Image processing procedures by ImageJ (version 1.51f)

For preparing images and their skeletonized images to be analyzed by persistence software produced by De La Cruz lab at Yale University. August 2016

The persistence software needs two images to process: a so called "original" image and its "skeletonized" image. Both of the "original" and "skeletonized" images can be single image or a stack of images. The "original" image has to be in one of 8, 16, or 32 bit grayscale or RGB formats, while the skeletonized image is a binary image in 8 bit grayscale with white signal pixels (value 255) and black background (value 0).

1. Load raw images.
   If there is only a single image, load directly or if you have a TIFF stack already, you can load using
   File-->Import-->TIFF Virtual Stack and click on the desired .tif file.
   If the images are in a series of numbered images:
   File-->Import-->Image Sequence and click the first image to import.
   Check "Sort names numerically" and convert to either 8-bit Grayscale or RGB (24 bit total and 8 bit per color) if in other formats than 8, 16 or 32 Grayscale or RGB.
   Note: This is critical because other formats of image won't be displayed correctly in the persistence software, although it can automatically recognize 8, 16 or 32 Grayscale or RGB.

   Note: You can press Ctrl+Shift+D to duplicate your stack of images at every processing step. This will allow you to take a step back in case a processing step does not work, otherwise you will start all over. <u>These processing steps cannot be undone</u>.

2. Subtract background to fix uneven illumination.
   Process--> Subtract Background.
   Set rolling ball radius to 10 pixels, because actin filament width is about 6-10 pixels. If you have thicker or thinner filaments, choose the pixel roughly equal your filament width. Make sure all other options are unchecked. *Optional:* You check preview option to make sure your subtraction is sufficient. Click "OK". Then click "Yes" to process all images.

3. Enhance contrast.
   Process-->Enhance Contrast.
   Saturation pixels: 0.3 or 0.4 % (ImageJ default). Check "Equalize histogram", "Process all # slices" and "Use stack histogram".

4. Save image stack as the "original image" for Persistence software analysis.
   File-->Save As-->Tiff.

We suggest saving to a different folder than the folder with raw images or a new name if in the same folder. After saving the "original image", don't close the image and proceed to prepare its skeletonized image.

5. Despeckle (optional).
   Process--> Noise--> Despeckle.
   Use if there are many breakages (darker spots) on the filaments or the image contains salt and pepper shot noise.

6. Smooth filaments and filament edges by Gaussian Blur.
   Process-->Filters-->Gaussian Blur.
   Choose Sigma (radius): 2 pixels or 1 pixel. With value of 2, it is more smooth locally and the background is much reduced, but details may be lost (e.g. more breaks in the filament). With value of 1, more detail is preserved while the edges of curves get smoothed. However, the background may remain too high. With 1 pixel, Gaussian blur combined with manual adjustment to eliminate background effects may work best. Other filters to smooth filaments locally can also be used, but you need to test which one is best for your data.

7. Make binary image using Threshold. *If the image is in RGB, you need to convert it to 8-bit grayscale before thresholding* (**Image-->Type-->8-bit**)*, but not if it is in 16 or 32 bit grayscale. If you don't convert from RGB, the new version of ImageJ would ask you to do so, while some old versions. e.g. V1.46r, still let you do the thresholding, but automatically switch to "color thresholding", which is not well defined or understood and you must experiment on your own to get correct image for the next steps.*
   Image-->Adjust-->Threshold.

   Please read through this entire step before proceeding, as it is somewhat complicated.

   Choose MaxEntropy or whichever thresholding method you prefer. Threshold divides pixels into two groups: the pixels with an intensity value below the threshold value go to background group and are given a uniform intensity value of 0, while the pixels with its intensity value above the threshold value go to image group with a uniform intensity value of 255 (if it is 8-bit grayscale). The difference between different methods (MaxEntropy, Mean, MinError, etc.) is the automatically-determined threshold value. Some methods have lower threshold values that may have more background noise. A higher threshold value will have much less background, but may lose some signal. Choose a method that gives minimal background noise and combine with some minor manual adjustment by dragging the threshold bar left or right. When opening the threshold dialogue or changing the threshold method, ImageJ automatically sets threshold

value for all images according to the current image in the stack. If you want to reset threshold value according to a particular image, scroll to that image and click “Auto”. Any manual threshold adjustment you make will affect all images. Other images in the stack may not look good with the chosen threshold, but this will be fixed later.

Choose display: Red or B&W or over/under doesn't matter. It is only for visualization.
Check: "Dark background" if your image is white on black background (threshold part now is white signal portion).
Optional: check "stack histogram". This not recommended. You should threshold each image individually.

Now, after choosing the threshold method on the threshold setting panel, use one of the following three ways to threshold the images. First way, threshold all images in the entire image stack using a uniform threshold value based on a particular image or "stack histogram" of the entire stack. Scroll to the particular image that you want the uniform threshold value to be based on and then click “auto”, or just check "stack histogram" if you want the uniform threshold value to be based on the entire image stack and click “auto” on the threshold setting panel, to see how good auto threshold value is. If the auto threshold value set by the chosen method is not good enough, adjust it by manually dragging the threshold bar left or right. Then go through each image to see if the uniform threshold value is good for all images. Remember, don’t adjust the threshold value, click “auto”, or change auto threshold method on any image other than the particular image you want the threshold to be based on. Otherwise, the threshold value will change based on the current image. If you need to change the threshold value, go back to that particular image. If the uniform threshold value is acceptable for all the images, click “Apply” and don’t check any box to uniformly threshold all images. Second way, go through each image and click “Auto” to see how well the image is auto-thresholded. If all images are acceptable, click “Apply” and check "Calculate threshold for each image" and ImageJ will automatically threshold each image individually. Third way, if you have some images that aren’t well auto-thresholded, you have to adjust threshold for each image manually by dragging the threshold bar left or right. After manually adjust threshold for an image, click “Apply” and check "Only convert current image". Repeat this procedure for every image in the entire stack.

In the "Apply" dialogue:
Method: whichever method you chose.
Background: "Dark" if you have white images on black background. This will produce a white background with black filaments; this is critical for the next step.

If you chose "stack histogram" in the previous window or if you want the same threshold for each image, do not check any boxes. Otherwise, check either "Calculate threshold for each image" or "Only convert current image".

Note: The user may save the name of the threshold method as well as the threshold values used. This is important to report those for later users or for verifying the data set again. Different thresholding numbers could lead to bias and might be cross-checked by a second user. The method as well as the values used can be saved directly in the file name so it is easy to find. Although doing this is only useful when you threshold all images using a uniform threshold value. For manually adjusting threshold for each image, it is difficult to record all the threshold values.

8. Skeletonization.
   Note: For skeletonizing images, after thresholding the filaments must be black on a white background if you threshold grayscale images. Otherwise, the ImageJ cannot properly skeletonize the images, meaning that it skeletonizes the background instead of signal pixels. If you follow this suggested procedure and the images are not black with a white background for some reasons, click "LUT" on the tool bar and then click "invert LUT". If you threshold using “color thresholding” because of your RGB images and an old version of ImageJ, the correct image for skeletonization is different and you have to experiment to find out.
   Process-->Binary-->Skeletonize.
   *If you get an error during skeletonize stating that the image needs to be 8-bitgrayscale, go to image -> type and convert to 8 bit. This also suggests that your images are thresholded by “color thresholding” and you have to make sure it is in correct “LUT"

9. Save skeletonized image.
   Now if you follow these suggested procedures, the images you get so far should be black skeletonized images in white background and you need to invert "LUT" for saving it to correct images. Click "LUT" on tool bar and then click "invert LUT" to get white filaments on a black background. This is critical because ImageJ saves black as "0" and white as "255". If you save black skeleton images on a white background, Persistence software won’t process them because black and white values are inverted and the persistence software recognizes value 255 as signal pixels and 0 as background.
   File-->Save as-->Tiff.
   Save it to the same folder as the original image stack and choose a filename to indicate it is a skeletonized image of the “original”.

You are now ready to proceed to persistence analysis by the persistence software. Please refer to the published article *Plos one*, 9 (4), e94766 (2014) and the documents accompanied to the software for using the software and about output files.

Comment: ImageJ keeps changing. The procedures listed here may not apply to future new versions. However, the most parts in this writing apply to most of versions so far and possibly future versions. Only two places need to be watched out: one is thresholding to get a binary image with correct “LUT” for skeletonization and the other is to save the skeletonized images with correct “LUT”, i.e. signal pixel value 255 and background value 0. If ImageJ changes and you cannot get correct images for either sckeletionization or the persistence software, try to invert image “LUT” before the troubled procedures.