Section 3.4 – Code Development for Exocytosis Analysis

An automated approach to analyse exocytic events is necessary given the large number of cells imaged. Several software solutions for analysis of exocytic events (ExEvs) exist[1,2] but they either do not work well or do not work at all for the detection of events in two-colour movies. Previously, a method was developed in our lab to detect ExEvs in HeLa cells[3]. This approach was extended to apply it to two channel movies and to improve the overall workflow for the image analyst.

This section describes in detail the workflow of the detection and graphing of exocytic events employed in this project. There are four steps: i) detection of event locations which maximises detection of all possible events in the cell, ii) recording fluorescence traces from each channel, iii) isolating exocytic events from these traces, and iv) validation of isolated events to winnow the traces down to only the true positive events. All steps are done in Fiji except for the third step which uses R.

3.4.1 – Defining Regions of Interests (ROIs) to Record

The first part of the ExEv detection process is split into two macros: i) def-cell-ROI.ijm to draw and save a region of interest (ROI) around a cell's perimeter to delimit the perimeter around which to search for ExEvs (Fig. 3.13). And ii) exo-analysis.ijm, which calls the saved ROI, detects the locations where ExEvs might have happened, and records the intensity data around it for the duration of the movie. Furthermore, exo-analysis.ijm is split into two key functions: *template_spot_detection* and *exo_analysis_general*.

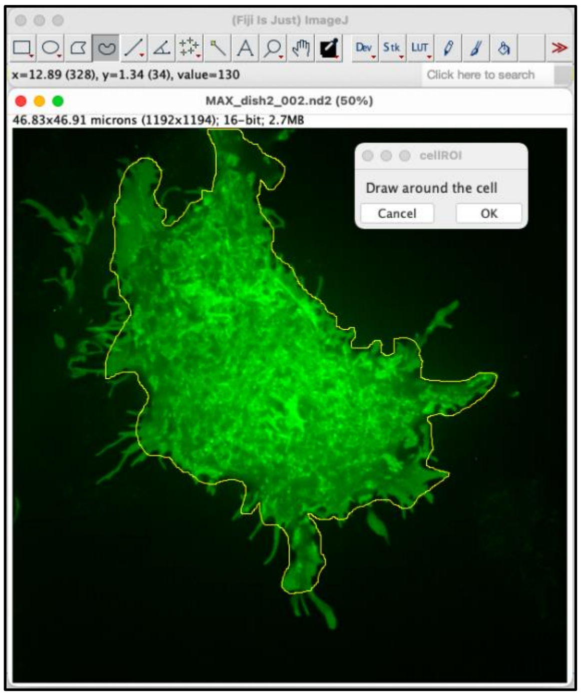


Figure 3.13. Representation of the def-cell-ROI Step of Exocytosis Analysis. The def-cell-ROI FIJI script opens each image and requires the user to draw around the perimeter of the cell. Once the 'OK' button is clicked, the outline is saved as an ROI.

template_spot_detection

Upon the opening of a two-channel image, each channel gets individually duplicated and assigned a name: *template* for the first channel and *subject* for the second (pHluorin and pHmScarlet respectively). Then, *template* undergoes a Z-projection for maximum intensity. This way, all potential ExEvs will appear as high-intensity puncta on a single channel (Fig. 3.13). Then, the *Find Maxima* tool is used to detect these puncta (Fig. 3.14). While this step usually requires user input, it has been automated to apply a prominence that would detect the maximum number of spots under a concentration of 0.55 spots μm^{-2} for the ROI drawn (Fig. S.1). This value was chosen as it was in large excess of the number of possible ExEvs, ensuring that fainter events may also be detected, while trying to keep the number of puncta detected low enough to facilitate the next steps. Lastly, all these points will be saved as a multipoint ROI.

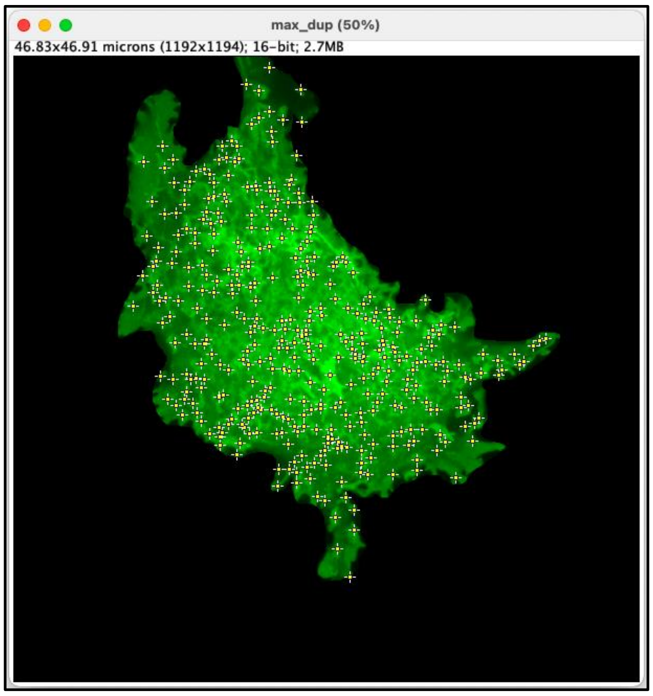


Figure 3.14. Maxima Detected During the template_spot_detection Step. The template_spot_detection step of exo-analysis FIJI script creates a Z-projection for maximum intensity of the selected movie. Here, the brightest value of each pixel is mapped onto a 2D image. This is helpful as any exocytic event that may have happened in the movie will appear as a punctum in the image. Then, the brightest points within the ROI defined in the def-cell-ROI step are saved as a multi-point ROI. The number of points saved can be defined as a number of points per area, as is currently done, or can more simply be set as a specific prominence value.

exo_analysis_general

With the maxima ROIs saved, this second function gets called onto *template*. For each point, it creates an oval ROI (7 px × 7 px) centred around the coordinates of the maxima and it saves the average intensity of pHluorin within the ROI for each frame of the file into a comma-separated value (CSV) file. Then, this function is applied onto *subject* to record the average intensity of pHmScarlet at the same coordinates. Once these steps are completed, all image windows except the original are closed.

To record ExEvs which happen in the second channel but not the first (pHmScarlet and pHluorin respectively), this whole process is ‘recycled’. Channel 2 gets duplicated and assigned the name *template*, from where puncta of potential ExEvs get detected. Then, the intensity data at the coordinates of these puncta get saved for both *template* and *subject*.

To keep track of the different datasets saved and their origin, files were named accordingly. Firstly, to know which channel was used as the template for finding ExEv sites, the term '*recycle'* is employed, with channel 1 (pHluorin) as the template in *recycle 0* and channel 2 (pHmScarlet) for *recycle 1*. Then, to each recycle is appended '*t*' or '*s*' for the data collected on *template* and *subject*, respectively.

Summary

In short, *recycle 0* begins and template_spot_detection is run to detect pHluorin puncta across the movie. Then, exo_analysis_general gets called a first time and the average pHluorin intensity at those puncta gets recorded for the entire duration of the movie. exo_analysis_general gets called a second time and records average intensity of pHmScarlet at the same coordinates across the entire movie. After this, *recycle 1* starts. template_spot_detection detects pHmScarlet puncta, exo_analysis_general records pHmScarlet then pHluorin intensity at the new coordinates. This can also be understood by looking at the diagram in Fig. 3.15 and the code in Fig. S.2.

Written more simply, we detect all events in green and also monitor what is happening in red; then we detect all events in red and also monitor what happens in those events in the red channel.

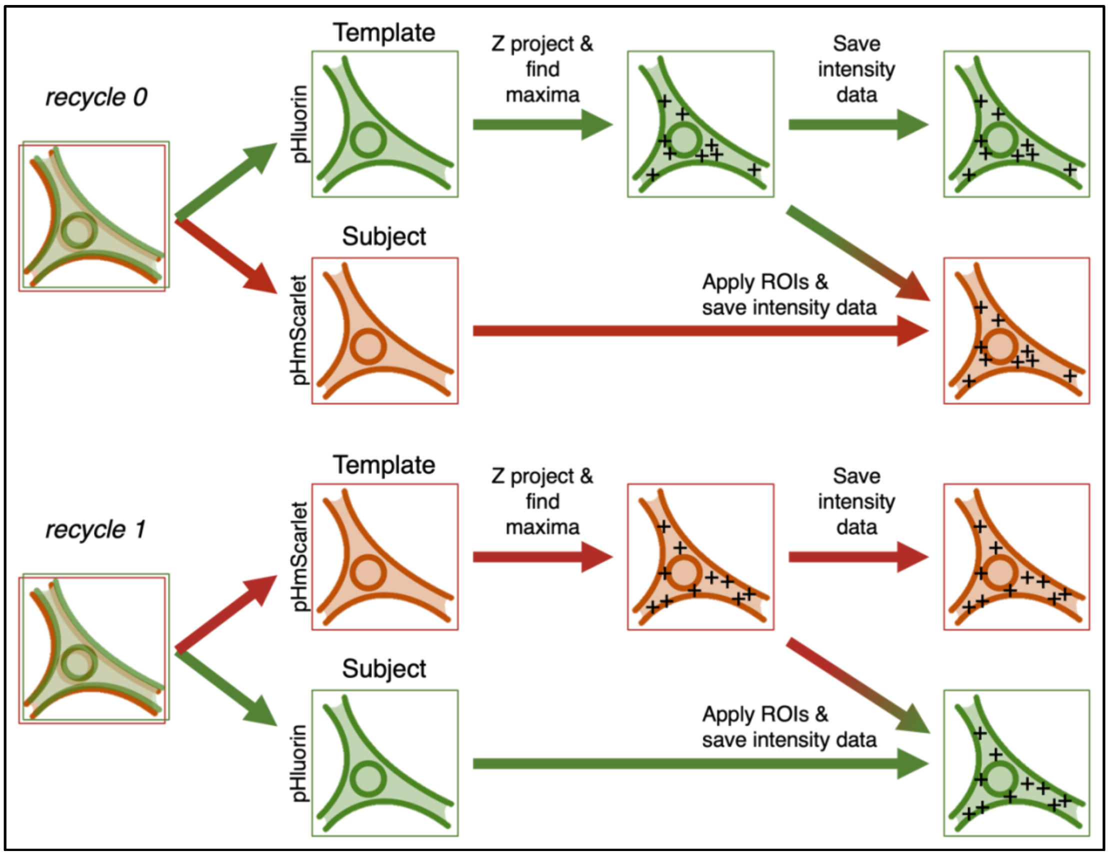


Figure 3.15. Diagram Depicting the Workflow of the Image Analysis Process in FIJI. In recycle 0, the pHluorin channel is used as the template to detect exocytic events. Then, the intensity data at the points detected from the Find Maxima step is recorded across the length of the whole movie. Then, in recycle 1, the process is repeated with pHmScarlet being used to detect potential exocytic events.

3.4.2 – Filtering Peaks for Exocytic Events

The CSV files generated in the previous step were then imported into R and exo_script.R is run, upon which the *template* data gets filtered using the pracma package's *findpeaks* function. To be classed as a positive event, the peak must have a minimum height of *50* and follow a pattern of *1* increasing step before the peak and *3* decreasing steps after. This produces a new data frame with only the intensity traces fitting the above pattern. Additionally, the data for each frame is paired with the same *recycle*'s *subject* data. Lastly, this data frame also includes a timestamp and frame number normalised around the peak.

This step is essential as most of the of the hundreds of 'maxima' returned in the previous steps may just be bright immobile structures of the cell (Fig. 3.14). The code helps filter them out and returns only peaks which reflect the high spatiotemporal variance that can only be observed with vesicles and ExEvs.

From this new data frame, sparkline plots can be generated showing the paired intensities of pHluorin and pHmScarlet at a specific spot (see section 3.5).

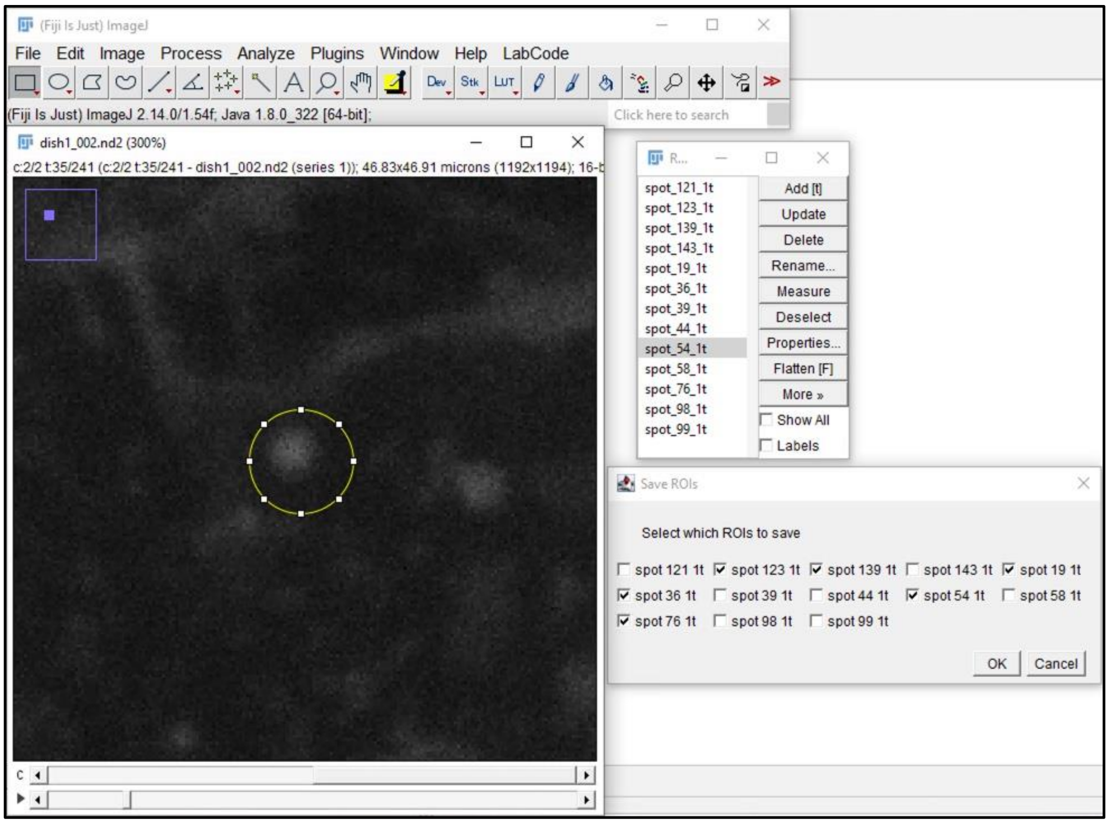
To then be able to return to the image and visualise the peak, it is necessary to have the *x*, *y*, and *frame* coordinates of a peak (also referred to as *xyt*, where *t* is time). A second data frame is therefore created, filtered to only contain the rows containing the frame of the peak. This also provides the unique identification code to that specific punctum, allowing the program to fetch its *x* and *y* coordinates. Once assembled, this data gets saved into a new CSV file.

3.4.3 – Displaying and Saving Exocytic Events

The final script, show-peaks.ijm helps automatically direct the user to the peak's frame and coordinates instead of manually moving to the frame and finding where the given peak's *x* and *y* coordinates are situated. It is used to validate which of the peaks detected in exo_script.R are truly ExEvs and helps generate the figures centred around those peaks. This macro imports the *xyt* coordinates into FIJI, then offers two modes to run the program.

show_all_peaks

In the first mode, an oval ROI centred around each peak at its specific *xyt* coordinates is created for each of the peaks associated with the image and is saved into the ROI manager. Therefore, to examine a peak, the user can click on one of the peaks in the ROI manager, which will set the frame to the start of the peak and display an ROI around the peak. The user can then advance the movie forwards and backwards, as well as be able to zoom in and out to appreciate the peak in the context of the whole cell (Fig. 3.16).



(Fiji Is Just) ImageJ 2.14.0/1.54f; Java 1.8.0_322 [64-bit];

c:2/2 t:35/241 (c:2/2 t:35/241 - dish1_002.nd2 (series 1)); 46.83x46.91 microns (1192x1194); 16-b

spot_121_1t
spot_123_1t
spot_139_1t
spot_143_1t
spot_19_1t
spot_36_1t
spot_39_1t
spot_44_1t
spot_54_1t
spot_58_1t
spot_76_1t
spot_98_1t
spot_99_1t

Select which ROIs to save

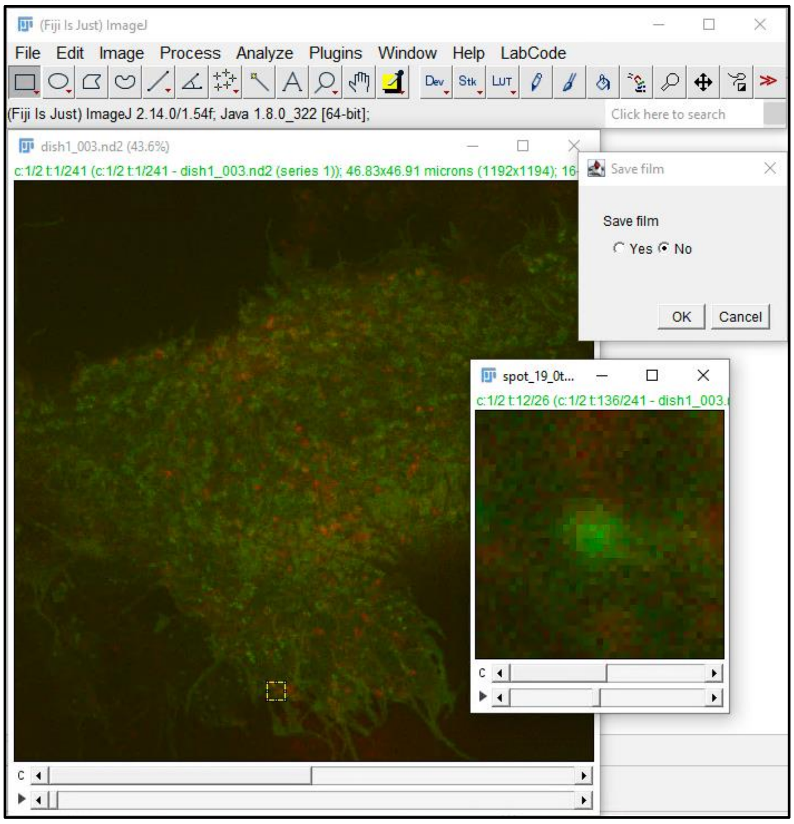
☐ spot 121 1t ☑ spot 123 1t ☑ spot 139 1t ☐ spot 143 1t ☑ spot 19 1t
☑ spot 36 1t ☐ spot 39 1t ☐ spot 44 1t ☑ spot 54 1t ☐ spot 58 1t
☑ spot 76 1t ☐ spot 98 1t ☐ spot 99 1t

Figure 3.16. Representation of the show_all_peaks Dialog and Options. When using the show_all_peaks mode of the show-peaks FIJI script, the spatiotemporal coordinates of every detected peak is called, and saved as individual into the ROI manager. The unique spot number and recycle of each point is also saved onto the ROI name. Selecting an ROI in the ROI manager sets the frame to that of the peak, and draws a circle around the point, aiding the user to visualise it. A dialog box with checkboxes next to each spot ID allows users to select which spots to save a movie for. Clicking 'OK' will generate and save those movies.

Simultaneously, a dialog box also appears to allow the user to select which ROIs, and therefore which peak, they would like to save (Fig. 3.16). The function fig_coord_setup sets up the parameters for saving the selected peaks. The dimensions of the saved image can be dictated within the function. Ultimately, this will return an array providing the information to create a movie centred around the ROI, beginning 10 frames before the peak and ending 15 frames after (Fig. S.3).

open_peaks_indiv

The second mode instead uses the data from fig_coord_setup to create the movie centred around a peak as before (Fig. 3.17). Then, the user has the option to save the movie or not. This process is then repeated for the next peak associated with that movie, and so on until all peaks have been processed.



(Fiji Is Just) ImageJ 2.14.0/1.54f; Java 1.8.0_322 [64-bit];

c:1/2 t:1/241 (c:1/2 t:1/241 - dish1_003.nd2 (series 1)); 46.83x46.91 microns (1192x1194); 16

c:1/2 t:12/26 (c:1/2 t:136/241 - dish1_003.

Save film
Yes [✓ No]

Figure 3.17. Representation of the open_peaks_indiv Mode of the show-peaks Script. In the open_peaks_indiv mode of the show-peaks script, fig_coord_setup is run for each of the peaks associated with the image. Each time, the short movie is created and a dialog box prompts the user if they want to save the movie. After clicking 'OK', the instruction is followed and the movie for the next peak is created.

3.4.4 – Summary

In conclusion, the code used in this project firstly tries to detect all possible ExEvs by mapping the maximum intensity of each frame across a movie onto a 2D image. Then, the fluorescence intensity at the brightest puncta are recorded. This data gets piped into R, where the code filters the traces for a pattern that matches that of an ExEv: a sharp increase in brightness followed by a gradual diffusion of fluorophore away from the site of exocytosis. Simply put, this filters out noise or the movement of certain vesicles across the field of view and ensures that only traces resembling ExEvs are returned. Finally, the user can winnow out false positives and easily save short, zoomed-in, movies of their choice. All the data generated by the scripts can be visualised in the following section.

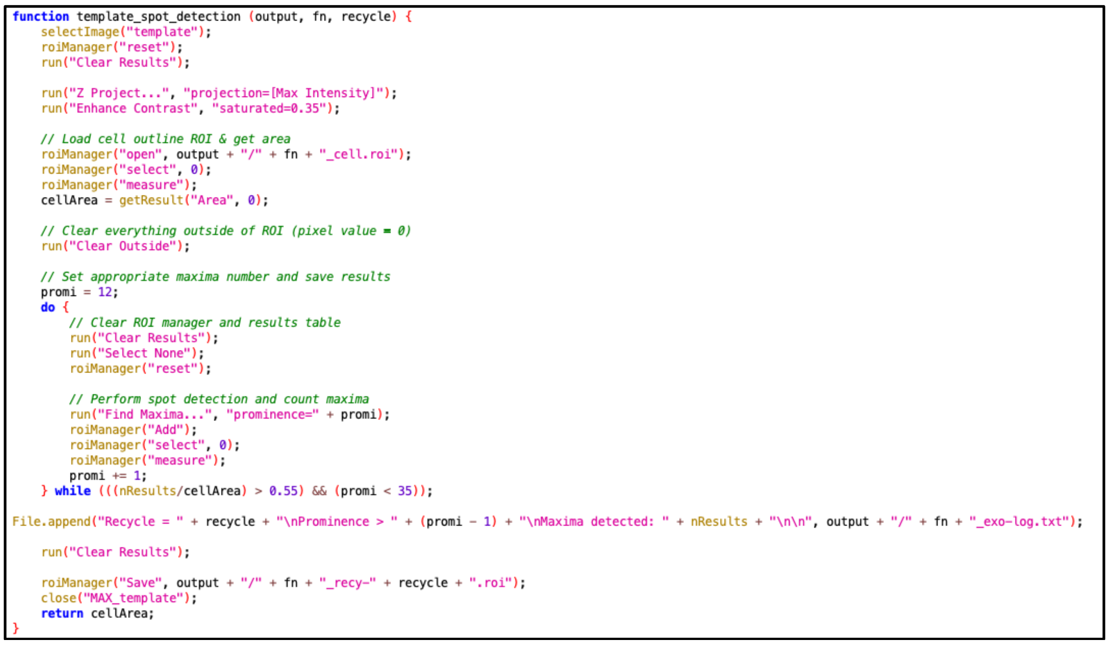
Appendix A – References

1. Liu J, Verweij FJ, Van Niel G, Galli T, Danglot L & Bun P (2024) ExoJ: an ImageJ2/Fiji plugin for automated spatiotemporal detection and analysis of exocytosis. *bioRxiv*, [Preprint] doi.org/10.1101/2022.1109.1105.506585.

2. O'Shaughnessy EC, Lam M, Ryken SE, Wiesner T, Lukasik K, Zuchero JB, Leterrier C, Adalsteinsson D & Gupton SL (2024) pHusion – a robust and versatile toolset for automated detection and analysis of exocytosis. *Journal of Cell Science*, 137 (20):

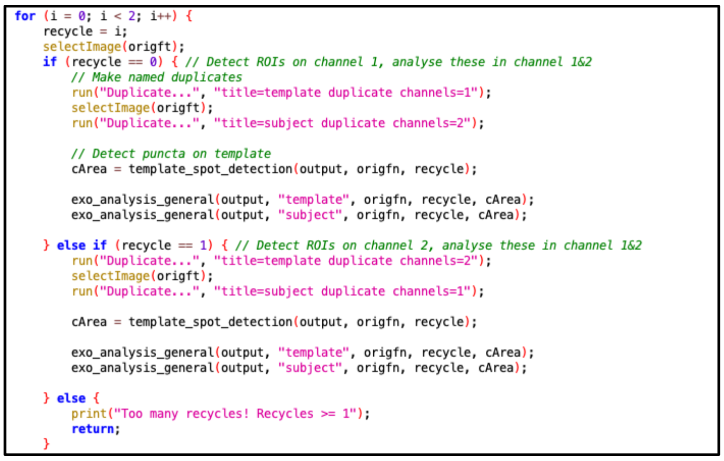
3. Sittewelle M & Royle SJ (2024) Passive diffusion accounts for the majority of intracellular nanovesicle transport. *Life Science Alliance*, 7 (1): e202302406.

Appendix B – Supplemental Figures



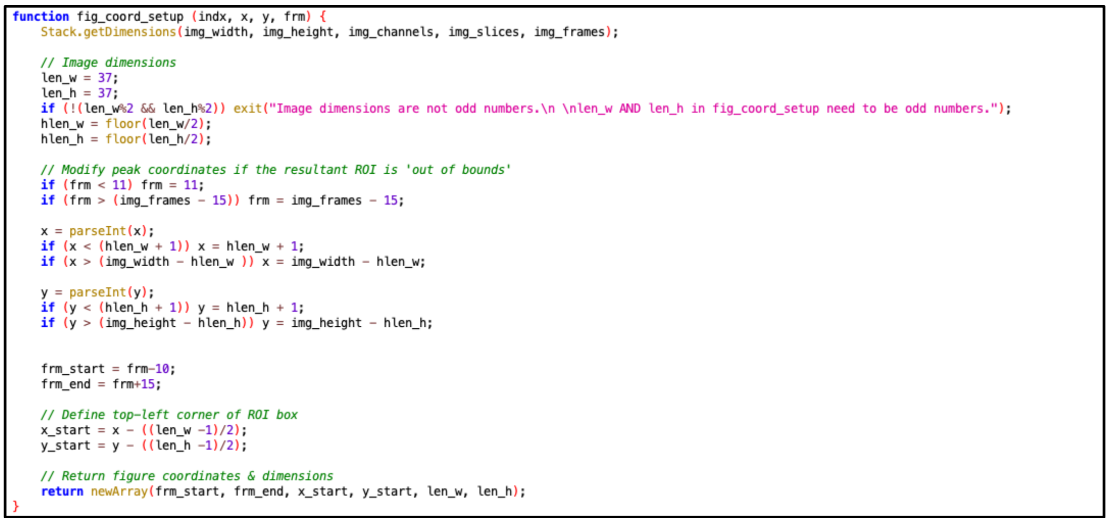
```
function template_spot_detection (output, fn, recycle) {
    selectImage("template");
    roiManager("reset");
    run("Clear Results");

    run("Z Project...", "projection=[Max Intensity]");
    run("Enhance Contrast", "saturated=0.35");

    // Load cell outline ROI & get area
    roiManager("open", output + "/" + fn + "_cell.roi");
    roiManager("select", 0);
    roiManager("measure");
    cellArea = getResult("Area", 0);

    // Clear everything outside of ROI (pixel value = 0)
    run("Clear Outside");

    // Set appropriate maxima number and save results
    promi = 12;
    do {
        // Clear ROI manager and results table
        run("Clear Results");
        run("Select None");
        roiManager("reset");

        // Perform spot detection and count maxima
        run("Find Maxima...", "prominence=" + promi);
        roiManager("Add");
        roiManager("select", 0);
        roiManager("measure");
        promi += 1;
    } while (((nResults/cellArea) > 0.55) && (promi < 35));

File.append("Recycle = " + recycle + "\nProminence > " + (promi - 1) + "\nMaxima detected: " + nResults + "\n\n", output + "/" + fn + "_exo-log.txt");

    run("Clear Results");

    roiManager("Save", output + "/" + fn + "_recy-" + recycle + ".roi");
    close("MAX_template");
    return cellArea;
}
```

Figure S.1. Code for the template_spot_detection Function. The template_spot_detection function works by performing a Z-projection of the movie, where the brightest value of each pixel is mapped onto a 2D image. Then, the ROI defined in the def-cell-ROI step is called to clear all data outside the ROI. The brightest points within the cell outline are then detected as their coordinates saved as a multi-point ROI. The number of individual points can either be defined as a set prominence value, or as is the case above, the prominence value is increased in a do-while loop until the first value where there are less than 0.55 spots µm-2 within the cell outline.



```
for (i = 0; i < 2; i++) {
    recycle = i;
    selectImage(origft);
    if (recycle == 0) { // Detect ROIs on channel 1, analyse these in channel 1&2
        // Make named duplicates
        run("Duplicate...", "title=template duplicate channels=1");
        selectImage(origft);
        run("Duplicate...", "title=subject duplicate channels=2");

        // Detect puncta on template
        cArea = template_spot_detection(output, origfn, recycle);

        exo_analysis_general(output, "template", origfn, recycle, cArea);
        exo_analysis_general(output, "subject", origfn, recycle, cArea);

    } else if (recycle == 1) { // Detect ROIs on channel 2, analyse these in channel 1&2
        run("Duplicate...", "title=template duplicate channels=2");
        selectImage(origft);
        run("Duplicate...", "title=subject duplicate channels=1");

        cArea = template_spot_detection(output, origfn, recycle);

        exo_analysis_general(output, "template", origfn, recycle, cArea);
        exo_analysis_general(output, "subject", origfn, recycle, cArea);

    } else {
        print("Too many recycles! Recycles >= 1");
        return;
    }
```

Figure S.2. Recycles Define Which Image Is Used to Detect Exocytic Events. In recycle 0, channel 1 (pHluorin) is duplicated as the template channel and channel 2 (pHmScarlet) is set as the subject channel. Maxima are detected using the template and the ROIs are saved. These ROIs are then called in exo_analysis_general, where the average intensity within a certain area around each detected maxima are saved across the entire movie. In recycle 1, the channels are inversed, where channel 2 (pHmScarlet) become the template and channel 1 (pHluorin) is the subject.



```
function fig_coord_setup (indx, x, y, frm) {
    Stack.getDimensions(img_width, img_height, img_channels, img_slices, img_frames);

    // Image dimensions
    len_w = 37;
    len_h = 37;
    if (!(len_w%2 && len_h%2)) exit("Image dimensions are not odd numbers.\n \nlen_w AND len_h in fig_coord_setup need to be odd numbers.");
    hlen_w = floor(len_w/2);
    hlen_h = floor(len_h/2);

    // Modify peak coordinates if the resultant ROI is 'out of bounds'
    if (frm < 11) frm = 11;
    if (frm > (img_frames - 15)) frm = img_frames - 15;

    x = parseInt(x);
    if (x < (hlen_w + 1)) x = hlen_w + 1;
    if (x > (img_width - hlen_w )) x = img_width - hlen_w;

    y = parseInt(y);
    if (y < (hlen_h + 1)) y = hlen_h + 1;
    if (y > (img_height - hlen_h)) y = img_height - hlen_h;


    frm_start = frm-10;
    frm_end = frm+15;

    // Define top-left corner of ROI box
    x_start = x - ((len_w -1)/2);
    y_start = y - ((len_h -1)/2);

    // Return figure coordinates & dimensions
    return newArray(frm_start, frm_end, x_start, y_start, len_w, len_h);
}
```

Figure S.3. Code for Setting Up the Coordinates of Exocytic Event Movies to Save. The coordinates for the movie to be saved are input when calling the function fig_coord_setup. These are the x, y, and frame coordinates at the centre and start of the peak. Then, according to the diameter input at the top of this function, the function adapts the those coordinates to return the coordinates necessary to define an ROI in FIJI (the coordinates of the top-left corner of the ROI.