Standard Operating Procedure

TU/e

SOP details

Title	Images analysis workflow
Description	This SOP describes cropping of the images and how to run the CellProfiler analysis.
Author	Aliaksei Vasilevich; Tim Kuijpers
SOP number	4.2
Version number	4

1 Purpose

The purpose of this SOP is to first crop the raw images and second, to obtain cell segmentation images and an image object with numerical parameters.

2 Principle

The principle of this SOP is to crop the whole TopoChip image to the individual TopoUnit images and to perform segmentation analysis of nuclei and cells. Finally, quantitative data of cell parameters is obtained from the CellProfiler output.

3 Important to know before starting

The Topochip analysis workflow assumes you are working in the TopoChipAnalysis folder (see SOP 4.1). Furthermore, we assume you have followed the introduction to CellProfiler before you start with the data analysis. If not, first finish the CellProfiler tutorial at https://cellprofiler.org/tutorials. Also, you have established a fully operational CellProfiler pipeline for your assay of choice.

4 Required materials

4.1 Workplace

This SOP can be performed on your Laptop/Desktop.

4.2 Equipment

- Laptop or desktop with MATLAB, NIS viewer and CellProfiler installed.
- CellProfiler pipeline located in the folder TopoScreenAnalysis. Note that this is a very basic pipeline and you should add modules where needed. Before you continue with the CellProfiler analysis it is important to have a fully operational CellProfiler pipeline.

4.3 Procedure

4.3.1 Raw images: convert nd2 files to tif files.

1. Open NIS Viewer to convert the images from .nd2 files to .tif files. TopoChip images tend to be large, therefore process one chip and one channel at a time. To open an image, go to *File* → *open*.
2. Once the images is loaded, you perform a visual check to see if the raw image is of good quality (Figure 1). Based on your own criteria (failed staining, defective TopoChip, wrong stitching, etc.) you may decide not to include a chip in further analysis. Furthermore, it is important to check whether the staining has stained the right cellular components.

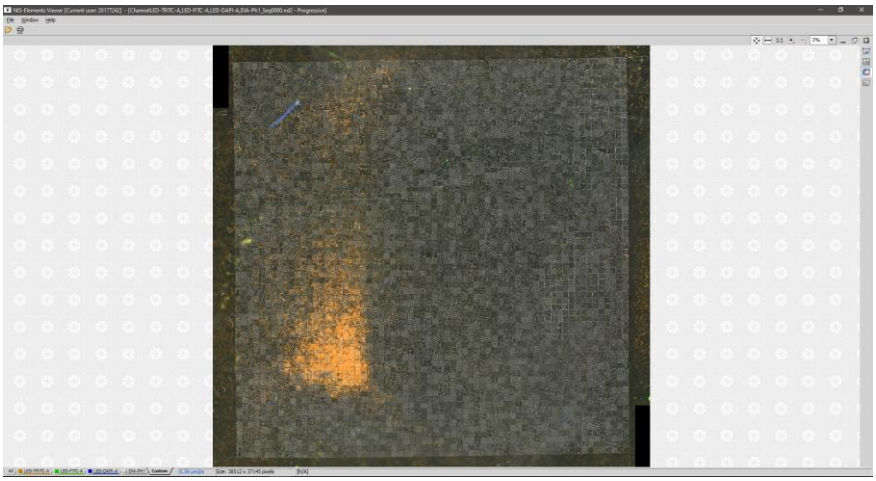

Figure 1 NIS viewer interface: NIS viewer can be used to inspect the image quality and to convert the images to .tif files.

3. Export the image as an .tif file by taking the following steps (figure 2):
    a. File -> choose save/export to tif files.
    b. Define the output folder.
    c. Select the option “standard tif”.
    d. Check the box “OME-metadata”.
    e. Repeat these steps for the other channels and all remaining TopoChips.

    Note: step 3 has to be done for each file in your folder. If Chip 1 contains three channels (DAPI, TRITC, and FITC), you have to perform step 3 three times.

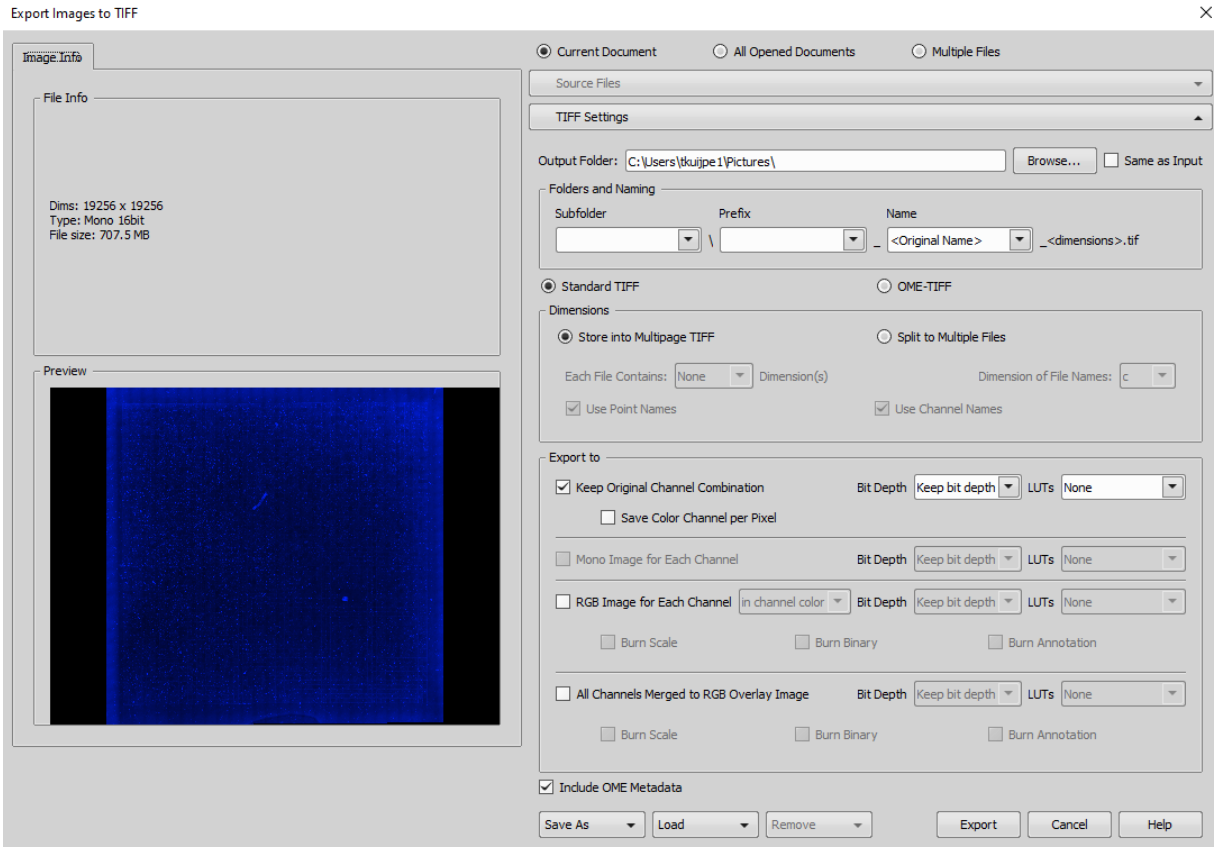

Figure 2 NIS Viewer: overview of the export settings to convert .nd2 files to .tif file.

4.3.2 Image cropping with MATLAB

Before you can run CellProfiler to analyze and segment your images, you need to crop the TopoChip images to get an image per TopoUnit. This will be done in MATLAB by running the file *topocropper.m* (This file is located in the *TopoScreen Data analysis* folder downloaded in SOP 4.1).

1. Open MATLAB and navigate to the folder *TopoScreen Data analysis* in the folder tree on the left in MATLAB.
2. In the same folder tree, right click on the map *RawImages* and click '*add to path*'. Now, MATLAB is able to access all files within the folder. Important note, this should be done after you completed section 4.3.1, MATLAB may not be able to access files that have been created after adding locations to the path.
3. In the MATLAB script set the following variables:
    a. Screen name (line 10 in script)
    b. Channel names (line 13,14, and 15 in script)
4. Click the play button in the menu bar to run the script. MATLAB will ask for your input to set:
    a. Chip Number
    b. Rotation angle
5. In some cases, the TopoChip image has a minor rotation (see figure 3). This needs to be corrected before you crop the image. The script will ask you to give the rotation angle and you can enter the following values:
    a. A value of zero to indicate no rotation is needed.
    b. A positive value to rotate the image counterclockwise.

c. A negative value to rotate the image clockwise.
The rotation value needs to be determined by visual inspection.

6. Next you have to specify the four corners of the TopoChip. A figure window will automatically appear, click on the corner piece of the chip with the crosshairs (Figure 3A). You will move from TOP-LEFT, to TOP-RIGHT, to BOTTOM-RIGHT to BOTTOM-LEFT.
7. To increase the accuracy of the cropping mesh, you are now asked to give another 8 locations on the outer wall of the Chip: 2 locations on the top, right side, bottom and left side of the TopoChip, respectively.
8. Finally you have to point out another 4 intersection of the walls. Here it is important to try to click on the center of the intersection.
9. This will create a mesh on top of the TopoChip (Figure 3B). You can zoom in and see if the points are located at the walls.
10. When the mesh on top of the TopoChip show good alignment (Figure 3C), you can proceed to the next section to save the cropped images.

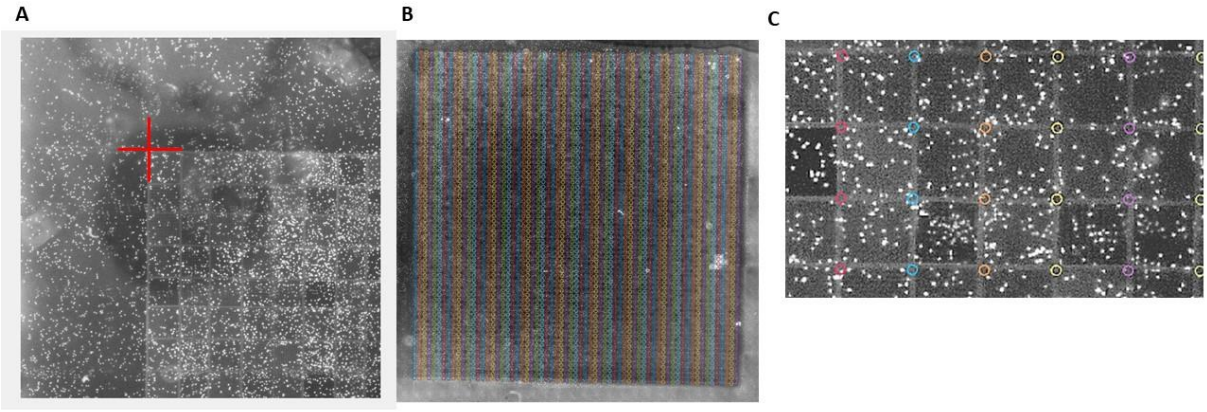

Figure 3 Image cropping in MATLAB. A: selection of the TopoChip corner. To achieve an accurate cropping, you should pay attention while clicking on the corner. **B:** the mesh defining the cropping region. **C:** You can zoom in to see if the points are on the corner of the TopoUnit. When the points are too far off (i.e. inside the TopoUnit), you need to redo the cropping.

4.3.3 Load the CellProfiler pipeline and upload the images

1. Open CellProfiler, click on *File* → *Open Project* and open your CellProfiler pipeline. This will open the image analysis pipeline you designed to work with the TopoChip screen analysis workflow.

4.3.4 Setup input parameters

1. First, we need to upload the images to the Cell Profiler pipeline. Therefore, you have to go the folder with the cropped images (created in step 4.4.1). To upload the cropped images, drag and drop them in the corresponding area in CellProfiler (Figure 4).

2. Next, switch to module Metadata from the panel on the left. The purpose of this module is to extract information from the folder and filenames of images, which will be used to link imaging data with corresponding replica and surface design.

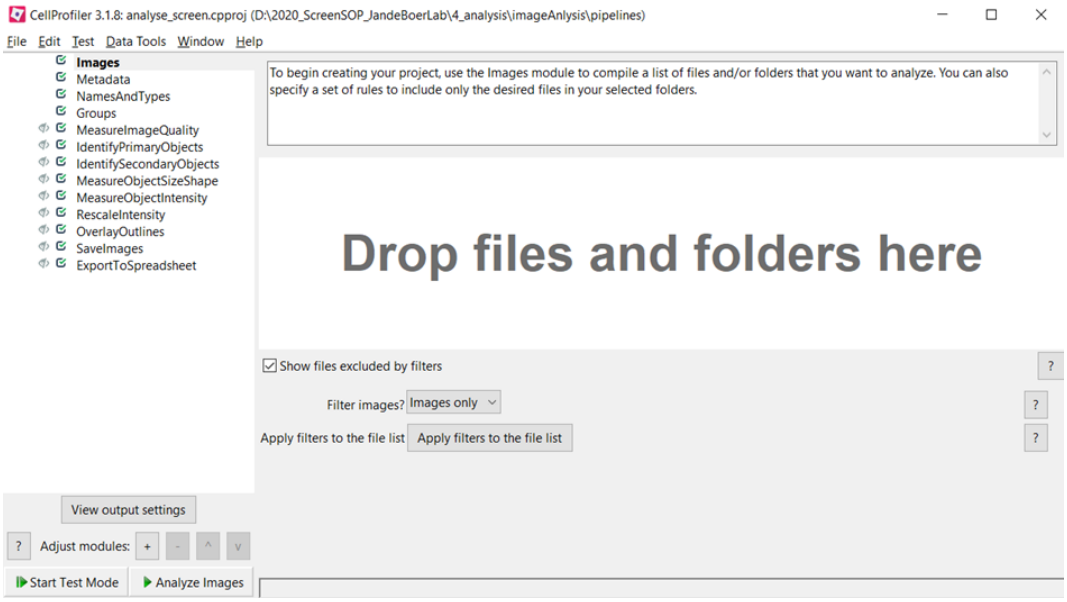

Figure 4: CellProfiler interface showing the area to drop and upload your images into the CellProfiler pipeline.

3. Next, click on the Metadata module to extract information from the filenames. Therefore we make use of a regular expression. The regular expression is based on the filenames given to the cropped images obtained with the MATLAB script. There is only one field you always have to change (Figure 5):

   ^Screenname_: the ^ and _ are characters used by CellProfiler to identify the screenname. You have to replace *Screenname* with the screenname you gave your files. For example, when your images start with M0_Screen, you will have to change the beginning of the regular expression to *^M0_Screen_* .

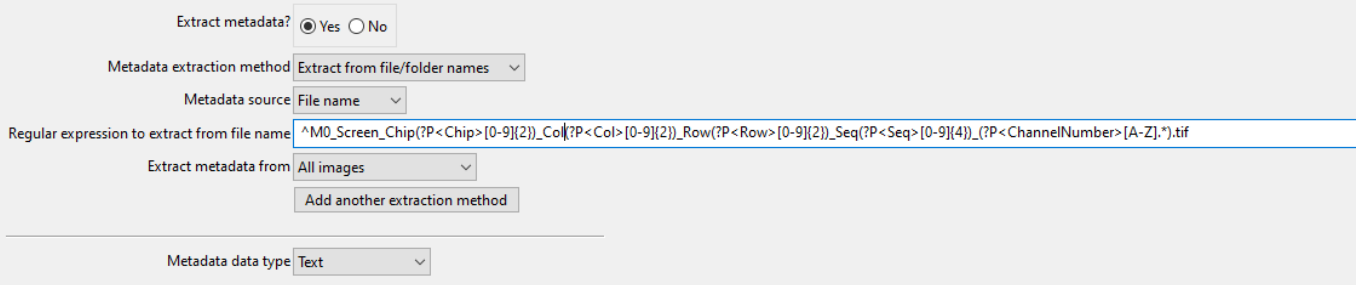

Figure 5 Regular expression used by CellProfiler to identify the metadata from a file name.

4. Click Update. Check that fields such as *Chip, Channel Number, Col, and Row* are filled with values in the Metadata module. When one column is empty, check the regular expression and filenames.
5. Switch to the NamesAndTypes module from the left panel. The purpose of this field is to provide unique names to images that were acquired by the microscope with the same optical settings. This information is stored in the metadata or filenames of the images. This information was extracted in the previous step. If you did not deviate from SOP 3.1 “Imaging of TopoChips”, and

you used the template provided by us, you do not need to make any adjustments. CellProfiler will use the metadata parameter ChannelNumber to name the images.

6. When you did not follow SOP 3.1, you have to define your own rules for the regular expression to extract the ChannelNumber in CellProfiler.

4.3.5 Create and save the output

After you have finished your pipeline and optimized the settings, two modules are used to save the segmentation images and measured object properties:

4.3.5.1 SaveImages

CellProfiler usually performs many image analysis steps on a large set of images. As by design, CellProfiler does not save any of the resulting images to the hard drive unless you specifically choose to do so with the *SaveImages* module. You can save any of the processed images created by CellProfiler during the analysis using this module.

4.3.5.2 ExportToSpreadsheet

In order to analyze the output with Python, you need to add this module at the end of the pipeline. The *ExportToSpreadsheet* function will export measurements in one or more files depending if you run one CellProfiler pipeline or a batch run of your pipeline.

In the field Default Output Folder you have to set the path to the folder /DataAnalysis/ to store the .csv files in the provided TopoChip analysis folder structure.

4.3.6 Start the CellProfiler Analysis

Before you can start your CellProfiler analysis, it is important to test the pipeline on your current experimental data. You have to perform a test run on a random set of images for each chip and check whether the segmentation is performed correctly.

Now you are ready to start your CellProfiler analysis. On a high-performance laptop this will take around four hours. When you need to analyze many TopoChips it is advised to make use of a High Performance system (Cloud computing).

4.3.7 Combine the CellProfiler image objects

When the CellProfiler analysis is finished, you have to transfer the image object csv files (those with _image.csv) to the folder */ImageObjects/* located in the data analysis folder. Open the Jupyter notebook to combine the individual image objects (per TopoChip) in one data file (the Jupyter notebook called *0_CombineImageObjects.ipynb). R*un step 1 and step 2 in the notebook, which will result in the *TotaImageObject.csv* that will be used to align the CellProfiler results to the TopoMap (SOP 4.3).