

ImageJ Lab Walkthrough

Information courtesy of Edward Evans , Michael Nelson, Ellen Dobson, as well as many others involved in ImageJ/FIJI development and production of educational materials, walkthroughs, and demonstrations.

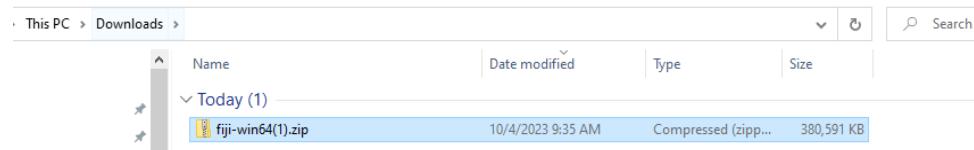


Installing ImageJ/FIJI	2
A Side Note on Updating	2
Install Plugins with Update Sites	3
Downloading a Practice Dataset	3
Common Keyboard shortcuts	4
Getting Started, Opening an Image	5
Starting with a basic image	5
Opening a Z-Stack	5
Hyperstacks	5
Duplicate A Channel Or Image	5
Adjusting Brightness and Contrast	6
Changing Channel Colors	6
Caution with Changing Image Types	7
3D Volume and Projections	7
Bit Depth	8
Set Scale and Scale Bars	9
Inspecting Images	11
Inverting the background	12
Changing the color scheme (LUT)	12
Colorblindness Consideration	13
Plot/Line Plot	13
Background Subtraction	14
Segmentation/Thresholding	14
Adding Selections to ROI Manager	16
Analyze Particles	16
Masks For Measurement	17
Making Measurements	18
Stitching Images	19
Scripting	21
Macro Recorder	21
Why does the recorder not show a response for all plugins?	22
Batch Processing	22

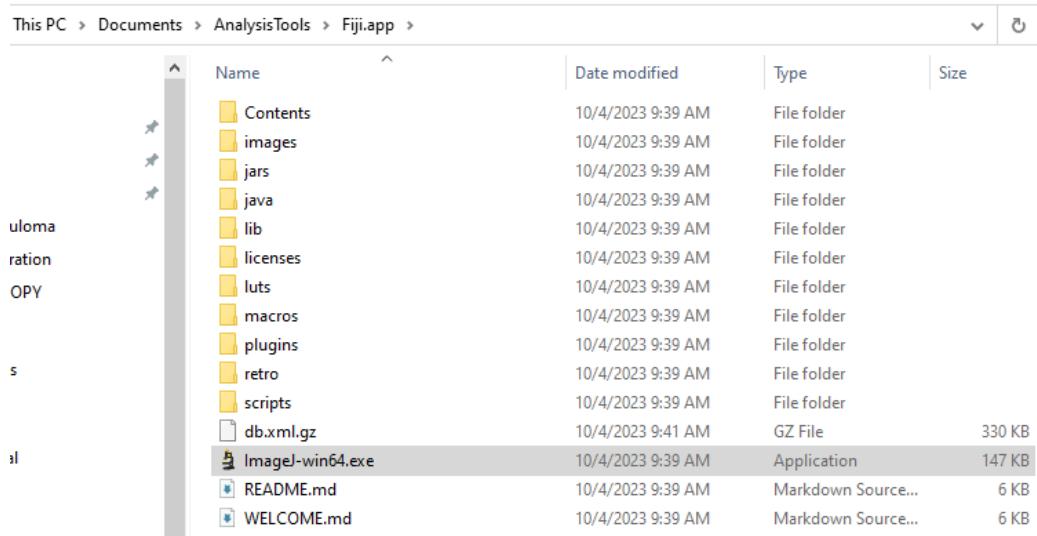
Deconvolution	24
Background Subtraction - Gaussian filters	26
Trackmate - Example from documentation	29
StarDist - Basic Example	36
Other Resources	39

Installing ImageJ/FIJI

1. Download FIJI package from the software site. FIJI has pre-installed plugins and is recommended for this lab.
 - a. Fiji: <https://imagej.net/software/fiji/downloads>
 - i. Look for the zip folder and extract the contents to the desired location on your computer. Keeping it in “Program files” is generally not recommended to avoid access/security issues.



- ii. Open the program with the “.exe” file



- iii. You may be prompted to start the updater after first opening the program

- b. Alternative to install ImageJ base package: <https://imagej.net/ij/download.html>

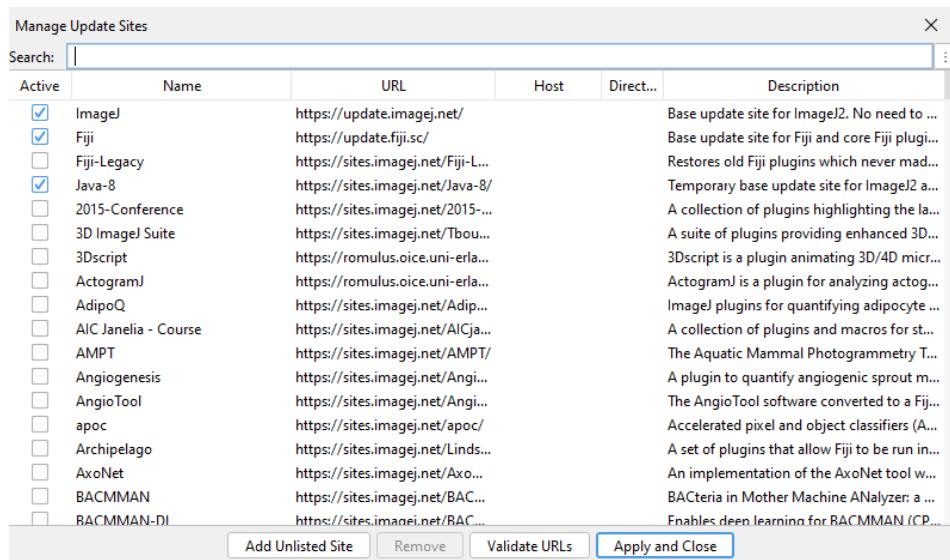
A Side Note on Updating

When using FIJI, it is recommended to use the **Help > Update** option rather than **Help > Update ImageJ**. Update ImageJ will specifically update only the base ImageJ package within FIJI, not the associated plugins. This makes it more likely that you will run into

version issues and other update problems. Generally, updating when prompted upon opening FIJI will also prevent issues.

Install Plugins with Update Sites

1. After running **Help > Update** there will be an option to **Manage Update Sites**. This provides access to common plugins that are compatible with FIJI. Use the checkbox to add a plugin, then click apply and close.



2. Click **Apply Changes**, then restart FIJI
3. You should now be able to search for and run the plugin

Downloading a Practice Dataset

The complete list of files may be downloaded at the following links:

Google Drive:

https://drive.google.com/drive/folders/1Pl-BitlQb2j0bm2s35dp70WV9USqiT0_?usp=sharing

Github:

https://github.com/hwilson23/FIJI-ImageJ_Lab_Tutorial

Common Keyboard shortcuts

+ - zoom in
- - zoom out

Ctrl/Command + S - save

*no shortcut for “save as”

Ctrl/Command + Shift + C - brightness and contrast

Ctrl/Command + Shift + Z - color channel control

Ctrl/Command + Z - undo

Ctrl + H - histogram *just H for Mac

Ctrl/Command + K - plot/line profile

Ctrl/Command + L - focus search bar

Ctrl/Command + M - measure

Ctrl/Command + Shift + T - threshold

Shift - to draw straight lines

Alt + arrow keys - to change current selection size

T - open ROI (region of interest) manager, and will add current selection as ROI

Ctrl/Command + Shift + W - close all

Also helpful:

Window > tile - will montage all open images/windows

Image> Transform - flip or rotate images

Help> Update - to update FIJI

Getting Started, Opening an Image

The easiest way to open a data set is to click-and-drag from the folder. However, some file types need to be specifically imported. To import a sequence of images that are saved within a folder, you can use the **File > Import > Image Sequence** option to open them as a stack or as separate images.

Additionally, using a **Virtual Stack** in the **Bio-Formats Importer** may be useful if trying to view large data files. This makes it easier to get a quick look at data when there is limited RAM. However, if you need to perform an image operation, such as background subtraction, then more memory will be required as the data needs to be loaded.

Starting with a basic image

1. Open “blobs.gif”
 - a. Also available though **File > Open Samples > Blobs**, or **Ctrl + Shift + B**

Opening a Z-Stack

1. Open “t1-head.tif”
 - a. Also available through **File > Open Samples > T1-Head (16-bits)**
 - b. Use the bar to scroll through the stack of images
2. We can also display the orthogonal view by using **Image > Stacks > Orthogonal Views** or **Ctrl/Command + Shift + H**
 - a. The yellow crosshairs can be used to change the display for each orthogonal view

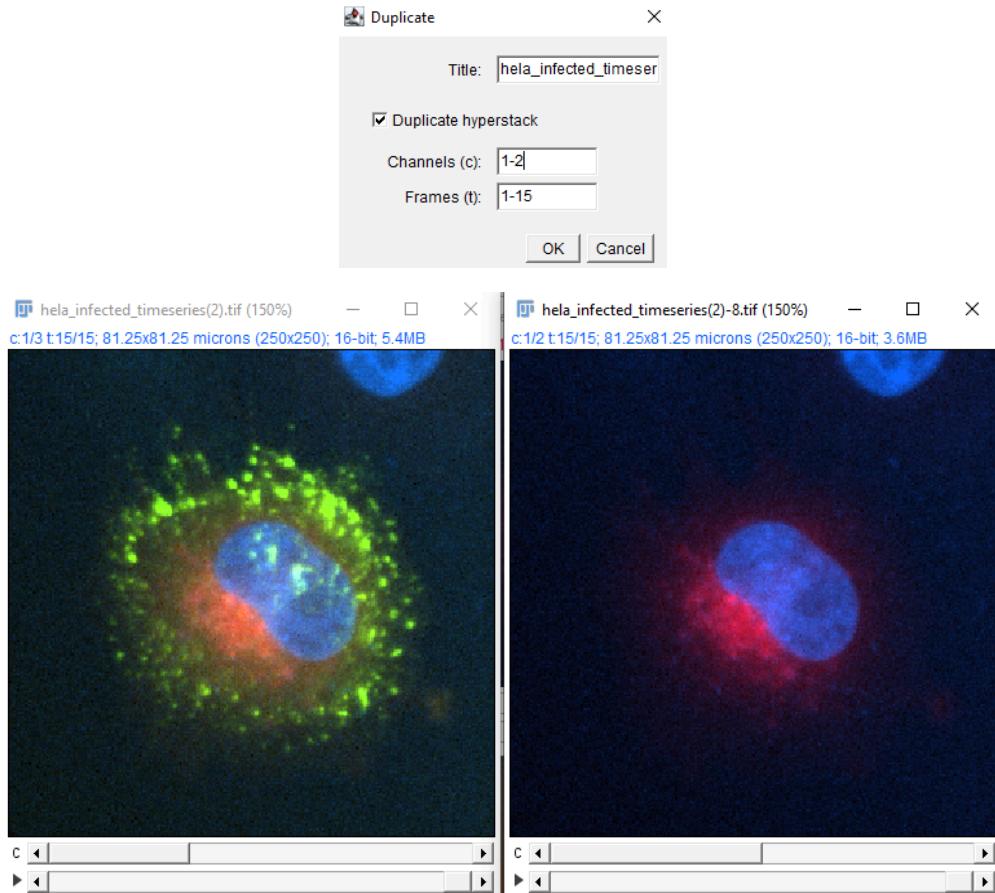
Hyperstacks

Hyperstacks are multidimensional data, such as a multichannel timeseries or z-stacks.

1. Start by opening the “hela_infected_timeseries.tif” image. This is available for download at: <https://media.imagej.net/workshops/data/3d/>

Duplicate A Channel Or Image

- a. Use **Image > Duplicate** to bring up the duplication options window. This can also be done with **Ctrl + Shift + D** or **Right Click > Duplicate**. Here, we can specify which RGB channels and timepoints we want to separate. If we use 1-2 in the channels menu, it should produce an image of just the red and green channels. The **duplicate hyperstack** box should be checked.

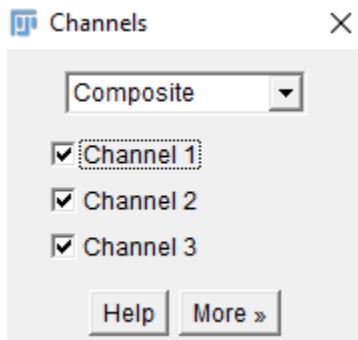


Adjusting Brightness and Contrast

- Changing the contrast in an image can be a good visualization tool. Using **Ctrl + Shift + C** or **Image > Adjust > Brightness and Contrast** will display the B&C menu. Using Auto or the B&C sliders is okay for visualization, but **using the Apply button will change the pixel values, which generally should not be done.**

Changing Channel Colors

- To change the display color of channels, use **Image > Color > Channels Tool** or **Ctrl + Shift + Z**.



- a. The “more” option can be used to apply colors to the selected channels.

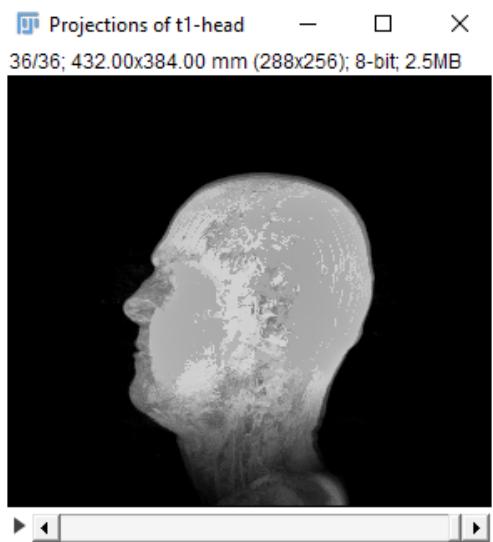
Caution with Changing Image Types

- a. Although it may be useful in specific cases, changing the image type through **Image > Type** can result in the loss of information as we will see in a later example. Not all image types can be converted back to the original. The following error is likely to display if you are not able to complete a type conversion.

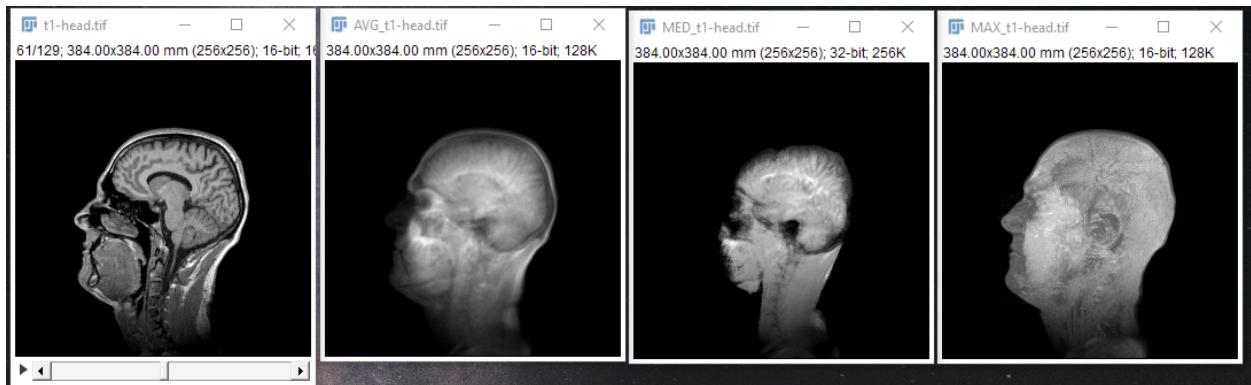


3D Volume and Projections

1. Open t1-head.tif, **File > Open Samples > T1 Head (16 bits)**
2. To display the 3D Volume, use **Image > Stack > 3D Projection > Click ok**
 - a. In 3D projection dialogue box, there are various options for projection including axis of rotation and setting the slice spacing (step size) which may be useful for known parameters



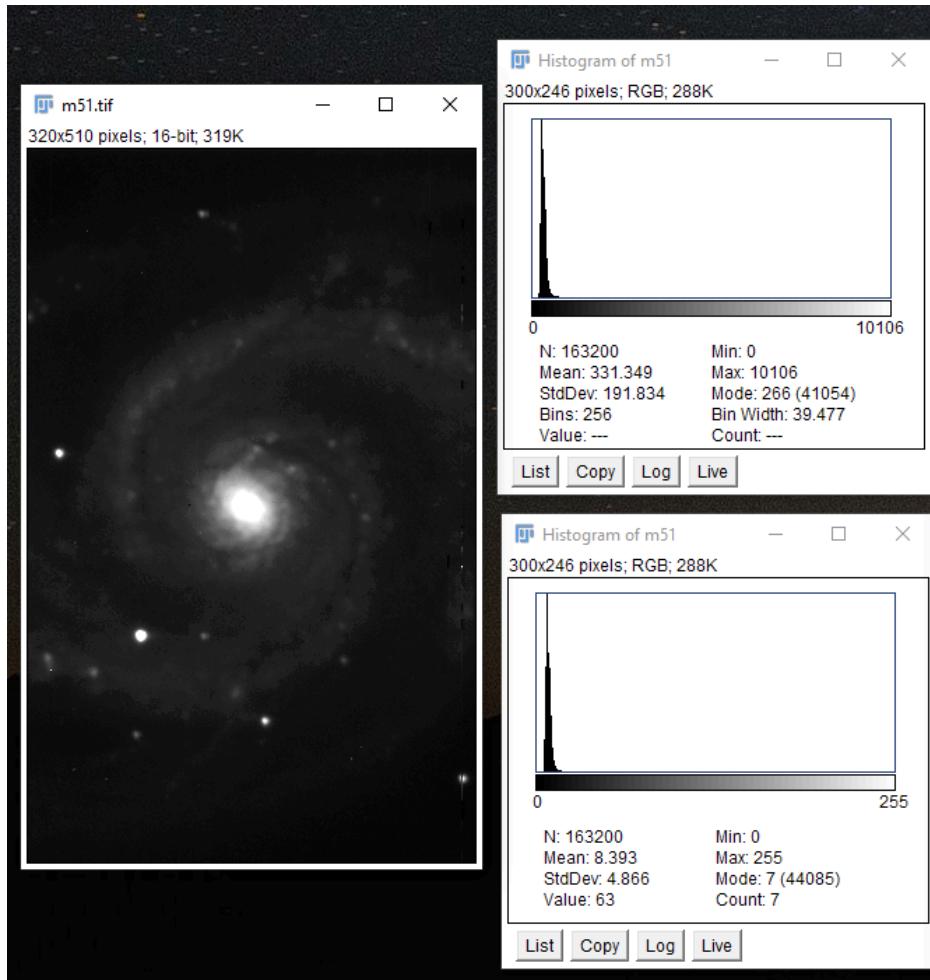
3. Additionally, the original image stack may be projected along one axis using **Image > Stacks > Z Project**. It is important to note that the information in the projection is highly dependent on the display parameter. A few examples are shown below (left to right: single slice of the stack, average projection, median projection, max projection).



Bit Depth

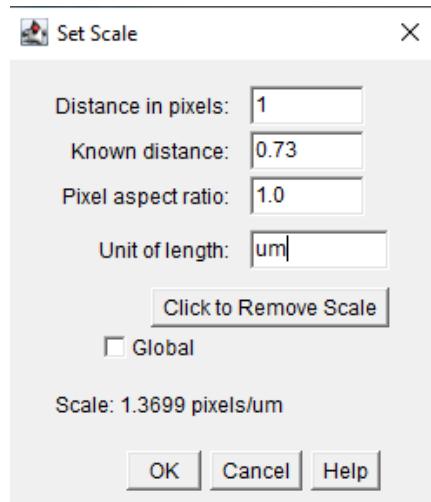
1. Open the “m51.tif” image, **Open > Open Samples > M51 Galaxy (16 Bits)**.
2. Adjust the contrast so we can see the image a bit better. **Image > Adjust > Brightness and Contrast**.
3. Make a histogram of the image, **Ctrl + H**, and notice the scale of the histogram and the mean value. Keep this window open.
 - a. 16-bit images have a maximum value of 65,536.
4. Convert the 16 bit image to an 8 bit image, **Image > Type > 8 bit**.
 - a. 8-bit images have a maximum of 256 values. This means the data resolution and range is lower compared to a higher bit depth, so the data is compressed.
5. Make another histogram and keep the window open. How did the values change?

6. Convert the 8 bit image back into a 16 bit image and make another histogram. Do you still have the original data?



Set Scale and Scale Bars

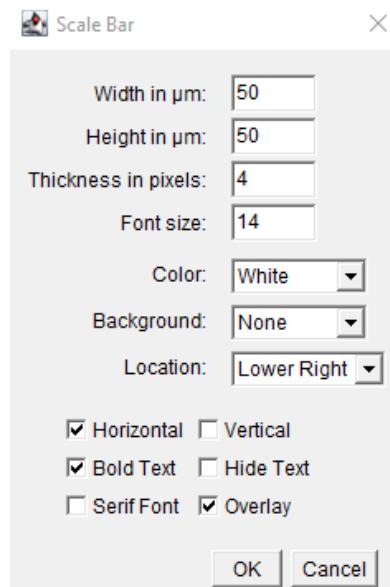
1. If the size per pixel is known, such as from a microscope calibration, this can be used to change the image dimensions from pixels to the known field of view.
2. Use **Analyze > Set Scale** to enter the known distance. For example, an image with a pixel size of 0.73 pixels/micron would be entered as the following image.



- Further info for setting the scale based on a measurement in an image can be found here:

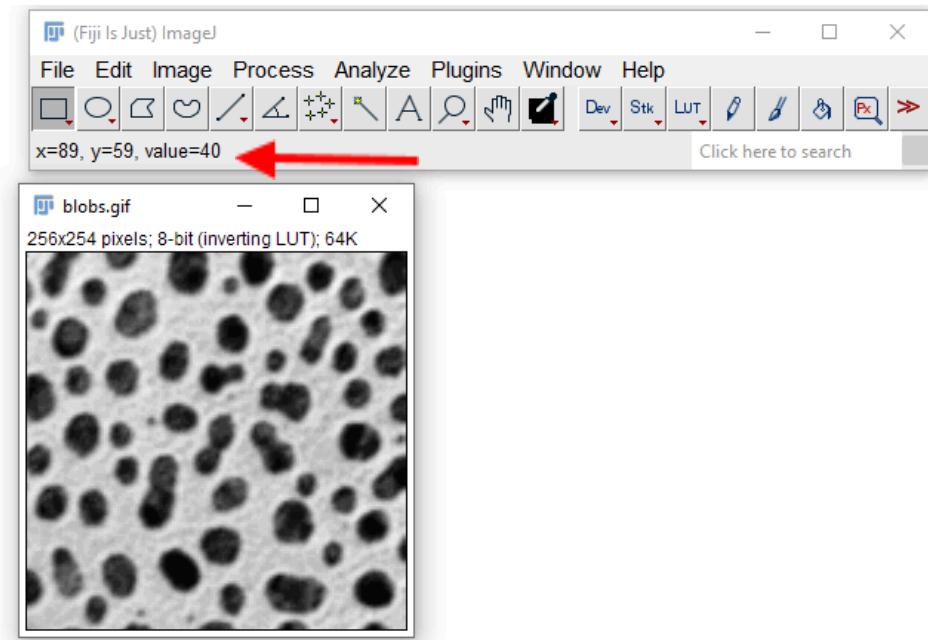
https://serc.carleton.edu/eyesinthesky2/week2/get_to_know_imagej.html

- To add a scale bar, use **Analyze > Tools > Scale Bar**.



Inspecting Images

1. Hover over an area in the image. In the status bar of Fiji, you should see the pixel you are on, with xy coordinates, and its associated value.
 - a. In an RGB type image, Fiji will also tell you the associated color values for the red, green and blue channels.



2. To see a larger area, use the **Pixel Inspection Tool**
 - a. In the Fiji window, click the pixel inspection icon
 - b. *If the icon is not present, you can use the double red arrows to add the icon, by opening the list and selecting “Pixel Inspector”*



- c. This brings up a window of pixel values that can be moved around the image. To adjust the parameters of this window, click **Prefs**. This can be used to adjust the size of the window.

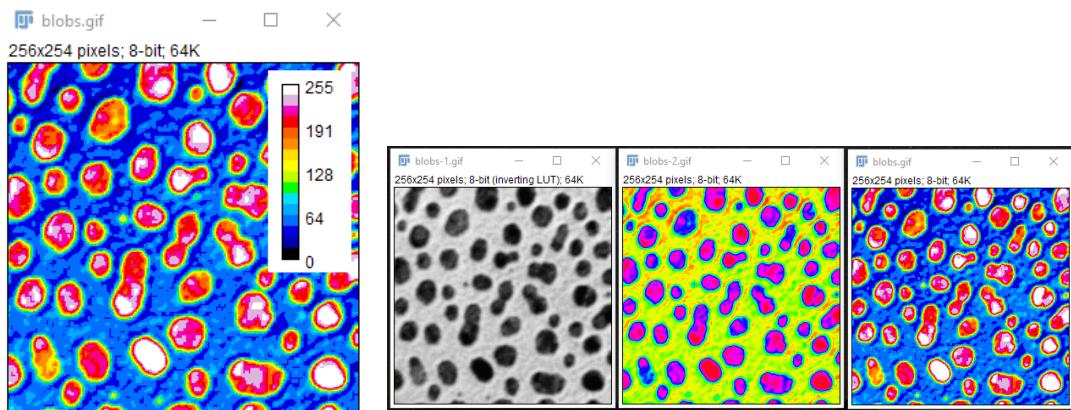
Prefs	127	128	129	130	131	132	133
100	56	48	48	48	56	56	64
101	56	56	56	56	64	64	64
102	64	56	64	64	64	64	64
103	56	48	56	56	56	56	56
104	48	40	40	40	48	48	48
105	40	40	40	40	40	40	48
106	40	32	32	32	32	32	40

Inverting the background

1. Sometimes, the background may be white instead of black, depending on what you are interested in within the image. The color scheme can be inverted by selecting **Edit > Inverse**.

Changing the color scheme (LUT)

1. A lookup table, or LUT, control the colormap of the images. This can easily be changed using the **LUT** button. Try a few and see how the information in the background and foreground changes.
 - a. Using **Image > Colors > Display LUTs** will display different types of LUTs.
 - b. It is also helpful to include a scale/calibration bar when changing the LUTs. This can be added with **Analyze > Tools > Calibration Bar**



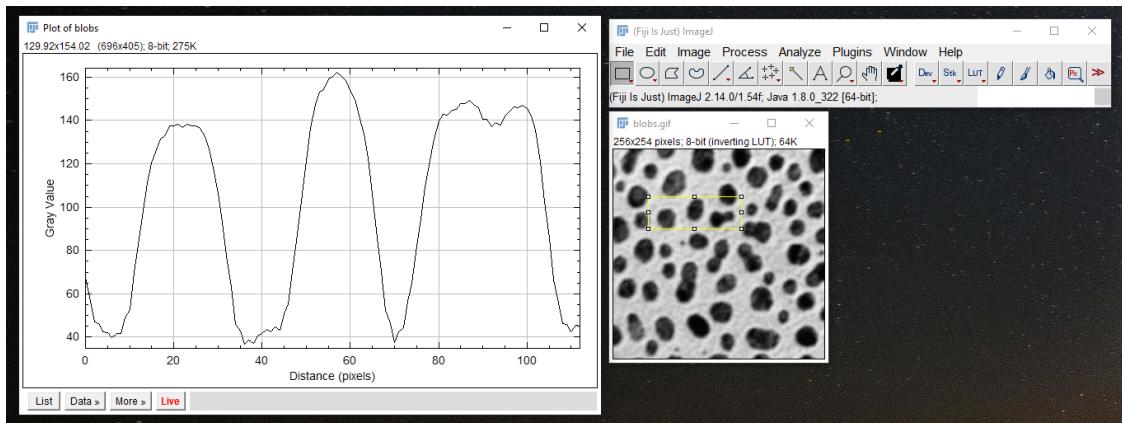
Colorblindness Consideration

To test how different images may appear to those with colorblindness, you can use the **Simulate Color Blindness** plugin if you have a RGB image. Mlp-viridis can be a good option as it is designed to be perceptually uniform (see <https://imagej.net/imaging/visualization>).

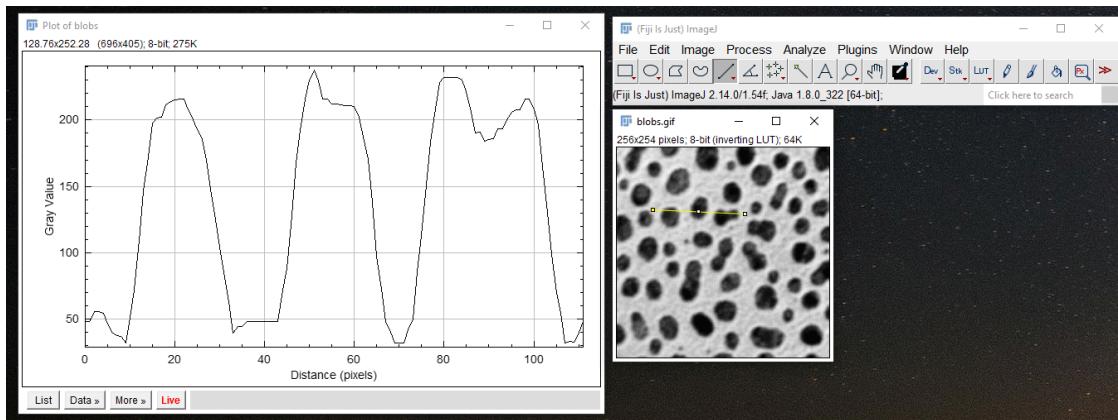
As a separate note, it may also be helpful to test any publication figures (images or otherwise) in grayscale, to see what color information may be lost if the paper is printed in grayscale. In many cases it may be more practical to develop images/figures in grayscale, especially for presentations, and it is still a valid representation of the data.

Plot/Line Plot

1. To see graphically how the pixel values vary across the image we can use the plot tool. This can be very useful for determining how much noise is in the background of an image, or for finding the width of an object, such as a point spread function.
 - a. Draw a line or rectangle across the image and use **Ctrl + K** to display the plot. The **Live** button in the plot window can be used to create an active plot that changes when the selection in the image is changed.
 - b. Here we can clearly see the increase in signal intensity across the three selected blobs. Since this is a rectangular selection, the values are averaged across the height of the rectangle.



- c. In comparison, the plot of a single line shows similar intensity changes when carefully drawn through the three blobs, but the plot is noisier and more sensitive to where the line is placed.



Background Subtraction

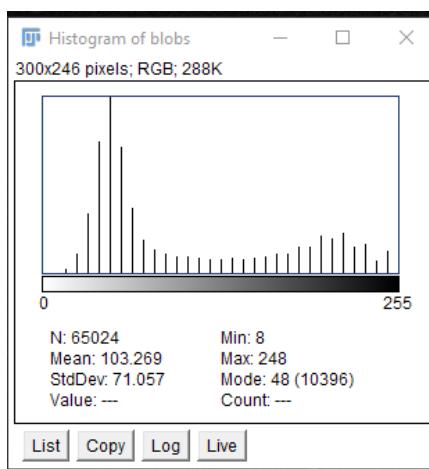
There are multiple methods to perform background subtraction in ImageJ, including **Subtract Background**, **Mean/Median Background Subtraction**, or **Gaussian Blur Subtraction**. However, any background subtraction used to process images should be carefully considered in relation to the original source of noise, how the image information is used, and any further processing or quantification of the image. In some cases the Subtract Background plugin, or other methods, can also introduce artifacts that are not present in the original image.

Some Image.sc forum discussion here:

<https://forum.image.sc/t/consensus-on-subtract-background-built-in-or-other/7061>

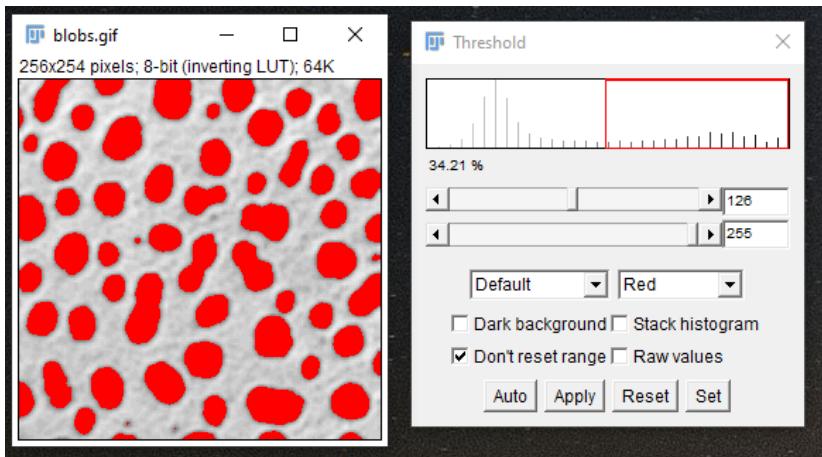
Segmentation/Thresholding

1. If we want to segment out the blobs we are interested in, one possible method is to use an intensity threshold. First, let's look at the histogram of the intensity values using **Ctrl + H**.

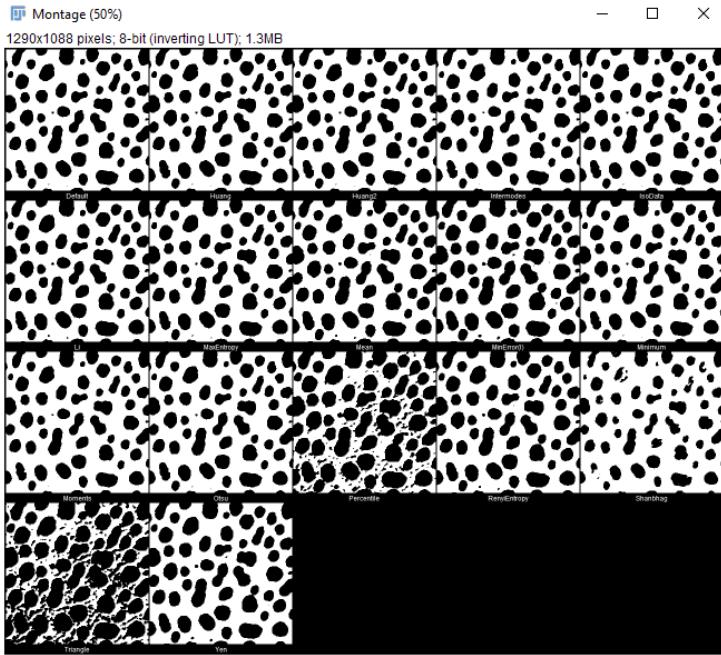


We can see that most of the background pixels have an intensity value around 50, whereas there is another grouping of pixel intensities around 210, which is likely the areas we are interested in because this image has a white background. Notice that the values are stretched outwards, this indicates that the original range was likely very small or somehow adjusted.

2. Use **Image > Adjust > Threshold** to see the automatic threshold applied based on the original histogram. This seems to segment the blobs pretty well. You can change the slider bars to see how the segmentation changes as the threshold moves. When ready, click **apply**, to create the binarized mask. Notice that the blobs now have a value of 255, and the background has been set to 0. This represents a **permanent change** in the image values, so going forward, most quantified analysis involving pixel values must be performed by applying the segmentation as a mask on the original image (see Masks section).



- a. There is also an auto threshold option, **Image > Adjust > Auto Threshold**, that you can use to compare different methods of thresholding. It is easiest to see the comparisons using the “Try All” method.



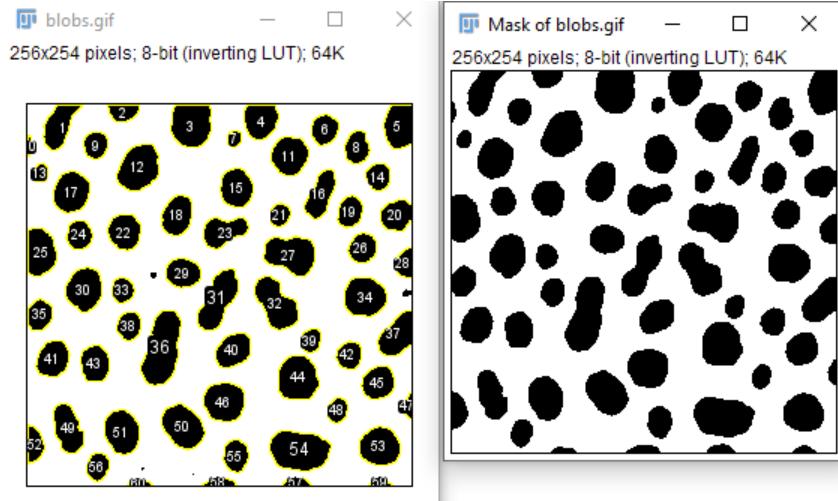
Adding Selections to ROI Manager

1. Press **T** to open the ROI manager.
2. Use **Edit > Selection > Create Selection** to select all of the blobs based on the current threshold.
3. In the ROI Manager window, use **More > Split** to divide the selection into multiple ROIs.
 - a. You can click on the regions to see the specific blob it corresponds to. If needed, the selected regions can be saved in a file and reopened later using **More > Save**

Analyze Particles

1. In this case, the Analyze Particles function can be useful for applying size exclusion, or removing any blobs that are below our desired size threshold. For example, we can see some smaller red circles in the image above that may represent noise rather than a full blob we are interested in. This can be especially useful if we have a biological problem, for example segmenting cells, where we know the cells must be above a certain size.
 - a. Open the ROI manager by pressing **T**.
 - i. If there are previous ROI selections that you would like to remove, click on the first one, hold shift, then click on the last one. This should be a simple way to select all of the ROIs at once. You can then press delete.
 - b. Use **Analyze > Analyze Particles** to open the function window.
 - i. The **Size** can be used to determine what size blobs are included. Here, we can specify “**30-Infinity**” to remove some of the smaller blobs in the mask. For many biological applications, it is best to know the size of the

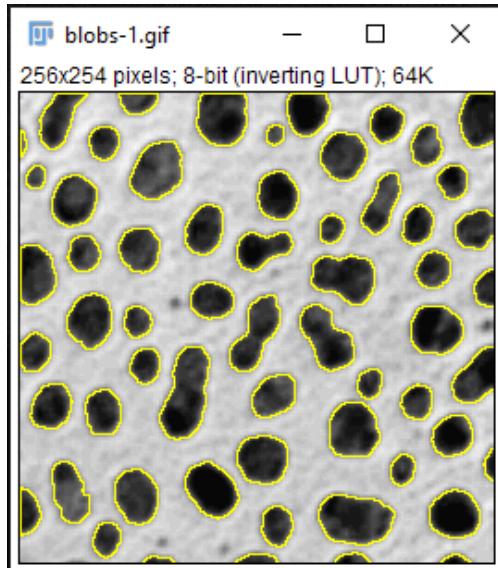
- pixels to know exactly how much area might be excluded, and if it corresponds to the correct size based on the experiment.
- ii. Ensure the **Add to Manager** box is selected to add the ROIs to the ROI manager.
 - iii. For the **Show** dropdown, use **Masks** to display the following:



Notice that some of the blobs that are present in the binarized image, like the one next to number 29, are not included in the mask.

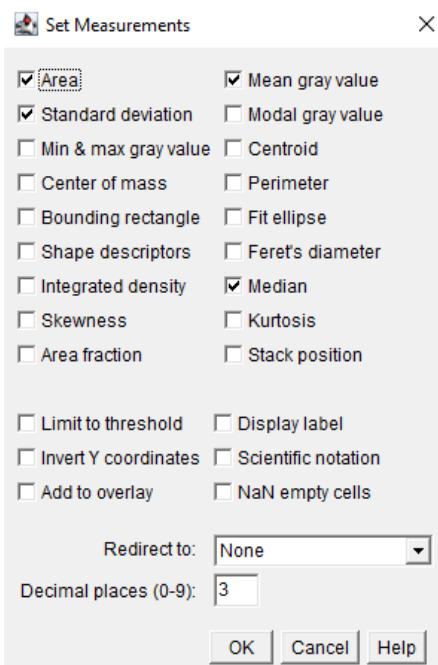
Masks For Measurement

1. Once you have a mask or the desired image, we can use the **Edit > Selection > Create Selection** tool to add the regions to the ROI manager.
 - a. After selection, you can also use **Edit > Selection > Create Mask** to make a mask of the desired areas if the image is not already binarized.
2. To copy the ROI selection onto the original image, click on the mask while the selection is open, then click on the original image and press **Shift + E** or use **Edit > Selection > Restore Selection**



Making Measurements

1. To set the desired measurements to be collected use **Analyze > Set Measurements**. This window will determine what results are displayed or saved from the image.



2. To create the measurements, use **Analyze > Measure** or **Ctrl + M**, while the desired image is active.
 - a. This may produce one measurement for the whole image if an ROI is not active. To measure all ROIs at once, use **Measure** in the **ROI manager**. The

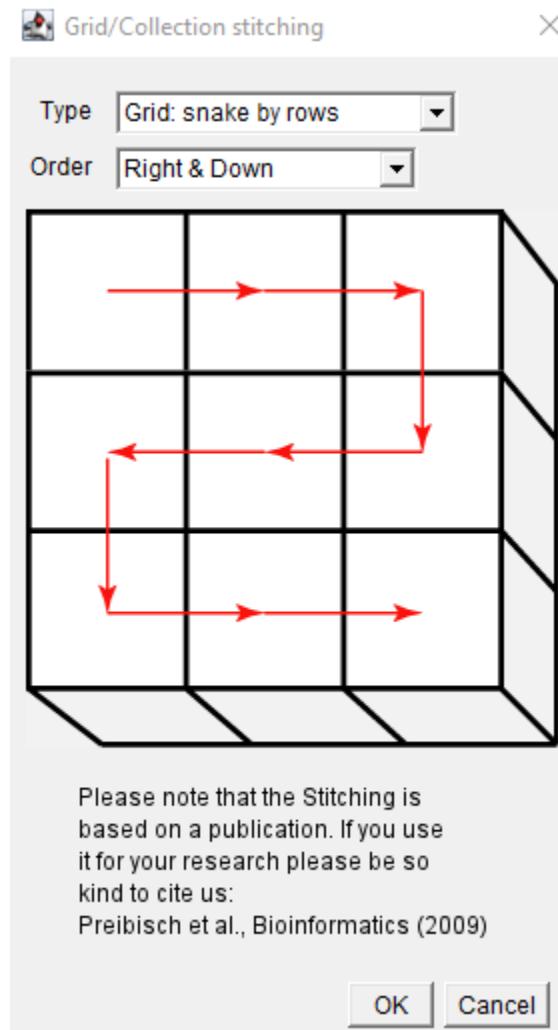
checkboxes for Show All and Labels can be useful to see what regions are being measured.

- b. If you want, use **Edit > Selection > Make Inverse**. Notice this now selects the entire background. You can measure again to get the value for the background.

Stitching Images

There are multiple ways to stitch images using the **Grid/Stitching** plugin, including with metadata or file position.

1. Open the Grid/Stitching plugin. In this example, we will use “Grid: snake by rows” with the order of “Right & Down”



2. In the dialogue box, specify that the following parameters:
 - a. Grid size is **x = 3 and y = 3**

- b. **Tile overlap [%] = 0.** Importantly, this is a known value, as the image we are stitching was part of a larger image. In an experimental context, this may be part of the image acquisition settings during imaging.
- c. The **first file index is 1**
- d. Directory is to the **Leaf_stitch folder**
- e. The file name is **leaf-{i}.tif**
 - i. {i} specifies where in the filename to iterate through values. If this is written as {ii}, there will be an error because the first file is 1 and not 01.
- f. **Uncheck the “Compute overlap” box** as there is a known overlap value.



- 3. Also try re-running the plugin with an overlap value of 10%. Notice how there are errors near the borders of the leaf and on the ruler (especially the faded 9 value), but errors at the center of the leaf may be harder to spot. This is why knowing the expected overlap value is very important, as spotting errors in experimental data may be even more difficult. In many cases, using stitching information from the image metadata is more helpful because the position information comes from the microscope.

Scripting

<https://imagej.net/scripting/>

Scripts can be generated in a variety of ways, including the macro recorder. For functions included in the ImageJ Macro language, please see this link:

<https://imagej.net/ij/developer/macro/functions.html>

Macro Recorder

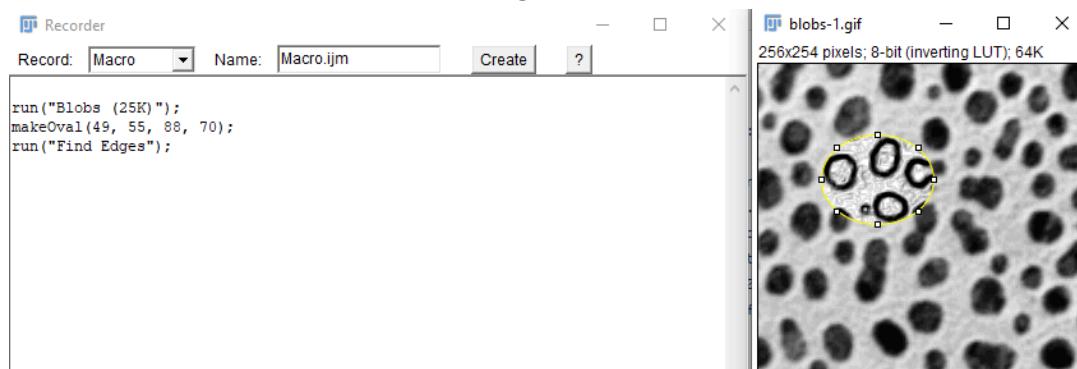
<https://imagej.net/scripting/macro>

1. Use **Plugins > Macros > Record** to start recording steps of simple functions and begin to generate a code. Note that when clicking around in FIJI, you may generate unnecessary or overly specific commands, which means macros generated through the recorder will typically require some level of editing to use on other images.

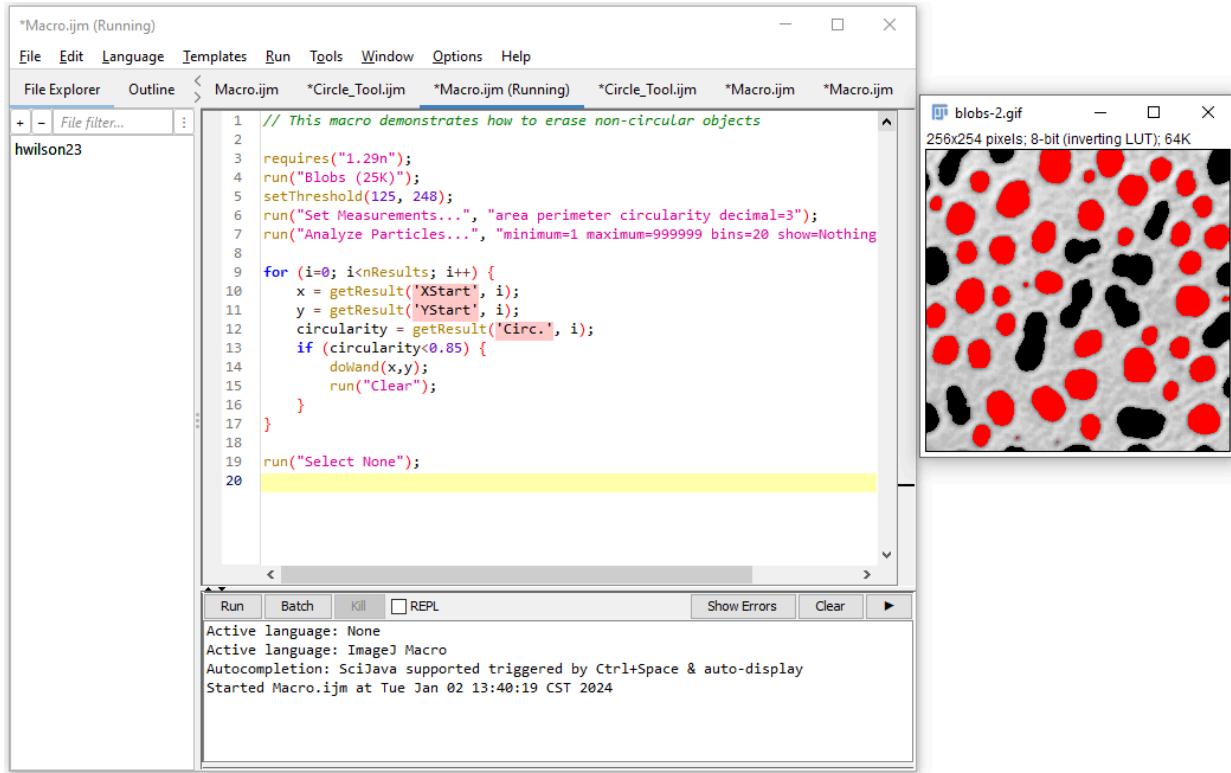


For example:

- a. Open Blobs with **File > Open Samples > Blobs**
- b. Using the tools, draw a circle or rectangle
- c. Run **Process > Find Edges**



Other examples can be found on the ImageJ website, such as the CircularParticles.txt macro shown below: <https://imagej.net/ij/macros/>

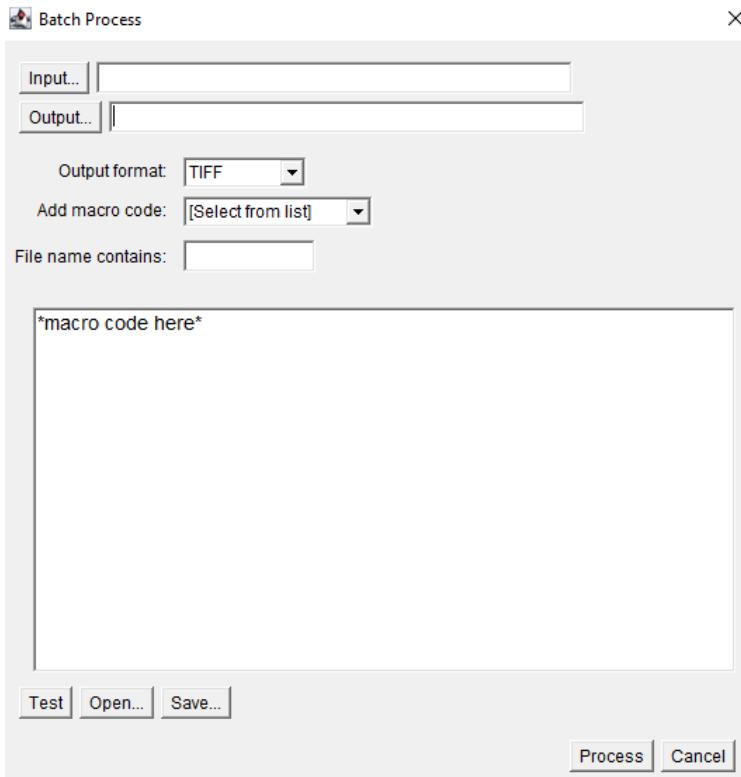


Why does the recorder not show a response for all plugins?

Not every developer has made the plugins compatible with the macro recorder, which may cause various bugs or a lack of response from the recorder. In some cases, a script may be used to generate similar functionality, but other times this may just be a limitation of the plugin. The Image.sc forum would be a good resource for help if you are able to provide enough information about the error and the analysis goals.

Batch Processing

1. To run a macro on a folder of files, batch processing can be used. With minimal code adjustments, this can be done with **Process > Batch > Macro**. This method has some limitations as it will usually overwrite the data, and the loop is not as customizable as a script.
2. Add the macro code to the following window:



3. Alternatively, the batch processing can be directly added to the macro script in the script editor, such as with the following template available on the ImageJ website:
<https://imagej.net/scripting/batch#option-2---script-template>
 - a. Open the script editor: **Plugins > New > Macro**
 - b. The template can be accessed in the script editor by using **Templates > ImageJ 1.x > Batch > Process Folder (ImageJ Macro)**.
 - i. This template contains an existing loops to process each files in the folder using the “processFolder” and “processFile” functions. It requires the user to input the input and output directory as well as the file information.
 1. When you are inserting the macro information, remember to open and save the file using the **open()** and **saveAs()** functions.
 - ii. The benefits of this is that the files can be more specifically iterated if needed with minor adjustments to the code. For example, try changing “i++” to “i=i+2” to work with every other file.
 - iii. Under **Options** in the script editor, there are features for “Enable Autocompletion” and “Show Completions Without Ctrl+ Space” which may help with developing the initial script.

The screenshot shows the ImageJ macro editor interface. The menu bar includes File, Edit, Language, Templates, Run, Tools, Window, Options, and Help. The title bar says "*New_.ijm". The main window displays a Groovy script for deconvolution. The code defines a function `processFolder` that scans a directory for files and processes them if they have a specific suffix. It also includes a `processFile` function. The status bar at the bottom shows "Active language: None" and "Active language: ImageJ Macro".

```

/*
 * Macro template to process multiple images in a folder
 */
#@ File (label = "Input directory", style = "directory") input
#@ File (label = "Output directory", style = "directory") output
#@ String (label = "File suffix", value = ".tif") suffix

// See also Process_Folder.py for a version of this code
// in the Python scripting language.

processFolder(input);

// function to scan folders/subfolders/files to find files with correct suffix
function processFolder(input) {
    list = getFileList(input);
    list = Array.sort(list);
    for (i = 0; i < list.length; i++) {
        if(File.isDirectory(input + File.separator + list[i]))
            processFolder(input + File.separator + list[i]);
        if(endswith(list[i], suffix))
            processFile(input, output, list[i]);
    }
}
function processFile(input, output, file) {
}

```

Deconvolution

Deconvolution is a method of denoising a microscopy image using information about the optical setup. In this case, we estimate a point spread function for the deconvolution using the user input parameters of numerical aperture, resolution, emission wavelength, and refractive index.

An ImageJ/FIJI macro for this has been developed by Ed Evans and can be accessed here: <https://github.com/elevans/fiji-scripts/blob/main/imagej2/deconvolution/decon.groovy>

1. Copy and paste the code into a macro script. **Change the script language to Groovy**, then run for the selected image.

```

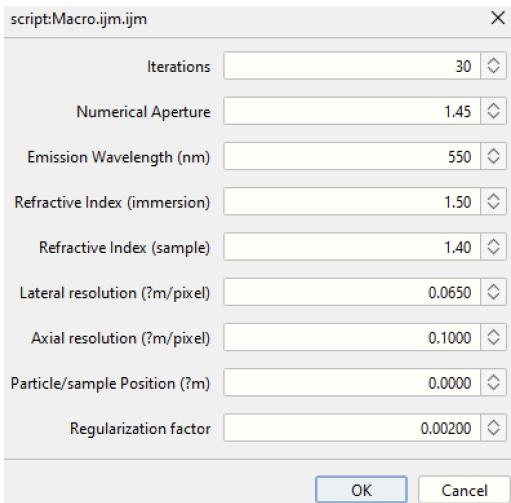
File Edit Language Templates Run Tools Window Options Help *Macro.ijm.ijm
File Explorer Outline < *Macro.ijm.ijm
Owner + - File filter...
1 #@ ImgPlus img
2 #@ OpService ops
3 #@ Integer iterations(label="Iterations", value=30)
4 #@ Float numericalAperture(label="Numerical Aperture", style="format:0.00", min=0.0, max=1.0)
5 #@ Integer wavelength(label="Emission Wavelength (nm)", value=550)
6 #@ Float riImmersion(label="Refractive Index (immersion)", style="format:0.00", min=1.0, max=1.7)
7 #@ Float riSample(label="Refractive Index (sample)", style="format:0.00", min=1.0, max=1.7)
8 #@ Float lateral_res(label="Lateral resolution (?m/pixel)", style="format:0.00", min=0.001, max=1.0)
9 #@ Float axial_res(label="Axial resolution (?m/pixel)", style="format:0.0000", min=0.0001, max=1.0)
10 #@ Float pZ(label="Particle/sample Position (?m)", style="format:0.0000", min=-10.0, max=10.0)
11 #@ Float regularizationFactor(label="Regularization factor", style="format:0.0000", min=0.0001, max=1.0)
12 #@output ImgPlus psf
13 #@output ImgPlus result
14
15 import ij.IJ
16 import net.imglib2.FinalDimensions
17 import net.imglib2.type.numeric.real.FloatType
18
19 // convert integer parameters to float
20 wavelength = wavelength.toFloat()
21

```

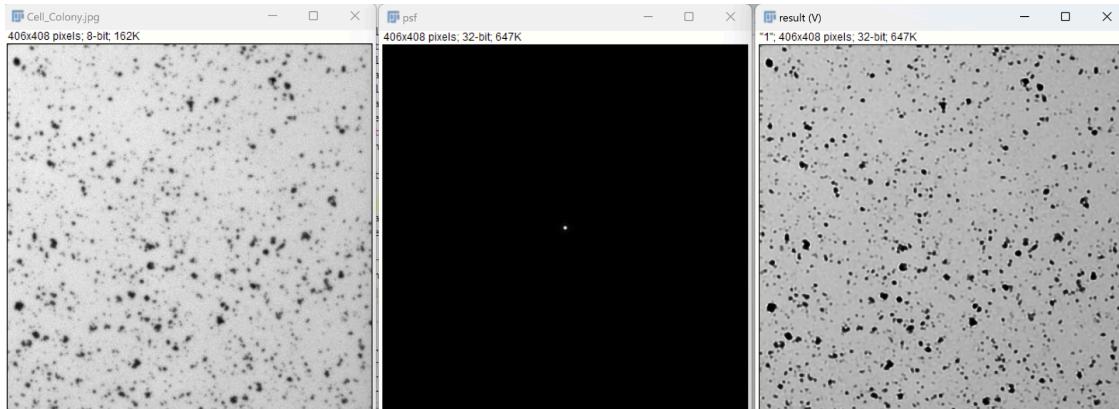
Run Batch Kill REPL Show Errors Clear ►

Active language: None
 Active language: ImageJ Macro
 Autocompletion: SciJava supported triggered by Ctrl+Space & auto-display
 Started Macro.ijm.ijm at Thu Jan 04 16:25:40 CST 2024

2. The parameter box will be displayed. Enter the known parameters.



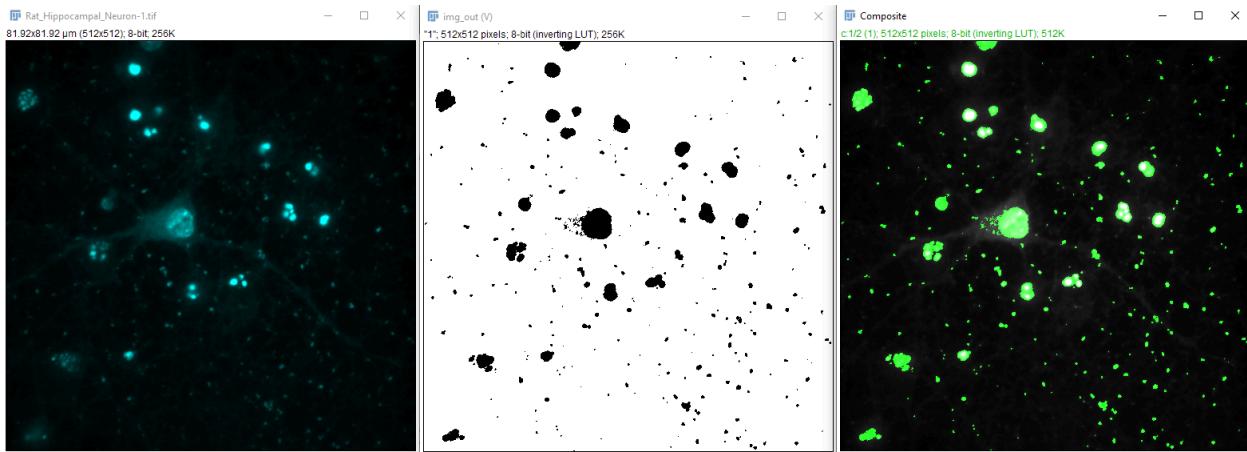
3. The output should display the estimated point spread function (PSF) and the deconvolved image. The following example is produced with the Cell Colony FIJI sample and the default code parameters. Although this is not representative of what happens with the correct parameters from a real experiment, it works as a demo to show what happens with the deconvolution.



Background Subtraction - Gaussian filters

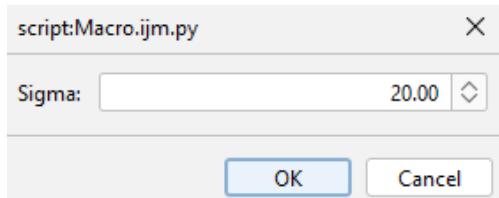
Performing background subtraction with a gaussian filter can be an effective method of resolving objects from a noisy background. More info here:

<https://bioimagebook.github.io/chapters/2-processing/4-filters/filters.html#gaussian-filters>

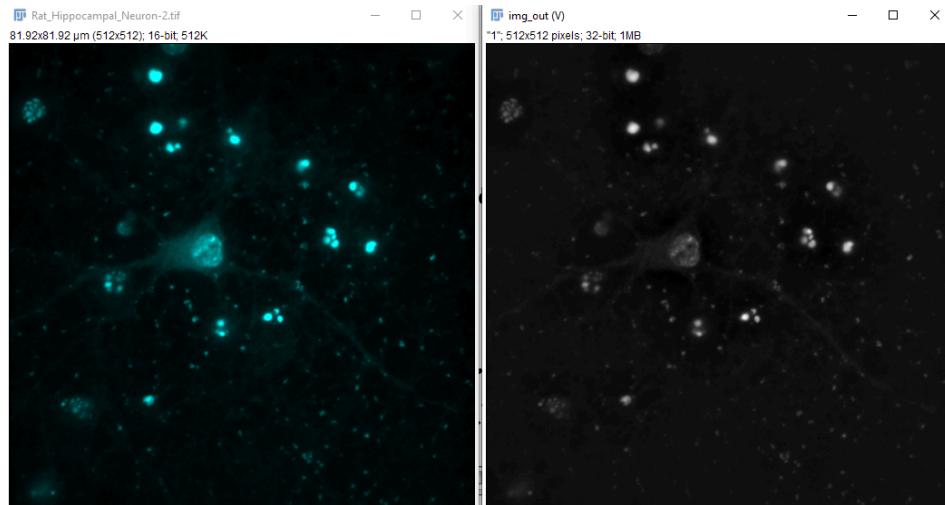


1. Use **File > Open Samples > Neuron (5 channel)** to open the starting image. We only want one channel to work with, so right click and **duplicate channel 4**.
 - a. Try a variety of thresholds on this image, including the triangle method. Here we will try to improve this threshold by separating out the smaller point-like structures.
2. Open the gaussian subtraction script found here, developed by Ed Evans:
 https://github.com/elevans/fiji-scripts/blob/main/imagej2/filters/2D_gaussian_subtraction.py
 - a. This script is written in python, so be sure to change the macro language to python before running, otherwise an error will be produced.
3. With the single channel image highlighted, run the script. You should be prompted to input a sigma value for the gaussian filter. Larger values will blur out larger and larger

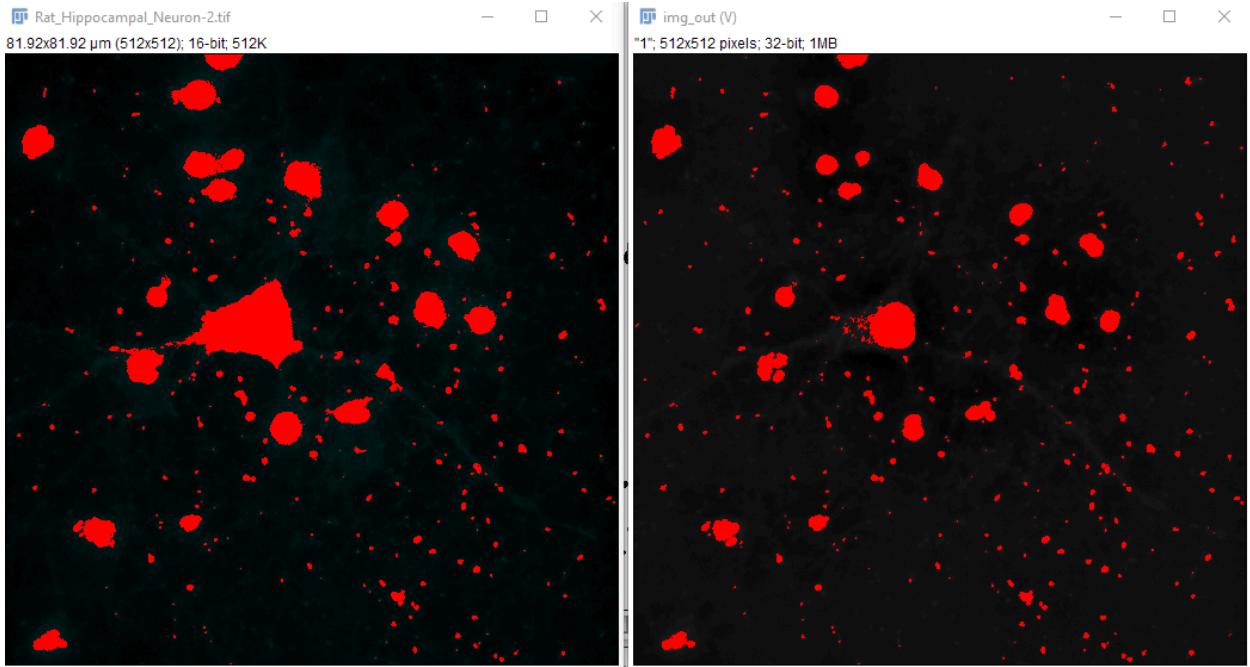
objects. For this example, use 20 and then use 1. Feel free to try a variety of values to see how the results change.



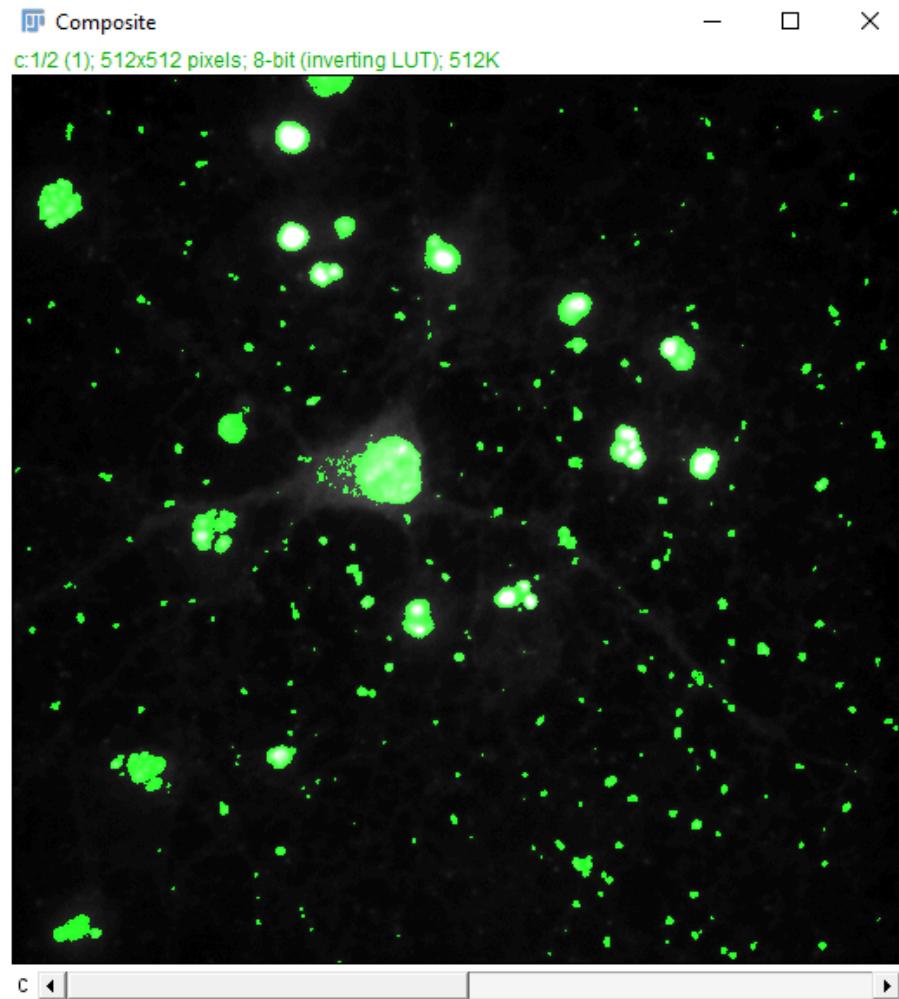
4. After running the script, we now see that some structures are highlighted a bit differently in img_out (V) (notice how the cell body is harder to see):



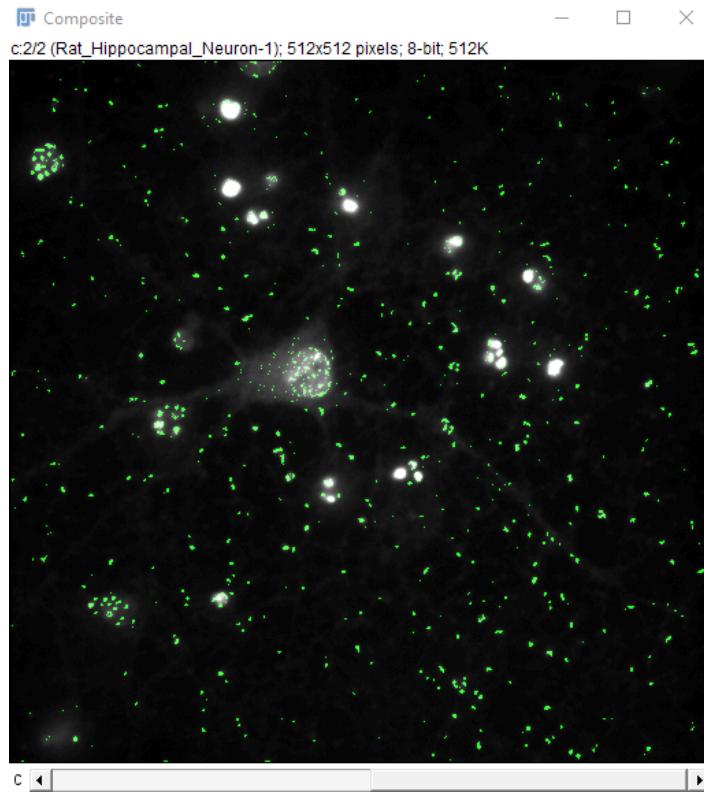
5. Try thresholding the output image again, including the triangle method as shown below. Below are the differences between the original (left) and the output image (right) using the automated triangle threshold. Notice the changes in the amount of cell body and axon signal, as well as a few smaller dots that were not originally picked up in the upper region.



6. If we decide this mask is sufficient for our data, we can then apply the mask to the output image and merge the display.
 - a. With the output image still selected, choose **Apply** on the threshold window, while making sure the dark background box is selected. It then will ask you to convert the output to an 8-bit mask.
 - i. The areas of interest should now have values of 255 while the background has a value of 0. If this is inverted, it is likely that the dark background box in the threshold window was not selected, you can simply use **Edit > Invert** to switch the values.
 - b. To overlay the mask on the original image, we can make a composite. This requires that the original image is also converted to 8-bit for display. Use **Image > Type > 8-bit**
 - c. To make the composite, use **Image > Color > Merge Channels**. Here, I will set the original channel duplicate to C4 (gray) and the mask to C2 (green). This should produce the following composite image, that can be used for display:



If we use a sigma value of 1 and repeat the process, we can see there are further differences in the cell body and other structures, although this seems to also pick up more noise:



Trackmate - Example from documentation

Ershov, D., Phan, M.-S., Pylvänäinen, J. W., Rigaud, S. U., Le Blanc, L., Charles-Orszag, A., ... Tinevez, J.-Y. (2022). TrackMate 7: integrating state-of-the-art segmentation algorithms into tracking pipelines. *Nature Methods*, 19(7), 829–832. [doi:10.1038/s41592-022-01507-1](https://doi.org/10.1038/s41592-022-01507-1)

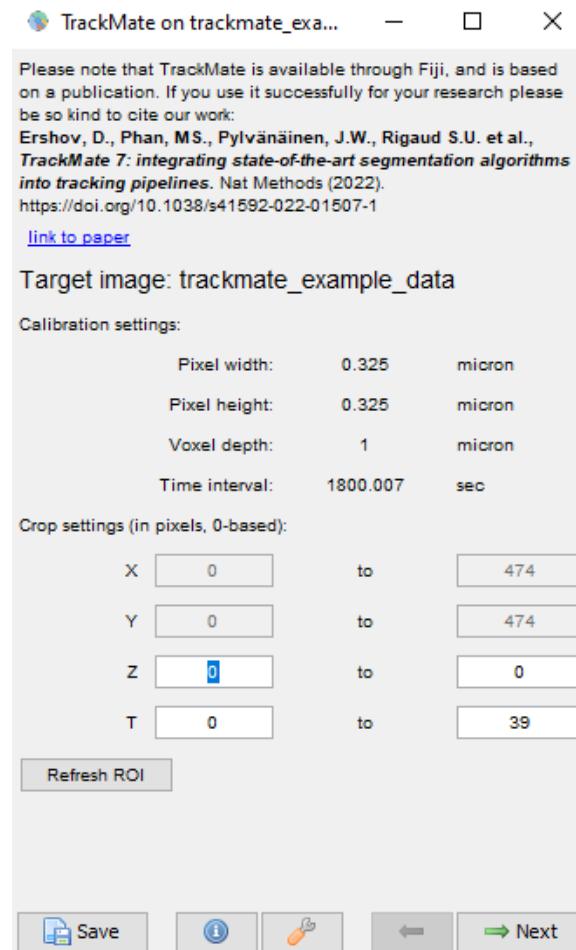
Trackmate documentation and tutorials: <https://imagej.net/plugins/trackmate/>

Trackmate manual: <https://imagej.net/media/plugins/trackmate/trackmate-manual.pdf>

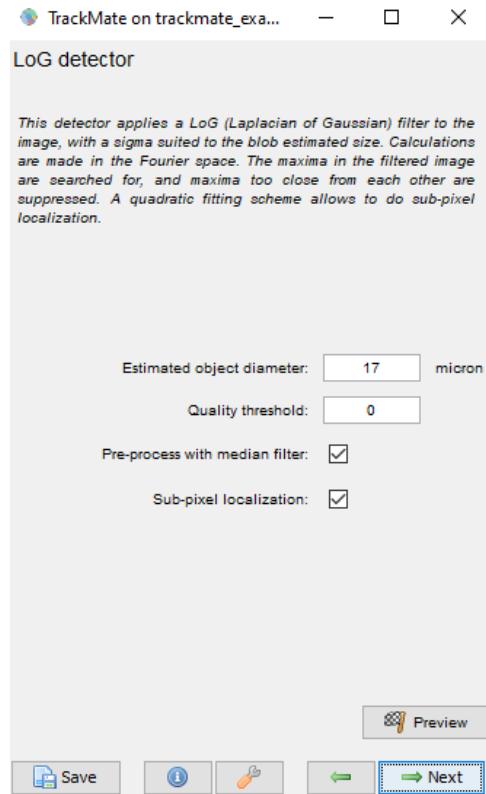
The following demo can be accessed here:

<https://napari.imagej.net/en/latest/examples/trackmate.html>

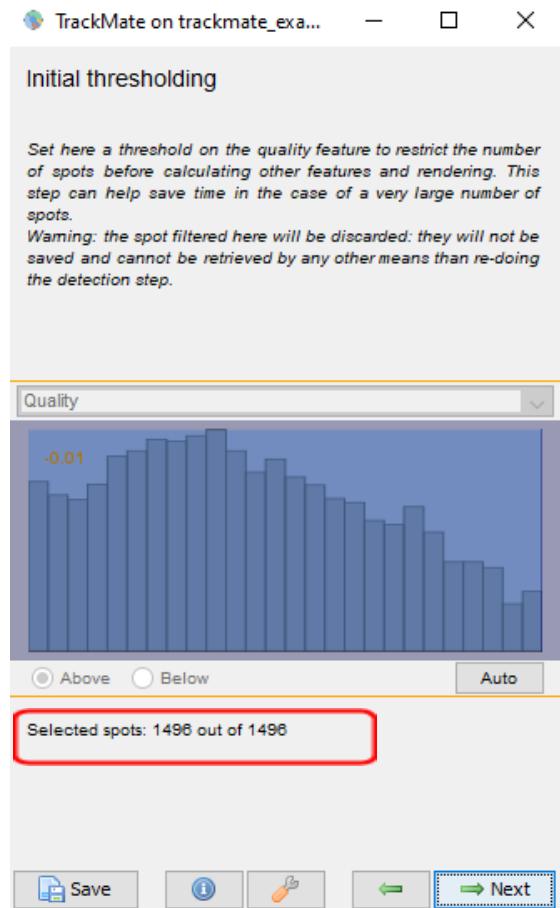
1. Open the “trackmate_example_data.tif”
2. Run the Trackmate plugin (**Plugins > Tracking > Trackmate**). Select “Next” as we don’t need to make any changes. The settings in this window allow you to crop the original data as needed.



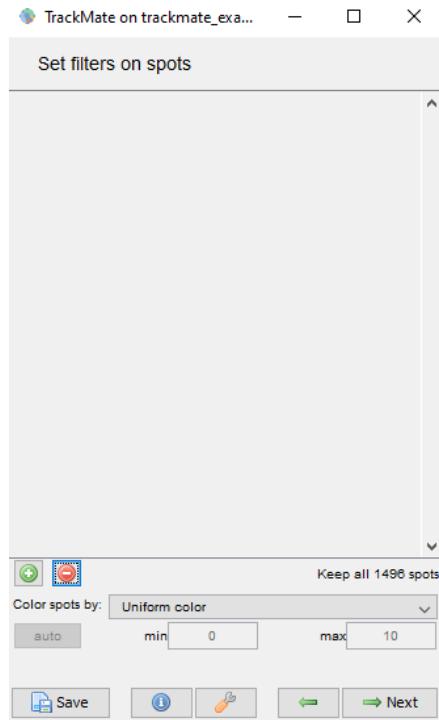
3. Use the Laplacian of Gaussian (LoG) detector. The plugin provides various algorithms to help process the image, also including the difference of gaussian and hessian detectors.
4. Enter 17 as the estimated object diameter and 0 as a quality threshold, "Next."



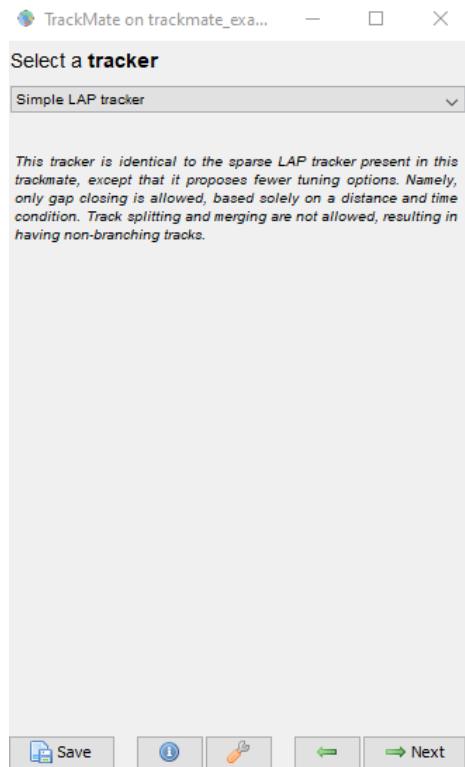
5. The next page provides a summary. Select “Next”
6. For Initial Thresholding, verify you see 1496 spots selected, and select “Next.”



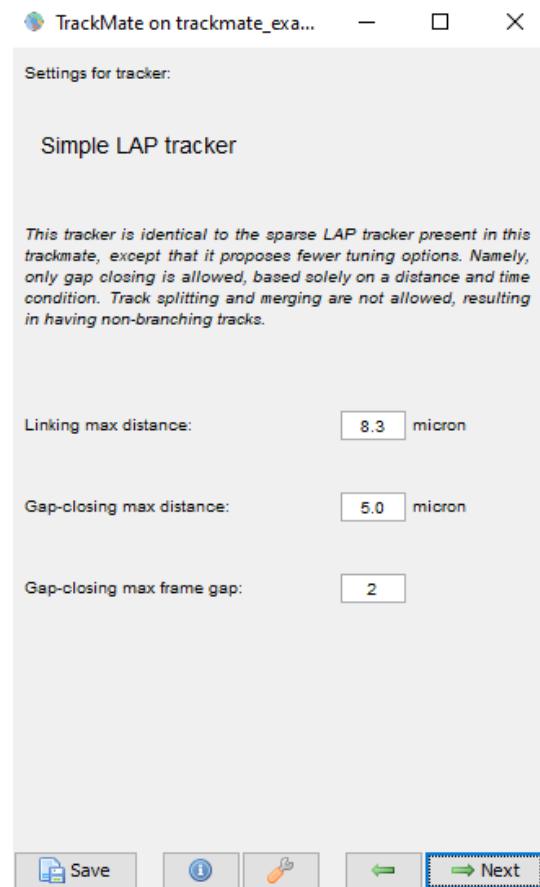
7. Select “Next” as we do not need a filter. If a filter was required, we could use the “Plus” button to add a filter and adjust the selection metric.



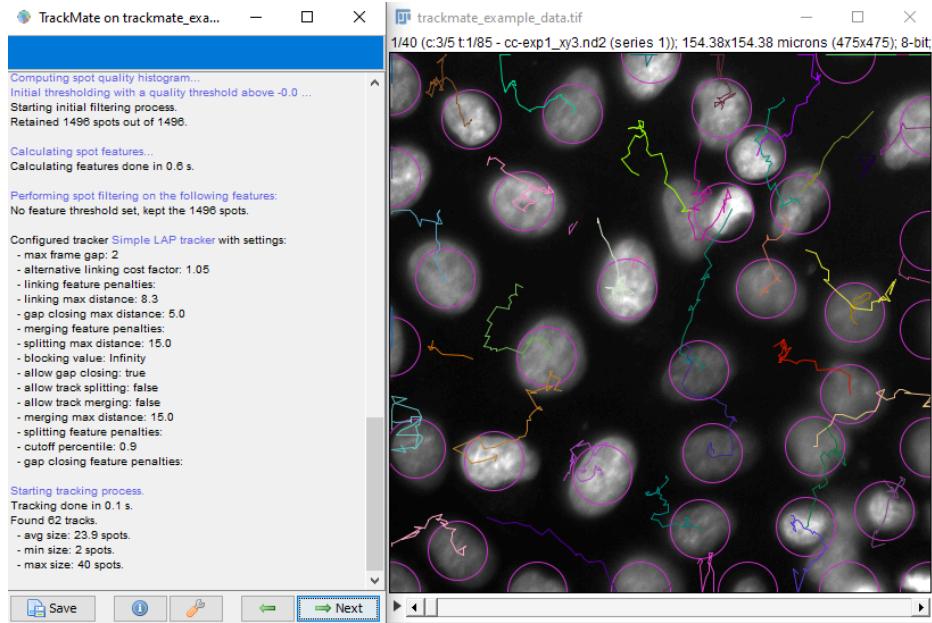
8. Use the “Simple LAP tracker”, then select “Next.” Trackmate also provides a variety of other methods.



9. Then we set the parameters for the tracker. The linking max distance will be 8.3 microns, the gap-closing max distance is 5 microns, and the gap-closing max frame gap is 2 micron. Select “Next.”



10. A summary page is displayed. You can scroll through the image and see the proposed tracks for each cell. Select “Next.”



11. Select “Next” three more times as we will not filter any of the tracks and do not need to edit the display options.
12. Select “Execute” for the final image, which can be run over all of the frames.

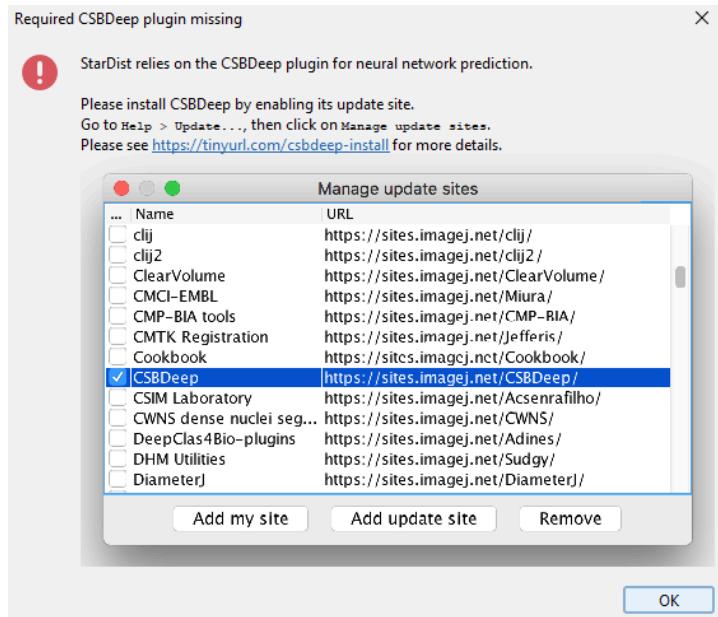
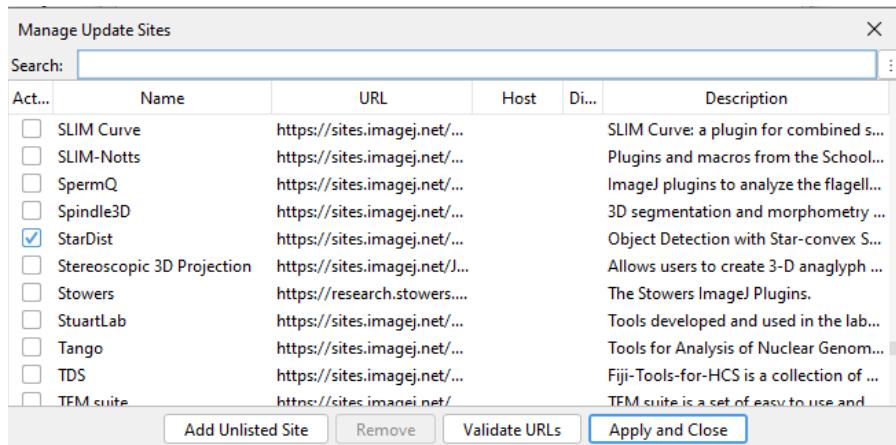
StarDist - Basic Example

Source: <https://github.com/stardist/stardist>

Additional info: (FAQs) <https://stardist.net/index.html>

(ImageJ Docs) <https://imagej.net/plugins/stardist>

1. StarDist is a deep learning tool to segment star-convex shapes (usually blobs, such as cells or nuclei). It can be installed as a FIJI plugin. Add **StarDist** through **Update > Manage Update Sites**. Also add the **CSB Deep** plugin (<https://imagej.net/plugins/csbdeep>), or you will receive the error message below.

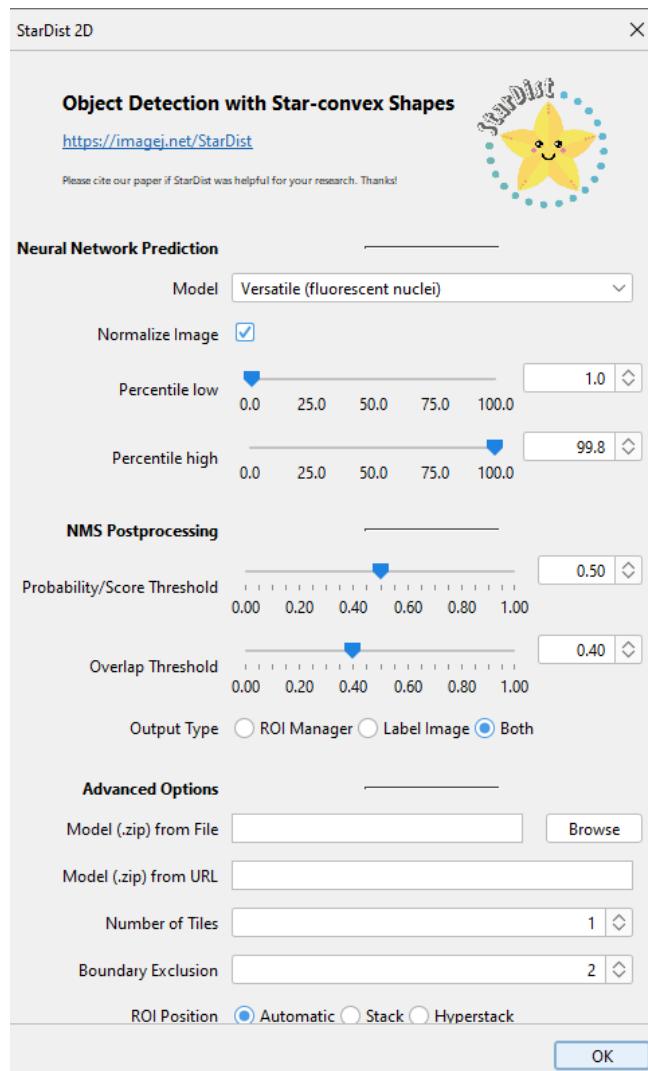


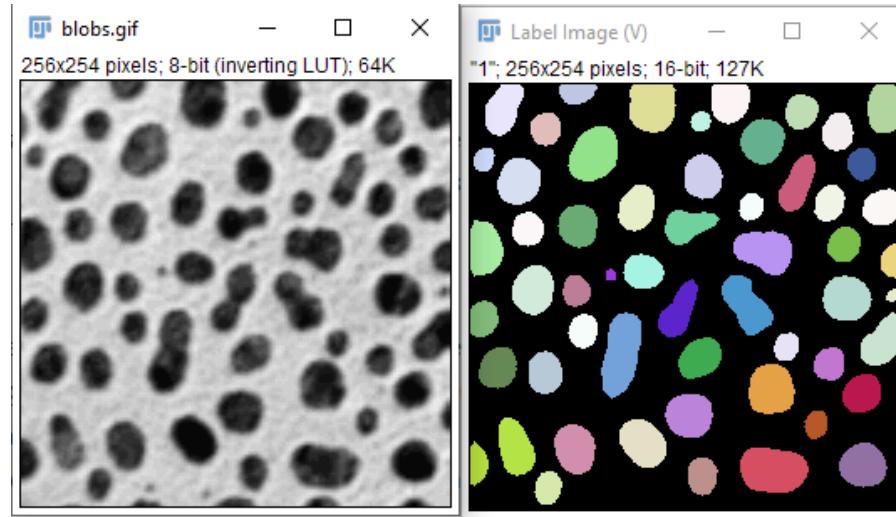
2. Open the blobs image as a simple test case: **File > Open Samples > Blobs**.
3. Run **Plugins > StarDist > StarDist2D**. If we run with the default parameters, shown below, with the "Versatile (fluorescent nuclei)" model we get the following result.

- a. The other options for the Neural Network Prediction allow us to adjust the percentile for image normalization
- b. NMS (non-maximum suppression) Postprocessing parameters operate as follows¹: *Probability/Score Threshold* - higher values lead to fewer segmented objects, but will likely avoid false positives.
Overlap Threshold - higher values allow segmented objects to overlap substantially.

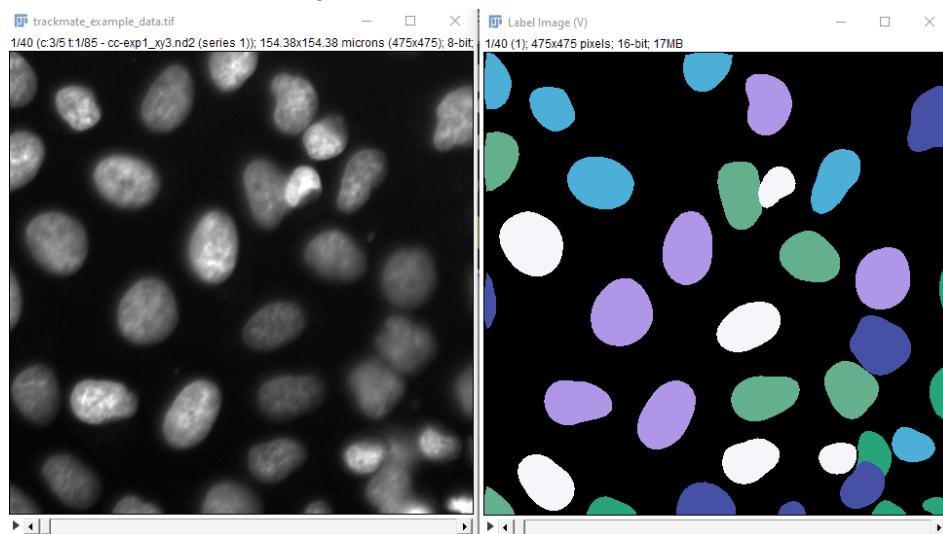
¹From: <https://imagej.net/plugins/stardist>

- c. Advanced options can be used to apply a user-trained model.





4. We can also test this using the **trackmate_example_data.tif** for a timeseries example.
Again run with default settings:



Other Resources

Introduction to Bioimage Analysis (Pete Bankhead):

<https://bioimagebook.github.io/README.html>

ImageJ User Guide (Ferreira and Rasband):

<https://imagej.net/ij/docs/guide/146.html>

Introduction and Segmentation in FIJI:

https://www.youtube.com/watch?app=desktop&v=CZExS_mkGsQ

ImageJ Tutorials:

<https://imagej.net/imaging/>

COBA: Center for Open Bioimage Analysis (YouTube):

<https://www.youtube.com/@cobacenterforopenbioimagea1864/featured>

Scientific Figure Making with Fiji and Inkscape (Jan Brocher):

<https://www.youtube.com/watch?v=F6lI37NOgXc>

ImageJ Documentation: Built-in Macro Functions:

<https://wsr.imagej.net/developer/macro/functions.html>

ImageJ Macro Cheatsheet (Robert Haase, Github haesleinhuepf):

https://github.com/BiAPoL/imagej-macro-cheat-sheet/blob/master/ImageJ_macro_cheatsheet.pdf

ImageJ Documentation: ImageJ Ops:

<https://imagej.net/libs/imagej-ops/>

Trackmate Information:

<https://imagej.net/plugins/trackmate/>

StarDist Information:

<https://github.com/stardist/stardist>

I2K Conference YouTube:

<https://www.youtube.com/@I2KConference>

Colocalization:

- [NEUBIAS “Deconstructing Colocalization Workflows](#) (video)
- [A practical guide to evaluating colocalization in biological microscopy](#) (publication)