Graphical User Interface for Semi-Automatic Spot Detection and Colocalization

Johannes Stegmaier[1,2], Maayan Scharzkopf[3], Harry M. Choi[3], Alexandre Cunha[2]

[1]Institute for Applied Computer Science, Karlsruhe Institute of Technology, Karlsruhe, Germany; [2]Center for Advanced Methods in Biological Image Analysis (CAMBIA), California Institute of Technology, Pasadena, CA, USA; [3]Division of Biology & Biological Engineering, California Institute of Technology, Pasadena, CA, USA;

Contents

1 Quick Start Guide

1.1 Installation Instruction

1. Download the latest software package and extract the zip-archive to your hard disk drive. Note that the file path you extract the files to should not contain any spaces or special characters and you'll need to have write permissions in order to temporarily store processed data.

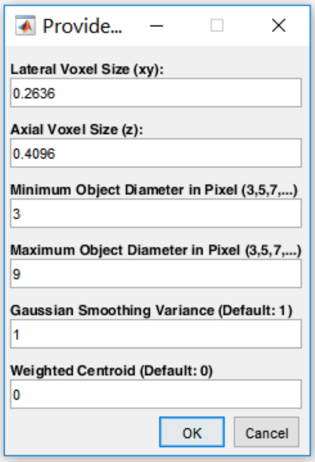
2. In order to run the spot colocalization software, you'll need to install MATLAB (we tested the software on MATLAB R2017a on Windows, Ubuntu and Mac OSX). Open up the file "SpotDetectionGUI.m" in the MATLAB editor and execute the script with the "F5" button or by pressing the green play button on top of the MATLAB code editor window.

3. In case you observe permission denied errors, try changing the permissions to the extracted software folder, such that read/write/execute are enabled. On Unix-based systems this can be performed by navigating to the respective folder (called $DIR in the following command) using the Terminal application and by executing the following command "chmod 755 -R $DIR". This should change the permissions to read+write+execute for the current user and to read+execute for all other users.

1.2 Example

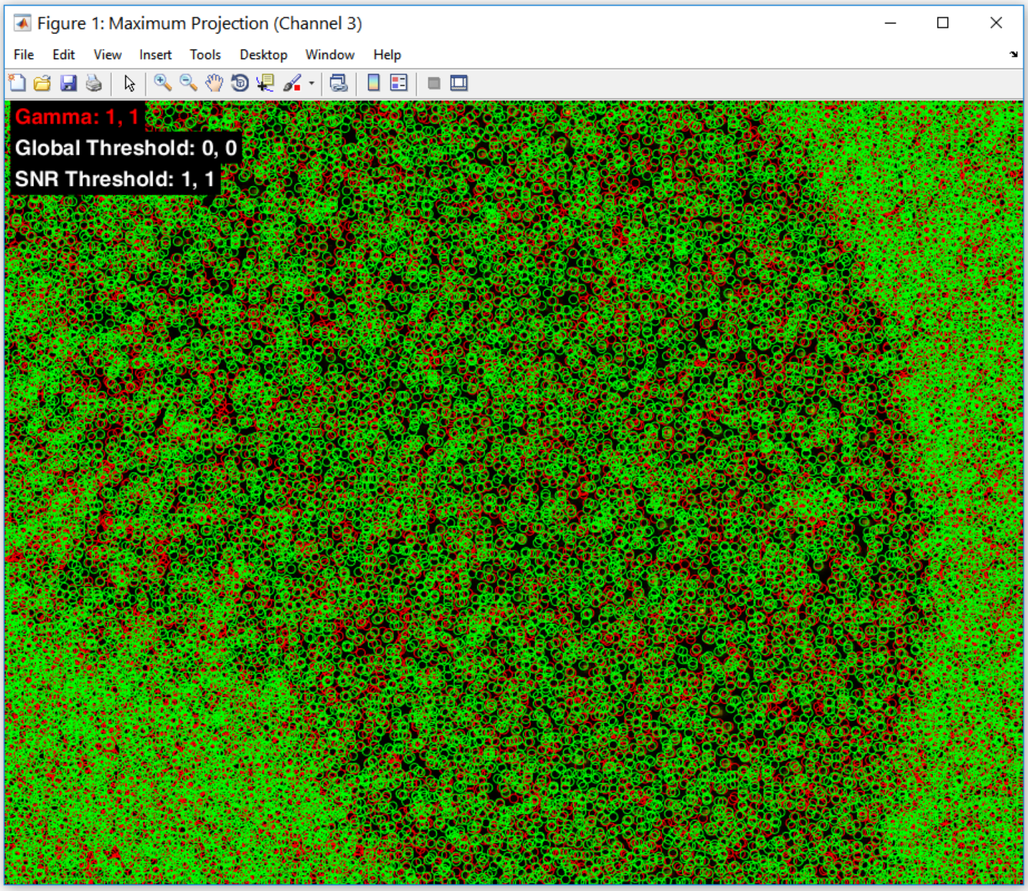
1. Open and run the “SpotDetectionGUI.m” in MATLAB as described above.

2. Adjust the project settings shown in the following dialog such that it matches your image data. This comprises the physical voxel size in microns and the minimum/maximum object diameter measured in pixel. For noise removal, the raw images are initially subtly smoothed with a Gaussian filter. For noisy images, it may help to increase the variance of the Gaussian filter for further suppression of high frequency noise. Optionally, the centroids identified by the Laplacian-of-Gaussian blob detector can be refined if the “Weighted Centroid” option is set to 1 (disabled by default).

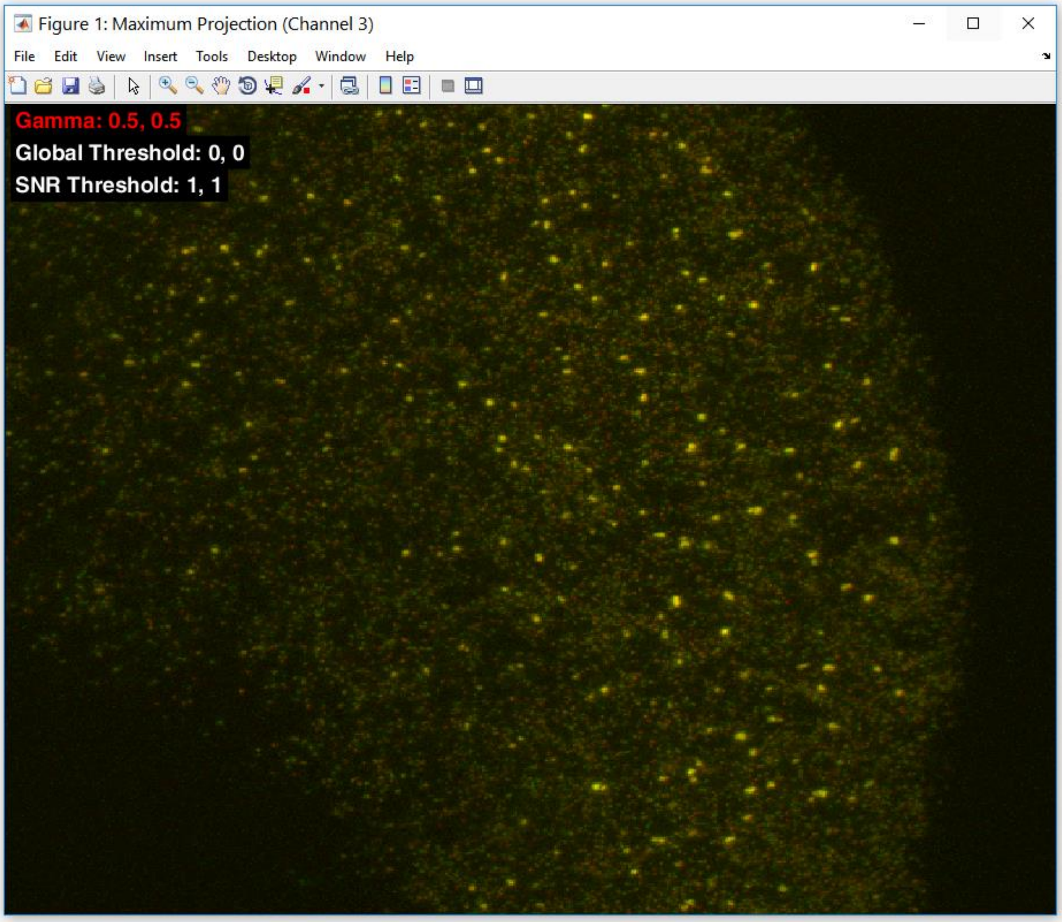


3. After confirming the project settings, three file selection dialogs will pop up. Select the two input images to use first (Note that the dimensions of the images need to be identical). In the third file selection dialog, select the output folder that will be used for saving the intermediate and the final results. Two example images can be found in the “Data” folder contained in the original zip archive of the software. Choose, e.g., the “Processing” folder also contained in the root directory of the software folder.

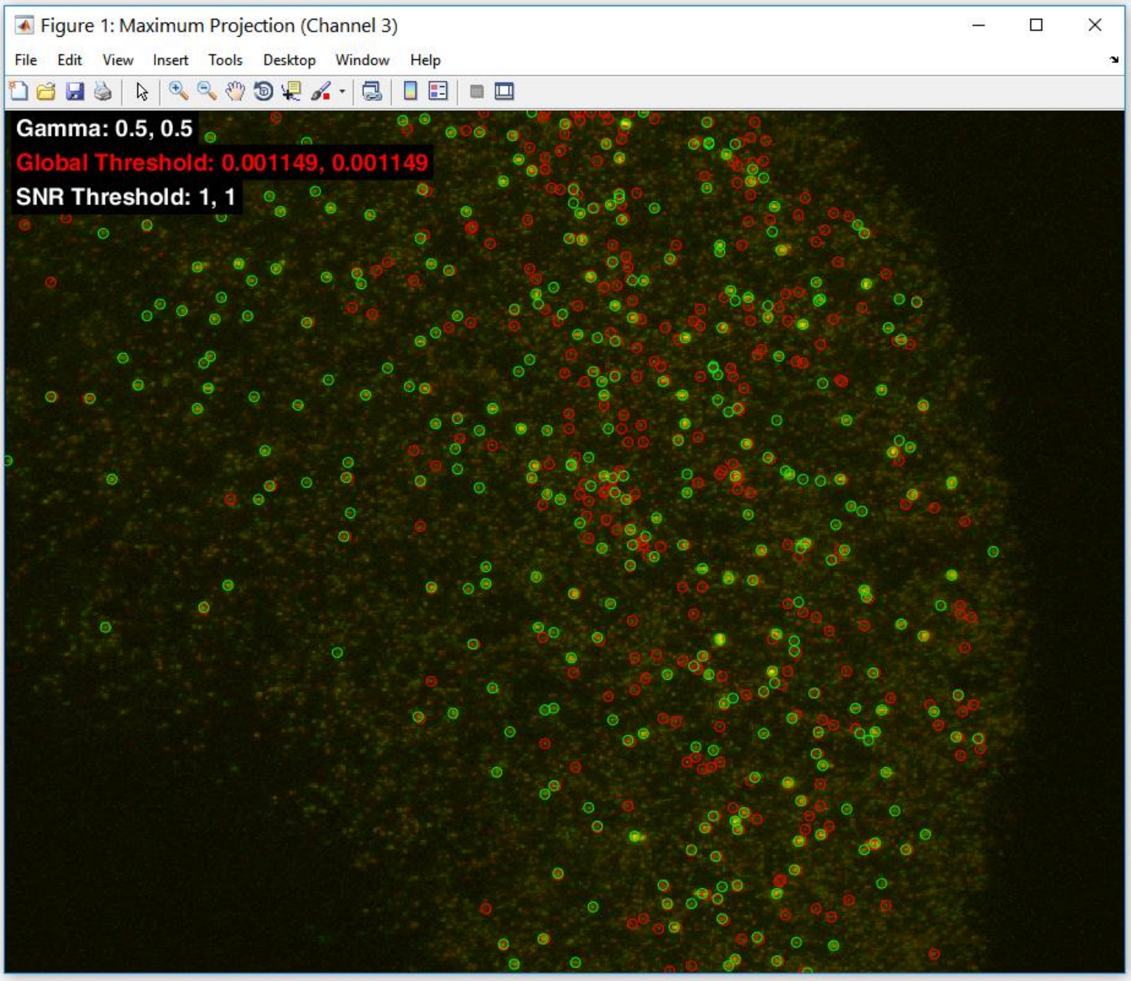
4. The images are automatically analyzed and the detected spots are imported to MATLAB for further analysis as shown in the following screenshot:



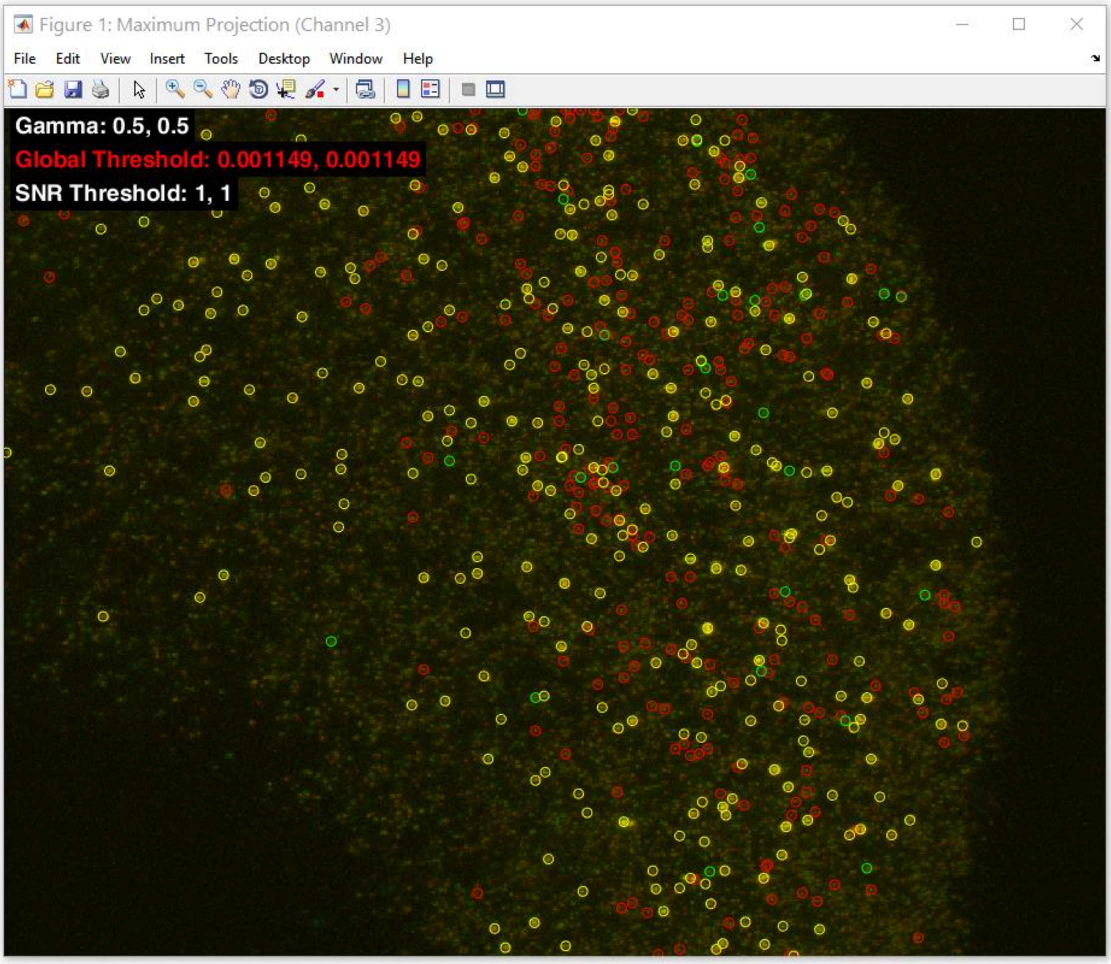
5. As a first step, it is usually advisable to adjust the contrast of the images in order to properly see the structures of interest. The buttons “1”, “2” or “3” can be used to switch between channel 1, 2 and an overlay of both channels, respectively. Select “3” and press the “Arrow Down” button to decrease the gamma value used for contrast adjustment.



6. Now the thresholds for filtering the detections can be adjusted. Press the “T” button once to switch the threshold mode to the global threshold. By pressing the “Arrow Down” and “Arrow Up” button, the global intensity threshold can be decreased and increased, respectively. All detections with an average intensity lower than this threshold are discarded from the further analyses. You can again switch between channels 1 and 2 or use both channels using the “1”, “2” or “3” keys, in case you need to separately adjust the threshold differently for both channels. By pressing “T” again, you can switch to the SNR threshold mode and adjust this threshold to further refine the filtering of the remaining detections. The SNR threshold uses a neighborhood surrounding the detection and compares the mean intensity of the detected spot to the neighborhood region excluding the central part. The higher the threshold, the more prominent a detection has to be. For instance, a value of 2 indicates that the mean intensity of the detection has to be at least twice as high as the mean intensity of its local neighborhood.



7. Once the detections are properly filtered, the actual colocalization can be performed by pressing the “C” key. Colocalized detections are highlighted in yellow as shown on the following screenshot:



8. Finally, results can be exported by pressing the “E” key. The uncolocalized and colocalized detections are separately stored for each channel in a CSV file format that can be used for further processing, e.g., in Excel. Furthermore, a result overview file is generated in plain text format.

1.3 Keyboard Shortcuts

The following hot keys can be used to control the software (a summary also shows up when pressing the “H” button for help).

- **1,2,3:** Toggles the displayed image (channel1, channel2, overlay)
- **Up Arrow:** Increase selected threshold (highlighted in red)
- **Down Arrow**: Decrease selected threshold (highlighted in red)
- **Left Arrow:** Go to previous slice (only works in slice mode)
- **Right Arrow:** Go to next slice (only works in slice mode)
- **A:** Toggle between streched mode and fixed aspect ratio
- **B:** Toggle background detections for intensity comparisons (auto-detection uses convex hull of colocalized dots and freehand tool allows arbitary masks)
- **C:** Toggle visibility of colocalized detections
- **D:** Toggle visibility of detections
- **E:** Export results
- **H:** Show this help dialog
- **I:** Show IDs of detections by hovering them with the mouse
- **L:** Show ScaLeBar at a user-defined location (lower-right corner of the scale bar)
- **M:** Switch between slice-mode and maximum projection mode
- **O:** Zoom out to the original view
- **P:** Preview of the percentage of colocalization before the final parameters are fixed for export
- **R:** Reset all thresholds
- **S:** Show scatter plots of the mean intensities of the current colocalizing detections
- **T:** Change current threshold (use 1,2,3 for changing only channel1, channel2 or both parameters, respectively)
- **Mouse Wheel:** Scroll through slices (only works in slice-mode)
- **CTRL + Mouse Wheel:** Zoom in/out in a Google-Maps like behavior

Hint: In case key presses show no effect, left click once on the image and try hitting the button again. This only happens if the window loses the focus.

2. Methods

2.1 Spot Detection

The first processing step was to detect spots in the 3D images separately in both channels. We performed the spot detection using a multi-scale Laplacian-of-Gaussian (LoG) filter as described in (Stegmaier *et al.*, 2014). To suppress image noise, the images were Gaussian-filtered with a variance $\sigma^2 = 2$. The noise reduced 3D input images were then filtered using a 3D LoG filter with kernel sizes that matched the expected size of the spots of interest. Based on the relationship $r = \sqrt{2} \cdot \sigma$ of the spot radius r and the standard deviation σ of the LoG filter, we used the standard deviations $\{\frac{1}{\sqrt{2}}, \frac{2}{\sqrt{2}}, \frac{3}{\sqrt{2}}, \frac{4}{\sqrt{2}}\}$ to emphasize spots of radius 1, 2, 3 and 4 pixels. As described in (Stegmaier et al., 2014), a maximum projection image was then formed from these filtered images, *i.e.*, at each voxel location, the maximum response of the differently filtered LoG images was used to construct a combined LoG scale-space maximum projection image (LoGSSMP). In addition to storing the maximum filter response of each voxel, the scale (standard deviation) yielding the maximum value was stored as well, to obtain a size estimate of each detected spot. We then searched for local maxima in the 3D LoGSSMP image, *i.e.*, we identified every voxel location with an intensity value larger than its neighboring pixels in each of the channels. These detections were then used for manual refinement and the actual colocalization analysis.

2.2 Manual Threshold Adjustment

As we report all local maxima in the preprocessing step, even the dimmest local maxima were extracted from the images. However, the detected spots still contained many false positive detections in the image background that needed to be discarded. We thus used two intensity-based criteria to manually reject detections in the background regions. First, we extracted the mean intensity of each detection using a cube with side length equal to $2r + 1$, where r is the estimated spot radius identified during the LoGSSMP spot detection step. Furthermore, we computed the ratio of this mean intensity to the mean intensity of a second, larger cube excluding the voxels contained in the smaller inner cube. The size of the outer cube can be adjusted relative to the size of the inner cube and for the presented experiments, a factor of 4 worked well in practice. This mean intensity ratio has the flavor of a signal-to-noise ratio, *i.e.*, it compares the average spot intensity to the average intensity of its environment. Values close to 1 indicate that these two intensities are similar. The higher this ratio becomes, the higher the contrast of the detected spot is in comparison to the surrounding background signal. This measure is particularly useful if intensities of the image are inhomogeneous and if it is thus not sufficient to use only a global intensity threshold. These two thresholds were then used to reject false positive detections that should not contribute to the remaining analyses.

2.3 Colocalization Analysis

After having adjusted the manual thresholds such that the remaining detections corresponded to the spots of interest and ideally all false positive detections have been rejected, we performed a colocalization analysis of the remaining spots. Therefore, we identified the respective nearest neighbor of each of the remaining spots of the first channel in the second channel. Based on the size estimate of the spots, a pair of neighbors was considered a valid colocalization if their spatial distance d was less than the sum of their radii, i.e., if $d \leq (r_1 + r_2)$. Alternatively, a fixed maximum distance of centroids can be specified, such that only nearest neighbor pairs with centroid distances lower or equal than this maximum distance are considered a valid colocalization. Identified colocalizations were then visualized and we additionally computed the percentage of colocalizations as well as scatter plots of the intensities of the spots in the different image channels. Furthermore, it is possible to automatically generate a set of background detections that can be used for comparison. We used a Delaunay triangulation of the locations of the colocalized spots of both channels and then placed artificial background detections in the center of the identified triangles. To avoid background detections being located too close to real spots, a minimum distance criterion was added, such that only background spots with a minimum distance of $d_{min} = 8 \cdot r_{mean}$ were accepted, with r_{mean} being the average radius of all colocalized detections.

2.4 Implementation and Availability

All analyses described in this contribution were performed using the open-source C++ software XPIWIT for spot detection (Bartschat *et al.*, 2016) and using a newly developed graphical user interface (GUI) implemented in MATLAB for the final analysis. The GUI takes care of automatically processing selected image pairs with XPIWIT and imports the detected spots back into MATLAB. The threshold values for the two intensity features can be manually tuned in the GUI with an immediate visual feedback such that the optimal threshold parameters can be interactively tweaked. Furthermore, all subsequent analyses including colocalization, scatter plots, single detection information, background detection generation and the generation of result summaries can be performed with the GUI. We provide software packages for Windows, Mac OSX and Linux including a copy of the XPIWIT binaries for each platform from https://github.com/stegmaierj/spotdetectionandcolocalizationgui .

References

Bartschat, A.; Hübner, E.; Reischl, M.; Mikut, R. & Stegmaier, J. XPIWIT - An XML Pipeline Wrapper for the Insight Toolkit, *Bioinformatics*, 2016, 32, 315-317

Stegmaier, J.; Otte, J. C.; Kobitski, A.; Bartschat, A.; Garcia, A.; Nienhaus, G. U.; Strähle, U. & Mikut, R. Fast Segmentation of Stained Nuclei in Terabyte-Scale, Time Resolved 3D Microscopy Image Stacks, *PLoS ONE*, 2014, 9, e90036