DotAnalysis 1.0 User Guide

Step-by-step instructions on using Mathematica notebook `DotAnalysis.nb`

Mark E. Fornace,[1] Maayan Schwarzkopf,[2] Harry M.T. Choi,[2] and Niles A. Pierce[2,3]

dHCR imaging enables digital mRNA absolute quantitation via single-molecule imaging in thick autofluorescent samples (Choi *et al.*, 2018). This User Guide provides step-by-step instructions on using the Mathematica notebook `DotAnalysis.nb` to perform dot detection and dot colocalization for dHCR images. For technical support, please email `support@moleculartechnologies.org`.

Overview of Dot Detection and Dot Colocalization Algorithms

Following Section S1.4.6 of Choi *et al.* (2018), consider a 2-channel redundant detection experiment in which a target mRNA is detected using two independent probe sets and HCR amplifiers (see Figure 1 for an example). All images displayed in the User Guide figures and in the Mathematica notebook are maximum intensity projections over all focal planes.

Let N_1 denote the number of dots detected in channel 1, N_2 the number of dots detected in channel 2, and N_{12} the number of colocalized dots appearing in both channels. We define the colocalization fraction for each channel:

$$C_1 = N_{12}/N_1,$$
$$C_2 = N_{12}/N_2.$$

As the false-positive and false-negative rates for single-molecule detection go to zero, C_1 and C_2 will both approach 1 from below, providing a quantitative basis for evaluating performance.

For Choi *et al.* (2018), single molecules were identified in each channel using the following dot detection algorithm applied to a three-dimensional confocal image stack:

- **Step 1: Blur noise.** To remove noise smaller than the dots of interest, the image was convolved with an isotropic Gaussian blur (standard deviation σ_{blur}). See Figure 2 for an example.
- **Step 2: Local background subtraction.** To eliminate variations in pixel intensity arising from background variations that occur on a length scale larger than the dots of interest, local background subtraction was performed by subtracting the mean pixel intensity of a cube (edge length d_{back}) from the intensity of the pixel at the center of the cube. See Figure 3 for an example.
- **Step 3: Global threshold on pixel intensity.** To eliminate dim features, the resulting pixel intensities were subjected to a global threshold ($t_{\mathrm{pixel}}^{\mathrm{min}}$), and the range [$t_{\mathrm{pixel}}^{\mathrm{min}}$,1] was renormalized to a [0,1] scale. See Figure 4 for an example.
- **Step 4: Watershed dot detection.** To identify single mRNA molecules as dots within the image, regional image maxima were segmented using the minimum salience watershed method (Couprie & Bertrand, 1997; Yoo *et al.*, 2002; Cates *et al.*, 2005) with threshold $t_{\mathrm{watershed}}$. Dot coordinates were calculated as the intensity-weighted centroid of each watershed region, and dot intensities were calculated as the total integrated pixel intensity within each region. See Figure 5 for an example and Figure 6 for a summary of the watershed method.

[1]Division of Chemistry & Chemical Engineering, California Institute of Technology, Pasadena, CA 91125, USA. [2]Division of Biology & Biological Engineering, California Institute of Technology, Pasadena, CA 91125, USA. [3]Division of Engineering & Applied Science, California Institute of Technology, Pasadena, CA 91125, USA.

- **Step 5: Global threshold on dot intensity.** To eliminate dim dots, the resulting dot intensities were normalized on a [0,1] scale and a global threshold (t_{dot}) was applied. The resulting number of dots N_i was recorded for channel i. See Figure 7 for an example.

After identifying the dots in each channel of a 2-channel redundant detection image, a dot i in Channel 1 and a dot j in Channel 2 were considered colocalized if all of the following four statements were true:

- **Test 1:** The xy centroids differed by less than a lateral distance threshold ($r_{xy} = 0.22$ μm).
- **Test 2:** Dot i is the closest dot in Channel 1 to dot j in Channel 2.
- **Test 3:** Dot j is the closest dot in Channel 2 to dot i in Channel 1.
- **Test 4:** The z centroids differed by less than the axial distance threshold ($r_z = 0.42$ μm).

Note that due to the lower axial resolution, dot colocalization was tested separately for xy and z. See Figure 8 for an example.

Run an Example Dot Analysis Job

To run a dot analysis job on sample data from the paper:

- Install Mathematica 11.0 or higher. Previous versions are untested but may work.
- Double-click on the notebook file `DotAnalysis.nb` to open it in Mathematica.
- Within the notebook click the link: "Click here to run complete analysis".
  - Click "yes" for the popup asking if initialization cells should be run.
  - Select the parameter file `BRAF.json` using the file browser.
  - As the notebook runs, results and graphics will appear.
- Click the link: "Export final EPS images and JSON" near the bottom of the notebook. Enter a prefix for the files names. For example, a prefix of `test` will yield files: `test-ch1.eps`, `test-ch2.eps`, `test-ch1ch2.eps`, and `test.json`. The image plots are formatted as in Figure 7 of Choi *et al.* (2018).
- Sample output files for this job can be found in the folder `sample-output` (including a pdf file showing the Mathematica notebook with all the results loaded).

Run a Dot Analysis Job

To run a dot analysis job on your own data:

- **Data acquisition**
  - Perform a 2-channel redundant detection dHCR experiment (Choi *et al.*, 2018). Save each channel as a 1-channel (grayscale) 3D tif file. For example: `BRAF-1.tif` and `BRAF-2.tif`.
- **Set job parameters**
  - Save a copy of the Example JSON file and specify parameters for your job:
    - *File paths:* image files for the two channels. For example: `[BRAF-1.tif, BRAF-2.tif]`
    - *Pixel dimensions* (μm): the x, y, and z dimensions of a single pixel (assumed to be identical for both channels). For example: `[0.0624, 0.0624, 0.42]`

* *Image scaling*: scale factors to upscale (> 1) or downscale (< 1) the images via linear interpolation. For example, setting `[0.5, 0.5, 0.5]` on an image of size (1024, 1024, 20) would run the analysis on a (512, 512, 20) image. Paper analyses used: `[1, 1, 1]` (which has no effect)
* *Flip image horizontally*: Flip image horizontally so that display convention matches ImageJ. Paper analyses used: `true`
* *Blur standard deviation* σ_{blur} (μm): Standard deviation for Gaussian blur to remove noise. Specified for each channel. Value should be above the length scale of pixel noise but below the length scale of the dots of interest (e.g., no larger than the diffraction limit). For example: `[0.2, 0.2]`
* *Blur extent* (μm): The truncation length of the convolution kernel used for the Gaussian blur. A value of 3-5 times the blur standard deviation is appropriate. Paper analyses used: `5`
* *Background subtraction length* d_{back} (μm): Edge length of cube used for background subtraction. Specified for each channel. Value should be above the length scale of the dots of interest but below the length scale of global variations in autofluorescent background. For example: `[2.5, 2.5]`
* *Background subtraction coefficient*: For a pixel with intensity g at the center of a background subtraction cube with mean intensity $\bar{g}$, the background-subtracted pixel intensity is $g - \alpha\bar{g}$, where α is the coefficient. Paper analyses used: `1`
* *Min pixel threshold* $t_{\text{pixel}}^{\text{min}} \in [0, 1]$): Minimum threshold applied to all pixel intensities in the mean-subtracted image. Specified for each channel. Example: `[0.003, 0.012]`
* *Max pixel threshold* $t_{\text{pixel}}^{\text{max}} \in [0, 1]$: Maximum threshold applied to all pixel intensities in the mean-subtracted image. Specified for each channel. Paper analysis used: `[1, 1]` (which has no effect)
* *Watershed threshold* $t_{\text{watershed}} \in [0, 1]$): See Figure 6 for a summary of the watershed method. Specified for each channel. Example: `[0.15, 0.15]`
* *Dot intensity definition (1: total, 2: mean, 3: max, 4: median)*: The approach used to define the intensity of each dot. Paper analyses used: `1` (denoting total intensity)
* *Dot intensity threshold* $t_{\text{dot}} \in [0, 1]$: The threshold applied to all dots in each channel is t_{dot} times the intensity of the brightest dot in that channel. Specified for each channel. Example: `[0.01, 0.005]`
* *Coordinate offset* (μm): an (x, y, z) offset that is added to Channel 1 in order to align the two channels before performing the colocalization analysis. Paper analyses used `[0,0,0]` except for one image where the two channels were out of alignment.
* *Keypoint alignment*: use keypoint algorithm to align the two channel images (did not work well in our hands). Paper analyses used: `false`.
* *Colocalization distance thresholds* r_{xy}, r_z, r_{xyz} (μm): Lateral distance threshold r_{xy}, axial distance threshold r_z, and Euclidean distance threshold r_{xyz}. Some subset of these is used for checking whether dots are colocalized between the two channels. Example: `[0.22, 0.42, 0.22]`
* *Colocalization mode* (1: test r_{xy} only, 2: test r_{xy} and r_z, 3: test r_{xyz}): Approach 1: test if xy centroids differ by less than lateral distance threshold r_{xy}. Approach 2: test if xy centroids differ by less than lateral distance threshold r_{xy} and z centroids differ by less than axial distance threshold r_z. Approach 3: test if xyz centroids differ by less than the Euclidean distance threshold r_{xyz}. Paper analyses used: `2` (denoting test r_{xy} and r_z).
* *Dot counts* (N_1, N_2, N_{12}): The dot counts N_1 and N_2 are outputs of the dot detection analysis and the dot count N_{12} is an output of the dot colocalization analysis. These outputs are saved to the JSON file at the end of an analysis job and need not be included prior to running the job. Example: `[129, 136, 110]`
* *Colocalization fractions* (C_1, C_2): The colocalization fractions C_1 and C_2 are outputs of the dot detection and colocalization analysis. These outputs are saved to the JSON file at the end of an analysis job and need not be included prior to running the job. Example: `[0.85 , 0.81]`
* *Notes*: Any notes the user may wish to attach to the JSON file.

- **Run the job and save output**
  - Run the job and save files as described above for the Example.
  - Alternatively, advanced users can edit parameters and run jobs interactively as described below.

Run a Dot Analysis Job in Interactive Mode (Advanced Users)

- Open the file `DotAnalysis.nb` in Mathematica.
- Edit the JSON file as described above to specify initial job parameters.
  - If the image file paths are not set in the JSON (or are invalid), select the two tifs using the file browser when prompted (choose 1 file at a time).
- It is easiest to first run the notebook non-interactively as described above for the Example, then change parameters afterwards.
- Select a cell by clicking the arrowed vertical line to the right of the cell (see Figure 9). To run a selected cell press `shift-return`. Alternatively, to run a selected cell and progress to the next cell press `shift-enter` (where `enter` is *on the number keypad*).
  - If running multiple notebooks concurrently, you should check that `Evaluation→Notebook's Default Context` is set to `Unique to This Notebook`.
  - The title bar of the notebook window will display (`Running...`) if any computations are being run. It is easiest to wait until this is not the case before continuing to the next cell.
- Using the the manipulation GUIs provided, follow the instructions for each cell to modify parameters where appropriate.
  - The display of parameters under section "Load parameters and images" should update as any parameters change.
  - Once all parameters are as desired, run any cells remaining in the notebook.
- `DotAnalysis.nb` is a notebook with control flow running from top to bottom. *If you make interactive updates in a given cell, every cell below that will have to be rerun to get valid results.*
  - You can double-check your results by rerunning the whole notebook via `Evaluation→Evaluate Notebook` from the Mathematica menu (any parameter values that you set interactively will be used).
  - You can triple-check your results by reopening Mathematica and running the notebook non-interactively using the new JSON file.
- Once parameters and results are as desired, you can click the "Save JSON" button to save the JSON results without images. The saved parameters should match the provided inputs if no parameters were modified interactively.
- To save figures, click the "Export final EPS images and JSON" button towards the bottom of the notebook and specify a file prefix as described in the Example above.
- Save the notebook if desired. We recommend saving a new notebook if running multiple analyses. Mathematica can be slow to reopen notebooks pre-loaded with images.
- To delete all output, select `Cell→Delete all output` (deleting output makes saving and loading faster since images are not stored in the notebook). To quit the session and clear all variables, select `Evaluation→Quit Kernel→Local`.

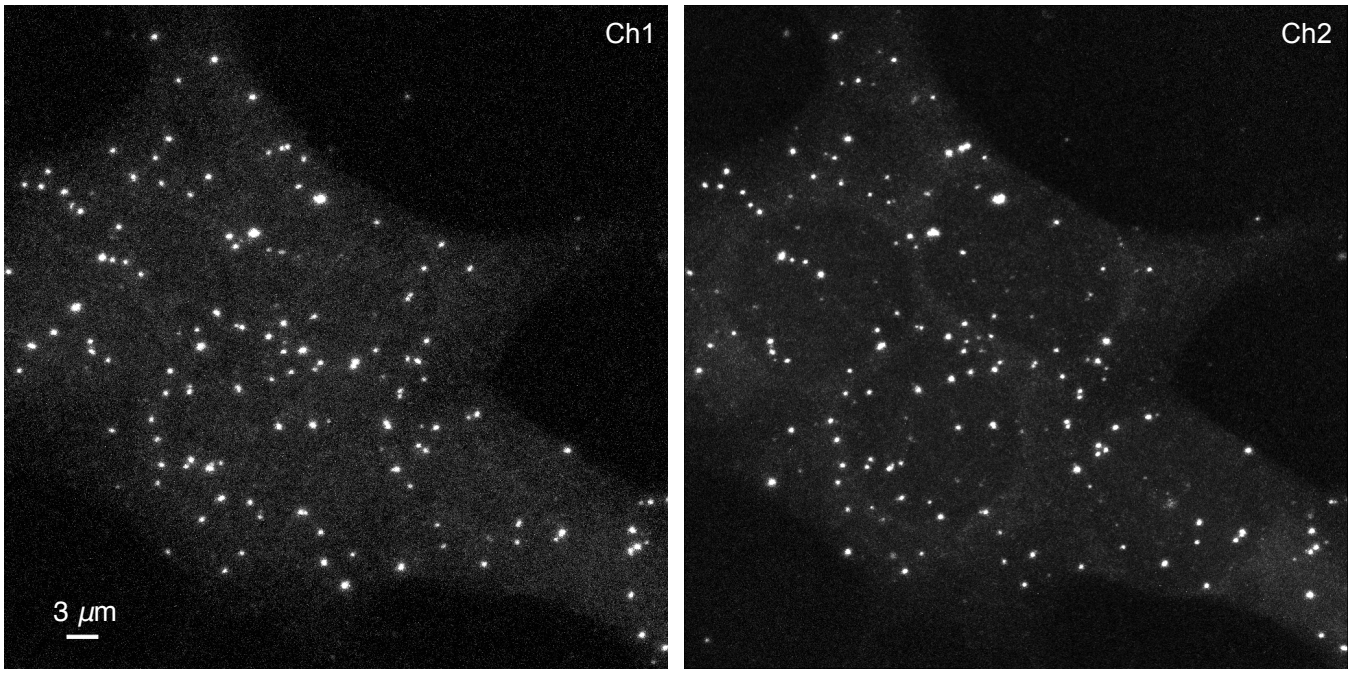

Figure 1. Raw images for a 2-channel dHCR redundant detection experiment. Maximum intensity projection in the axial direction over 7.14 μm (17 focal planes). Target mRNA: *BRAF*. Sample: HEK cells. Pixel size: $0.0624 \times 0.0624 \times 0.42$ μm. Corresponds to Replicate 1 of Figure S29 and Table S25 in Choi *et al.* (2018).

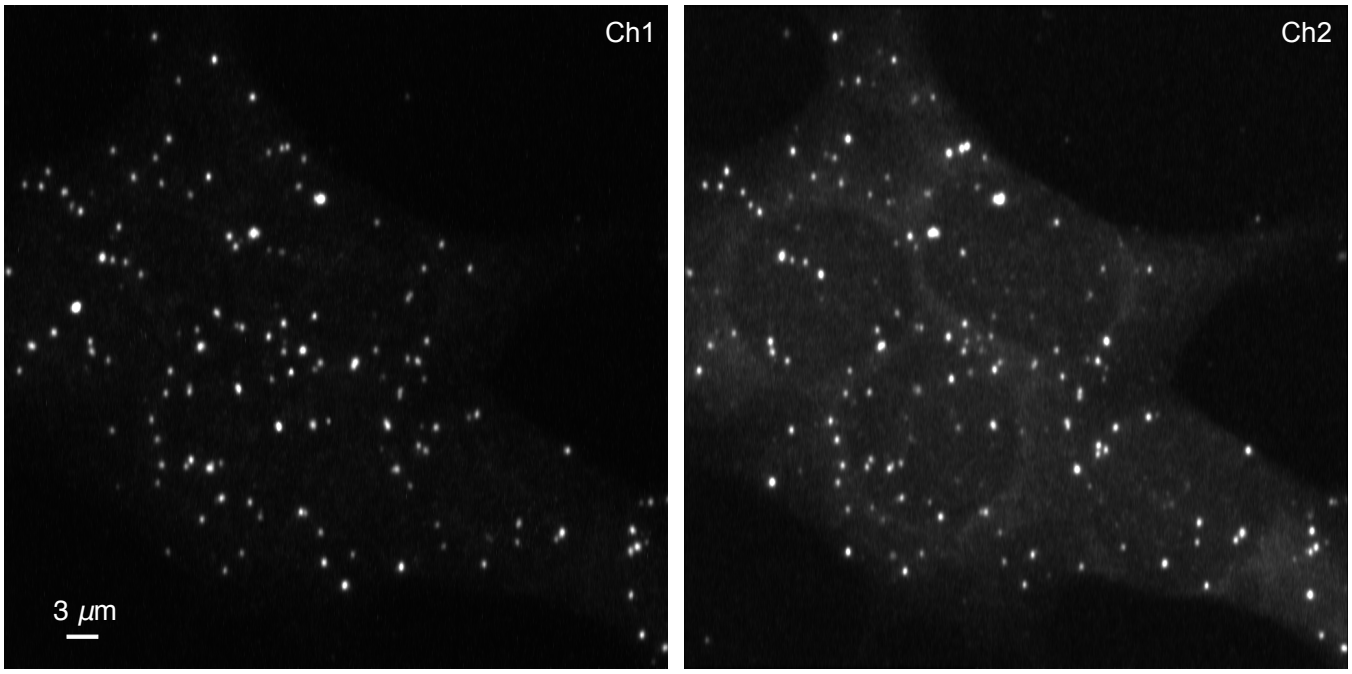

Figure 2. Blur noise. Images after blurring noise with an isotropic Gaussian blur. The blur standard deviation should be above the length scale of pixel noise but below the length scale of the dots of interest (e.g., no larger than the diffraction limit). Blur standard deviation σ_{blur} : 0.2 μm for both channels.

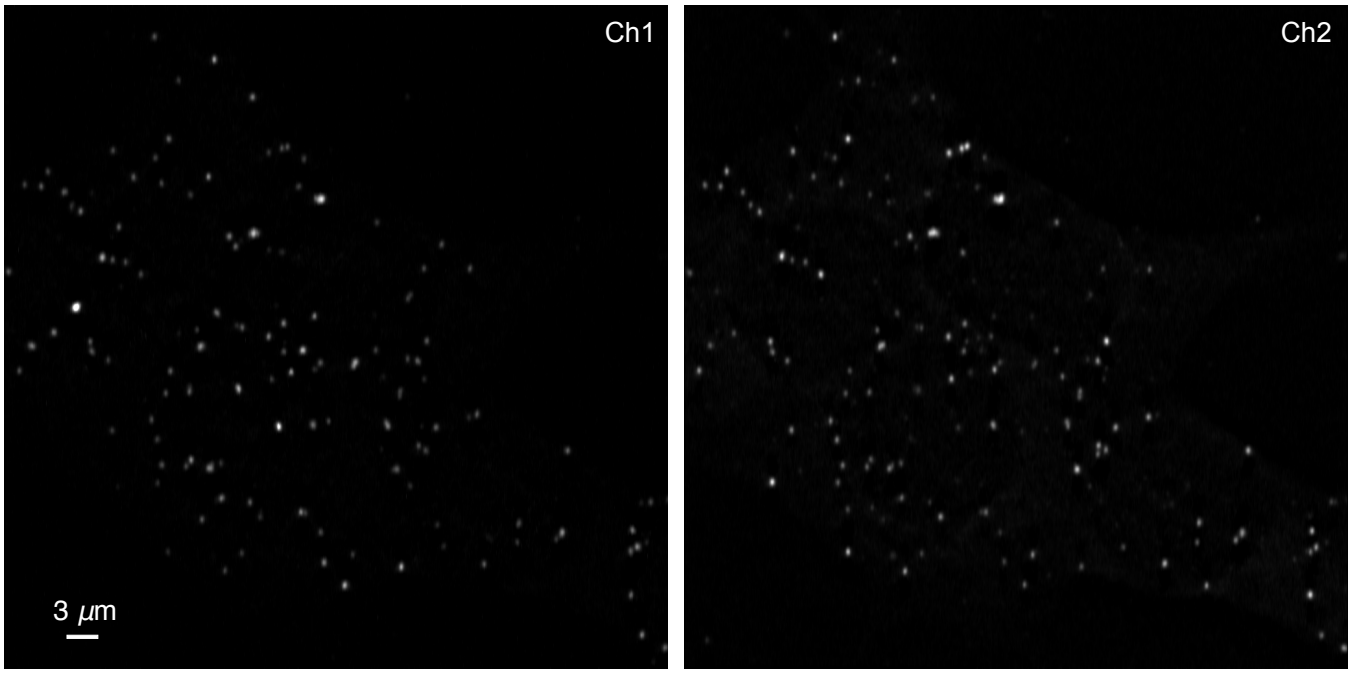

Figure 3. Local background subtraction. Images after background subtraction to remove large-scale variations in background level across the image. The background subtraction length should be above the length scale of the dots of interest but below the length scale of global variations in autofluorescent background. Background subtraction length d_{back}: 2.5 μm for both channels.

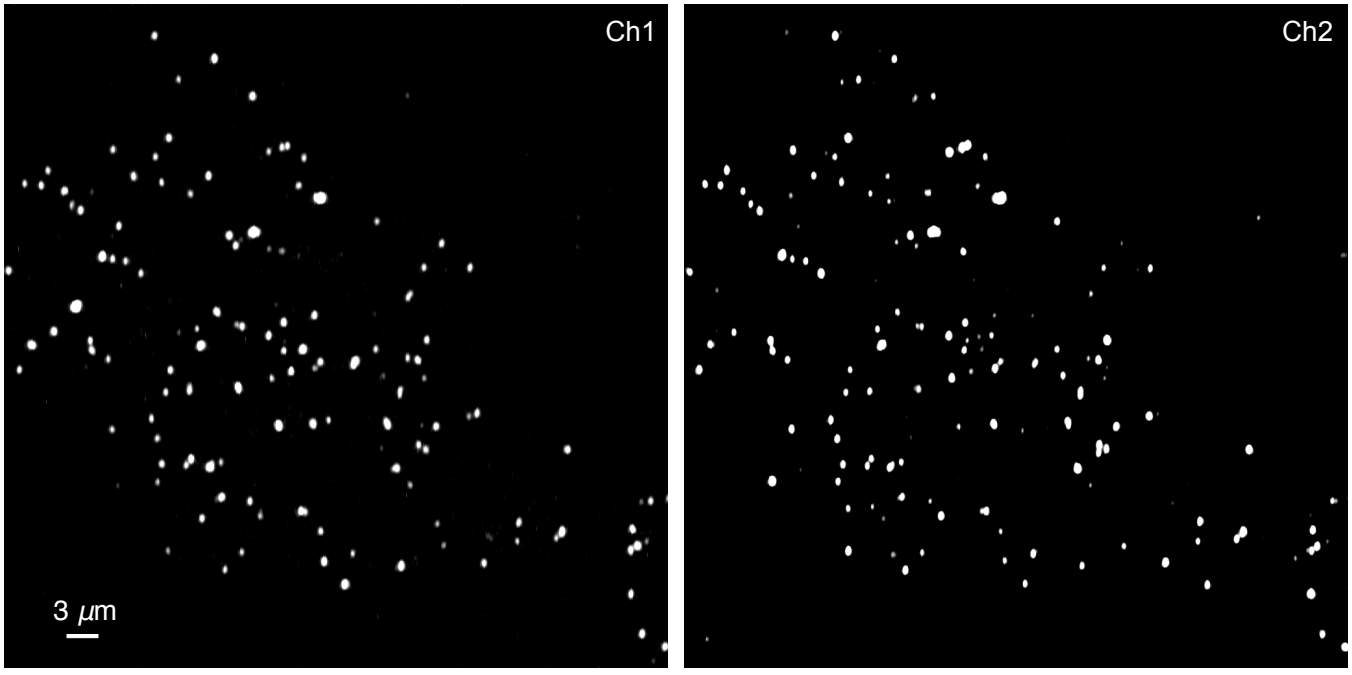

Figure 4. Global threshold on pixel intensity. Images after thresholding to remove dim features. The minimum pixel threshold should be lower than the intensity of pixels within the dots of interest. The maximum pixel threshold should be brighter than the intensity of pixels within the dots of interest (we did not use this parameter for the paper). Minimum pixel threshold $t_{\mathrm{pixel}}^{\mathrm{min}}$: 0.003 for Ch1, 0.012 for Ch2. Maximum pixel threshold $t_{\mathrm{pixel}}^{\mathrm{max}}$: 1 for both channels (which has no effect). After thresholding, the pixel intensities for each channel are renormalized to a [0,1] scale.

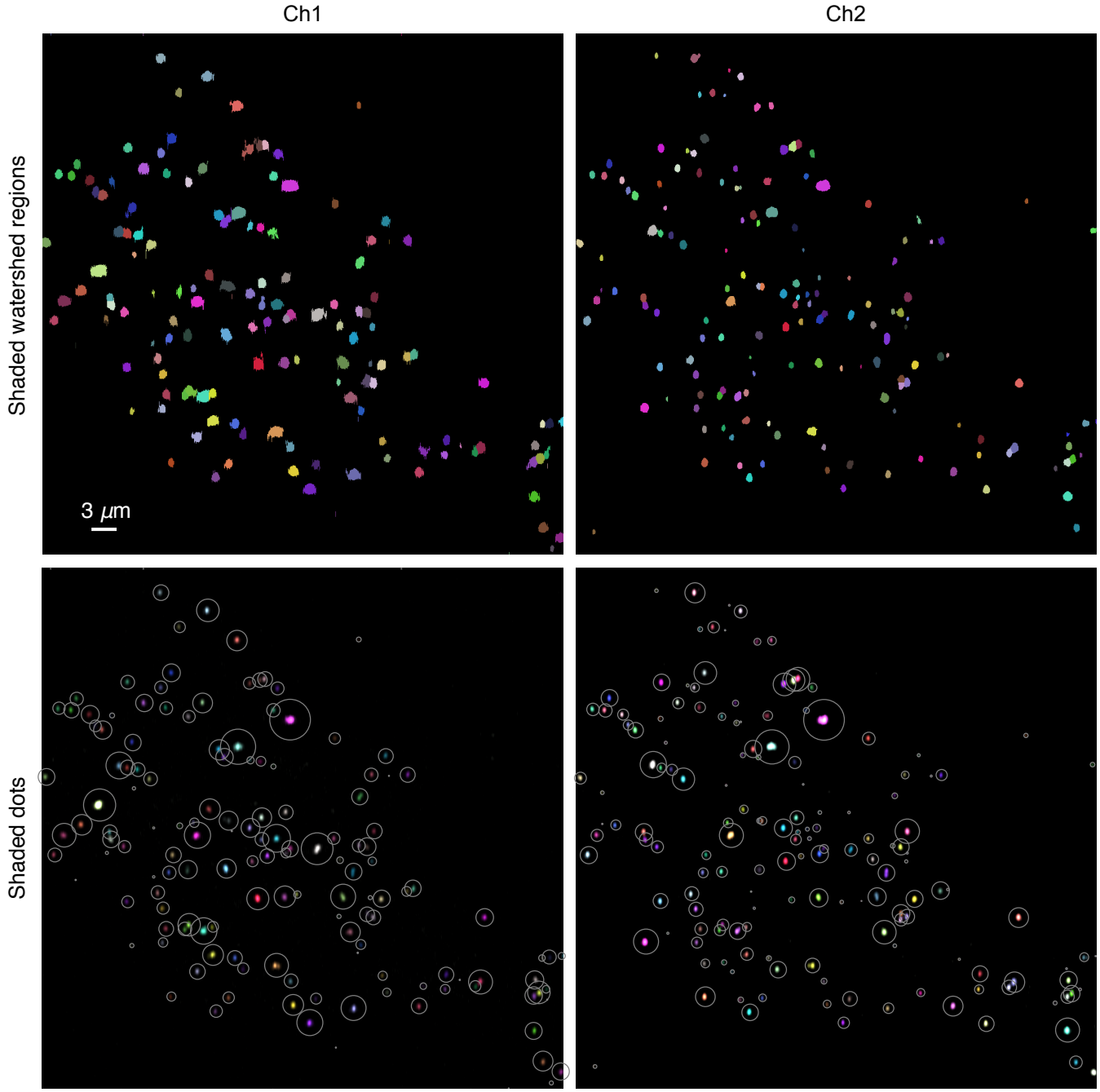

Figure 5. Watershed dot detection. Images after segmenting regional maxima using the minimum salience watershed method (Couprie & Bertrand, 1997; Yoo *et al.*, 2002; Cates *et al.*, 2005). Top: Shaded watershed regions (interpreted as one dot per region). Bottom: shaded individual dots with centroids depicted as circles with area proportional to total dot intensity (bottom). Watershed threshold $t_{\mathrm{watershed}}$: 0.15 for both channels. Dot coordinates are calculated as the intensity-weighted centroid of each watershed region, and dot intensities are calculated as the total integrated pixel intensity within each region. After dot identification, dot intensities for each channel are normalized to a [0,1] scale. See Figure 6 for a summary of the watershed method.

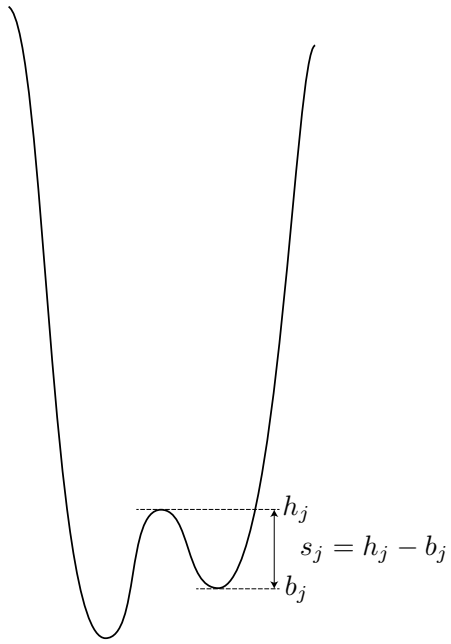

Figure 6. Summary of the minimum salience watershed method. The minimum salience watershed method (Couprie & Bertrand, 1997; Yoo *et al.*, 2002; Cates *et al.*, 2005) segments regional minima in an image by analogy to water flowing downhill within cachement basins, and then merges basins separated by low saddle points to avoid oversegmentation (e.g., to avoid segmenting single dots into multiple basins). For convenience, since we are interested in finding regional *maxima* (bright dots in the image), we apply the watershed method to an inverted image by replacing each normalized pixel intensity $g \in [0, 1]$ with $1-g$. Each pixel is assigned to a cachement basin B_j by following the path of steepest descent to the local minimum b_j within B_j. The *salience* $s_j = h_j - b_j$ is calculated for each basin B_j corresponding to the difference between the lowest saddle point h_j bordering another basin and the basin minimum b_j. Basin merging is performed by merging the minimum salience basin with its neighbor until all basins have salience above threshold $t_{\text{watershed}} s_{\max}$, where $s_{\max} = \max_j s_j$ is the maximum salience over all basins in the image. Choosing $t_{\text{watershed}} = 0$ corresponds to merging no dots, and choosing $t_{\text{watershed}} = 1$ corresponds to merging all dots into a single dot.

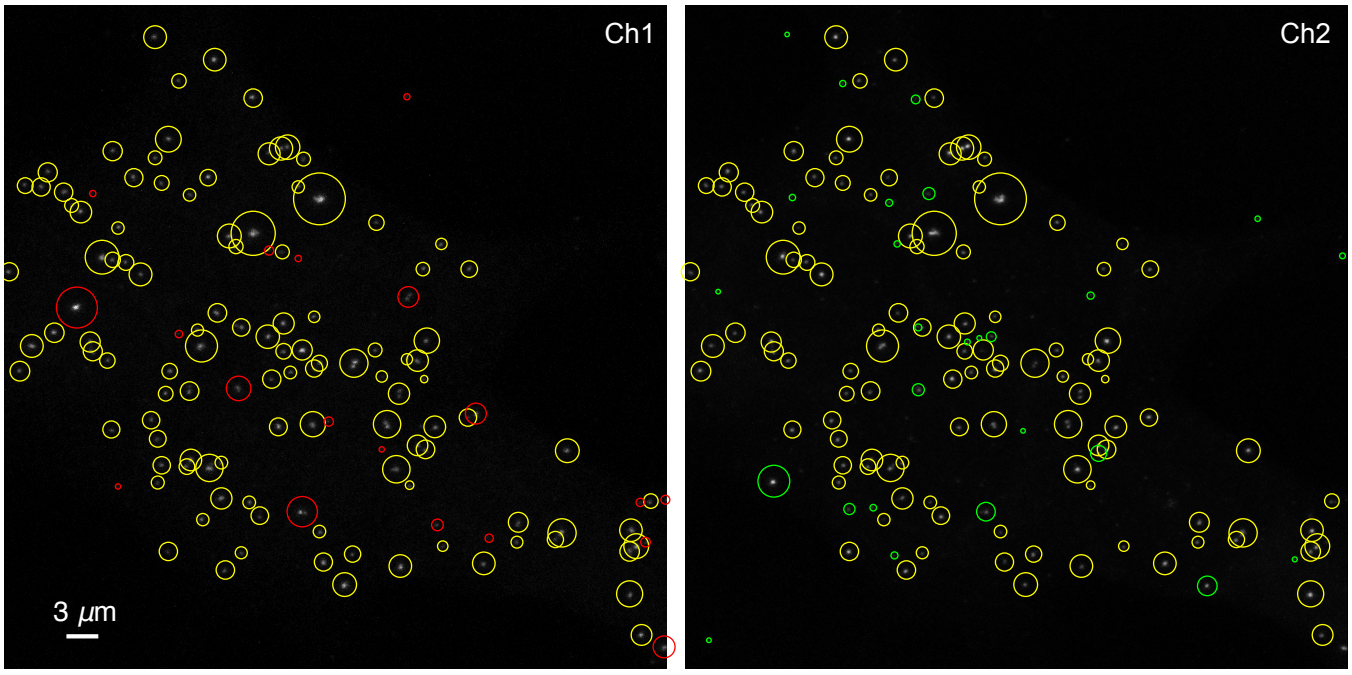

Figure 7. Global threshold on dot intensity. Images after thresholding to remove dim dots. Increasing the threshold may cause false negatives in a given channel; decreasing the threshold may cause false positives in a given channel. Dot intensity threshold t_{dot}: 0.01 for Ch1, 0.005 for Ch2. The resulting number of dots for each channel were $N_1 = 129$, $N_2 = 136$.

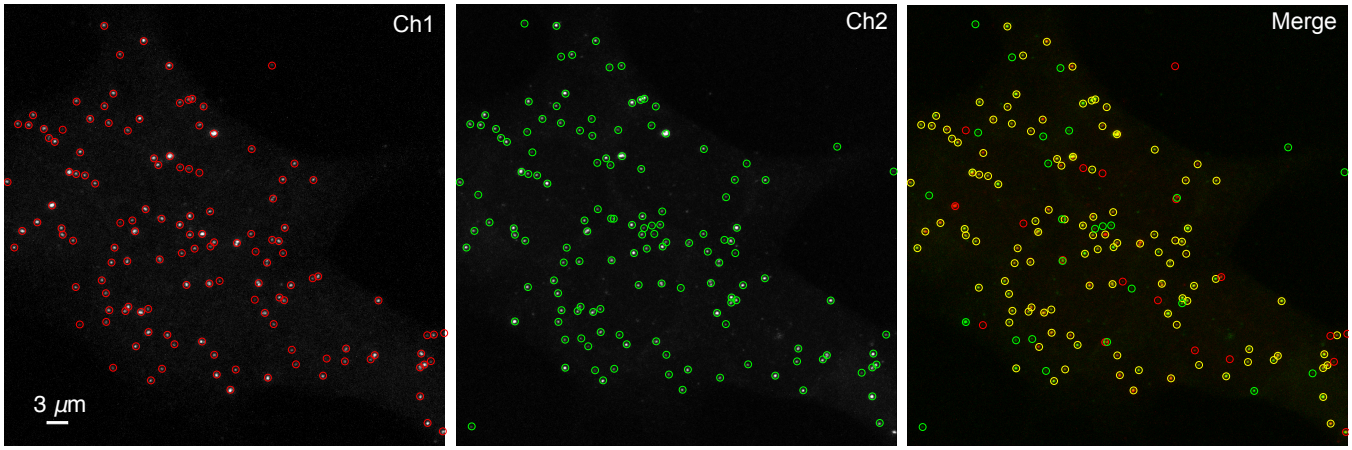

Figure 8. Dot colocalization. Images depicting dot colocalization between channels. Left: Dots in Ch1 image (red). Center: Dots in Ch2 image (green). Right: dots colocalized between channels (yellow). Dot counts: $N_1 = 129, N_2 = 136, N_{12} = 110$. As the false-positive and false-negative rates for single-molecule detection go to zero, C_1 and C_2 will both approach 1 from below, providing a quantitative basis for evaluating performance. Colocalization fractions: $C_1 = 0.85, C_2 = 0.81$. Colocalization distance thresholds: lateral distance threshold $r_{xy} = 0.22$ μm, axial distance threshold $r_z = 0.42$ μm. Colocalization thresholds are applied to the distance between the intensity-weighted centroids of dots that are mutual nearest neighbors between the two channels. These threshold distances should be as small as possible but large enough that dots clearly present in both channels have centroids closer than the thresholds. Lateral resolution and lateral centroid accuracy are better than axial resolution and axial centroid accuracy so we test the xy distances separately from the z distances so that needed leniency in the z colocalization test (due to lower axial resolution) does not undermine the stringency of the xy colocalization test.

Figure 9. Running a cell in Mathematica. Click the arrowed vertical line to the right of the cell (left) and press `shift-return` to see the output (right). Alternatively, to run a selected cell and progress to the next cell press `shift-enter` (where `enter` is *on the number keypad*).

References

Cates, J. E., Whitaker, R. T., & Jones, G. M. (2005). Case study: an evaluation of user-assisted hierarchical watershed segmentation. *Med. Image Anal.*, **9**(6), 566–578.

Choi, H. M. T., Schwarzkopf, M., Fornace, M. E., Acharya, A., Artavanis, G., Stegmaier, J., Cunha, A., & Pierce, N. A. (2018). Third-generation in situ hybridization chain reaction: multiplexed, quantitative, sensitive, versatile, robust. *Development*, **145**, dev165753.

Couprie, M., & Bertrand, G. (1997). Topological gray-scale watershed transform. *Pages 136–146 of: Vision geometry VI*. Proceedings of SPIE, vol. 3168.

Yoo, T.S., Ackerman, M.J., Lorensen, W.E., Schroeder, W., Chalana, V., Aylward, S., Metaxas, D., & Whitaker, R. (2002). Engineering and algorithm design for an image processing API: a technical report on ITK - the Insight Toolkit. *Stud. Health Technol. Inform.*, **85**, 586–592.