

Journal of Visualized Experiments

Automated Detection and Analysis of Exocytosis

--Manuscript Draft--

Article Type:	Invited Methods Collection - Author Produced Video
Manuscript Number:	JoVE62400R3
Full Title:	Automated Detection and Analysis of Exocytosis
Corresponding Author:	S. L. Gupton University of North Carolina at Chapel Hill CHAPEL HILL, NC UNITED STATES
Corresponding Author's Institution:	University of North Carolina at Chapel Hill
Corresponding Author E-Mail:	sgupton@email.unc.edu
Order of Authors:	Fabio Urbina S. L. Gupton
Additional Information:	
Question	Response
Please indicate whether this article will be Standard Access or Open Access.	Standard Access (US\$1200)
Please specify the section of the submitted manuscript.	Biology
Please confirm that you have read and agree to the terms and conditions of the author license agreement that applies below:	I agree to the Author License Agreement
Please provide any comments to the journal here.	the standalone app will also be added to dropbox as it does something strange when dropped into the files here
Please indicate whether this article will be Standard Access or Open Access.	Standard Access (\$1400)
Please confirm that you have read and agree to the terms and conditions of the video release that applies below:	I agree to the Video Release

TITLE:

Automated Detection and Analysis of Exocytosis

AUTHORS AND AFFILIATIONS:

Fabio Urbina, Stephanie L. Gupton

UNC Neuroscience Center, Department of Cell Biology and Physiology, University of North Carolina at Chapel Hill

Email Addresses of Co-Authors:

Fabio Urbina (fabiou@email.unc.edu)

Stephanie L. Gupton (sgupton@unc.edu)

Email Address of Corresponding Author:

Stephanie L. Gupton (sgupton@unc.edu)

SUMMARY:

We developed automated computer vision software to detect exocytic events marked by pH-sensitive fluorescent probes. Here, we demonstrate the use of a graphical user interface and RStudio to detect fusion events, analyze and display spatiotemporal parameters of fusion, and classify events into distinct fusion modes.

ABSTRACT:

Timelapse TIRF microscopy of pH-sensitive GFP (pHluorin) attached to vesicle SNARE proteins is an effective method to visualize single vesicle exocytic events in cell culture. To perform an unbiased, efficient identification and analysis of such events, a computer-vision based approach was developed and implemented in Matlab. The analysis pipeline consists of a cell segmentation and exocytic-event identification algorithm. The computer-vision approach includes tools for investigating multiple parameters of single events, including the half-life of fluorescence decay and peak $\Delta F/F$, as well as whole-cell analysis of the frequency of exocytosis. These and other parameters of fusion are used in a classification approach to distinguish distinct fusion modes. Here a newly built GUI performs the analysis pipeline from start to finish. Further adaptation of Ripley's K function in R Studio is used to distinguish between clustered, dispersed, or random occurrence of fusion events in both space and time.

INTRODUCTION:

VAMP-pHluorin constructs or transferrin receptor (TfR)-pHuji constructs are excellent markers of exocytic events, as these pH-sensitive fluorophores are quenched within the acid vesicle lumen and fluoresce immediately upon fusion pore opening between the vesicle and plasma membrane¹. Following fusion pore opening, fluorescence decays exponentially, with some heterogeneity that reveals information about the fusion event. Here, a graphical user interface (GUI) application is described that automatically detects and analyzes exocytic events. This application allows the user to automatically detect exocytic events revealed by pH sensitive markers² and generate features from each event that can be used for classification purposes³

(**Figure 1A**). In addition, analysis of exocytic event clustering using Ripley's K function is described.

The automated classification of exocytic events into different exocytic modes was recently reported³. Two modes of exocytosis, full-vesicle fusion (FVF) and kiss-and-run fusion (KNR) exocytosis have previously been described⁴⁻⁷. During FVF, the fusion pore dilates, and the vesicle is incorporated into the plasma membrane. During KNR, the fusion pore transiently opens and then reseals^{4,5,8-10}. Four modes of exocytosis were identified in developing neurons, two related to FVF and two related to KNR. This work demonstrates that both FVF and KNR can be further subdivided into fusion events that proceed immediately to fluorescence decay (FVFi and KNRi) after fusion pore opening or exocytic events that exhibit a delay after fusion pore opening before fluorescence decay begins (FVFd and KNRd) (**Figure 1B**). The classifier identifies the mode of exocytosis for each fusion event. Here this analysis has been incorporated into a GUI that can be installed in MATLAB in Windows and Mac based operating systems. All analysis files can be found at <https://github.com/GuptonLab>.

PROTOCOL:

1. Choose datasets and directory

1.1. To select datasets for analysis, click the **Find Datasets** button (**Figure 2A, red box 1**) to navigate to the folder where data is deposited (e.g., RawData folder). Datafiles will automatically populate the Data Files as a list. There can be more than one dataset in the folder.

1.2. Click the **Choose Directory** button and select the directory (e.g., Test) where analyzed files will be deposited (**Figure 2A, red box 2**). A set of folders and finished analysis files as well as intermediate temporary images will be created in this directory when the Analysis button is pushed. Errors will be produced if a directory is not chosen.

2. Set the pixel size and framerate

2.1. Fill in the appropriate frame rate and pixel size of the images in the appropriate "Framerate" or "pixel size" box (**Figure 2A, green box**). If no values are provided (they are set to the default "0"), then the program will search the metadata associated with the file for the framerate and pixel size. If these values cannot be found, the program will default to per-pixel for measurement and per-frame for time points.

NOTE: In the example provided, the framerate is 100, and the pixel size is 0.08.

3. Choose or make masks

3.1. Use the **Mask Maker** button to automatically create cell masks for the data in the file data setlist (**Figure 2A, blue box**). Upon using the **Mask Maker** button, a new folder in the chosen directory will be created called MaskFiles. The "Run indicator" will turn yellow while running and return to green when completed.

NOTE: A mask for each file in the Data Files list will be created from the first 10 frames of the image file (or from all of the frames if less than 10 are in the image file) and deposited in the MaskFiles folder using the proper naming scheme (described below).

3.2. Ensure that the mask files automatically populate the Mask Files list. The user may proceed directly to the analysis.

NOTE: Always visually check mask files and confirm they capture the entire region of interest. The first frame of the data file and the mask file are displayed on the UI when selected. (**Figure 2B**). The Mask Maker may produce errors in the case of low signal-to-noise, so validating that mask files are appropriate is critical for quality control.

3.3. As an alternative to using Mask Maker, if the signal to noise of images is insufficient, create masks manually in ImageJ.

3.3.1. First, open the raw image file in ImageJ (**Figure 3A**).

3.3.2. Click the **Polygon Selection** button and click to draw a mask around the cell. Once finished, double-click on the last point to complete the polygon.

3.3.3. Once finished, navigate to **Edit | Selection | Create Mask (Figure 3B)**. A new inverted mask will be created based on the polygon drawing. Save masks in a designated MaskFiles folder in the chosen directory. The mask file naming scheme must match the corresponding individual data files followed by “_mask_file”. For example, if a data file is named “VAMP2_488_WT_1.tif”, the corresponding mask file must be named “VAMP2_488_WT_1_mask_file.tif”.

3.3.4. Use the **Find Maskfiles** button to navigate to the chosen folder of deposited custom mask files. The masks will populate the Mask Files as a list.

NOTE: It is important to have a mask for every data file before running the analysis.

4. Analysis and feature extraction

4.1. After the directory is chosen and the Data Files list and Mask files list are populated, click the **Analysis** button (**Figure 4A**). If the classification of exocytic events is required, move to step 5.

NOTE: The **Analysis** button click will perform a series of automated tasks to analyze the data. It will create individual folders in the chosen directory to deposit analyzed data. While running, the “run indicator” will change from green to yellow (**Figure 4A, red box**). After the analysis is finished, this will change back to green.

4.2. Find a DataFiles folder (**Figure 4B**) with the complete set of analysis files (as well as feature

extraction files, to be used in classification later) named according to each datafile (**Figure 4C**).

NOTE: A description of these analysis files is included below in the Representative Results section.

5. Classification of exocytic events

5.1. To perform simultaneous classification of exocytic events with automated detection, check the **Classification** checkbox before clicking on the **Analysis** button. For each exocytic event, assign a probability score between 0-1 for each class. An exocytic event is considered classified as one of the four classes if the probability score for that class is > 0.5 .

NOTE: Once the analysis is complete, a new data file folder will appear within the chosen directory. The folder will contain analysis files corresponding to each image file.

6. Spatiotemporal analysis of exocytosis using Ripley's K values

6.1. Create a separate "neurite" and "soma" mask file. First, segment the soma from the neurite. There is no unbiased method for segmenting the soma from the neurites, so the user should be blinded to the conditioning experiment and use the best judgment; an ellipsoid with no obvious neurite extensions is suggested.

6.2. Open ImageJ/Fiji.

6.3. Drag-and-drop the mask file or use **File | Open** and then select the mask file.

6.4. Use the color-picker tool and click on any of the black pixels in the background of the mask to set the color to black.

6.5. Use the polygon-selection tool or freehand to draw an outline around the soma, separating it from the neurites. This requires manual decision-making.

6.6. Click **Edit | Selection | Make Mask**. A new image will open with the circled soma segmented from the rest of the image. This will create a soma mask file.

6.7. Without moving the drawn region, click on the header of the original mask file.

6.8. Click **Edit | Fill** to fill in the circled soma so that only the neurites remain the mask.

6.9. Once separate neurite and mask files are obtained, **Save** both the mask files.

6.10. Open "neurite_2D_network" in Matlab.

6.11. In MATLAB, navigate to directory with all analysis data.

6.12. Change the path “mask name” to the name of the neurite mask, i.e., “MaskFiles/VAMP2pHluorin_488_wt_4_mask_file_neurite.tif”

6.13. Change the “csv_file_name” to where fluorescent_traces.csv file is located, i.e., “MaskFiles/VAMP2pHluorin_488_wt_4_mask_file_neurite.csv”.

6.14. Click **Run** to skeletonize the neurite mask file. This creates a skeletonized version of the neurite mask file and deposits it as a CSV file under the maskfiles folder.

6.15. Next, generate a CSV file for the soma. Open “CSV_mask_creator.m” in Matlab.

6.16. Put in the path for the “mask name” to the name of the soma mask, i.e., “MaskFiles/VAMP2pHluorin_488_wt_4_mask_file_soma.tif”

6.17. Change the “writematrix” to csv filename to be created, i.e., “MaskFiles/VAMP2pHluorin_488_wt_4_mask_file_soma.csv”

6.18. Click **Run**. This creates the new VAMP2pHluorin_488_wt_4_mask_file_soma.csv file.

6.19. Repeat 6.14 for every mask file.

7. RStudio setup

7.1. Open Rstudio and open ripleys_k_analysis.R file.

7.2. Install the package “spatstat” in RStudio by going to **Tools | Install Packages** and typing in “spatstat” followed by clicking **Install**.

NOTE: This only needs to be performed once per Rstudio Installation.

7.3. Run the library spatstat at the beginning of each session.

7.4. Pay attention to two main variables in this case: “neuron_mask” and “neuron_datapoints.” Neuron Mask points to the set of mask files to run, i.e., soma mask files.”

NOTE: Run all soma mask files together, separately from Neurite mask files, and vice versa.

7.5. Read in the .csv of the mask files for each of the neurons to be analyzed.

7.6. Run multiple files at once by copying additional neuron_mask_n+1. This will allow to aggregate Ripley’s analysis together, using the script described in section 8.

7.7. Now check the second variable, “neuron_datapoints”.

7.8. Read in the "X_fluorescent_traces.csv" file generated by the analysis program (by features all extracted_R file) with the x,y,t positions of the exocytic events, as well as the neurite-specific file of the x,y positions for the 2D network. This goes in the "neuron_datapoints" position.

7.9. In RStudio **Select Code | Run Region | Run all**. This generates several plots, including the grouped Ripley's K values as well as density plots.

7.10. Save plots by going to **Export | Save Image As | Select** appropriate image format and directory, and input appropriate file name, then click **Save**.

NOTE: The plotting function for the heatmaps is called in the script using "plot(density(soma_data,0.4))". The number "0.4" here represents how smoothed-out the density function should be. It can be changed to fit user data in a meaningful way, but if comparisons are to be performed between different heatmaps, the number must be the same between them.

7.11. **Export** or **Save** image from Rstudio. If a heatmap requires further editing, choose an appropriate file type (SVG or EPS).

8. Ripley's analysis

NOTE: The Ripleys_k_analysis.R file also automatically generated Ripley's k value plots. Running the entire script will automatically run the functions mentioned below, but it is included in detail if one wishes to run each portion of the script individually or make changes to the analysis.

8.1. First, run the envelope function for each cell. This function simulated complete spatial randomness (CSR) to test the Ripley's K value of the exocytic event point pattern against.

```
Data_envelope_1 = envelope(soma_data_1, Kest, nsim = 19, savefuns = TRUE)
```

```
Data_envelope_2 = envelope(soma_data_2, Kest, nsim = 19, savefuns = TRUE)
```

8.2. Next, pool these envelopes together and create one estimate of CSR for the group:

```
Pool_csr = pool(Data_envelope_1, Data_envelope_2,...)
```

Next, run the Ripley's K function for all data points.

```
Data_ripleys_k_1 = Kest(soma_data_1, ratio = TRUE)
```

```
Data_ripleys_k_2 = Kest(soma_data_2, ratio = TRUE)
```

8.3. Once complete, pool the Ripley's K values together and bootstrap their confidence intervals with the following command:

```
Data_pool = pool(Data_ripleys_k_1, Data_ripleys_k_2,...)
```

8.4. Bootstrap with the following command:

```
Final_Ripleys_K = varblock(fun = Kest, Data_pool)
```

8.5. Plot the data.

8.6. If a hard-statistical difference is required, Studentised Permutation Test is included in the spatstat package to test for a difference between groups of point patterns:

```
Test_difference = studpermu.test(all_points_to_test, exocytic_events ~ group, nperm = np).
```

REPRESENTATIVE RESULTS:

Here the GUI (**Figure 2A**) was utilized to analyze exocytic events from three VAMP2-pHluorin expressing neurons at 3 DIV using TIRF (total internal reflection fluorescence) microscopy. E15.5 cortical neurons were isolated, followed by transfection with VAMP2-pHluorin and plating using the protocols as outlined in Winkle et al., 2016 and Viesselmann et al., 2011^{11,12}. The methodology of imaging parameters is as outlined in Urbina et al., 2018². Briefly, TIRF microscopy was used to image the basal plasma membrane of neurons every 100 ms for 2 min. **Figure 2, Figure 3, Figure 4** show a step-by-step guide to analyze exocytic events. The folder where the neuron images are located is selected, and a directory to deposit the final analysis datafiles is chosen (**Figure 2A**). Using the MaskMaker function, a mask is generated for the neurons, which is inspected in the GUI (**Figure 2B**). In this instance, the cell mask is of good quality, and the analysis can proceed. Should a mask be insufficient, a mask can be created in ImageJ (**Figure 3**). After using the MaskMaker function or creating a mask in ImageJ and selecting the directory where the mask files are located, the analysis is performed (**Figure 4A**). Results are generated in the DataFiles folder when the analysis is finished (the yellow indicator changes back to green) (**Figure 4B**).

Datafiles are automatically generated and named according to the raw data files provided.

Assuming the datafile is named X:

X_tracking: This file includes x,y position and frame number of each event as well as bounding boxes which can be used to draw boxes around each event. Age indicates the number of frames past the initial detection where an event is a distinct gaussian puncta. If classification is checked, the classification results will appear in this file, which indicates the probability of an exocytic event belonging to one of four classes. If the probability is greater than .5, and greater than other probabilities, then an exocytic class was chosen.

X_fluorescent traces: This file includes x,y position and frame number of each event. In addition, it includes fluorescent intensity measures in a region of interest around each event 2 seconds before and 10 seconds following the peak $\Delta F/F$ for each event (indicated by the Timepoint columns).

X_cell_statistics: This file includes the cell area, total image time, and automatically calculated frequency of exocytic events for each cell (in events/mm²/minute).

Feature extraction files include:

X_contrast: Contrast. A measure of the intensity contrast between a pixel and its neighbor over

the whole image.

X_correlation: Correlation. A measure of how correlated a pixel is to its neighbor over the whole image.

X_energy Total energy. Defined as the squared sum of the pixel intensity.

X_homogeneity measures the closeness of the distribution of elements in the ROI to the ROI diagonal.

X_ring_fluorescence: the average fluorescence of border pixels.

X_SD: Standard Deviation. This is defined as the standard deviation of the ROI.

Examples of average fluorescence traces \pm SEM from each exocytic class were plotted from X_fluorescent traces file (**Figure 5A**). Using the Cell_statistics file, the frequency of exocytosis for each class was plotted for each neuron (**Figure 5B**). With the Classification checkbox clicked, the program assigns each exocytic event to a class, plotted in **Figure 5C**. Following classification, the Ripley's K analysis code to determine if exocytic events are random, clustered, or dispersed in space and time. Density heatmaps of the localization of exocytic events (**Figure 5D**) are generated. These reveal expected clustered "hotspots" in distinct regions of the neuron. Next, Ripley's K analysis was performed for the soma, neurite, and clustering over time (**Figure 5E**). The Ripley's K value and SEM (black line and blue shaded region, respectively) rise about the line of complete spatial randomness (red dotted line), suggesting statistically significant clustering.

FIGURE AND TABLE LEGENDS:

Figure 1: Representation of exocytic analysis and classification. (A) Outline of the analysis pipeline for the GUI. Cells are segmented from the background before exocytic events are identified and tracked. Parameters such as the peak $\Delta F/F$ and $t_{1/2}$ are calculated from fluorescent traces of exocytosis in a ROI around the event pre and post-fusion. (B) illustration of the four modes of exocytosis and example image montages. After fusion, events may proceed instantaneous to FVF or KNR (FVFi and KNRI), or a delay may be present before the onset of fusion fate to FVF or KNR (FVFd and KNRd).

Figure 2: Step by step example analysis. (A) First, datasets are chosen (1., red box), and a directory is chosen to place analysis files (2., red box). Next, the framerate and pixel size are specified (3., green box). Here, 0.08 μm pixel size and 100 ms framerate were used. The MaskMaker function button is then pushed (4., blue box). A folder titled "MaskFiles" is automatically created in the chosen directory containing a mask file for each image file in the dataset. (B) When datasets are loaded, selecting a data file and/or mask file will display the first frame of the images for ease of comparison (Green box). Mask files may not be completely correct; this mask file can be corrected for errors, or a new mask file may be made manually

Figure 3: Creating a mask file manually in ImageJ. (A) First, open the file for making a mask. The

“Polygon Selections” button is outlined in red. By clicking around the edge of the cell, a polygon outline is created. **(B)** How to create a mask from the polygon outline. By selecting **Edit | Selection | Create Mask**, a black and white mask will be created from the polygon (right image).

Figure 4: Analysis of exocytic events. **(A)** Demonstration of the analysis button (orange box) and a running analysis. Notice the run indicator turns yellow while an analysis is being performed. **(B)** Example analysis files that are created in the chosen directory when the analysis is complete. DataFiles contain all analysis files of the exocytic event. **(C)** Analysis files generated in the DataFiles Folder. Color boxes represent the open files in subsequent images. **(D)** Three open files, “X_fluorescent_traces.csv” (red) and “X_Cell_statistics”(green), and “X_tracking”(blue). Fluorescent traces contain the x,y position and frame number of each event, as well as fluorescent intensity at each ROI Cell_statistics contains summary information of whole-cell exocytic statistics such as frequency of exocytosis. X_tracking contains position and time information for each exocytic event as well as the probability of each class of exocytosis for each event, represented as a number between 0-1 (>0.5 indicates an event belongs to a particular class).

Figure 5: Representative results. **(A)** Average fluorescence traces +/- SEM from each exocytic class **(B)** Frequency of events plotted for three murine cortical neurons for each exocytic class. These data values were plotted from “X_Cell_statistics”, using the classes assigned in “X_tracking”. **(C)** Distribution of the classes of exocytosis for the same three cells used in A). Here, the ratio of each mode is plotted. **(D)** Density plot of where exocytic events are occurring as generated in the Ripley’s K Analysis portion of the protocol. This can be interpreted as a “heat map” of the spatial likelihood of where events are occurring. **(E)** Ripley’s K analysis of three cells used for A) and B). The red line indicates what value completely spatially random distribution of exocytic events would be. The black line indicates the aggregate Ripley’s K value for the three cells in this example, and the blue shaded region represents the confidence interval. Here, the shaded region notably falls outside of the line of complete spatial randomness between ~0.25-1 μm , suggesting exocytic events are clustered at those distances.

DISCUSSION:

When using the exocytic detection and analysis software, please consider that the program only accepts lossless compression .tif files as input. The .tif image files may be 8-bit, 16-bit, or 32-bit grayscale (single channel) images. Other image formats must be converted into one of these types before input. For reference, examples used here are 16-bit grayscale images.

Inherent in the automated detection process, the timelapse image sets are processed for the automated background subtraction and photobleaching correction. For background subtraction, the pixels outside of the masked region of the mask file are averaged over the timelapse of the whole image, and the average value is subtracted from the image set. For photobleaching correction, a mono-exponential decay fit is applied to the average fluorescence of pixels in the mask over the course of the video, with the corrected intensity adjusted as follows:

Corrected intensity = (Intensity at time t) \div $\exp(-k \cdot t)$ where k = decay constant

Therefore, no pre-processing is necessary prior to input. These processes, however, critically rely on the image mask to effectively separate the cell from the background, and thus a proper cell mask is necessary for good results.

The automated cell mask creator requires a uniform signal with a sufficient signal-to-noise ratio (ideally, at least 2x the standard deviation of the average background signal) to perform well. Empirically, the CAAX box tagged with a fluorescent protein, which inserts into the plasma membrane upon prenylation, works well. There is no requirement that the signal can be maintained throughout the timelapse imaging of exocytosis, as a suitable mask can be created from a high signal in the first 10 frames of the sequence. However, if the cell morphology changes significantly during the imaging paradigm, care should be taken.

When using the automated detection software, include the framerate and pixel size for accurate temporal and spatial outputs. If no framerate or pixel size is declared, the output will be per-pixel and per-frame. As a rule-of-thumb, vesicle diameters in developing neurons are on the scale of ~ 100 nm¹³, and thus the automated detection of events may work for vesicles similar in size. As it stands, there is no hard limit on the size of vesicles that can be detected (by pixel area), as automated detection relies on Gaussian-shaped intensity over a large range of gaussian widths. Fusion of vesicles smaller than the width of a pixel can be detected accurately if the intensity of the event meets the signal-to-noise criteria of 2x the standard deviation of the average background signal as the fluorescence diffraction expands over multiple pixels.

This program was developed for detecting exocytic events in developing neurons. However, the software has been exploited to successfully detect exocytic events in other cell lines², indicating the detection algorithm is robust. Although we have used the software for detecting exocytic events in non-neuronal cell types, differences in exocytic event mode between cell types¹⁴ indicate that the classification algorithm may not be suitable for other cell types. Exocytic events in developing neurons were originally classified using three different methods: Hierarchical clustering, Dynamic Time Warping, and PCA, revealing four different classes³. Here all three classifiers are employed in the GUI to categorize exocytic events into one of these established four classes. For each exocytic event, a probability score between 0-1 is assigned for each of the four possible classes. Any class with a probability score > 0.5 is considered to be of that class. This classification has only been used in developing neurons to date. Whether these classes exist in other cell types or at later developmental time points in neurons is not known. A large number of events with probability scores of < 0.5 for any of the classes suggest the classification procedure may not be appropriate for the exocytic events currently being evaluated. If low probability scores are assigned to exocytic events of different cell types, this would suggest that de novo classification is needed as new or alternate modes of exocytosis may exist. The same classification methods used here would need to be applied to the automatically detected exocytic events.

To explore the spatial and temporal clustering of exocytic events, Ripley's K analysis¹⁵ is exploited. Analyzing clustering for neurons involves three separate analyses: One for the soma, one for the neurites, and one for time. The reason for splitting the soma and neurites is to account for the

extreme morphology of neurites, which are often thin enough that they can be treated as a 2D network and apply a 2D variant of Ripley's K analysis. For time, a 1D Ripley's K analysis is implemented for temporal clustering. Ripley's K analysis is a robust method for detecting clustering of point processes and colocalization. The graph of Ripley's K can be interpreted as such: the Ripley's K value + confidence interval has a 5% chance to fall outside of the line of complete spatial randomness at any point, analogous to a p-value of 0.05. Where the value falls outside of the line of complete spatial randomness, exocytic events are clustered (above CSR) or are separated at regular intervals (below CSR) at those distances (x-axis).

The creation of the mask file relies on a high initial signal-to-noise of the cell. pH sensitive markers may not always perform well in illuminating a cell, depending on what protein the probe is attached to. Another option for making mask files if the signal to noise of the exocytic marker insufficiently highlights the cell border is to employ a second fluorescent channel/image for mask creation. If using a different fluorescence marker (tagRFP-CAAX, for example) to make masks, when choosing a dataset, first navigate to the folder with the images to make masks from (for example, a folder containing the tagRFP-CAAX images). Use the Mask Maker button here. Importantly, remember to rename the mask files to match the exocytic data file to be analyzed with the above naming scheme (following the example above, the mask files named "CAAX_1_mask_file.tif" will need to be renamed to "VAMP2_488_WT_1_mask_file.tif" to match the VAMP2 image set to be analyzed). Once mask files are appropriately named, return to the **Choose Dataset** button again to navigate to exocytic dataset files to analyze.

The detection of exocytosis is remarkably sensitive, and exocytic events were accurately detected at signal-to-noise ratios as low as a $\Delta F/F$ of 0.01. The sensitivity of the detection depends partially on the variance of the background fluorescence, and events less than 4 standard deviations above the background signal will not be detected.

The detection of exocytic events relies on their transient nature. As a feature of the analysis of transient events, our detection code explores a 20-second window around the exocytic event. In some cell types, kiss-and-stay exocytosis may "stay" for a much longer temporal window than in developing neurons, and the automated detection may not robustly capture these less transient events. This feature is an easily modifiable variable for those comfortable with MATLAB. Similar to the limitation of the 20 second ROI, exocytic events that appear but do not disappear before the last time frame of the video may not be counted as true exocytosis, which may influence the frequency, which is calculated based on the entire length of the time series.

The mask maker included is automated by design and performs well on segmenting complex shapes from the background; however, the fully automated design limits the range of signal-to-noise and signal type that can be used to generate a mask automatically. If the user cannot include a fluorescent cell marker that is uniform and of a significant signal-to-noise ratio, the cell mask will need to be drawn manually.

User based analysis of exocytosis is a time-consuming process and subject to personal bias, as events are not always clearly separated from other fluorescence. The use of an automated

analysis program to correctly identify and analyze exocytic events in an unbiased manner increases analysis efficiency and improves reproducibility and rigor.

Not only does this exocytic event detection work for accurately capturing pH sensitive fluorescence in developing neurons but other cell types as well (Urbina et al., 2018). Whether classification works for other cell types will require determination. Future applications of this technique may be useful at synapses, in non-neuronal cell types, or with novel markers of exocytic vesicle docking and fusion, such as recently described pHmScarlet¹⁶.

ACKNOWLEDGMENTS:

We thank Dustin Revell and Reginald Edwards for testing code and the GUI. Funding was provided by the National Institutes of Health supported this research: including R01NS112326 (SLG), R35GM135160 (SLG), and F31NS103586 (FLU).

DISCLOSURES:

The authors declare nothing to disclose.

REFERENCES:

1. Miesenböck, G., De Angelis, D. A., Rothman, J. E. Visualizing secretion and synaptic transmission with pH-sensitive green fluorescent proteins. *Nature*. **394** (6689), 192-195 (1998).
2. Urbina, F. L., Gomez, S. M., Gupton, S. L. Spatiotemporal organization of exocytosis emerges during neuronal shape change. *Journal of Cell Biology*. **217**(3), 1113-1128 (2018).
3. Urbina, F. L. et al. TRIM67 regulates exocytic mode and neuronal morphogenesis via SNAP47. *Cell Reports*. **34** (6),108743 (2021).
4. Alabi, A. A., Tsien, R. W. Perspectives on kiss-and-run: Role in exocytosis, endocytosis, and neurotransmission. *Annual Review of Physiology*. **75**, 393-422 (2013).
5. Albillos, A., et al. The exocytotic event in chromaffin cells revealed by patch amperometry. *Nature*. **389** (6650), 509-512 (1997).
6. He, L., Wu, L. G. The debate on the kiss-and-run fusion at synapses. *Trends in Neuroscience*. **30** (9), 447-455 (2007).
7. Elhamdani, A., Azizi, F., Artalejo, C. R. Double patch clamp reveals that transient fusion (kiss-and-run) is a major mechanism of secretion in calf adrenal chromaffin cells: high calcium shifts the mechanism from kiss-and-run to complete fusion. *Journal of Neuroscience*. **26** (11), 3030-3026 (2006).
8. Bowser, D. N., Khakh, B. S. Two forms of single-vesicle astrocyte exocytosis imaged with total internal reflection fluorescence microscopy. *Proceedings of the National Academy of Sciences of the United States of America*. **104** (10), 4212-4217 (2007).
9. Holroyd, P., Lang, T., Wenzel, D., De Camilli, P., Jahn, R. Imaging direct, dynamin-dependent recapture of fusing secretory granules on plasma membrane lawns from PC12 cells. *Proceedings of the National Academy of Sciences of the United States of America*. **99** (26), 16806-16811 (2002).
10. Wang, C. T. et al. Different domains of synaptotagmin control the choice between kiss-and-run and full fusion. *Nature*. **424** (6951), 943-947 (2003).
11. Winkle, C. C., Hanlin, C. C., Gupton, S. L. Utilizing combined methodologies to define the

529 role of plasma membrane delivery during axon branching and neuronal morphogenesis. *Journal*
530 *of Visualized Experiments*. (109), e53743 (2016).

531 12. Viesselmann, C., Ballweg, J., Lombard, D., Dent, E. W. Nucleofection and primary culture
532 of embryonic mouse hippocampal and cortical neurons. *Journal of Visualized Experiments*. (47)
533 e2373 (2011).

534 13. Plooster, M. et al. TRIM9-dependent ubiquitination of DCC constrains kinase signaling,
535 exocytosis, and axon branching. *Molecular Biology of the Cell*. **28** (18), 2374-2385 (2017).

536 14. Urbina, F. L., Gupton, S. L. SNARE-mediated exocytosis in neuronal development. *Frontiers*
537 *in Molecular Neuroscience*. **13**, 133 (2020).

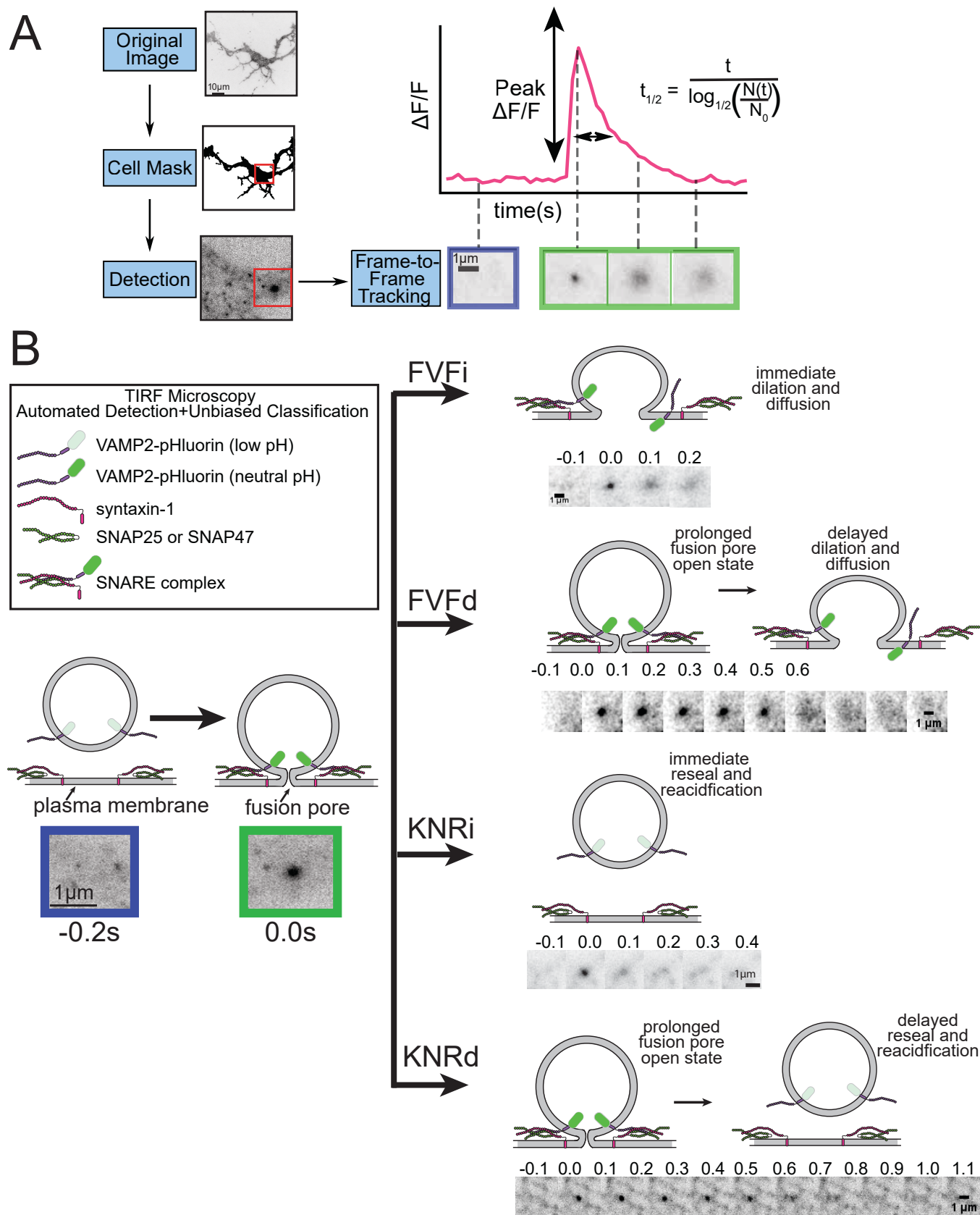
538 15. Ripley, B. D. The second-order analysis of stationary point processes. *Journal of Applied*
539 *Probability*. **13** (2), 255-266 (1976).

540 16. Liu, A. et al. pHmScarlet is a pH-sensitive red fluorescent protein to monitor exocytosis
541 docking and fusion steps. *Nature Communication*. 12 (1), 1413 (2021).

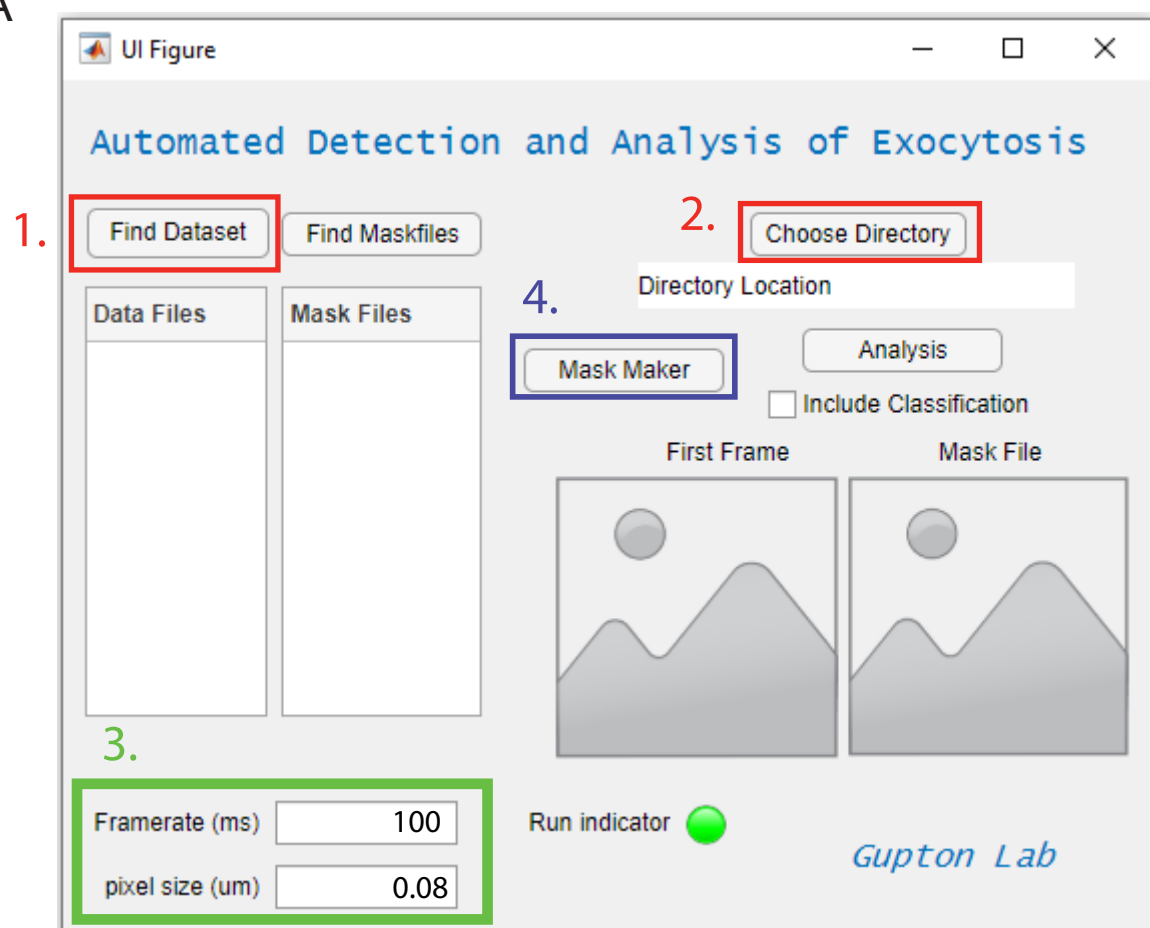
542

Figure 1

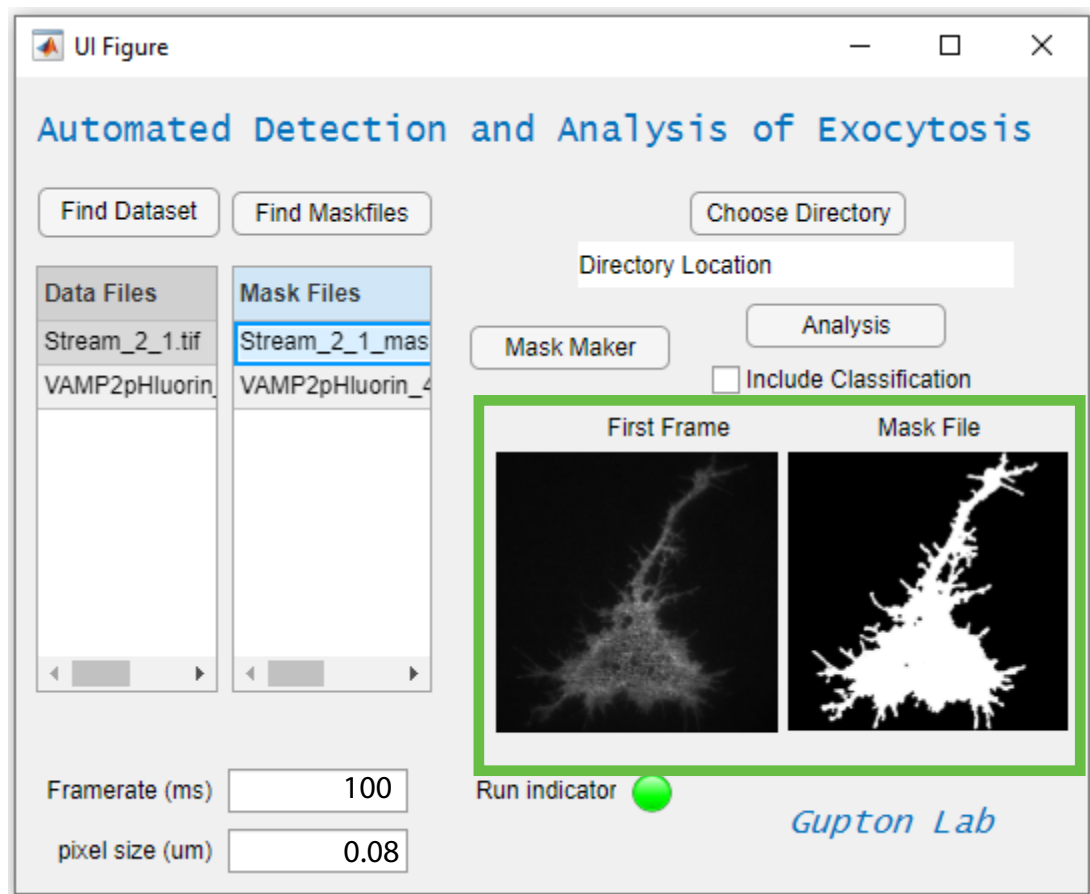
[Click here to access/download;Figure;Figure1.pdf](#)



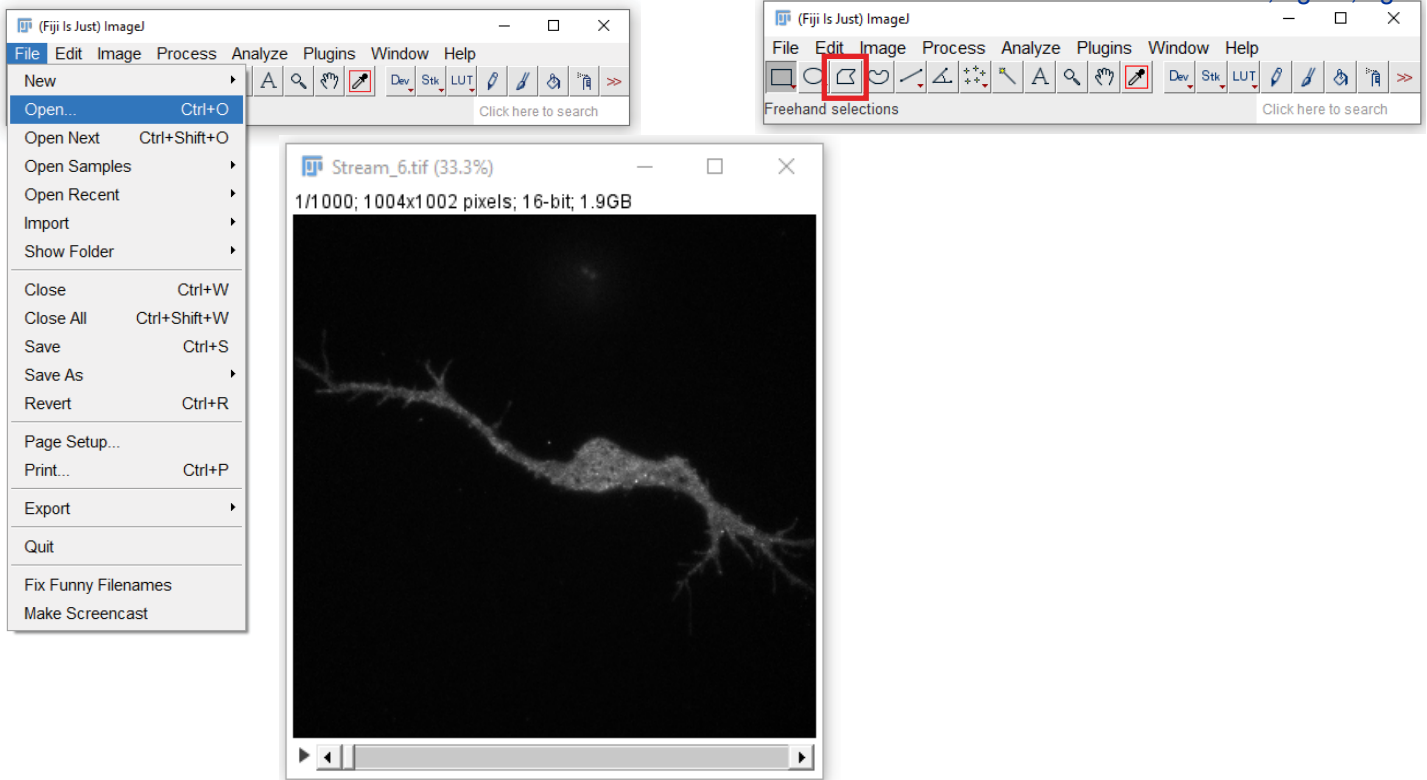
A



B



A



B

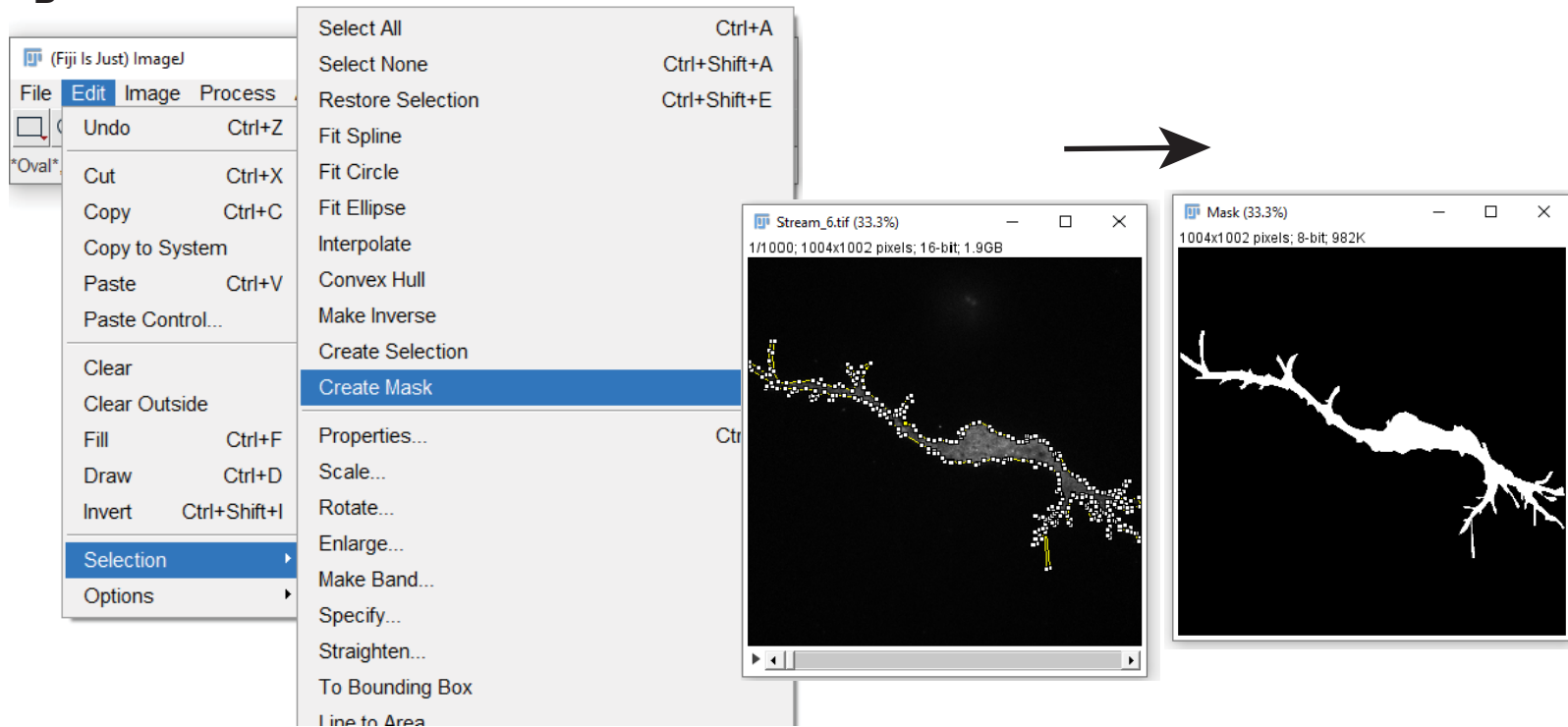
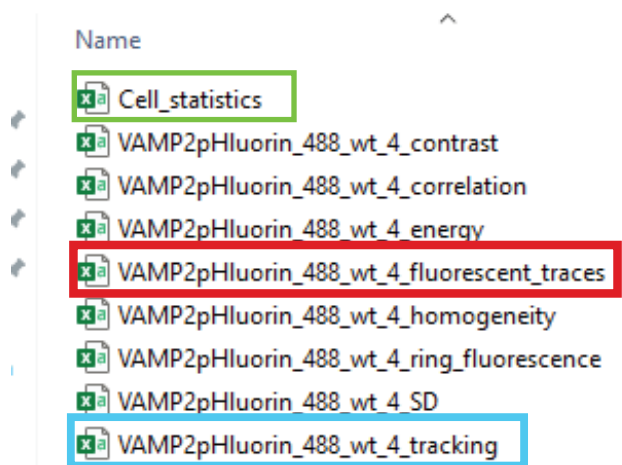


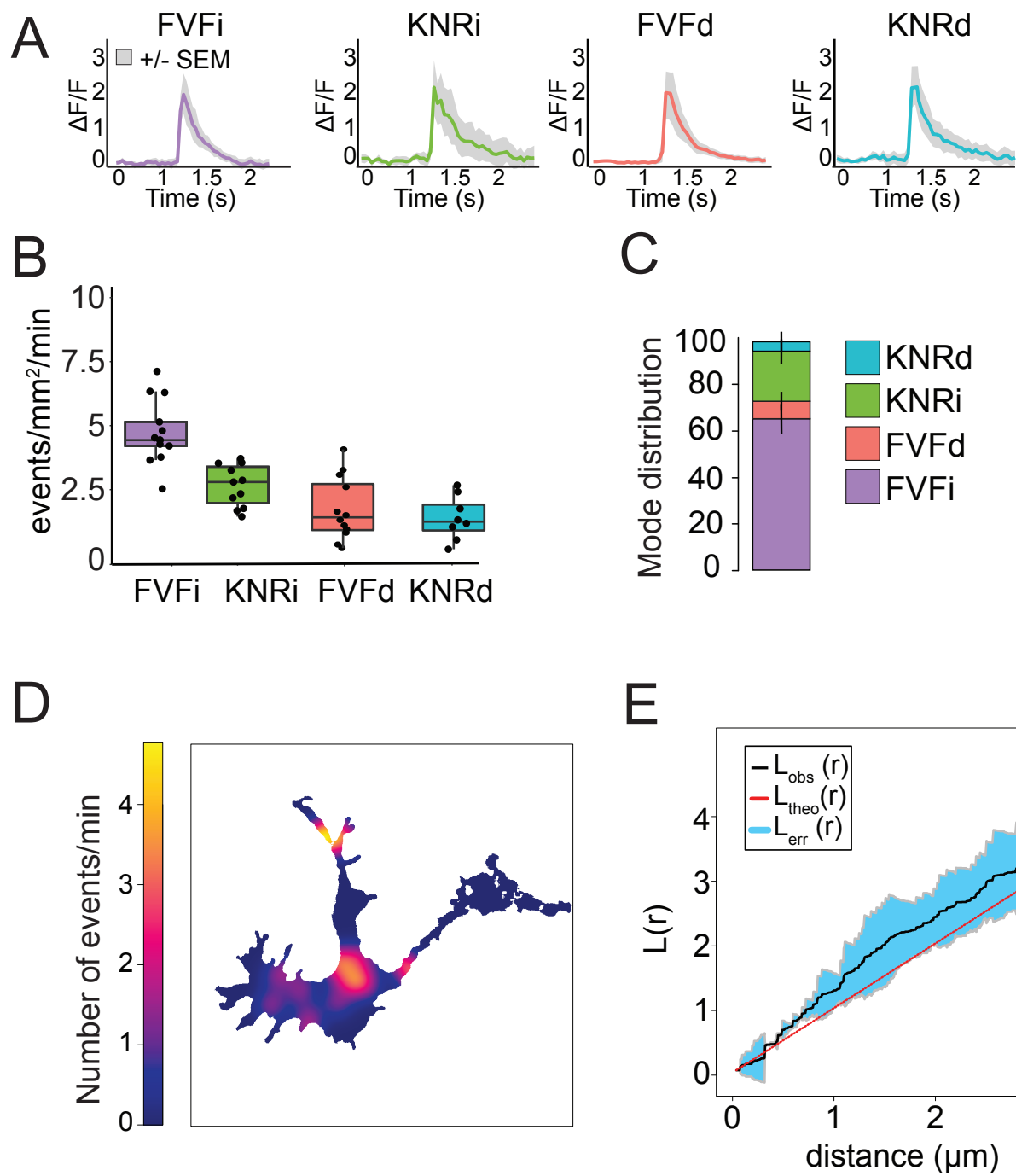
Figure A



	A	B	C	D	E	F
1	index	x_pos	y_pos	frame_num	TimePoint_1	TimePoint_2
2	1	451.5	534.4	17	0	0
3	2	636.752	717.336	18	0	0
4	3	642.8636	685.1364	37	35.19501134	32.75283447
5	4	320.5672	460.2463	40	42.75736961	41.68480726
6	5	682.2571	600.4857	43	36.25396825	35.88208617
7	6	630.2378	773.1297	46	17.39002268	17.62585034
8	7	716.6517	621.7865	54	39.29251701	38.70975057
9	8	677.2798	853.7339	77	16.36054422	14.53061224

	Name	number of events	area(pixels)	area(mm2)	time	frequency
2	VAMP2pHluorin_488_wt_4.tif	25	88409	5.658176	0.3333333333	13.255155
3						

H	I	J	K	L	M
centroid_1	centroid_2	FVFi_probability	FVFd_probability	KNRi_probability	KNRd_probability
451.5	534.4	0.6	0.3	0.1	0.3
636.752	717.336	0.3	0.1	0.8	0.1
451.5	534.4	0.7	0.2	0.3	0.4
636.752	717.336	0.1	0.9	0.1	0.1





Click here to access/download
Table of Materials
JoVE_Materials_list.xls



THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

DEPARTMENT OF CELL BIOLOGY AND PHYSIOLOGY
MEDICAL BIOMOLECULAR RESEARCH BUILDING
111 MASON FARM ROAD
CHAPEL HILL, NC 27599

Friday August 6 2021

To the *JoVE* Editorial Staff:

We have uploaded a new version of our video to dropbox and submit here a revised manuscript addressing all editorial concerns.

We thank you for your time and consideration and we look forward to hearing from you.

Sincerely,

A handwritten signature in black ink that reads "Stephanie L. Gupton". The signature is written in a cursive style.

Stephanie L. Gupton, Ph.D.
Associate Professor, Department of Cell Biology and Physiology
Neuroscience Center
University of North Carolina at Chapel Hill



Click here to access/download
Supplemental Coding Files
AutomatedExocytosisAnalysis.exe





Click here to access/download
Supplemental Coding Files
AutomatedExocytosisAnalysis





Click here to access/download
Supplemental Coding Files
applauncher



Click here to access/download
Supplemental Coding Files
prelaunch



Click here to access/download
Supplemental Coding Files
MWOpenAccessoryView.nib





Click here to access/download
Supplemental Coding Files
MainMenu.nib





Click here to access/download
Supplemental Coding Files
MWSaveAccessoryView.nib





Click here to access/download
Supplemental Coding Files
About.nib





Click here to access/download
Supplemental Coding Files
unresolvedSymbols.txt





Click here to access/download
Supplemental Coding Files
manifest.xml



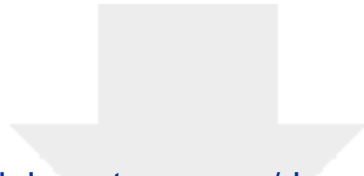


Click here to access/download
Supplemental Coding Files
requiredMCRProducts.txt





Click here to access/download
Supplemental Coding Files
token



[Click here to access/download](#)

Supplemental Coding Files

[inspectorProxyViewMapCache_en_US.mat](#)





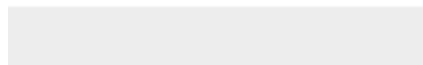
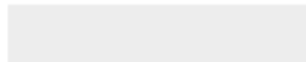
Click here to access/download
Supplemental Coding Files
apps_toolset_apps.xml





[Click here to access/download](#)

Supplemental Coding Files
ml drivetripwireaccess.mlsettings



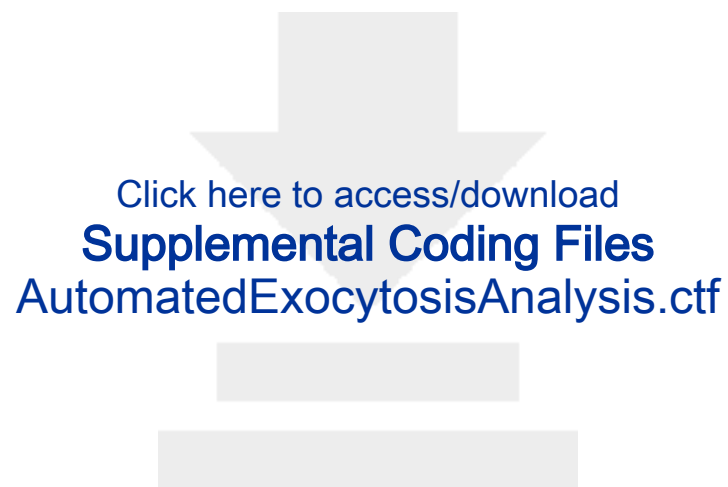


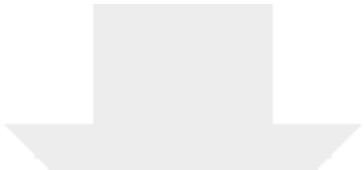
Click here to access/download
Supplemental Coding Files
matlabCompiler.mlsettings



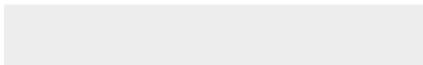



Click here to access/download
Supplemental Coding Files
matlab.mlsettings





Click here to access/download
Supplemental Coding Files
matlab.prf





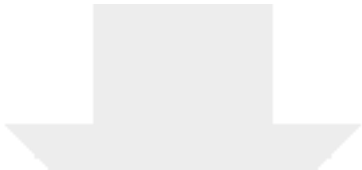
Click here to access/download
Supplemental Coding Files
migratePref.txt



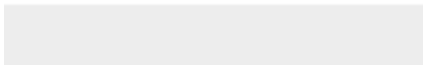
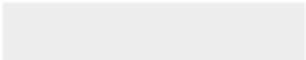


Click here to access/download
Supplemental Coding Files
doc_toolbar_documents.cwd_r7.xml






Click here to access/download
Supplemental Coding Files
Contents.m





Click here to access/download
Supplemental Coding Files
printopt.m



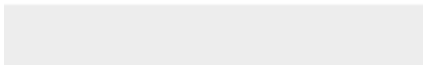


Click here to access/download
Supplemental Coding Files
matlabrc.m





Click here to access/download
Supplemental Coding Files
ctfroot.m





Click here to access/download
Supplemental Coding Files
system_dependent.m





[Click here to access/download](#)

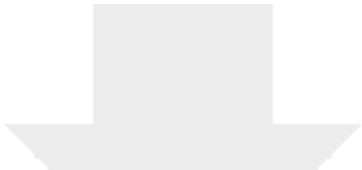
Supplemental Coding Files

[AutomatedExocytosisAnalysis_11921.mlapp](#)

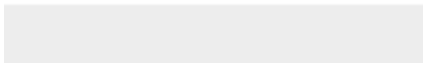
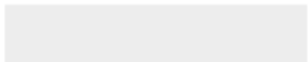





Click here to access/download
Supplemental Coding Files
membrane.icns



Click here to access/download
Supplemental Coding Files
Info.plist





Click here to access/download
Supplemental Coding Files
CSV_mask_creator.m

