Jul 23, 2024   Version 2

🌐 An ImageJ macro for the quantification of ciliary accessible cholesterol V.2

DOI

dx.doi.org/10.17504/protocols.io.q26g71d88gwz/v2

Jonas Nikoloff[1,2], Sreeja V Nair[1,2], Suzanne R Pfeffer[1,2]

[1]Department of Biochemistry, Stanford University School of Medicine; [2]Aligning Science Across Parkinson's

ASAP Collaborative Rese…	Suzanne Pfeffer

👤 **Suzanne R Pfeffer**
Stanford University School of Medicine

OPEN 🔓 ACCESS

DOI: **dx.doi.org/10.17504/protocols.io.q26g71d88gwz/v2**

Protocol status: Working
We use this protocol and it's working

Created: July 19, 2024

Last Modified: July 23, 2024

Protocol Integer ID: 103941

Keywords: ASAPCRN

Abstract

ALOD4, a toxin-based probe, can be used to visualize the accessible pool of cholesterol (1). Here we describe an algorithm for segmenting fluorescently-labeled cilia using SSTR3 as a ciliary marker. The segmented masks are used to determine the intensity of ALOD4-bound accessible cholesterol in the ciliary membrane. The algorithm asks for a directory path and searches for .tif-files. Each identified .tif-file is cropped for a user defined region of interest (ROI) which must contain the cilia in the image series. Then the Z-stack is maximum intensity projected, the contrast of channel 1 (SSTR3-mApple, red) is optimized for visualization, and both channels are duplicated. The MorphoLibJ library helps to identify the SSTR3-positive cilia. SSTR3 signals are then thresholded, binarized, and intensity-corrected. A shape filter prevents false positive signals by preventing non-ciliary structures to be assigned as objects. The identified objects are applied onto the channel 2 signal (mNeon-ALOD4, green) and intensities are measured and saved.

1. Endapally S, Infante RE, Radhakrishnan A. Monitoring and Modulating Intracellular Cholesterol Trafficking Using ALOD4, a Cholesterol-Binding Protein. Methods Mol Biol. 2019;1949:153-163. doi: 10.1007/978-1-4939-9136-5_12. PMID: 30790255; PMCID: PMC6459602.

Materials

- Fiji/ImageJ (Ref. 2) with morphoLibJ library (Ref. 3) and ShapeFilter (Ref. 4) plugin

- .tif-files of images to be analyzed

2. Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675. **doi:10.1038/nmeth.2089**

3. Legland, D., Arganda-Carreras, I., & Andrey, P. (2016). MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics*, *32*(22), 3532–3534. **doi:10.1093/bioinformatics/btw413**

4. Wagner, T. & Lipinski, H-P. (2013) IJBlob: An ImageJ Library for Connected Component Analysis and Shape Analysis. *Journal of Open Research Software*, *1*(1), e6. **doi:10.5334/jors.ae**

Description of the Macro: Four basic steps

1   Step 1: Drag and drop the macro to ImageJ/Fiji

2   Step 2: Click "Run". The macro requests the user for a folder containing .tif files of images to be analyzed. Choose the folder, click 'Open'.

3   Step 3: Create a target directory ("processed") to save processed images in .tif format and the result, mean ALOD4 intensity in .csv format

4   Step 4: The macro will go through each .tif file, one at a time, and apply the 'function' for image processing

5   **NOTE**: The user performs steps 1-3. Once the macro is finished, the user can manually verify the cilia objects identified by the macro, the ciliary mask in channel 1, and the ROI generated in channel 2 for measuring mean ALOD4 intensity. All files are saved in the target directory/ output folder.

6   Details of the functions follow:

Code details

7   Line 47-52: this code instructs macro to go through all images and obtain the image title

8   Line 53: this code renames the images to raw01

9   Line 57-58: this code crops the image in channel 1.

10 Cropping is done in such a way that the cilia is located at the center of the image. Cropping also minimizes the size of the images, thereby increasing the speed of the macro. Also, it can help reduce unwanted, high intensity pixels in the background.

11 NOTE: All images analyzed here are acquired in 512 X 512 format and the command " makeRectangle (205, 225, 100, 100)" ensures cilium is indeed at the center of the cropped images. 205, 225- represents the coordinates of the upper left corner of the rectangle and

100,100 represents the size of the rectangle. Users can verify the cropped images in the output folder and adjust the parameters by changing the values for cropping if needed.

12   Line 59: renames the cropped image as 'crop_raw01'

13   Line 65: for maximum intensity projection (mip) of 'crop_raw01'

14   Users can define the Z-slices for maximum intensity projection by typing the start and stop number of Z in line 65 as cilia only appear in 1-2, Z-slices.

15   NOTE: If the user wants to include the whole Z-stack for maximum intensity projection, replace the command in line 65 with run("Z Project...", "projection=[Max Intensity]")

16   NOTE: We did not see any differences with predefined Z-slices versus whole Z-stack for maximum intensity projection. However, working with predefined Z-slices eliminates some background and makes cilia segmentation easier in subsequent steps.

17   Line 68-69: this code adjusts the brightness and contrast of the maximum intensity projected cilia in channel 1 for visualization purpose only.

18   Line 71: renames the maximum intensity projected files of channel 1 and 2

19   Line 72-78: these steps duplicate channel 1 and channel 2. These images will be used in subsequent steps for creating a cilia mask in channel 1, and creating a ROI in channel 2 based on the cilia mask in channel 1.

20   Line 88-102: these steps detail cilia segmentation in channel 1.

21   Line 88-90 : These steps are for using the morphological Laplacian of MorphoLibJ library to enhance the edges of the cilia object. The resulting image is renamed in line 90.

22   Line 92: this code is for smoothing the "cilia object" identified by the macro in line 90 by the Gaussian blur function with a sigma of 1.

23   Line 96-97: the cilia object is further subjected to thresholding by "Auto threshold-Yen dark stack". The user can change the thresholding method by replacing Yen from line 97 with the method of choice.

24   NOTE: Yen works well with the images that are being analyzed here.

25   Line 98: this code is for converting the cilia object in channel 1 to a mask.

26   Line 99-101: these steps are for using morphological filters in combination to further refine the segmented objects. A combination of dilation followed by erosion used in these codes removes structures that are smaller than the cilia mask.

27   Line 102: This code is for shape filter. Here, parameters are set for morphological filtering of elongated structures representing cilia. This step ensures objects being analyzed in the subsequent steps are indeed cilia.

28   Line 106-108: this code is for creating a ROI in channel 2 and measures the mean intensity, based on the cilia mask in channel1. Each ROI is saved in the output folder.

29   Line 118,120,122, and 130: this code saves the maximum intensity projected images of channel 1 and 2, cilia mask on channel 1, and ROI in channel 2.

30   Line 147: summarizes all features into a single .csv and saves it in the output folder/ target directory created by the user in step 3.
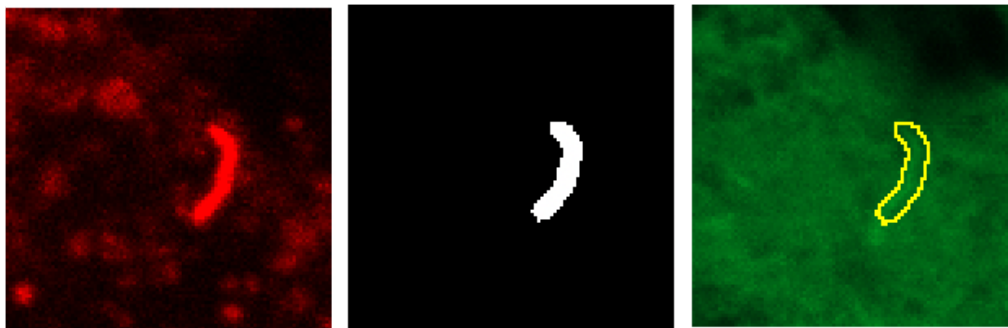
31   Example:

Fig. 1: Representative image showing maximum intensity projected cilium object in channel 1 (red, left), cilia mask created in channel 1 (white, middle), and the ROI in channel 2 (green, right) based on the cilia mask in channel 1.

Macro (Written by Jonas Nikoloff)

32

// Free memory

```
run("Collect Garbage");
IJ.freeMemory();

// clear ROI Manager
close("ROI Manager");
directory = getDirectory("Choose a Directory");

// get image IDs of all open images
for (i=0;i<nImages;i++) {
    selectImage(i+1);
    title = getTitle;
    print(title);
    saveAs("tiff", directory+title);
    }

run("Close All");

///### When selecting images manually
directory = getDirectory("Choose a Directory");//

// Get image directory and list image numbers
list = getFileList(directory);
print(list.length);

// Create directory for processed images
File.makeDirectory(directory+"processed");
processed_Dir = directory+"processed";

// Start looping through the save tiff files applying the 'operation' function
    for(i=0; i < list.length; i = i+1){//list.length ; i = i++){
        if (endsWith(list[i], ".tif")){
            operation();
            }
            };

// Print at the end
print("Macro Finished");

// Applied function
function operation(){

filename = directory + list[i];
open(filename);
```

```
title = getTitle;

//*directory = getInfo("image.directory");
File.makeDirectory(directory+"processed");
processed_Dir = directory+"processed";
title = getTitle;//*/
rename("raw01");

//# Cropping
selectWindow("raw01");
makeRectangle(205, 225, 100, 100);
run("Crop");
rename("crop_raw01");

//# Max Int Proj
selectWindow("crop_raw01");

//# With z limits
run("Z Project...", "start=8 stop=9 projection=[Max Intensity]");

run("Brightness/Contrast...");
run("Enhance Contrast", "saturated=0.35");

rename("MIP_selstack");
run("Duplicate...", "title=MIP_selstack_ch01_dup duplicate channels=1");

selectWindow("MIP_selstack");
run("Duplicate...", "title=MIP_selstack_ch01 duplicate channels=1");

selectWindow("MIP_selstack");
run("Duplicate...", "title=MIP_selstack_ch02 duplicate channels=2");

close("MIP_selstack");

close("crop_raw01");

// Filtering
selectWindow("MIP_selstack_ch01");
rename("01");
run("Duplicate...", "title=02");
run("Morphological Filters", "operation=Laplacian element=Square radius=1");
imageCalculator("Subtract create", "01","02-Laplacian");
```

```
rename("MIP_selstack_ch01");
rename("MIP_selstack_ch01");
run("Gaussian Blur...", "sigma=1");

//# Thresholing
selectWindow("MIP_selstack_ch01");
run("Threshold...");
setAutoThreshold("Yen dark stack");
run("Convert to Mask");
run("Dilate");
run("Erode");
run("Fill Holes");
run("Shape Filter", "area=0.5-Infinity area_convex_hull=0-Infinity perimeter=0-Infinity
perimeter_convex_hull=0-Infinity feret_diameter=0-Infinity min._feret_diam    eter=0-Infinity
max._inscr._circle_diameter=0-Infinity area_eq._circle_diameter=0-Infinity
long_side_min._bounding_rect.=0-Infinity short_side_min._bounding_rect.=0-Infinity
aspect_ratio=1.6-Infinity area_to_perimeter_ratio=0.1-Infinity circularity=0-62 elongation=0.69-1
convexity=0-1 solidity=0-1 num._of_holes=0-Infinity thinnes_ratio=0.2-1 contour_temperatur=0-
1 orientation=0-180 fractal_box_dimension=0-2 option->box-sizes=2,3,4,6,8,12,16,32,64
draw_holes black_background exclude_on_edges");
run("Analyze Particles...", "size=50-Infinity pixel circularity=0.00-0.90 show=Masks exclude
include add");
run("Invert");

// measuring ROI
selectWindow("MIP_selstack_ch02");
roiManager("measure");

// renaming and saving
nROIs = roiManager("count");
print(nROIs);

//nROIs=0
if    (nROIs>0){
    print("option1");
    selectWindow("MIP_selstack_ch01_dup");
    saveAs("tif", processed_Dir + "/" + title + "_MIP_selstack_ch01_dup.tif");
    selectWindow("MIP_selstack_ch02");
    saveAs("tif", processed_Dir + "/" + title + "_MIP_selstack_ch02.tif");
    selectWindow("MIP_selstack_ch01");
    saveAs("tif", processed_Dir + "/" + title + "_Mask_of_MIP_selstack_ch01-1.tif");
    nROIs = roiManager("count");
    print(nROIs);
```

```
        sel_array = newArray(nROIs);
        for (i = 0; i < nROIs; i++) {
        sel_array[i] = i;
        }
        roiManager("Select", sel_array);
        roiManager("Save",processed_Dir + "/" + title + "_cilia_masks.zip");
        run("Select All");
        roiManager("Deselect");
        roiManager("Delete");
    }

    else {
        print("no ROIs");

        }

        roiManager("Reset");

        //run("Close All");
    }

    // Saving results
    saveAs("Results", processed_Dir + "/" + "Results.csv");
    run("Clear Results");
```

Protocol references

1. Endapally S, Infante RE, Radhakrishnan A. Monitoring and Modulating Intracellular Cholesterol Trafficking Using ALOD4, a Cholesterol-Binding Protein. Methods Mol Biol. 2019;1949:153-163. doi: 10.1007/978-1-4939-9136-5_12. PMID: 30790255; PMCID: PMC6459602.

2. Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, *9*(7), 671–675. **doi:10.1038/nmeth.2089**

3. Legland, D., Arganda-Carreras, I., & Andrey, P. (2016). MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. *Bioinformatics*, *32*(22), 3532–3534. **doi:10.1093/bioinformatics/btw413**

4. Wagner, T. & Lipinski, H-P. (2013) IJBlob: An ImageJ Library for Connected Component Analysis and Shape Analysis. *Journal of Open Research Software*, *1*(1), e6. **doi:10.5334/jors.ae**