Quant**Escape** | Quantification tools for endosomal escape and damage

Version 2020a

Hampus Du Rietz
Department of Clinical Sciences, Oncology, Lund University, Lund, Sweden.
hampus.du_rietz@med.lu.se

Table of contents

1. Introduction

The microscopy image processing and analysis toolkit presented here were developed with the aim to streamline the analysis of endosomal escape of siRNA from intracellular vesicle, using cytosolic galectin-9 as a sensor of endosome damage. The data organization and workflow are designed for to fit the experimental and technological requirements for this aim. Although the original use was with widefield deconvolution microscopy data, this toolkit can be used with other kinds of live-cell microscopy data as well, and potentially other sensors or markers than galectins.

Requirements that were central in the development of these tools include

- Automated image handling and analysis with batch processing options
- Possibility of processing large-size image datasets
- Efficient and adaptive automated or semi-automated tracking of intracellular structures
- Efficient options for data visualization
- Streamlining the analysis of many events of interest

To achieve this, three main strategies or components have been adopted

1. Pre-defined folder architectures and data organization
2. Standardized operations for data handling and processing
3. A MATLAB program with a graphical user interface

To use the tools, a licenced MATLAB program is required, with the following toolboxes installed:

- Signal Processing Toolbox
- Image Processing Toolbox
- Curve Fitting Toolbox
- Parallell Computing Toolbox

The open source Java image processing program Fiji (ImageJ) is used in some steps of the analysis workflow. Fiji can be downloaded for free from https://imagej.net/Fiji/Downloads. Note that the MATLAB program was used with Fiji 1.52i, and that other versions might have changes that could cause problems with running some functions in the program.

2. Download and setup

Make sure that MATLAB and Fiji are installed on the computer.

The MATLAB App and code, together with the starting folder structure and example datasets for a walk-through demo of the vesicle tracking and analysis are accessible at figshare.com. Follow the link provided in the README-file on GitHub https://github.com/hdurietz/QuantEscape.
All MATLAB files required for running the analysis program are also accessible via GitHub.

For a walk-through demo, follow the instructions in the boxes in the corresponding sections.

3. Home folder and data organization

The MATLAB program works against a well-defined folder architecture. In the provided files, the Home folder is named QuantEscape. This directory contains the following subfolders

- Data repository containing all data in a hierarchical fashion, where acquisitions are organized in the folder of their corresponding experiment. The acquisition folder contains all the data associated with the specific dataset.
- Processing tools containing tools necessary for processing and analysis, including
  - Matlab containing all Matlab functions and code.
  - Chromatic aberration correction model containing the model for using automatic correction of chromatic aberration.
  - Tracking masks containing tracking masks used for object tracking and measurements. For more information regarding tracking mask options, see the Compartment tracking section
- Output data the folder where collected data is exported

WALK-THROUGH: Step 1

The QuantEscape example files and MATLAB code are accessible on figshare.com. The figshare Collection containing the files can be reached via the link in the README-file on GitHub https://github.com/hdurietz/QuantEscape. Download the three files:

- siRNA Example Dataset
- Rab5 Example Dataset
- QuantEscape – Quantification tools for endosomal escape and processing

The MATLAB code can be accessed through GitHub, but is also included in the QuantEscape.zip file. Unarchive the QuantEscape.zip file, and the example dataset files. The QuantEscape App and all MATLAB functions are located in QuantEscape > Processing tools > Matlab. Run the App by clicking the file QuantEscape2020a.mlapp. The App opens in a separate window from the main MATLAB program. In the main MATLAB window, set the current directory to QuantEscape > Processing tools > Matlab.

4. QuantEscape MATLAB App

The QuantEscape graphical user interface enables a more intuitive and easier way to set up experiment parameters and use the processing tools. Experiment and metadata parameters are entered in the Experiments section as described in the following sections. Some parameters are given as default values upon start of the application. These can be changed by configuring the application itself in the App Designer in MATLAB.

The path of the QuantEscape folder should be entered in the Settings tab in the Processing section .

For most of the operations to be executed, the MATLAB current directory should be set to the folder containing the functions to be used, *i.e.* …> QuantEscape > Processing tools > Matlab. The MATLAB current directory is often changed during the execution of the processing operations. In most cases, the Backwards button in the MATLAB Current Folder section can be used to quickly reach the function-containing folder.

5. Setting up experiment parameters

Experiment metadata should be specified in the Experiment section in the App. To activate an experiment tab for processing, the top left box Activate experiment tab must be ticked. A maximum of ten experiments can be processed during the same run. The following information should be entered:

- Experiment | The name of the experiment with appropriate syntax.

- Time series | This value specifies the number of acquisitions in the experiment. The value only needs to be specified to execute the Create Folders operation.
- Ch. A, Ch. B, Ch. C | The name of the channels in the dataset. The order of the channels entered here need to match the order of the channels in the dataset (c1, c2, c3). Enter any name or choose from the list.
- Emission | The emission wavelength for the corresponding channel (nm). This information is used to match image channels during chromatic aberration correction.
- x Dim, y Dim, z Dim | The spatial dimension of the datasets (pixels).
- Include Time Series: All; Selected | Specifies if all acquisitions in the experiments should be processed, or only hashtag-selected acquisitions.

If individual acquisitions of the same experiment have different *z*-dimensions, the acquisitions should be processed separately (with matching datasets) using hashtag-selection.

WALK-THROUGH: Step 2

In the first tab in the Experiment section, check the Activate experiment tab box to activate the tab. As experiment name, enter exambleDataset_Rab5. Set the number of subfolders for the experiment (acquisitions) to 1 in the Time series field. In the Ch. A list, select YFP-galectin-9, and in the Ch. B list select mCherry-Rab5 and enter 610 in the Emission field to the right. As the example images are 512x512 pixel 10 plane *z*-stacks, enter 512 in the xDim and yDim fields, and 10 in the zDim field.

In the second tab in the Experiment section, again check the Activate experiment tab box to activate the tab. As experiment name, enter exambleDataset_siRNA and set the number of subfolders (time series) to 1. In the Ch. A list, select YFP-galectin-9, and in the Ch. B list select DY547-chol-siRNA and enter 547 in the Emission field to the right. Set the xDim, yDim and zDim parameters as above.

6. Adding experiment folders and data

To add new data into the folder structure, the App can be used to generate the appropriate folder structure of single or multiple experiments. After providing experiment information in the Experiments section, press the Create Folders button in the Pre-processing tab to create all folders with the correct structure. Experiment names must start with a letter, and can only contain letters, numbers, underscores or dashes.
For the image pre-processing step, it is required that the raw images are exported as single-plane tiff-files, with the appropriate strings for identification of the time, plane and channel dimensions. The string pattern contained in the file name should be for example image_ t001z01c1.tiff . Naming should follow the convention

- t1, t2, … t9 for datasets with less than 10 time points
- t01, t02, … t99 for datasets with less than 100 time points
- t001, t002, …t999 for datasets with less than 1,000 time points

- c1, c2 … c9 is usually sufficient for the channels

- z1, z1, … z9 for datasets with less than 10 *z*-planes
- z01, z02, … z99 for datasets with less than 100 *z*-planes

The raw image tiff-files should be manually transferred to QuantEscape > Data repository > Experiment > Acquisition > rawImages, with no additional subfolders.

WALK-THROUGH: Step 3

In the Processing section, under the Pre-processing tab, press the Create Folders button. Two new folders should appear in QuantEscape > Data repository, with the names exambleDataset_siRNA and exambleDataset_Rab5 as entered in the Experiments section.

Transfer the image files in the downloaded rawImages_siRNA and rawImages_Rab5 folders into the respective experiment folders and time series folder, and rename them to simply rawImages. Note that no additional subfolders can exists within this folders.

7. Hashtag selection of datasets

The processing or analysis of specific datasets within experiments can be achieve by adding a hashtag [#] as the last character to the folder name of the datasets that should be included in the operation. Selecting individual acquisitions an experiment folder can be used for

- Image processing
- Creating event panels
- Creating event ROIs

For generating event panels and event ROIs, even more specific data selections can be used

- Selection of data associated with individual cells in an acquisition
- Selection of data associated with individual events

Make sure to deselect the folder after completing the operation.

WALK-THROUGH: Step 4
No hashtag-selections of datasets are required for the walk-through processing.

8. Data pre-processing

After setting up the experiment parameters in the App, and adding the raw image tiff-files to the rawImages folder, pre-processing of the data is possible. This can be executed with multiple expeiments activated in the Experiments section, or only one active experiment at a time. During the pre-processing, images from each channel are sorted into new folders, and then resampled so that all z-planes in a stack is contained within a single tiff-file. Last, maximum intensity projection (MIP) images are generated from from z-stacks, and exported to mipImages in the respective acquisition folders.

WALK-THROUGH: Step 5
Pre-processing of the example datasets can be executed in sequence during the same run, by activating both experiments in the Experiments section, or one at a time. Click the Pre-process button in the Processing section. The progress can be monitored in the main MATLAB window, and a message should appear when the pre-processing is completed.

9. Opening images with Fiji

Visualizing full multidimensional dataset becomes more challenging as the size of the data increase. Images can be opened with Fiji as virtual stacks, where the data is only read into the random-access memory (RAM) when the specific image is being view when toggling the dataset. Although minimizing the RAM used, the virtual stack mode is often unsatisfactory because of significant lag as the data is constantly being transferred to the RAM during toggling of the dataset, and also because of difficulties maintaining the appropriate window settings (brightness-contrast) when toggling. One efficient way to visualize the data is to read all of the data into the RAM, in which case the maximum intensity projections often have to be used limit size of the data. For identification of *e.g.* galectin-9 recruitment events, maximum intensity projections are also easier to evaluate than the full 3D volume (plane by plane).

Import the image files using the Bioformats plugin, that is provided as a standard with Fiji. As the data is organized in separated folders for each channel, the channels are opened separately and then merged in the software.

WALK-THROUGH: Step 6

In Fiji, go to File > Import > Bioformats. Select a image file in the channel subfolder in the mipImages folder. Press Open. In the import options, select View stack with: Hyperstack. Under Dataset organization, select Group files with similar names and Swap dimensions. Click OK. In the next window, click OK. In the third window, make sure that the dimensions of the stack are in the correct order, *i.e.* Z=1, C=1, T=500. Click OK. Repeat the process for the second channel. When both channel hyperstacks have been opened, stitch them together using Images > Color > Merge Channels...

10. Event calling in Fiji

For subsequent analysis, individual events of interest are identified manually in Fiji, and regions of interests (ROIs) of the events can then be created using the MATLAB App. Open the dataset of interest, and open the ROI manager located in Analyze > Tools. The procedure is performed on a per cell level:

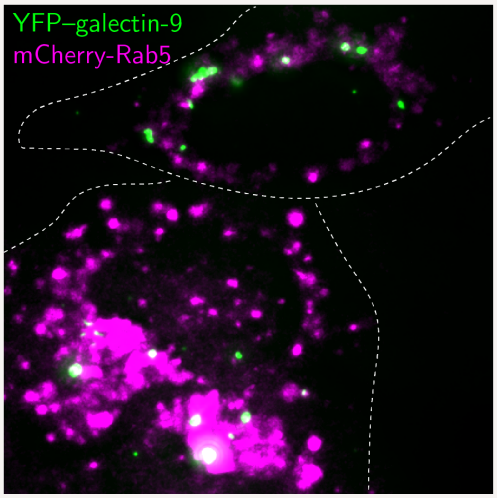
1. Pan the data and identify events of interest
2. Pin-point the event of interest using the Point tool (hair cross). Make sure not to use the Multi-point selection tool. In the case of galectin-9 recruitment, pin-point the galectin-9 foci as close to their appearance as possible (within 10 frames is usually sufficient) and as close to the intensity maximum as possible The x,y–accuracy is more important if galectin objects are crowded, otherwise pin-pointing any position just near the object is usually fine.
3. Click Add in the ROI manager, or press T on the keyboard to add the point to the ROI list.
4. Continue and add the remaining events of interest in the cell to the ROI list one by one.
5. After adding all points of interest, change to the Freehand selections tool.
6. At the time point of the first event of interest identified in the cell, draw an outline of the cell and add to the ROI list. Repeat this process at the time point for the last event. If only one event of interest has been identified, simply add two identical copies of the ROI to the list. It is necessary that the two cell outline ROIs are listed last in the ROI list and in the correct order.
7. Save the list in the data organization in Experiment > Acquisition > cellMetadata by creating a new folder named cell_*n*, where *n* is the number of the cell evaluated in the dataset, written as 01, 02, 03, ... To save the list, click More > List, right-click in the window and select Save as. Save the list as data.txt in the folder. Be sure to add the .txt extension to the name of the file.

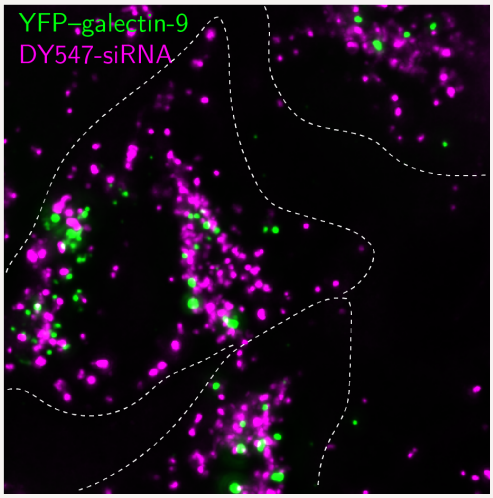
8. Save the cell outline ROIs by selecting the individual ROIs in the list (highlighted), go to Analyze > Tools > Save XY Coordinates. Save the file as startFrame or endFrame, respectively. No file extension is required when naming these files. The files have to be saved separately.
9. Clear the ROI manager list and repeat the process for all cells to be analyzed.
10. After identifying the events of interest in all cells, select a region of the background in the image at the first time point in the dataset and add this to the ROI list. Save this ROI in Experiment > Acquisition > background in the same way as described in above, with the filename background without extension.

WALK-THROUGH: Step 7

In the Rab5 example dataset, details from two cells are visible. In total, 4 events can be identified. These can be analysed using the automated tracking approach outlined below.

In the siRNA example dataset, details from three cells are visible. In total, 11 *de novo* galectin-9 event can be identified. Out of these, 5 are considered to be damage events to siRNA-containing vesicles, that are also possible to quantify using the semi-automated analysis approach described below.





Perform the event calling and save the files as outlined above for each datasets. For each example, the resulting files from the event calling are also provided, in the folders exampleDataset_siRNA_analysed and exampleDataset_Rab5_analysed. If you want to use these files instead, simply copy all folders in the cellMetadata and background folders to the corresponding folders in the exampleDataset_siRNA or exampleDataset_Rab5 experiment folders.

11. Creating ROIs of events

After object or event identification, the relevant data for further analysis of individual objects can be extracted as event ROIs of the full dimensional dataset, with some associated whole-cell data and metadata. In the App, options for creating the ROIs should be specified accordingly:

- Extend ROI specifies the lateral xy-dimension of the ROI, indicated as the pixel distance from the center given by the event location. For example, the value 75 will create a ROI with $2 \times 75 + 1 = 151$ pixels in both x- and y-dimension.
- Times Pre specifies the number of time points before the indicated event to include in the ROI. To include all time points, set the value to the number of time points in the complete dataset.
- Times Post specifies the number of time points after the indicated event to include in the ROI. To include all time points, set the value to the number of time points in the complete dataset.

Note that all z-planes and all channels will be included in the ROI.

If the experiment and ROI parameters have been provided in the App, the processing will begin by creating a new directory for all events identified in each cell in Experiment > Acquisition > eventdata. Each event will have its own folder, containing all the metadata and other data used for further analysis. The event ROIs will be saved as mat-files in Acquisition > eventROIs > cell_*m* > event_*n*. Some data of the cell in which the event occurred, as well as image background information, will be saved in the event folder as well.

WALK-THROUGH: Step 8

When events have been identified in Fiji, and all required files have been saved in their designated subfolders, ROIs can be created of individual events., The default parameter settings in the Create ROIs tab in the Processing section in the App can be used. Both example datasets can be processed in the same run, or datasets can be processed one at a time. Click the Create ROIs button to start the processing. The progress can be monitored in the MATLAB main window, and a message should appear when the step has been completed.

12. Chromatic aberration correction

Some specific processing tools and options are used to minimize effects of chromatic aberration in the microscopy system. A linear correction model for each spatial dimension and for each channel can be generated using images of fluorescent beads. The correction

model can then be applied to the event ROIs to shift the frames of the channels to minimize the aberration. The correction procedure can be automatically applied every time when already generated event ROIs are processed, *e.g.* when creating event panels or performing tracking, although the original event ROI data is always kept uncorrected.

If a correction model is not currently available or should not be used, the Skip chromatic aberration correction option should be enabled under the Event panel or Tracking tab in the App. Correction models for the exmaple datasets are provided, located in QuantEscape > Processing tools > Chromatic aberration correction model. For more information on how to map the chromatic aberration and generate a correction model for the microscopy system, please refer to Appendix 1. Chromatic aberration modeling.

WALK-THROUGH: Step 9
The chromatic aberration correction is enabled by default and will correct the example datasets according the the correction models present in the Processing tools directory. To turn off the correction, check the box next to Skip chromatic aberration correction in the Event panels or Tracking tabs as appropriate.

13. Create event panels

To make inspection and visualization of events easier and less time consuming, ROIs can be put together in new collage – here referred to as event panels. Event panels can be put together with only the event ROI data itself, or with the addition of tracking data in a separate channel. All event ROIs are required to have the same number of channels, but can have spatial and temporal dimensions of different sizes. Different options are available as how to display the z-dimension and any available tracking data, in the Mode drop-down window in the Event panels tab:

- MIP Untracked will create a maximum intensity projection of the event ROIs and will not include any available tracking data in the event panel. This option is suitable for obtaining a quick view of the event ROI data, *e.g.* verifying the appropriate cropping of the ROIs. The $t = 0$ time point will be the frame in which the original coordinates of the event were located during event identification.

- MIP Tracked will create a maximum intensity projection of the event ROIs and will also include available tracking traces in a separate channel. This option is suitable for inspecting the accuracy of the tracking procedure. The $t = 0$ time point will be the frame in which the event is detected during tracking.

- *Z*-Cropped Tracked will create a dataset containing a number of individual *z*-planes as specified by the Extend Planes option, and will also include available tracking traces in a separate channel. The $t = 0$ time point will be the frame in which the event is detected during tracking. Importantly, note that the tracked plane will always be kept at the center *z*-plane in the event panel data stack.

- *Z*-Complete Untracked will create a dataset containing all individual *z*-planes of the event ROIs, and will not include any available tracking data in the event panel. The size of the *z*-dimension is likely to generate event panels of a significant size using this option, making visualization in Fiji cumbersome.

There are some specific variables determining the layout and the dimensionality of the event panels

- Extend Planes specifies the number of planes to include in the maximum intensity projections in MIP Tracked and MIP Untracked, or the number of planes to include in the Z-Cropped Tracked visualization options. Typically, this value should correspond to the number of *z*-planes in the dataset.

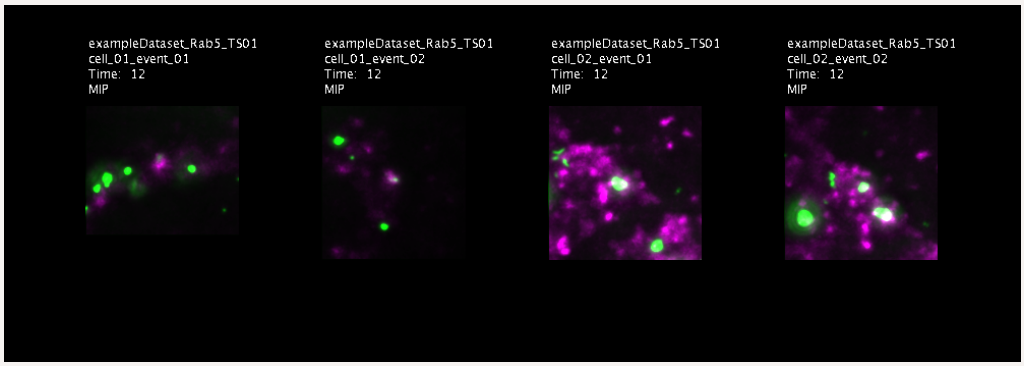
- X Spacers, Y Spacers specifies the horizontal and vertical distances, respectively, separating individual event ROIs in the panel.

- X Panel Dim, Y Panel Dim specifies the horizontal and vertical dimension of the event panel, respectively.

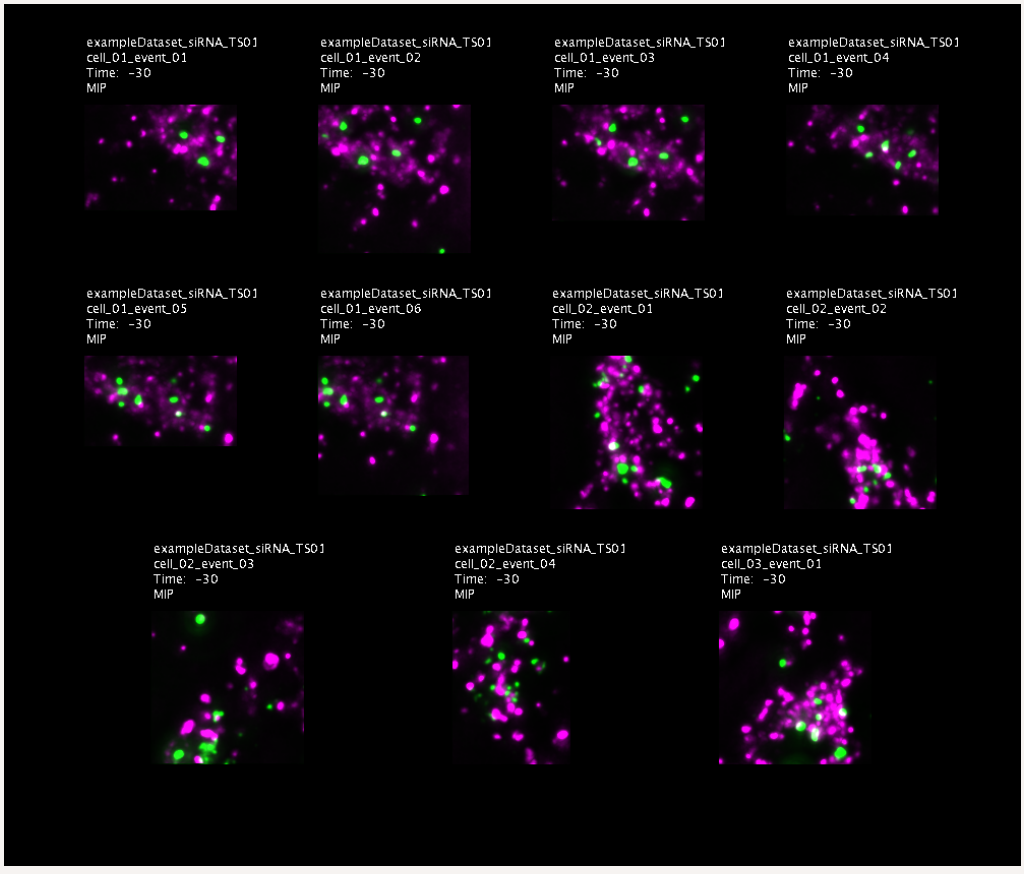
A path to a folder where the event panel should be saved should be provided in the Export Folder field. Typically, this folder can be the first acquisition folder of the experiment processed. If multiple experiments are included in the event panel, a single panel will be generated and exported to the location specified. If desirable, provide a non-standard name of the panel. In the specified folder, the event panel image files will be saved in … > eventPanels > Panel01 > Panel files. The panel files can be imported and visualized in Fiji in the same way as described previously in section 9 .

WALK-THROUGH: Step 10

Create event panels for each example dataset one at a time by unchecking or checking the Activate experiment tab boxes. Select the Event Panels tab in the Processing section. Use all the pre-set parameters. In the Export folder field, enter the path of the respective example dataset being processed, *e.g.* ... > QuantEscape > Data repository > exampleDataset_siRNA > exampleDataset_siRNA_TS01. Click the Create ROI Panel button. The progress can be monitored in the main MATLAB window, and a message should be displayed when the process is completed and all files are exported. The panel files are located in the selected Export folder, in a subdirectory called eventPanels > Panel01 date > panelFiles. Open the exported panel and all channels in Fiji using the Bioformats import tools as described previosly. Select Images > Color > Make composite to overlay the channels, and select look-up tables and adjust brightness-contrast settings. The resulting panels are displayed below. Note that the results depend on the event calling, if the provided example event calling files are not used.



Event panel of example_Dataset_Rab5.



Event panel of example_Dataset_siRNA.

14. EventKey for scoring and curating data

An Excel-file can be created for each experiment, containing a register of all events identified in the previous step,This file can be used to improve the analysis process and track-keeping. The file is saved in the first subdirectory of the experiment folder by default. This file is a working file for keeping track of tracking status and validity, and manual classification or annotation of events. The file can also be used to automatically include subsets of events when creating event panels or reiteration tracking, as described below. It is also used for a final curation of the data during the data collection step. The file can be generated after event ROIs have been created, by activating the experiment in the Experiment section, and clicking the Create Event Key button under the Create ROIs tab in the Processing section. The file can be opened with any standard spreadsheet software.

The identified events are organised in separate rows, where the column contains the following information (left to right): experiment name, acquisition name, cell ID, event ID.

For compartment and cargo tracking, additional notations should be made in the fifth column of the spreadsheet (E). Eligible notations are their meanings are:

- ok | Tracking is correct and no action is needed. If the Skip events validated in eventKey option is enabled, this event will not be included in the next evet panel or tracking procedure.

- exclude | This event will be removed during the final data collection step. Typically, not notation can be used if events are not trackable, or have been misidentified or duplicated during event calling.

- Numbers, *e.g.* 31, can be used to exclude tracking data after a certain time-point (frame number). This is typically a good option if a object is tracked correctly during the first part of the track, but is then lost or confused with some other object, and a longer track is not necessary. The tracking data will be excluded during the collection step after the provided frame number, corresponding to the aligned frame number (0 = time of galectin detection) in the event panel with tracked events.

Other non-numerical notations can also be used, e.g. for memos ('track again') or prior to any tracking ('NA'). Importantly, make sure that the file is saved as a csv-file (comma separated value), named eventKey_curated.csv in the same folder as the original file, and that no empty cells in a column or row containing other notations exists. Notations should not contain commas.

Typically, it is also favourable to make notations regarding cargo contents or release (*e.g.* pos/neg) in the sixth column, and if the event was quantifiable or trackable (*e.g.* yes/no) in the seventh column. If you are using and updating the spreadsheet continuously when tracking events and tuning tracking parameters, remember to save the file after each change so that MATLAB will update the list of events to process. Only annotations in the fifth column (E) are used for any processing or analysis purposes. Other notes can nonetheless serve as an efficient summary of the data and analysis results. For examples of eventKey files for automated and semi-automated workflows, see the provided example spredsheets in ... > QuantEscape > Data repository > exampleDataset_siRNA_analysed > eventKey or ... > QuantEscape > Data repository > exampleDataset_Rab5_analysed > eventKey

WALK-THROUGH: Step 11
Each experiment is processed individually. Activate the experiment in by ticking the box in the Experiment tabs section. Uncheck any other experiment. In the Event ROIs tab, click the Create Event Key button. Open the file with *e.g.* Microsoft Excel or any other equivalent spreadsheet software, supporting editing and saving comma separated value (csv) files.

15. Automated object tracking

Objects or structures in the event ROIs can be analyzed using automated tracking, under the Tracking tab in the Processing section. In principal, the default tracking procedure consists of:

1. Identification of the time point and location at which the event of interest occurs.
2. Forward tracking of the object of interest.
3. Pre-event stationary measurements performed at the location where the object or event is first identified.

This specific procedure is referred to as compartment tracking. A set of tracking parameters are needed to properly perform the analysis, and some adjustments might be needed to successfully track a set of events with a certain variability:

- Track Times Pre-Event specifies the number of frames to include in the analysis prior to the event (the number of static measurements)
- Track Times Post-Event specifies the number of frames to include in the analysis after identification of the event (the number of tracking measurements)
- Min Times Pre specifies the minimum number of time points required.

- Event Detection Search Radius specifies the maximum *xy*-distance from the given coordinates of the object within which to initially detect the event. If there are many objects present in the near vicinity of the object to be detected, this value should be set lower.
- Event Tracking Search Radius specifies the maximum *xy*-distance to search for the objects between frames after detection. If there are many galectin-9 positive objects or objects with higher galectin intensity, they can be erroneously tracked if they are located within the event tracking search radius. If the tracked objects are highly mobile, this parameter value should be adjusted to correspond to the maximum *xy*-distance travelled by the object between two frames.
- z-Search Distance specifies the maximum *z*-distance to search for the objects between frames. If the tracked objects are highly mobile, this parameter value should be adjusted to correspond to the maximum *z*-distance travelled by the object between two frames.
- Start Event Detection specifies the number of time points before the seeding coordinate to start the event detection.
- Confirm Event Detection specifies the number of time points used in the confirmation of the event identification. It represents the number of time points following the initial identification of an object with an intensity higher than the event detection threshold at which the intensity must also exceed this limit.
- Event Detection Threshold specifies the sensitivity of the event detection. The parameter represents the number of standard deviations from the mean signal in the object mask that corresponds to a significant recruitment or accumulation of the marker that defines the event.
- Object Mask Threshold specifies the sensitivity of the algorithm determining object size of galectin-posiyive structures. This is only used if instead of using a fixed mask size, the size of the tracking mask is fitted to each object during tracking.

The channel order of the dataset should be specific in the App to properly execute the tracking of objects. This order is used both for the Automated (compartment) Tracking or Semi-automated (Cargo) Tracking method, and for the data normalization and collection steps as well.

- Primary channel is the channel used for event (recruitment) detection and for post-recruitment tracking, i.e. the galectin channel (c1).
- Secondary channel is the channel containing compartment markers (automated tracking), or cargo (semi-automated tracking), i.e. the Rab5 or siRNA channels (c2).
- Tertiary channel the channel containing any second compartment marker (if dual compartment tracking), or the single compartment marker (if cargo tracking with cargo as secondary channel).

After parameter setup, the tracking analysis is performed for experiments and events selected in the Experiments section. During the analysis, only the raw image data in the event ROIs and the ROI metadata are required. This makes the analysis fast and efficient, and the work required for re-analysis or parameter adjustment is minimized.

WALK-THROUGH: Step 12

Perform the automated tracking analysis using the Rab5 example dataset only. Select the Tracking tab and change the Event Detection Search Radius to 0, and the Event Tracking Search Radius to 5. Use all other pre-set parameter settings. Click the Track Compartments button to start the object tracking. The progress can be monitored in the main MATLAB window, and a message should be displayed when the tracking is completed. To visualize the trajectories, continue to Visualize and validate object tracking below.

16. Semi-automated object tracking

A different strategy is required to reliably track small and highly mobile structures in a crowded environment, where a successful automated event tracking is hard to achieve. This approach uses on the automated tracking procedure of the structure of interest described above, but with extension of the tracking of the object before recruitment of a certain marker. This alternative strategy relies on manual identification of the specific structure of interest until significant recruitment or accumulation of the marker has occurred. As an example, to measuring the release of endosomal cargo from intracellular vesicles in the presence of a membrane damage sensor like a galectin, the releasing vesicle is manually traced up until a clear galectin recruitment has occurred. In the following quantification process, the structure of interest will be automatically tracked using the galectin signal after significant recruitment, and the manual tracking data will be used for the analysis at the time points prior to the galectin accumulation.

To perform the semi-automated tracking procedure, start by creating a MIP Untracked event panel with the events to be analyzed (events that will not be tracked can still be included in the panel), and export the panel and the panel metadata as describedin section 13. Create and open the EventKey file associated with the experiment, as described in section 14. After this:

1. Open the event panel in Fiji.
2. Go to Plugins > Tracking > Manual Tracking
3. Before starting the tracking, make sure that the first channel in the stack is selected with the channel slider. This affects the indexing of the tracking coordinates.

4. Click Add track.
5. Locate the structure of interest in the ROI, and start tracking it from a time point of choice prior to the recruitment of the marker.
6. After clear recruitment of the marker, stop the tracking by pressing End track.
7. Repeat the process for all events to be analyzed. Tracks can be deleted using the plugin options.
8. When the tracking of the events is completed, save the coordinates by right clicking on the window of the coordinate table, and save the data as manTracks.csv in the same folder as the event panel and event panel metadata file.
9. In the App, enter the path to the folder containing the event panel metadata file and the tracking coordinates in the field under the Partition Manual Cargo Tracks section in the Tracking tab.
10. Click the Partition Tracks button to partition all tracking coordinates. A message should appear in the main MATLAB window when the process in completed.

The EventKey file is used as an aid in the above process. In the 5^{th} column, add NA to all rows. In the 6^{th} column, event classification (determined manually) could be entered, *e.g.* positive/negative, low/high etc. depending on what parameters or processes are being evaluated. In the 7^{th} column, note if the event was manually tracked or not (yes/no). Save the EventKey-file as a csv file, named eventKey_curated.csv.

After generation and partitioning the manual tracking coordinates, the automated tracking and quantification can be performed using the App:

1. Specify the tracking parameters, as described in the Automated event tracking section.
2. Fit Cargo Tracking specifies the number of pixels within which to fit the object mask to the object in the secondary channel (cargo), in order to reduce inaccuracy in the manual tracking.
3. Run the tracking by pressing Track Cargo.

WALK-THROUGH: Step 13

Activate the Experiments tab with exampleDataset_siRNA, and create a MIP Untracked event panel as described in Step 10, and open it with Fiji. Perform the manual tracking process as described above. Save the resulting coordinate file as manTracks.csv in the even panel folder, ... > QuantEscape > Data repository > exampleDataset_siRNA > exampleDataset_siRNA_TS01 > eventPanels > Panel01. Paste the path of this folder in the field in the Partition Manual Cargo Tracks section in the App, and click on Partition Tracks. Next, change the Event Detection Search Radius to 3 (default value), and the Event Tracking Search Radius to 1 in the tracking parameters section. Click the Track Cargo button to perform the second step of the tracking. The progress can be monitored in the main MATLAB window, and a message should appear when the tracking is complete.

(Continued)
In the exampleDataset_siRNA_analysed folder, a file containing the manual tracking coordinates are also provided, as manTracks.csv. To use this file instead, the same event ROIs need to be created as in the analysed dataset. Copy and replace the cellMetadata folder from the analysed folder to the new experiment folder, exampleDataset_siRNA > exampleDataset_siRNA_TS01, and remove any files already saved to the eventMetadata folder. Perform the Create event ROIs and Create event panel steps as described above, using the provided data files. Copy the provided manTracks.csv file into … > QuantEscape > Data repository > exampleDataset_siRNA > exampleDataset_siRNA_TS01 > eventPanels > Panel01, and proceed with the next steps of the tracking as described above. To visualize the trajectories, continue to Visualize and validate object tracking below.

17. Visualize and validate object tracking

To inspect the tracking results, create a new event panel with the vesicle tracks as a separate channel. In the Event Panels tab, change Mode from MIP Untracked to MIP Tracked. In the Select Data list, select Compartment tracking data or Cargo tracking data as appropriate, to include the tracking data. Use the same event panel export folder as before, a new subdirectory will be created for the new event panel. Import and overlay the panel files in Fiji as described above. Open the EventKey file to register tracking validity. The following aspects of the object tracking typically requires validation:

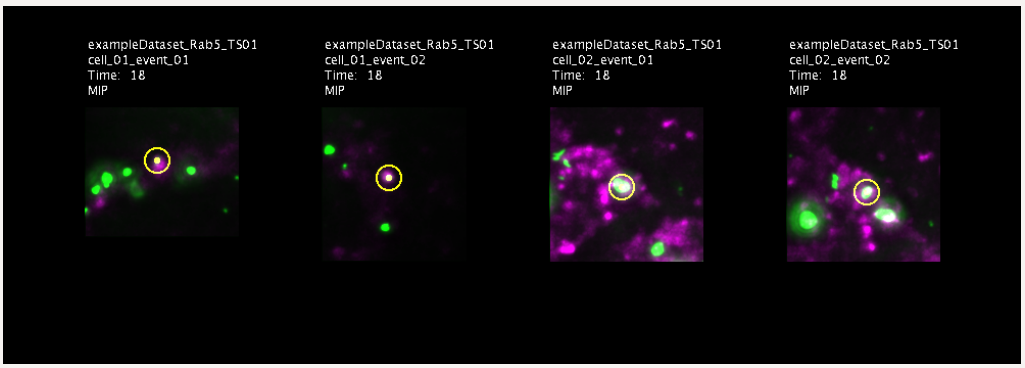
- Event detection ($t = 0$) is set to the first frame with detectable marker recruitment.
- Fitting of the tracking mask and adequacy for the cargo tracking.
- Pre-recruitment stationary tracking is performed at the position of event detection.
- The tracking mask follows the center of the object throughout the tracking. If the object makes a leap exceeding the maximum tracking distance (Event Tracking Search Radius) of the algorithm, either the object center will be missed, typically generating a notch in the quantitative data, or the track of the object will be fully lost.
- The object tracking mask adequately fit the apparent size of the object.

If the object tracking is adequate, replace NA with ok in the 5th column of the EventKey spreadsheet. If you wish to modify the tracking parameters of the event and do the tracking again (as described below), replace NA with *e.g.* redo or some other notation to keep track of the events you are currently working on improving the tracking.

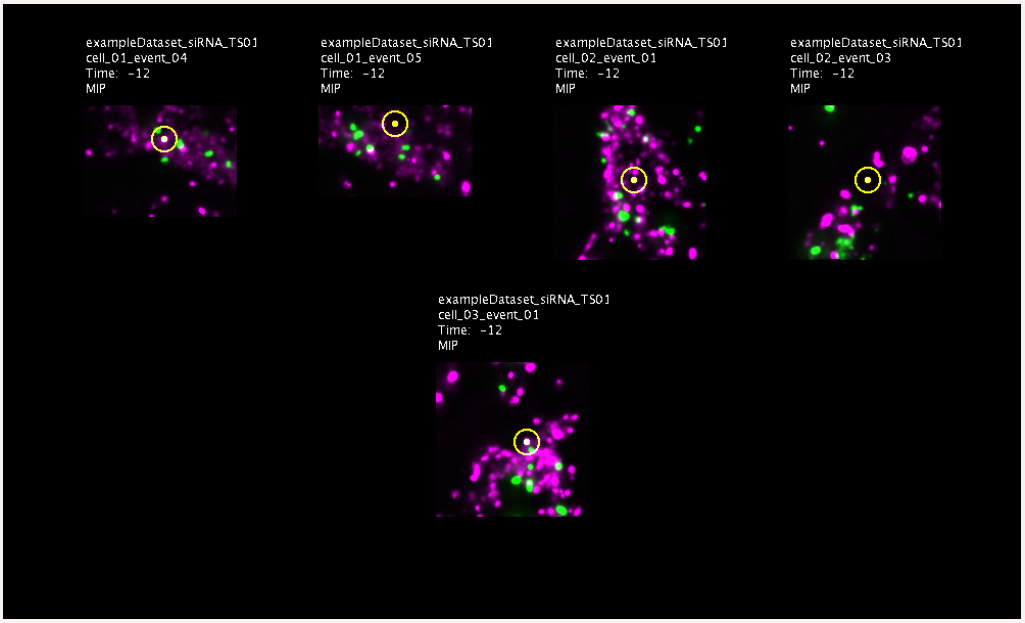
If the track of the object is lost or disrupted, but an acceptable part of the event has been correctly tracked, enter the last frame where the object was correctly tracked (*e.g.* 30) in the 5th column of the spreadsheet. The erroneously tracked last part will then be manually excluded from the dataset during the curation process of the data collection step, without the need of further optimizing the tracking procedure. Save the EventKey-file as eventKey_curated.csv.

WALK-THROUGH: Step 14

Create a new event panel of the respective example datasets, including the tracking data as described above, and open them with Fiji. A third channel, containing the tracking masks, should now be added to the image stack, as shown below for the event ROIs generated with the provided cellMetadata in the exampleDataset_siRNA_analysed or exampleDataset_siRNA_analysed folders. Matching annotated EventKey spreadsheets for the datasets can also be found there, in the eventKey folder.



Event panel with tracking masks overlayed, exampleDataset_Rab5.



Event panel with tracking masks overlayed, exampleDataset_siRNA.

(Continued)
If the tracking parameters setting provided above were used, 2 out of the 4 events in the Rab5 dataset are correctly tracked, and 2 need some tracking parameters adjusted. With the siRNA dataset, 2 events need some parameters to be tuned in order to achieve acceptable tracking performance. In the other cases put ok in the 5th column of the EventKey spreadsheet and save it as described above.

18. Tuning tracking parameters

The tracking parameters of individual events can be modified in the Tune Tracking Parameters tab. The specified parameters will be saved in the roiMetadata.mat file of the event ROI, and will overrun the general tracking parameters provided in the App during analysis. To tune parametes of individual events, specify the following parameters in the Tune Tracking Parameters tab:

- Experiment | The full experiment name, *e.g.* exampleDataset_Rab5.
- Acquisition | The full name of the acquisition, *e.g.* exampleDataset_Rab5_TS01
- Cell | The cell ID, *e.g.* cell_01.
- Event | The event ID, *e.g.* event_01.

The parameters that can be tuned are described further in the Automated object tracking section above:

- Event Detection Search Radius
- Event Tracking Search Radius
- z-Search Distance
- Start Event Detection
- Event Detection Threshold
- Object Mask Size

To tune individual tracking parameters, enter a value in the parameter boxes. If parameters are not specified, the general parameter settings in the App will be used during the tracking.

Keep in mind that the tracking algorithm only maximizes the signal in the mask by fitting it to the area determined by the tracking parameters in-between adjacent frames. It has no other properties where by it can separate or otherwise quality check the tracking.

Some general strategies to optimize the tracking performance can be used, either by changing the general tracking parameters in the App or tuning event specific tracking parameters:

The event is not detected

1. The Event Detection Threshold is set to high, try with a lower threshold. This is typically relevant if the event has a low signal to background ratio.
2. The object is moving to fast, try increasing the Event Detection Search Radius.
3. The seeding-point was not proberly set during event calling in Fiji.

Adjacent object is erroneously detected as the event

1. If neighboring, high intensity object is detected, try increasing the Event Detection Threshold (increments of ~2–3 is usually a good strategy, depending on general object intensities etc.).
2. If the errounosly detected object is very close to the true event, try to reduce the Event Detection Search Radius. This reduces the area from the seeding-point in which the event detection is permitted. This strategy could be limited by the precision by which the seed was originally placed during the event calling step.
3. If a neighboring object pass the same region as the subsequent event of interest, this may be errounosly detected. Try reducing the Start Event Detection parameter (specifiec as the number of framed prior to the seeding timepoint where the event detection should be initiated. Remember that this parameter has to be more than the setting determining the least number of timepoints that have to be analysed before detecting any event (Min Times Pre), that is by default set to 5 frames.

The event is correctly detected but the tracking jumps to an adjacent object.

1. Try reducing the Event Tracking Search Radius.

The object moves faster than tracking algorithm can keep up with.

1. This is importat to check during tracking validation, as the object could be completely lost during tracking. Note that even a single timepoint with the tracking mask placed suboptimally could produce a false "notch" in the quantitative data, contributing to reduced signal-to-noise ratio. Try increasing the Event Tracking Search Radius, but avoid the detection of other objects in the near region.

After modifying tracking parameters, re-tracking of the events can be performed. If the general tracking parameters in the app have not been changed, the outcome of the tracking will not be changed unless tracking parameters of individual events have been modified. This means that either all events can be included in the re-analysis, or events

that are already marked as ok in the EventKey file can be omitted by checking the Skip events validated in eventKey box in the Tracking tab.

WALK-THROUGH: Step 15

If the example cellMetadata files that are provided were used to generate event ROIs for the example datasets, tune the parameters for the respective datasets using the information below by entering it into the corresponding fields in the Tune Tracking Parameters tab. If event calling was performed from scratch in Fiji, and event ROIs were generated from this data, cells and events will have ID's that will most probably not correspond to the provided examples (unless all events were identified in the exact same order). After entering the information into the fields, click the Tune Parameters button. A message will appear in the main MATLAB window when the parameters have been updated. Remember to clear parameters that should not be updated when moving to the next event.

Rab5 Example Dataset

Experiment: exampleDataset_Rab5
Acquisition: exampleDataset_Rab5_TS01
Cell: cell_02;
Event: event_02
Parameter: Event Tracking Search Radius = 1

siRNA Example Dataset

Experiment: exampleDataset_siRNA
Acquisition: exampleDataset_siRNA_TS01
Cell: cell_01;
Event: event_04
Parameter: Event Detection Search Radius = 1

Experiment: exampleDataset_siRNA
Acquisition: exampleDataset_siRNA_TS01
Cell: cell_03;
Event: event_01
Parameter: Event Detection Search Radius = 1

When the parameters have been updated, execute the tracking step again by clicking either Track Compartments or Track Cargo, depending on the example dataset being analysed. Events that were properly tracked can be left out by ticking the Skip events validated in eventKey box, if the EventKey-file was updated as described in the sections above. Create a new event panel with the updated tracking results, as described previously. Here as well, events that were properly tracked can be left out by ticking the Skip events validated in eventKey box in the Event Panels tab. Having tuned the tracking parameters for the above events, all events should now be properly tracked, and ok can be added to all tracked events in the respective EventKey files.

19. Normalize and collect tracking data

Since the data associated with individual events are located in individual folders after tracking, normalizing and collecting (concatenating) tracking data is performed through the App. There are several options for background correction and normalization:

Normalization

- Cell objects | The intensity of marker-labeled objects in each cell is calculated as the median of all pixel values higher than the mean + n × the standard deviation of base 10 log-transformed cell pixel values, where n is the Object Intensity Threshold specified in the Normalize tab (default value = 1). Tracking measurements in the corresponding cell are normalized against this value.

- Pre-release | Tracking measurements are normalized against the mean object intensity before galectin recruitment was detected (i.e. before cargo release), between the start of object tracking up to 3 frames before galectin detection.

- Local background | Object intensity measurements are normalized against the local background intensity, calculated as a rolling average over the number of frames specified as the Local background rolling mean parameter in the Normalize tab (default value = 5).

- Cell background | Object intensity measurements are normalized against the median pixel intensity value of the entire cell.

- No normalization | No normalization of object intensity measurements is performed.

Background correction

- Local | The local background intensity, measured by the ring surrounding the object mask, is subtracted from the object intensity in matched frames. The local background is calculated as a rolling average over the number of time points specified as the Local background rolling mean parameter.

- Cell | The background is defined as the median pixel value of the entire cell. This value is subtracted from the object intensity measurement.

- Image | The image background intensity is measured in the image area defined as the background during the event calling step in Fiji. The mean pixel intensity value in this area is subtracted from the object intensity measurement.

For the Cell objects and Cell background normalization options, and the Cell background correction option, a linear regression model is fitted to the two cell measurements performed using the first frame in the ROI of the first event and the last frame in the ROI of the last event in the cell, using the cell outlines created in Fiji during the event calling step. The models are then used to normalize the object intensity values, or correct for cell background, at the corresponding time-points throughout the trace. This is to avoid problems with normalization related to potential bleaching of the fluoropores during time-lapse acquisition experiments.

Two options exist for estimating object and local background intensities – Mean or Median of the respective tracking mask. In some cases, the mean value measured by the local background mask can be pulled up by high-intensity objects in the vicinity of the tracked object. In this case, using the median mask value might be more suitable for local background correction.

Select if Compartment tracking data or Cargo tracking data should be processed using the Select Data list. This is so that one dataset can be analysed using both strategies, if desirable. Note that with each strategy, previously normalized data of the same type will be written over when the step is executed mulipe times.

Collecting normalized tracking data can be performed in the Collect tab, after selecting the kind of data to be collected. Tick the Curate data after collecting to curate the tracks as specified in the EventKey file, as described in the previous sections.

Several files will be exported to the the data collection folder in the Data output directory during the collection step

- cellObjectData.mat, cellObjectData_primary.xlsx and cellObjectData_secondary.xlsx contains whole-cell measurements in the various channels.

- collectedEvents.mat, collectedEvents_primary.csv and collectedEvents_secondary.csv contains measurements of the tracked objects. Collected events are organized in columns and arranged so that $t = 0$ (*i.e.* the first timepoint at which the event was detected) is located at row 100 for all events.

- collectedCuratedEvents.mat, collectedCuratedEvents_primary.csv and collectedCuratedEvents_secondary.csv contains measurements of the tracked objects after curating as specified in the EventKey file.

- listEvents.mat contains a list of all events collected, and associated metadata on data processing and normalization.

The collected data is exported to a new folder in the Output data directory each time the collection step is executed.

WALK-THROUGH: Step 16

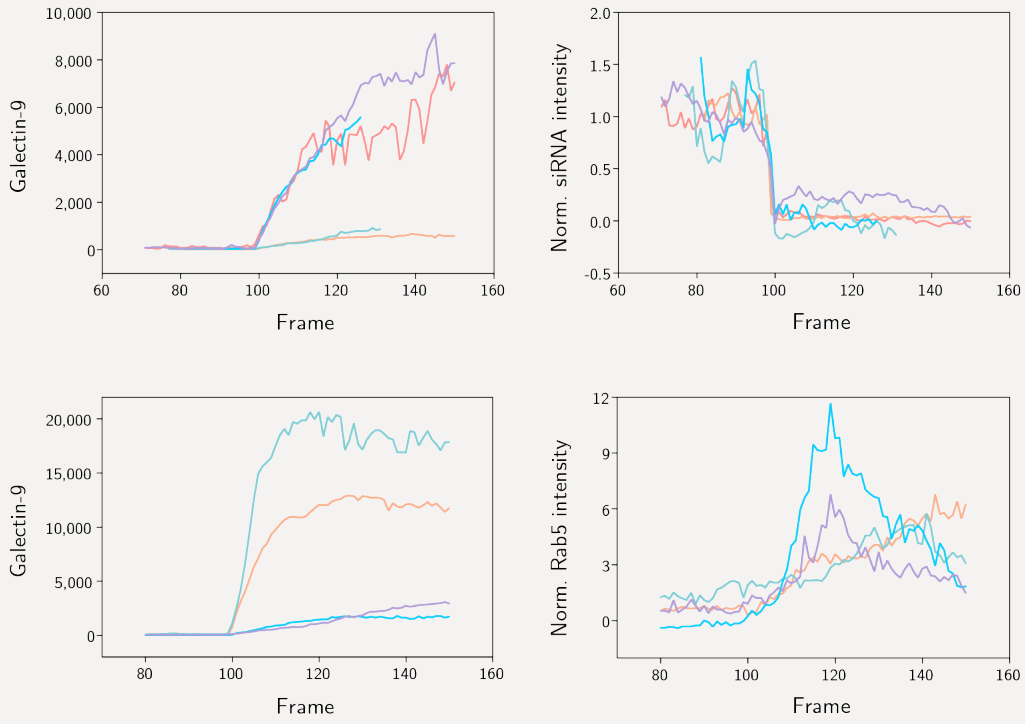
The data normalization and collection is performed separately for each example dataset. In the Normalize tab, select the options for the primary channel (Ch. A) and secondary channel (Ch. B) as outlined below:

Rab5 Example Datset	**siRNA Example Datset**
Normalization	Normalization
Ch. A: No normalization	Ch. A: No normalization
Ch.B: Cell objects	Ch.B: Pre-release
Background correction	Background correction
Ch A: Image	Ch A: Image
Ch B: Local	Ch B: Local
Object measurement	Object measurement
Ch A: Mean	Ch A: Mean
Ch B: Mean	Ch B: Mean
Local background measurement	Local background measurement
Ch A: Mean	Ch A: Median
Ch B: Mean	Ch B: Median

Select the correct data type in the Select Data list. Use the default parameter values for Object Intensity Threshold and Local background rolling mean. Click the Normalize Data button. Follow the progress in the main MATLAB window. When the processing is completed, go to the Collect tab and select the data type in the drop-down list. Tick the Curate data after collecting box, and press Collect Data. When the collection step is completed, the data can be opened by navigating to QuantEscape > Data output > Data collection. Rename the folder, so that it is not written over during the next data collection process.

(Continued)

Open the data files with MATLAB or any other spreadsheet software. If plotted, traces should look similar to what is shown below for the respective example dataset.



20. Post-processing analysis – Endosome profiles

For post-processing data of endosomal markers, a stand-alone MATLAB script can be used, to do quality control and further classification of tracked objects. The folder structure used for this procedure is based on the markers analysed organized in the first layer, with the possibility to have multiple contidions in each subfolder (second layer). For instance, the example datasets provided in QuantEscape > Post-processing > Data repository are mCherry-LAMP1 and mCherry-Peroxisomes, containig the subfolders LLOMe, Chloroquine and Siramesine.

1. In a new folder, create subdirectories with appropriate names for the data collected in the previous step (*e.g.* LAMP1, Rab5 etc.).
2. Create subfolders with names of the different conditions or treatments used. If only one condition was used, a subfolder still have to be created (*e.g.* treated, control, drug A etc.).

3. Add the files in the Data collection folders to the folders in the new structure. Do not use additional folders here.

Open the script named postprocessingEndosomes.m located in QuantEscape > Post-processing > Scripts.

Provide the following parameters:

- p.intensityThresh is the object intensity threshold is the threshold for classification of traces as positive or negative for the markers analysed. This value should be >0 and typically <1.0, which is the mean intensity of labeled cell objects as calculated previously during the data normalization step.
- p.snrThresh is the threshold for signal-to-noise ratio of objects in the cell relative to the image background, to exclude cells with too low marker expression. The signal-to-noise ratio is calculated as by subtracting the image background intensity from the mean cell object intensity, divided by the standard deviation of the image background.
- p.sdThresh is the signal stability threshold for objects, and removes traces with high signal fluctuations from the positive class, to avoid false positives. The parameter value defines the maximum allowed signal standard deviation during the assessment interval. Events with a mean marker intensity above the intensity threshold (*i.e.* marker positive) but a signal standard deviation over the stability threshold will be reclassified as marker negative.
- p.firstFrame, p.lastFrame and p.minVals determines the frames to be used for event classification (assessment interval), and the minimum number of measurements needed to be evaluated, respectively.
- Select one or several of the plot types Pie, Dotplot, Cutoff, Boxplot, and Heat (heatmap) to create, by changing the parameter value from 0 to 1.
- Provide the full path to the main QuantEscape folder as the p.home variable.

WALK-THROUGH: Step 17

Additional example datasets (collected tracking data) are provided for demonstrating the post-processing of endosome tracking data, structured in folders in QuantEscape > Post-processing > Data repository. Open the postprocessEndosomes.m script in the Post-processing > Scripts folder. Use all default values for parameters, and select the type of plot to generate. Provide the full path to the QuantEscape main folder as the p.home variable, and execute the script by clicking Run in the MATLAB Editor menu. Figures with LAMP1 and Peroxisome tracking data will be generated for LLOMe, Chloroquine and Siramesine, and the associated data is available in the MATLAB workspace.

Appendix 1. Chromatic aberration modeling

To investigate and map the chromatic aberration of the microscope, subdiffraction limit fluorescent beads with multiple excitation/emission wavelengths can be used. We have found Tetraspeck™ 0.1 μm Microsphere beads (Thermo Fisher Scientific) useful for this application, but several other options exist. This method is to correct for lateral (*x,y*) abberation. In our experience, axial (*z*) aberration was less significant at focus depths typical for monolayer cell samples – especially when considering the lower axial resolution. Scripts for chromatic aberration modeling are located in QuantEscape > Processing tools > Chromatic aberration odeling.

Note that the beads are typically sensitive to photobleaching and should be protected from ambient light as much as possible.

1. Dilute beads 1:100, 1:1,000 and 1:10,000 in ethanol.
2. Using a pair of tweezers, hold on to a No. 1.5 cover slip, submerge it in 70% ethanol and swiftly move the glas through the flames of a bunsen burner. This step will ensure that dust that often present on the glass will be removed.
3. Depending on the coverslip size, pipett ~4 μL diluted beads on the glass.
4. Place the glass with the bead side up on a heatblock at ~90°C for 5 min, to allow the beads to better adhere to the glass.
5. Clean three microscopy glass slides using a Kimtech wipe.
6. Pipett ~10 μL 100% glycerol on the cleaned microscope glass slides.
7. Place the cover glass on the glass slide bead side down, and allow for the glycerol to spread.
8. Place the prepared slide in the microscope, with the cover glass facing the objective lens. Use the same objective, immersion medium, illumination and filter settings to match the data for which the chromatic correction should be applied.
9. Acquire ≥3 *z*-stacks starting ~2 μm below and above the planes where the signal from the beads is perceptible.
10. Export data as uncompressed raw image tiff-files, all in one folder.

11. Open the prepBeadImages.m MATLAB script. Enter the dimensions of the image stack and select a suitable variable name for the image channel. Provide a identifier for the channel to be read, *e.g.* c1 or similar. Set the MATLAB working directory to the folder containing the images. Note that this processing step can require a significant amount of RAM, depending on the dimensions of the dataset and number of channels.

12. Execute the script, and save the variable as a mat-file.

13. Repeat for all channels in the image stack.

14. Open the processBeads.m script and setup the variable names to match the ones chosen previously. Run the first section of the script, where the bead segmentation is performed.

15. Run the second section. The script will save the fitted linear model for the beads of the stack in the mat-file linmdl.mat. Additional files saved are beadData.mat, that can be used to concatenate data from several stacks, and mdlData.mat, that contains all the data used for the linear fitting.

16. For use with the QuantEscape App, the linear correction model should by copied to QuantEscape > Processing tools > Chromatic aberration correction model.