

# Documentation for photoActivationAnalysis: tracking spatiotemporal evolution of photoactivated proteins in the ER

E. F. Koslover

Last updated May 29, 2020

The code in this package runs on Matlab and is used to analyze signal traces at different positions around a photoactivated region.

## 1 Requirements

- Matlab software and image processing toolbox. The code has been tested on Matlab R2018b.
- Image stacks. Each stack should be a separate (single-channel) .tif file. Currently the code assumes that for each image movie you have:
  - One stack containing frames prior to activation, with photoactivated signal channel.
  - One stack containing frames during activation, with photoactivated signal channel.
  - One stack containing frames prior to activation, with constitutive ER marker.
  - One stack containing frames during activation, with constitutive ER marker
  - One stack containing frames during activation, showing the photoactivated region only.

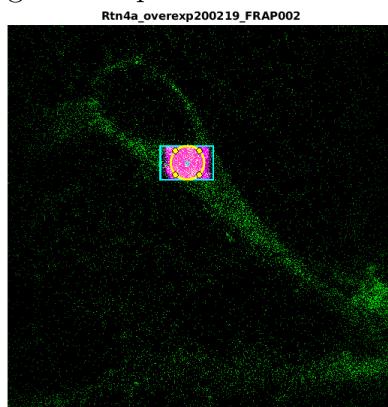
## 2 Workflow

1. Load a .lif file in Fiji (ImageJ). Select all "pre" and "bleach" series of interest. Select "split channels" and "Display metadata"
2. Save all loaded channels to individual .tif files. Also save the metadata as a .csv
3. Process the resulting images to load general information about the cell, set up ROIs etc. (Section 3). Save to .mat files.

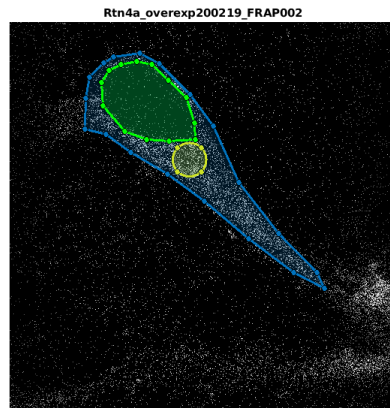
4. Fit a single exponential rise to each wedge-shaped ROI. Average all ROIs a certain distance away from the center of the activated region. If starting from preloaded data `allcells_*.mat`, just do this step.

### 3 Processing images to make cell objects

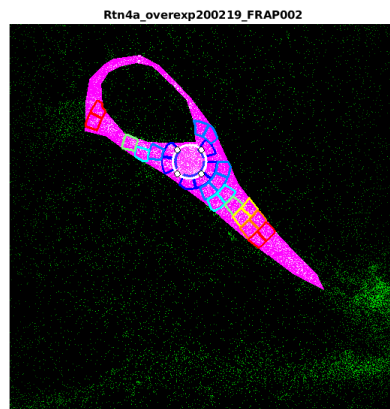
1. Open `processManyCells.m` and step through it in Matlab's cell mode (hit ctrl-enter to run each chunk of code).
2. You will need to modify directory and file names to match your system.
3. The activation region is detected automatically (only one region is detected in a movie). If a movie has more than one cell to be analyzed, you will need to run it through multiple times and set the "manualrect" flag to 1 to pick the activation region manually.



4. You will need to manually select the outline of each cell and its nucleus. Click on the image to select a polygonal region, then drag the control points to adjust, then hit enter when done.

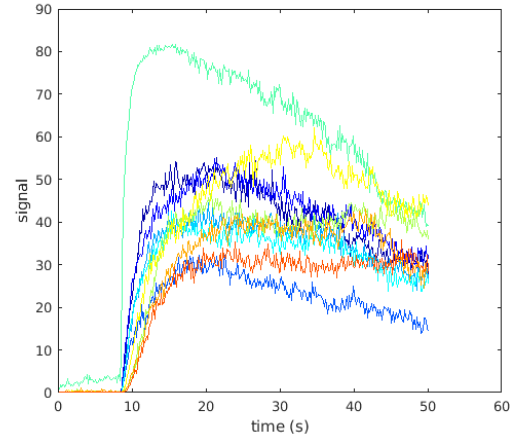
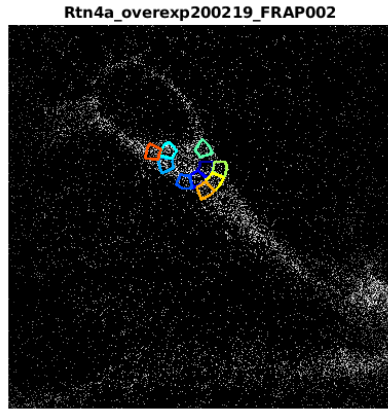


5. The code will automatically pick out wedge and ring regions. Look through the comments for parameters to alter (eg: how wide to make the rings and wedges).



6. The code then calculates the signal (per px) over time in each wedge and ring ROI and saves the result in `CL.ROIs.avgsignal`.
7. To visualize (*eg*) the first ten ROIs and their signals do:

```
CL.showROI(1:10)
```



8. The cell object is saved to a .mat file. Only one image is included (CL.ERing) to allow the cell to be visualized while keeping file size small.

## 4 Get average half-time for all regions at the same radius

1. Load previously processed cell objects (with all regions defined).
  - .mat files provided: `allcells_ctrl.mat` has control cells and `allcells_RTN40E.mat` has Rtn4a overexpression cells. These were obtained as described in Section 3.
2. Open the file `exploreHalfTimes.m` and step through the sections of code.
3. Important parameter: 'maxtscl'. All regions where the fitted half-time is above this factor times the length of the movie are excluded from the average.

