ISM-2023 Tutorial Workshop: Incorporating Machine learning tools into Image Analysis workflows using Fiji, Ilastik and StarDist

By: Ofra Golani, MICC Cell Observatory, Weizmann Institute of Science

1 Introduction

A typical image analysis workflow includes segmentation of regions and objects (eg nuclei or cells), measuring multiple features for objects and regions, and quantifying relations between objects. Fiji is an open-source software platform based on ImageJ and a collection of compatible plugins focusing on general purpose image analysis for life-sciences. It is scriptable and enables fast prototyping of image analysis workflows. Ilastik, StarDist and other Fiji plugins provide an easy way to exploits recent advances in machine-learning and deep-learning based algorithms as handy components for use within image analysis workflows.

In this hands-on workshop we will introduce you to building image analysis workflows using Fiji, Ilastik and StarDist.

We will NOT cover the basics of image analysis. If you are not familiar with this, we highly recommend you to go through one of the following resources

Mooc on Image Processing and Analysis for Life Scientists by EPFL image analysis team, OR

Introduction to Bioimage Analysis by Pete Bankhead.

Please make sure to follow the *Get Prepared* section (at the end of the document) in order to install the software and download the tutorial data.

Contents

2 Dataset used in the workshop

The Images are cropped from stitched images of osteoclasts in different conditions, imaged with the Hermes automated widefield microscope. Osteoclasts (OCL) tend to fuse together under certain conditions and form multi-nucleated cells. We aim to quantify the osteoclast size and shape and to count the number of nuclei within each cell. Further analysis of interest (not introduced here) include subcellular distribution of the nuclei within the cell (random vs clustered, central vs peripheral), relationship between cell size and number of nuclei etc.

The images have 2 channels:

Ch1:	Osteoclast (OCL)	green
Ch2:	Nuclei	white

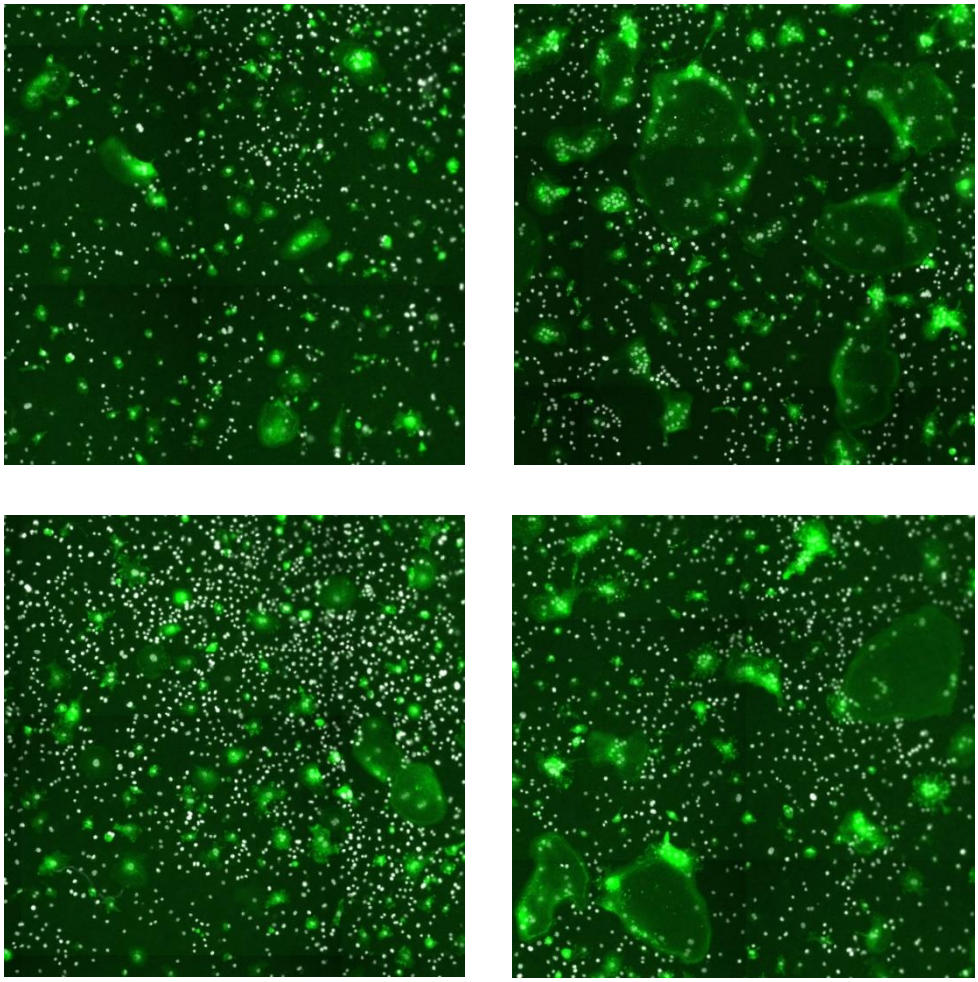
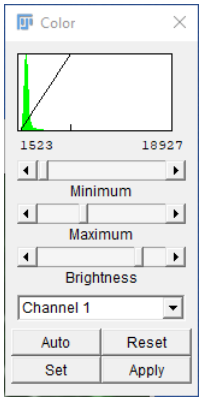


Figure 1: Images used in the workshop. Clock wise starting from upper left corner: A5, B5, C4, A9

3 Warm-up Exercise: Image inspection and proper image comparison

Drag and drop the images: **A5.tif, C4.tif** from the **OCL-Orig** subfolder into Fiji

Put them side by side. Do the following for both images:



- Click the image to select it
- `Analyze>Histogram` : when prompted wether to include all 2 images: click No
- Click on the image you want to analyze, `Image>Adjust>Color Balance…`
- select `Channel 1`
- Find a good values that will allow you to visualize the green osteoclasts cells properly.
- Click `Auto`

Inspect the histograms, and the values set by automatic intensity adjustment for each image and write them down

	Histogram Min	Histogram Max	AutoSet Min	AutoSet Max
A5				
C4				

Is it OK to present the images together in a figure this way? Why?

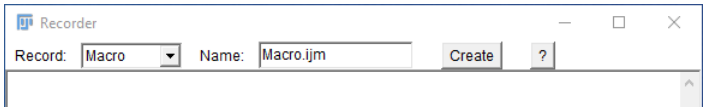
Use the adjust contrast window to select proper min/max value?

What should be the criteria for this? _____

From the contrast adjustment window click the `Set` button and set the minimum and maximum to your selected values. Check "`Propagate to all other 2 channel images`" option and click `OK`.

4 Exercise 1: Segment Nuclei using StarDist

Image: **OCL\A9.tif** or any of the other images

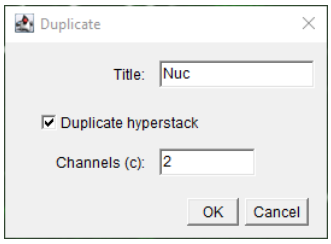


1. Open image in Fiji

2. Fiji has a powerful way to assist you with reproducing results and automation. It allows recording of (almost) every action you do through the graphic user interface (GUI). This can be later on used to create macros to reproduce the set of actions you did.

   Start the recorder: `Plugins>Macros>Record…`

   Make sure `Record` is set to `Macro`



3. To segment the nuclei, we need to have the nuclei channel as a separate image.

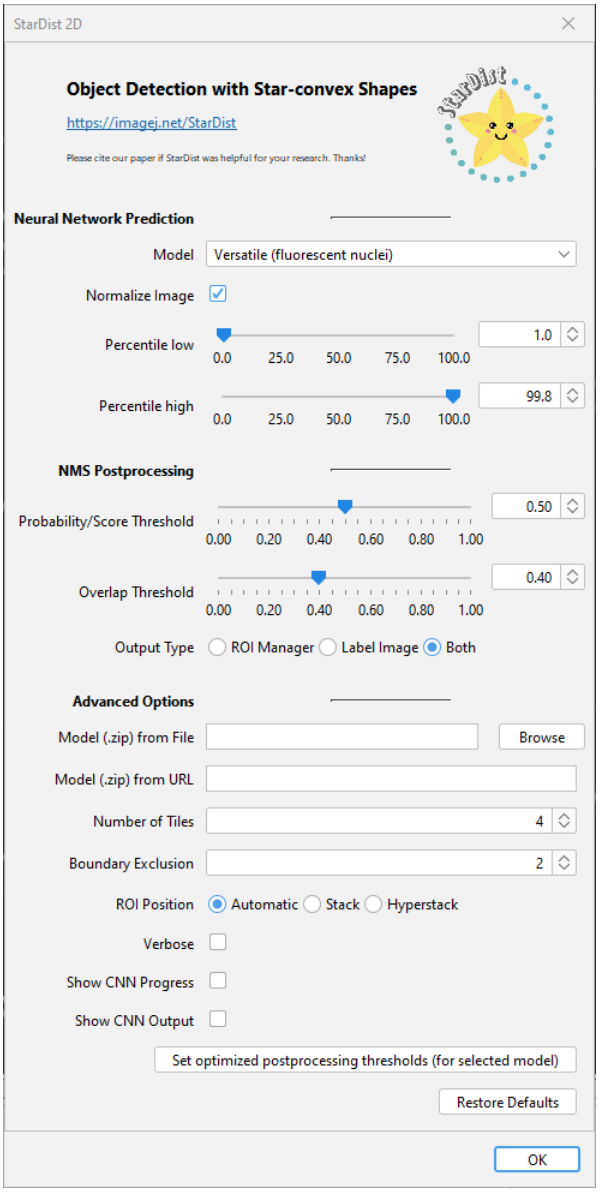
   Duplicate ch2 and rename to `Nuc` : `Image>Duplicate` or `Ctrl+Shift+D` .

4. Select the `Nuc` image and Run StarDist: `Plugins>StarDist>StarDist 2D`

   Use the default settings. You may need to increase the number of Tiles to make it work on your own computer.

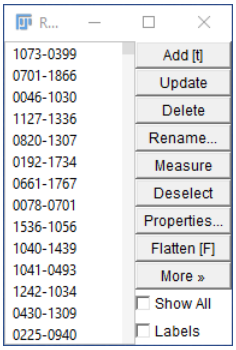
   Be patient, it may take some time, when processing is done you'll get the results in two forms:

   - A label image with unique value for each segmented nuclei
   - A set of Regions of Interest (ROIs) that will be added to the RoiManger



5. Select `Label Image`, rename it using `Image>Rename…` to `NucLabel` (for further steps)

6. Select the original `Nuc` image and click `Show All` at the RoiManger window to overlay the Rois on top of the image.



7. Click the Flatten button for the RoiManger to create a quality control image with the overlay imprinted on.

8. A new image will be create. Save it as `A9_Nuc_Overlay.jpg`

9. From the recorder window, click the `Create` button. The Script Editor window will be opened (or a new tab will be added if it is already open). Save the newly created script as `MySegmentNuc.ijm`

10. **Inspect Results (on top of original image)**

    You can

    - Show/hide the ROIs on top of the original image and/or
    - Place Nuc and Label Image side-by side, and use `Analyze>Tools>Synchronize windows` to point to the same pixel on both images. (make sure to select both images)

- Do we miss any nucleus
- Do touching nuclei merge into a single one
- Is there a nucleus that get over segmented
- Inspection of results is extremely important
- To assist inspection create quality control image
- (if needed) How can we improve?

Note: StarDist comes with out-of-the-box ready models for fluorescent and H&E stained nuclei. You can select them by setting Model in the GUI. You can (re)train your own model

Further StarDist information:

Introduction to nuclei segmentation with StarDist - [NEUBIASAcademy@Home] Webinar: https://www.youtube.com/watch?v=Amn_eHRGX5M

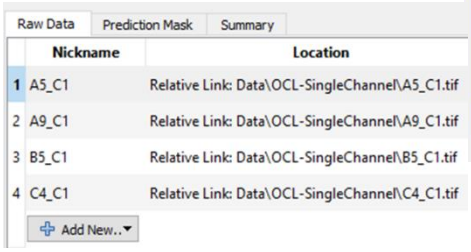
FIJI (ImageJ): Segmentation with StarDist: https://www.youtube.com/watch?v=xkViEAFf6Bs

5 Exercise 2: Train Ilastik Pixel Classifier to detect osteoclasts pixels

Images: **OCL-SingleChannel\{A5, A9, B5, C4}_C1.tif**

1. Start Ilastik
2. Create New Project: Pixel Classification

   Save it into MyOCLPixelClassifier.ilp

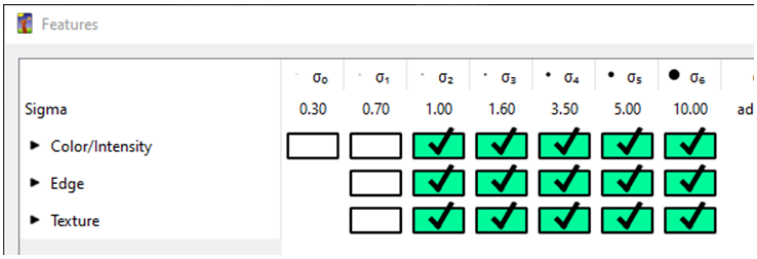


3. Add input Images: click on `Add New…`

   And select all the C1 Images

4. Feature Selection:

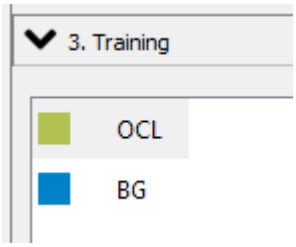
   Click on `Select Features…` and choose all the σ values between 1-10



5. Set Training Classes: `OCL`, `BG`

   Double click on the class name to edit it

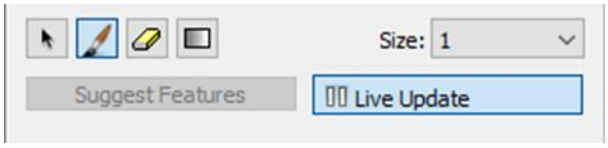
   Double click on the color box on the left to set the color



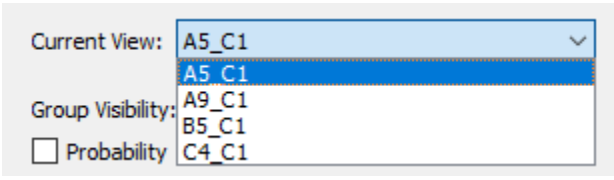
6. Interactively, Iteratively Train ilastik:

   a. **Draw / Correct** labels for each of the classes

   b. **Live Update** – will retrain the classifier using all the available labels



   c. Switch images – don't try to achieve perfect performance for specific image, but rather aim for good performance on an image, and then switch to another image. After going through all the training images, go back to the first image and see if any corrections are required. This strategy will help you to iteratively improve the classifier performance and generalization. Use the Current View drop down menu to switch between images.

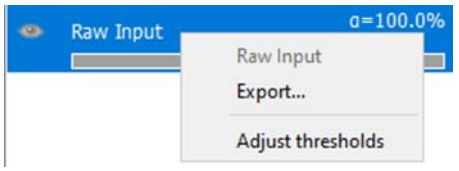


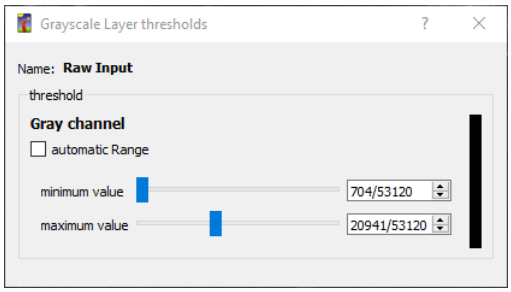
 Navigation mode,  Drawing mode,  Eraser,  Contrast adjustment

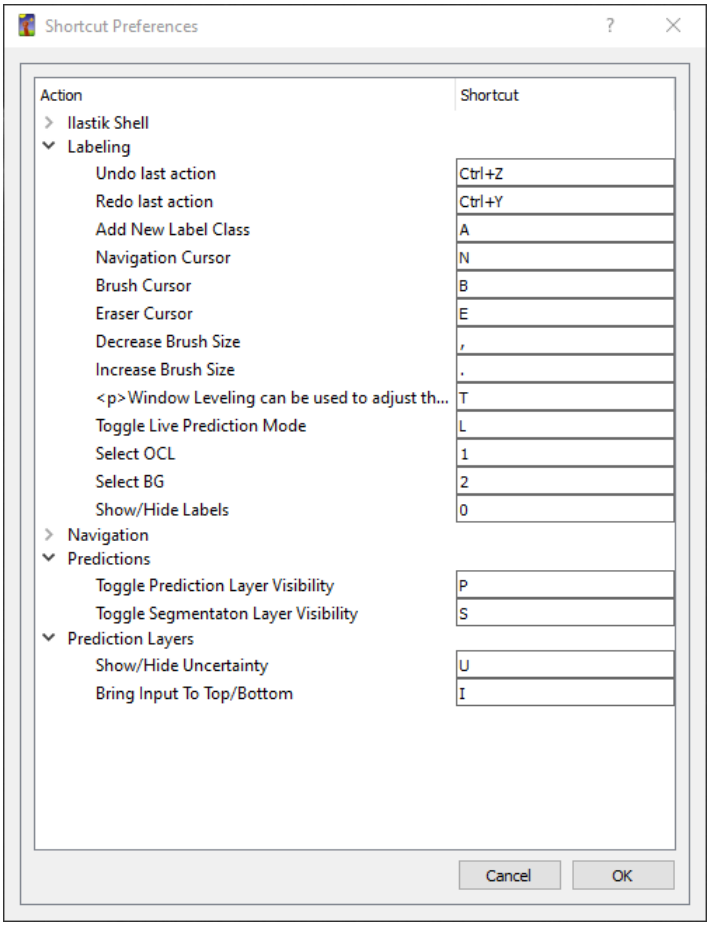
7. Save Project: `Project>Save` Project or `Ctrl+S`

Tips:

- **(de)activate Live Update**: to save time and prevent Ilastik from updating after each and every drawing, deactivate Live update, make few corrections and then activate it again to the effect of your corrections
- **Choose your drawing budget wisely**: if you see that certain type of appearance is correctly identified, don't add drawings (labels) for this appearance. The total number of pixels in your labels affect the time it takes to train the classifier. To make your training more efficient, add labels for regions that were not classified correctly.
- **Adjust contrast**: the big OCLs are quite deem. To see them better you should adjust the contrast. You can do this either by using the contrast adjustment icon and then dragging the mouse while holding the left mouse button.
  Alternatively you can right-click on the `Raw Input` image (usually at the bottom), select `Adjust thresholds`
- **Keyboard shortcuts** are very useful, you can see them from the `Settings>Keyboard shortcuts`.







<u>Ilastik video tutorials:</u>

Tutorial Image Analysis with Machine Learning Basic Principles and Examples Using Ilastik 28 03 2021, by Anna Kreshuk: https://www.youtube.com/watch?v=K0b4209NxP0

Pixel Classification using Ilastik, by Anna Klemm: https://www.youtube.com/watch?v=QfMO9L3ZWto

6 Ex 3: Apply Ilastik Classifier from Fiji

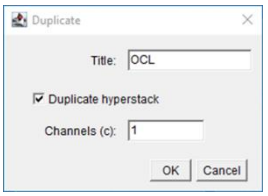
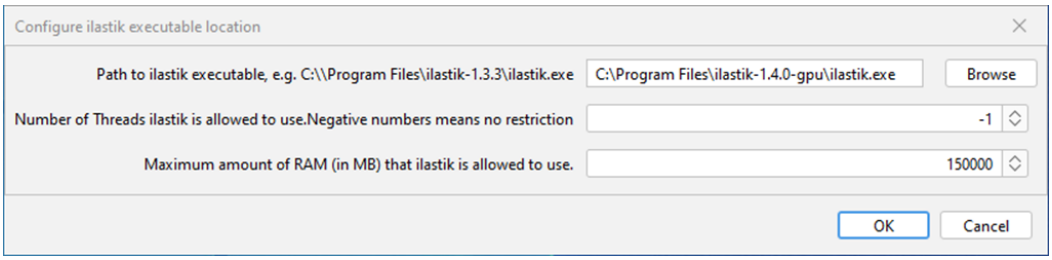


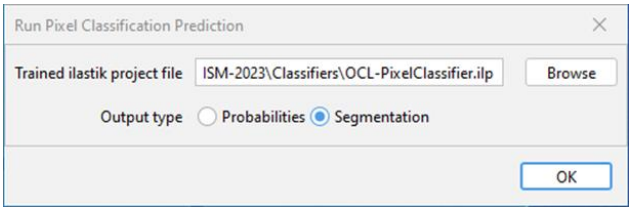
Image: **OCL\A9.tif** or any of the other images

Once you have a trained Ilastik classifier, switch back to Fiji

1. Open image in Fiji
2. Start recorder: `Plugins>Macros>Record…` or clear it if it is already open.
3. Duplicate ch1 and rename to `OCL` (`Image>Duplicate` or `Ctrl+Shift+D`)
4. Initialize Ilastik:



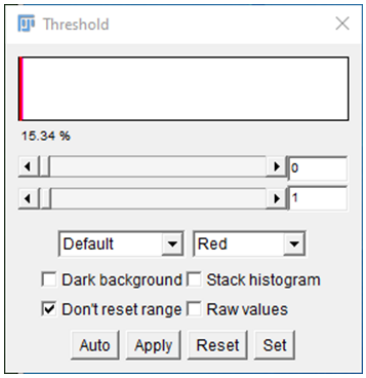
   `Plugins>Ilastik>configure Ilastik executable location`
   Use the `Browse` button to select *Ilastik.exe* file from the folder in which Ilastik is installed on your own computer.
5. Apply Ilastik classifier: `Plugins>Ilastik>Run Pixel Classification Prediction`
   Use the Browse button to select the classifier that you trained in Ex2.
   Alternatively you can use the catch-up classifier provided with the workshop material `Classifiers\OCL-PixelClassifier.ilp`
   Be patient, it may take some time to get Ilastik results.
   You may see red warning lines in the console window, they are usually OK



6. Select Ilastik output image, `Image>Rename…` to `ilastikLabelImage`
7. As you have only 2 classes, the output image will appear black.
   Change LUT of the output image: `Image>Lookup Tables…>glasbey on dark`

7 Ex 4: Use Ilastik output to Segment Osteoclasts

Ilastik classify each image pixel as being OCL or not. However in order to work with cell objects we need to combine adjacent pixel together into single cell object and ignore non-cell objects based on size and shape.

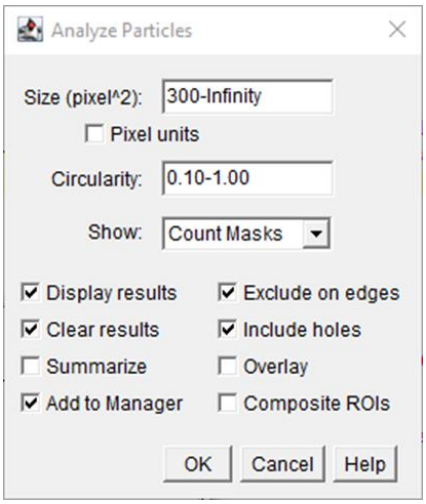


1. In Ilastik segmentation image all pixels of each class have different (integer) numeric value. In our case there are two classes, so the image have pixels with values of either 1 or 2. As a first step we need to convert the label image into a Binary mask, in which OCL pixels have value of 255, and background pixels have zero value.

   `Image>Adjust>Threshold…`, use `Set` to select the range to be [0,1] , click `Apply`

2. Connected Component Analysis: `Analyze>Analyze Particles…`

   Set values for minimal size to discard too small objects and minimal circularity to discard very elongated objects.

   Note to set the `Show` value to `Count Masks` .

   This will create a label image of the final created objects in addition to adding the objects as ROIs to the Roi manager. This is needed for the next step in which we will associate nuclei objects to cell objects.

   It is possible also to create it from the RoiManager eg using `Plugins>BIOP>ImageAnalysis>ROIs>ROIs to Label Image`

   Note: in this case the images are not scaled to real units. So the size filters are set as number of pixels. In general case, it is best to set it according to size in real units like um^2

3. Select the Count Mask image, `Image>Rename…` to `OCL_Label` (for further step)

You will now show the results on top of the original image, flatten it, to create a quality control image that can be saved for further inspection.

4. Select the `OCL` window
5. From RoiManager : `Show All`
6. `Flatten` to imprint the overlay on the image. This will create a new RGB image
7. Save the flatten image into `A9_OCL_Overlay.jpg`: `File>Save As>Jpeg…`

8. From the recorder: `Create` draft macro: Save it as `MySegmentOCL.ijm`

9. Inspect Results (show/hide on top of original OCL image):

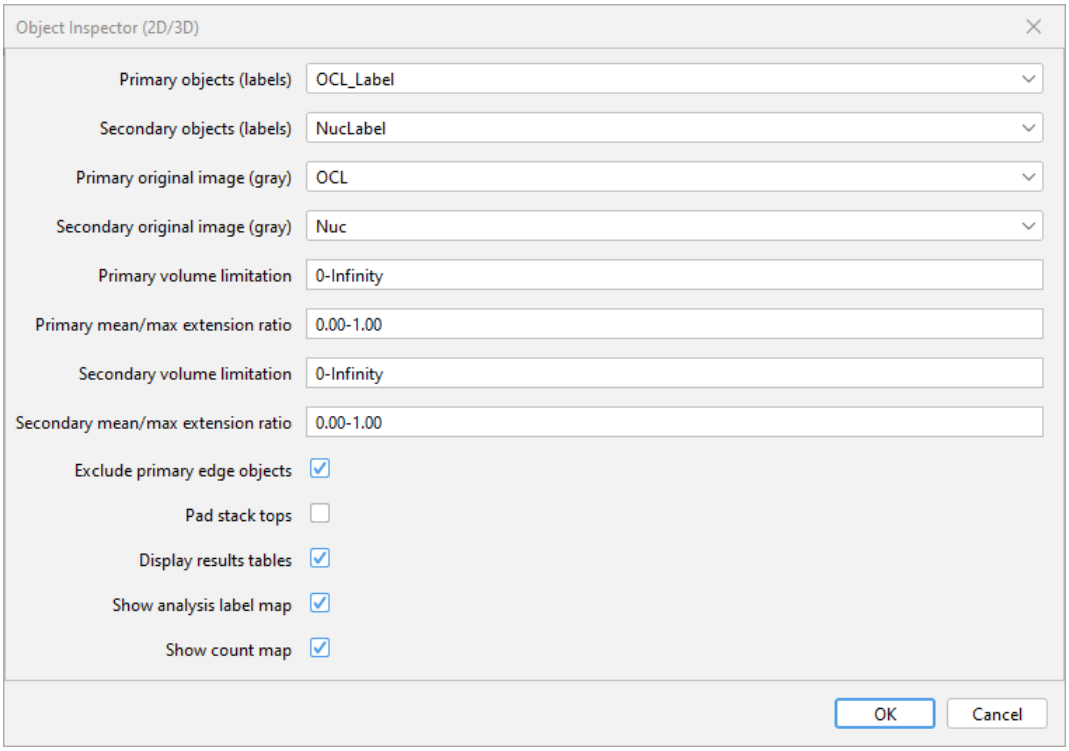
- Do we miss any OCL?
- Do touching OCL merge into a single one?
- Is there an OCL that get over segmented?
- How can we improve?

8 Ex 5: Associate Nuclei to Osteoclasts

You should now have two label images: one for Nuclei (`NucLabel`) and one for OCL (`OCL_Label`). If you are missing any of them, you can easily get to this stage by following catch-up step below.

Our current goal is to associate each nuclei to its parent OCL and discard all nuclei that are not within OCLs. There are many ways to achieve this in Fiji, we will use `Object Inspector` plugin which is one of the easiest ways to do this.

1. Run `BioVoxxel 3D Box>Analysis>Object Inspector (2D/3D).`
   It will create three images and two tables.



2. Select `CountMap_OCL_Label`
3. Show calibration bar: `Analyze>Tools>Calibration Bar…`

Catch-up step (only if you don't have both label images opened

If you are missing the label images:

a. Close all images : File>Close All
b. Close tables / RoiManger
c. Drag & drop an image into Fiji
d. Clear the recorder window or reopen it: `Plugins>Macros>Record…`
e. Drag and drop `SegmentNuc.ijm` and `SegmentOCL.ijm` from the `Macros` subfolder into Fiji.
f. Select the image A9.tif
g. Run `SegmentNuc.ijm`
h. Select the image A9.tif
i. In `SegmentOCL.ijm` macro tune the following lines to match your own computer

```
var IlastikExeLocation = "C:\\Program Files\\ilastik-1.4.0\\ilastik.exe";
var maxRAM = 24000;      // MB , set to 50-80% of available RAM
var pixelClassifierLocation = "D:\\YourFolder\\YourClassifier.ilp";
```

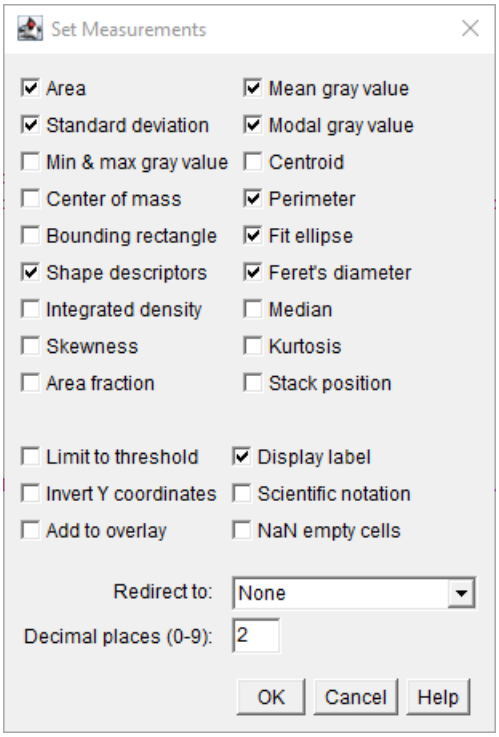
j. Run `SegmentOCL.ijm`

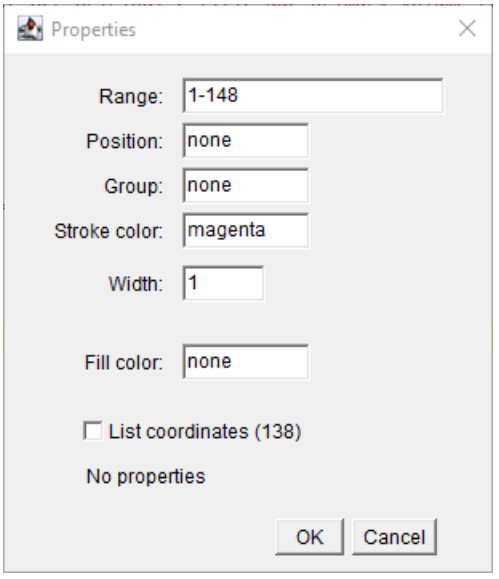
9 Ex 6: Save Results

We will now record all the steps needed to save output files:

- Count Map with overlay of the OCLs (to make sure we see OCL with zero nuclei) and calibration bar
- Overlay image with both nuclei and OCL shown on the original image
- Results table with one line for each OCL with shape measurements and File name. The Mode column will show the nuclei count
- Final OCL ROIs
- Final Nuc Rois

1. Make sure Recorder is activated
2. Select `CountMap_OCL_Label`
3. Display calibration bar: `Analyze>Tools>Calibration Bar...`, Flatten, Save
4. Select `CountMap_OCL_Label`
5. Rename to `A9_CountMap_OCL_Label`
6. `Analyze>Set Measurements…`
7. Close Results table if it is open
8. From the RoiManager: click `Measure`
9. Save `Results to A9_Results.csv`
10. From the RoiManager: `More>Save… A9_OCLRois.zip`
11. Select original image (A9.tif), from RoiManager `Show All`, `Flatten`
12. Close RoiManager
13. Select final_NucLabel, Plugin>BIOP>Image Analysis>ROIs>Label Image to ROIs
13. From RoiManager: Deselect, Properties…
14. Select the output flatten image of step 10, From the RoiManager:`Show All`, `Flatten`
15. Save new flatten image to `A9_FinalOverlay.jpg`
16. From the Recorder: `Create` macro save it as `SaveResults.ijm`





10 Guidelines for automating your work

The recorded macros include almost all the steps needed to reproduce the actions on the same image.

Few easy steps are required to make them more generalized so they can be applied to any image.

Once we have a general macro for a single image, it is then quite easy to pack it in a function and use the available template to make it into a macro that runs for all images in a given folder.

Teaching macro writing is beyond the scope of this workshop.

We will give below some guidelines that will assist you with the above, and links that you can use to learn more.

The workshop material includes both macros discussed above which you can use to reproduce all the steps that we did:

- `QuantifyOCL-SingleFile.ijm`
- `QuantifyOCL-WholeFolder.ijm` – which we will use in the next exercise

10.1 Key points for automating your code

- While recording rename temp images to fixed names: eg Nuc, IlastikLabelImage
- Record everything, including results saving
- Start with recorded code : from the recorder click Create
- Replace hard-coded values by variables (var): Ilastik path, classifier path and name, all numbers (min/max values).

  var precedes the variable name as in the example lines below and generates global variable.

```
var IlastikExeLocation = "C:\\Program Files\\ilastik-1.4.0\\ilastik.exe";
var maxRAM = 24000;      // MB , set to 50-80% of available RAM
```

- Store original image name and path for further image saving : getTitle
- Remove redundant lines
- Try to run on the same image and debug
- Try to run on another image
- Optionally pack in a function (ProcessFile)
- Once you have working code for single image, Use template code example for going over all image
- Add documentation

10.2 Converting single file macro into a macro for whole folder

- From the script editor: Templates>ImageJ1.x>Batch>Process Folder (ImageJ Macro)
- Open the single file macro (`QuantifyOCL-singleFile.ijm`)
- Copy the variables sections to just above the `processFolder` call
- Copy the rest of the code into the `processFile` function
- Fix indentation (select all and use Tab to indent inside)
- Add command for opening the file

```
open(input + File.separator + file);
```

Resources for self-learning ImageJ Macro writing:

Anna Klemm's video on ImageJ/Fiji Macro Language - [NEUBIASAcademy@Home] Course

Robert Haase's video on: "In introduction to ImageJ macro, variable and operations including best practices in writing code".

Ofra Golani's Hands-on exercises for ImageJ/Fiji macro writing

ImageJ Macro Language Reference

11 Ex 7: Apply to all images

1. Drag and drop the macro `QuantifyOCL-WholeFolder.ijm` into Fiji
2. From the Script Editor, Click `Run`
3. When prompted, select the folder in which all the images are located
4. Results will appear in `Results` subfolder
5. Compare color coded count maps.

   How can we make fair comparison?

12 Get Prepared

1. **Install Fiji**

    a. Download from https://imagej.net/software/fiji/downloads and unzip

    Caution: "Program Files" not recommended! on Windows. We strongly recommend that you store your directory somewhere in your user space eg C:\Fiji.app rather than C:\Program Files

    b. Create desktop shortcut

2. **Install Fiji plugins**
    a. Start Fiji
    b. Help > Update…
    c. Click Manage update sites…
    d. Select the following update sites (keep the default ones)
        i. 3D ImageJ Suite
        ii. BioVoxxel
        iii. BioVoxxel 3D Box
        iv. Clij
        v. Clij2
        vi. Clijx-assistant
        vii. Clijx-assistant-extensions
        viii. CSBDeep
        ix. DeepImageJ
        x. IJPB-plugins
        xi. Ilastik
        xii. PTBIOP
        xiii. StarDist

    e. Click Close
    f. Click Apply Changes
    g. Restart Fiji

3. **Update ImageJ**

    a. Help>Update ImageJ…

    b. Upgrade to v1.54d

    c. Restart Fiji

4. **Install Ilastik**

    a. Download Ilastik version 1.4.0 from https://www.ilastik.org/download.html

    a. Install the downloaded exe

    b. Create a desktop shortcut

5. **Download Workshop Material** to your computer from the provided link

13 Acknowledgement

The images used throughout this workshop are courtesy of Sabina Winograd-Katz and Benny Geiger.

If you use any of these software make sure to cite them:

Fiji:

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., … Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods*, *9*(7), 676–682. doi:10.1038/nmeth.2019

Ilastik:

interactive machine learning for (bio)image analysis
Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht & Anna Kreshuk
in: Nature Methods, (2019) Link at publisher, BibTex file

StarDist:

Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers.
Cell Detection with Star-convex Polygons.
International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Granada, Spain, September 2018.

BioVoxxel

https://biovoxxel.github.io/bv3dbox/ , doi: 10.5281/zenodo.7691609

MorphoLibJ

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
MorphoLibJ's code repository has its own DOI.

Clij2

Robert Haase, Loic Alain Royer, Peter Steinbach, Deborah Schmidt, Alexandr Dibrov, Uwe Schmidt, Martin Weigert, Nicola Maghelli, Pavel Tomancak, Florian Jug, Eugene W Myers. *CLIJ: GPU-accelerated image processing for everyone*. Nat Methods 17, 5-6 (2020) doi:10.1038/s41592-019-0650-1