Cell Segmentation Overview:

- Step 1. Streak removal
- Step 2: Rearrange hyperstack, generate substack, and optional signal enhancement
- Step 3: Prepare for ilastik
- Step 4: Train pixel classifier
- Step 5: Segmentation
- Step 6: Export
- Step 7: Spillover compensation
- Step 8: Normalization
- Step 9: Clustering and cell type identification

Notes:

1. Image format (x,y,c,z) = (1000,1000,1,46).
2. Pipeline built under Cell Profiler version 3, ilastik 1.3.3, and R 4.0.2.

Cell Segmentation Overview:

Modified based on:
https://nbviewer.jupyter.org/github/BodenmillerGroup/ImcSegmentationPipeline/blob/development/scripts/imc_preprocessing.ipynb

Step 1: Technical artifacts (streaks of high intensity pixels) are removed using a customized R script (vigenettes/cell_segmentation/step1_streak_removal/ removing_streaks.R).

1. Start with folders containing multi-channel tiff files. Run the script by supplying path to input directory and a path to output folder. Images with streaks removed will be saved to the output directory, following the same directory structure as input directory.

Script description: The first two channels of our multi-channel images are empty channels, targeting ions that are not present in the panel. We used these two channels to detect technical artifact. If streaks of strong intensity pixels are present in these two channels, meaning they are among the top 2% of pixel intensity in the channel and it is 5-times greater than the average in the 5x5 neighborhood (excluding the neighbors on the same row), these locations will be evaluated for the rest of channels. For the rest of the channels, if it is 5-times greater than the average in the 5x5 neighborhood (excluding the neighbors on the same row), the pixel will be replaced with the neighborhood average,

Step 2: Change the hyperstack of the images, select out the channels for Ilastik training and to perform local contrast enhancement for CD99
(vigenettes/cell_segmentation/step2_hyperstack_subtack/preprocssing.ijm).

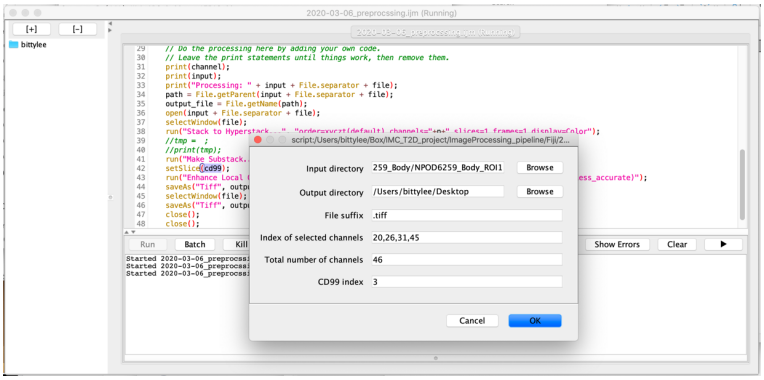
1. Open Plugins > Macros > Edit > Open 2020-03-06_preprocssing.ijm file
2. Click Run button and input the parameters.

Parameters:

1. Input directory: directory containing your images, all files matching the file suffix will be included under that folder.
2. Output directory: where all the finished files will be saved to
3. File suffix: should be unique to the images you want to input into the algorithm. If other files under you input directory also have this suffix, they will be inputted to the algorithm as well and may cause error.
4. Index of selected channels: Specify channels to be selected out of the multi-channel images for Ilastik training, such as membrane channels and DNA channels.
5. CD99 Index (put -1 if no channel needs to be enhanced): For the pancreas dataset, we need to enhance the signal of CD99, so here you need to input the CD99 index contained in the list of newly selected channels.
    a. Enhancement is done through “Enhance Local Contrast (CLAHE) function in Fiji, with blocksize = 39, histogram = 256, maximum = 40 and mask = None. These parameters can be changed directly in the ijm file.

Outputs:

1. An image with the same name as input image but has its hyperstack order changed.
2. Substacked image, with channels selected and one channel potentially enhanced.

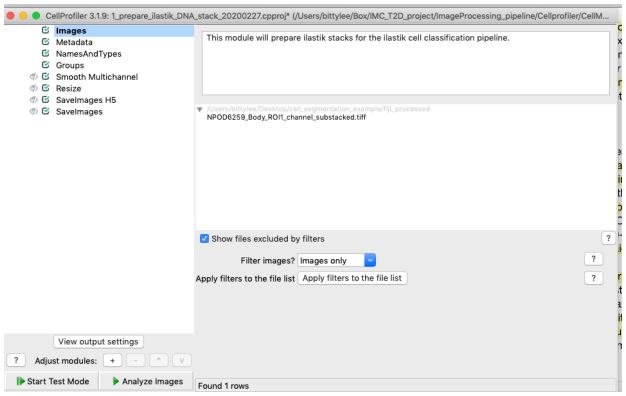


Step 3: Prepare image of selected channels for Ilastik (vigenettes/cell_segmentation/step3_prep_for_ilastik/prepare_ilastik_DNA_stack.cpproj).

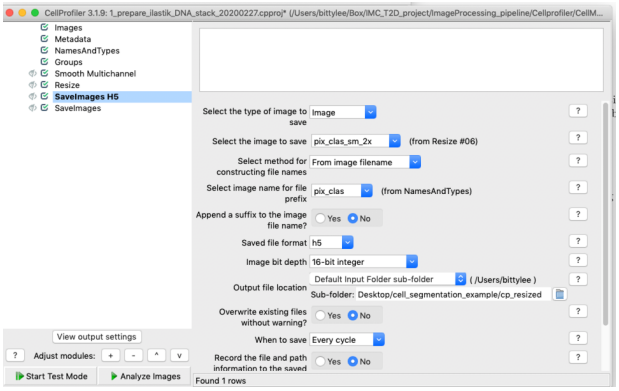
Note: Before you go through the pipeline:

- Open Cell Profiler, go to Preferences.
- Specify CellProfiler plugins directory (available in https://github.com/BodenmillerGroup/ImcPluginsCP).
- Copy-paste the whole ImcPluginsCP/plugins folder to your own working directory.
- Specify default input and output folders
- Click OK
- Exit out of Cell Profile to access IMC Plug-ins. You only have to do this once as long as your plugs folder stays in the same location.

1. Input images to Cell Profiler (these should be substacked images from step 2).



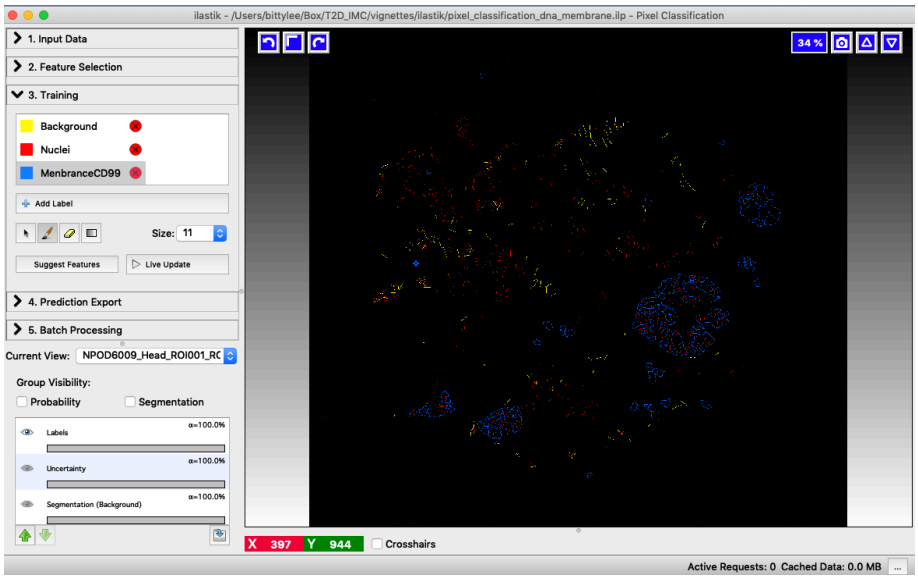
2. Smooth multichannel: This filter will remove strong, single outlier pixels from the images.
3. Scaling up the images 2x.
4. Save the image as a h5 file, remember to modify the output file location.



5. Save the image as tiff file for visualization purpose, remember to modify the output file location.

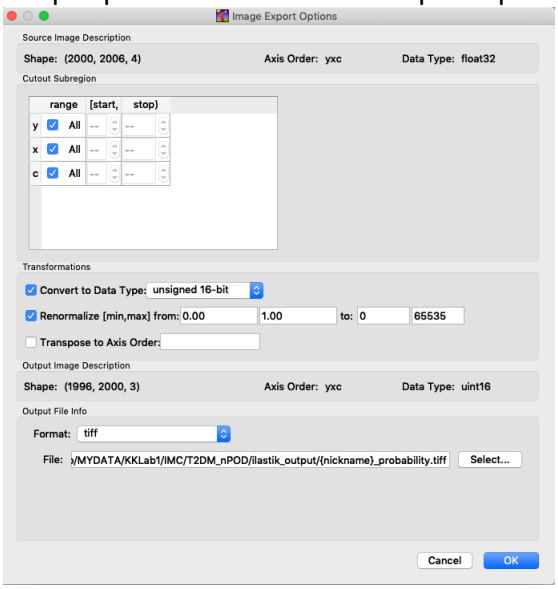
Step 4: Train nuclei-membrane classifier in Ilastik (vignettes/cell_segmentation/step4_pixel_classifier/pixel_classification_dna_membrane.ilp).

1. Load selective .h5 images to ilastik.
2. Edit image property so that “Storage” is “Copy into project file”.
3. Select all 37 features.
4. Train with three types of labels: Background, Nuclei, and MembraneCD99.
    a. Label Nuclei based on DNA channel.
    b. Label MembraneCD99 based on CD99 channel.
    c. Background is all other pixels.
    d. After labeling one image, click live update to train the model. Then move to the next image, click live update to view trained result. For wrongly classified pixels, add in more labels and then retrain the model.



    e. We trained 4 full size images.
5. Output prediction.

a. Output probabilities with the export options shown below.



6. Batch processing.
    a. Load all h5 files and click process all files.

Step 5: Segmentation in Cell Profiler (vigenettes/cell_segmentation/step5_segmentation/segment_ilastik.cpproj).

- Input: probabilities file generated from ilastik. These files should have 3-channels, indicating the probability for each pixel in each class.
- ColorToGray: splits input file into 3 images, one for each class.
- Smooth: smooth over the nuclei probability map.
- IdentifyPrimaryObjects: segment the nucleus using the nuclei probability map, with an expected diameter of 5 ~40 pixels. Minimum cross entropy is used to identify differentiate foreground from background. Specific parameters might need to be readjusted for other tissues/studies.
- Nuclei masks are converted to images and saved in the following modules.
- FilterObjects: nuclei with less than 5 pixels in area are filtered out.
- Smooth: smooth over membrane probability map.
- IdentifySecondaryObjects: expand from primary objects to include the cytoplasm of the cells. Distance-B method is used in which the expansion is guided by the membraneCD99 probability map and a maximum of 10 pixels is allowed.
- FilterObjects: nuclei with less than 15 pixels in area are filtered out.
- Cell masks are converted to images and resized by 0.5, returning it to the original scale.
- Resized cell masks are saved in the following modules.
- Cell outlines can also be saved for visual examination of segmentation performance.
- Tiff file with "_cell_mask_resize_image" suffix is the final segmentation result.

Step 6: Export mean intensity of segmented cells in Cell Profiler (vigenettes/cell_segmentation/step6_segmentation/measure_cell.cpproj).

Input: full size images generated from step 2 and cell mask generated from step 4.

- ConvertImageToObjects: cell mask images are converted to objects
- MeasureObjectIntensity Mutichannel: Mean-intensity value per cell per channel is calculated.
- ExportToSpreadSheet: mean-intensities, cell location, and object number are exported in a csv file. Fill free to adjust what to export by clicking "Press button to select measurements".

Note:

1. The next two steps are ran after merging all mean-intensity results across images and having one big cell-by-protein table.
2. Mean-intensity values exported here are renormalized by the bit depth. Our images are unit16, so mean-intensity values will be multiplied by 65535 (2^16) in R before downstream analysis.

Step 7: Spillover compensation (vigenettes/cell_segmentation/step7_spillover_compensation/spillover_compensation.R).

- Rationale: signal crosstalk between channels sometimes happens and contaminate protein signals in some channels. Therefore, we use CATALYST to compensate for the signal spillover. More details can be found in the original publication (Chevrier et al., Cell Systems, 2018).
- A toy example is included in the folder, with three input files:
    - imc_panel.csv: linking metal tag to protein of interest.
    - Isotope_Purity_Matrix.csv: crosstalk percentage between metal ion channels, provided by Fluidigm.
    - NPOD6009_Head_ROI1.csv: an example cell-protein matrix.
- spillover_compensation.R can be ran on this toy example to demonstrate spillover compensation.

Step 8: Normalization (vigenettes/cell_segmentation/step8_normalization/cell_normalization.R).

- Cells with area <=25 pixels are filtered out.
- For each image, mean-intensity is normalized by 1) log2(value+1), 2) subtracted by the value with highest density in the current image, for each channel, 3) negative values are set to zero.

Step 9 : Clustering (vigenettes/cell_segmentation/step9_clustering/fsom_clustering.R).

- The script is an example of our clustering method, using flowSOM.
- More details about how clustering was done can be found in our manuscript.
- In short, there are two major steps involving the flowSOM algorithm.

- We first run this script on all cells, dividing the cells into three major cell types based on proteins markers: “endocrine cells”, “immune cells”, and “others”.
- The same clustering workflow is done again on each major cell type, and they are re-clustered into 50 clusters for finer annotation.
- We used boxplot showing protein distribution per cluster to guide manual annotation of clusters, as well as mapping the cells back to images.