer will need an Internet connection. The process will take ~10 minutes.

Once the installation is completed, you should see an app called *detect* in your computer's list of applications. You should be able to run it like any other Windows app. (Note: Depending upon your computer's security settings, the first time you run detect, you might get a message asking whether you want Windows Firewall to accept actions *detect* wants to make. Select ACCEPT.)

GETTING STARTED

To start, MATLAB users should open MATLAB and enter "detect" at the MATLAB command line, whereas other users should select *detect* from their computer's list of applications.

detect_folder

The very first time you do this, the program will create a folder called *detect_folder* within the Documents folder. (Note well: This folder will be created in the Documents folder on the C drive. In many Windows installations, there is also a cloud storage Documents folder associated with Microsoft OneDrive -- that's not the one we mean.) The folder *detect_folder* contains three subfolders.

Name	Date modified	Type
AVI_files	2/25/2024 11:52 AM	File folder
CNMF_outputs	2/25/2024 11:52 AM	File folder
detection_parameters	2/25/2024 11:52 AM	File folder

AVI_files is the default location for saving the AVI files created by the *detect* software. For the sake of uniformity, regardless of the file type of the raw data, *detect* saves images as AVI files. All other parts of the program, including motion correction, cell detection, and analysis, work on these AVI files. This folder also is the default location in which to save the definition of regions of interest.

CNMF_outputs is the default location for saving the results of all analyses. These analyses are saved in a single file of type MAT, which is the default MATLAB file format. The default filename is *outCNMF.mat*, though this can be renamed to something else. In the last tab of the *detect* GUI, the contents of *outputCNMF.mat* may be saved to an Excel file for use by other programs.

detection_parameters is the default location for the parameters used in cell detection by the CNMF algorithm.

Although it is not necessary to use the subfolders of *detect_folder*, as the files could in principle be saved anywhere, doing so minimizes the potential for confusion. All the AVI files in *AVI_files* are ready for analysis by the program, every file in *CNMF_outputs* contains the variables of the CNMF analysis, etc. Best practice is to use *detect_folder* while using the *detect* GUI, and then to transfer the resulting files, perhaps renamed in more instructive ways, to other folders for long-term storage.

Graphical user interface (GUI)

Start the program by entering "detect" on the MATLAB command line or selecting *detect* from the list of Windows apps. A graphical user interface (GUI) that looks like this will open up:
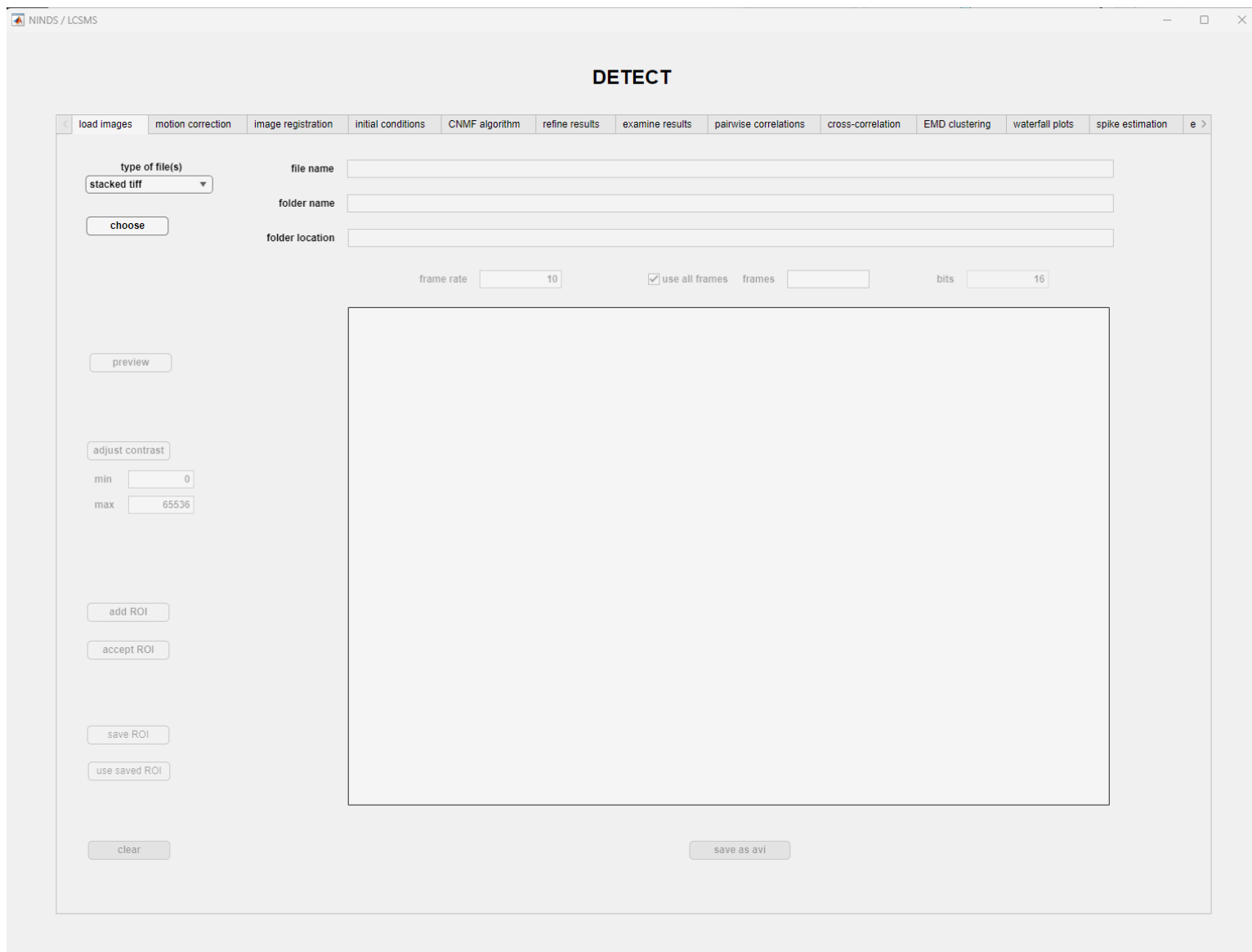
Figure 1. Graphical user interface. The GUI contains multiple tabbed panels. The first, the starting point, allows one to load images, adjust contrast, define regions of interest, and save processed images.

Note the tabs at the top. They allow you to navigate -- tab by tab, left to right -- between the program's functions.

Tabs

1. load images: Load raw images (tiff, stacked tiff, avi, or mp4). Preview the images and adjust contrast. Add regions of interest (ROIs). Save preprocessed images as AVI.
2. motion correction: Run the PatchWarp algorithm for motion correction.
3. image registration: Register images longitudinally using the PatchWarp algorithm.
4. initial conditions: Choose the parameters of the CNMF algorithm for cell detection.

5. CNMF algorithm: Run the CNMF algorithm.
6. refine results: Quality control on the cells detected by the CNMF algorithm. Includes both manual and automatic options.
7. examine results: Explore the fluorescence traces ($\Delta\frac{F}{F}$) of detected cells.
8. pairwise correlations: Calculate Pearson correlation between pairs of cells. Show how they cluster using a dendrogram and/or a correlation matrix.
9. cross correlation: Calculate and plot cross correlations between pairs of cells.
10. EMD clustering: Run a Fourier transform on fluorescence traces and calculate the Earth Mover's Distance (EMD) between pairs of cells. Display as a dendrogram and/or distance matrix.
11. waterfall plots: Display waterfall plots of fluorescence traces, grouped by pairwise correlation
12. export: Export analyses for display or further analysis in other programs.

LOAD IMAGES

The first tabbed panel of the GUI allows users to load raw images, adjust contrast, and define regions of interest.

Raw images must be in one of four formats:

- individual TIFF files saved in a single folder
- stacked TIFF
- AVI
- MP4

To load raw images, press the **choose** button, which will open a Windows dialog allowing you to either (1) select the folder containing the individual TIFF files or (2) select the file containing the stacked TIFF, AVI, or MP4 video of the raw images.

If necessary, adjust the **frame rate** field to reflect the acquired number of frames per second. This will only be necessary when working with individual TIFF files; the other formats have the frames per second included in their metadata.

Choose the frames to analyze using the **frames** field. (Unclick the **use all frames** checkbox to make the field editable.) The frames can be specified as a single range (like this 1:600 for the first 600 frames) or as a broken up vector (like this 20:422,488:512,552:592, to specify three noncontiguous sets of frames). MATLAB's usual rules for array indexing apply here: https://www.mathworks.com/help/matlab/math/array-indexing.html.

Press the **preview** button. This will display the mean image of the specified frames in the adjoining axes.

If the image does not look clear enough for you to identify structures of interest (e.g., the anatomical boundaries used to identify targeted brain areas), you can access the basic MATLAB tools for adjusting image contrast by pressing the **adjust contrast button** .

To add regions of interest, press the **add ROI** button. This allows you to draw an ROI using the mouse as described in the *drawpolygon* section of the MATLAB documentation: https://www.mathworks.com/help/images/ref/drawpolygon.html. (See also **Supplementary Video 1**, for an illustration.) You may add multiple, non-contiguous ROIs in this way, such as in the following example.
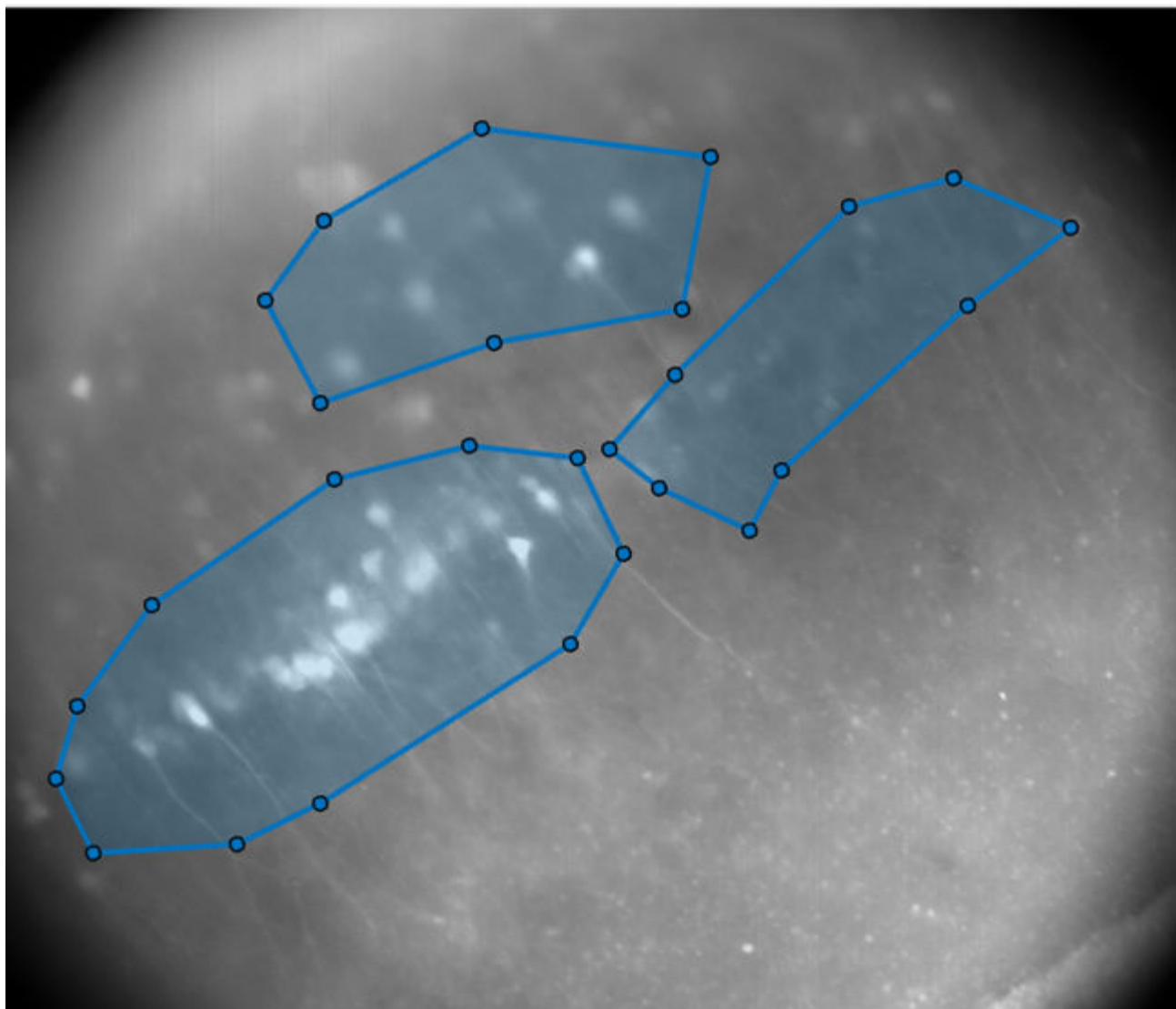

Figure 2. Defining regions of interest. Use MATLAB's builtin ROI tools to define one or more ROIs by clicking and dragging the mouse.

Using the mouse, you can adjust the shape and position of each ROI. If, for whatever reason, you are unhappy with the results, you can always press the **clear** button to reset everything and start again (press the **preview** button, etc.)

Once you are satisfied with the ROIs, press **accept ROI** to accept the defined ROis and crop out all the other parts of the image.

You can save your ROI definitions or call up previously saved ROI definitions using the **save ROI** and **use saved ROI** buttons, respectively.

Once satisfied, press the **save as avi** button. The default save location is within the *AVI_files* subfolder previously defined.
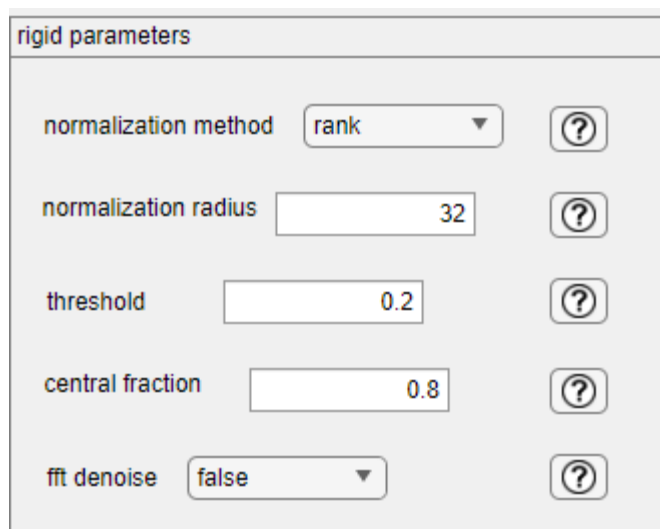
MOTION CORRECTION

All files that require motion correction (AVI files created by the *load images* subpanel) should be saved together in a single folder with no other files present. For example, creating a subfolder within the *AVI_files folder* is a good choice. They wlll be corrected together. Press the button **specify avi folder** to specify the folder name containing the AVI files.

Two separate forms of motion correction are availabe, as described in the main text: (1) rigid translation and (2) warp correction.

Rigid translation

The parameters of the rigid translation are set using the fields of the *rigid parameters* subpanel.

Figure 3. Rigid parameters subpanel.

The parameters are:

1. normalization method: Normalization of images is performed to make full use of dynamic range and increase contrast. The PatchWarp algorithm offers two choices: *rank* and *local*. In rank normalization, pixels are replaced by their fractional rank (rank divided by total number of pixels) within the image. In local normalization, the intensity of each pixel is normalized by dividing it by the intensity of neighboring pixels. (The parameter normalization radius determines the size of the neighborhood used.) In general, unless images are expansive or have obvious non-uniformities, rank normalization is preferred. But if motion correction appears unsatisfactory, the first thing to do is to try local normalization.
2. normalization radius: When the local normalization method is selected, images are normalized by dividing each pixel value by the mean of pixel values in the local neighborhood. The parameter normalization radius determines the size of the neighborhood. (When the rank normalization method is chosen, this parameter does nothing.) The radius should be expressed in pixels and correspond to the approximate size of expected non-uniformity.
3. threshold: The parameter threshold determines which frames are used to calculate template images (representative images against which other images are compared). The Pearson correlation coefficient is calculated between the mean across all frames and each individual frame. Frames with coefficients lower than the specified threshold will be ignored as outliers.
4. central faction: The parameter central fraction limits how much the edges of the template image affect image registration. For example, when the central fraction is set to 0.8, then 10% of pixels (those closest to the edges) will be ignored. This parameter does not affect results very much. The default choice of 0.8 is fine.
5. fft noise: If this parameter is checked true, then strong periodic patterns (e.g., ripple patterns of PMT noise) will be removed from template images by Fourier transform. In general this is not necessary, but can be useful if periodic rippling is evident in images.

Warp correction

Warp correction is employed when there are local distortions in subfields of the field of view. That is, when it is not sufficient to move the whole field in one direction or another, and individual subfields must be adjusted individually. This is a rare circumstance except when the field of view is very large. But if it is very large, select Yes fo the *warp correction?* pulldown menu and set the parameters accordingly.

Figure 4. Warp correction subpanel.

The warp parameters are:

1. affine radius: The parameter affine radius is the radius of the circular filter used for local intensity normalization before application of the gradient-based algorithm. The radius is in pixels and should reflect the observed non-uniformity.
2. moving average size: This parameter is the window size (number of frames) for smoothing downsampled images before estimation of affine transformation matrices. For good signals, this should just be 1. Increase this number if signal-to-noise is low.
3. block size: Row and column numbers for splitting an image's field of view into into blocks for affine transformation. Each image is split into [warp_blocksize]*[warp_blocksize] subfields.For moderate distortion, one should use a small number (<= 4). For severe distortion, a large number may be used. Note that processing time goes like n^2 and so larger block sizes will increase the required time by a lot.

Running motion correction

After setting the parameters, running the correction simply entails pressing the **run motion correction** button. You can view the original and corrected videos simultaneously by selecting them from the *original files* and *corrected files* listboxes respectively and then pressing the **play videos** button (Figure 5).

*Figure 5. After motion correction, you can select individual original and corrected files and play them simultaneously be pressing the **play videos** button. If they are acceptable, press the **accept correct files** button.*

If the correction is acceptable, press the **accept correct files** button. Use the corrected AVI in all other analyses.

IMAGE REGISTRATION

Image registration is tightly linked to motion correction. First run the motion correction algorithm above both on the AVI files of the original recording session (reference) and on those of the subsequent recording session (target).

Press the **choose reference folder** button and select the *downsampled* folder created for the original session (motion_correction > corrected > pre_warp > downsampled). Then press the **choose target folder** button and select the *downsampled* folder created for the subsequent session. created by the motion correction algorithm. Finally press the **calculate registration** button. This creates a registration file (named *regFile.mat*) at the location specified in the field.

To use this registration file on the target AVI files, select it using **load registration parameters**, specify the target AVI file using **specify file to be registered**, and press the **run registration algorithm** button. You may need to adjust the **frame rate** field beforehand.

INITIAL CONDITIONS

In this tabbed panel, you can choose the parameters used by the CNMF algorithm to detect active cells. (The whole process -- defining parameters, running the algorithm, doing quality control on the results -- is illustrated in **Supplementary Video 2**.)

Many parameters are listed, but practically speaking, three are most important:

1. estimated number: Choose the maximum number of possible cells in the field of view. If you think there are 30 cells, list 40. If you think there are 10, list 20. The algorithm uses your estimate as an *upper limit* -- so always overshoot.
2. typical diameter (pixels):   Estimate the approximate diameter of the typical neuron (soma size) in the field of view. This number need only be correct to within ~50%. Just set the scale so that the algorithm knows it should be searching for a neuron of ~30 pixels and not ~300 pixels or ~3 pixels.
3. merging threshold: This threshold (0-1) determines how correlated pixels have to be to be considered part of a single active neuron. For microendoscopic imaging, we have found that a large number (~0..95) is appropriate, but for brain slices a much small number (~0.25) seems more correct. Users will have to experiment a bit to get the correct number for their own preparations.

The other CNMF parameters are much less crucial and mainly affect how much time the CNMF algorithm takes to run. Pressing the question mark button next to each field brings up information on each parameter. But for the most part, leaving them at the defaul values is a reasonable choice. The three parameters noted above (estimated numer, typical diameter, and merging threshold) matter a great deal and require some consideration; the other parameters do not.

Pressing the **save parameters** button saves your choices. The default folder in which parameters are saved is *detection_parameters* and the default name is detectionParameters.mat.

CNMF ALGORITHM

One can choose the AVI file to analyze by pressing the **choose avi file** button and the CNMF parameters to employ by choosing the **choose parameters files** button.The names of the selected AVI and parameters files will appear in the designated fields.

(If you would like to combine multiple AVI files -- perhaps different recording sessions from the same brain slice or the same animal -- and analyze the combination together, simply choose multiple files after pressing the **choose avi file** button. One can choose multiple files from the dialog in the standard Windows way, namely hold the CTRL key down while clicking on the choices. The program will create a new AVI file with the word COMBINED appended to the title and save it in the *AVI_files* subfolder. Its name will appear in the **avi file** text field and it will be used in the CNMF analysis.)

Press the **choose parameters file** button to select the detection parameters (e.g., the detectionParameters.mat file created in the previous tab).

Then simply press the **run** button to run the algorithm.This can take a long time (5-30 min) depending on the size and number of image frames and the memory, number of processors, and speed of the computer.

Once the analysis has completed, you can save the results by pressing the **save** button.

The **reset** button zeroes everything so that, if for whatever reason you want to run the algorithm again, you can do so.

The clock icon keeps track of how long the process is taking. Also, the labels of different parts of the algorithm change from black to green to blue as each part is started, run, and completed.

Percent complete

run time (sec)　　　0.0

choose default CNMF file

apply to other AVI files

Figure 5. The clock indicates how much time is remaining in running the CNMF algorithm. The two buttons at the bottom may be used to select and apply previously saved CNMF cell detection analysis to other AVI files from the same preparation.

Once the analysis is complete, you can press the **save** button to save the results. The default save location is the aforementioned *CNMF_outputs* folder.

Reusing previously saved CNMF results

If you have run the CNMF cell detection analysis on data from one recording session, you may wish to use the detected cell boundaries on other recording sessions from the same brain slice or animal. This makes it simple to compare results longitudinally. To do this (Figure 5), (1) select the CNMF outputs from the original session by pressing the **choose default CNMF file** button, and (2) press the **apply to other AVI files** button to select the AVI files from the subsequent sessions. After the analysis has been applied, you can choose where to save the new analyses using a dialog box that will pop up.

REFINE RESULTS

The next panel allows you to inspect the output of the CNMF algorithm and perform quality control. After the algorithm completes, a mean image of the ROI appears in the central window, with the detected cell boundaries superimposed on top of it. The center of mass of each neuron is marked by a green dot.
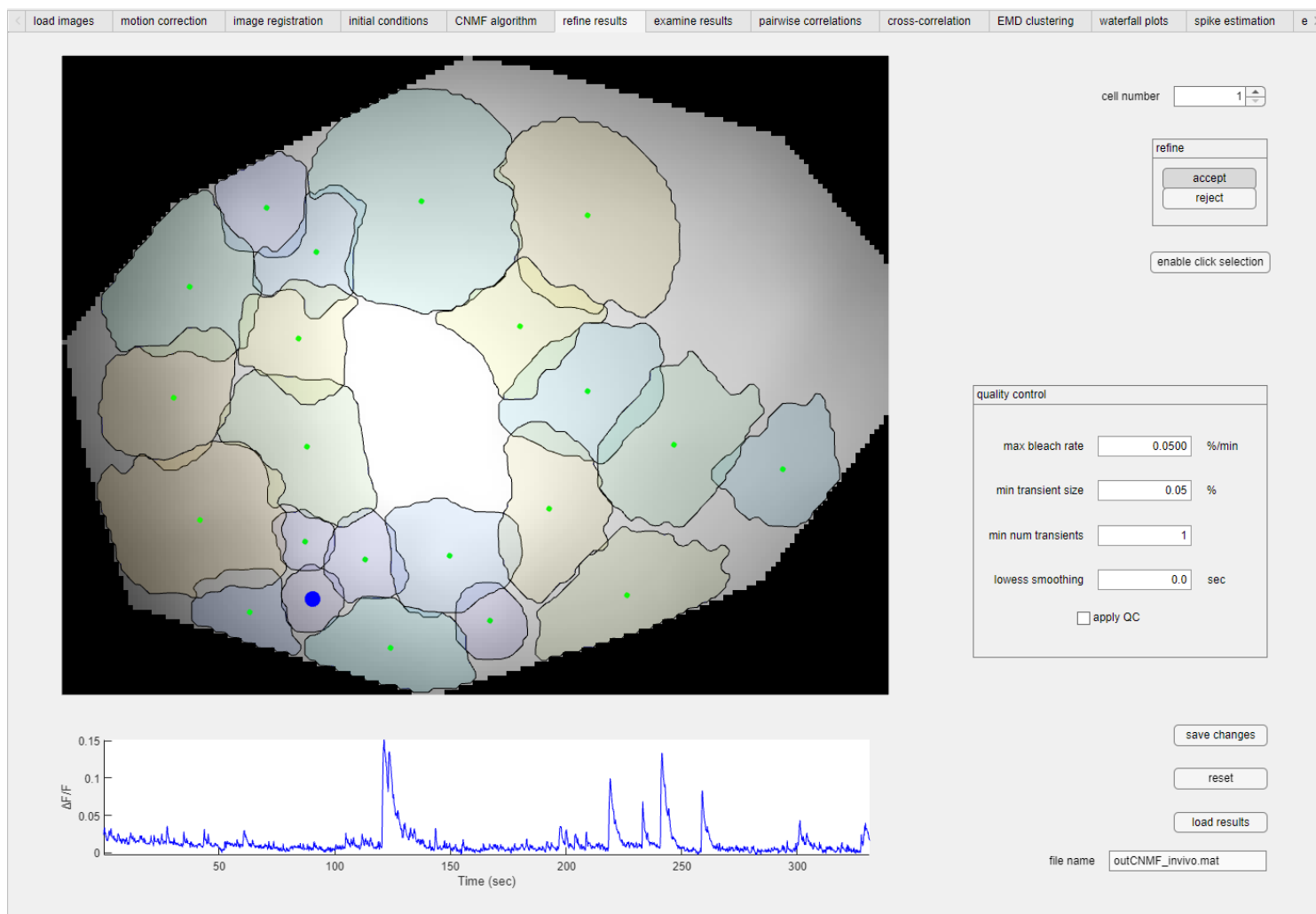
Figure 6. The Refine Results panel allows users to manually accept or reject individual detected neurons and to implement automated quality control. A mean image of the ROI is seen in the central window with outlines of detected neurons superimposed. A green dot is placed at the center of mass of each cell boundary. A selected neuron, chosen using the controls at upper right, is marked by a big blue dot and its fluorescence trace is plotted in the bottom graph.

One can navigate between neurons by using the controls at upper right. There are two ways of selecting an individual neuron. (1) Use the the **cell number** spinner to scroll up and down through the cell list. (2) Press the **enable click selection** button to put it in the "on" state (marked by darker shading). This will transform the cursor into a crosshair when it hovers over the image. Clicking on the green dot over a given neuron will switch focus to that neuron.(Disabling click selection by pressing the button again returns the cursor to normal function.)

Once a neuron is selected, its the position in the image will be marked by a big blue dot and the blue trace of the bottom graph shows its fluorescence trace over time. Traces that show problems like excessive noise, baseline drift, or "odd-looking" calcium transients may be rejected by pressing the **reject** toggle button. (Likewise switching a previously rejected neuron to accept status can be accomplished by pressing the **accept** toggle button.) If a neuron is rejected, the color of its dot in the image changes from green to red.

The *quality control* panel in the middle implements several forms of automated quality control. (1) It excludes neurons with fluorescence traces that drift by more than the max bleach rate (% change in $\Delta\frac{F}{F}$ per minute).

(2) It sets a minimum calcium transient size. Fluorescence traces without calcium traces at least this large are excluded. (3) It sets a minimum number of caclium transients. Fluorescence traces with fewer calcium transients are excluded from further analysis. (4) It smooths the fluorescence traces using a lowess method. Clicking the **apply QC** clickbox to active implements these quality control measures; unclicking it to inactive removes previously applied quality control measures.

Once you are satisfied with your refinements, press the **save changes** button to save them. The default folder is again *CNMF_outputs* and the default filename is outCNMF.mat.

You can load previously-saved results to work on them again by pressing the **load results** button. If you can make a mistake, you can return to baseline by pressing the **reset** button.

Combining spatial and temporal information

The CNMF algorithm uses both spatial and temporal information to identify the boundaries of neurons. The distinction between spatial and temporal is particularly evident when one considers preparations in which neurons are densely packed and only intermittently active. In those cases, an average over all images results in a static image that seems to be an undifferentiated blur. An example is microendoscopic imaging from the basolaterela amygdala (see picture at left of Suppl. Fig. 6 above) in which a static spatial image does not show any obvious cell boundaries. However, if one were to watch a movie of the same imaging session long enough (and if contrast were adjusted just so), one would see that there are, in fact, active cells embedded within the blur. When deciding whether the CNMF algorithm has worked, one has to consider both space and time. In particular, one must rely on the fluorescence time series (bottom graph of Suppl. Fig. 6).

EXAMINE RESULTS

The *examine results* panel allows users to plot the fluorescence traces of multiple neurons simultaneously. The idea is simply to get a sense of what local networks of neurons look like.
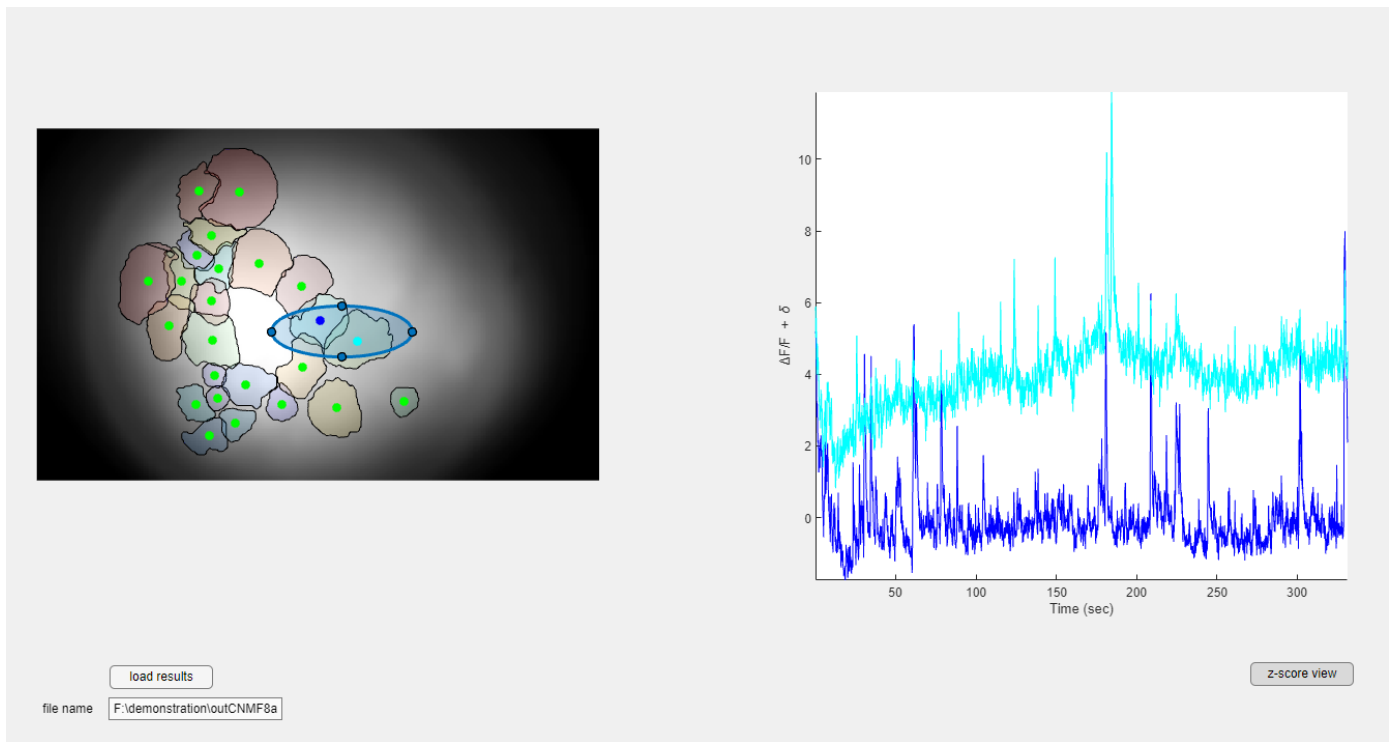
Figure 7. An image from the examine results *subpanel. The blue elipse at left indicates the selected local network of neurons; the ellipse's size and location can be manipulated using a mouse. The neurons' fluorescence traces will be plotted in the graph at right.*

As shown in Figure 5, a blue ellipse on the image can be moved and manipulated to select different groups of neurons. Their fluorescence traces will appear at right, color-coded and (if the **z-score view** button is depressed) z-score normalized.

PAIRWISE CORRELATIONS

The *pairwise correlations* panel allows users to calculate the Pearson coefficients between every pair of detected neurons by pressing the **run pairwise** button.

The results are plotted in two ways: (1) dendrogram and (2) correlation matrix.

The slider below the dendrogram (marked "cutoff") allows users to set the cutoff for the color clustering of the dendrogram. Smaller numbers indicate finer-grained clustering; larger numbers indicate coarser-grained clustering.

The correlation matrix displays the correlation coefficient between pairs of neurons. The color scale and the size of each marker indicate the size of the correlation (hotter colors and larger markers signify larger correlations). These are organized according to the clustering specified by the cutoff slider.

17

CROSS CORRELATION

The *cross correlation* panel allows users to calculate and examine the cross correlations between pairs of fluorescence traces, by pressing the **calculate** button. They select one or more neurons from the *neuron 1* listbox and one or more neurons from the *neuron 2* listbox (using shift and CTRL in the typical Windows way to select more than one item). All pairs between neuron 1 and neuron 2 are then identified, the fluorescence traces are displayed in the middle graph, and the cross correlations are displayed in the graph at right.
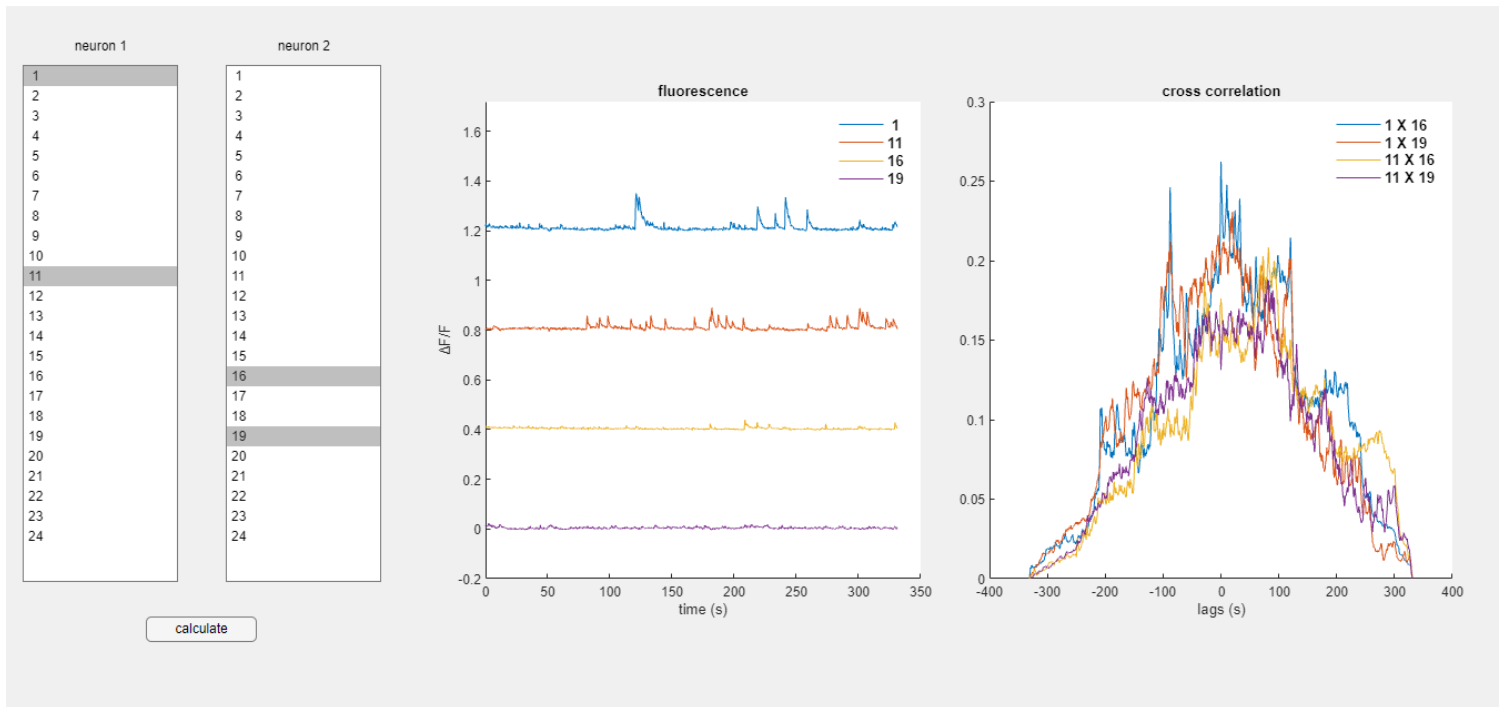
Figure 8. Cross correlations between neurons. The listboxes at left allow users to specify neuron numbers. The raw traces of these neurons are displayed in the middle. The cross correlations of every pair of neurons are displayed at right.

EMD CLUSTERING

In this program, Earth Mover's Distance (EMD) is applied to the discrete Fourier transforms of the fluorescence traces $\Delta \frac{F}{F}$ traces. The idea is provide a metric that characterizes how close or far the oscillatory activities of different neurons are from each other.

As in the other panels, pressing the **load file** button loads the outCNMF.mat file and displays the estimated cell boundaries in the image at left. Pressing the **run EMD** button runs the EMD algorithm and displays the results in a dendrogram (middle) and distance matrix (right).
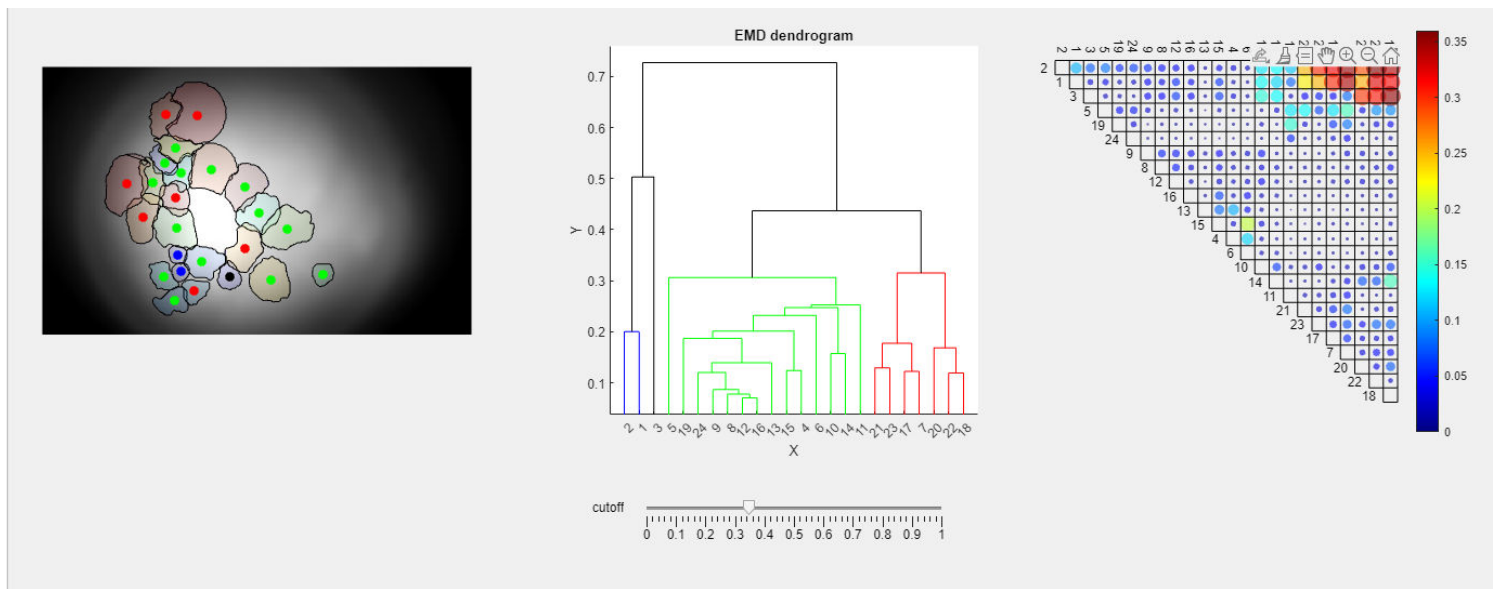
18

Figure 9. EMD metrics. The colors of the dendrogram (middle) indicate clustering of the closest distances. The matrix at right shows this in a different away, with the symbol sizes and colors indicating small (blue-ish, small circles) and large (red-ish, large circles) distances.

WATERFALL PLOTS

Waterfall plots are a useful way to visual time series data across related neurons. One dimension is time, one is the intensity being plotted, and one is the neuron number.

In the case of pairwise correlations, the result may look like this:
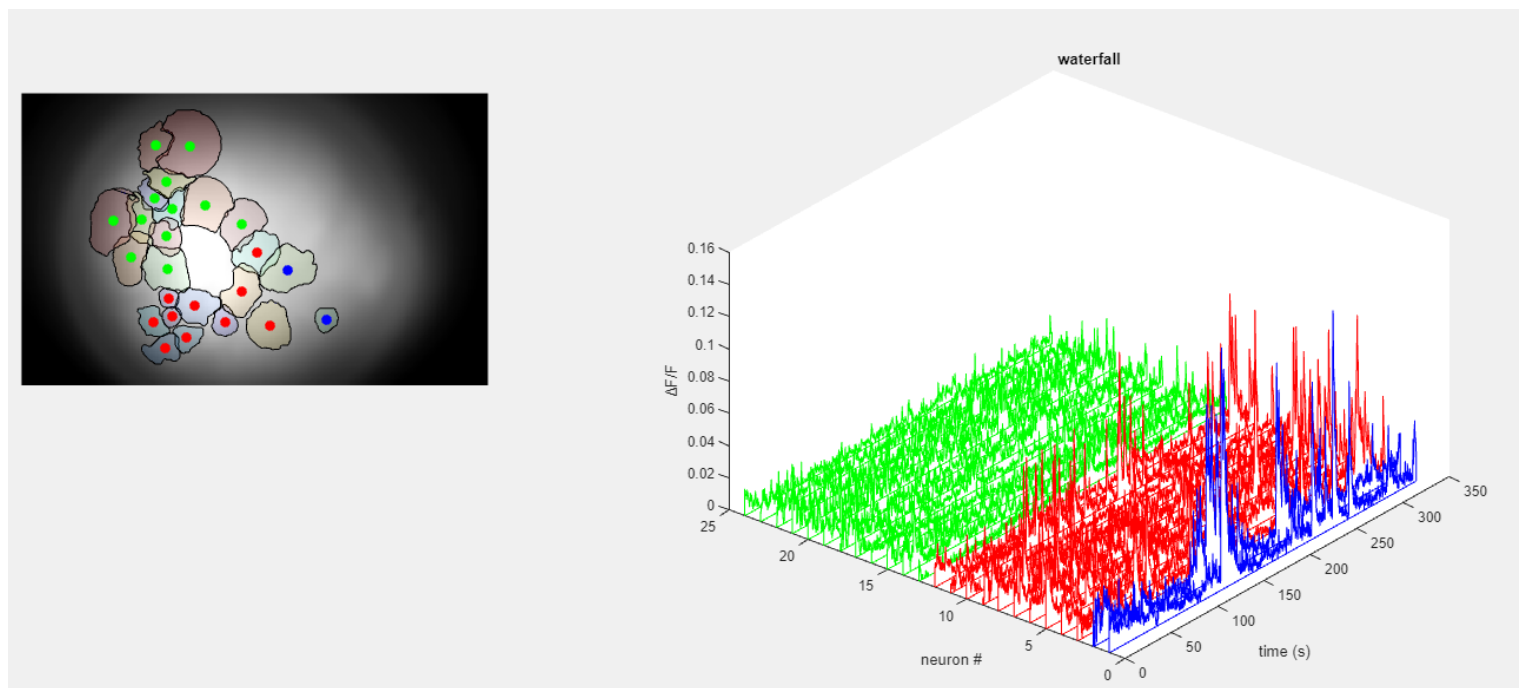
Figure 10. Waterfall plots.

Here, the green, red, and blue traces (neurons) are clustered with each other according to their pairwise Pearson correlation.

Users can use the **classification type** dropdown menu to choose other grouping types.

SPIKE ESTIMATION

This uses the MLSpike algorithm of Thomas Deneux and colleagues (Deneux et al., 2016) to estimate spike times from a fluorescence trace $\Delta\frac{F}{F}$. This estimation depends on the calcium flourophore being used (e.g., GCaMP6s, GCaMP6f, OGB). The algorithm uses the measured nonlinearities of these fluorphores to interpret calcium dynamics and estimate spike times.

Important: For spike estimation from recorded calcium fluorescence, a high frame rate (30 Hz or faster) is necessary.

The parameters of the MLSpike algorithm may be specified in this subpanel:
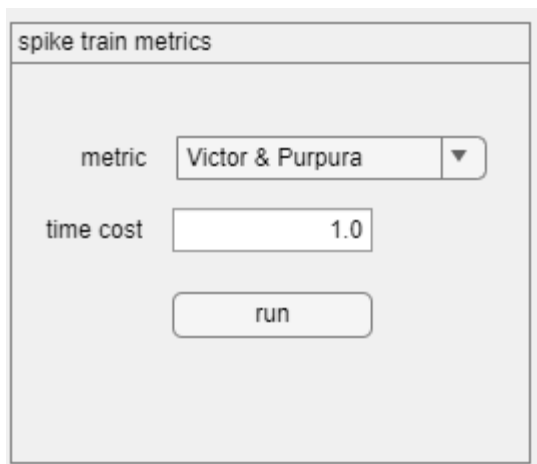
Figure 11. Parameters of the MLSpike algorithm.

Here, *Amplitude A* is the typical amplitude (in percentage $\Delta\frac{F}{F}$) of a calcium transient produced by a single action potential, *decay time tau* is the characteristic time for decay of a caclium signal (in seconds), *noise sigma* is the standard deviation of the baseline noise (in percentage $\Delta\frac{F}{F}$) , *baseline min* and *baseline max* are the typical minimum and maximum values of the background fluorescence (in percentage $\Delta\frac{F}{F}$).

Pressing the **autoML** button instructs the progam to run MLSpike's builtin estimation of the MLSpike parameters. Ideally, these will be close to the parameters one would select by eye.It **is important to be aware that the autocalibration algorithm should only be used with high-quality data, typically where single-spike transients are clearly identifiable by eye.** In other cases, setting parameters manually is preferred.

The **nonlinearity** pulldown menu allows users to select the calcium fluorophore in use.

Pressing the **run ML Spike** button runs the MLSpike algorithm.

The subpanel at right contains the functions for running spike metrics.

Figure 12. Parameters of the spike distance metrics.

The available metics are *Victor & Purpura*, *Van Rossum*, *SPIKE*, and I*SI*. Each can be parameterized by a time cost (in seconds).

Pressing the **run** button calculates the metrics and plots them in the *spike train distances* window.
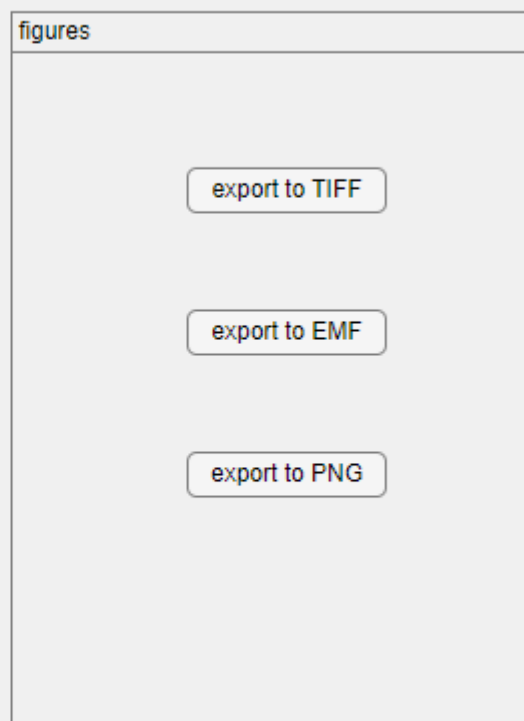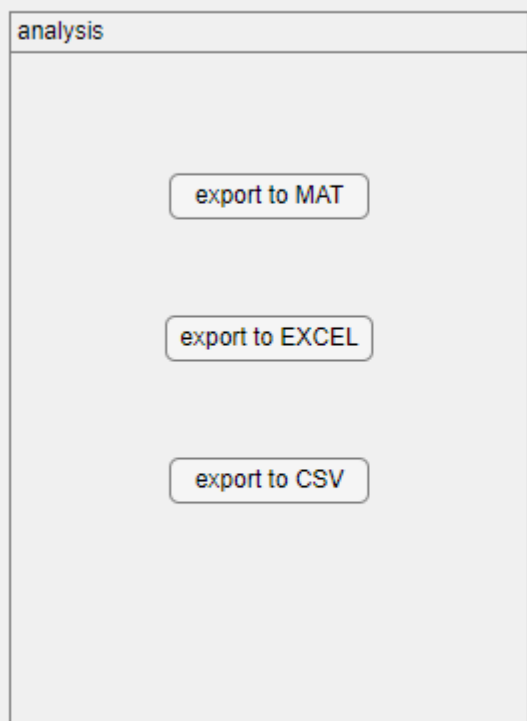
EXPORT

Figure 13. Export panel.

Analysis results may be exported from the program using the *analysis* subpanel on the *export* panel. Press one of the buttons at left to save results into MAT, Excel, or CSV formats.

Individial figures may be saved by hovering the cursor above the figure so that the interactive tools appear (see Figure 10). Then select the icon on the left.
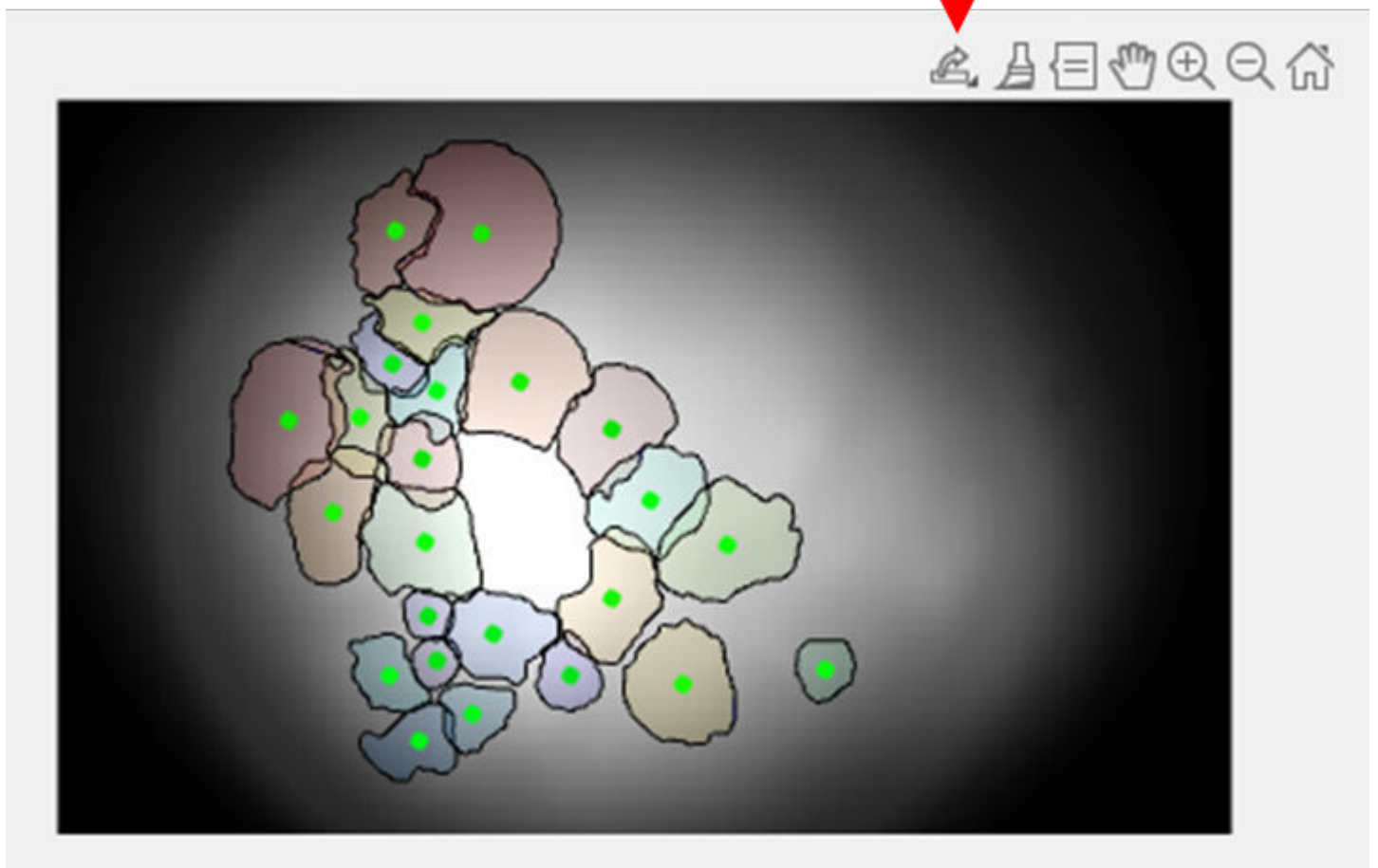
Figure 14. Hovering the cursor over the figure make the tools at upper right visible. The one on the far left allows you to save the image (as PNG, TIFF, etc.)

Alternatively, press one of the buttons on the far right of the *export* panel to save all figures in the TIFF, EMF, or PNG formats.