

Using the image processing toolbox

This image processing toolbox estimates the nuclear surfaces of individual nuclei from confocal image stacks containing multiple nuclei. First, the location of each nuclei in a stack is obtained by projecting the 3D stack into a 2D image and running a 2D active contour algorithm. Next, any incorrectly segmented nuclei is removed manually using a GUI based script. Finally, we use the location of individual nuclei to segment it in 3D using a 3D active contour algorithm. We utilize the following toolboxes from the mathworks website – (i) the active contour algorithm coded by D. Kroon, University of Twente, and (ii) geometry methods coded by David Legland, INRA.

1. Create a 'readme.txt' file in each of the folders containing your image stacks. This file contains information of the filename pattern for each stack of confocal images in a folder. Fig. 1 explains a sample 'readme.txt' file

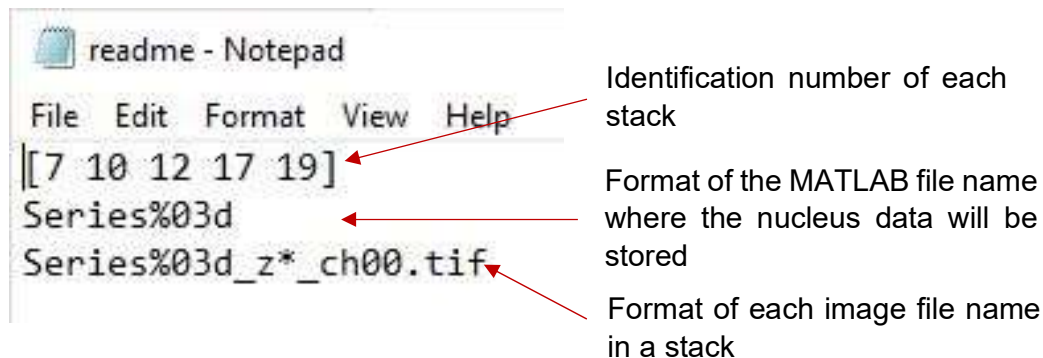


Fig 1: Format of the readme.txt file. The first row contains the identifiers for each stack, the second row is the format of the MATLAB file in which the nucleus shape data will be stored, and the third row is the format of each image file name in a stack. For example, the first stack will have image files with filename – 'Series007_z*_ch00.tif' and the nucleus shape will be stored in 'Series007.mat'

2. Open the 'proj_run2D_manyfolders.m' file and enter the path to the folders containing your image data. Change the *pxl* variable to the size of the image voxel in μm of your images. This information is typically available in the metadata of the image stacks. Now run the 'proj_run2D_manyfolders.m' file. This will read each stack, project the 3D stack into a 2D image file and obtain the 2D boundaries of each nuclei using a 2D active contours algorithm. A MATLAB binary file ('.mat' format) is created for each stack and the corresponding 2D nuclei boundaries are stored in it. Note: If your images are in RGB format, the program assumes that the nucleus is in Blue colour. If not, then the line number 25 – 'zimg = squeeze(tmp(:,:,3));' in 'read_stack_proj_run2D.m' should be updated.
3. A few nuclei might not be segmented properly. These need to be manually removed using the 'remove_nuclei.fig' GUI program. Point to the folder containing your image data and type the filename of your image stack without the extension (For example 'Series007'). The projected image and the 2D nuclear boundaries will be loaded in the GUI. Left click on any nuclei that hasn't been segmented properly to remove it from the analysis. A red cross on the nuclei, after you have clicked on it, shows that it will be removed from the

analysis. Manually check all the image stacks for nuclei that haven't been segmented properly and remove them. See figure below

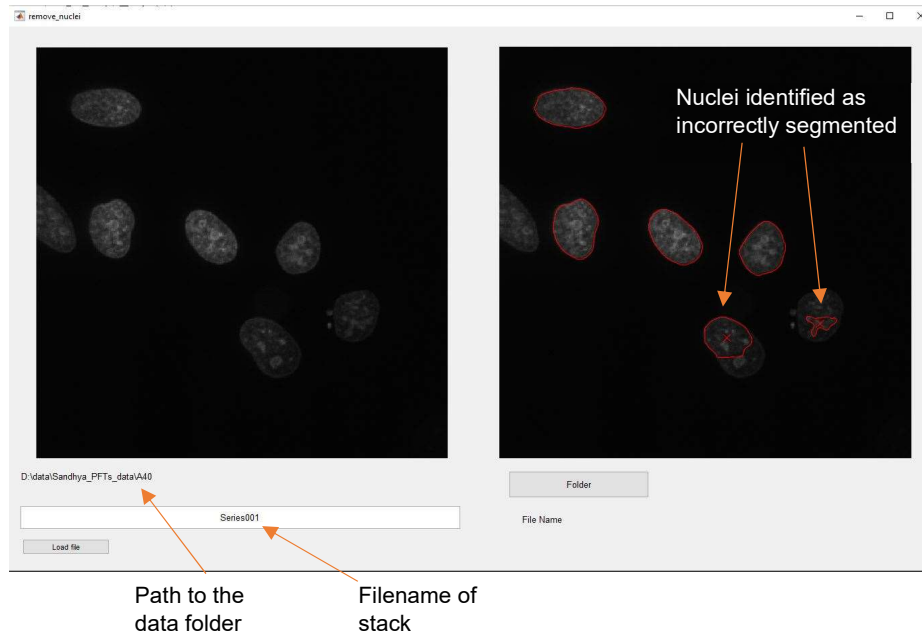


Fig 2: Removing incorrectly segmented nuclei. Run the 'remove_nuclei.fig' MATLAB GUI. Click on the 'Folder' button and navigate to the folder containing the data. Enter the name of the stack to load in the File Name textbox. Click 'Load File' and the projected image and segmented nuclei will be loaded as shown. Left click on incorrectly segmented nuclei to remove them from further analysis. Nuclei selected for removal will be marked with a red cross.

4. Open the 'volume_manyfolders.m' file and enter the folder names containing your image data. Change the *pxl* variable to the size of the image voxel in μm of your images. Run 'volume_manyfolders.m'. This will run the 3D active contours algorithm on individual nuclei and obtain the nuclear surface as a triangular mesh.
5. Open the 'save_folder_areas_vol.m' file, edit the *fldr* variable to the path of your data folder and run the script. This script will calculate the projected area, surface area, and volume of nuclei from the surface mesh.
6. To check if the 3D active contours algorithm has converged correctly to the nuclear surface, use the 'check_3D_convergence.m' script. Edit the *fldr* variable to the path to the folder containing the image stacks, *stck_num* to the serial number of the stack and *nuclei_num* to the serial number of the nuclei. The script will show the orthogonal sections of the image stack superimposed with the nuclear surface mesh and a 3D plot of the nuclear surface mesh.
7. For each image stack a corresponding '.mat' file is created. The shape data is stored in a struct array variable named *dat*. Each element of this array corresponds to an individual nuclei. *dat.surf* is the surface mesh of the nuclear surface, *dat.PA* is the projected area, *dat.SA* is the surface area and *dat.V* is the volume of the nucleus.