Bioimage Analysis with Fiji / ImageJ

Wei-Chen CHU (朱韋臣)

Project Scientist @Imaging Core Facility
Institute of Cellular and Organismic Biology (ICOB),
Academia Sinica

PowerPoint Slide:

Outline

Part I: Basic of Basic

- Terminology
- FIJI / ImageJ introduction
- Image format and visualization
- Region of Interest (ROI), Measurement

Part II: Typical bioimage analysis workflow

- Segmentation
- Selection, ROI List, Mask, Label
- ImageJ Macro
- CLIJ (GPU accelerated image processing)
- Interactive bioimage analysis and automatic code generation
- Future of bioimage analysis (AI)

Part of this document is adapted from the following material under CC BY 4.0

Dr. Robert Haase
Clusters Excellence “Physic of Life” (PoL),
TU Dresden, Germany

https://haesleinhuepf.github.io/

Dr. Peter Bankhead
University of Edinburgh, UK

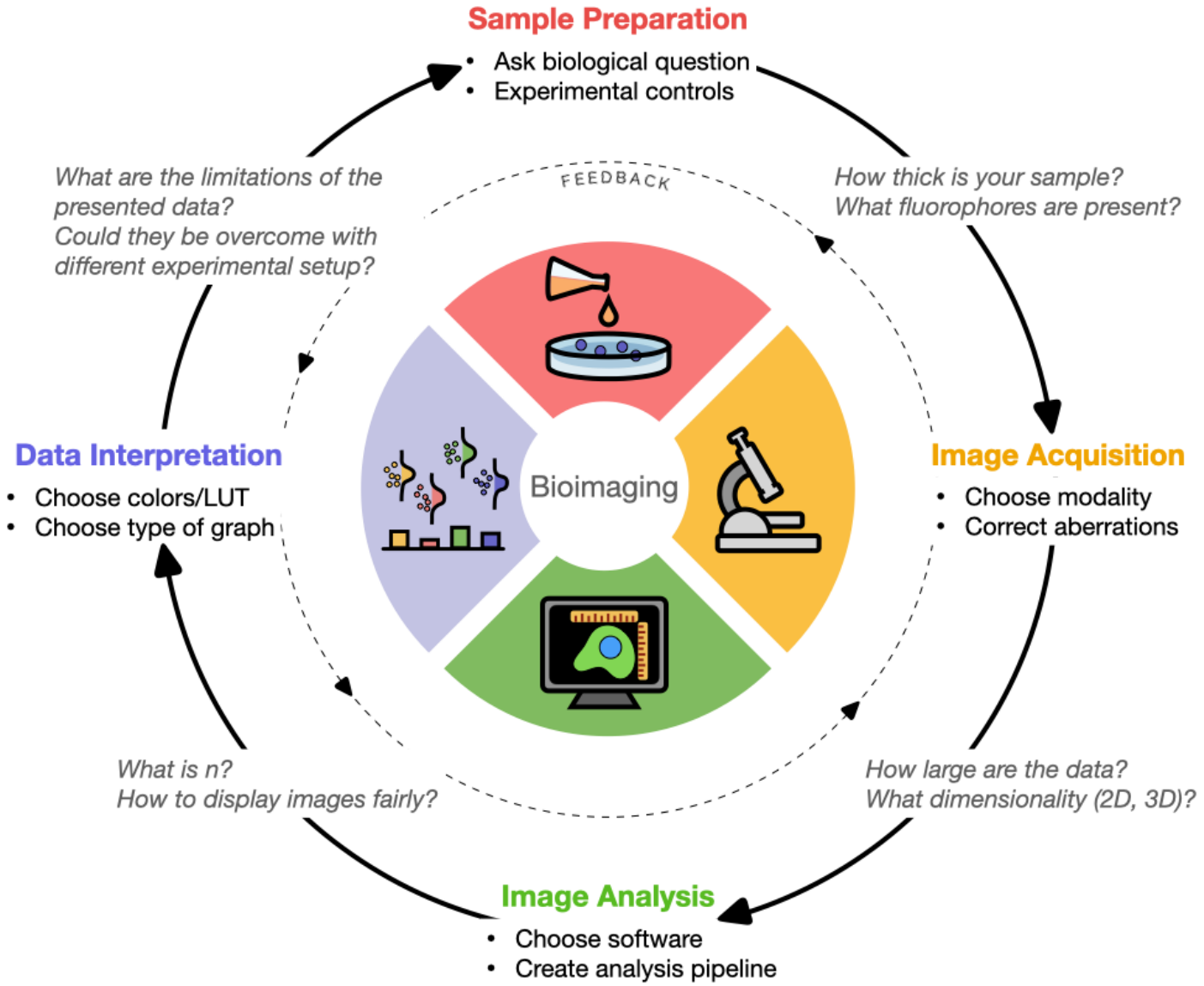
Introduction to Bioimage Analysis (2022)
https://bioimagebook.github.io/

neubias
network of european bioimage analysts

https://neubias.github.io/training-resources/index.html

https://www.youtube.com/@NEUBIAS

Quantitative bioimaging



https://www.bioimagingguide.org/
https://doi.org/10.1371/journal.pbio.3002167

Coverslip thickness is important!

- **Most objectives are designed to use #1.5 coverslips!**
- **Better to use #1.5H for high resolution image**

Most

Exception

Dipping objective

Objective with correction ring

Coverslip No.	Ideal thickness	Range
#0	0.10 mm	0.080 - 0.130 mm
#1	0.15 mm	0.130 - 0.170 mm
#1.5	**0.17 mm**	**0.160 - 0.190 mm**
#1.5H	**0.17 mm**	**0.165 - 0.175 mm**
#2.0	0.22 mm	0.190 – 0.250 mm

Performance Reduction

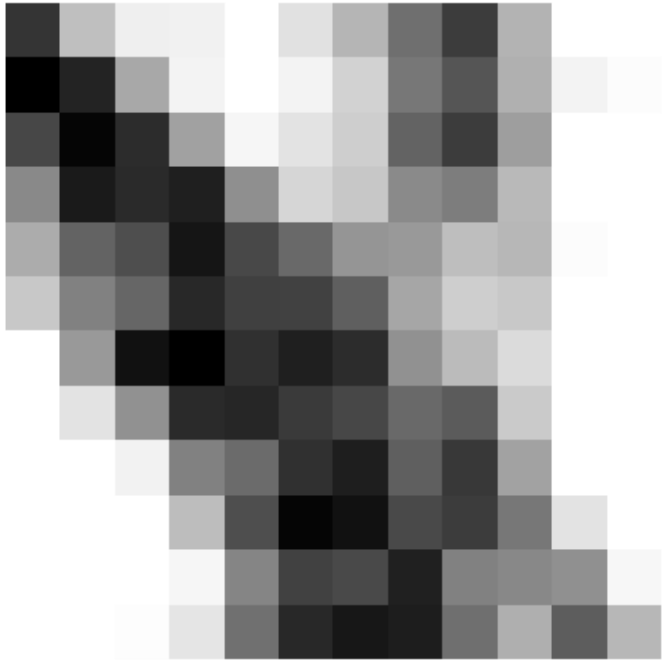
Objective NA.	0.01 mm Deviation	0.02 mm Deviation
0.30	-	-
0.45	-	-
0.70	-2%	-8%
0.85	-19%	-57%
0.95	-55%	-71%

https://www.microscopyu.com/microscopy-basics/coverslip-correction

Image are composed of pixels (picture element)

(A) Original image

(B) Enlarged view from (A)



(C) Pixel values from (B)

53	191	239	241	255	225	181	111	61	180	255	255
	35	168	244	255	243	210	119	85	176	244	252
71		45	161	246	227	206	99	60	158	255	255
137	26	42	31	143	214	199	138	125	185	255	255
172	99	78	21	72	106	149	153	190	183	252	255
200	129	102	41	64	65	95	166	206	200	255	255
255	153	17		49	31	44	145	187	219	255	255
255	227	145	42	38	58	71	106	91	202	255	255
255	255	242	129	107	48	30	95	57	162	255	255
255	255	255	189	78		17	74	60	119	228	255
255	255	255	246	133	65	73	32	129	136	144	247
255	255	253	229	112	40	23	29	111	175	93	183

0

255

Bit-depth

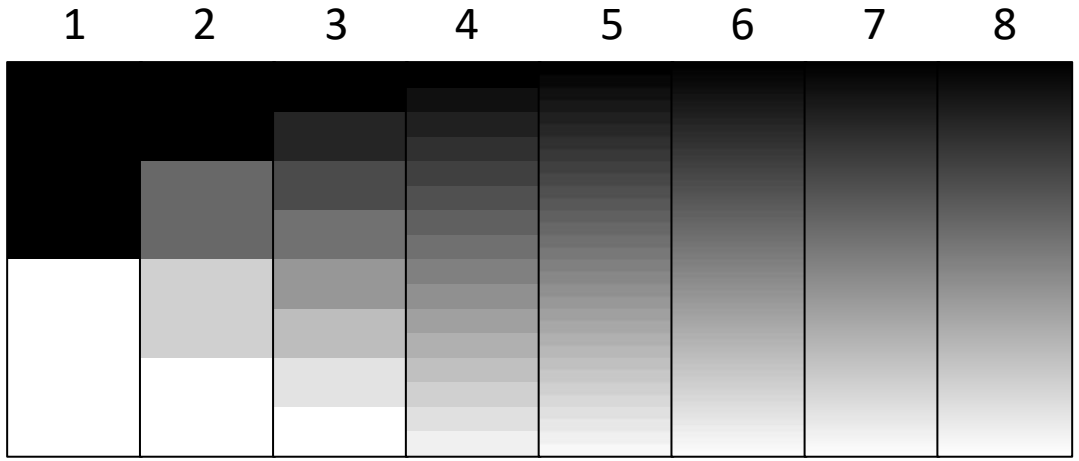
- A bits is the smallest memory unit in computers.
- The bit-depth *n* enumerates how many different intensity values are present in an image:
  - 2^n grey values
- In microscopy, images are usually stored as 8, 12 or 16-bit images.

Bit = “0” or “1”
1 Bytes = 8 bits -> basic store unit

8 bits (1 byte) = 0 ~ 255
16 bits (2 bytes) = 0 ~ 65535

12 bits (2 bytes!) = 0 ~ 4095
14 bits (2 bytes!) = 0 ~ 16383

32 bits (4 bytes, floating-point)



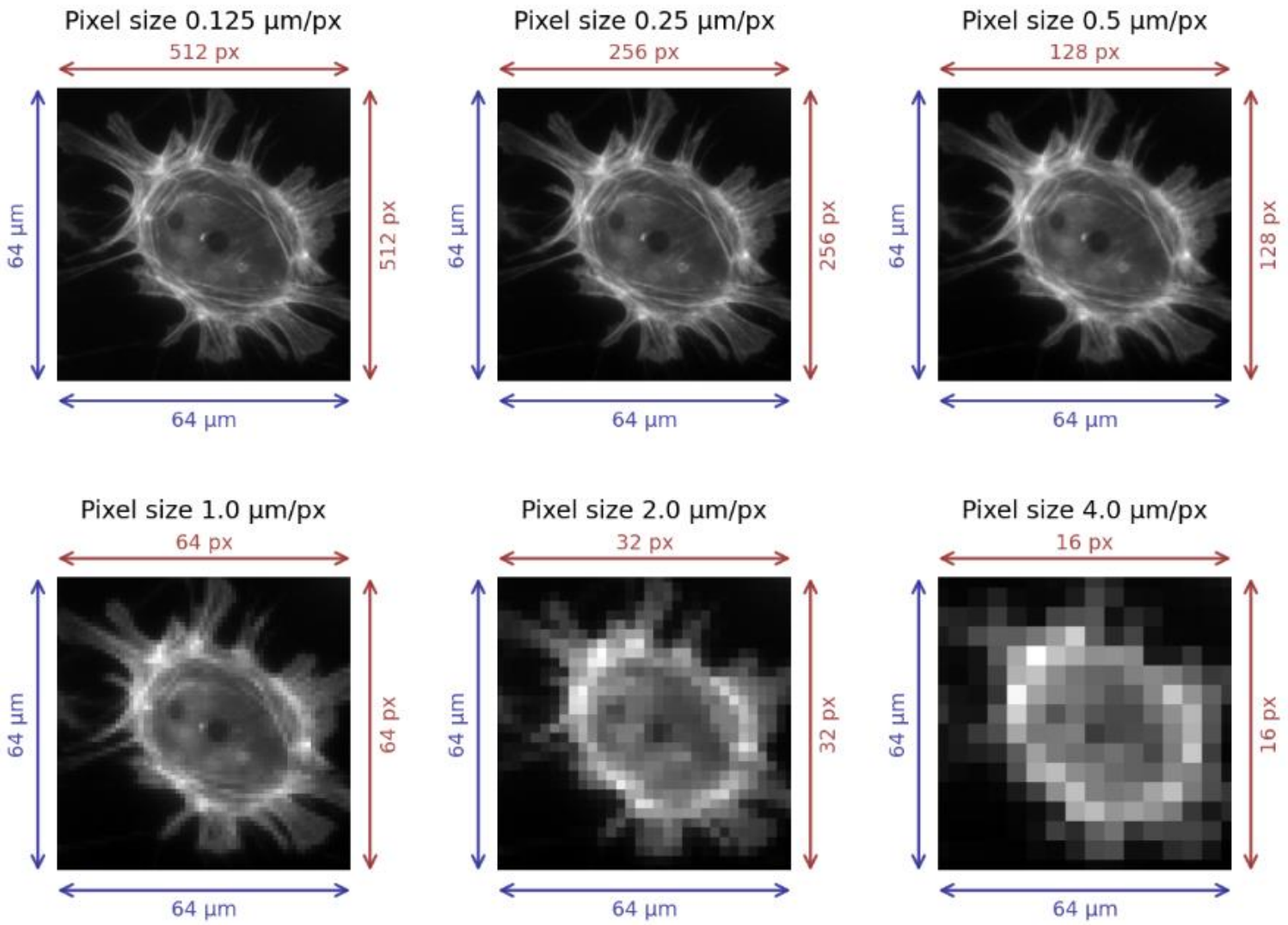
A *'Lookup table' (LUT)* determines which color will be used to represent each value

4	6	58	144	183	53	7	3
2	34	208	239	248	212	74	5
83	138	232	242	235	249	171	3
68	223	219	174	202	229	87	4
5	152	146	47	150	223	89	4
0	99	213	121	136	110	180	4
1	115	209	252	199	216	255	38
3	161	148	231	213	213	201	51

Pixel values

Lookup table (LUT)

Pixel Size



Quiz:

Pixel size: 4 µm

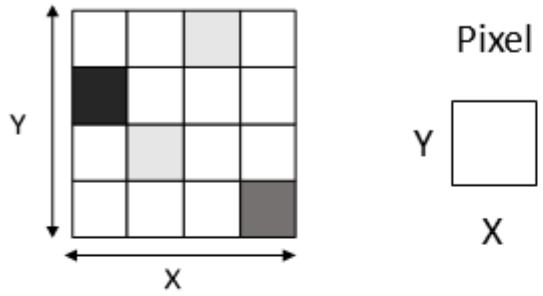
10 px

5 px

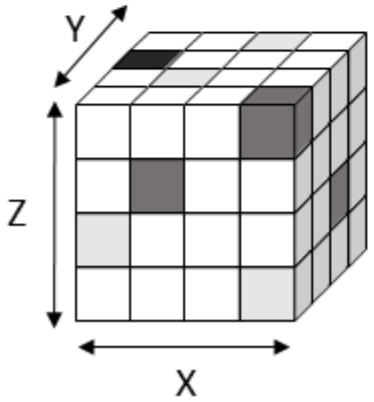
What is the area of the structure in µm^2 ?

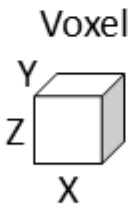
N-dimensional images

2D image

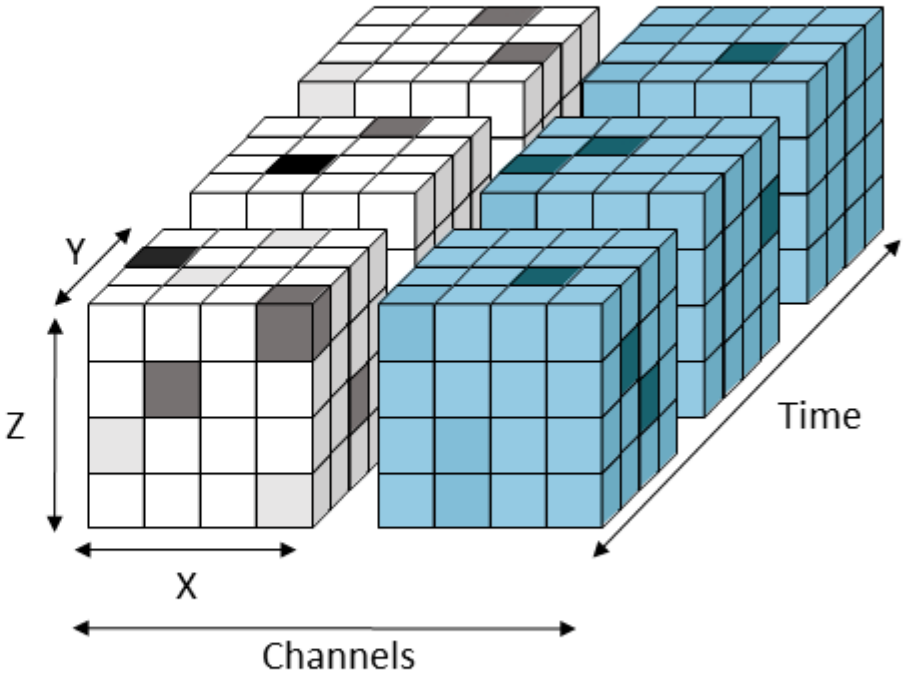


3D image



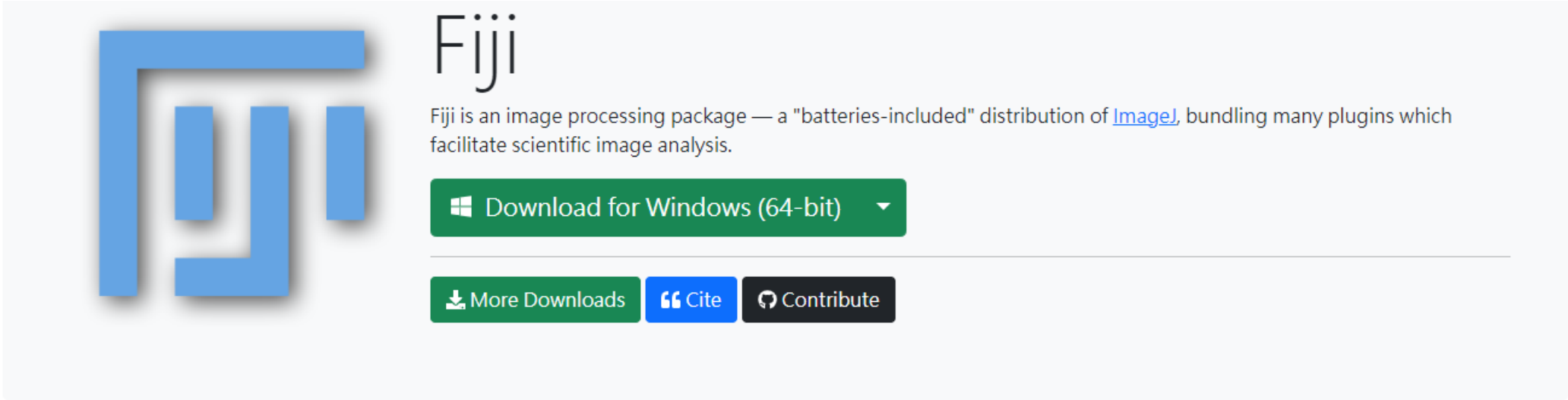


5D image



- Most fluorescence image:
  (n color -> n Channels)
- Data from color camera:
  RGB format
  (n color -> R/G/B Value, 24 bits)

(Fiji is just) imageJ



https://fiji.sc/

ImageJ, ImageJ2 and FIJI

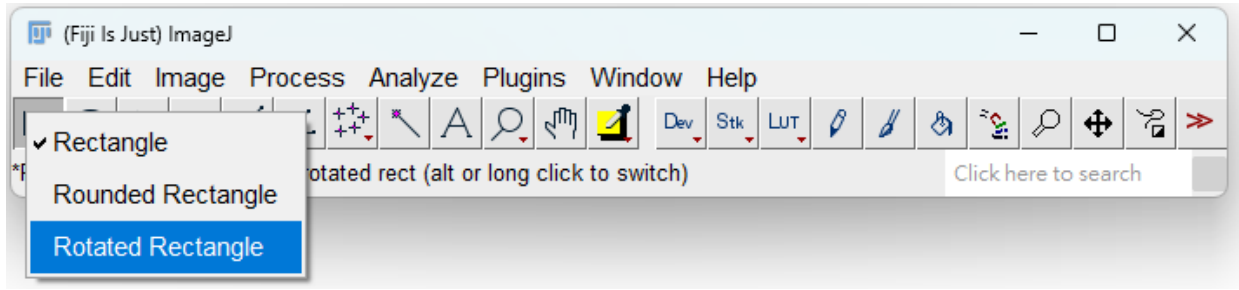
ImageJ Citation:
Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012).
NIH Image to ImageJ: 25 years of image analysis.
Nature Methods, *9*(7), 671–675. doi:10.1038/nmeth.2089

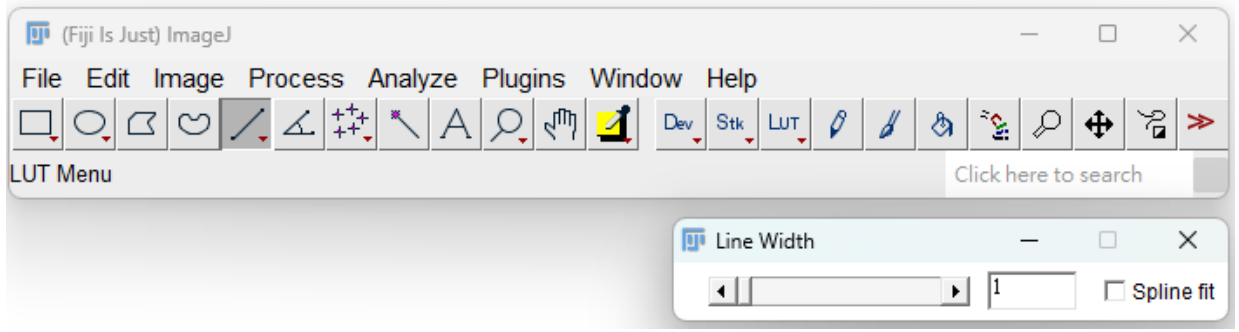
ImageJ2 Citation:
Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T., & Eliceiri, K. W. (2017).
ImageJ2: ImageJ for the next generation of scientific image data.
BMC Bioinformatics, *18*(1). doi:10.1186/s12859-017-1934-z

FIJI Citation:
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

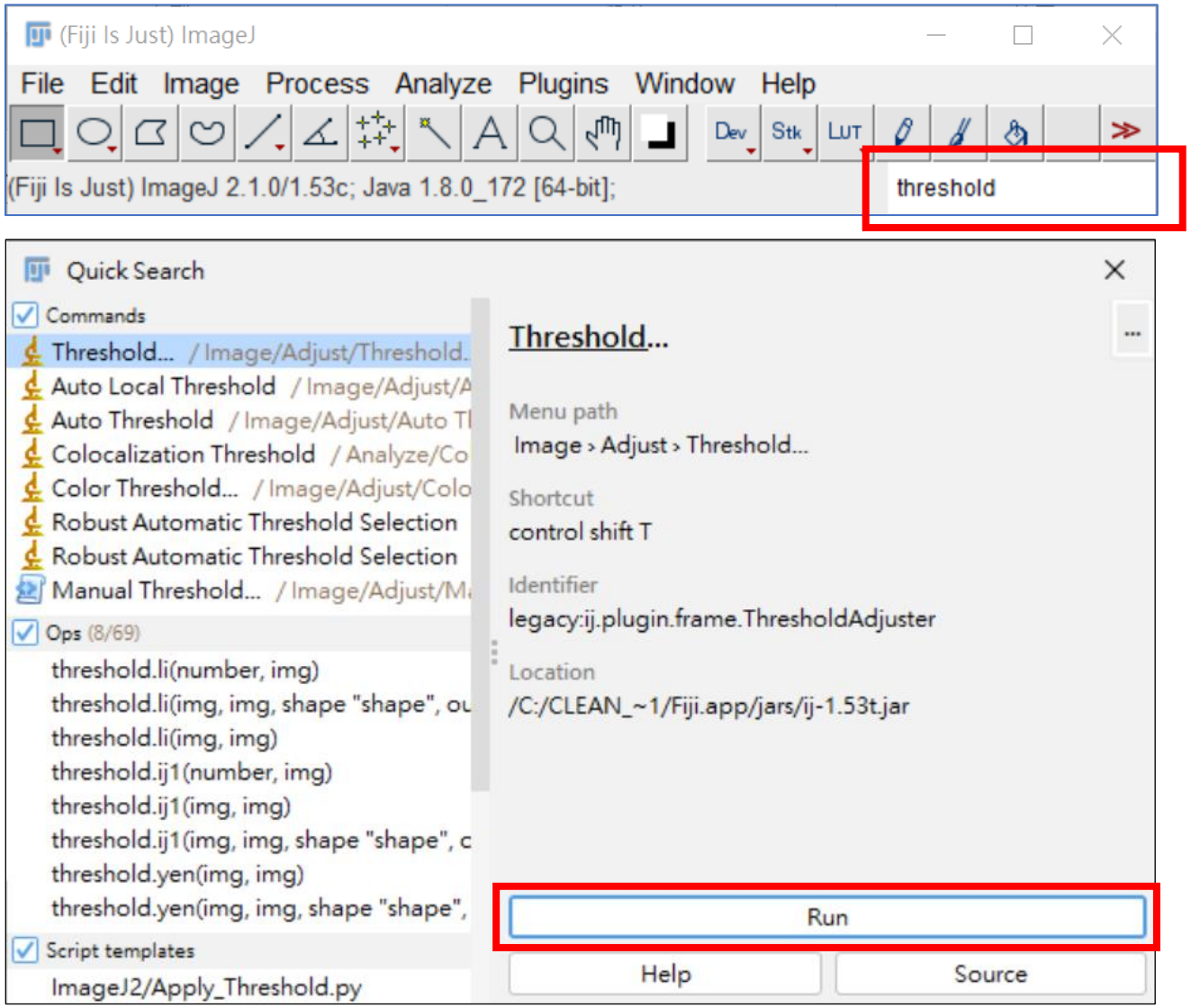
Fijis user interface

- There are more tools in the toolbar than expected…
- Use the right click or double click to discover them!



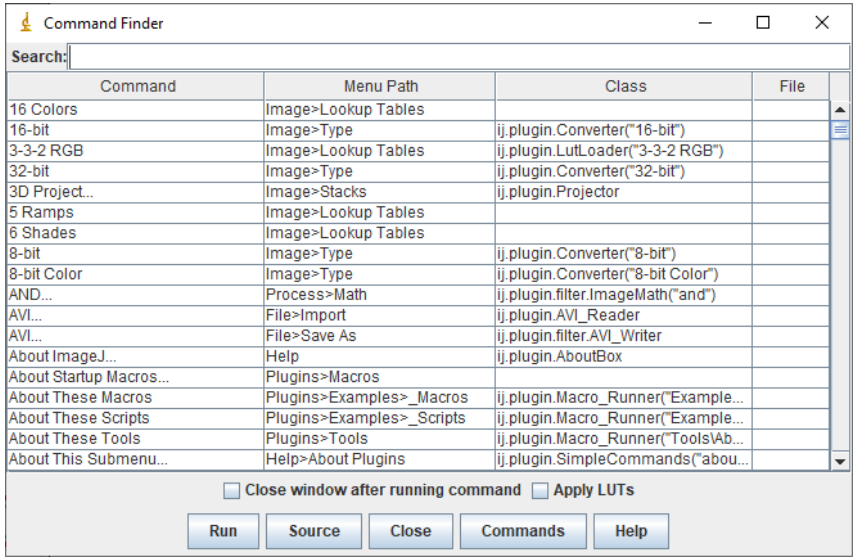


Search Bar

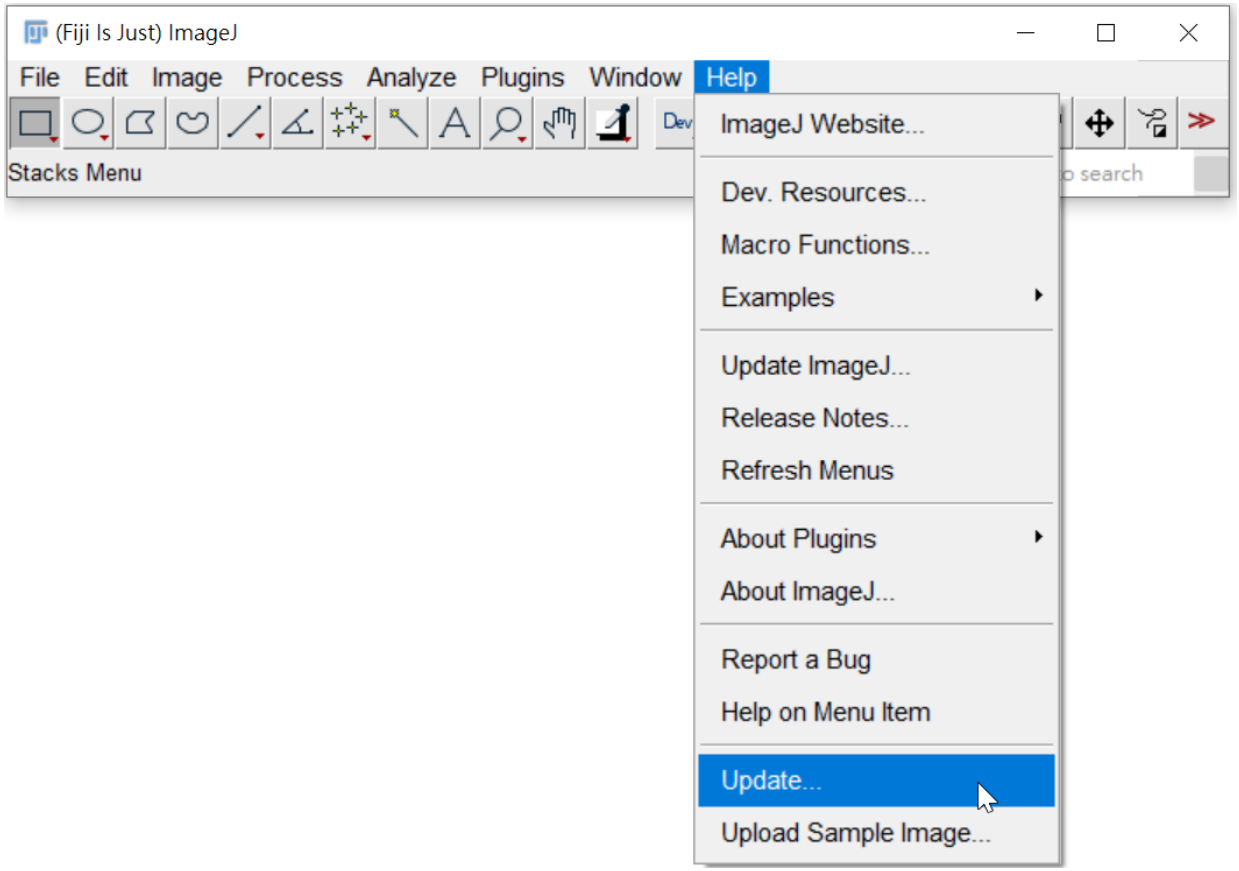


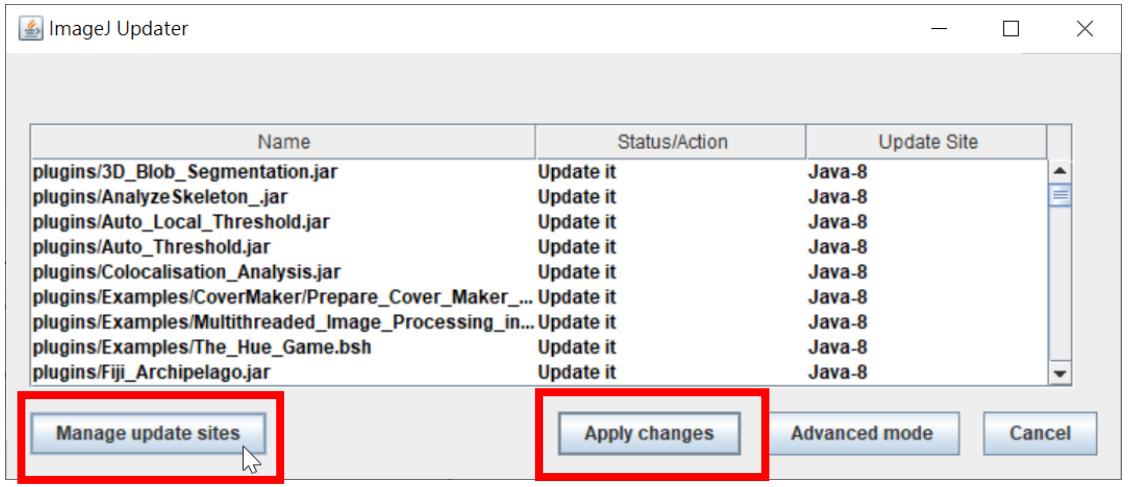
- It shows you where the plugin is located
- You can run it from here (press enter)

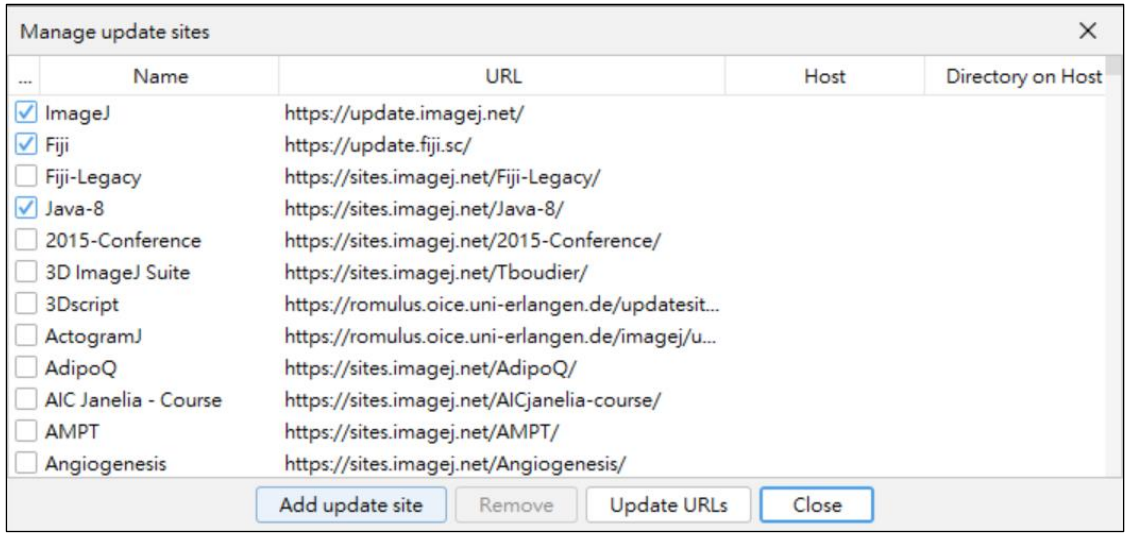
ImageJ : command finder (Hotkey “L”)



FIJI updater







- Keep a backup of working version before you update.

http://imagej.net/List_of_update_sites

Loading Data

Drag to FIJI and release

- Generic image format
- Folder (Import Image Sequence)
- ROI (.roi, .zip)
- Macro files (.ijm)

Bio-Format Importer

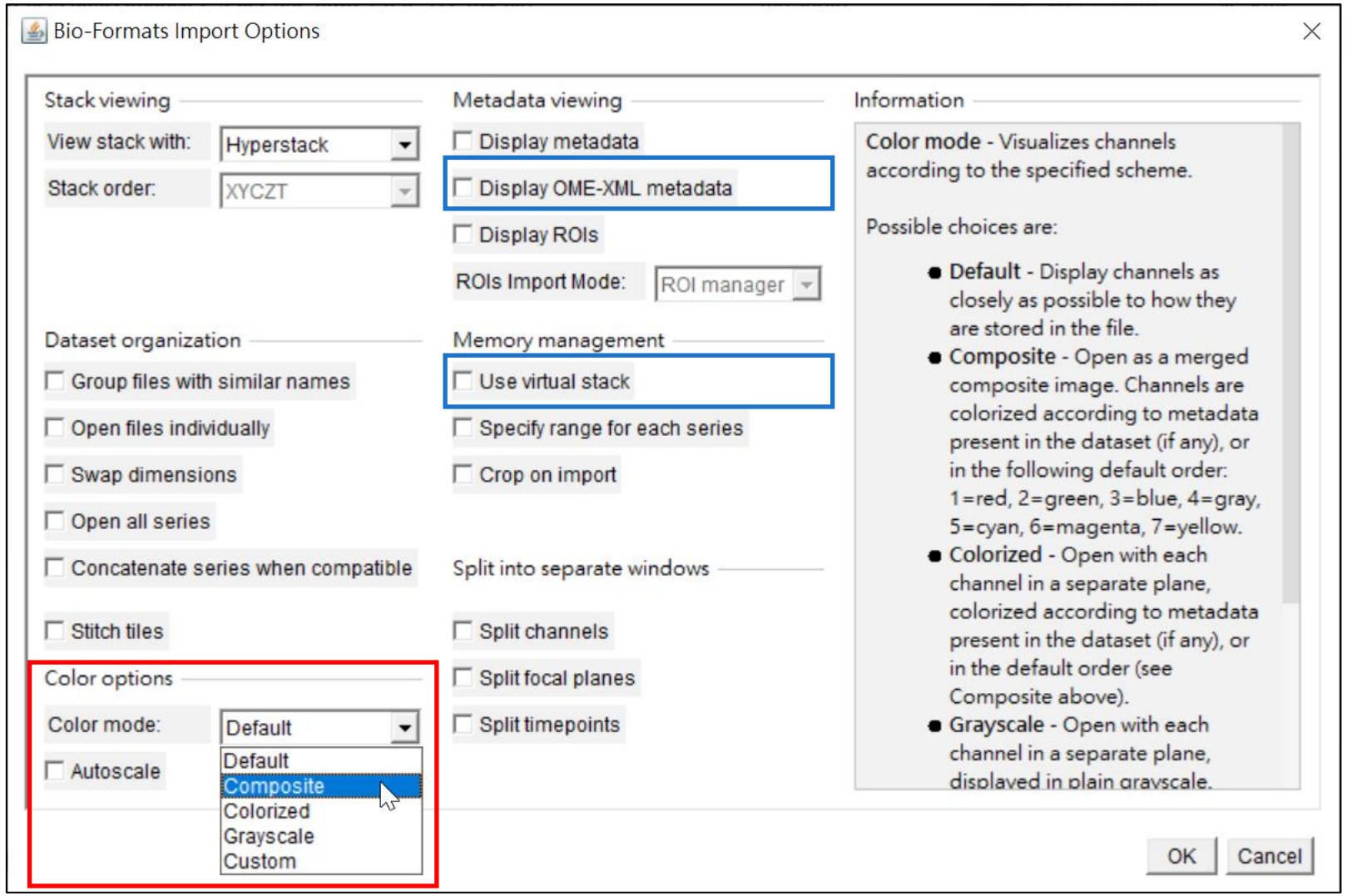
- Vender specific image format(.czi, .lif, .oir…etc)
- Huygens deconvoluted image ICS2 format (.ics)
- Big Data (Use Virtual Stack)
- Generic image format
- More details in the measurement “label”
  (Channel, Z, T, ROI)

Generic image format

- Tiff: 8 bits, 16 bits, 32 bits per channel
  24 bits RGB
  Can keep part of metadata (e.g. Pixel Size)
- Png: RGB mode only, metadata loss
- Jpg: lossy compression
- Zip: compressed Tiff (lossless)

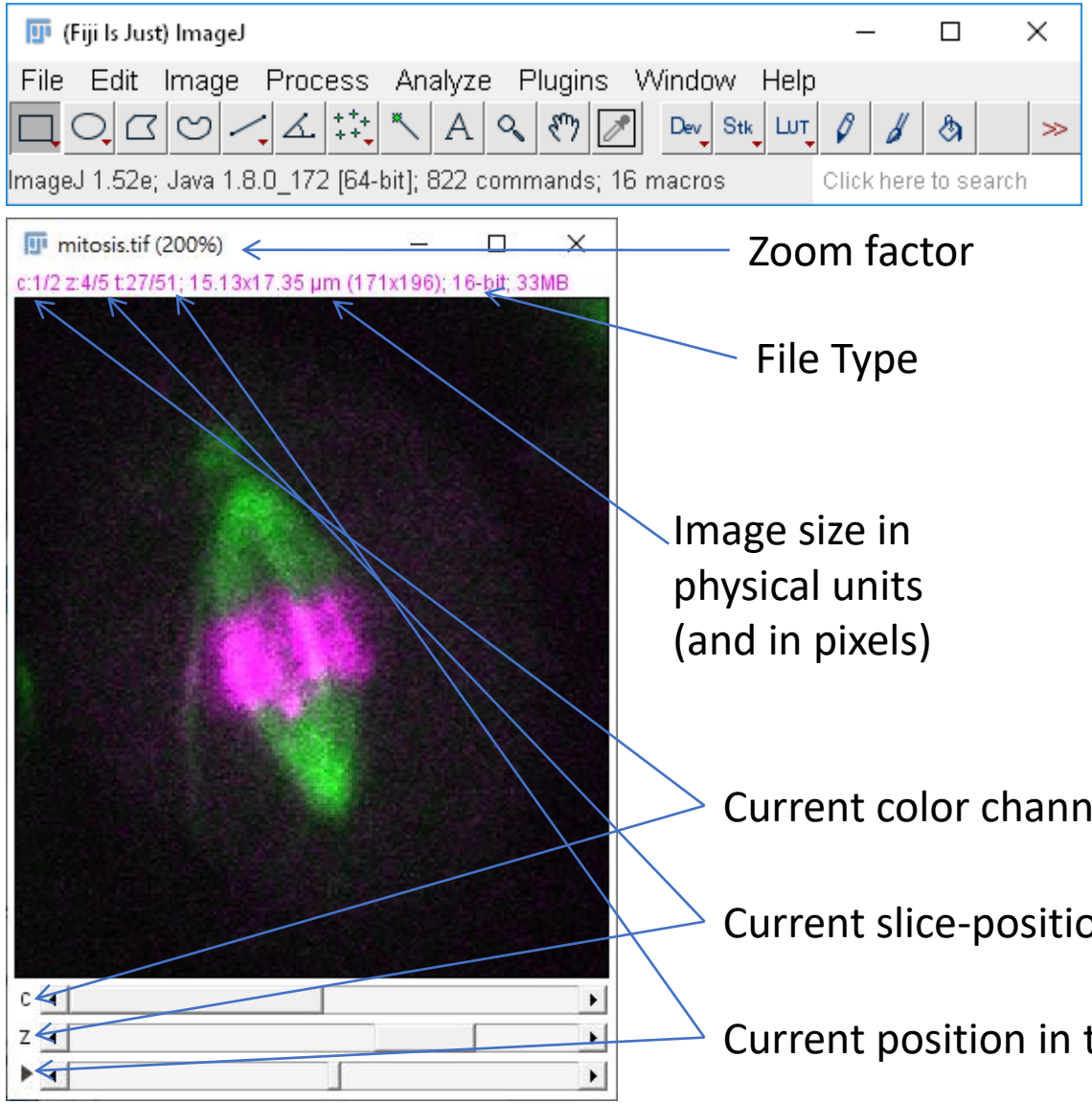
Bioformats importer

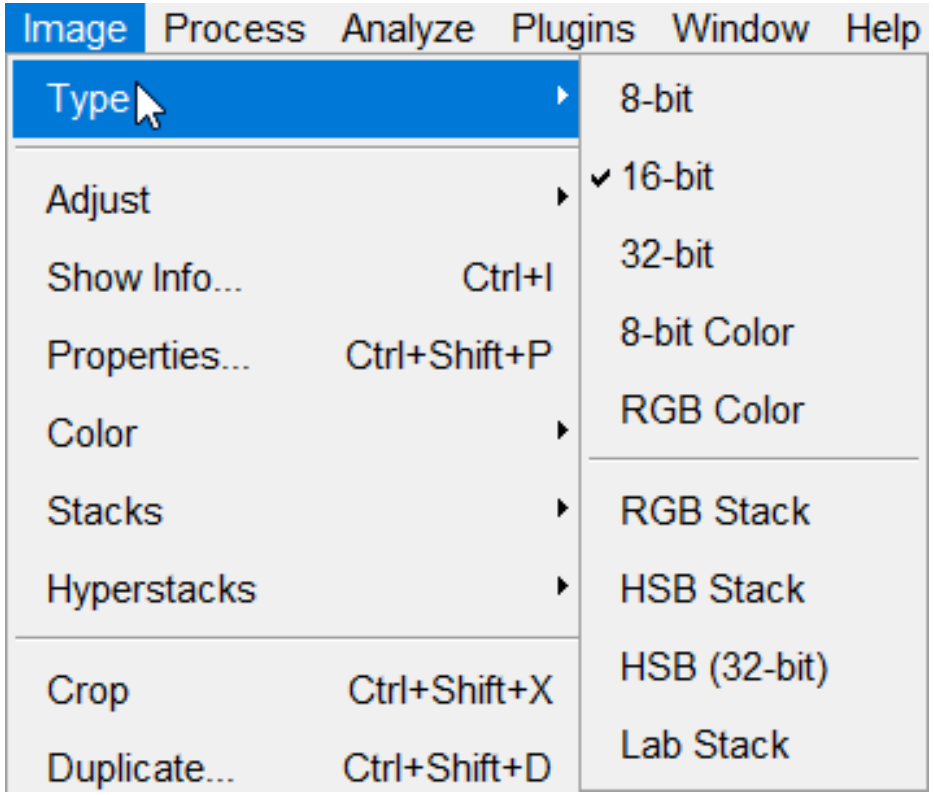
- **Composite** is useful for multi-color imaging
- **Autoscale** -> similar to auto contrast
- **Display OME-XML metadata**: easier to read image acquisition settings.
- **Virtual Stack** -> useful for large image
(Lazy loading)



Fijis user interface

Only 1 active window, 1 activate channel





Useful tools when you open an image

Image -> Channels Tool

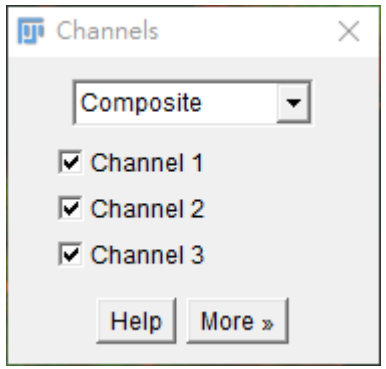
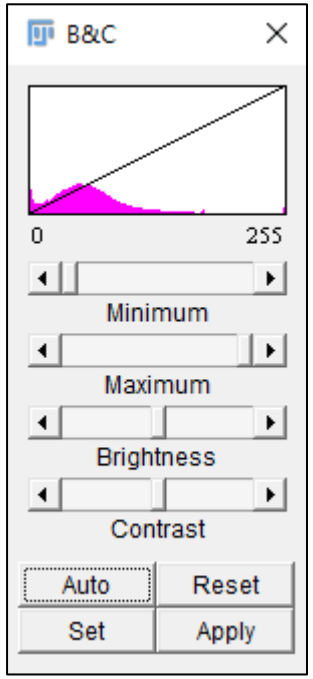
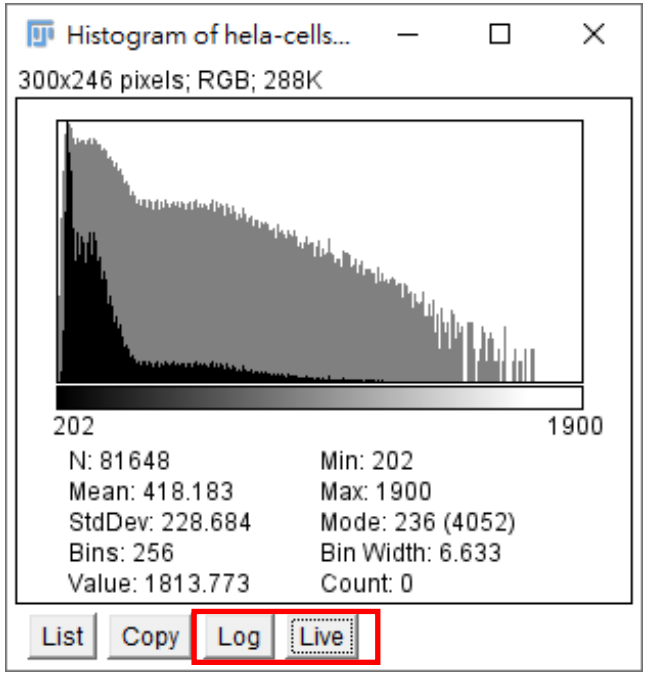


Image -> Adjust -> Brightness & Contrast



Analyze -> Histogram

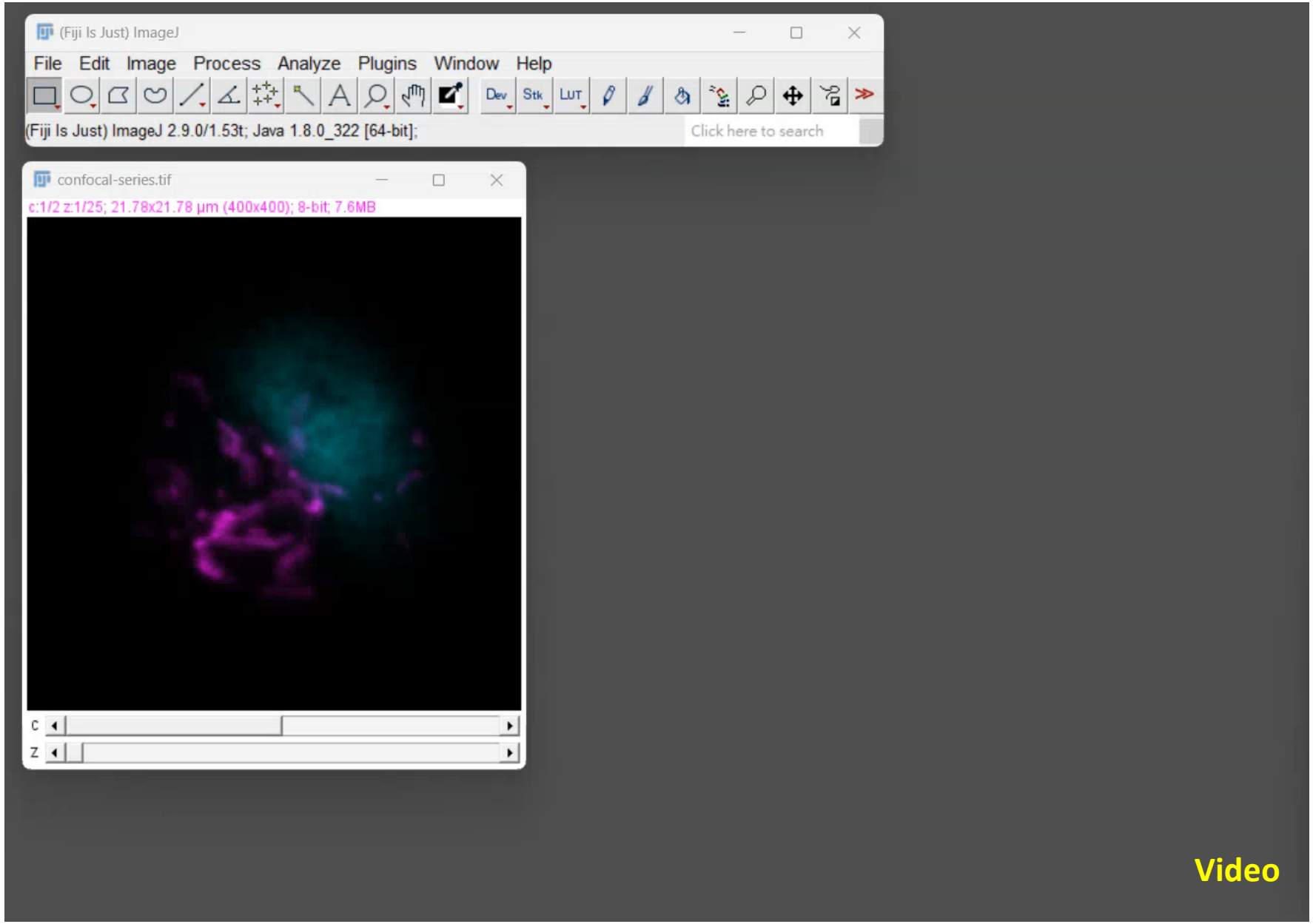


Explore your image

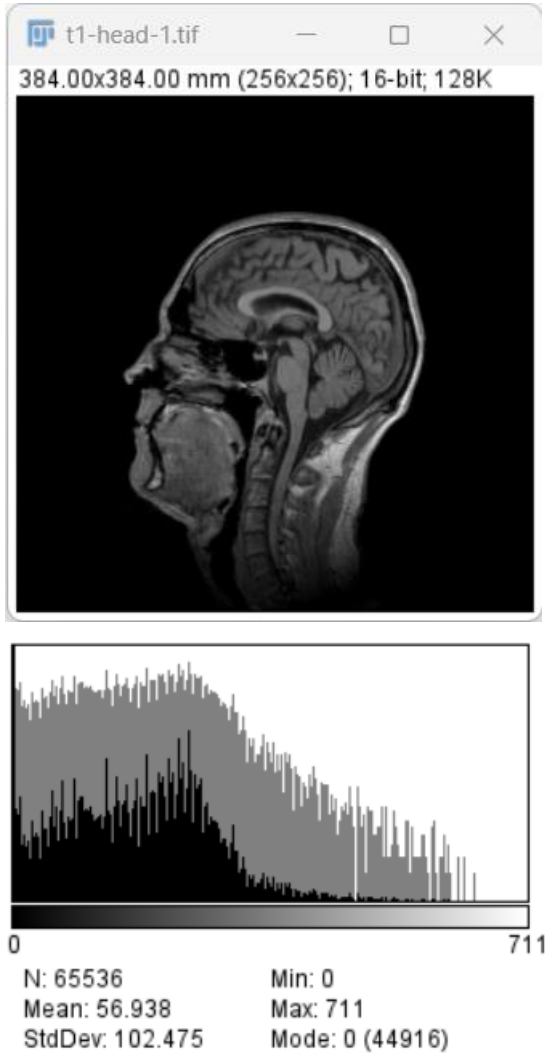
Channels Tool

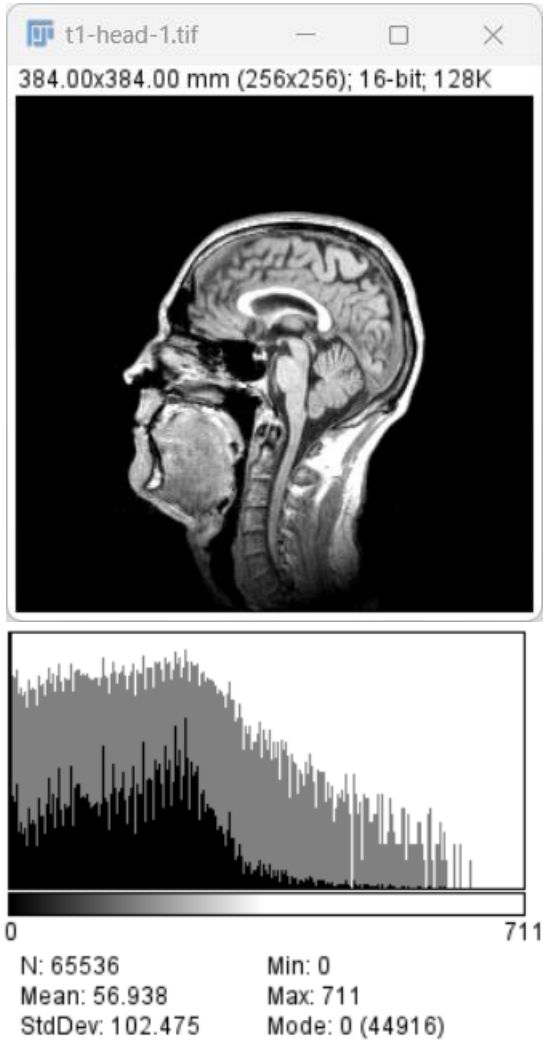
Brightness & Contrast

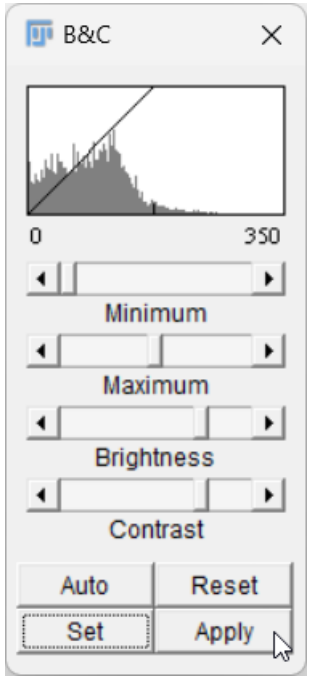
Histogram

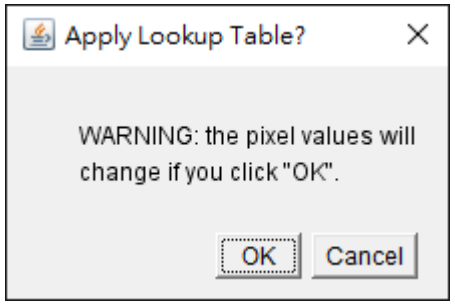


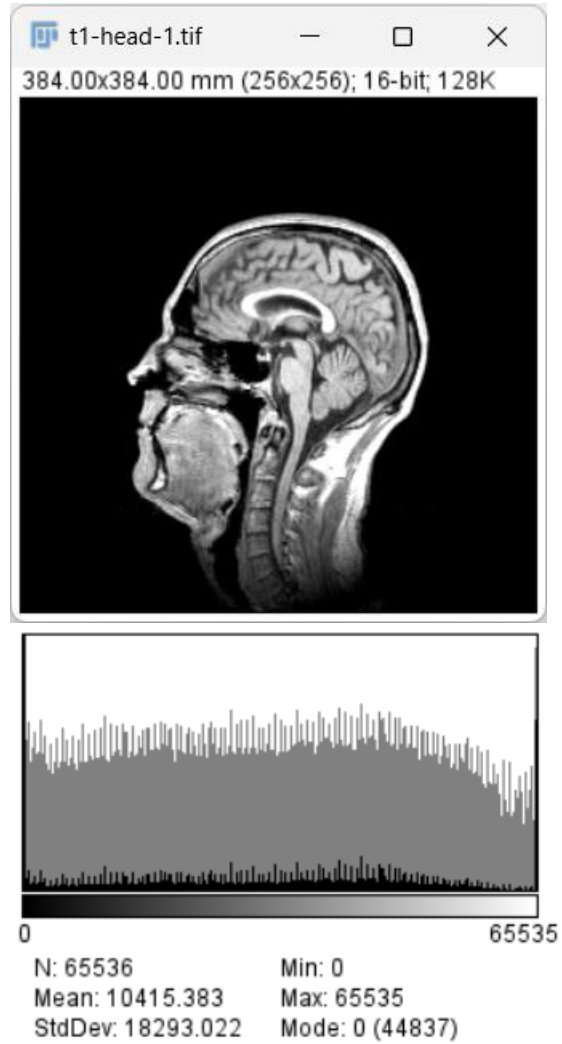
Don’t “Apply” when you adjust the Brightness & Contrast



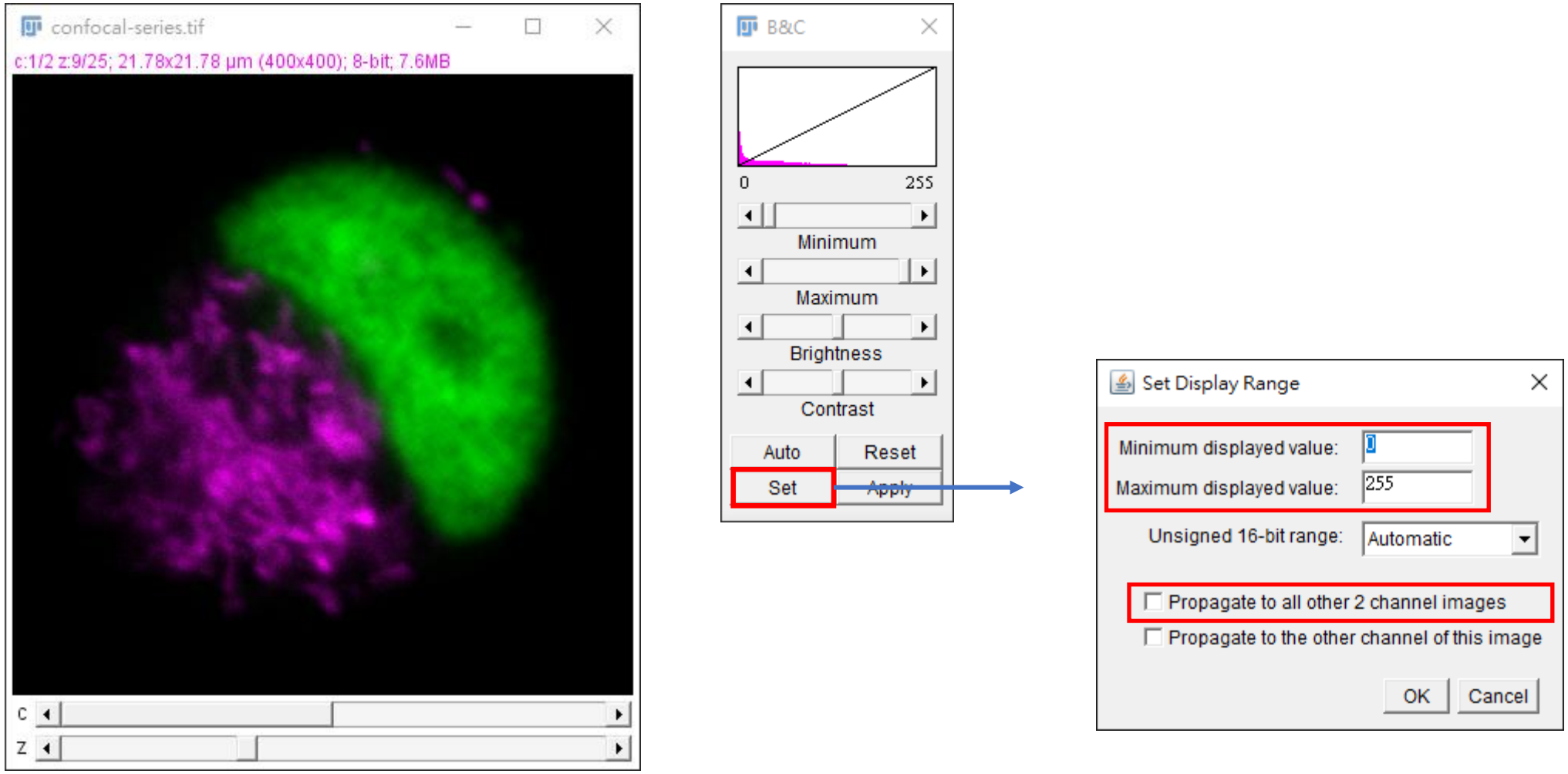






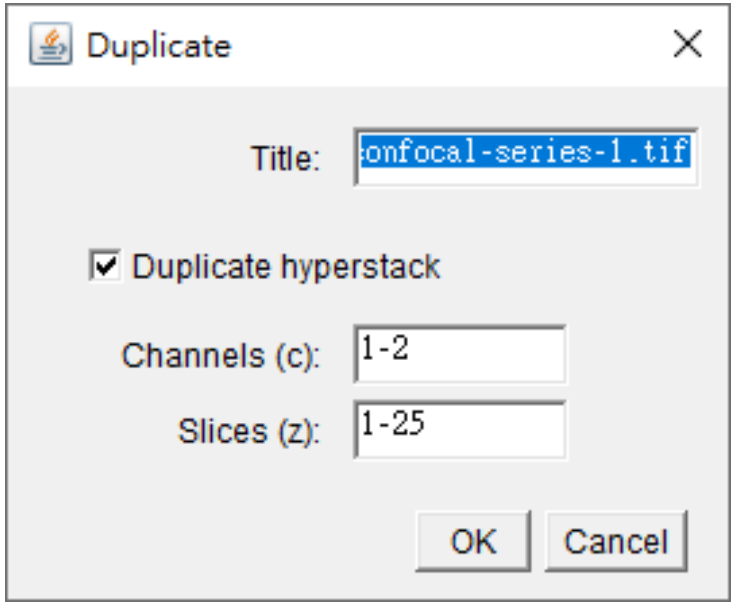


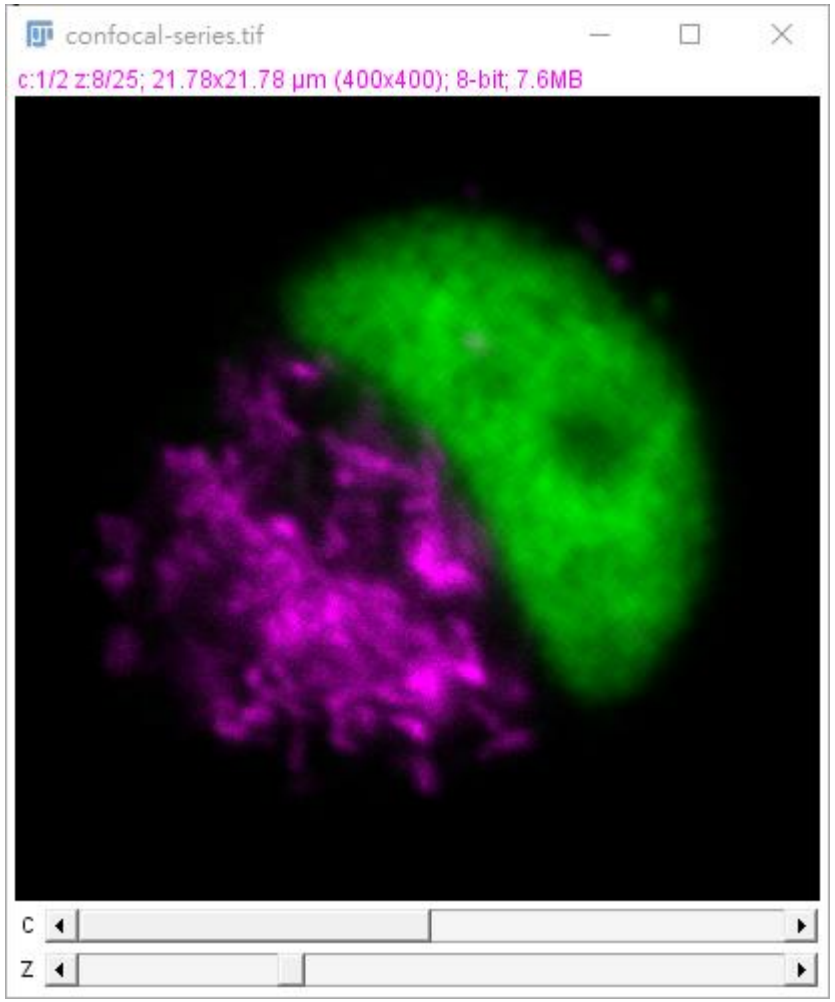
Use the same display range for the image you want to compare



Duplicate

- Image -> Duplicate ..

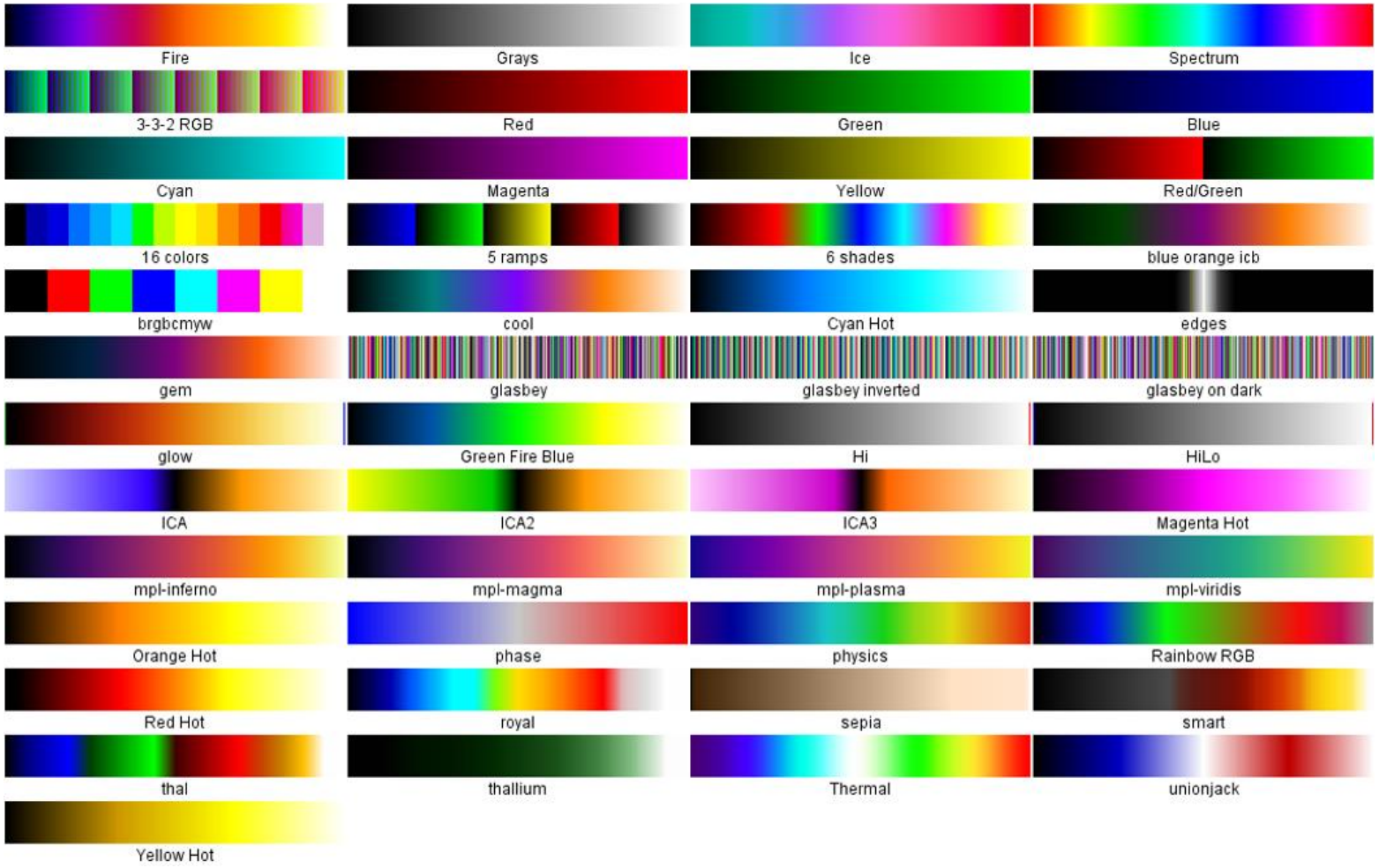




LUT

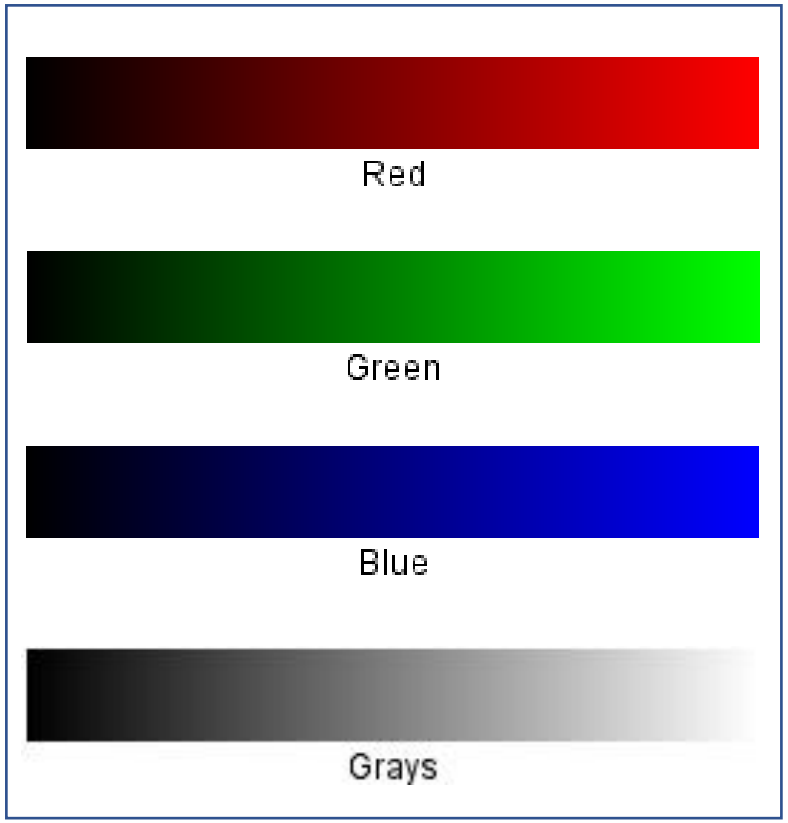
- **Image -> Lookup tables**

- **Image -> Display LUT**



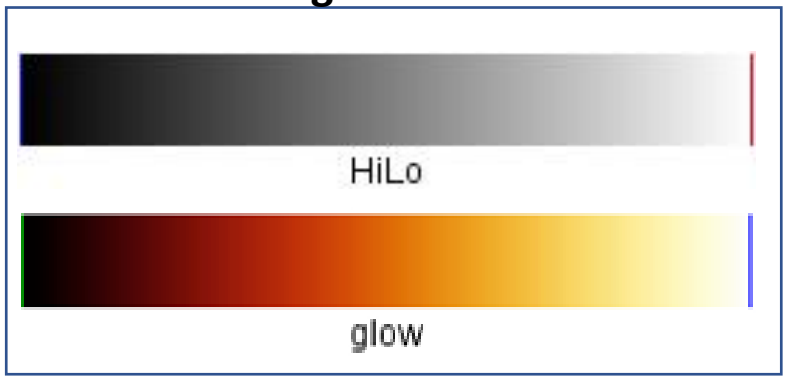
Useful LUT

"Classic"

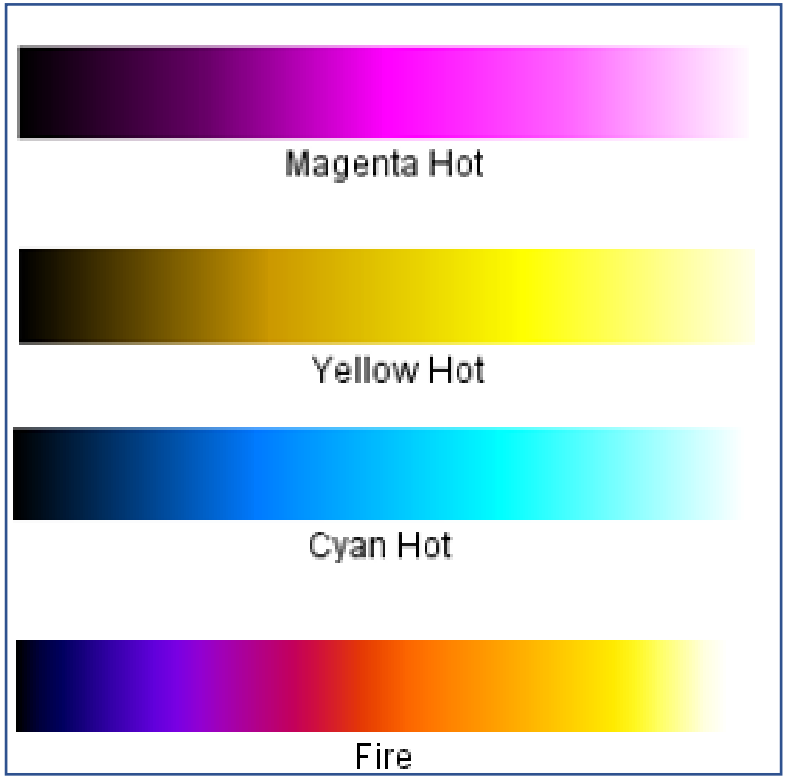


Color Blindness friendly

Range indictor

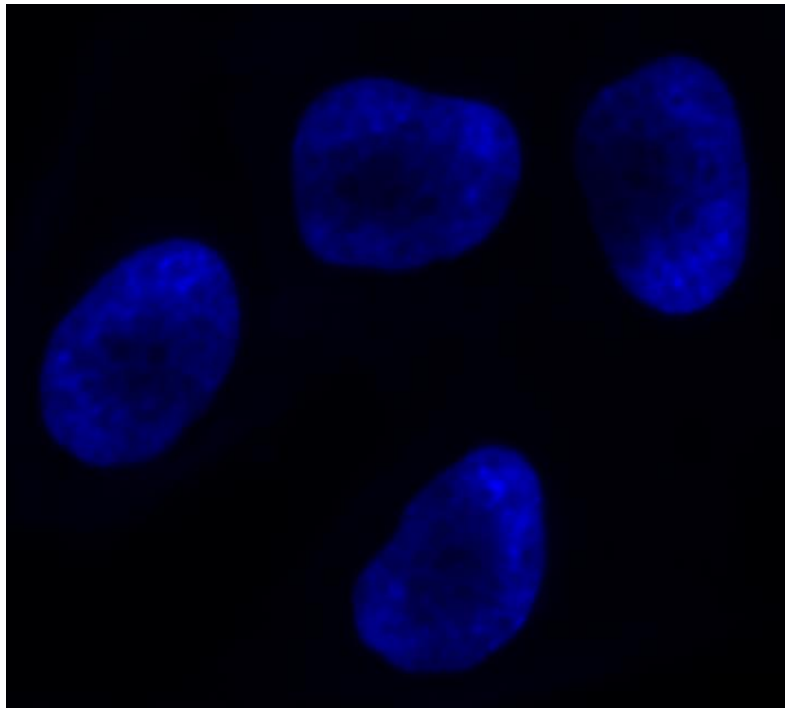


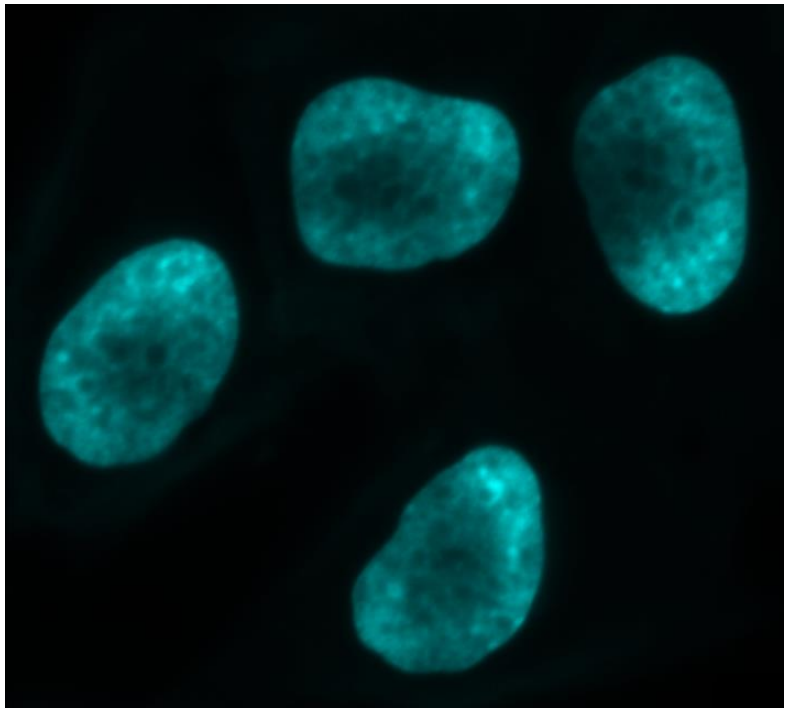
Good for weak signal visualization (nonlinear, calibration bar is necessary)

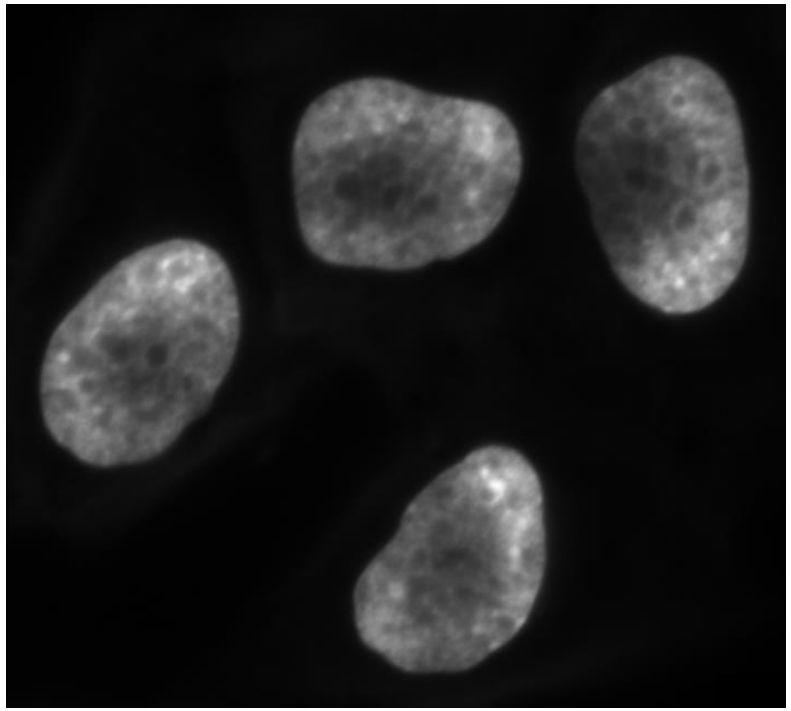


For labeling/ count mask

Our eye has different sensitivity to different colors





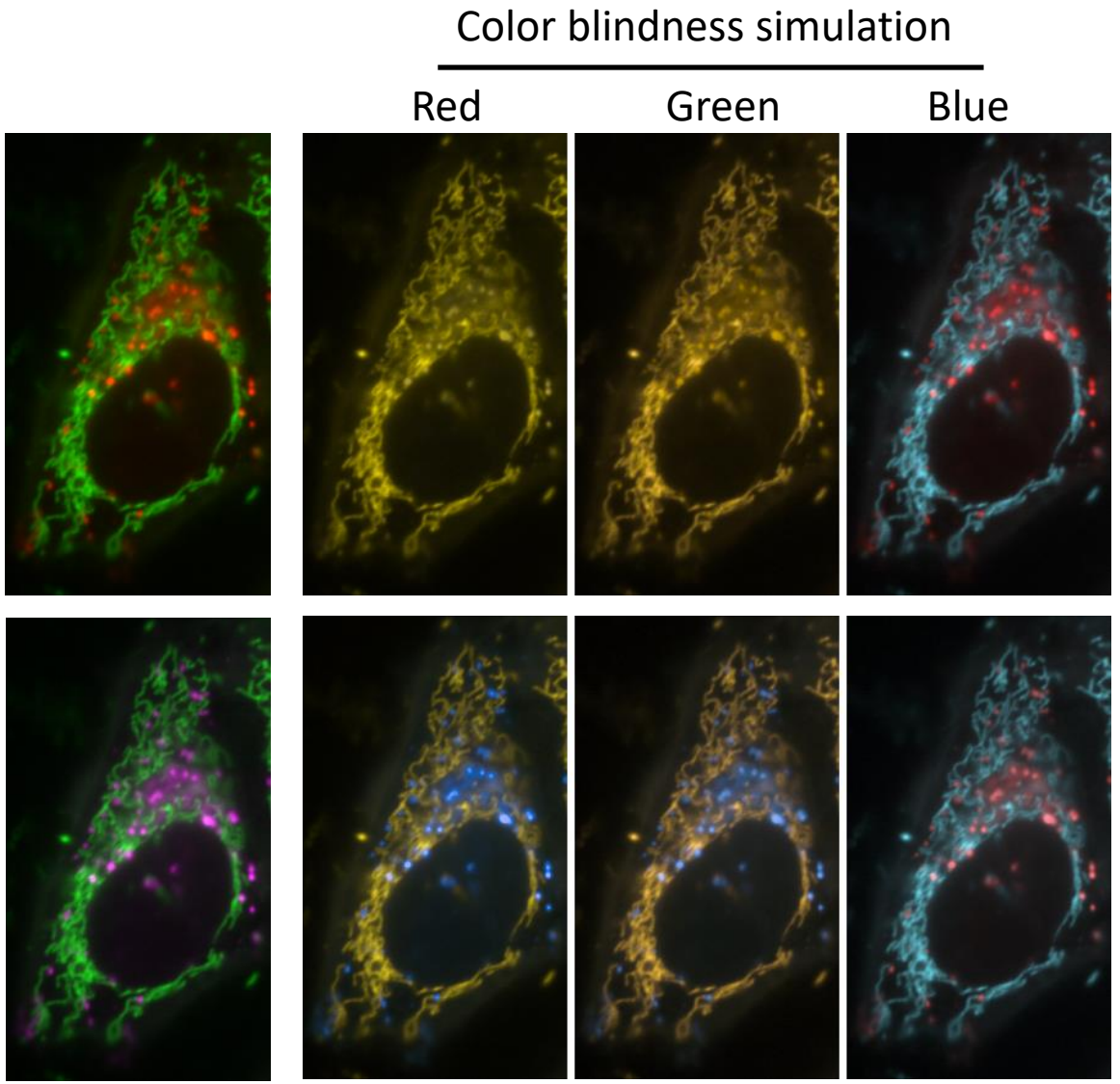


Be friendly to colorblindness

Don’t use red and green in the same image!

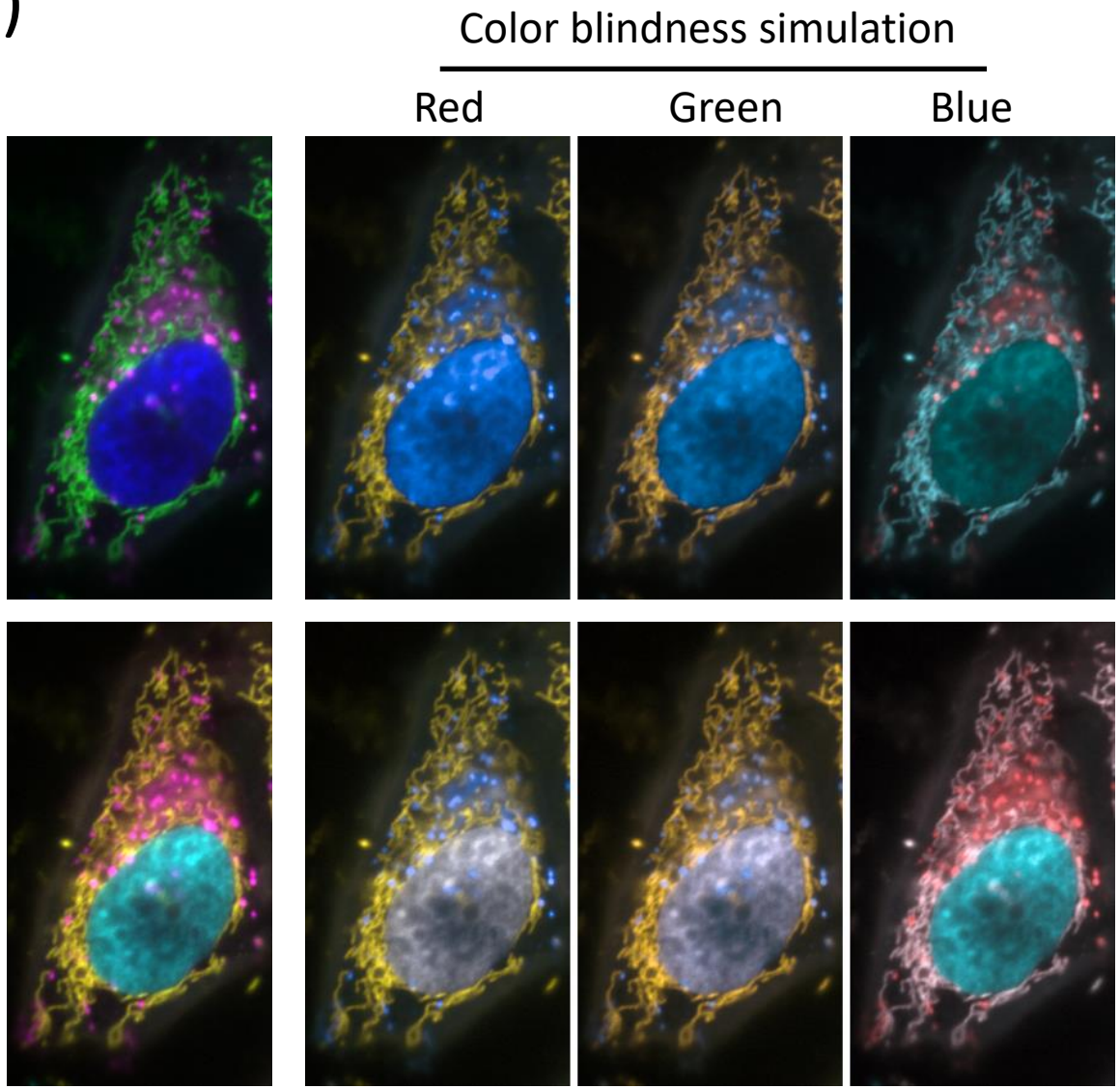
Red/Green only:

➔ Replace Red with Magenta



Be friendly to colorblindness (3 Color)

- Magenta/Yellow/Cyan
- Show grayscale pictures of each channel. (when distribution of each signal is interesting)
- Show combinations of two channels in magenta and green. (when spatial correlation between channels is interesting)

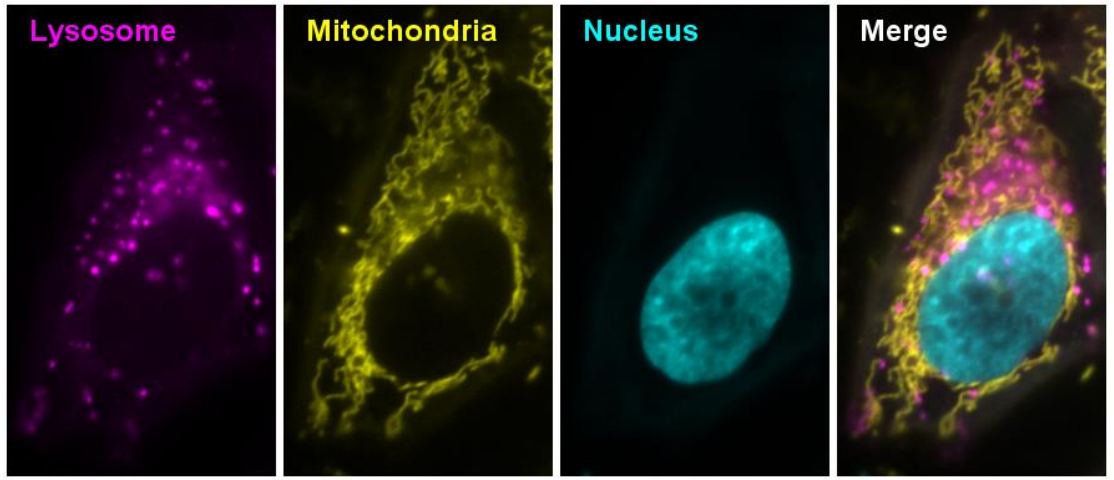


https://jfly.uni-koeln.de/color/

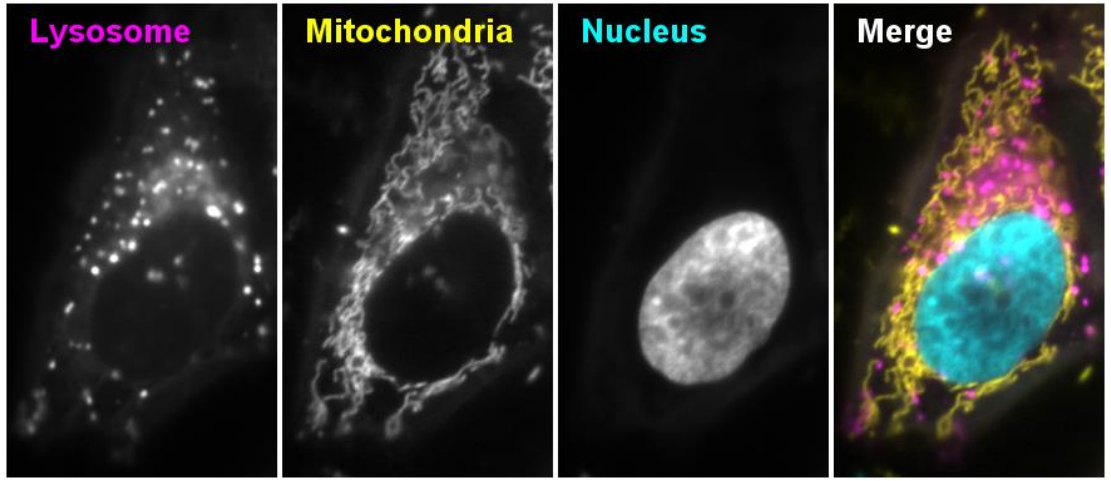
Colorblindness friendly presentation

- For individual channel presentation, the grey scale LUT is recommended!

Good

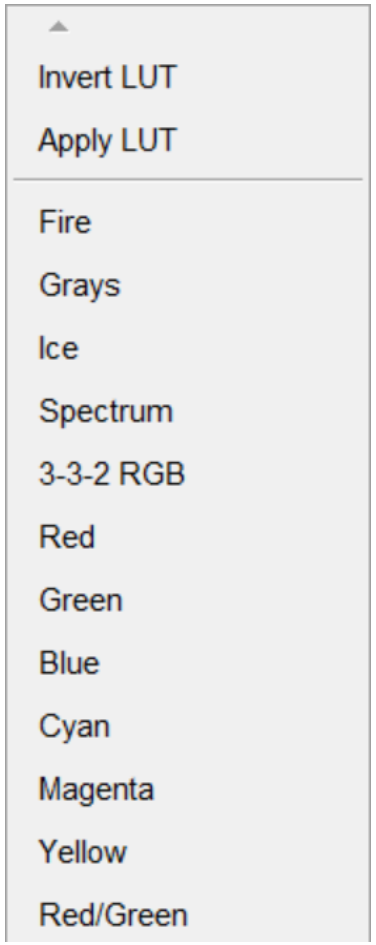


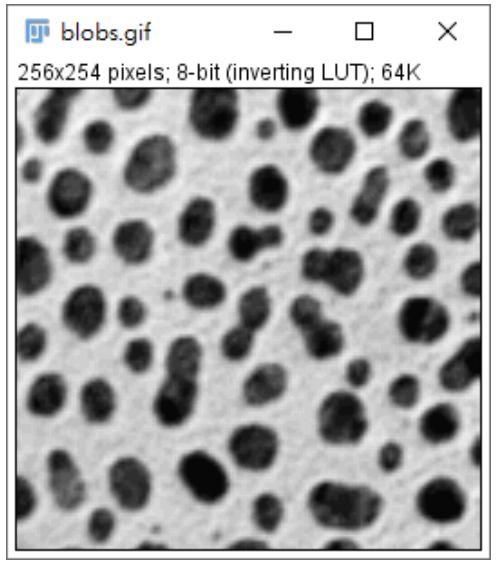
Best

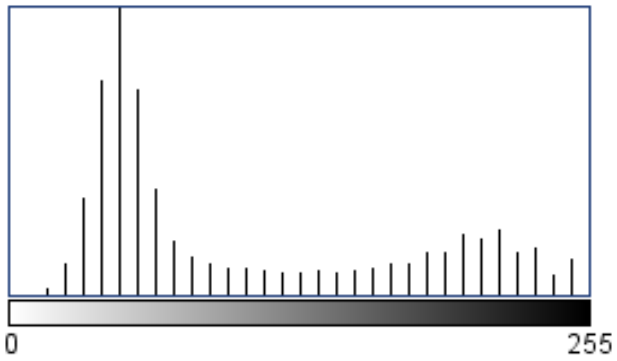


Invert LUT

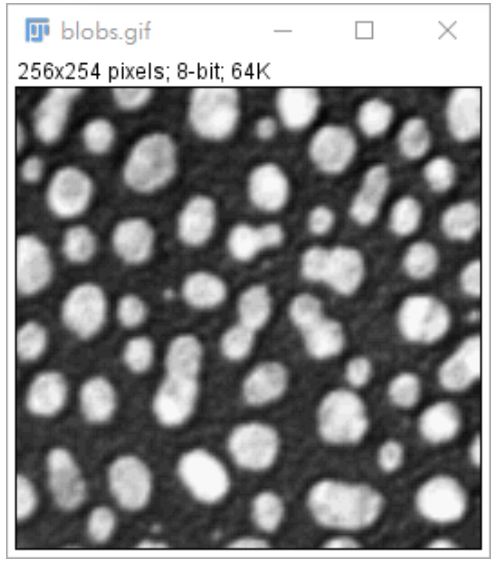
- **Image -> Lookup tables**

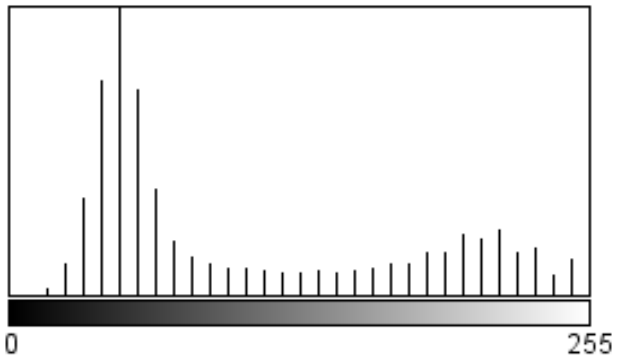






N: 65024 Min: 8
Mean: 103.269 Max: 248
StdDev: 71.057 Mode: 48 (10396)
Value: --- Count: ---

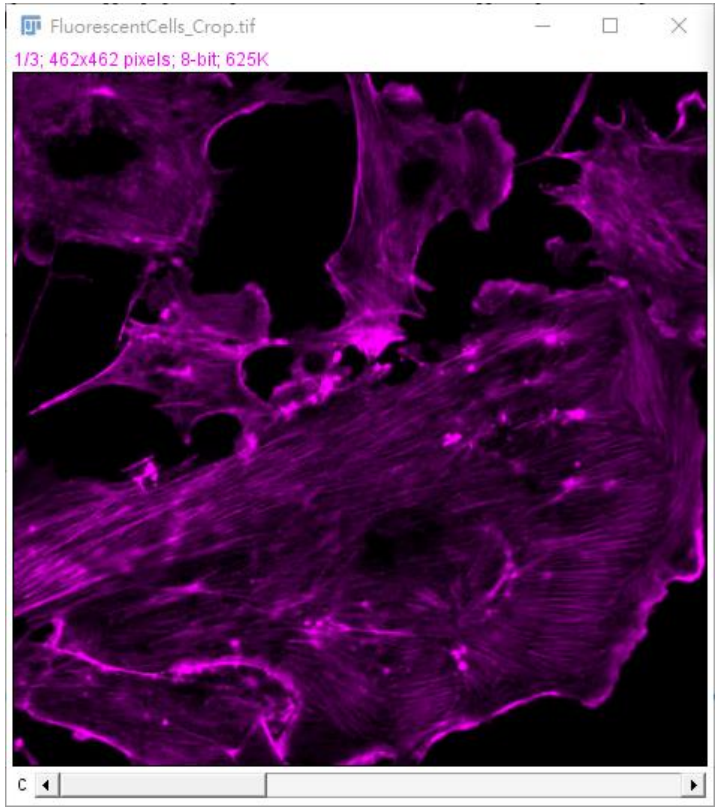


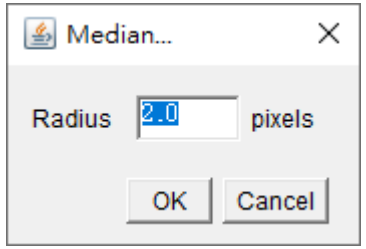


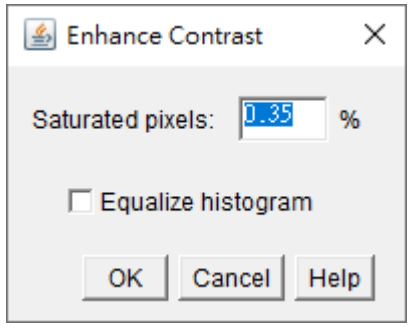
N: 65024 Min: 8
Mean: 103.269 Max: 248
StdDev: 71.057 Mode: 48 (10396)
Value: --- Count: ---

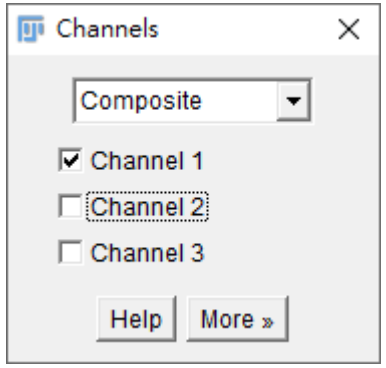
Images that look different may contain the same pixel values!

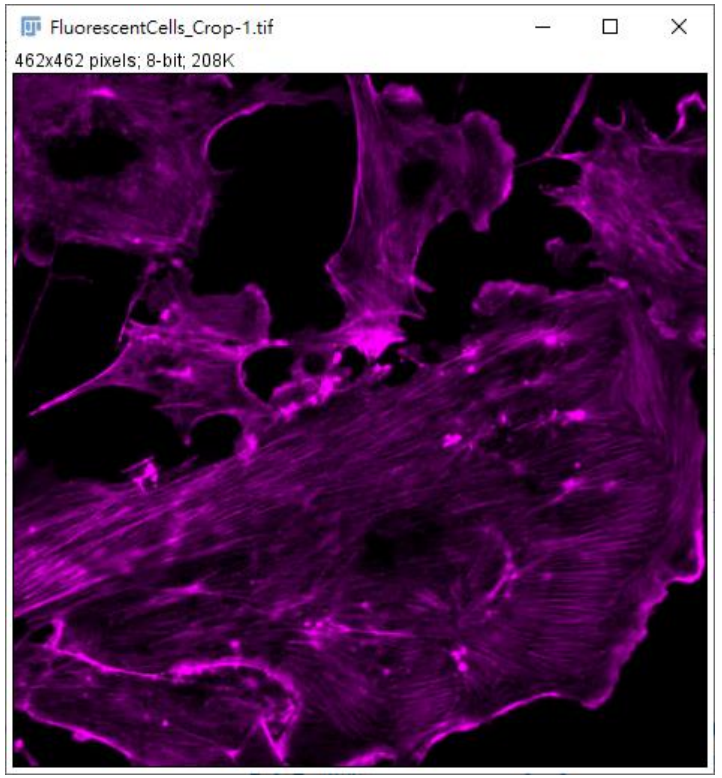
Some functions are only available when you use a single channel window

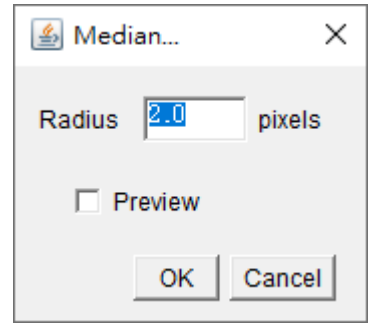


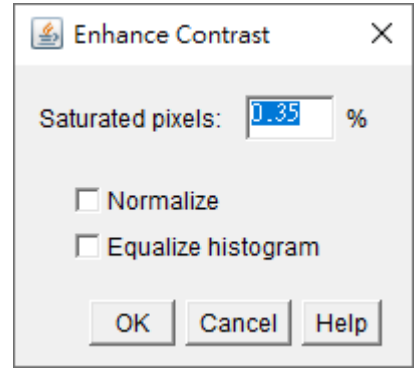






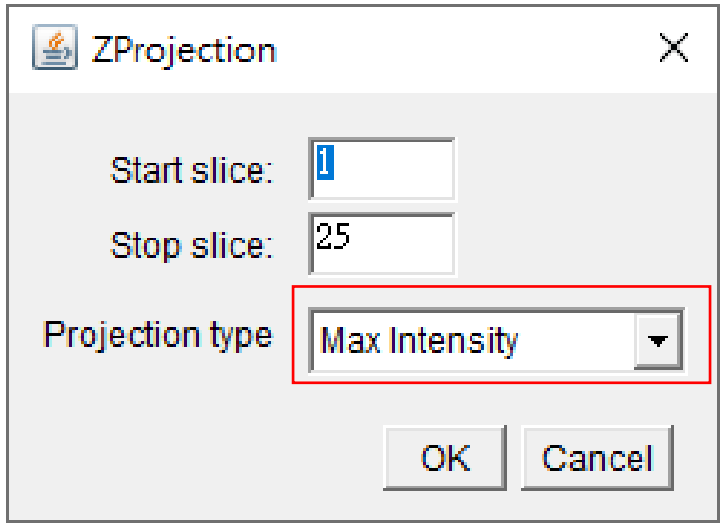


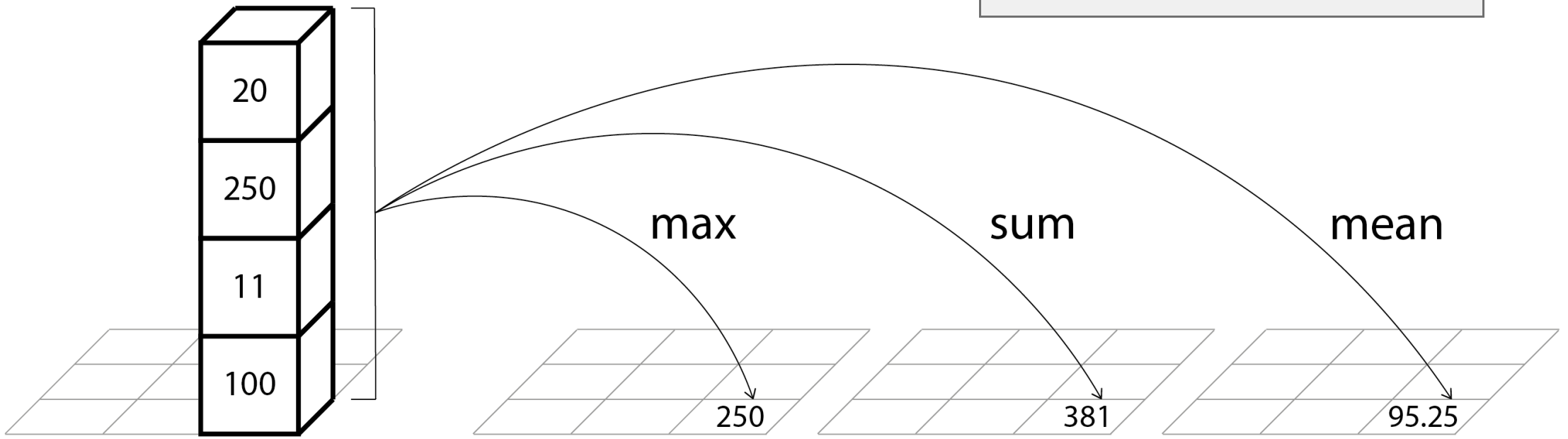




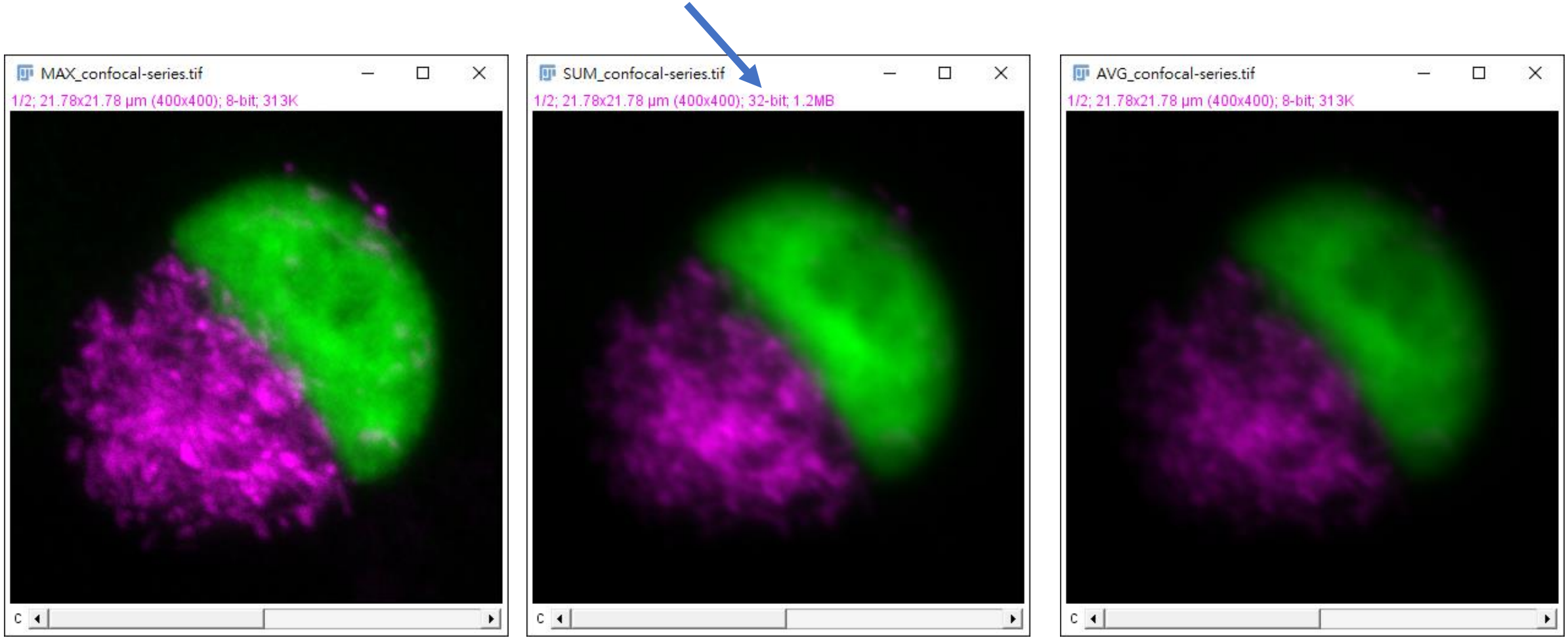
Z Projection

- Image -> Stacks -> Z Project…
- Max intension projection-> better for visualization
- Sum slices projection -> better for signal quantification (for comparison, use the same number of slices)
- Since Z projects lose 3D information, two signal colocalization may be false positive (e.g. two signals come from two different focal planes).

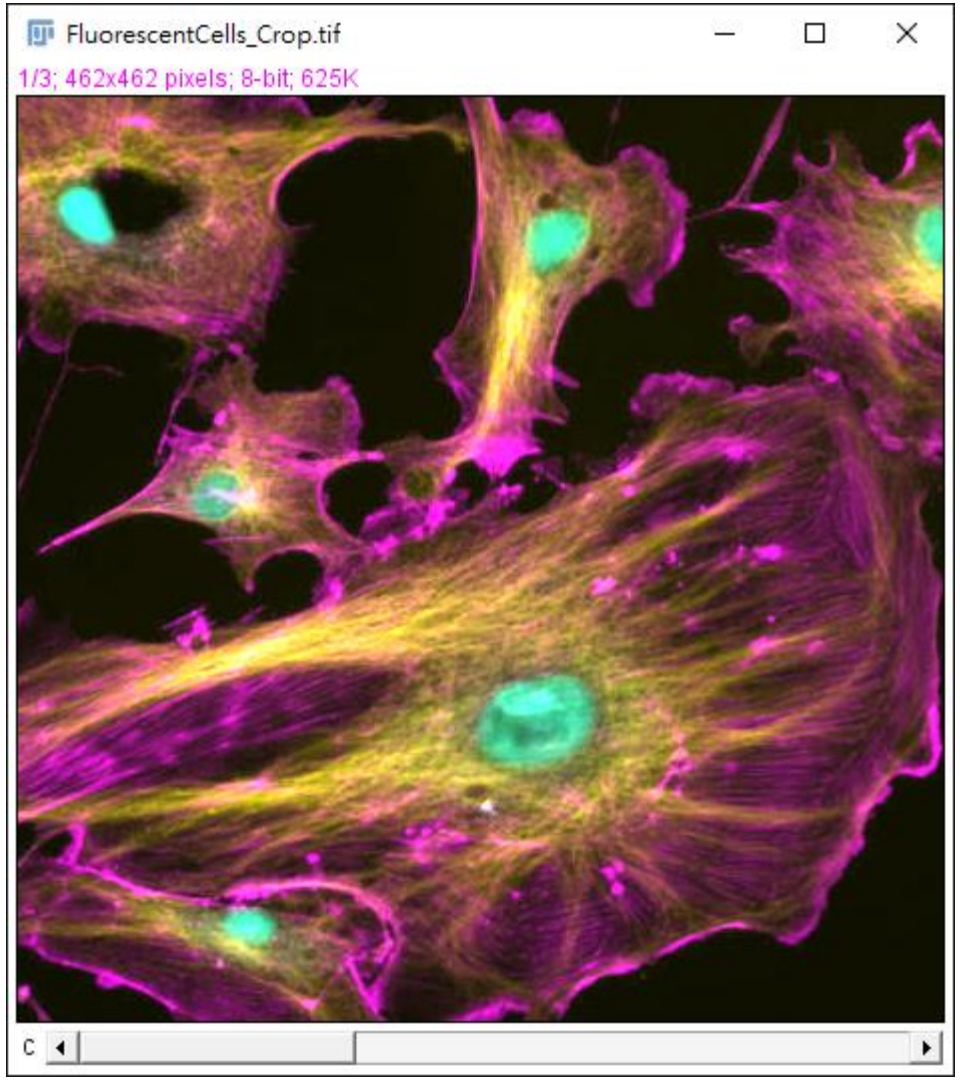


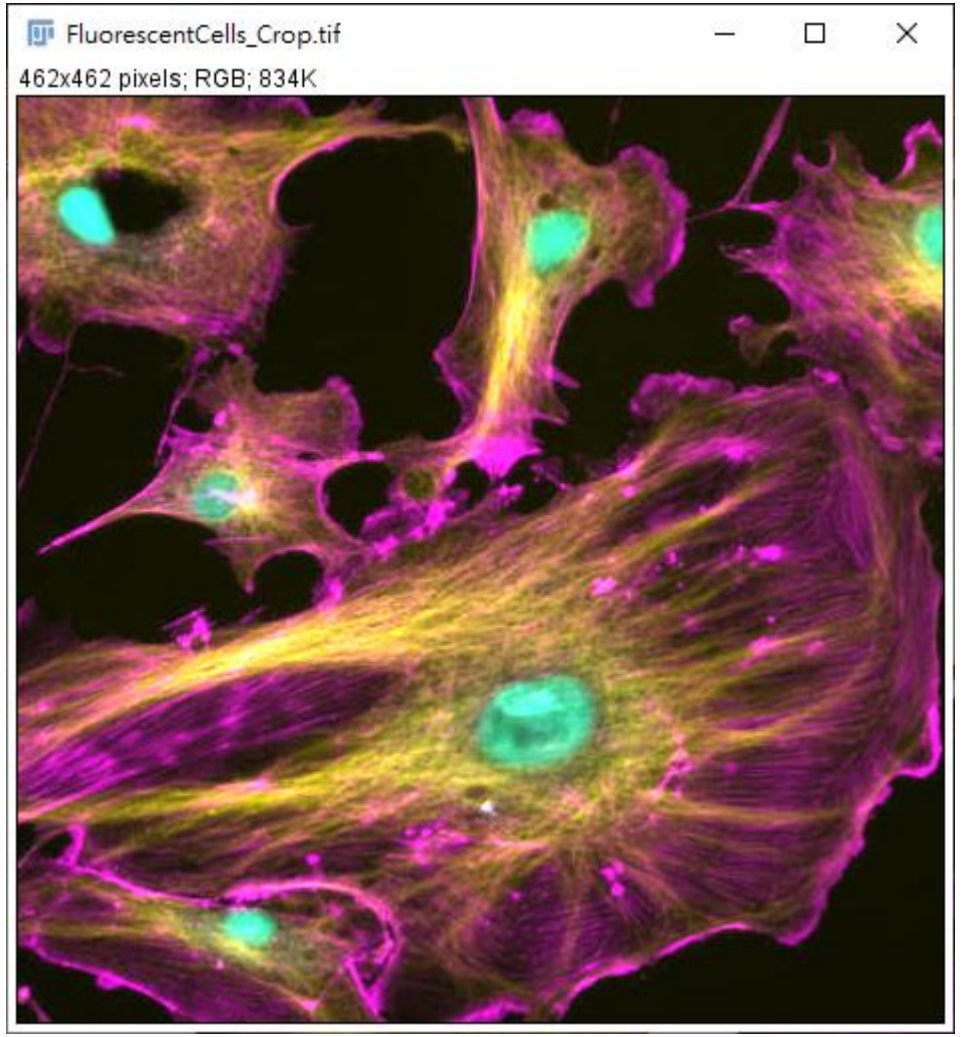


Quiz: Why Sum projection use 32-bit format?

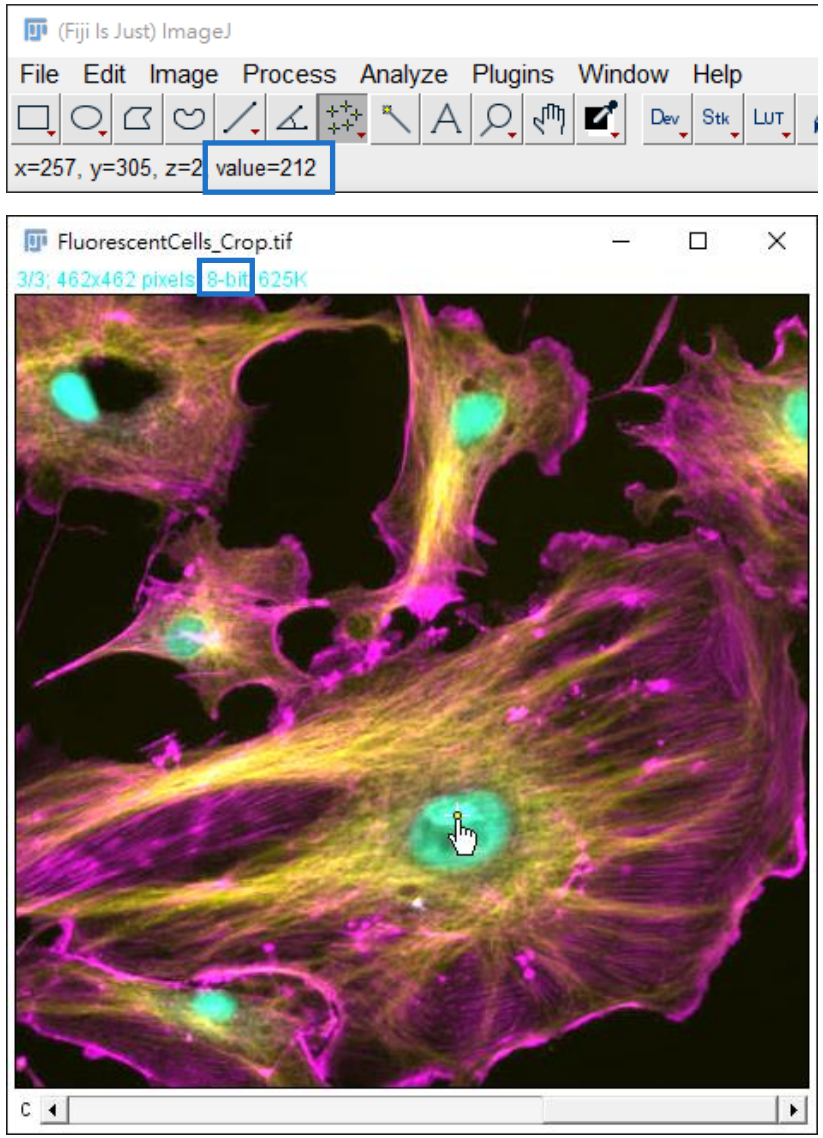


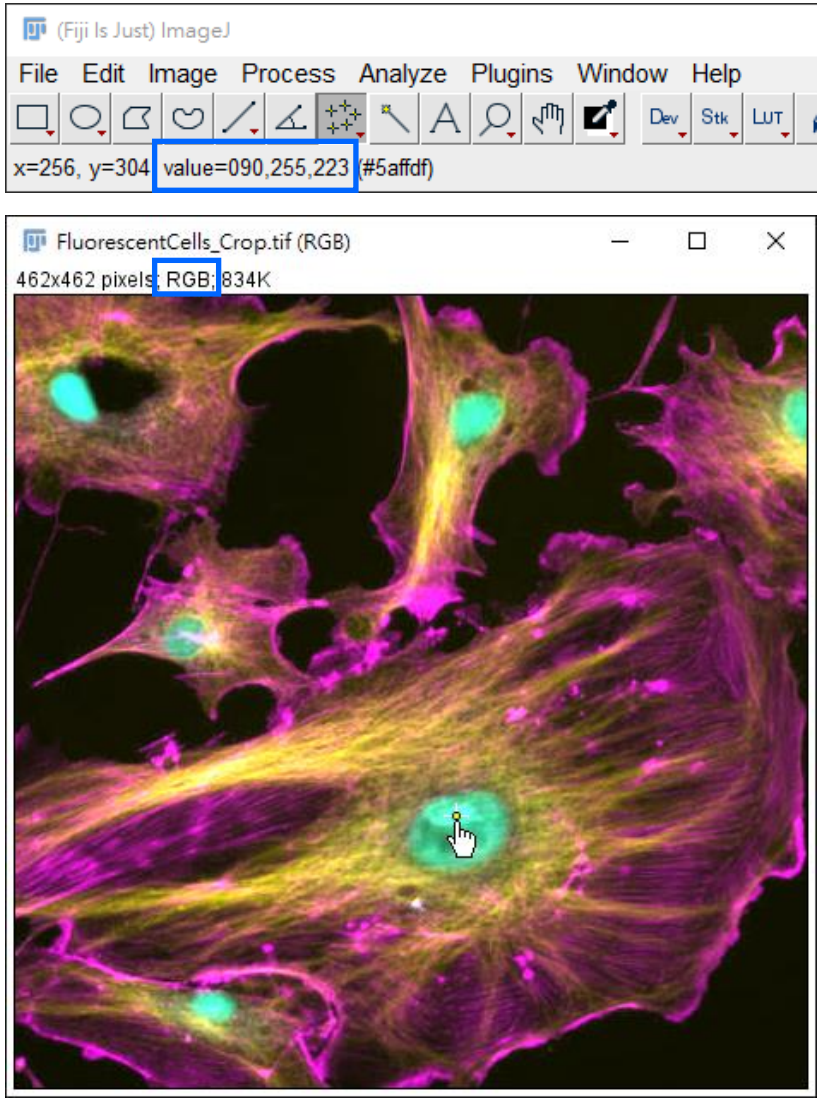
Quiz: Which one is suitable for signal intensity quantification?





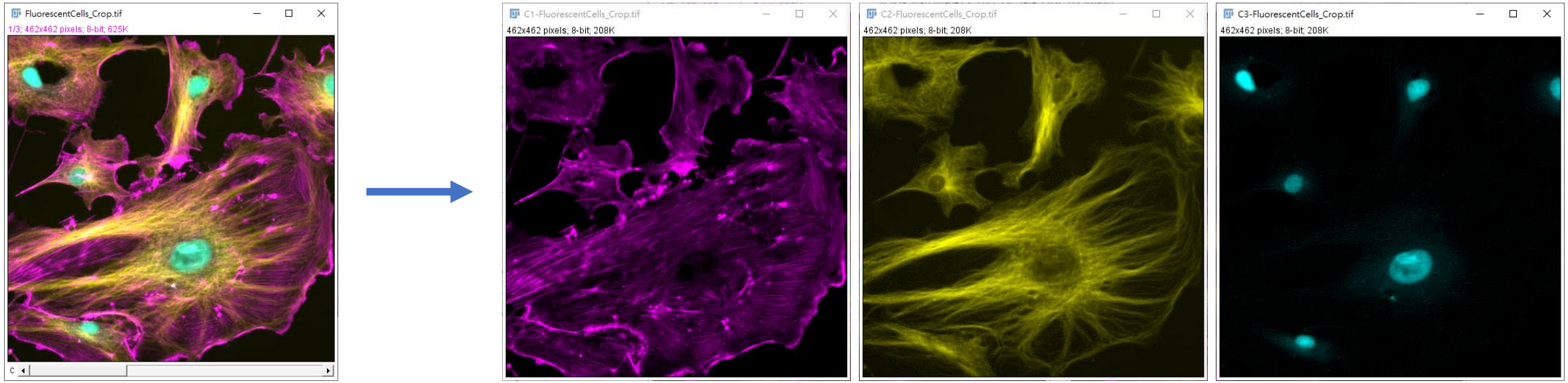
In general, don’t use RGB format for quantification

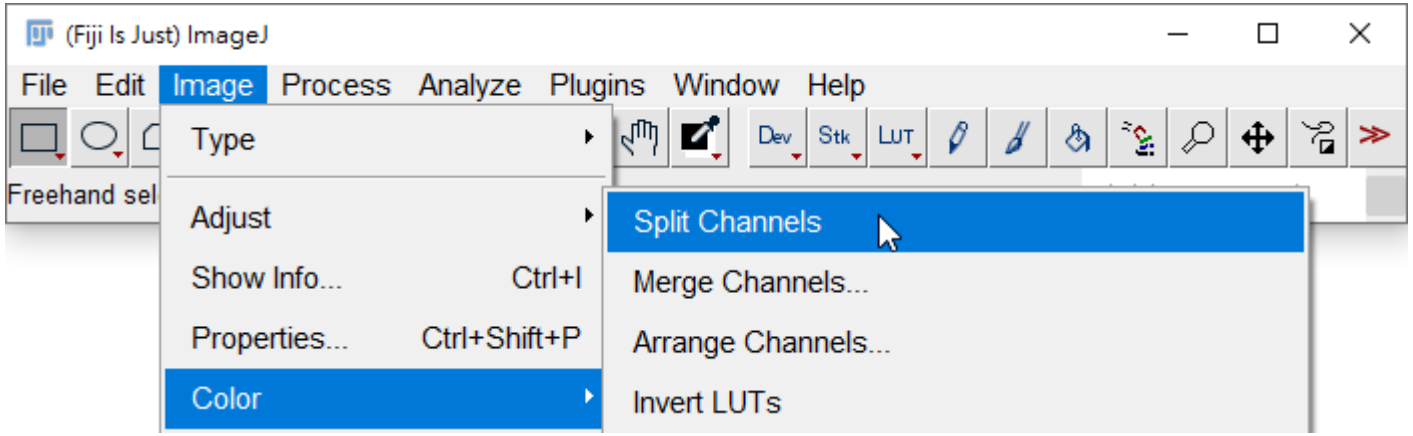




Images that look the same may give you different value!

Split channel



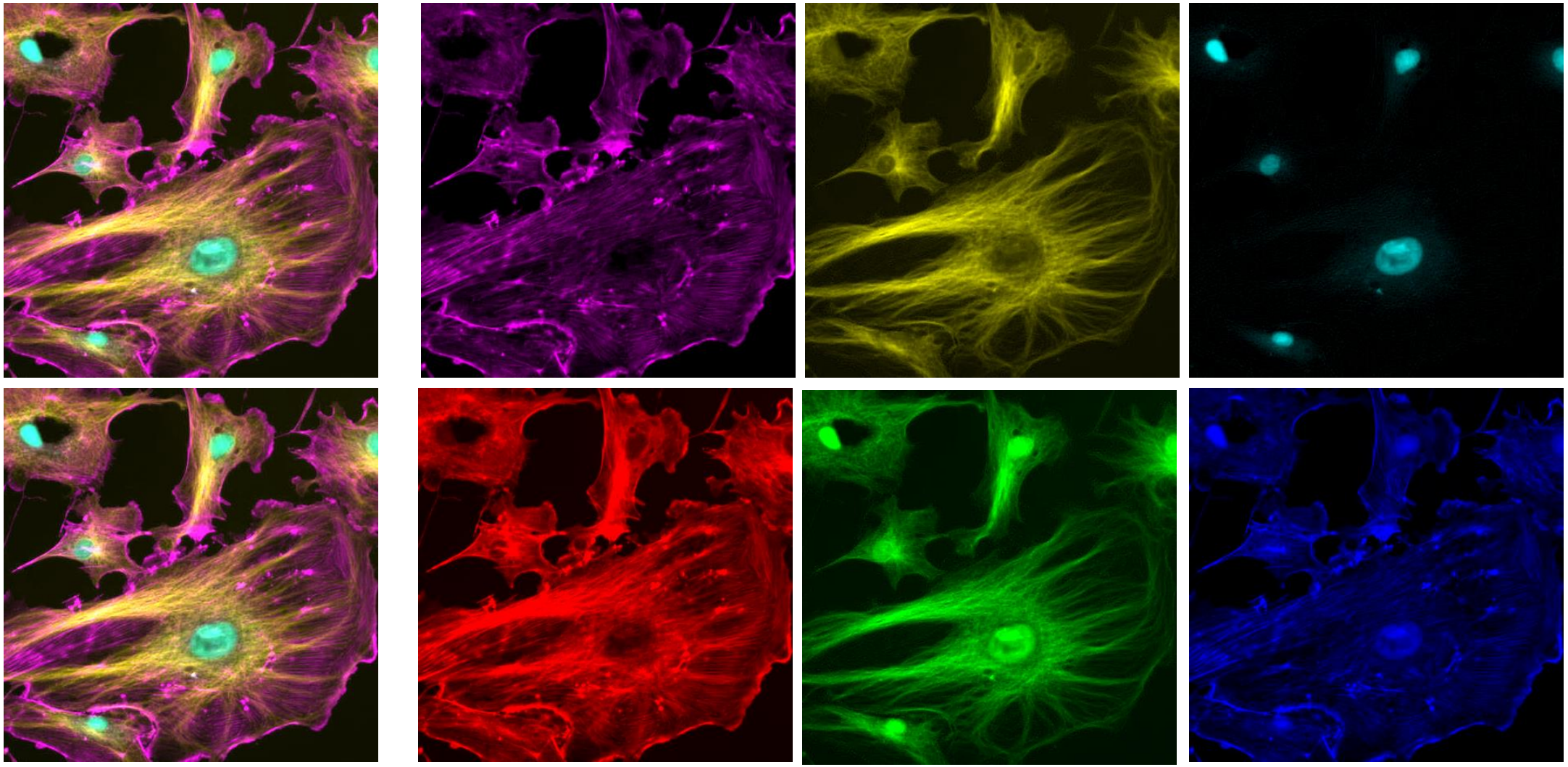


Original data vs RGB format

Split Channels

3 Channel Composite

RGB

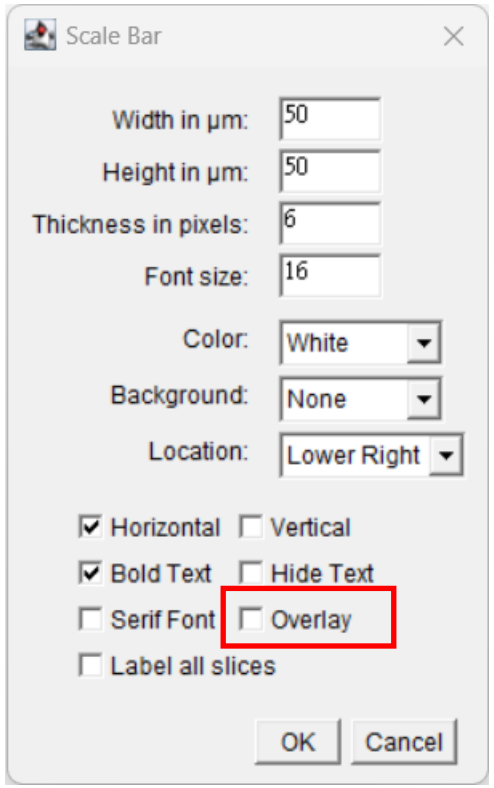


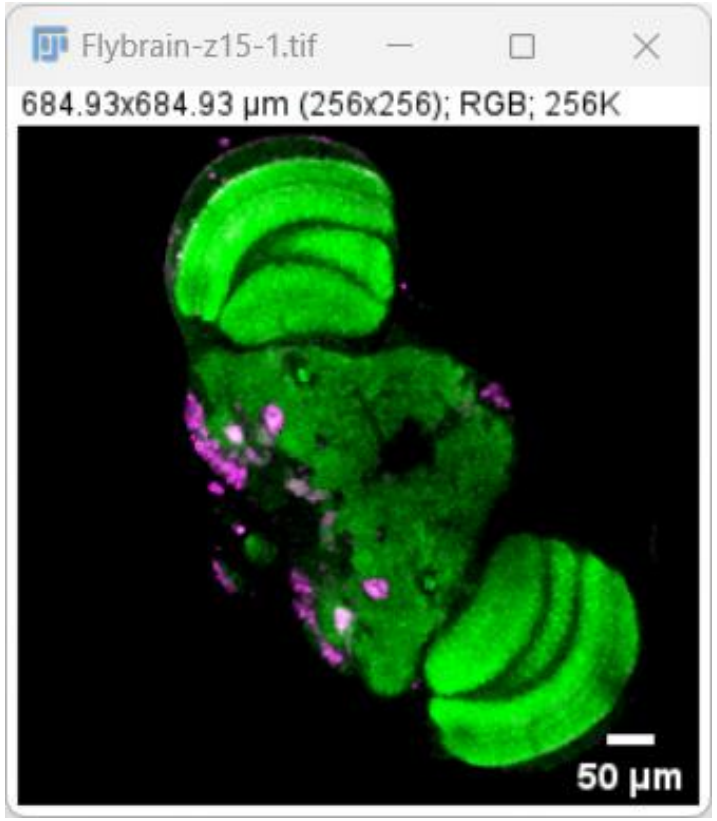
Convert your image data to RGB color for presentation

- Image -> Overlay -> Flatten
- Image -> Type -> RGB color
- SaveAs -> Png… (metadata loss)

Scale Bar

- Analyze -> Tools -> Scale Bar
- Confirm the pixel size first (“Ctrl + Shift + P”)
- Recommended to add a scale bar on the image files for presentation (RGB format)





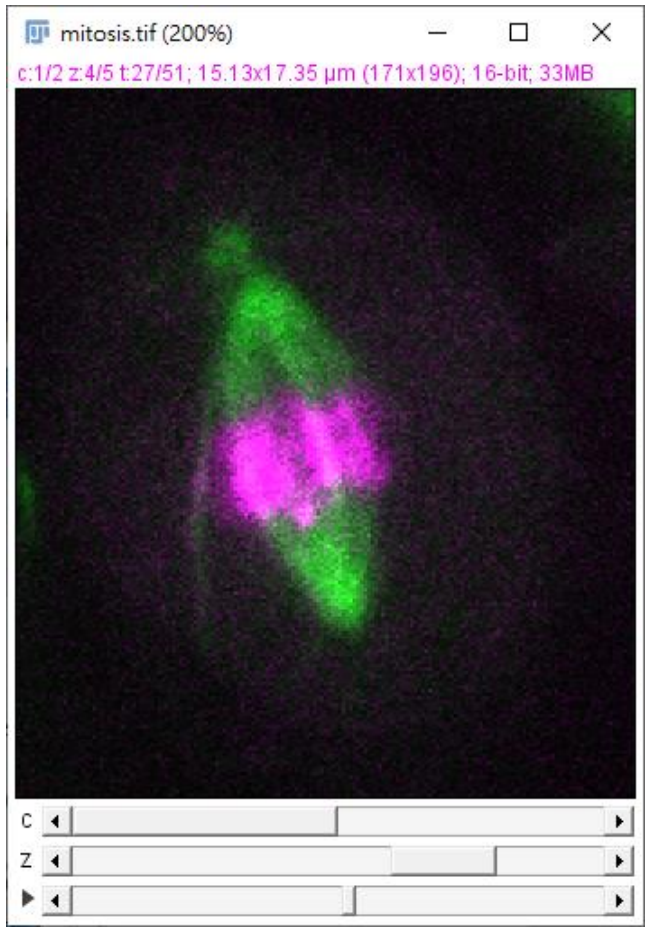
To allow other software can read the scale bar:

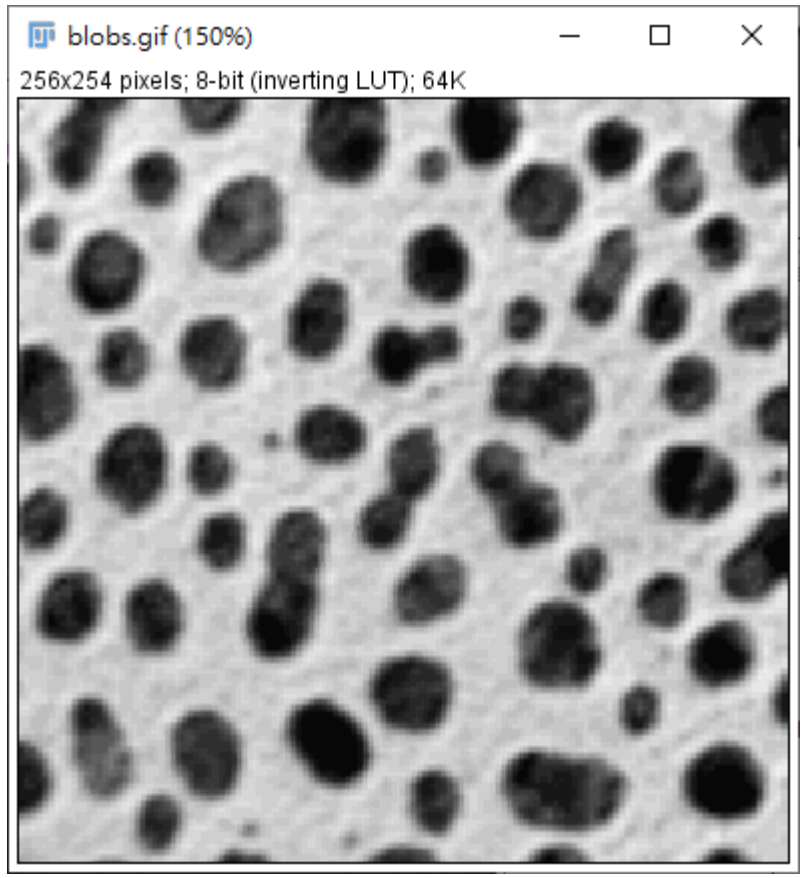
- Uncheck “Overlay”.
- If “Overlay” is checked -> Please “Flatten” it!

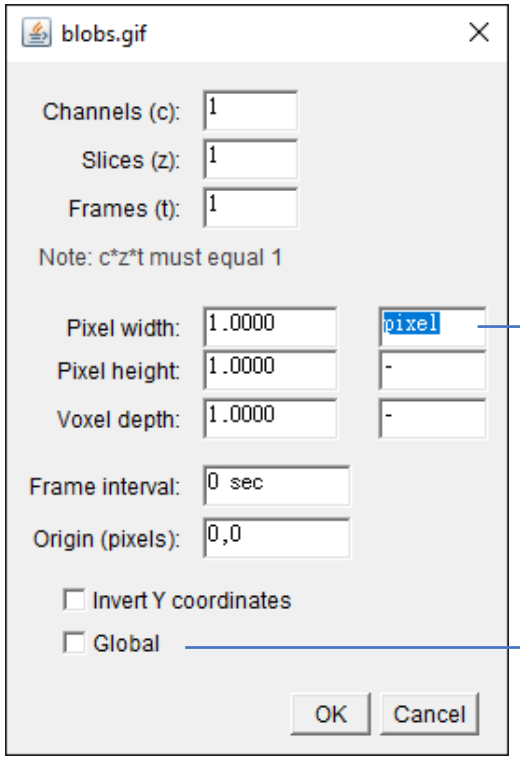
Image -> Overlay -> Flatten

Pixel size calibration

- Image -> Properties



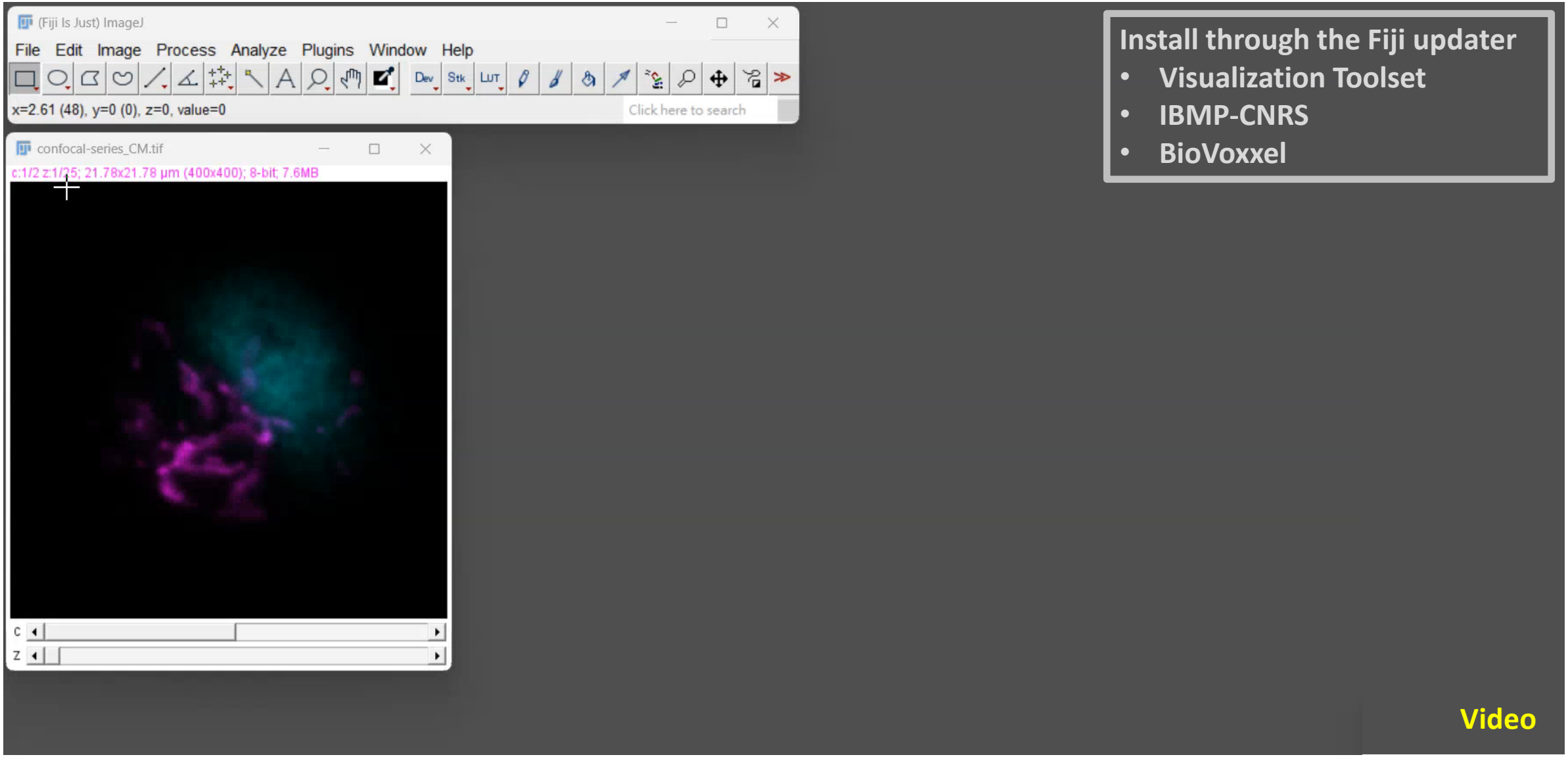




um
micron

Apply to
other image
windows

Visualization Toolset (super useful for presentation figure prepare)



https://github.com/kwolbachia/Visualization_toolset

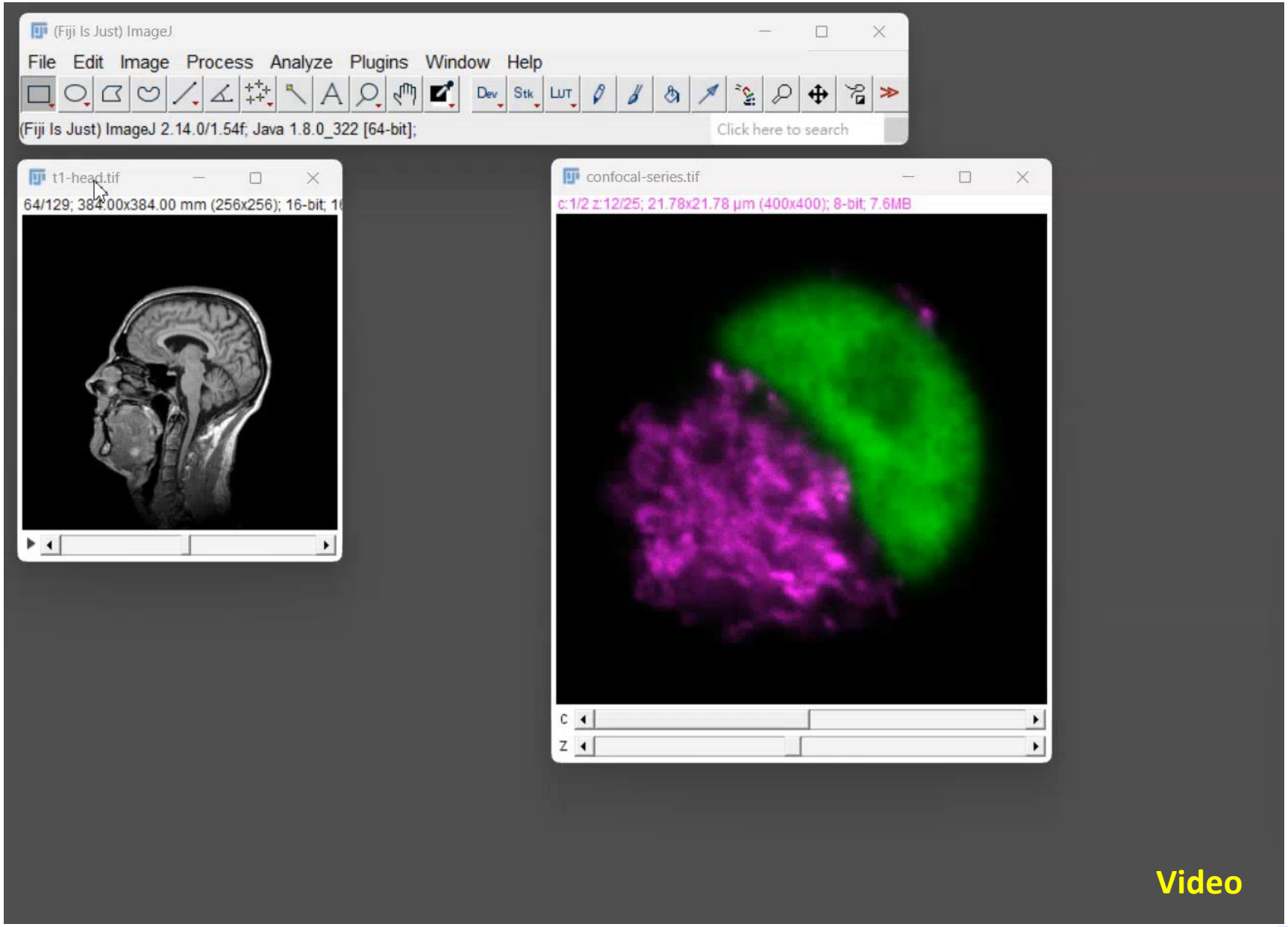
Summary

- **Always use the original image for analysis (especially for signal intensity)**
  - If the raw data isn't RGB, then don't convert it before analysis!
- **Create an RGB copy of your image for display**
  - Keep the RGB copy separate, so you always retain it *and* the raw image

Exception : Brainbow, Skinbow ……

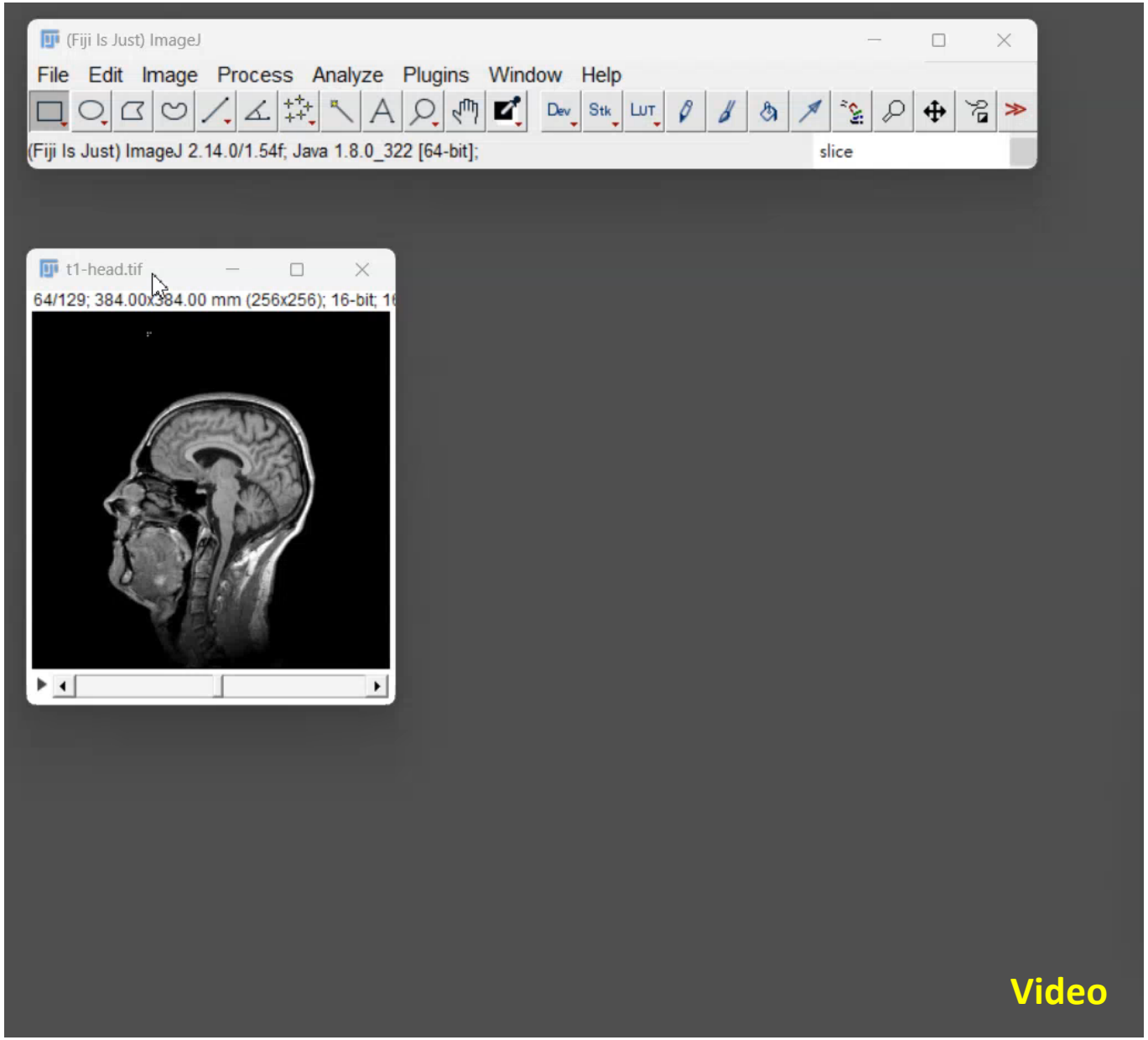
Orthogonal View

- Image -> Stacks -> orthogonal view
- Multiple Channel: XZ, YZ -> RGB Format

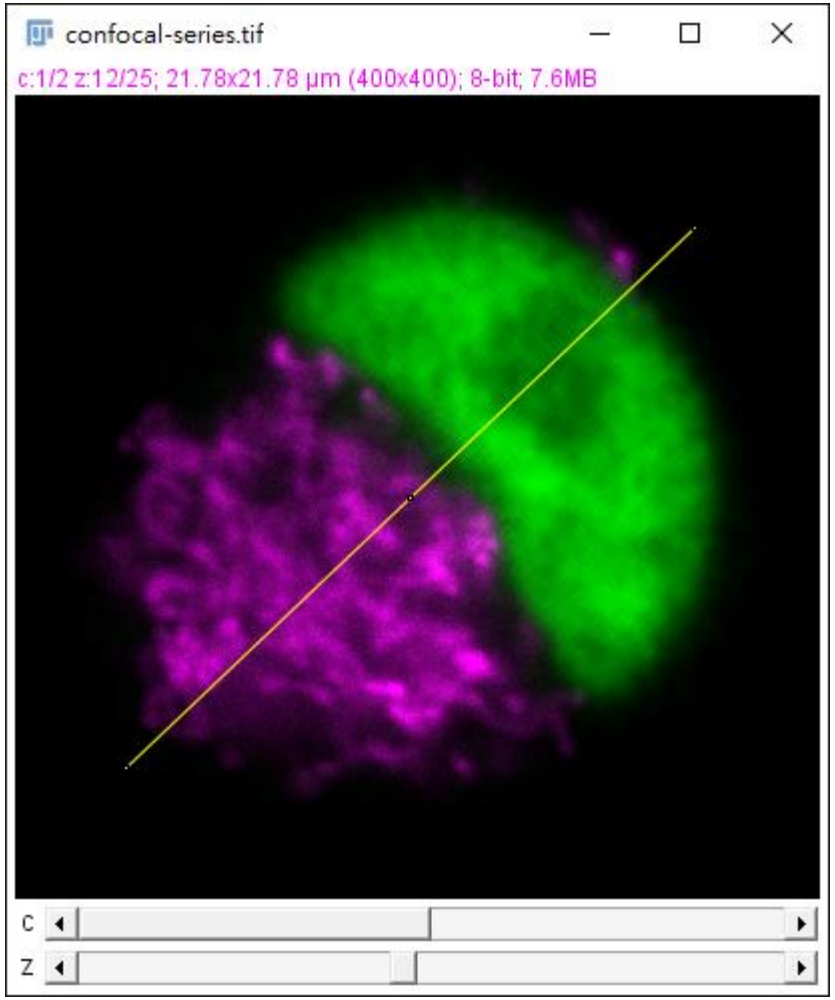


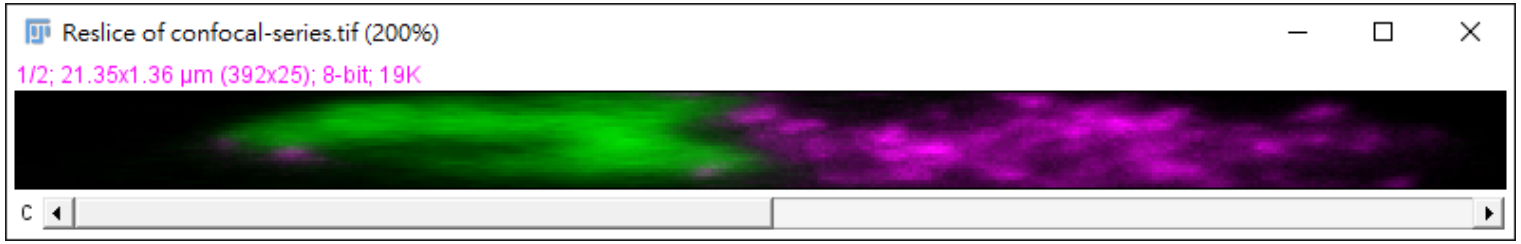
Dynamic slice and Reslice

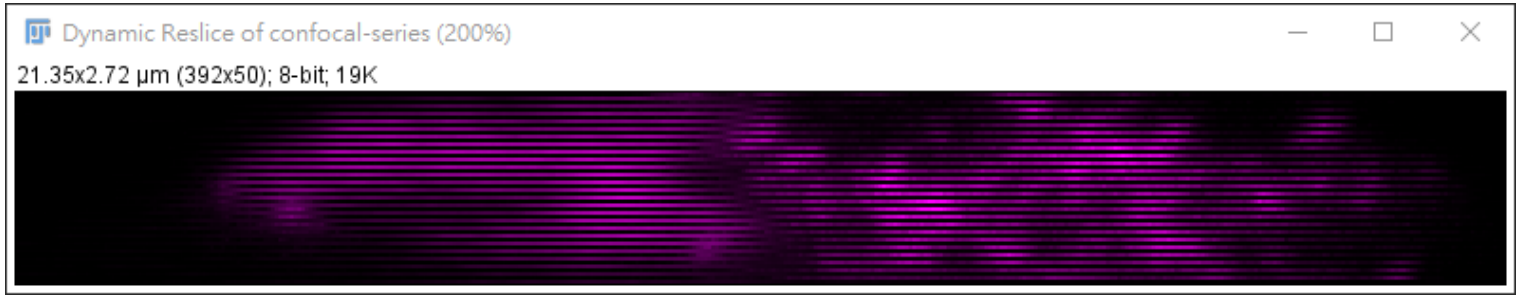
- Image -> Stacks -> Dynamic Slice
- Image -> Stacks -> Reslice [/]



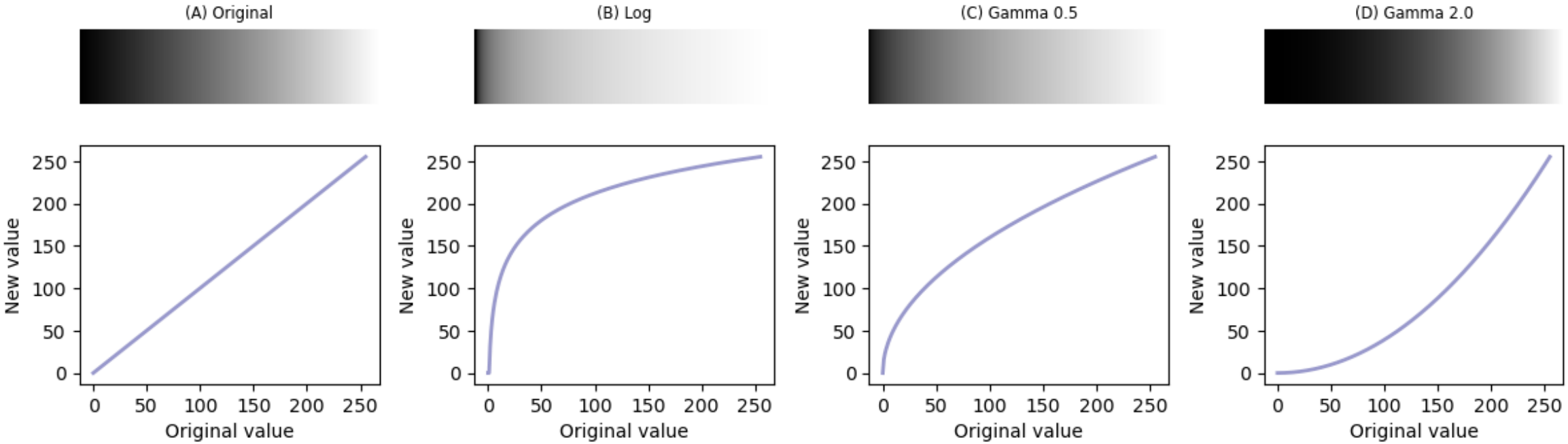
Reslice is more useful for multiple channel images







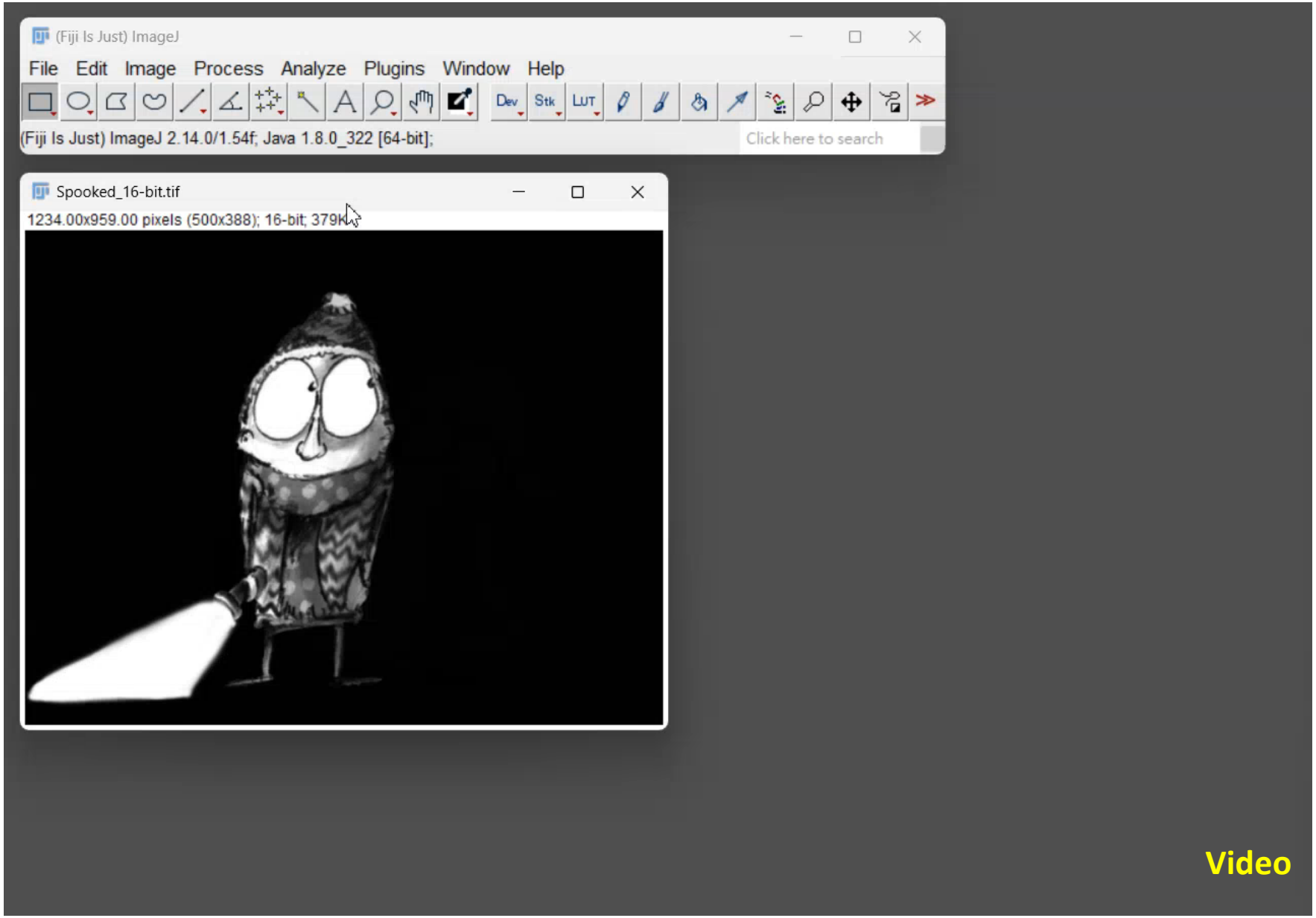
Gamma adjustment



- **In FIJI/ImageJ built-in function, Gamma adjustment change the pixel value! (in Zeiss Zen: change the display only!)**
- **If any nonlinear operations are used, these should always be noted in the figure legend!**

Gamma adjustment

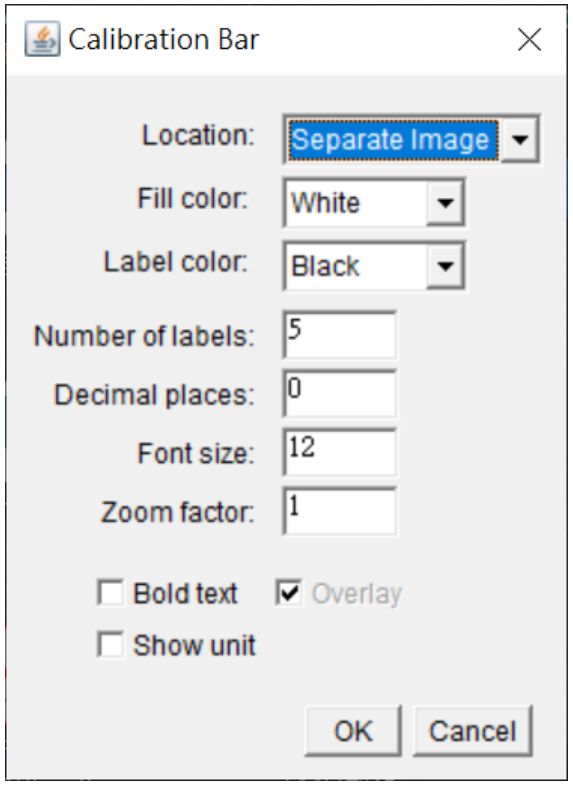
- Use Visualization Toolset
- Adjust gamma without changing pixel value
- Reset to linear LUT : Change to another LUT and then reselect the LUT you need!

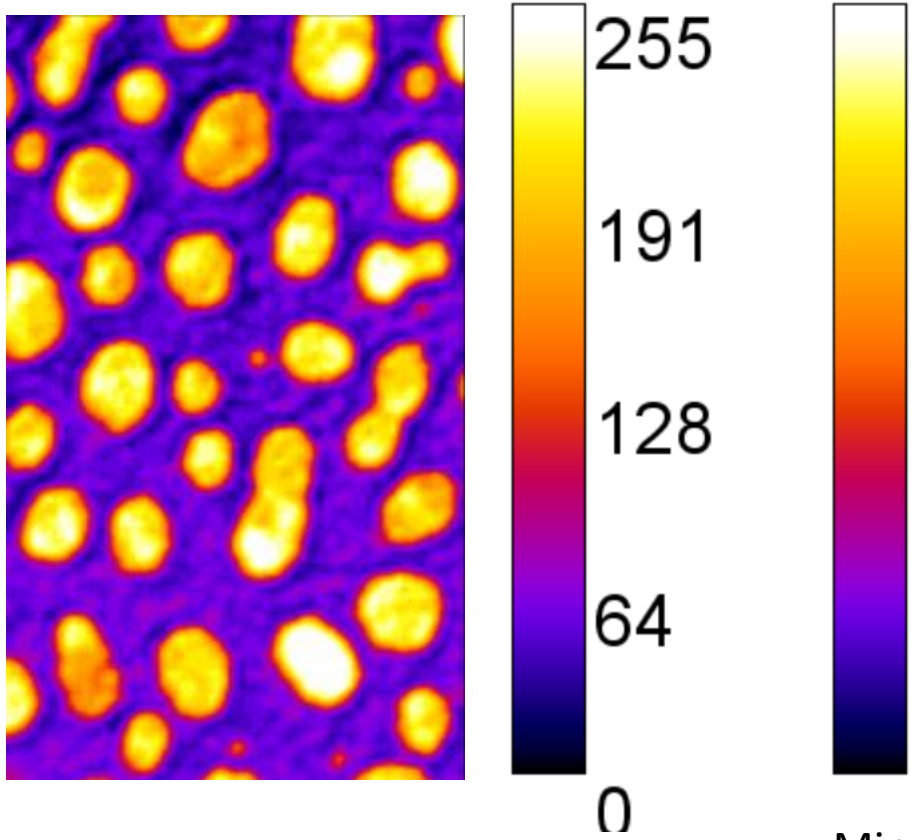


Example image from: https://bioimagebook.github.io/

Calibration Bar (Color scale)

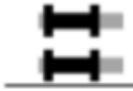
- Analysis -> Tools -> Calibration Bar
- Only work for single channel image
- Brightness and Contrast affect the Calibration





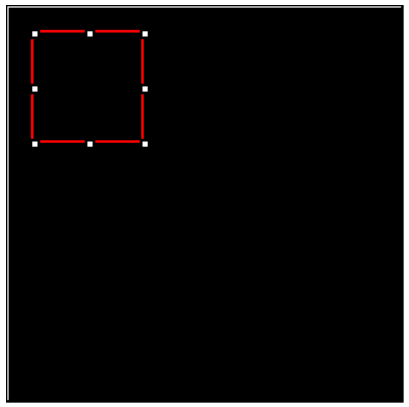
Community-developed checklists for publishing images and image analysis

Image colors and channels

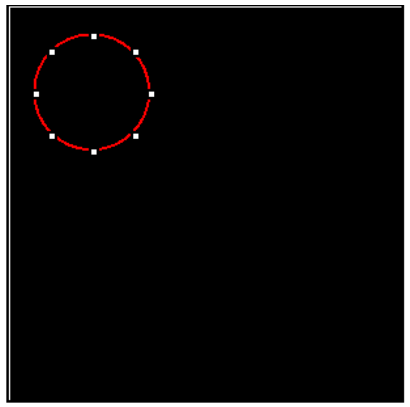
	Annotation of channels (staining, marker etc.) visible	Minimal
	Adjust brightness/contrast, report adjustments, use uniform color-scales	
	Image comparison: use same adjustments	
	Multi-color images: accessible to color blind	
	Channel color high visibility on background	
	Provide grey-scale for each color channel	Recommended
	Provide color scales for intensity values (greyscale, color, pseudo color…)	
	Pseudo-colored images: additionally provide greyscale version for comparison.	Ideal
	Gamma adjustments: additionally provide linear-adjusted image for comparison	

https://doi.org/10.48550/arXiv.2302.07005

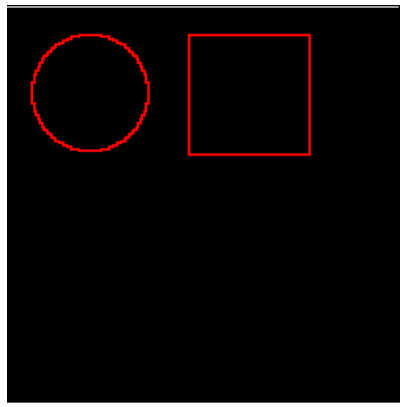
Regions of interest (ROI)



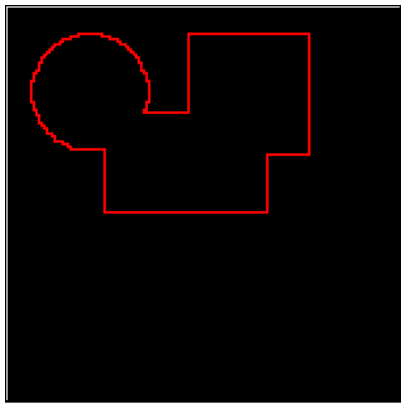
draw a rectangle
using the rectangle tool



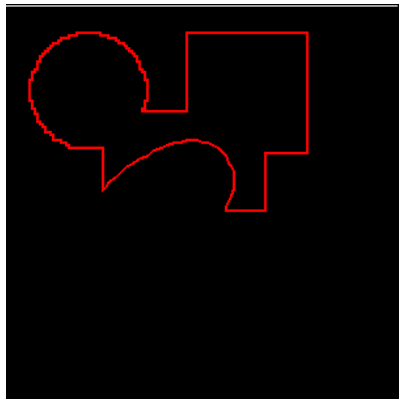
draw a circle
using the circle tool



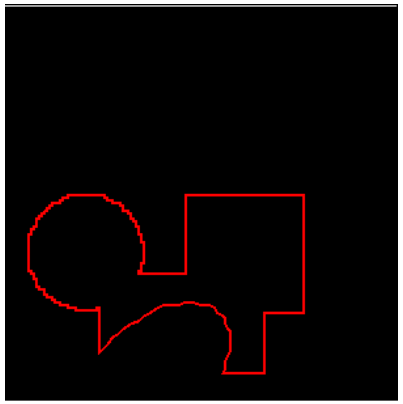
add objects to the ROI
by holding SHIFT



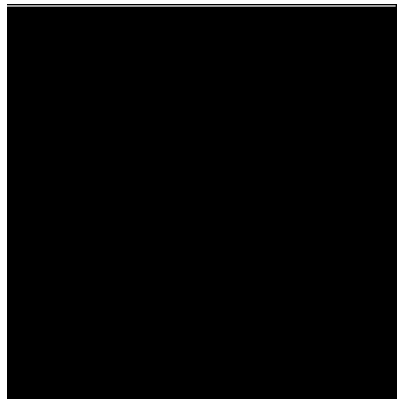
unite objects to the ROI
by holding SHIFT



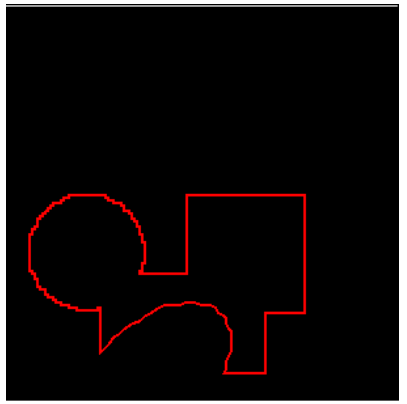
subtract objects to the
ROI by holding ALT



move the ROI
by clicking inside and
move the mouse



delete the ROI
by clicking outside
(Ctrl + Shift + A)



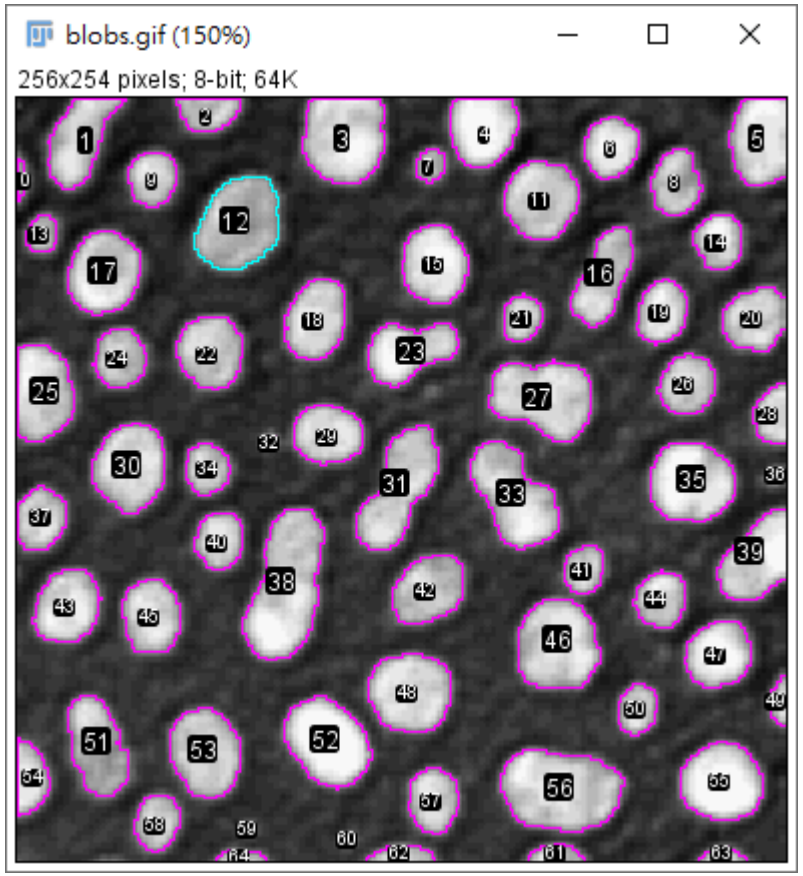
Recover the last ROI by using menu
Edit > Selection > Restore Selection
(Ctrl + Shift + E)

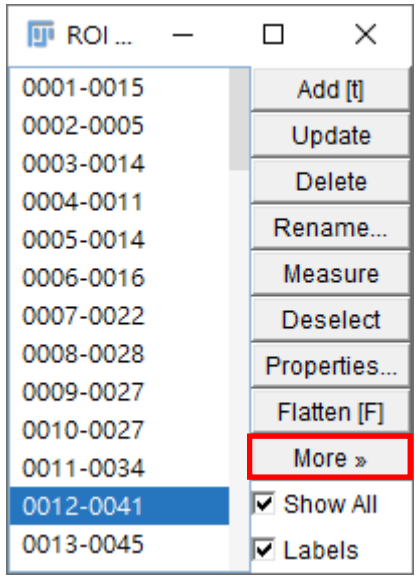
Adapted from the learning material of Dr. Robert Haase

ICOB Imaging Core

ROI manager

- Analysis -> Tools -> ROI Manager
- Save the ROI, so you can repeat the measurement!





- Single ROI: *.roi
- Multiple ROI: *.zip

- Deselect
  -> Not select a specific ROI
  -> “All”

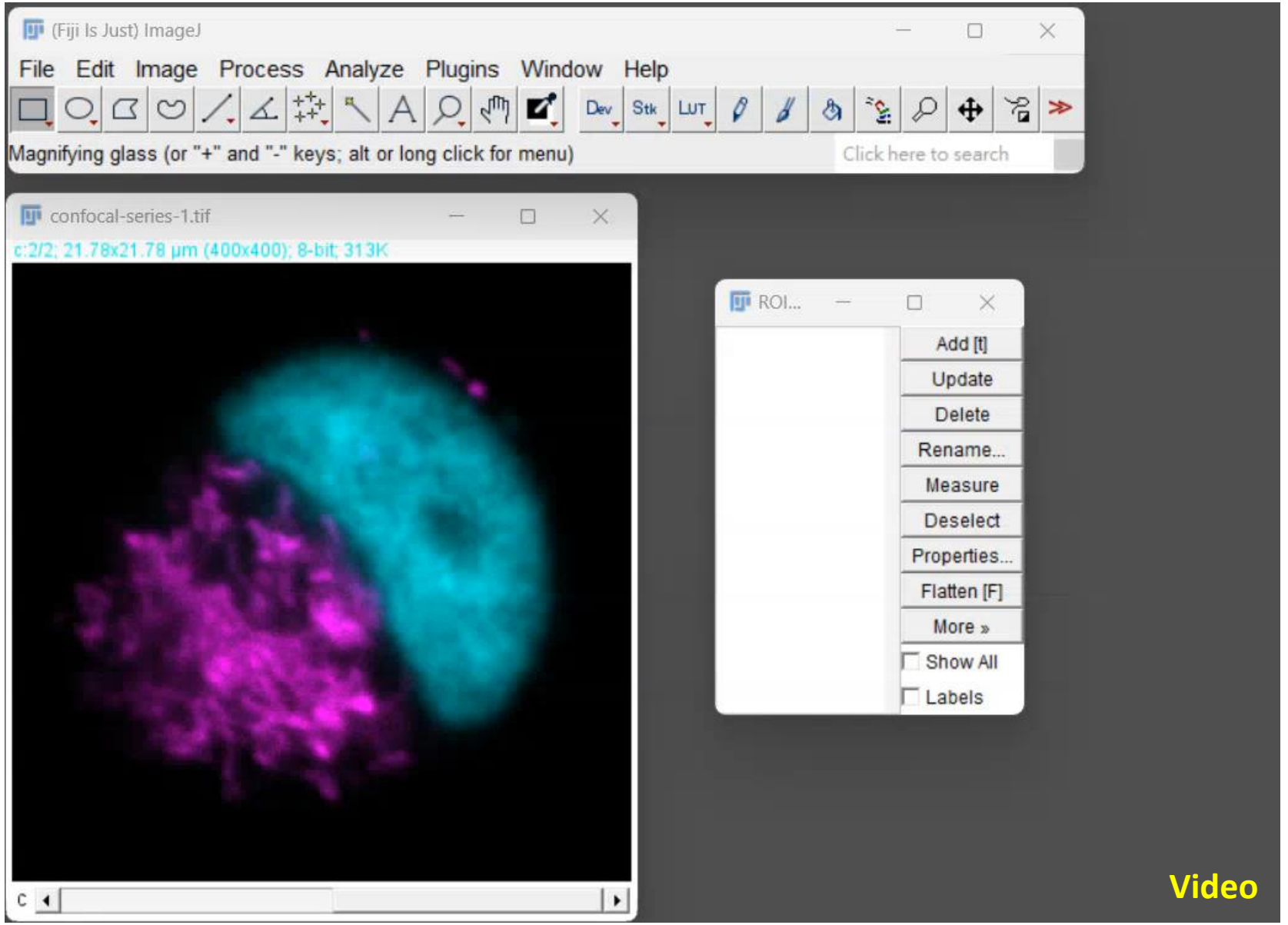
Useful Selection Tool

- **Selection Brush Tools**
- **Wand Tools**

Multiple ROI

- Add to ROI manager

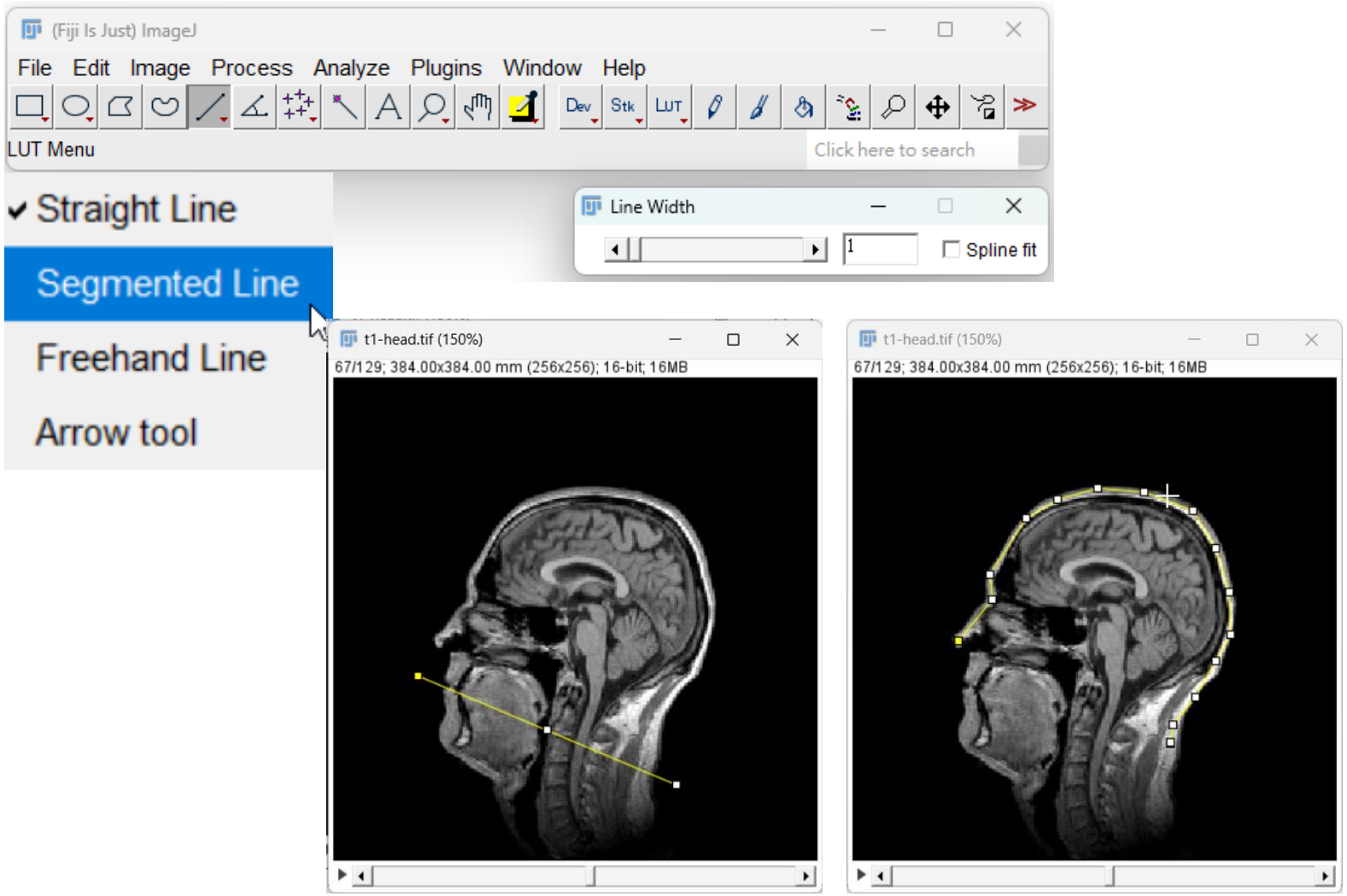
- Select None



Line tool

- Measure length
- Add to ROI manager(“T”)

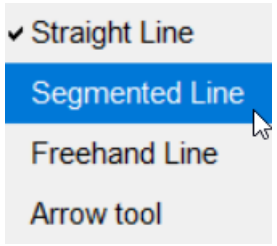
  Measure
- Plot profile



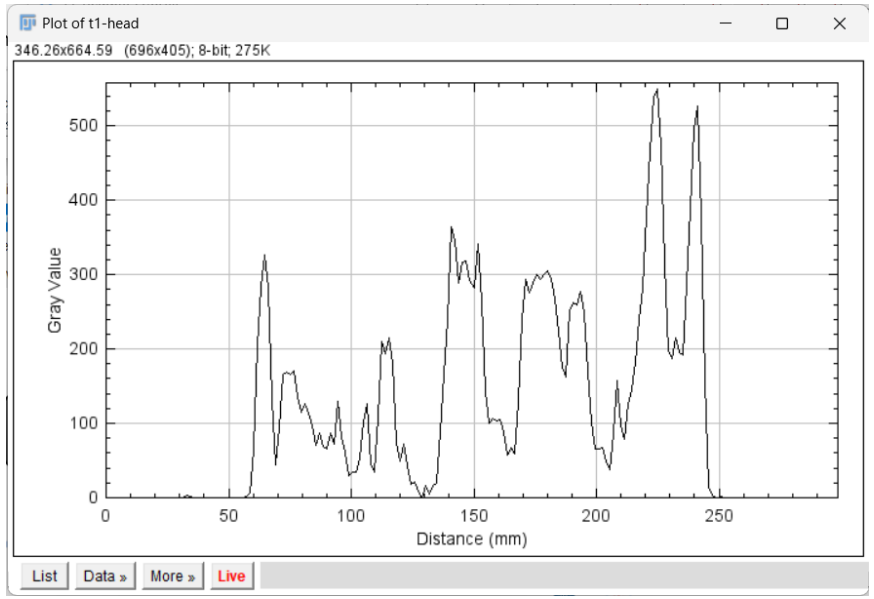
Plot Profile

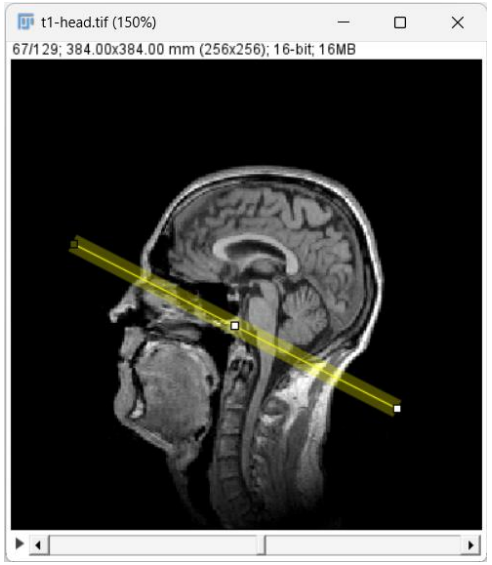
- Analyze -> Plot profile

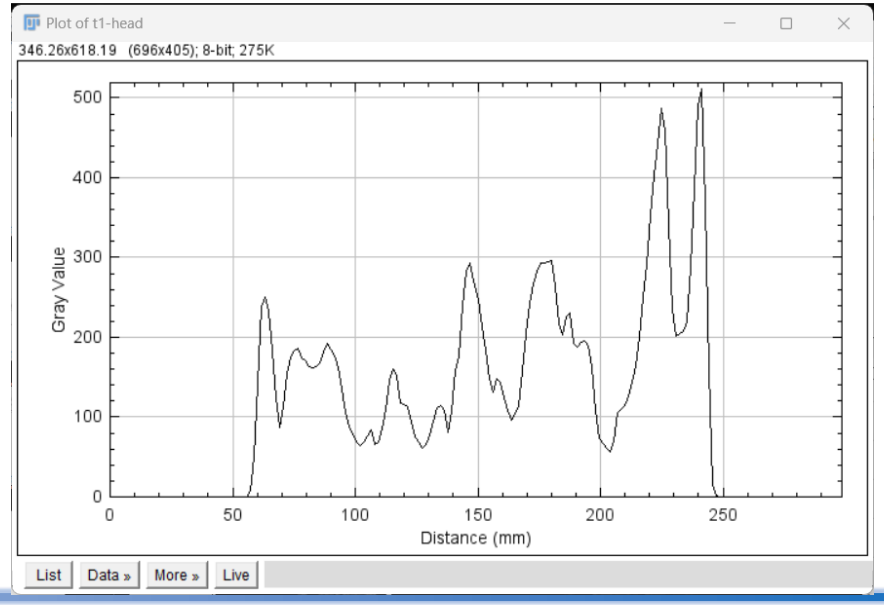
- Can apply to different Line type, width



- Note the scale information, unit!

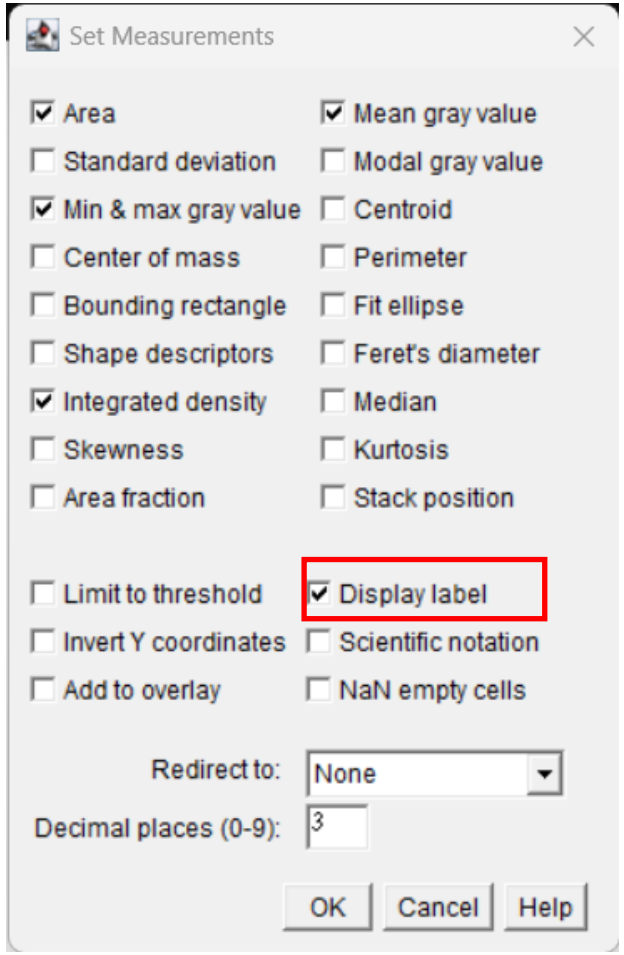




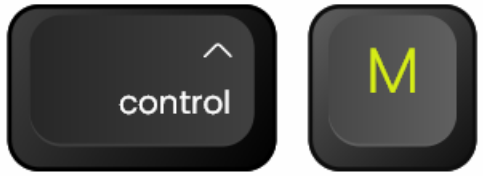


Measurement

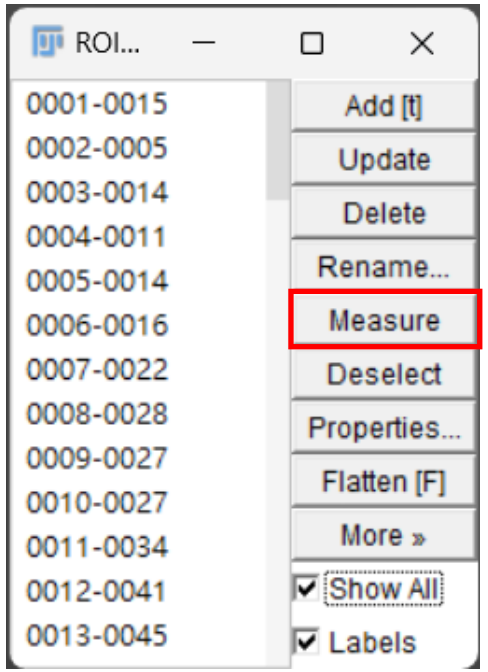
- Analyze > Set measurements

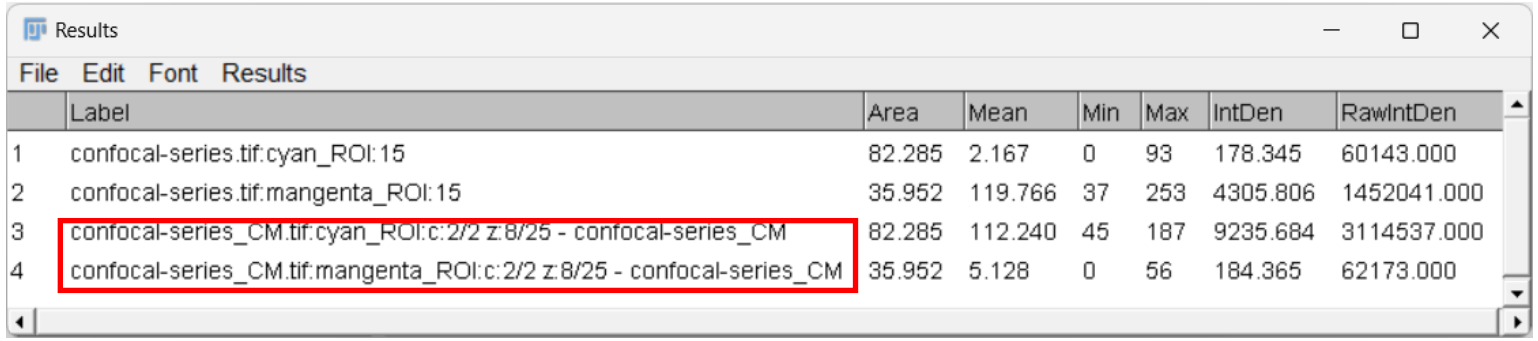


- Analyze > Measure



- ROI Manager > Measure
- Unit: Image -> Properties.. (Ctrl + Shift + P)





	Label	Area	Mean	Min	Max	IntDen	RawIntDen
1	confocal-series.tif:cyan_ROI:15	82.285	2.167	0	93	178.345	60143.000
2	confocal-series.tif:mangenta_ROI:15	35.952	119.766	37	253	4305.806	1452041.000
3	confocal-series_CM.tif:cyan_ROI:c:2/2 z:8/25 - confocal-series_CM	82.285	112.240	45	187	9235.684	3114537.000
4	confocal-series_CM.tif:mangenta_ROI:c:2/2 z:8/25 - confocal-series_CM	35.952	5.128	0	56	184.365	62173.000

- Use Bio-Format importer to open the image can reveal more details on the measurement label!

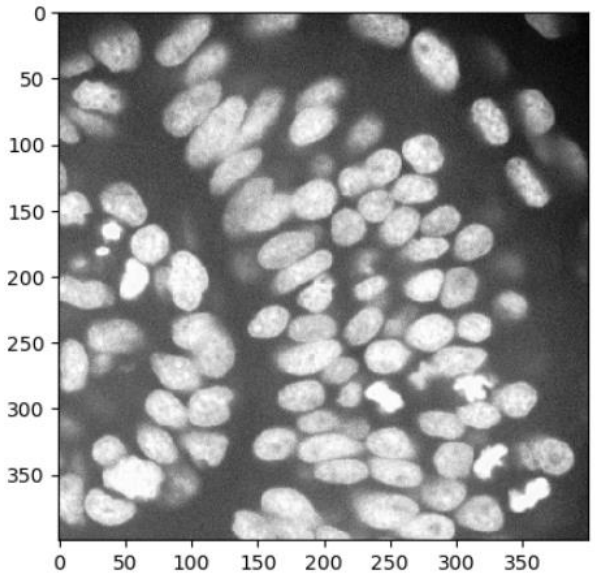
Part II: Bioimage Analysis Workflow

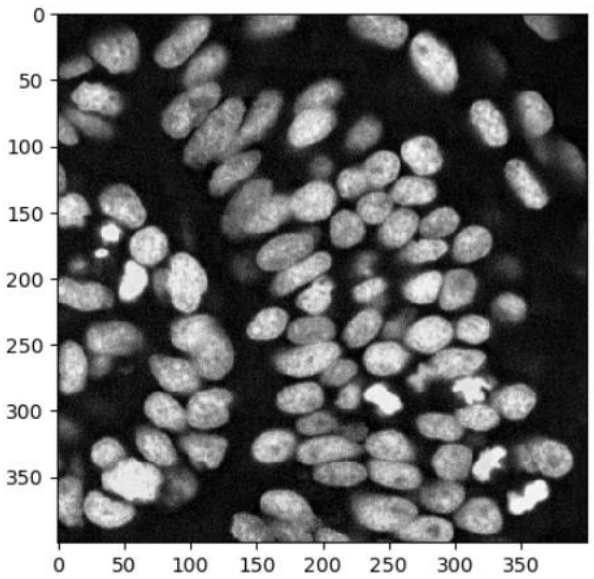
Part II: Typical bioimage analysis workflow

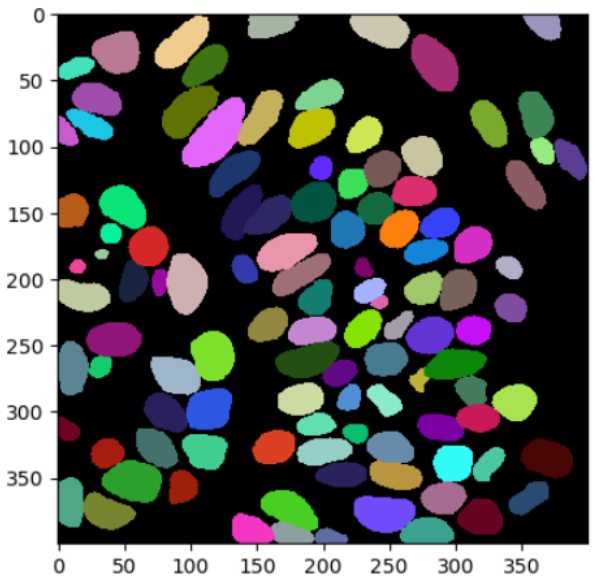
- Segmentation
- Selection, ROI List, Mask, Label
- ImageJ Macro
- CLIJ (GPU-accelerated image processing)
- Interactive bioimage analysis and automatic code generation
- Future of bioimage analysis (AI)

Bio-image Analysis

- Image Data Analysis workflows
- Goal: **Quantify observations, substantiate conclusions with numbers**







	area	mean_intensity	major_axis_length	minor_axis_length
0	594.0	40572.809764	28.611591	26.537947
1	645.0	43764.872868	33.511511	24.566916
2	1105.0	51970.561991	45.232031	31.456308
3	718.0	47015.487465	31.023274	29.520883
4	791.0	49132.515803	36.382253	27.718301
...	...	...	...	...
105	238.0	30477.126050	20.252197	15.276536
106	615.0	32886.154472	41.030760	19.874280
107	110.0	33042.445455	14.366347	9.945911
108	222.0	43304.180180	25.370549	11.637599
109	167.0	43378.808383	17.895110	13.015369

Image filtering → Image segmentation → Feature extraction

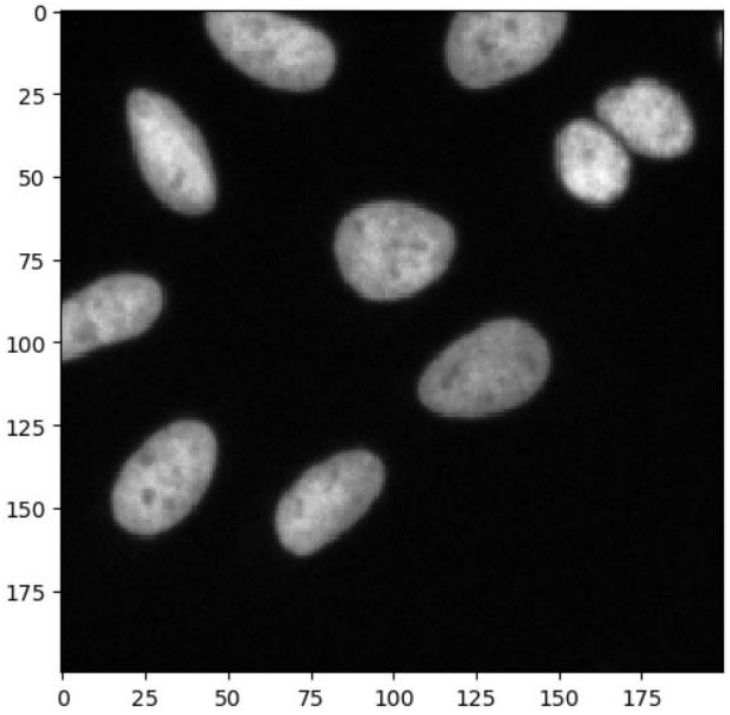
Image source: Mauricio Rocha Martins (Norden/Myers lab, MPI CBG)

Adapted from the learning material of Dr. Robert Haase

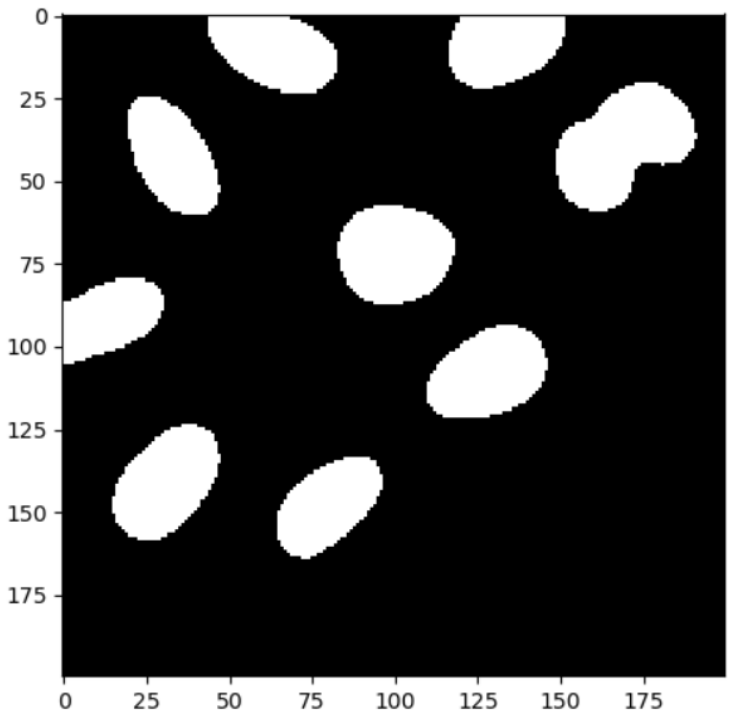
ICOB Imaging Core

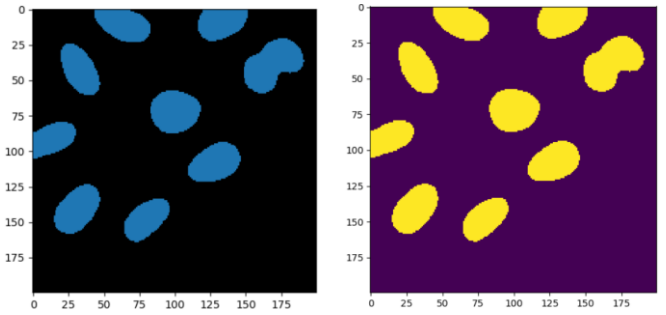
Terminology

Intensity image

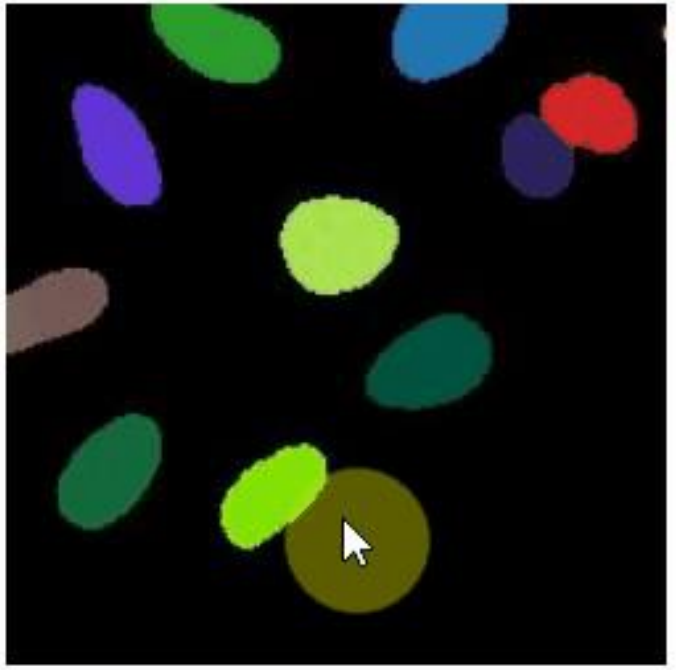


Binary image (Mask)





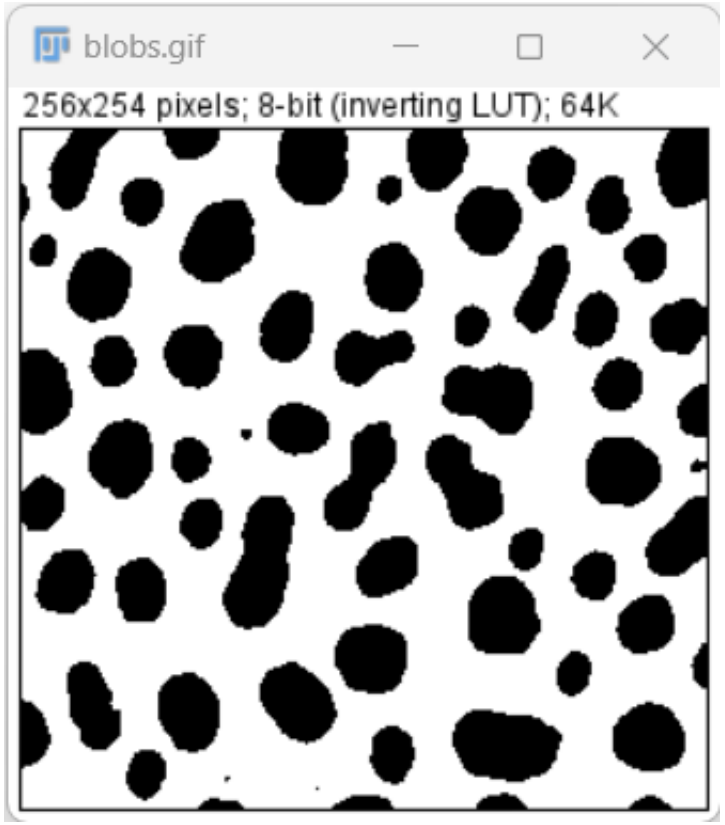
Label image



[y=152, x=92] = 0

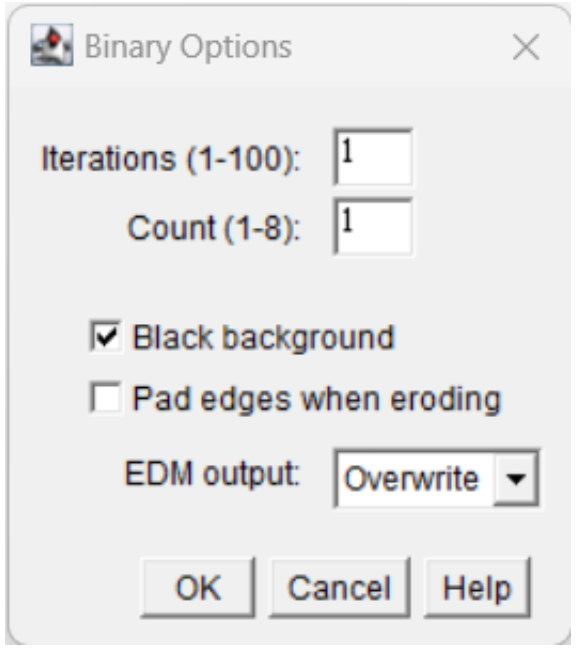
Mask: different display mode

“Default”



0	255

- **Process -> Binary -> Options…**

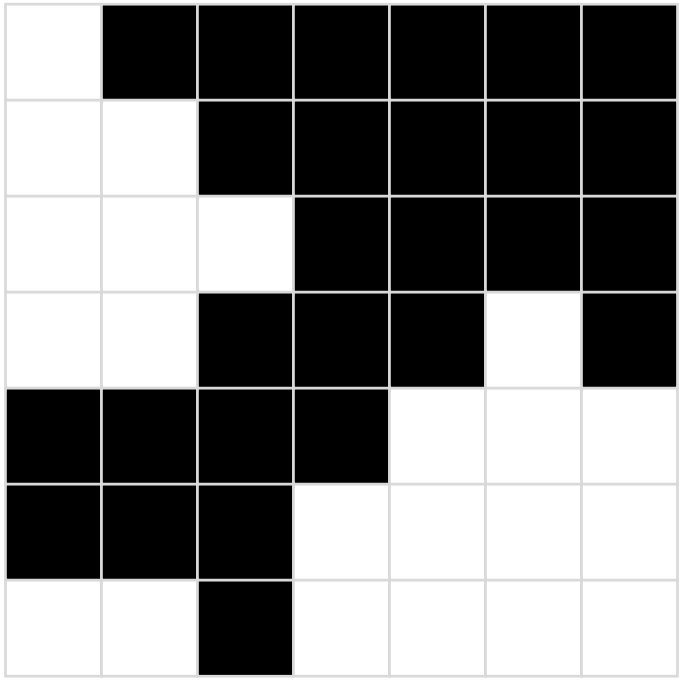


“Black background”

0	255

Connected component labelling

- In order to allow the computer differentiating objects, connected component analysis (CCA) is used to mark pixels belonging to different objects with different numbers
- Background pixels are marked with 0.
- The maximum intensity of a labelled map corresponds to the number of objects.



CCA

1	0	0	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0
1	1	0	0	0	3	0
0	0	0	0	3	3	3
0	0	0	3	3	3	3
2	2	0	3	3	3	3

4-connectivity vs 8-connectivity

4-connectivity

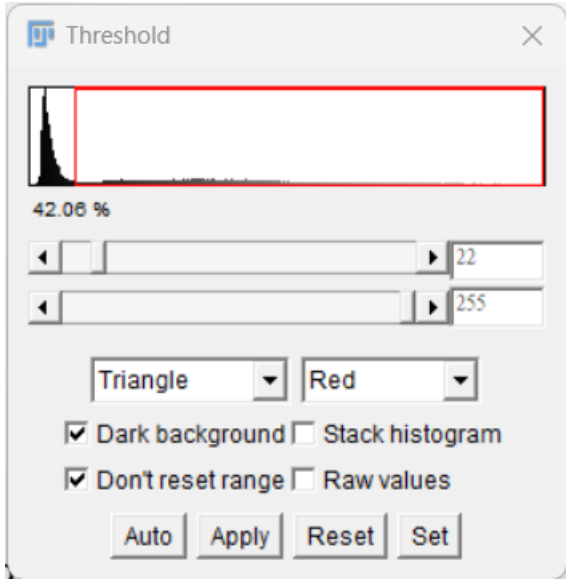
8-connectivity

- Analyze Particle in FIJI/ ImageJ: 8-connectivity
- 6-connectivity is similar to 4-connectivity, but in 3D.
- 26-connectivity is similar to 8-connectivity, but in 3D

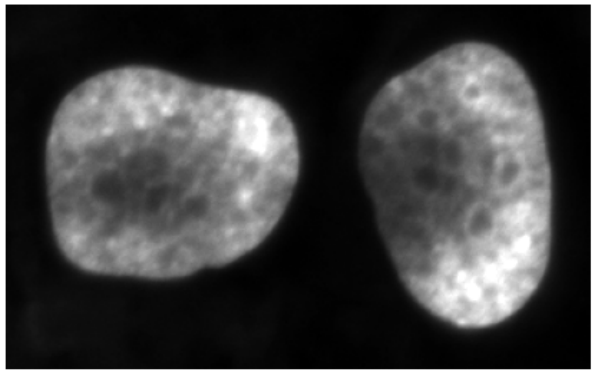
Thresholding

- Image -> Adjust -> Threshold…

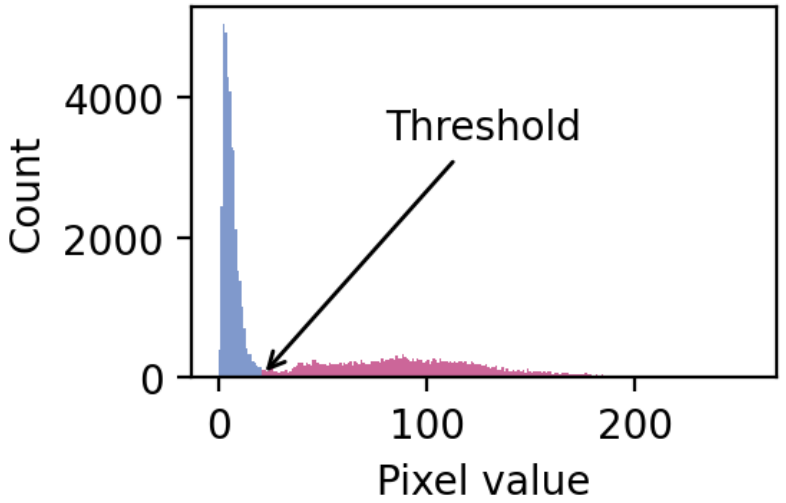
- Find an algorithm that fit to your data set!



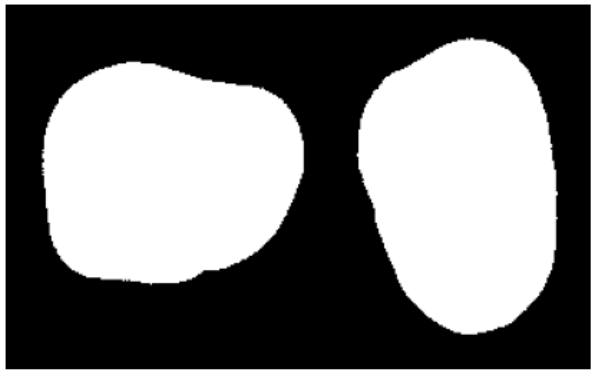
(A) Image



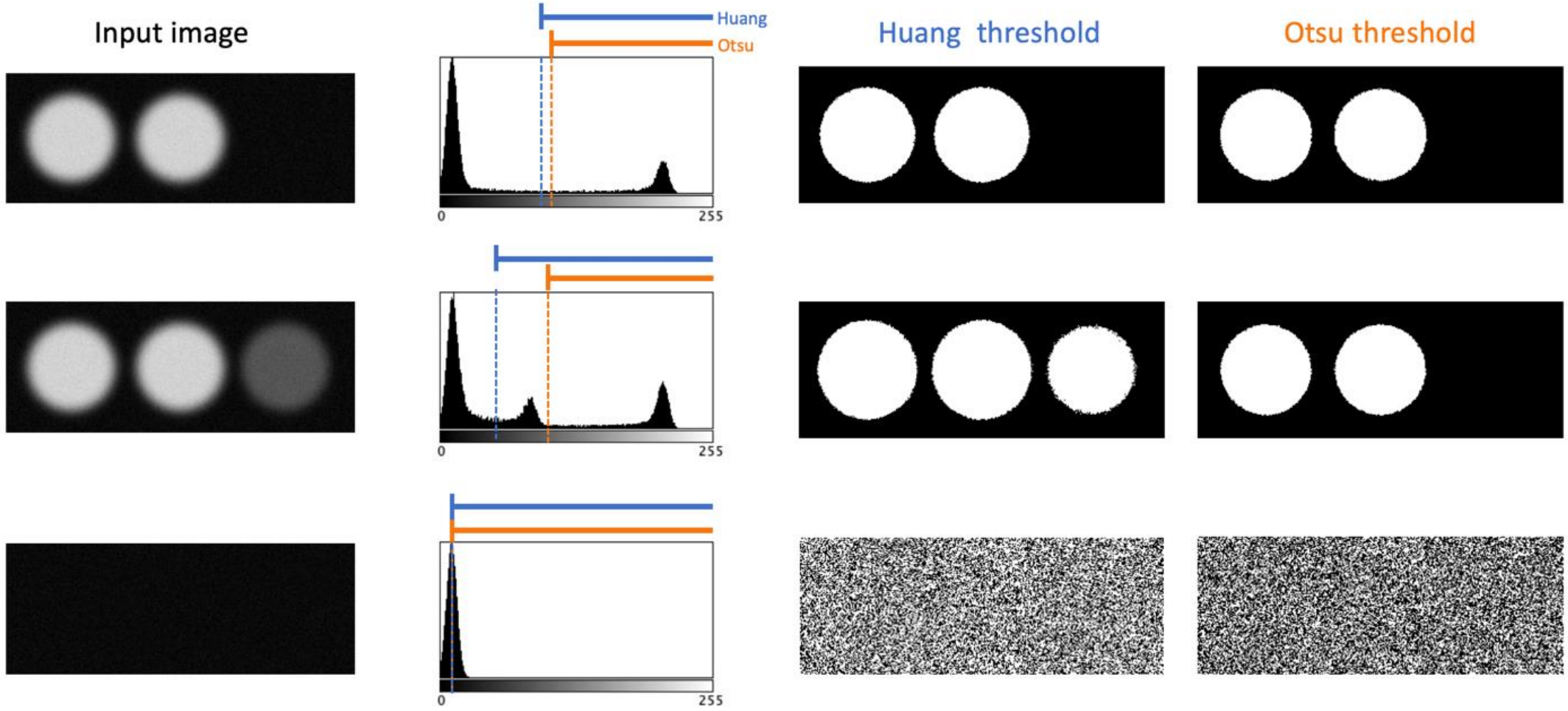
(B) Histogram



(C) Thresholded



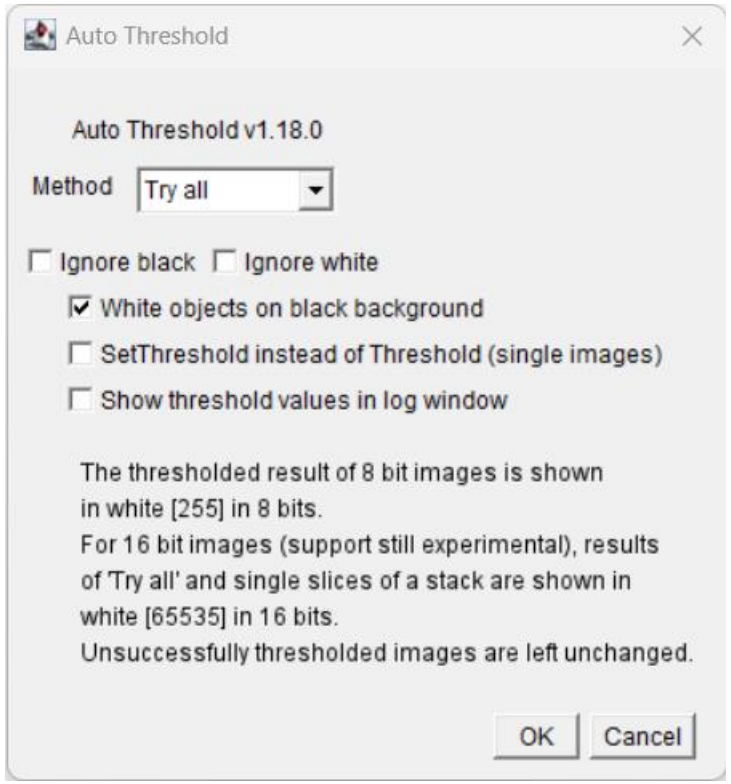
Automatic thresholding based on histogram

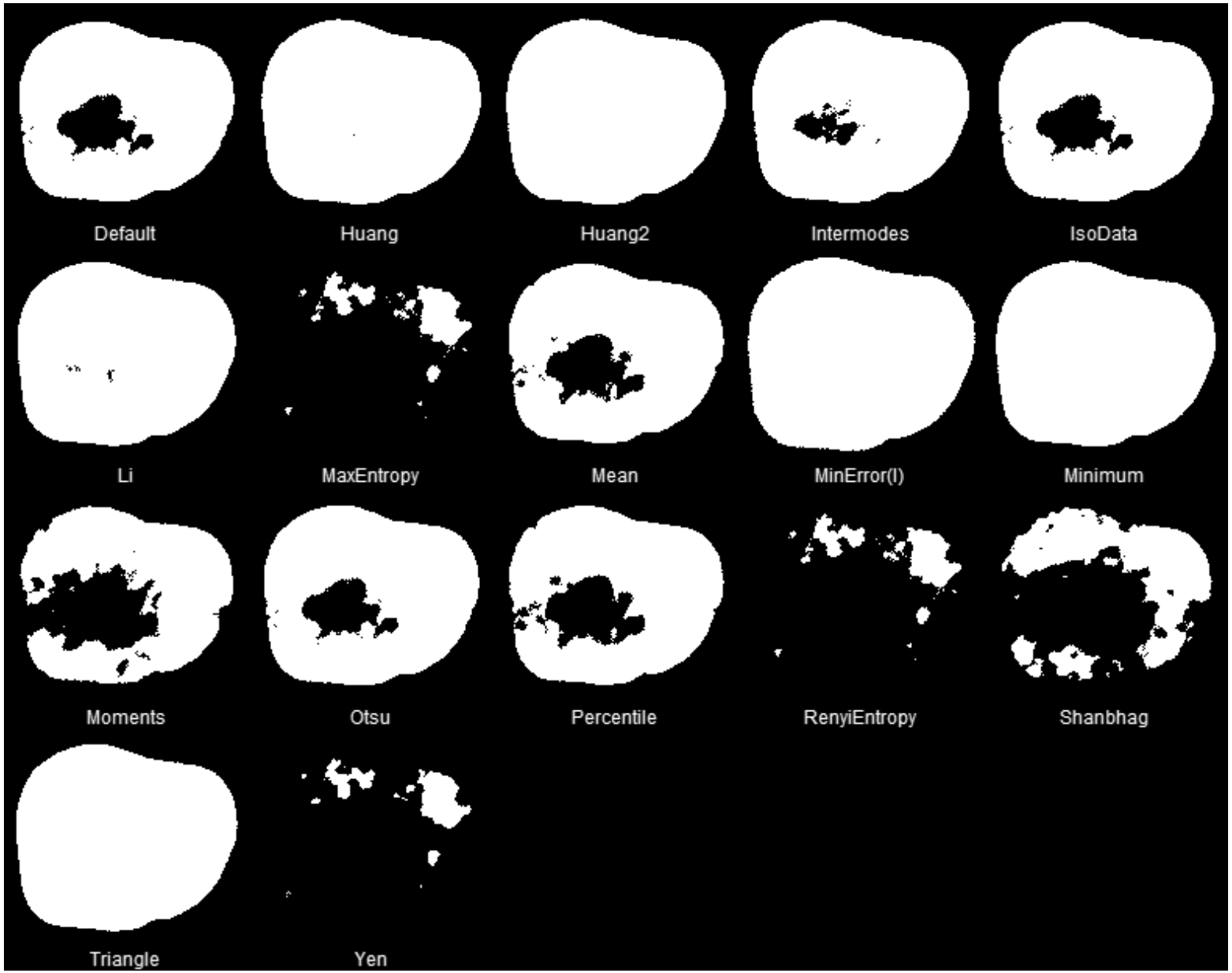


https://neubias.github.io/training-resources/auto_threshold/index.html

Try all

- Image -> Adjust -> Auto Threshold
  Method: Try all





Thresholding: Citing

- Cite the thresholding method of your choice properly

"We segmented the cell nuclei in the images using Otsu's thresholding method (Otsu et al. 1979) implemented in Fiji (Schindelin et al. 2012)."

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. SMC-9, NO. 1, JANUARY 1979

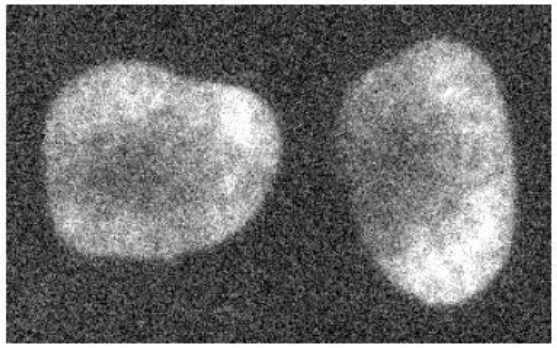
A Threshold Selection Method from Gray-Level Histograms

NOBUYUKI OTSU

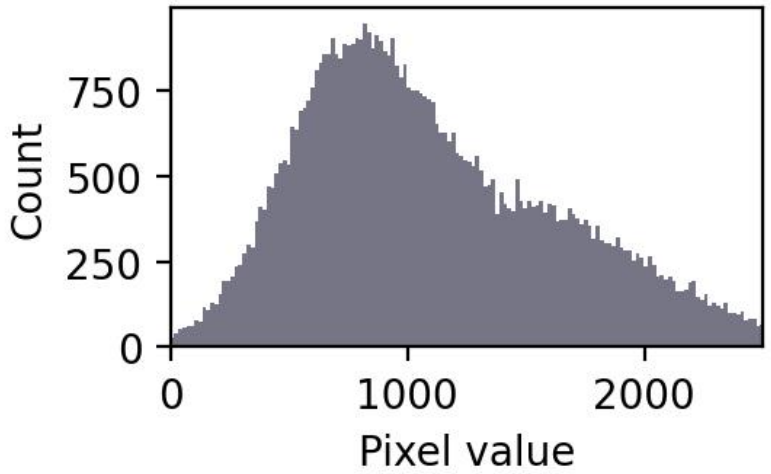
Abstract—A nonparametric and unsupervised method of automatic threshold selection for picture segmentation is presented. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth- and the first-order cumulative moments of the gray-level histogram. It is straightforward to extend the method to multithreshold problems. Several experimental results are also presented to support the validity of the method.

Filtering for improving thresholding results

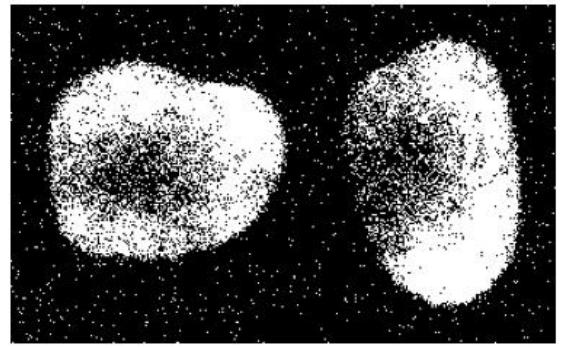
(A) Noisy image



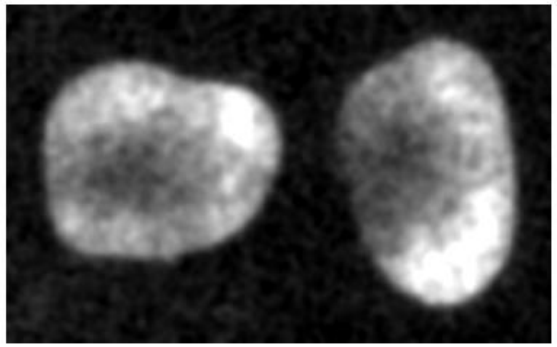
(B) Histogram of (A)



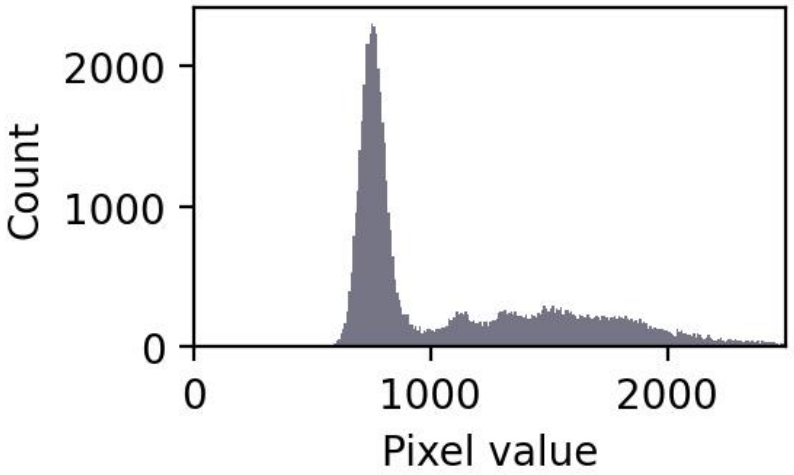
(C) Threshold applied to (A)



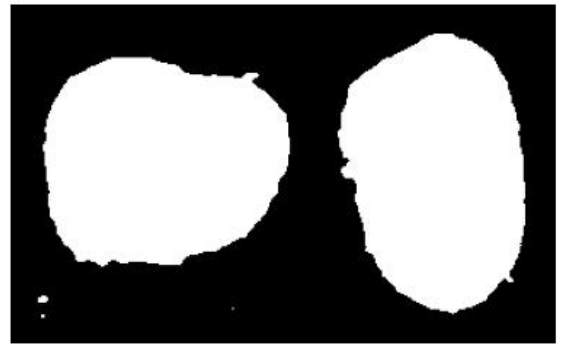
(D) Gaussian filtered image



(E) Histogram of (D)

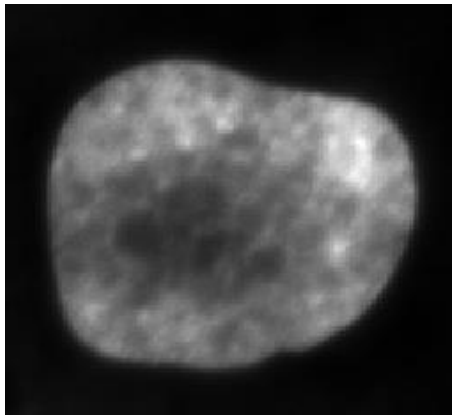


(F) Threshold applied to (D)

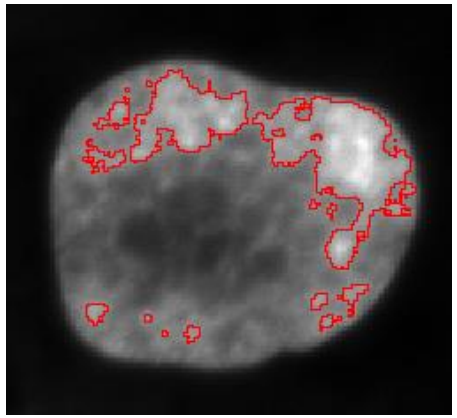


Filtering for improving thresholding results

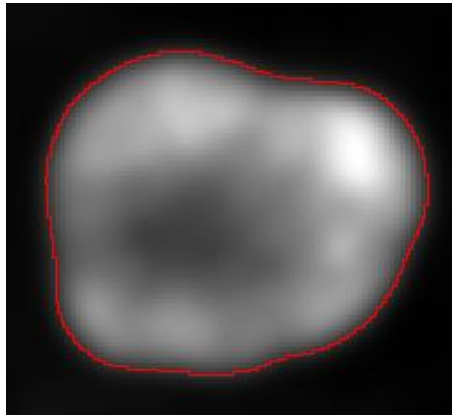
- In case thresholding algorithms outline the wrong structure, blurring in advance may help.
- However: **Do not** continue processing the blurred image, continue with the original!



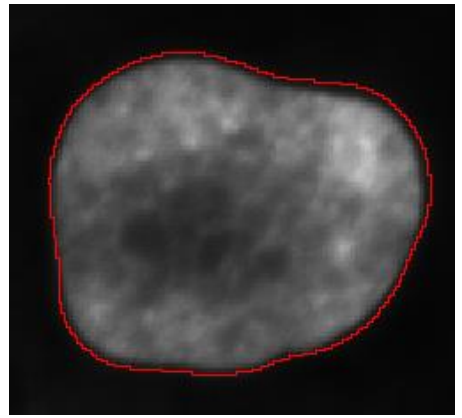
Thresholding ☹



Blurring +
Thresholding ☺



(duplicated image)



Transfer ROI to intensity image for measurement

Noise removal

- Gaussian filter
- Median filter (computationally expensive)

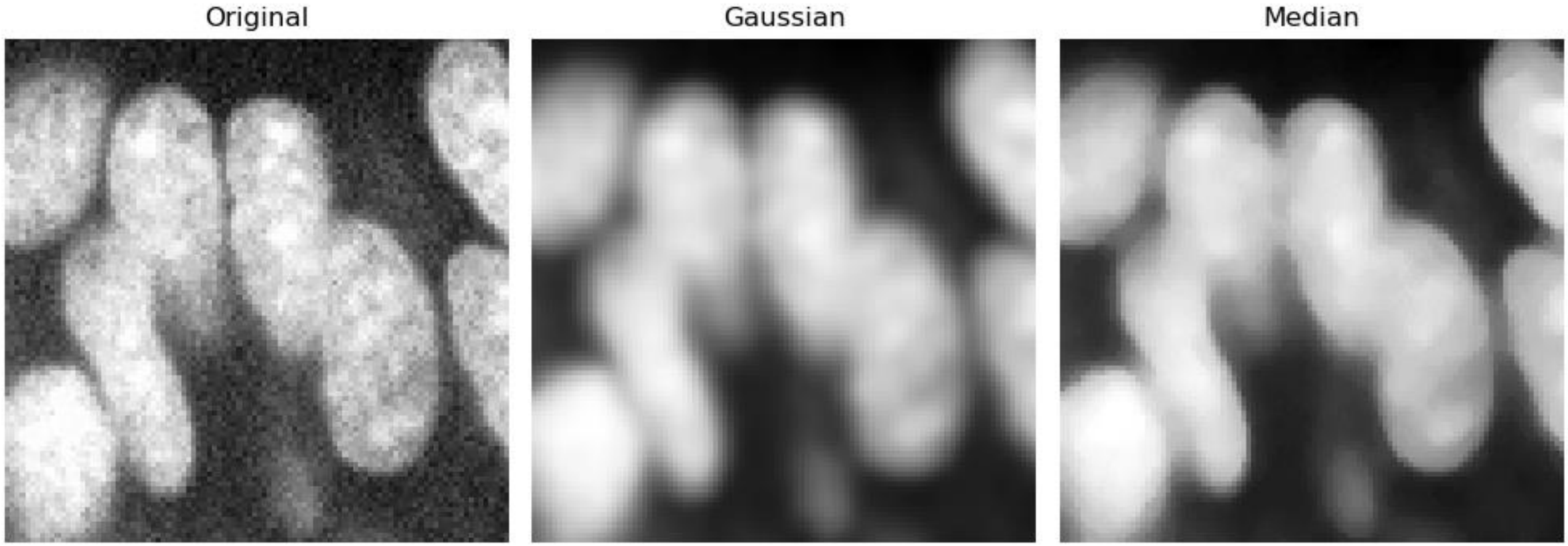
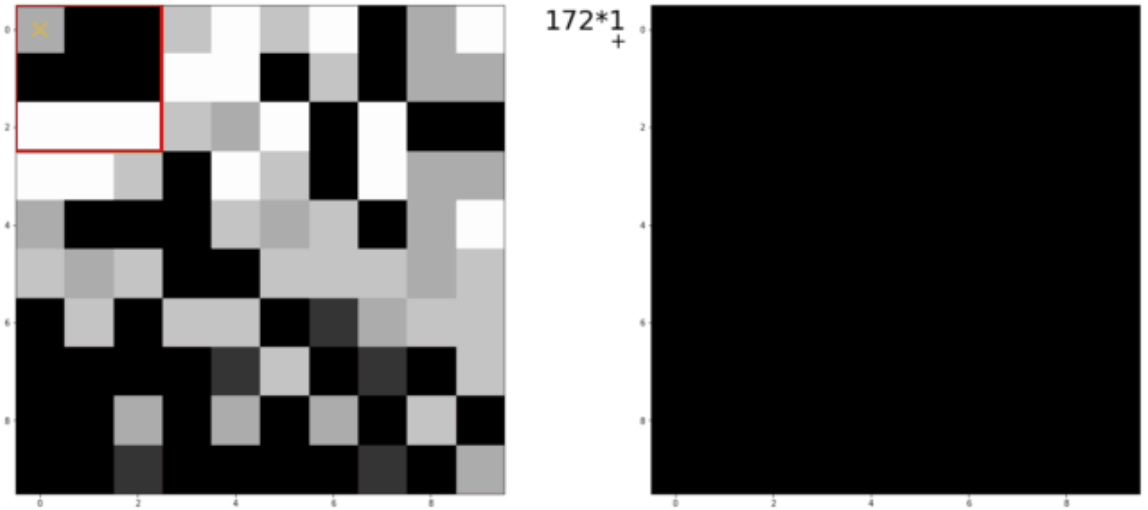


Image source: Mauricio Rocha Martins (Norden/Myers lab, MPI CBG)

Linear Filters

- *Linear filters* replace each pixel value with a weighted linear combination of surrounding pixels
- Filter *kernels* are matrices describing a linear filter
- This multiplication of surrounding pixels according to a matrix is called *convolution*



Mean filter, 3x3 kernel

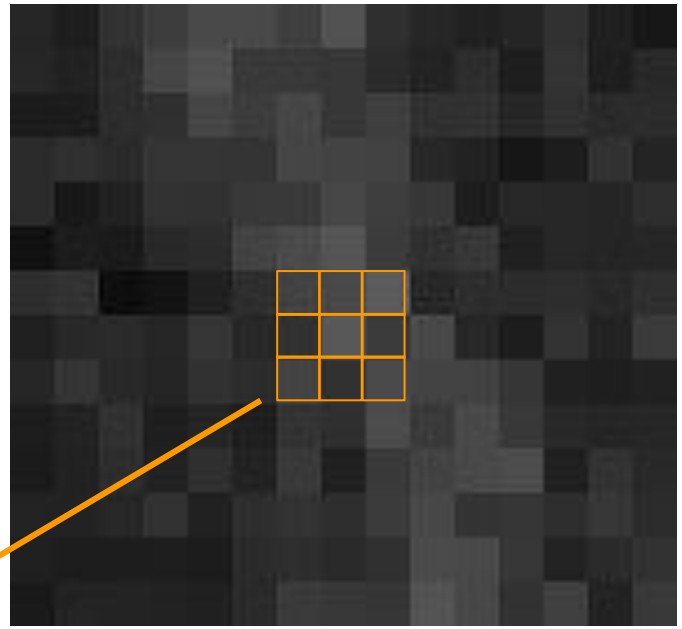
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

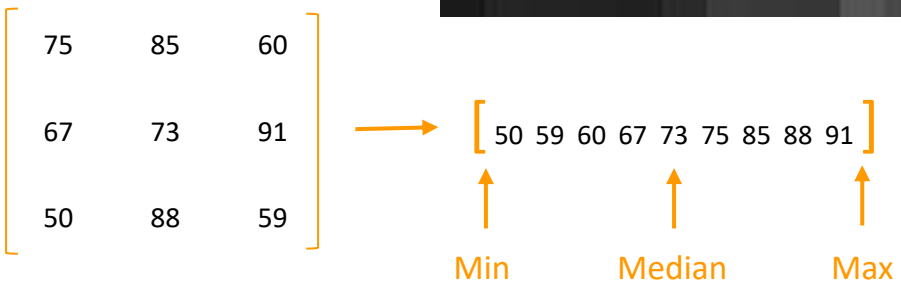
Animation source: Dominic Waithe, Oxford University
https://github.com/dwaithe/generalMacros/tree/master/convolution_ani

Nonlinear Filters

- Non linear filters also replace pixel value inside as rolling window but using a non-linear function.

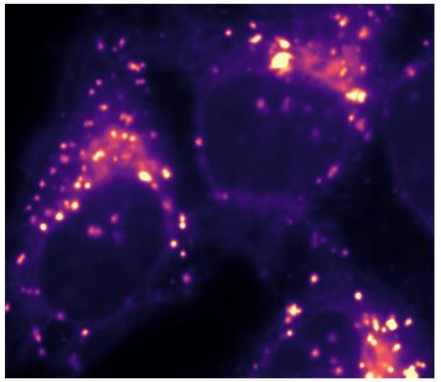
- Examples: order statistics filters
  - Min
  - Median
  - Max
  - Variance
  - Standard deviation



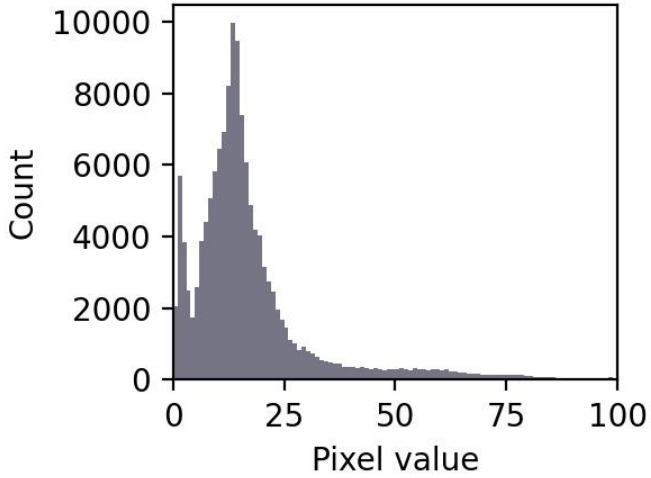


Background subtraction improving thresholding results

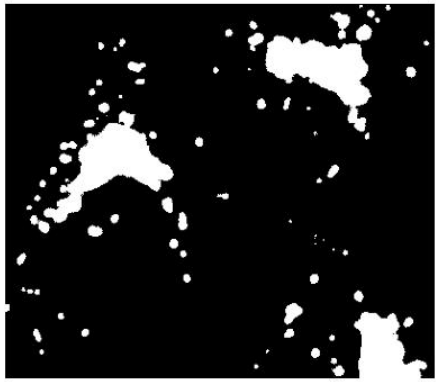
(A) Original image



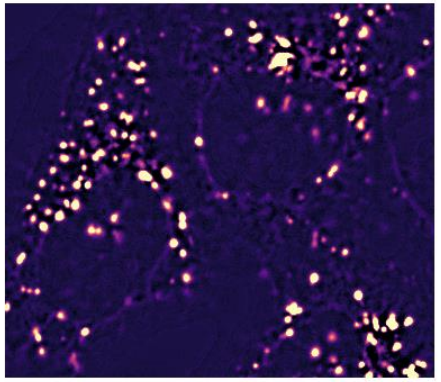
(B) Histogram of (A)



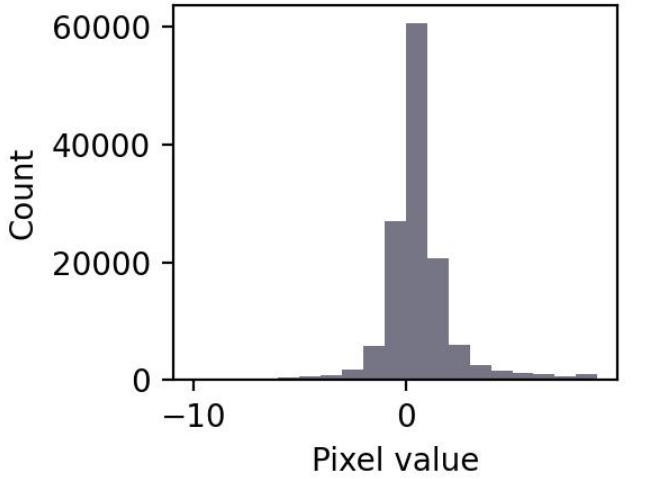
(C) Triangle threshold applied to (A)



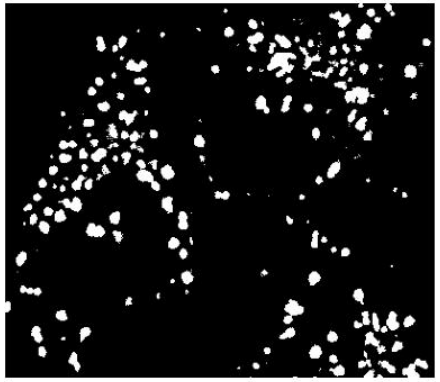
(D) 'Background' subtracted image



(E) Histogram of (A)

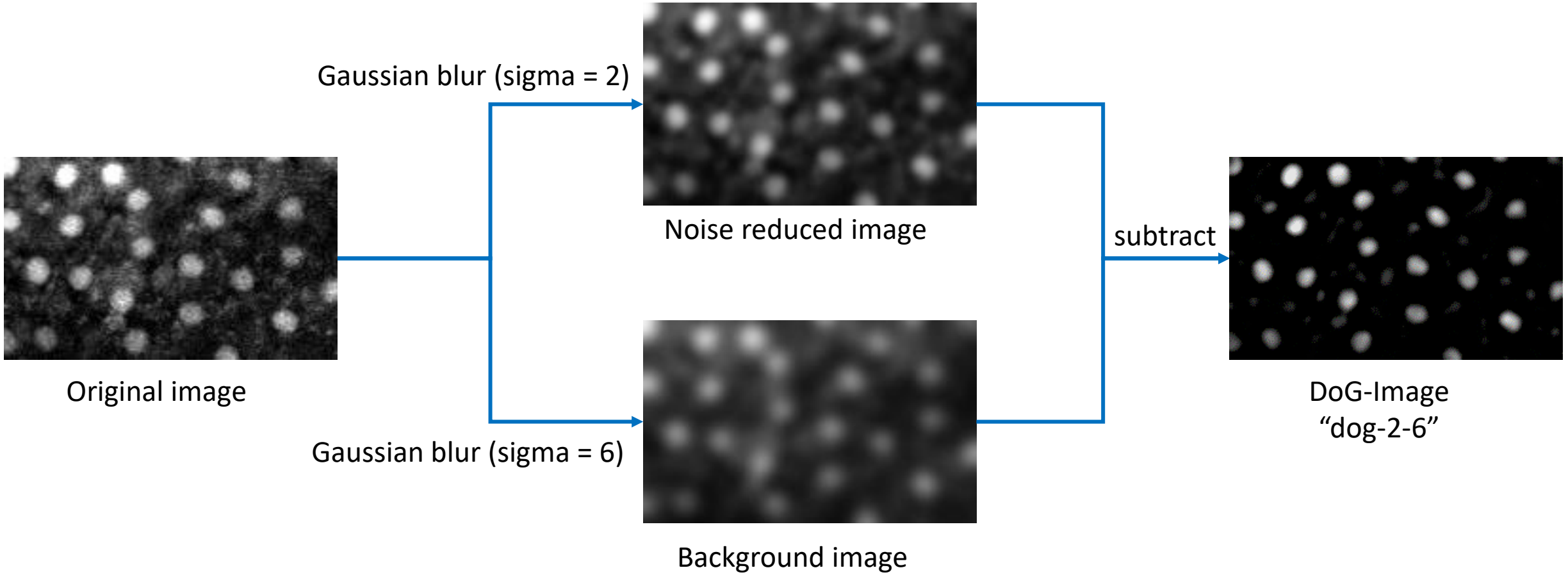


(F) Triangle threshold applied to (D)



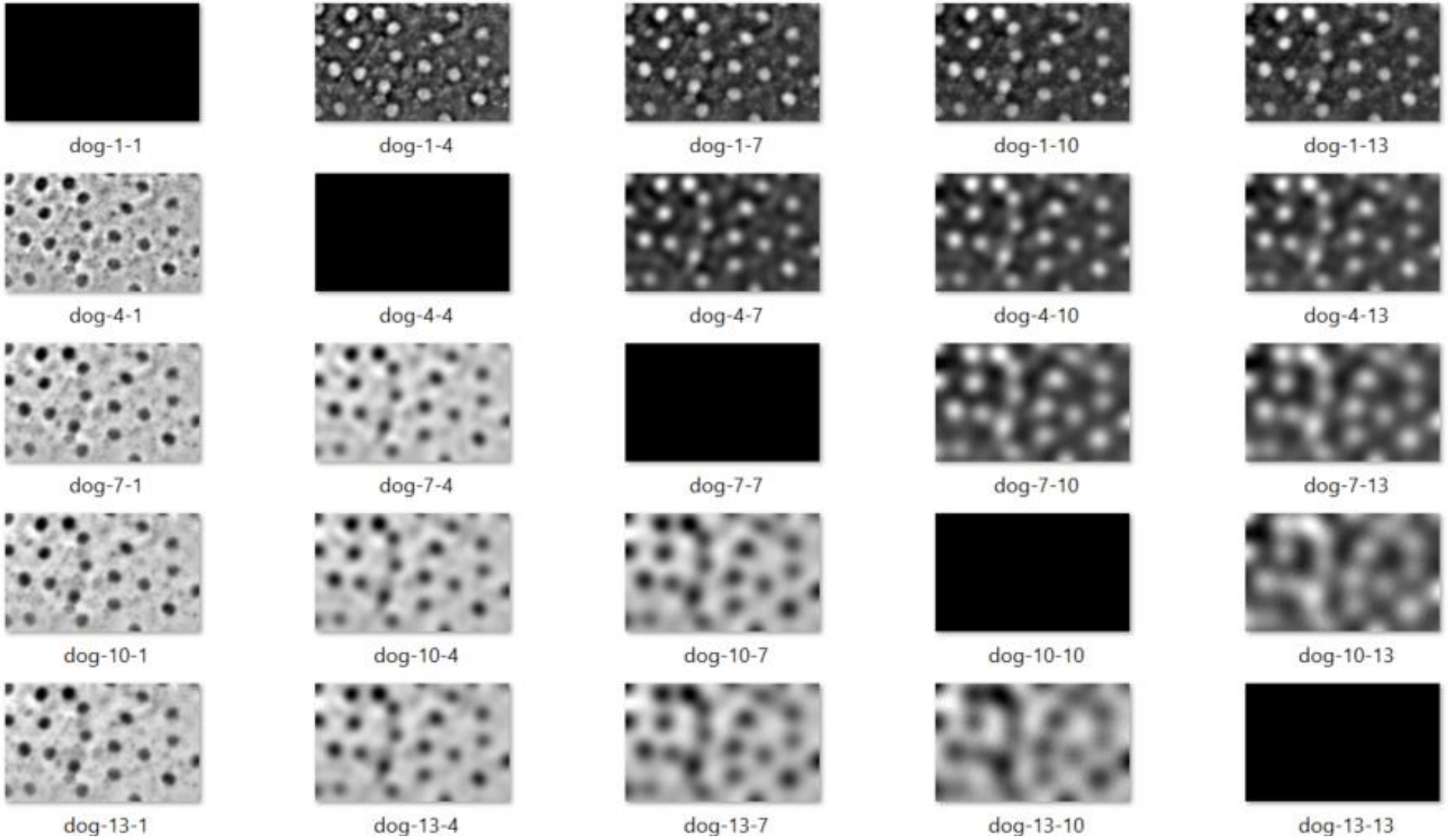
Difference-of-Gaussian (DoG)

- Improve image in order to detect bright objects.
- Band-pass filter



Difference-of-Gaussian (DoG)

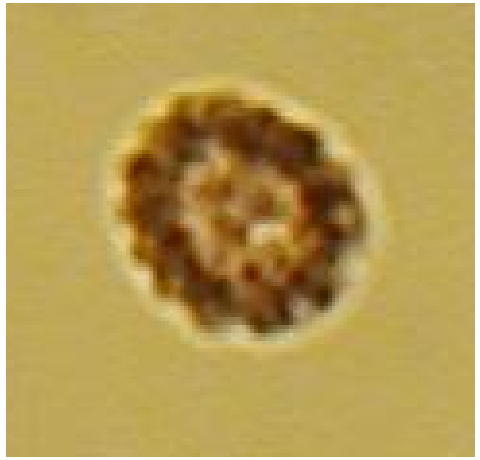
- Example DoG images

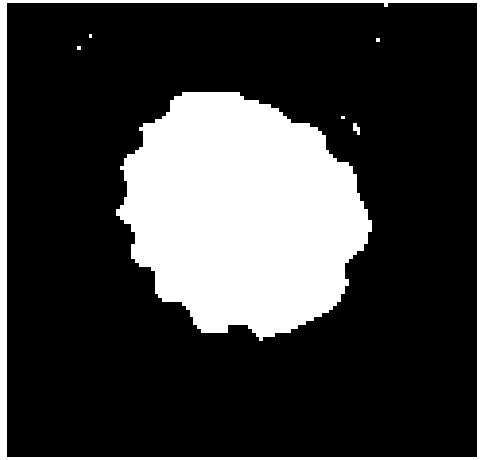
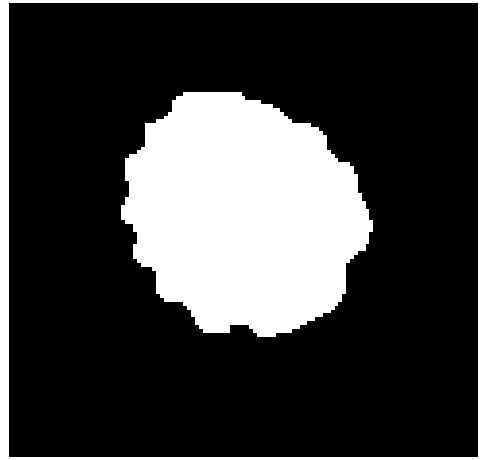


Refining masks

- Binary mask images may not be perfect immediately after thresholding.
- There are ways of refining them

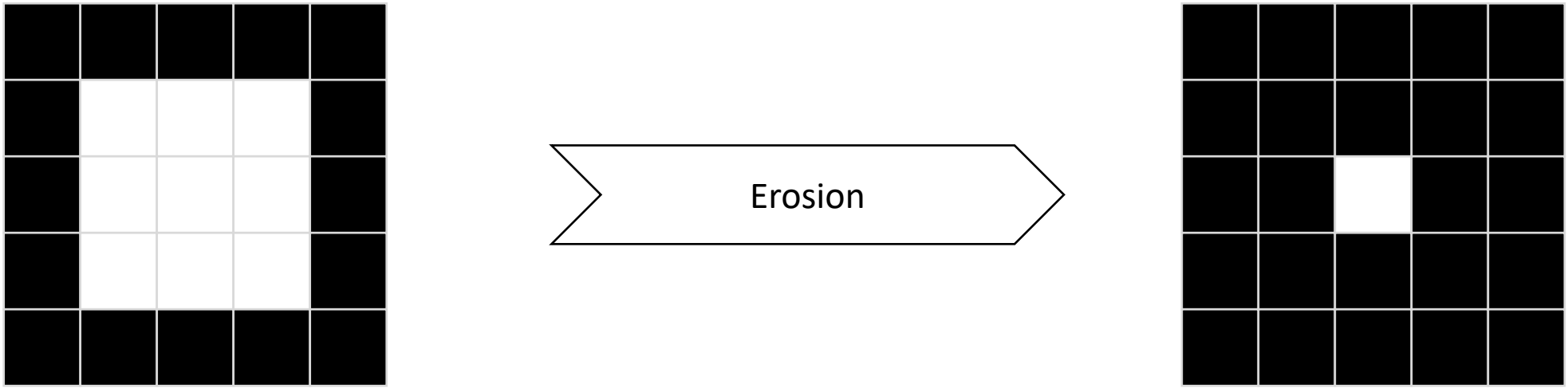
Thresholding → Closing → Opening



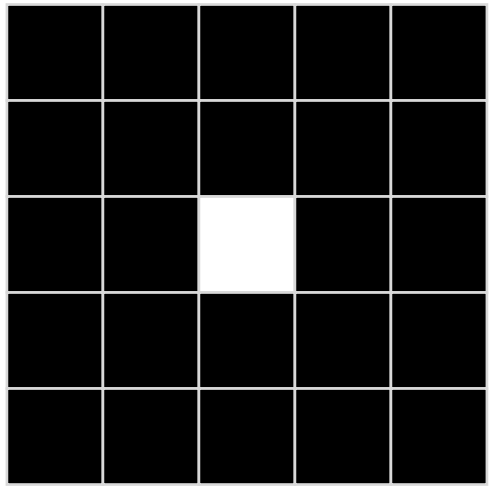
Erosion

- Erosion: Every pixel with at least one black neighbor becomes black.



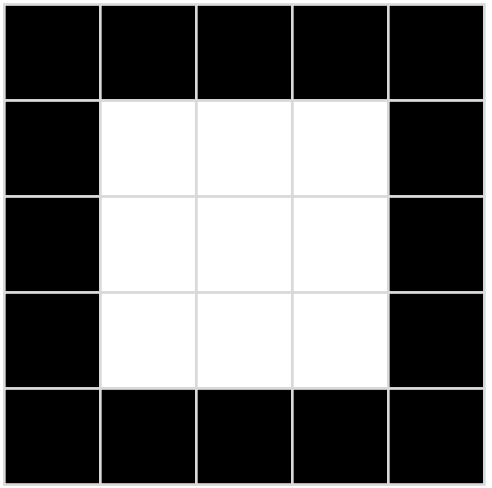
Dilation

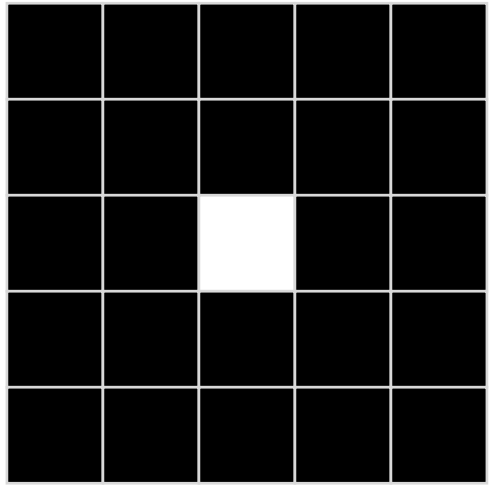
- Dilation: Every pixel with at least one white neighbor becomes white.



Dilation

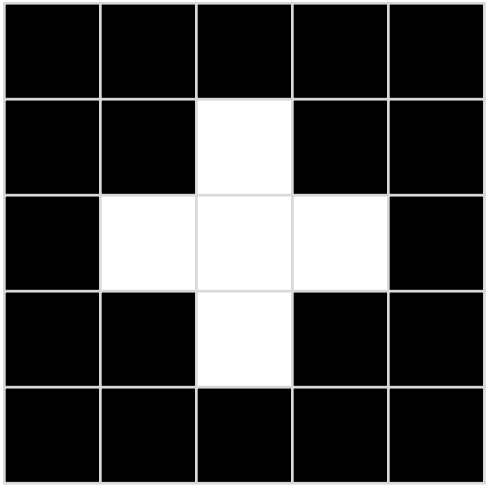
8-connected neighborhood
(BOX, ImageJ Default)





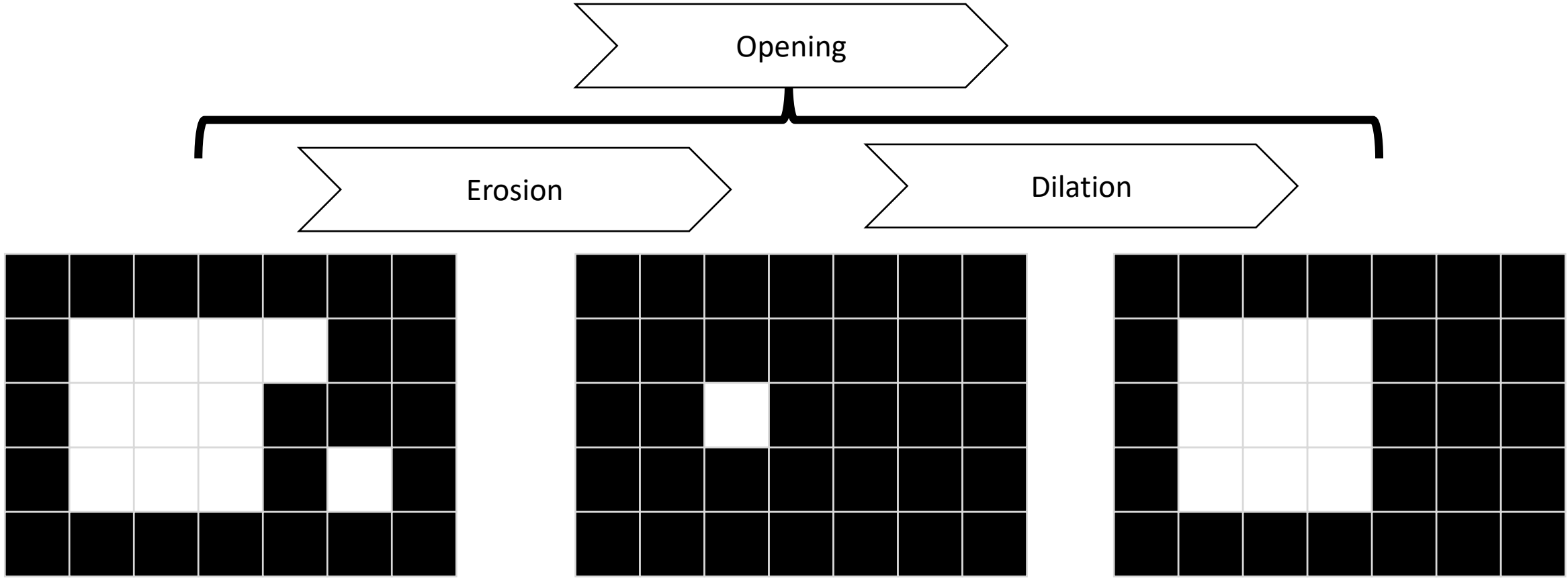
Dilation

4-connected neighborhood



Opening

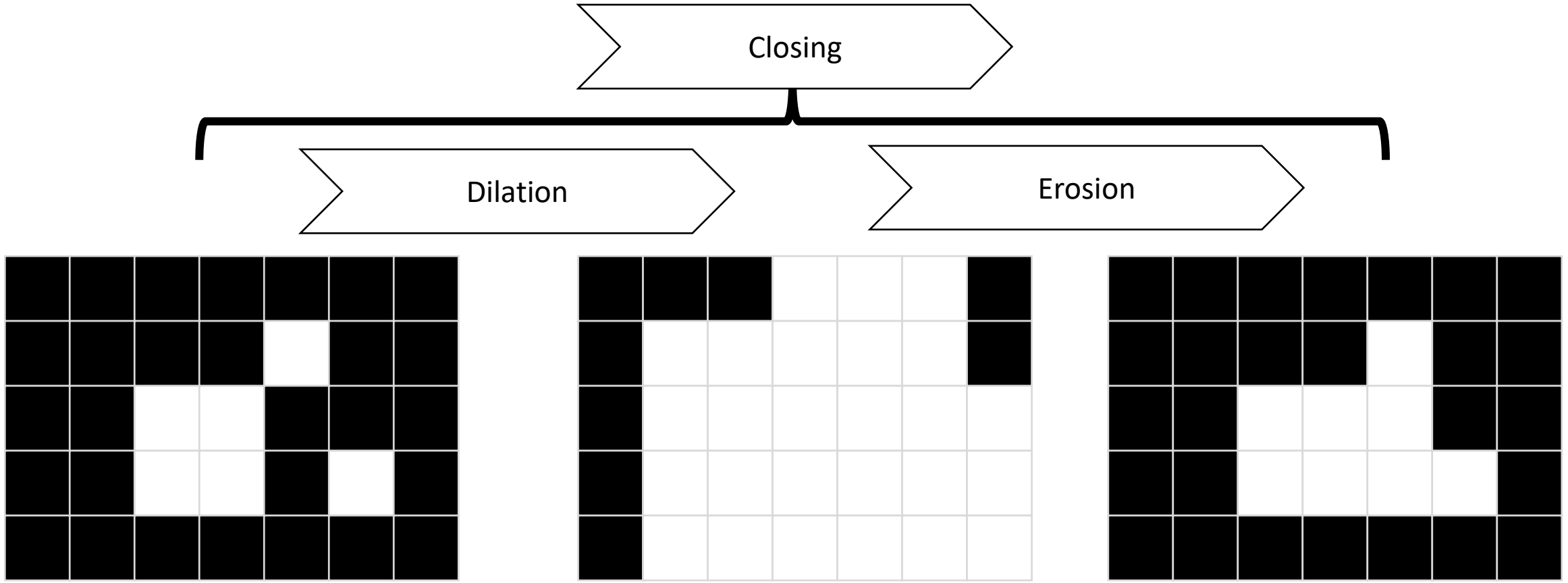
- Erosion and dilation combined allow correcting outlines.



- It can separate white (high intensity) structures that are weakly connected
- It may erase small white structures

Closing

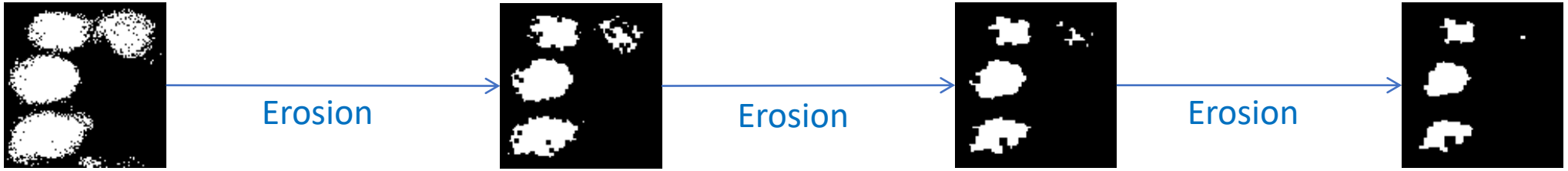
- Erosion and dilation combined allow correcting outlines.



- It can connect white (high intensity) structures that are nearby
- It may close small holes inside structures

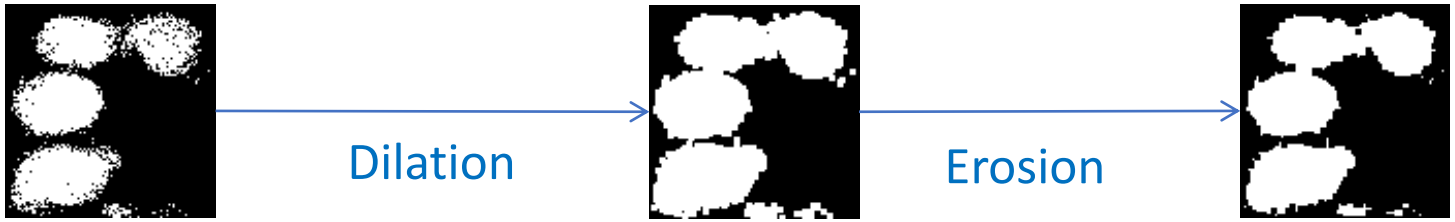
Chaining erosion and dilation

- Erosion: Set all pixels to black which have at least one black neighbor.



- Dilation: Set all pixels to white which have at least one white neighbor.

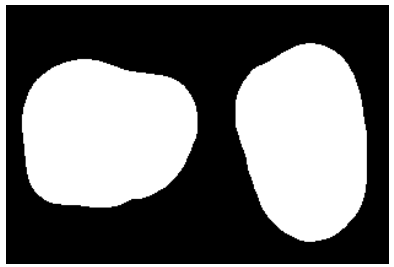
- Closing: Dilation + Erosion



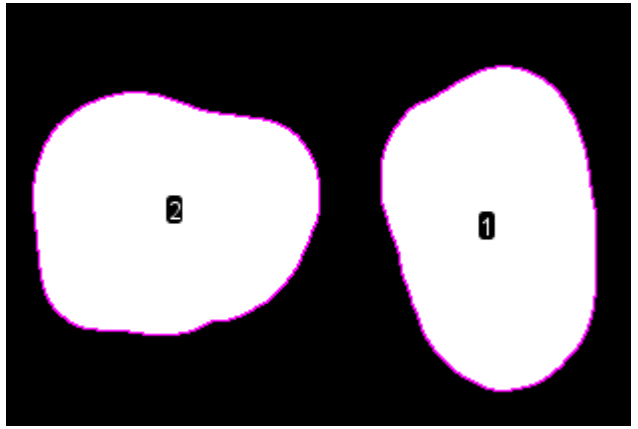
- Opening: Erosion + Dilation

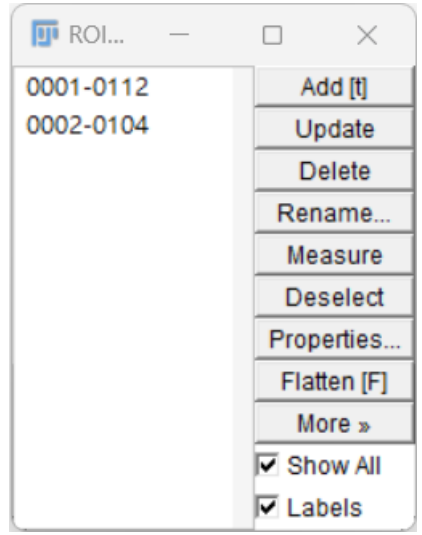
Analysis Particle: Convert Mask to ROI or Label

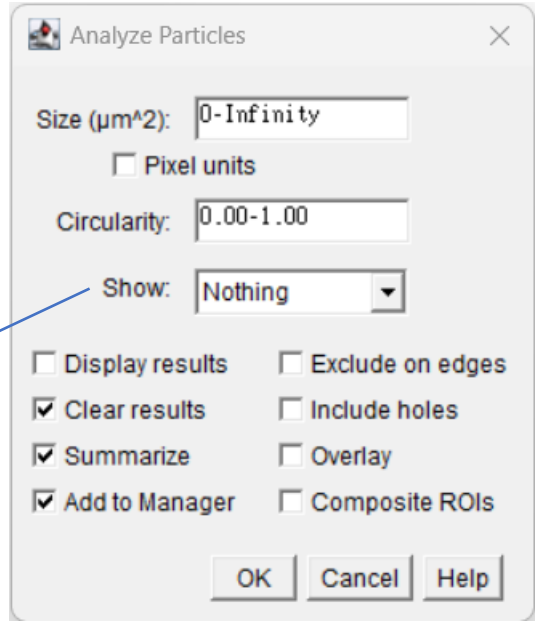
- Analyze -> Analyze Particle

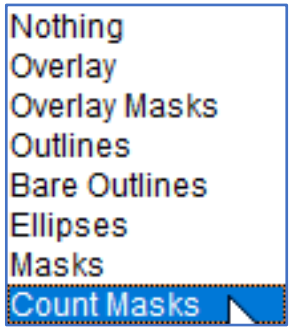


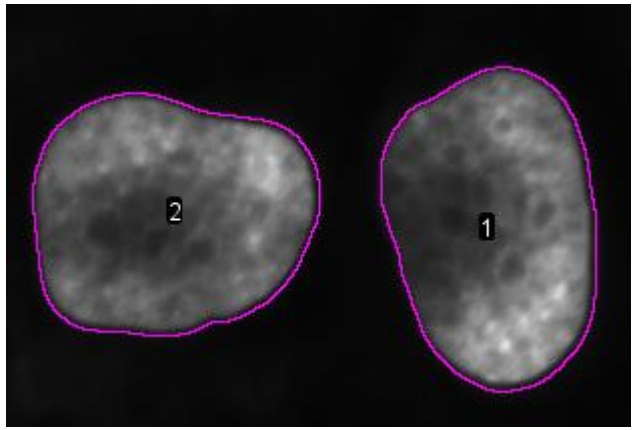
Analyze Particle

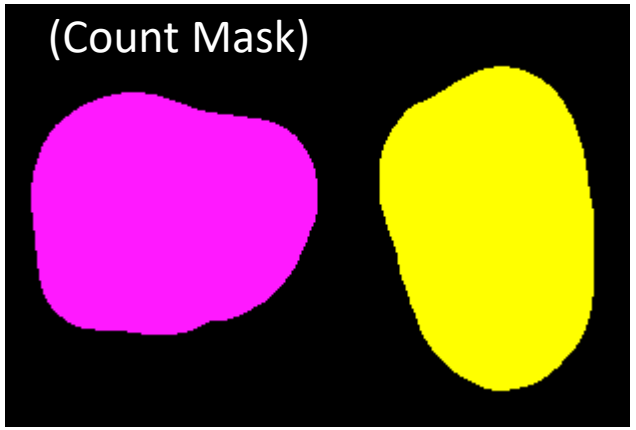


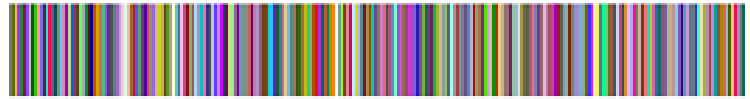










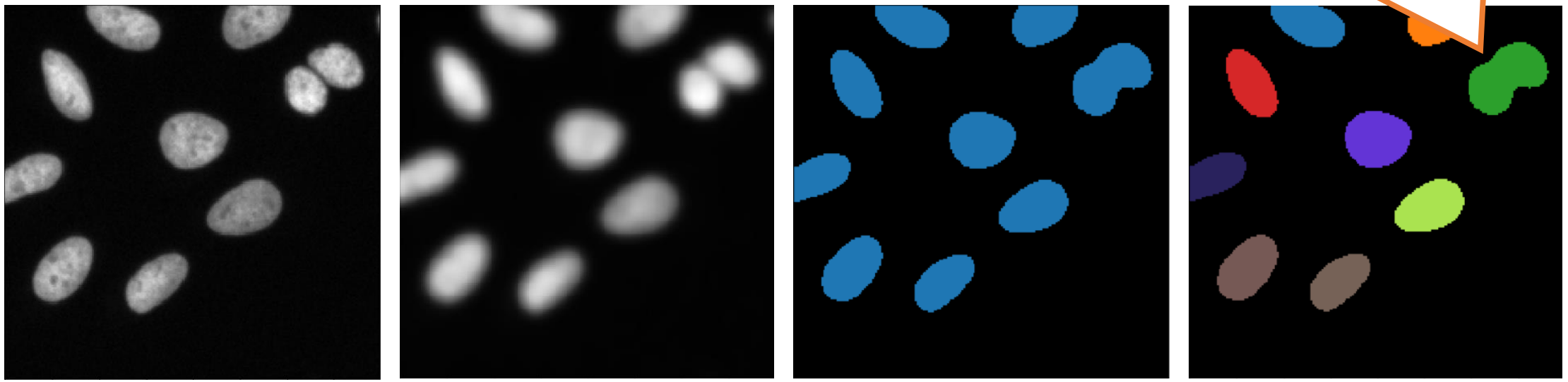


glasbey on dark

Common image segmentation workflows

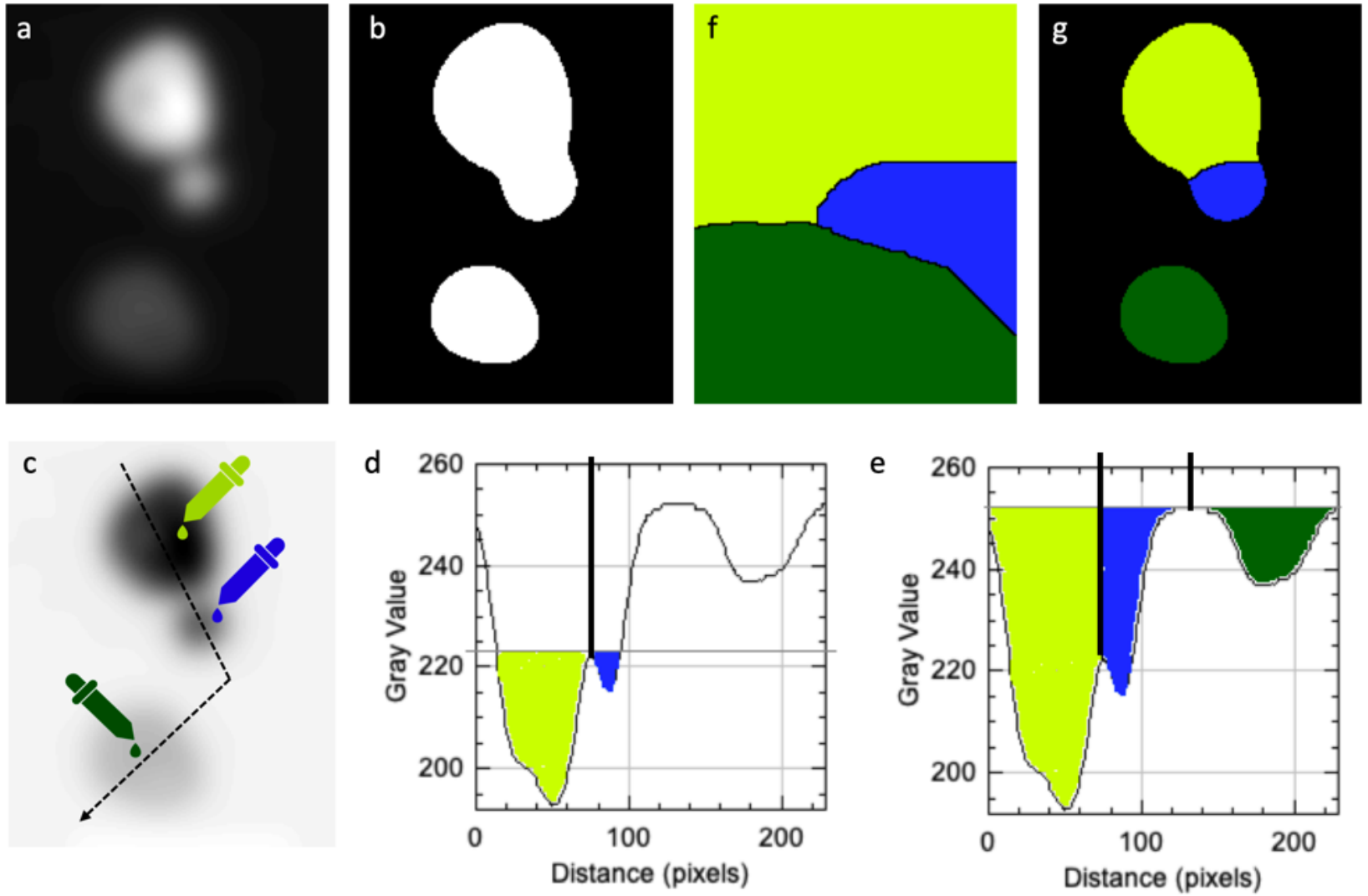
- Presumably the most common segmentation algorithm used for fluorescent microscopy images:
- Gaussian blur, Otsu's Threshold, Connected Component Labeling

Limitation:
Dense objects



Denoising → Binarization → Labeling

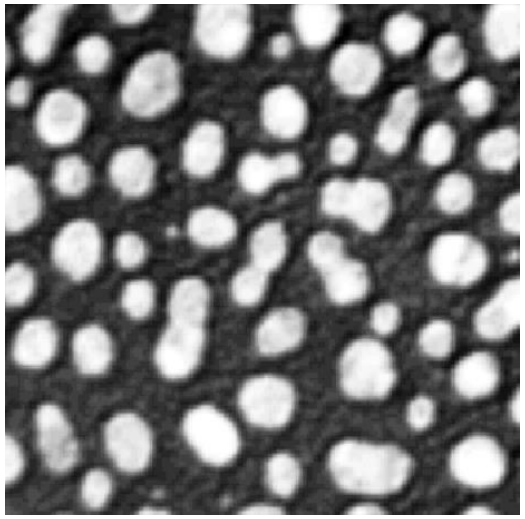
Watershed

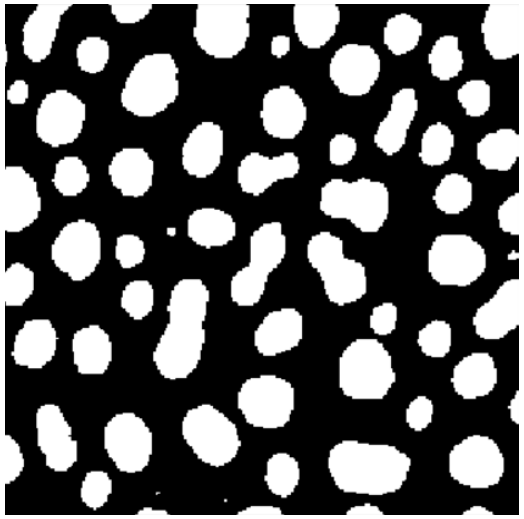


https://neubias.github.io/training-resources/watershed/index.html

Watershed use-cases

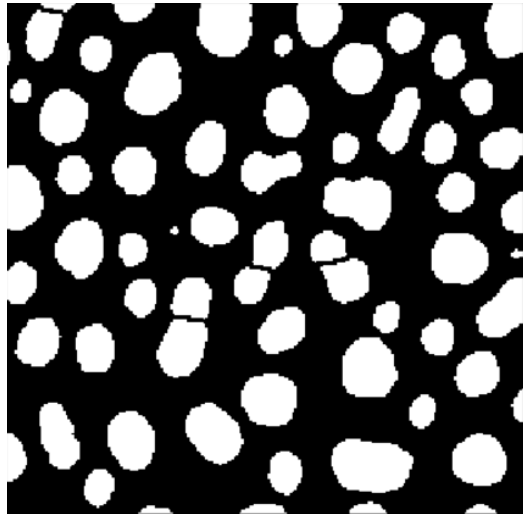
- Split dense objects

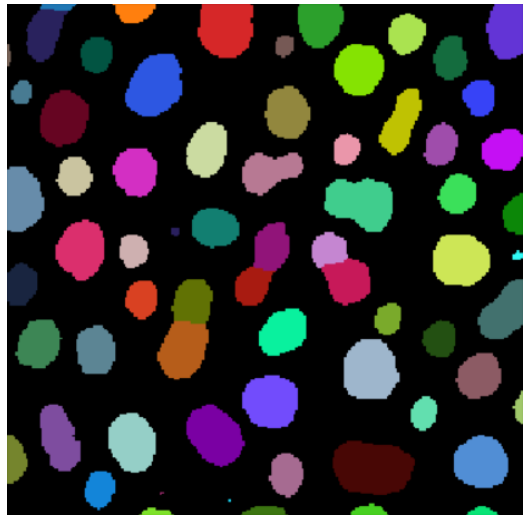




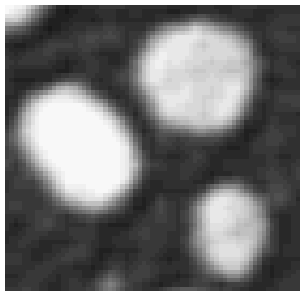
Binarization

Watershed

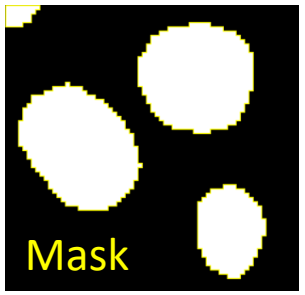




Segmentation with ROI list, Mask, and Label

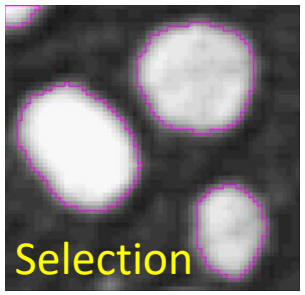


Thresholding



Create Selection
(Whole mask area selection)

Create Mask



Analysis Particle

Add to ROI Manager
Split

Select ROIs

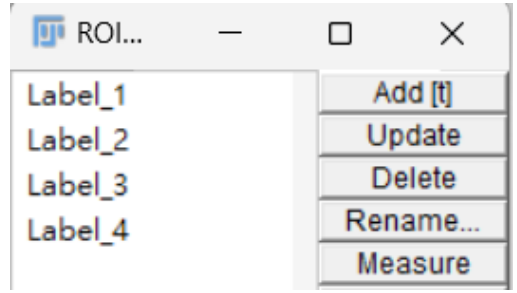
Analyze Particle
Connected Component
Labeling (CLIJ2)

Thresholding

AI-based segmentation tools

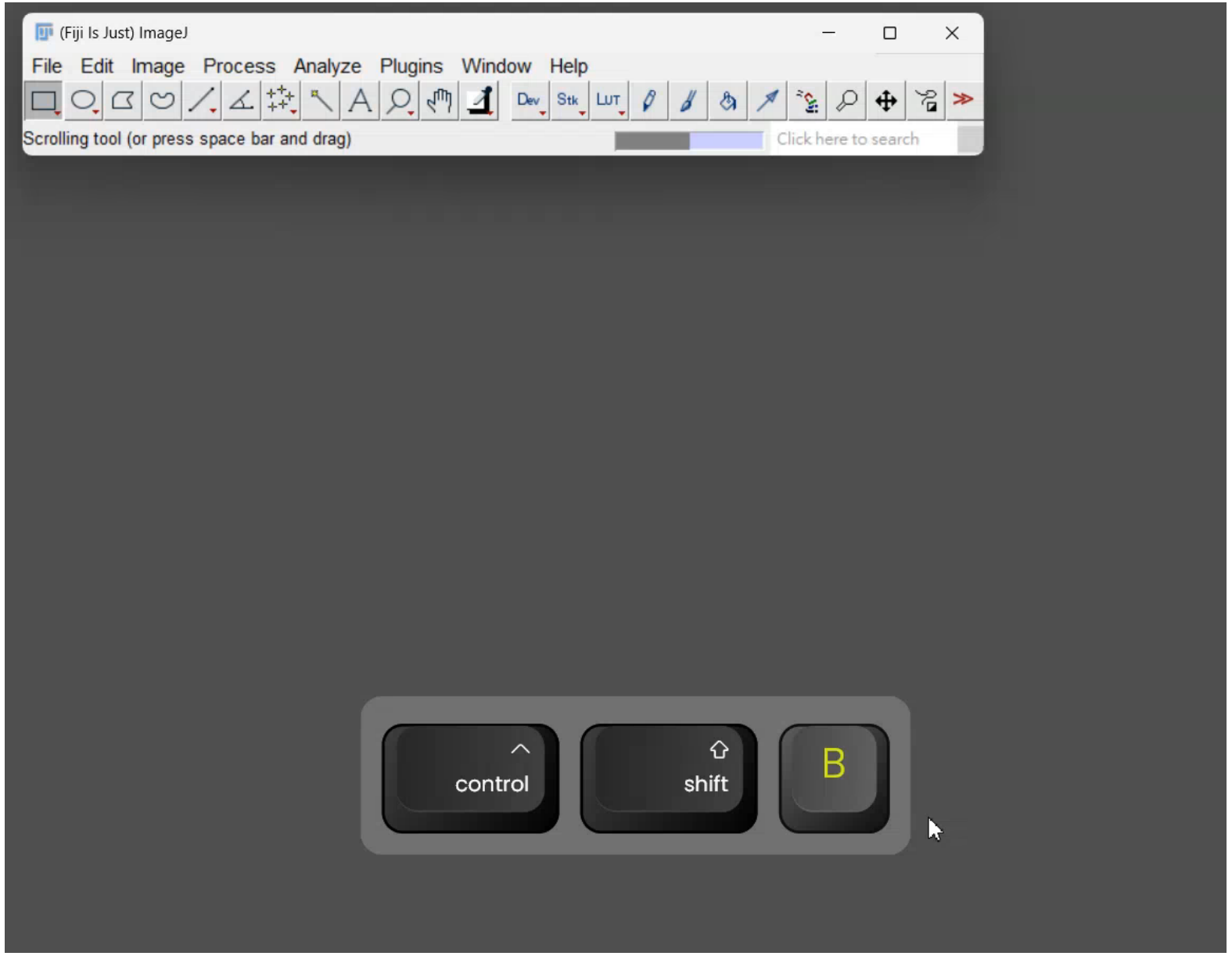
Label to ROI (BIOP)
pullLabelsToROIManager (CLIJ2)

ROI to Label (BIOP)



This slide is important!

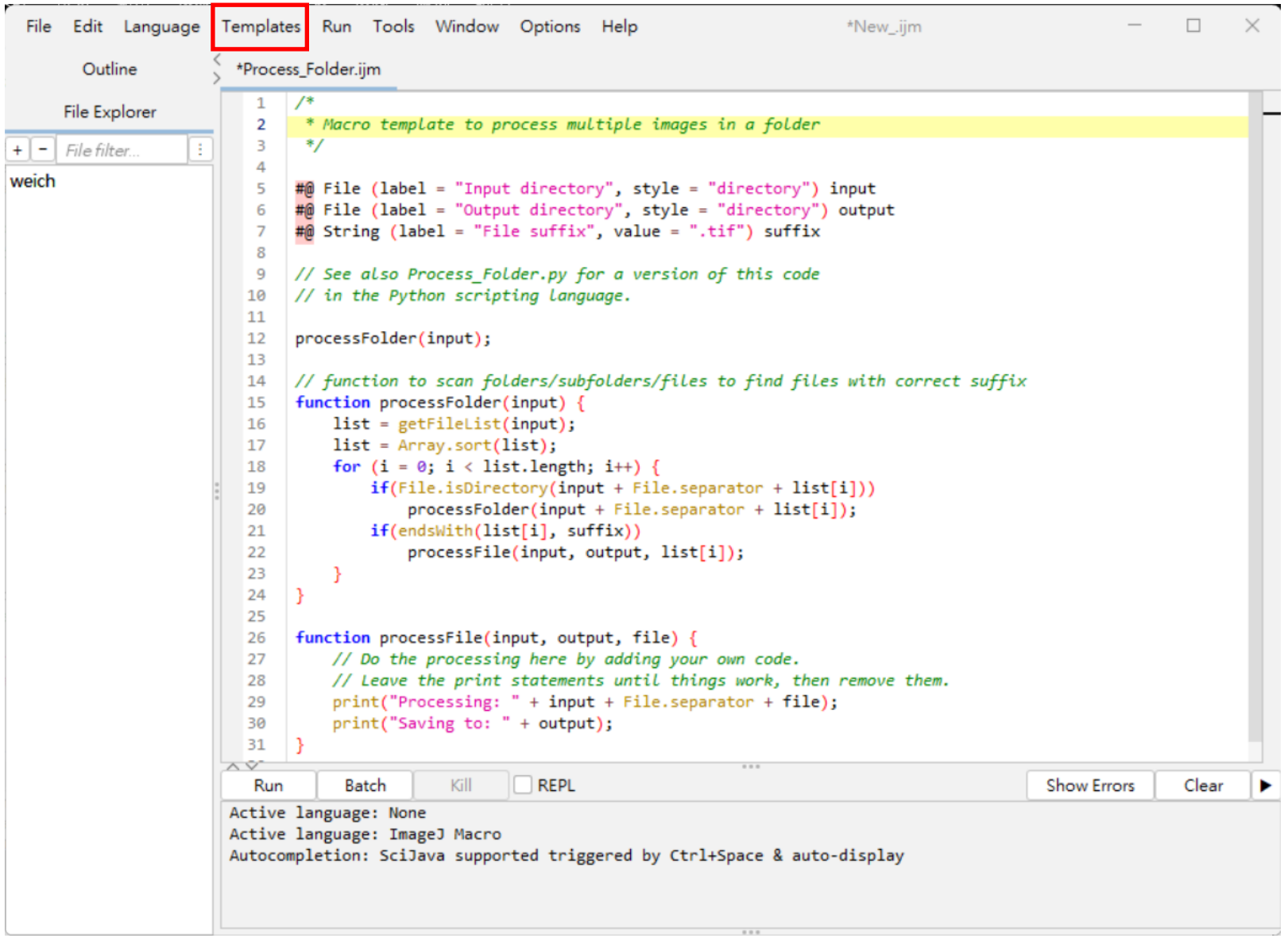
Typical segmentation workflow in Fiji



ImageJ macro

- Plugins -> Macros -> Record…
- File -> New -> Script
- Check Templates
- Use “Auto code completion”

  (Thanks to Dr. Robert Haase)
- Print a “Cheat Sheet” as reference
- Try and Error, Debug



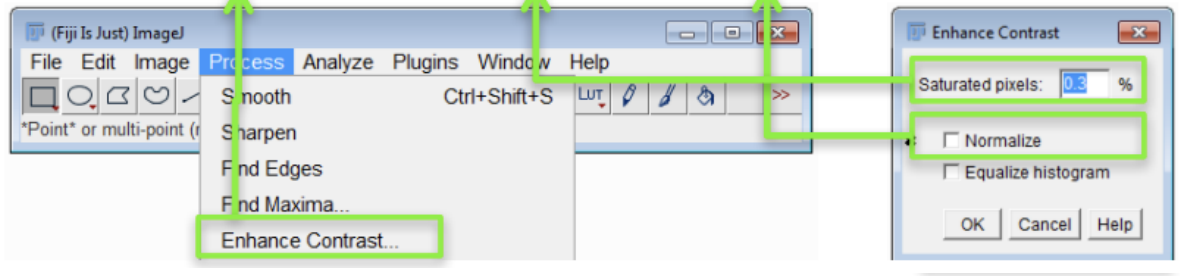
ImageJ macro CHEAT SHEET

Macro language elements

```
// comments for code documentation
numericVariable = 5;
stringVariable = "text value";
builtInCommand();
```

Calling any ImageJ/FIJI menu

```
run("Enhance Contrast...", "saturated=0.3 normalize")
```



Switch between image windows

```
titleOfCurrentImage = getTitle();
selectWindow(titleOfAnyImage);
```

Navigation in image stacks

```
Stack.getDimensions(width, height,
channels, slices, frames);
Stack.setSlice(slice);
Stack.setChannel(channel);
Stack.setFrame(frame);
Stack.setDisplayMode("color");
Stack.setDisplayMode("composite");
Stack.setDisplayMode("grayscale");
```

Handle image files and folders

```
open(folder+imageFilename);
close();
fileList = getFileList(folder);
numFiles = lengthOf(fileList);
for (i=0;i<lengthOf(fileList);i++){
    file = fileList[i];
    open(file);
    // actual image processing...
    close();
}
```

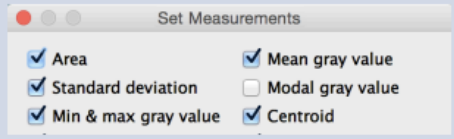
Reading image calibration

```
getPixelSize(unit, pWidth, pHeight);
getVoxelSize(vWidth, vHeight,
vDepth, unit);
```

Result tables

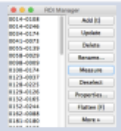
```
run("Set Measurements...", "area
mean standard min centroid");
```

corresp. to this:




```
run("Analyze Particles...",
"add clear display");
```

```
roiManager("Measure");
```



```
rowCount = nResults();
value = getResult("column title",
                  rowNumber);
setResult("column title",
rowNumber, newValue);
saveAs("Results","myResults.xls");
run("Clear results");
```

ROI manager

```
roiManager("add");
roiManager("split");
roiManager("delete");
roiManager("reset");
roiManager("measure");
roiManager("count");
roiManager("open", filename);
roiManager("save", filename);
roiManager("save selected", filename);
roiManager("select", index);
roiManager("select", newArray(index1,
index2, ...));
roiManager("deselect");
roiManager("show all");
roiManager("show all with labels");
roiManager("show none");
roiManager("and");
roiManager("combine");
```

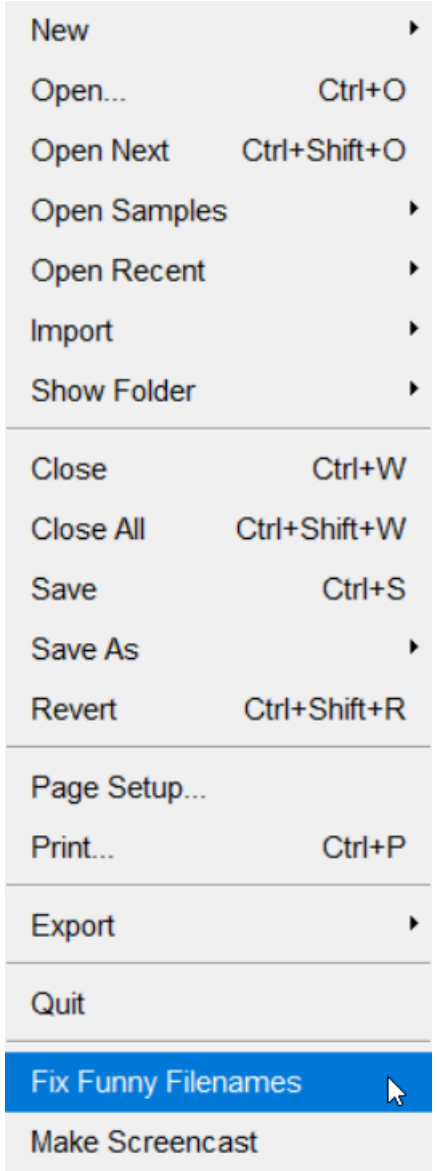
Keep in mind:

- Only one active window!
- One activate channel, slices!
- One ROI list
- One result table!

Fix funny filename

- Rename files with problematic file or folder names automatically
  image 001.tif → image_001.tif

- Try it if your ImageJ macro can not find you file or folder!



CLIJ: GPU accelerated image processing for everyone

Integrated GPU-acceleration into ImageJ / Fiji using ImageJ Macro “instead” of OpenCL.

CLIJ

OpenCL ImageJ

Dr. Robert Haase
Clusters Excellence “Physic of Life” (PoL),
TU Dresden, Germany

https://haesleinhuepf.github.io/

https://clij.github.io/

Benchmarking in Fiji

Comparison of processing performance Fiji versus CLIJ: 1-2 orders of magnitude speedup are possible!

vs.

Workstation CPU

2x Intel Xeon Silver 4110

Laptop CPU

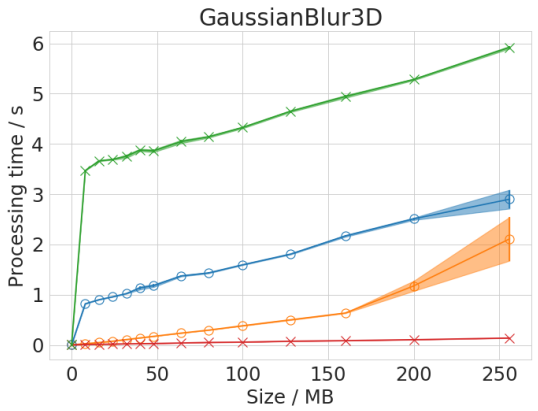
Intel Core i7-8650U

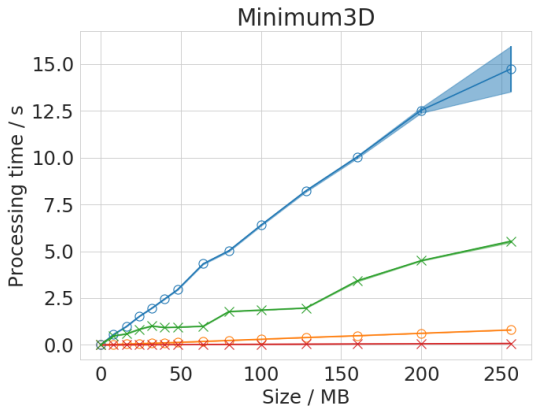
Workstation GPU

Nvidia Quadro P6000

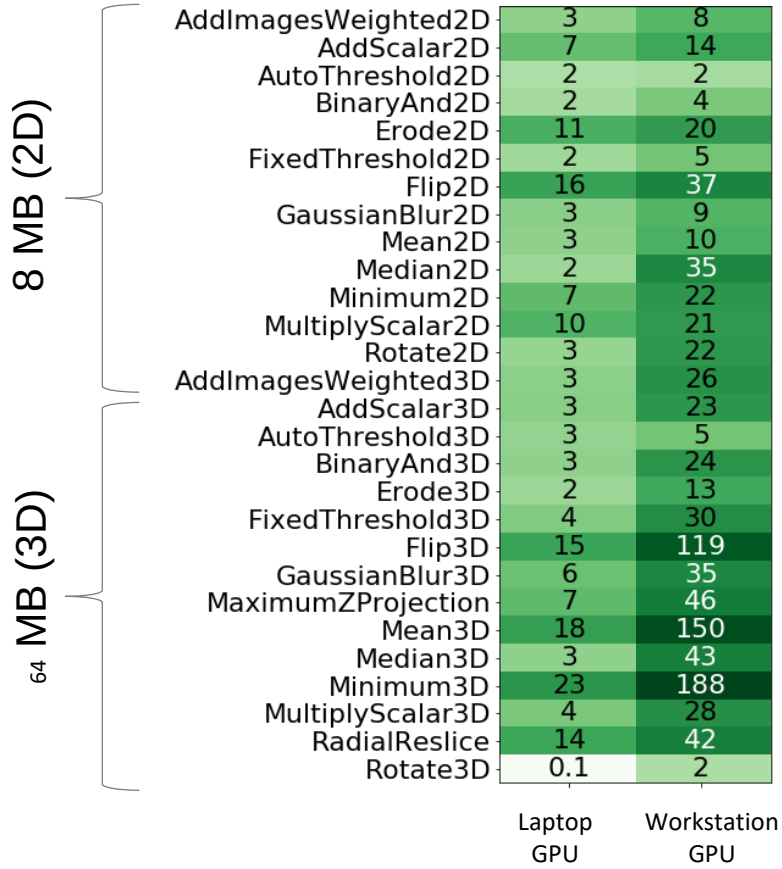
Laptop GPU

Intel UHD 620 GPU





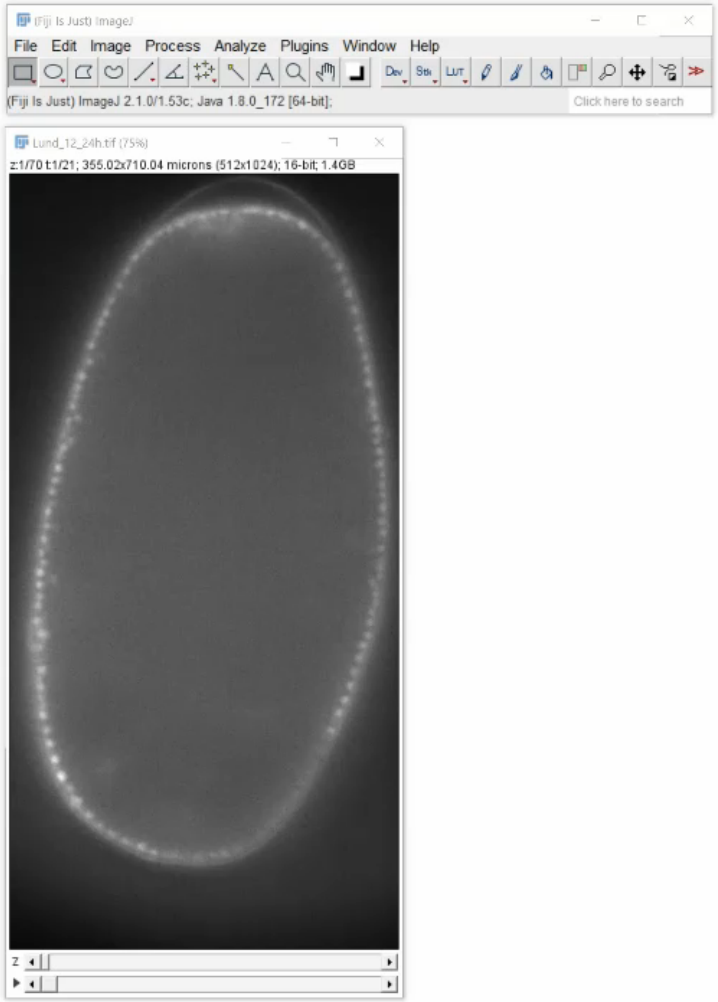
Speedup factor compared to Laptop CPU



https://www.youtube.com/watch?v=uMj0OS1TiQE&t=873s

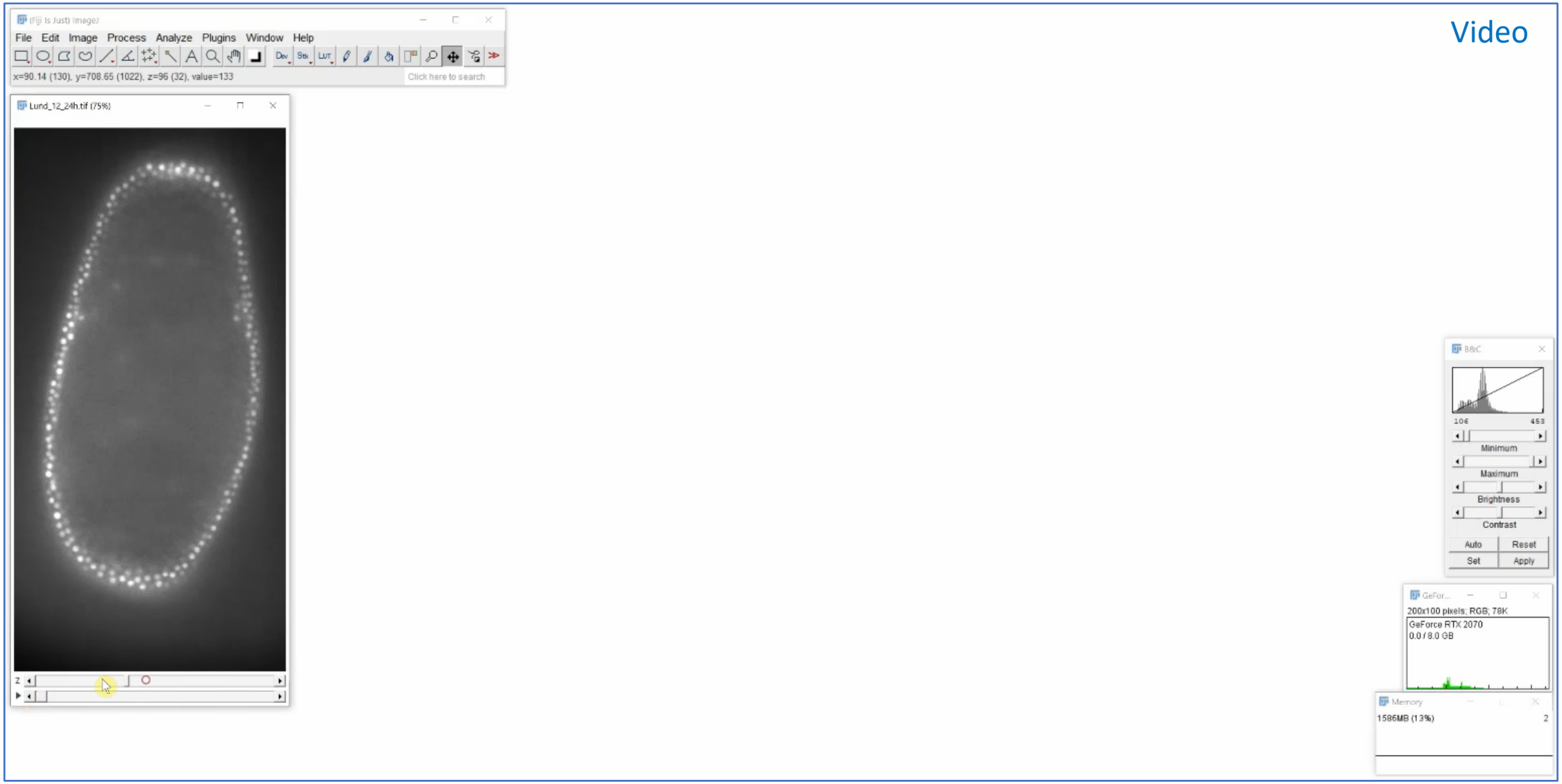
Nat Methods 17, 5-6 (2020) doi:10.1038/s41592-019-0650-1

Image processing in life-sciences (Fiji)



Video

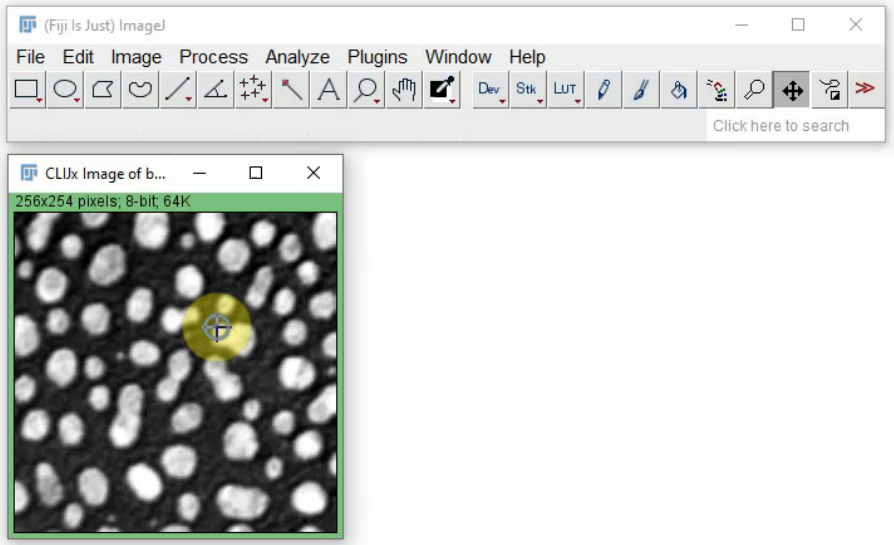
How image processing is supposed to be



Adapted from the learning material of Dr. Robert Haase
https://f1000research.com/slides/10-978

CLIJ-Assistant

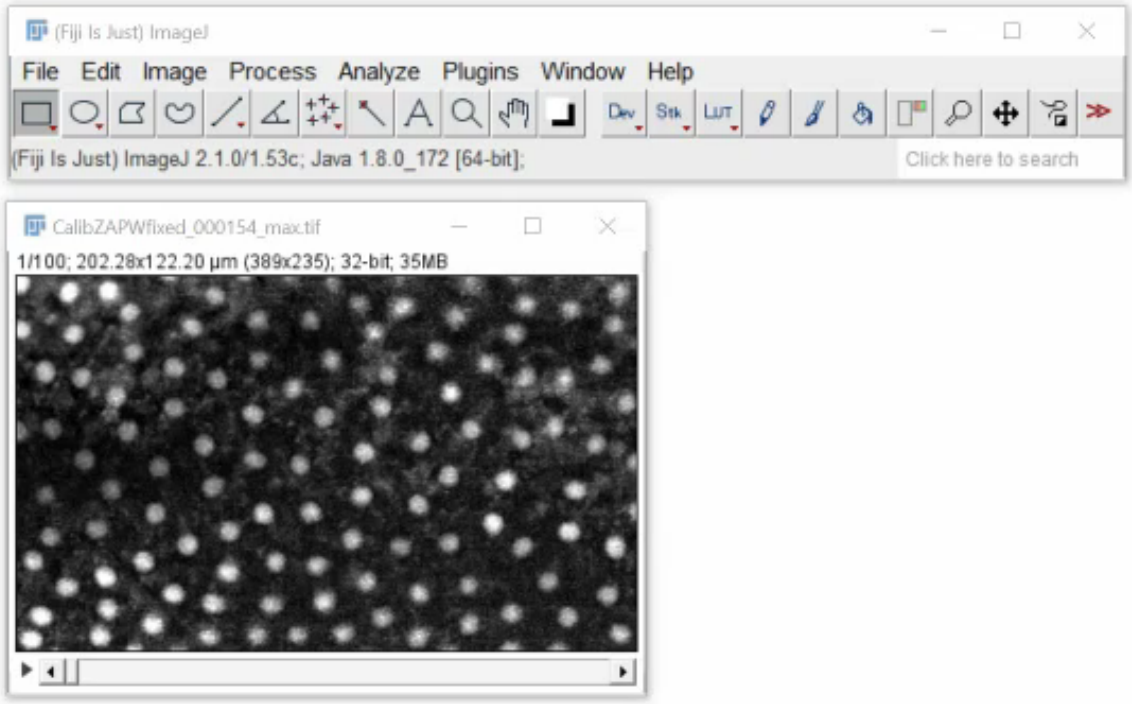
- Interactive workflow design



Video

Timelapse processing

Setup the workflow on one timepoint and check if it’s working on other timepoints



Video

GPU acceleration + code generation

After setting up the workflow, generate code!

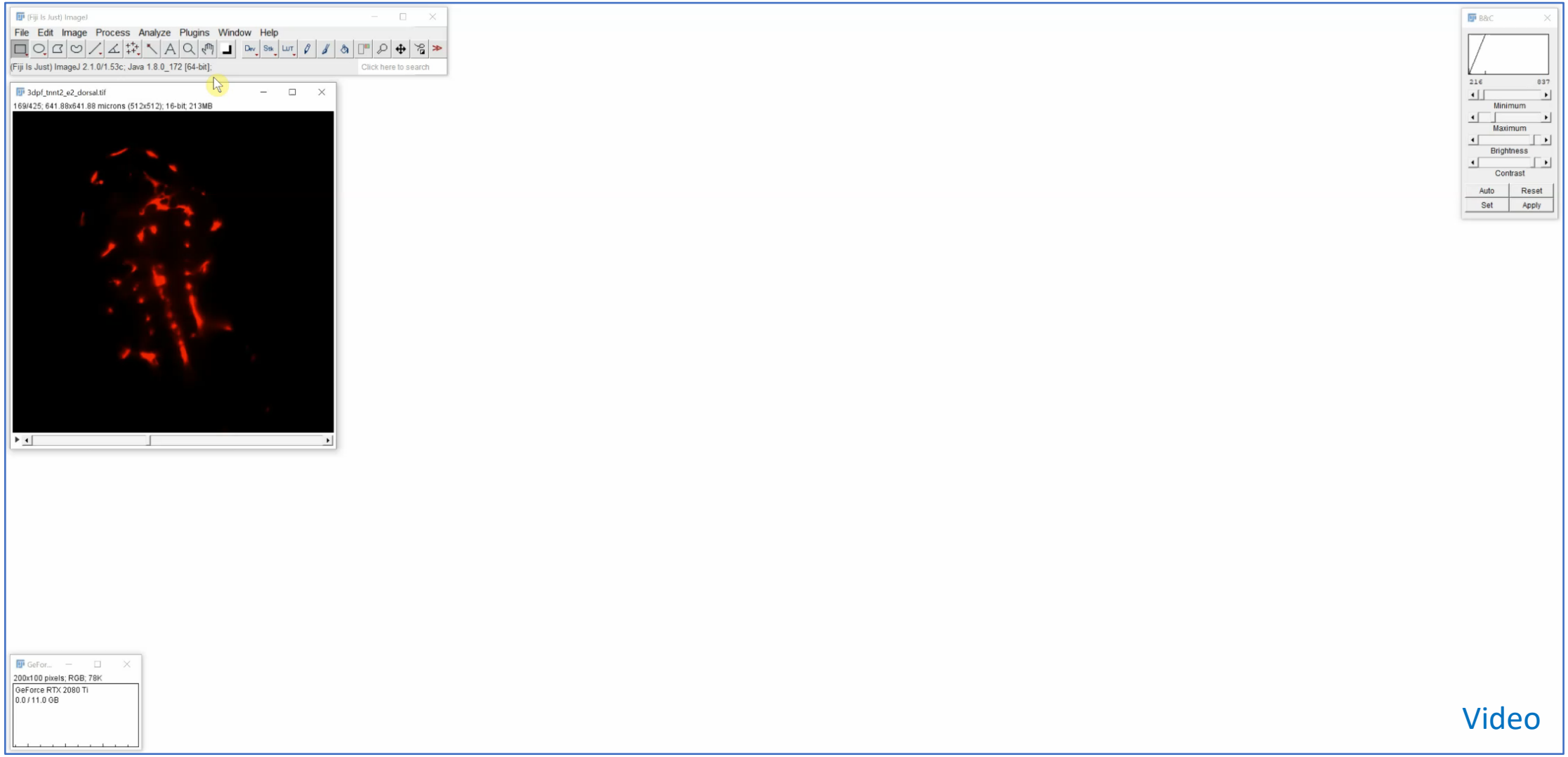


Image data source: Elisabeth Kugler; labs of Tim Chico and Paul Armitage, The University of Sheffield (UK)" https://zenodo.org/record/4204839#.X8DCRGj7Q2w
https://f1000research.com/slides/10-978

CLIJ2: What every ImageJ Macro script must have

Load data

Initialize GPU

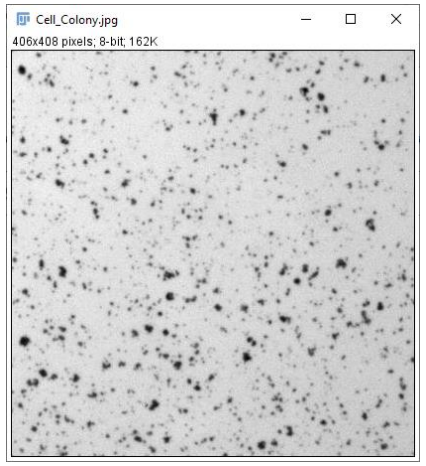
Push

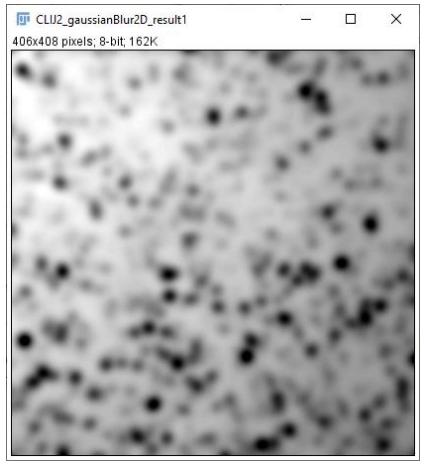
Process images

Pull

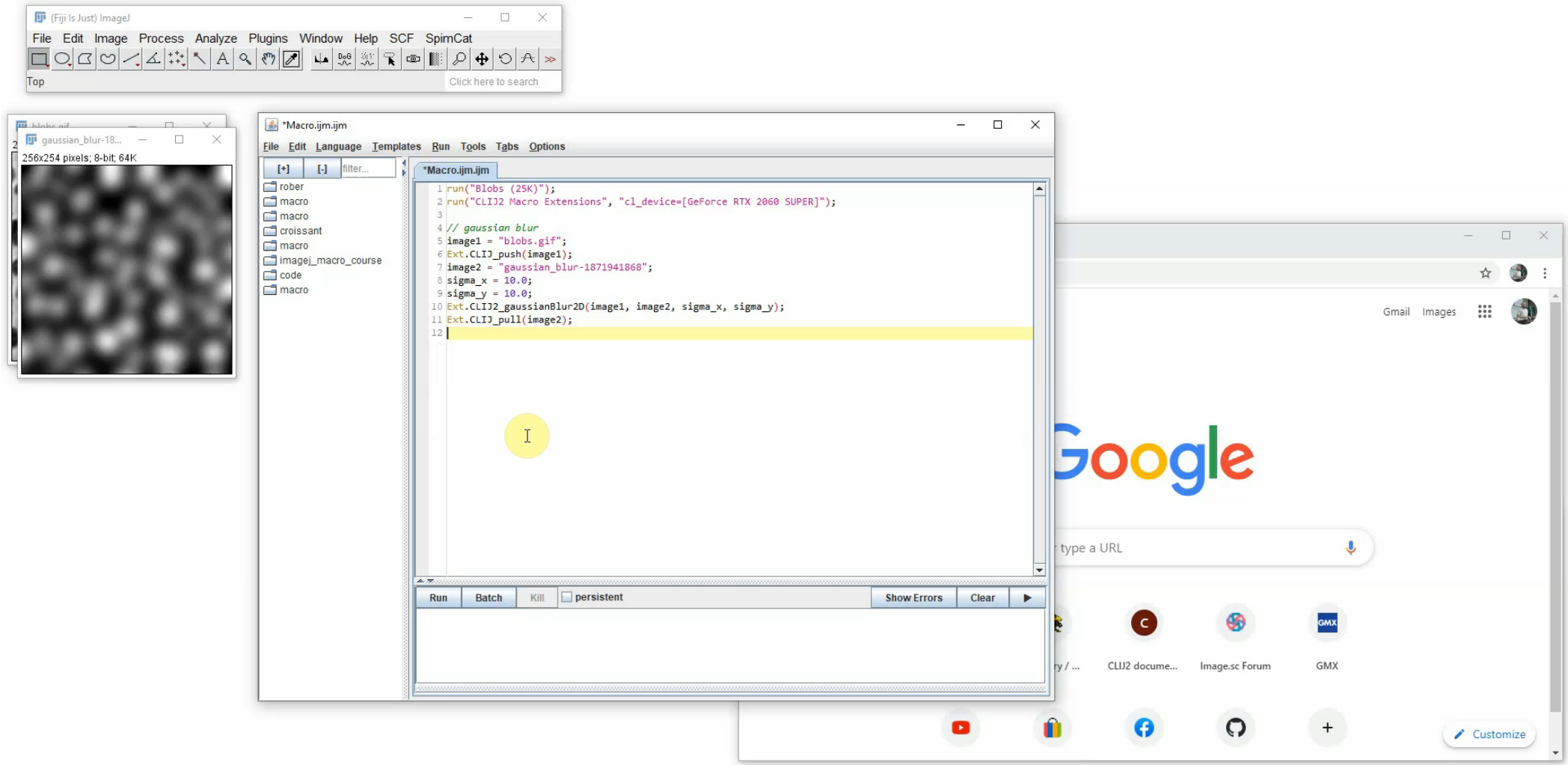
Cleanup

```
// load data
run("Cell Colony (31K)");

// initialize GPU
run("CLIJ2 Macro Extensions", "cl_device=");
Ext.CLIJ2_clear();

// push image to GPU
input_image = getTitle();
Ext.CLIJ2_push(input_image);

// process image
sigma = 5;
Ext.CLIJ2_gaussianBlur2D(input_image, result_image, sigma, sigma);

// optional: release input data
Ext.CLIJ2_release(input_image);

// pull result back from GPU
Ext.CLIJ2_pull(result_image);

// clean up by the end
Ext.CLIJ2_clear();
```





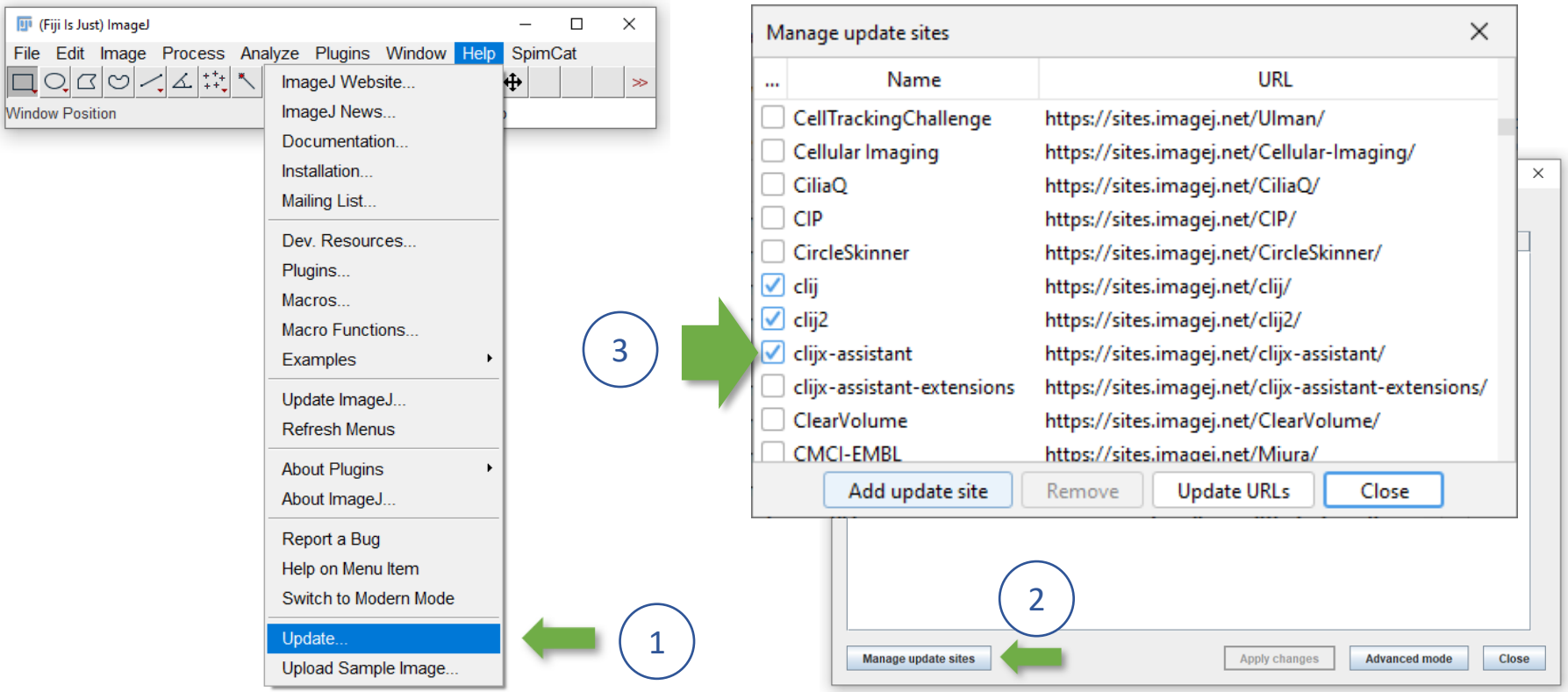
CLIJ2: Macro editing



Adapted from the learning material of Dr. Robert Haase

CLIJ: installation

- Just activate the CLIJ update sites, in menu Help > Update…



installation:

- clij
- clij2
- clijx-assistant

eXperimental

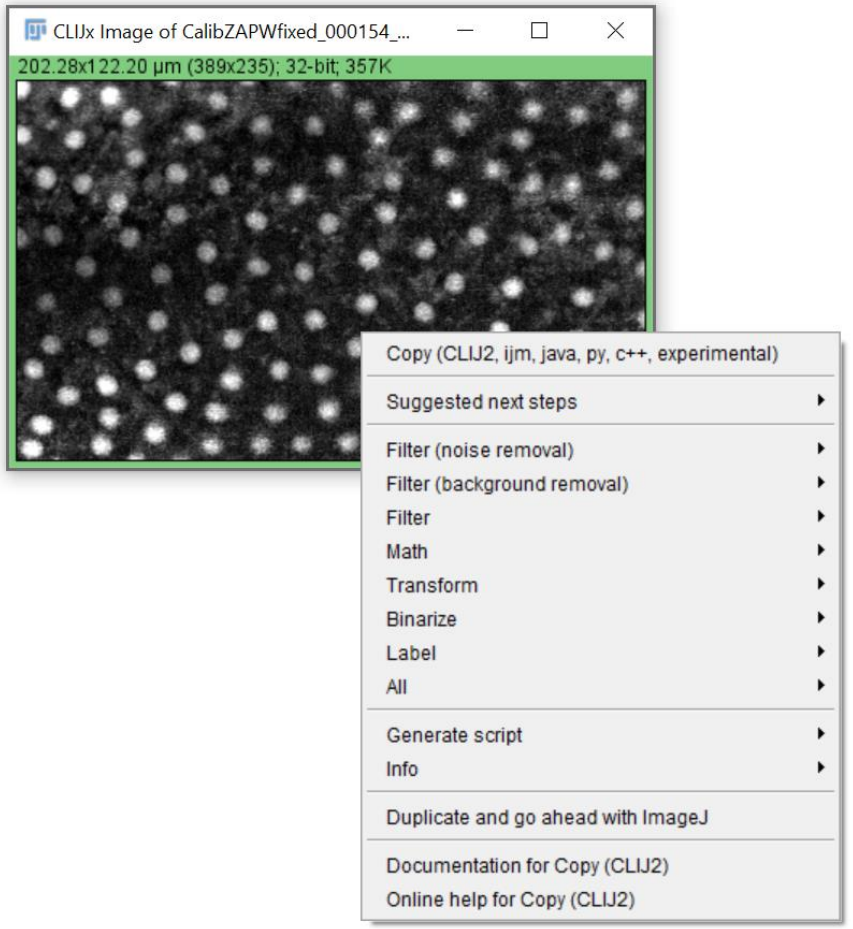
Advanced installation:

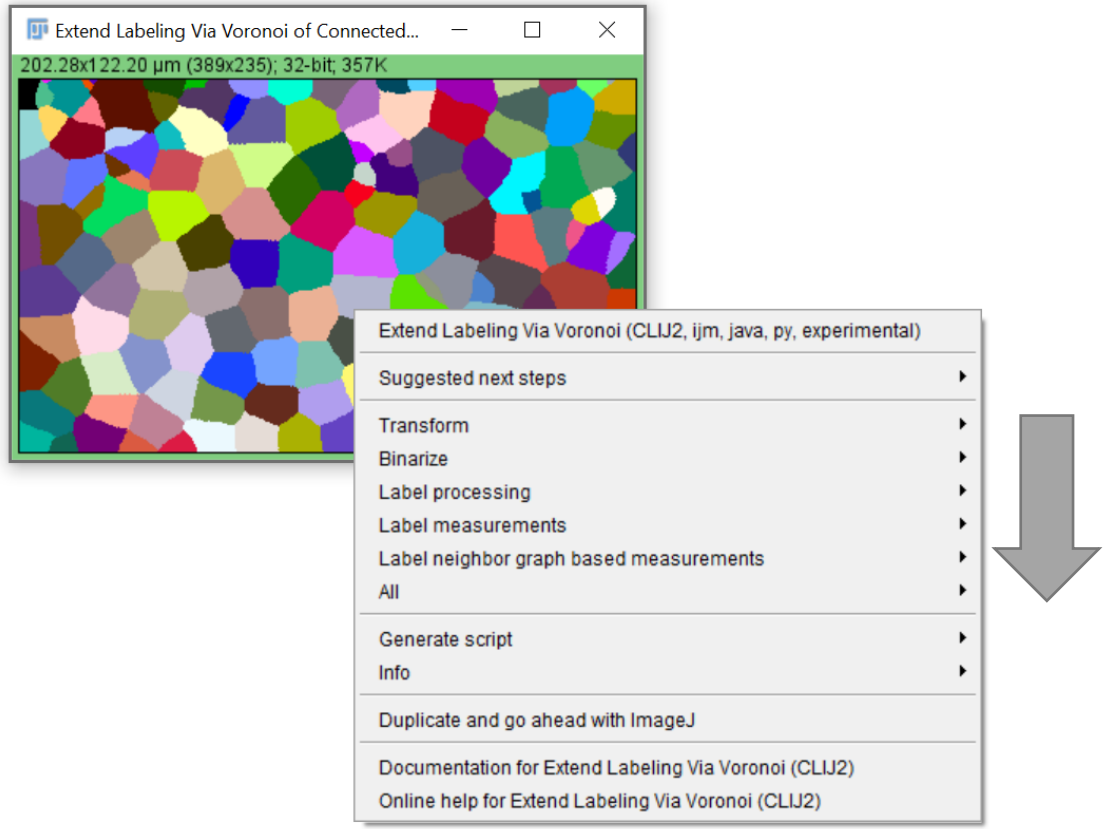
- 3D ImageJ Suite
- BoneJ
- IJPB-Plugins
- clijx-assistant-extensions

https://clij.github.io/clij2-docs/installationInFiji

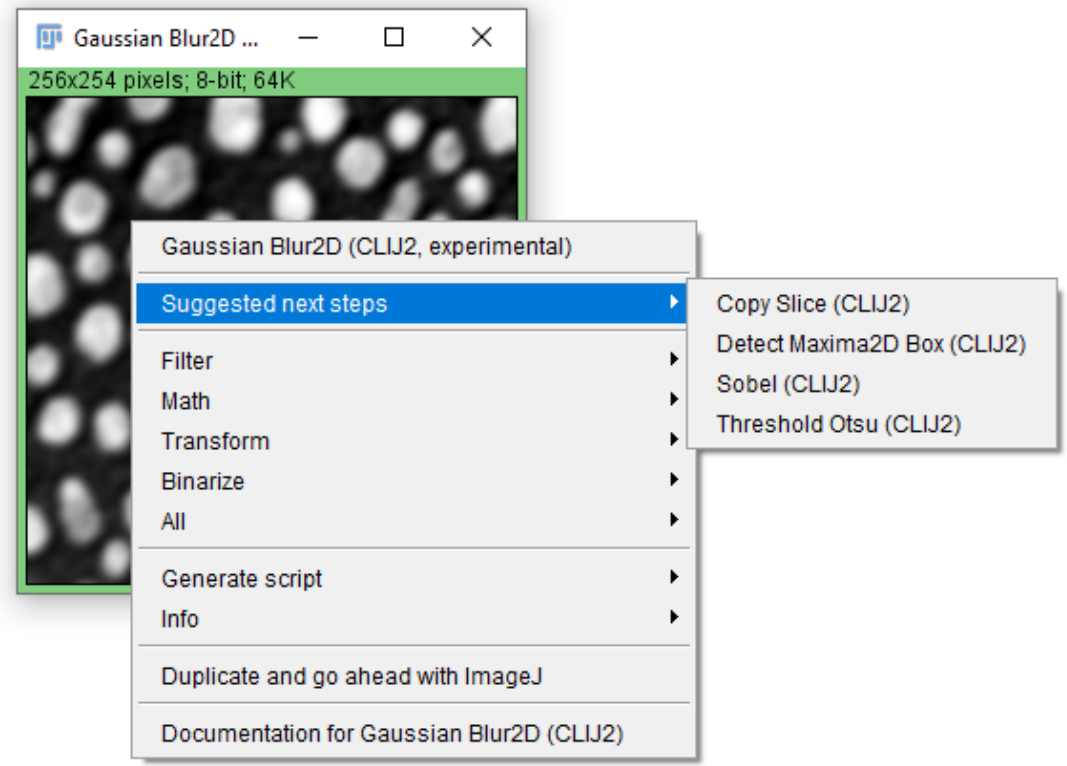
Image Data Flow Graph design

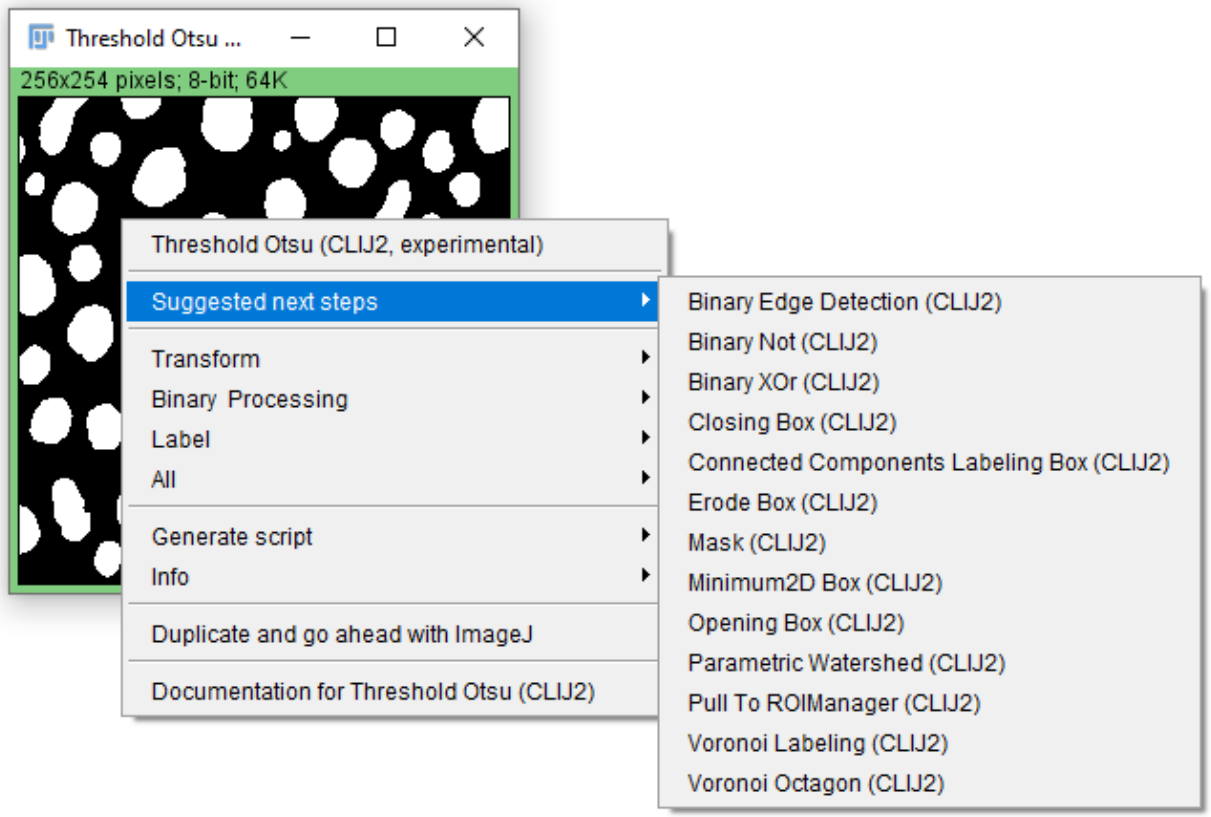
The menu order is intentional: From preprocessing to analysis





Expert system: Suggestions

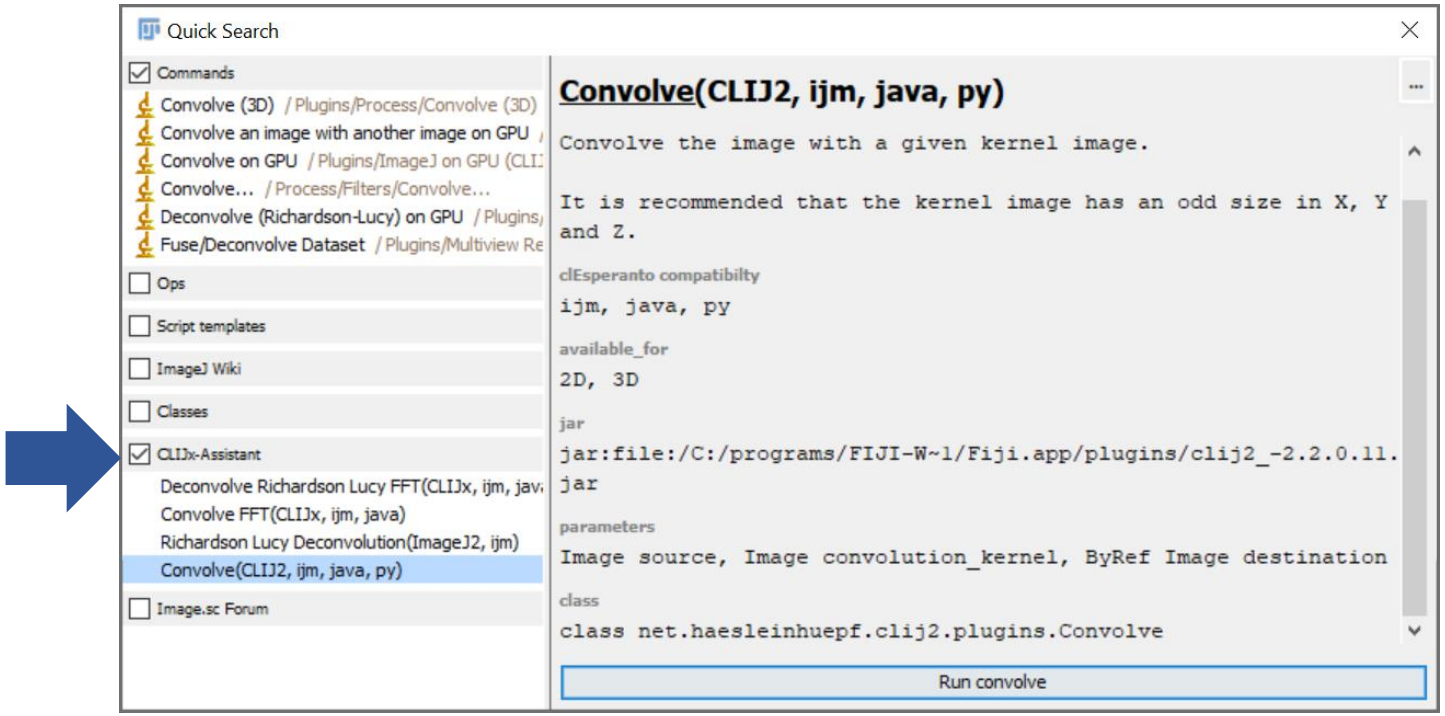




Adapted from the learning material of Dr. Robert Haase
https://f1000research.com/slides/10-978

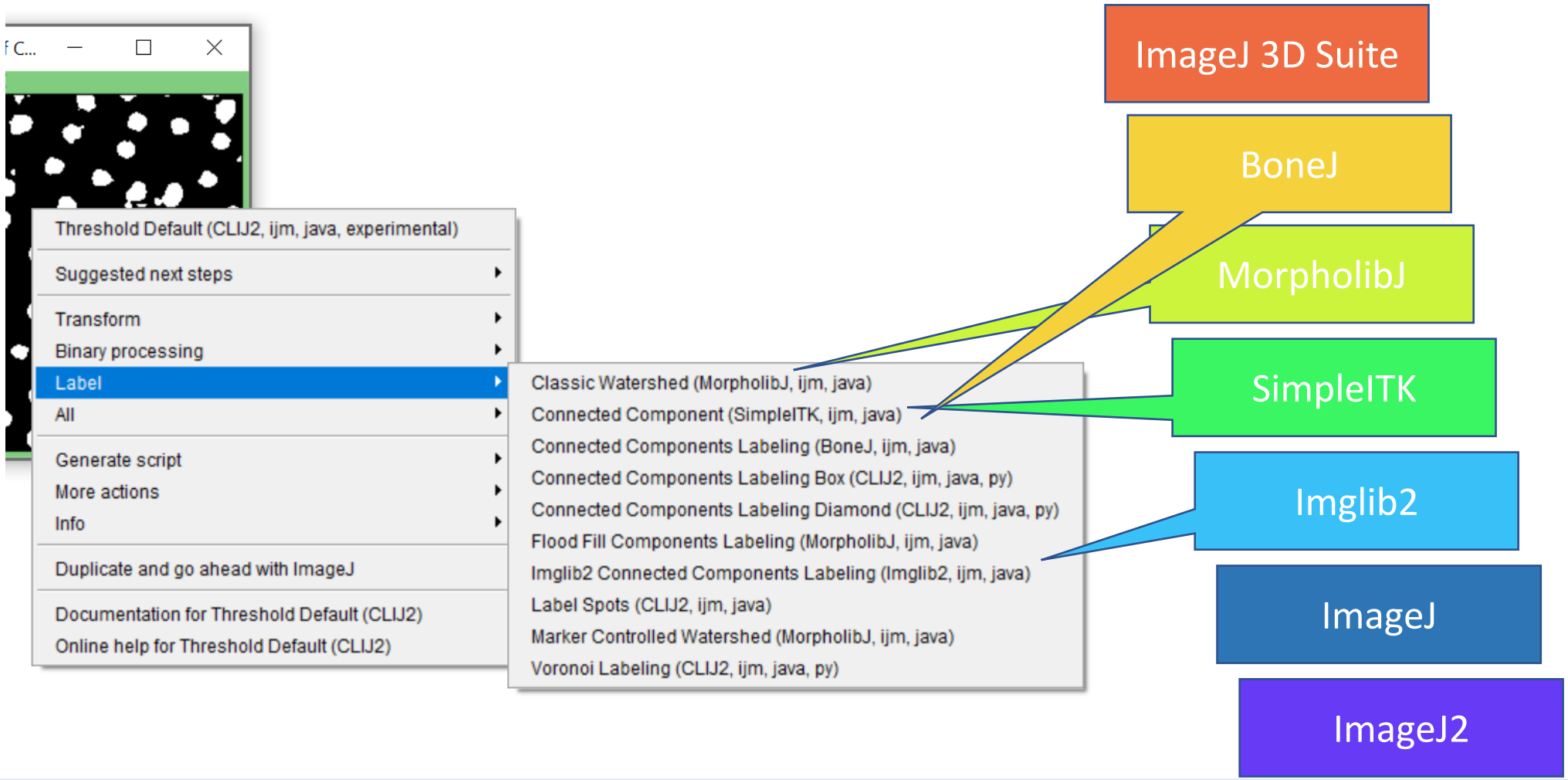
Fiji search bar

In Fijis search bar result, there is a new category: CLIJ2-assistant which offers operations



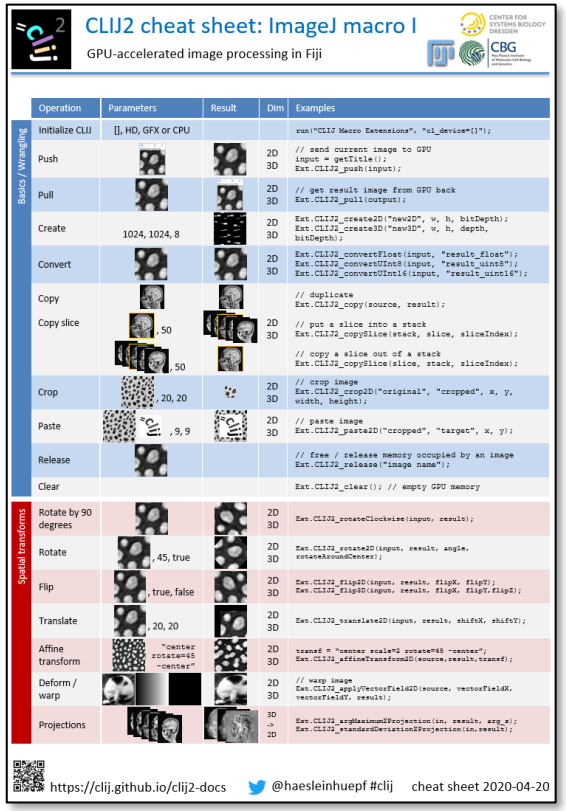
Extensibility

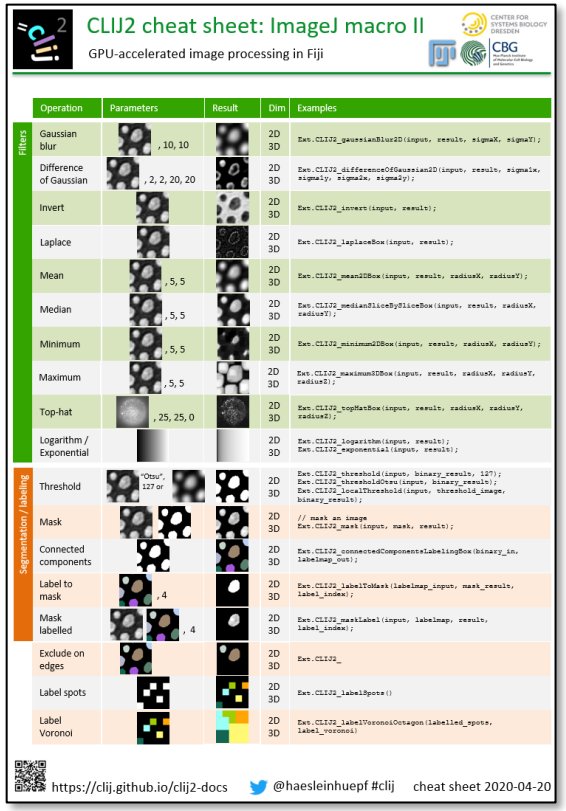
- Install: https://clij.github.io/assistant/installation#extensions

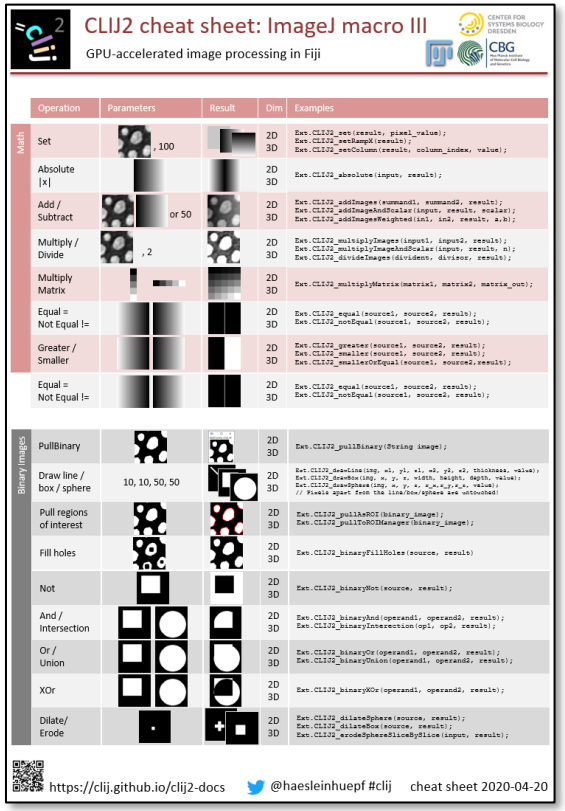


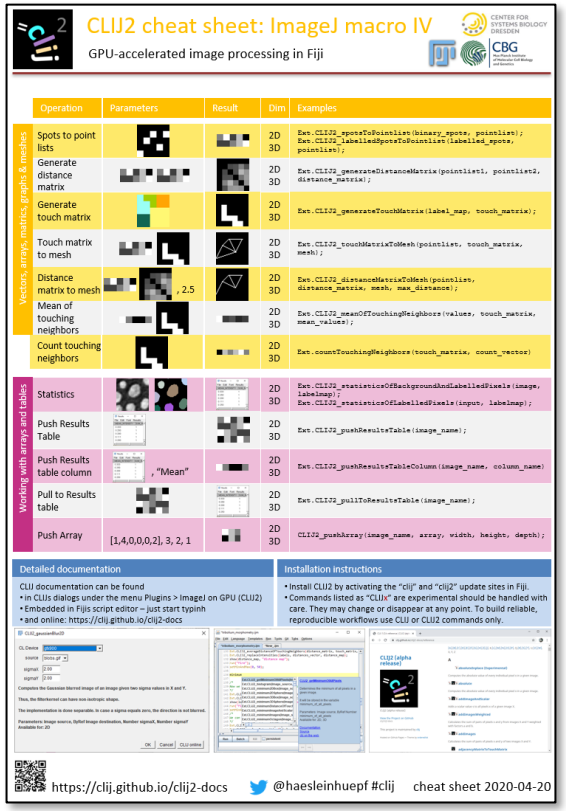
CLIJ Cheat sheets

Cheat sheets show the most important methods with input and output parameters visually.









Online support

Wei-Chen CHU weichen — Mar 25

Hi, Robert
I have a similar related question:
In case of large images that exceeds the capacity of GPU memory.
Are there any functions in CLIJ can allow lazy- loading/processing?
(Something like "Using virtual stack" in the Bioformat importer)

1 ♥ Reply

https://image.sc

Robert Haase haesleinhuepf clij & clesperanto maintainer — Mar 26

Hi @weichen ,

we had a discussion earlier about potential strategies in these threads:

> CLIJ2 Blockwise analysis ■ Usage & Issues
>
> Hey @rfrs , what kind of denoising / analysis where you thinking of? I'm also working with light sheet data a lot and we use downsampleSliceBySliceHalfMedian in routine to reduce our amount of data while imaging. To fit your data into GPU memory of your computer, you have several strategies: Process it slice by slice using pushCurrentZSlice. There is also an example macro for this. Process tiles using the methods pushTile and pullTile as also discussed in this thread. Please note that pushTil...

> Segmentation of islets in fluorescence image ■ Image Analysis
>
> Awesome! Let me know how it goes! For working with hugh images, there are two new commands in CLIJx: pushTile and pullTile which allow you to process a huge image piece wise. Here is some example code: https://github.com/clij/clij2-docs/blob/master/src/main/macro/processTiles3D.ijm for (x = 0; x < numTilesX; x++) { for (y = 0; y < numTilesY; y++) { for (z = 0; z < numTilesZ; z++) { Ext.CLIJx_pushTile(original, x, y, z, tileWidth, tileHeight, tileDepth, margin, margin, margin); Ex...

If you work with Python, you could also use dask for processing the data tile-by-tile: https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/32_tiled_image_processing/tiled_nuclei_counting_quick.html 1

Furthermore, cropping and downsampling are often alternatives for answering a scientific question without processing every single pixel.

Let us know if this helps!

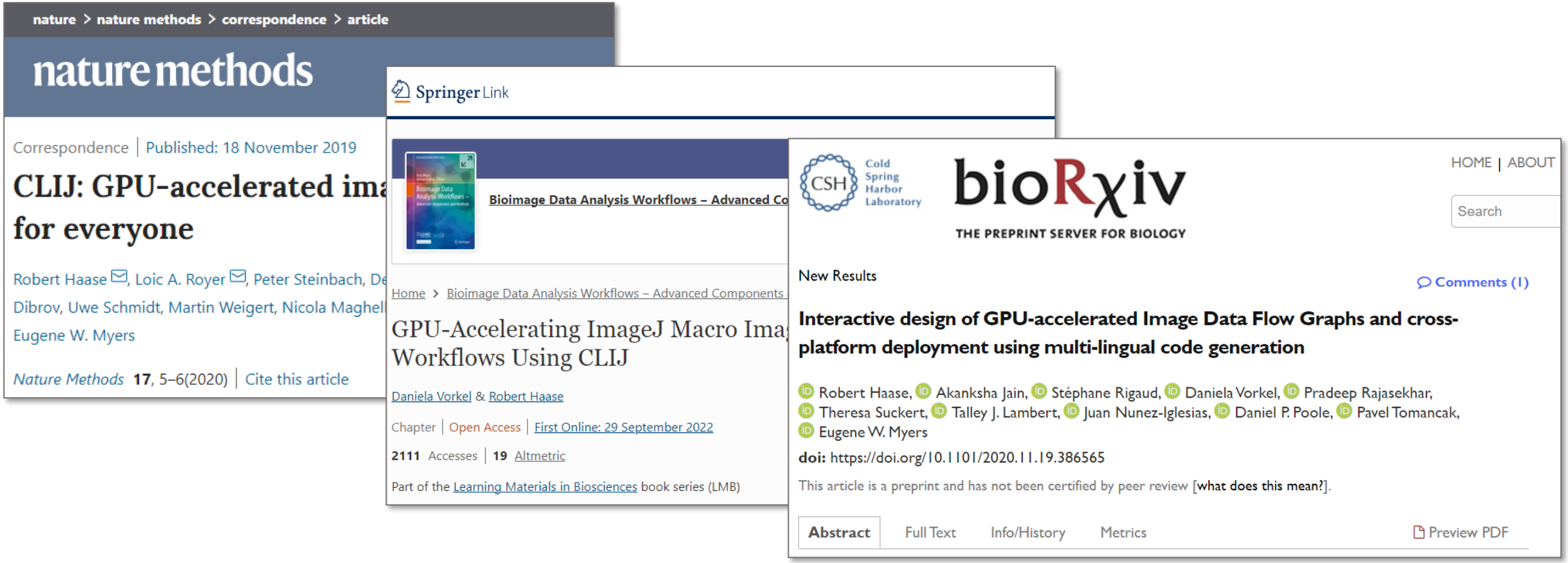
Best,
Robert

Solution 2 ♥ Reply

https://forum.image.sc/t/clij-plugin-gpu-error/79034/5

Citation of CLIJ

- If you work with CLIJ and friends, please cite the paper(s).



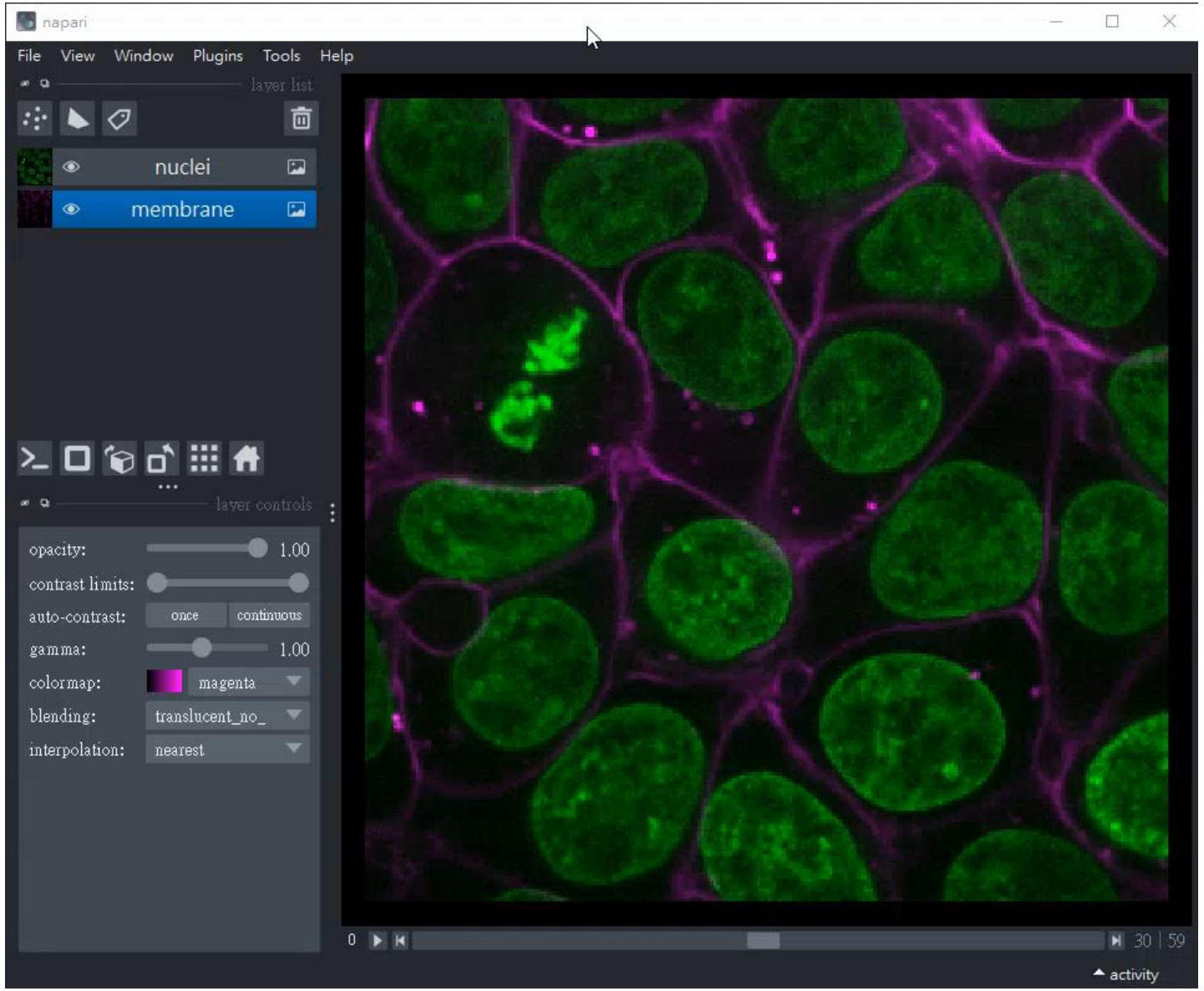
https://www.nature.com/articles/s41592-019-0650-1
https://link.springer.com/chapter/10.1007/978-3-030-76394-7_5
https://www.biorxiv.org/content/10.1101/2020.11.19.386565v1

Adapted from the learning material of Dr. Robert Haase

From FIJI to Napari

napari

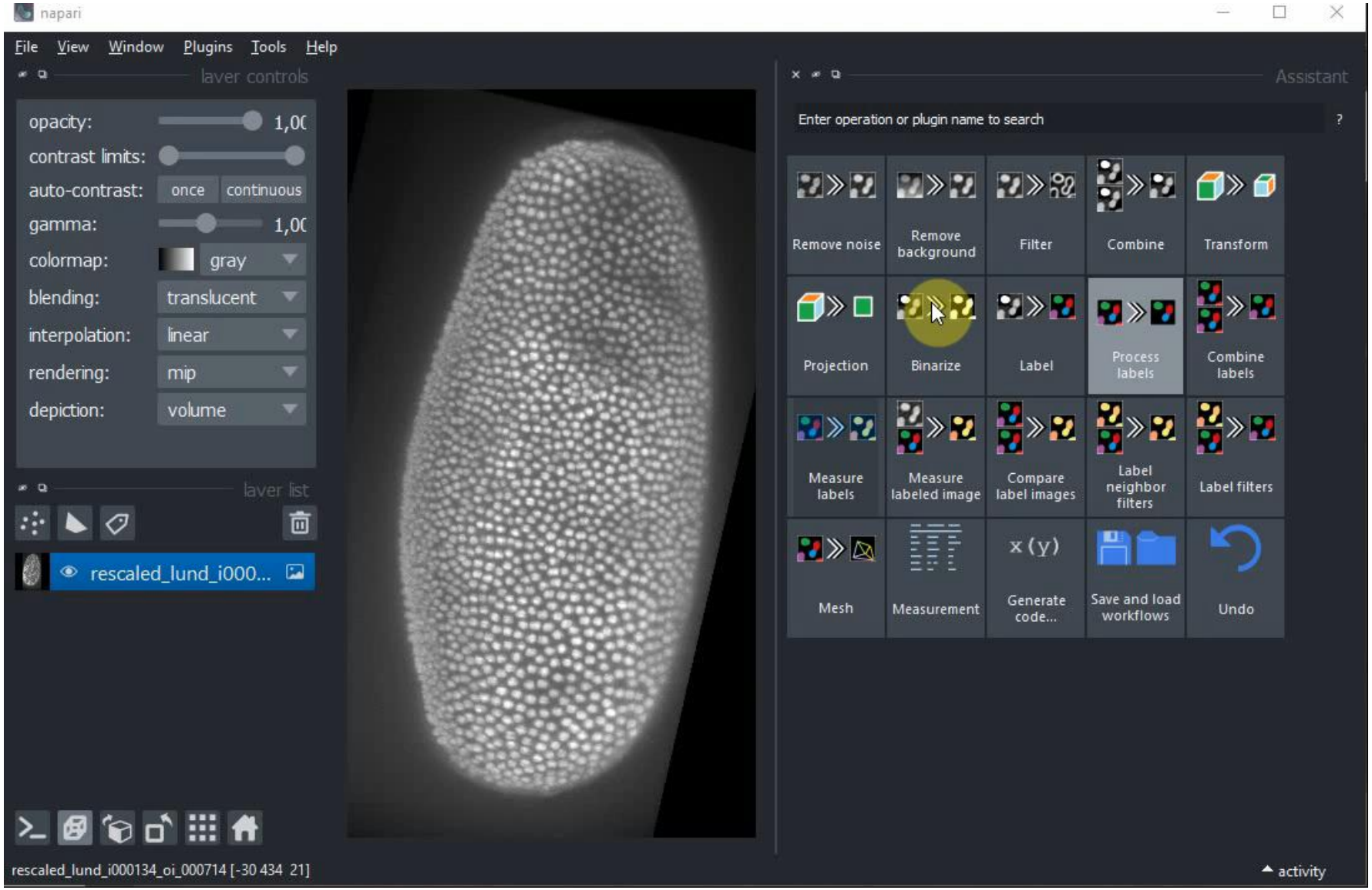
- Multi-dimensional image viewer in python
- 3D Visualization!



https://napari.org/

The Napari Assistant

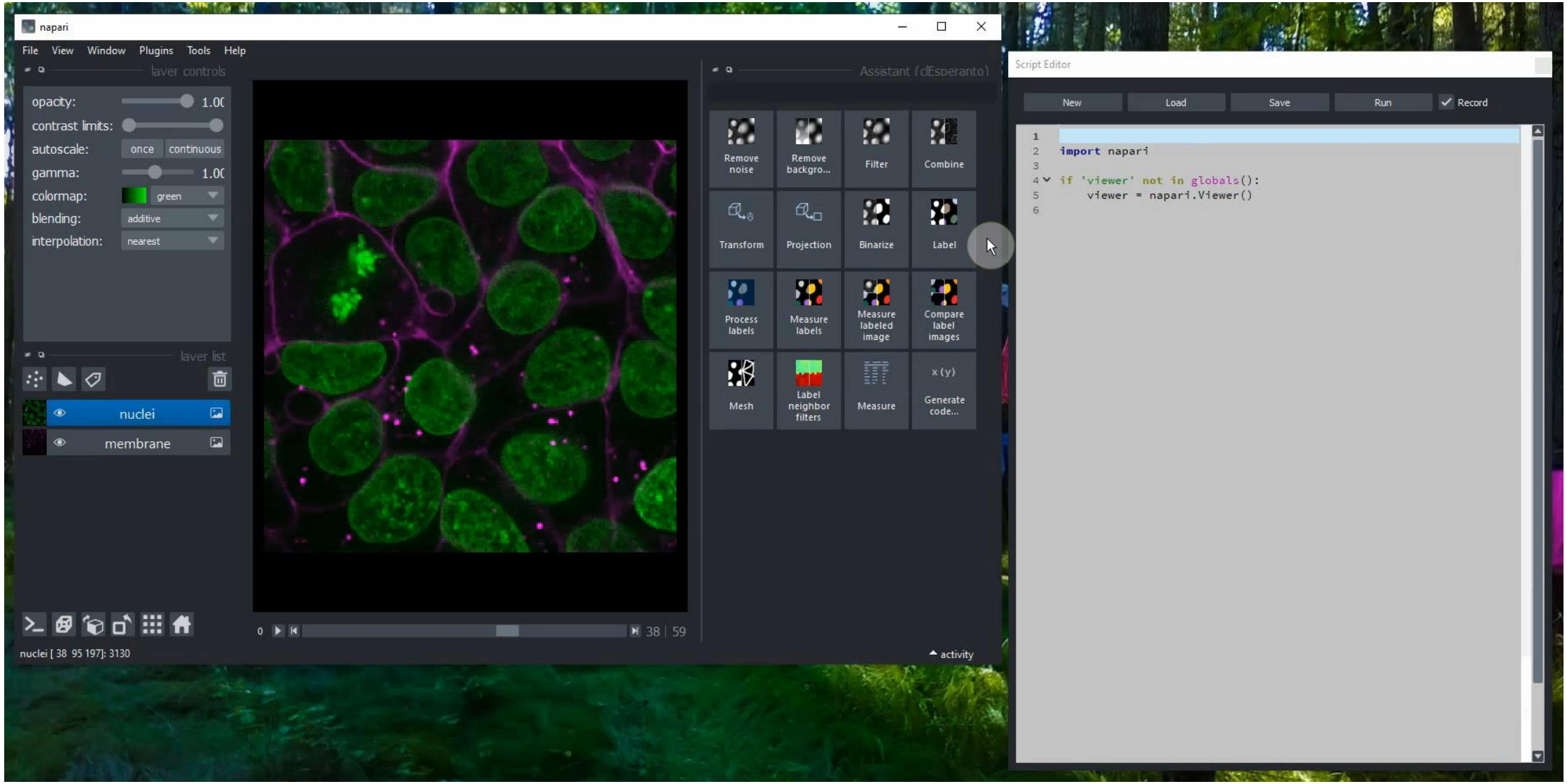
- 3D visualization
- Undo [redo]



https://www.napari-hub.org/plugins/napari-assistant

Adapted from the learning material of Dr. Robert Haase

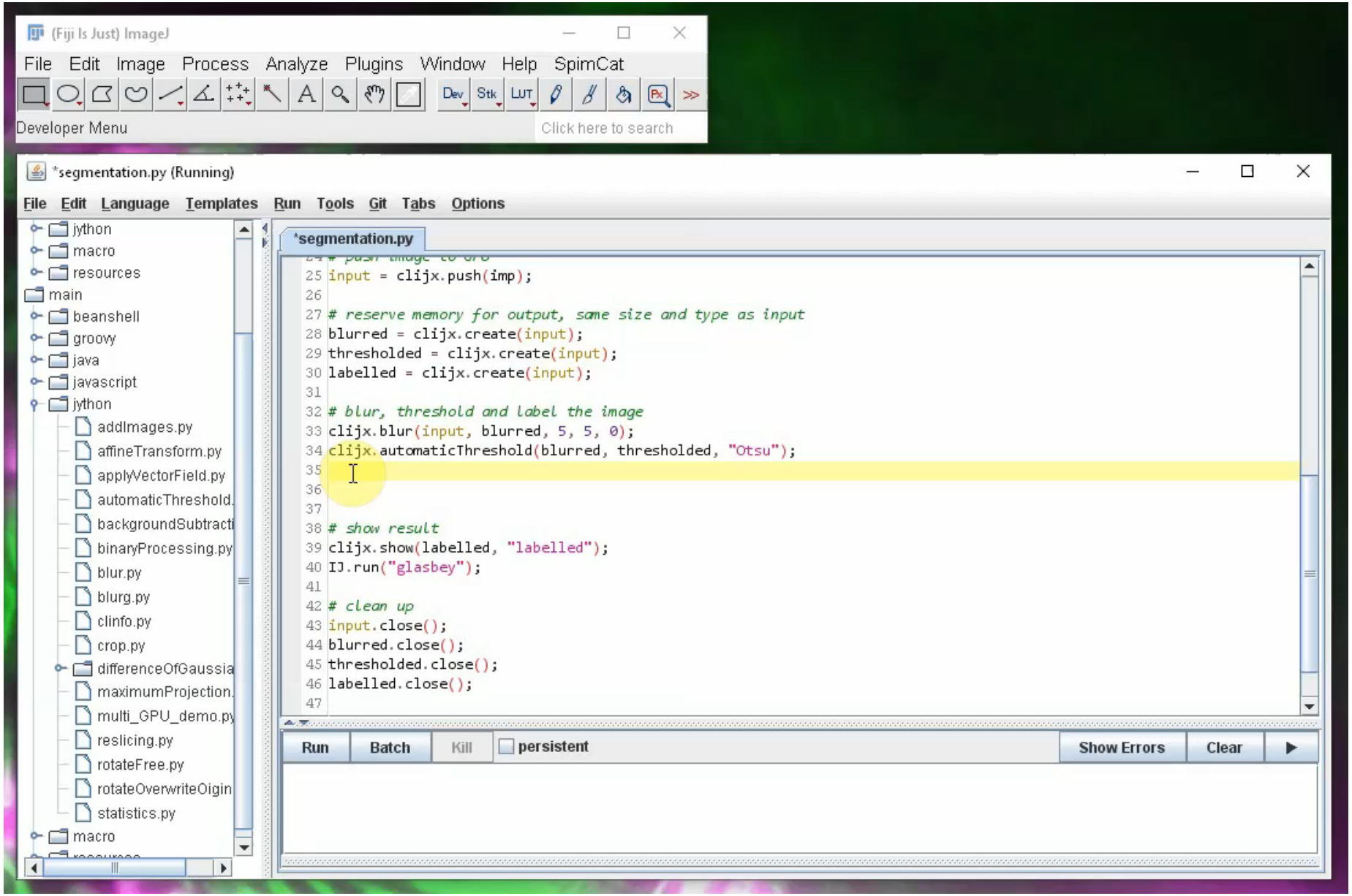
Record code instead of writing it!



clEsperanto

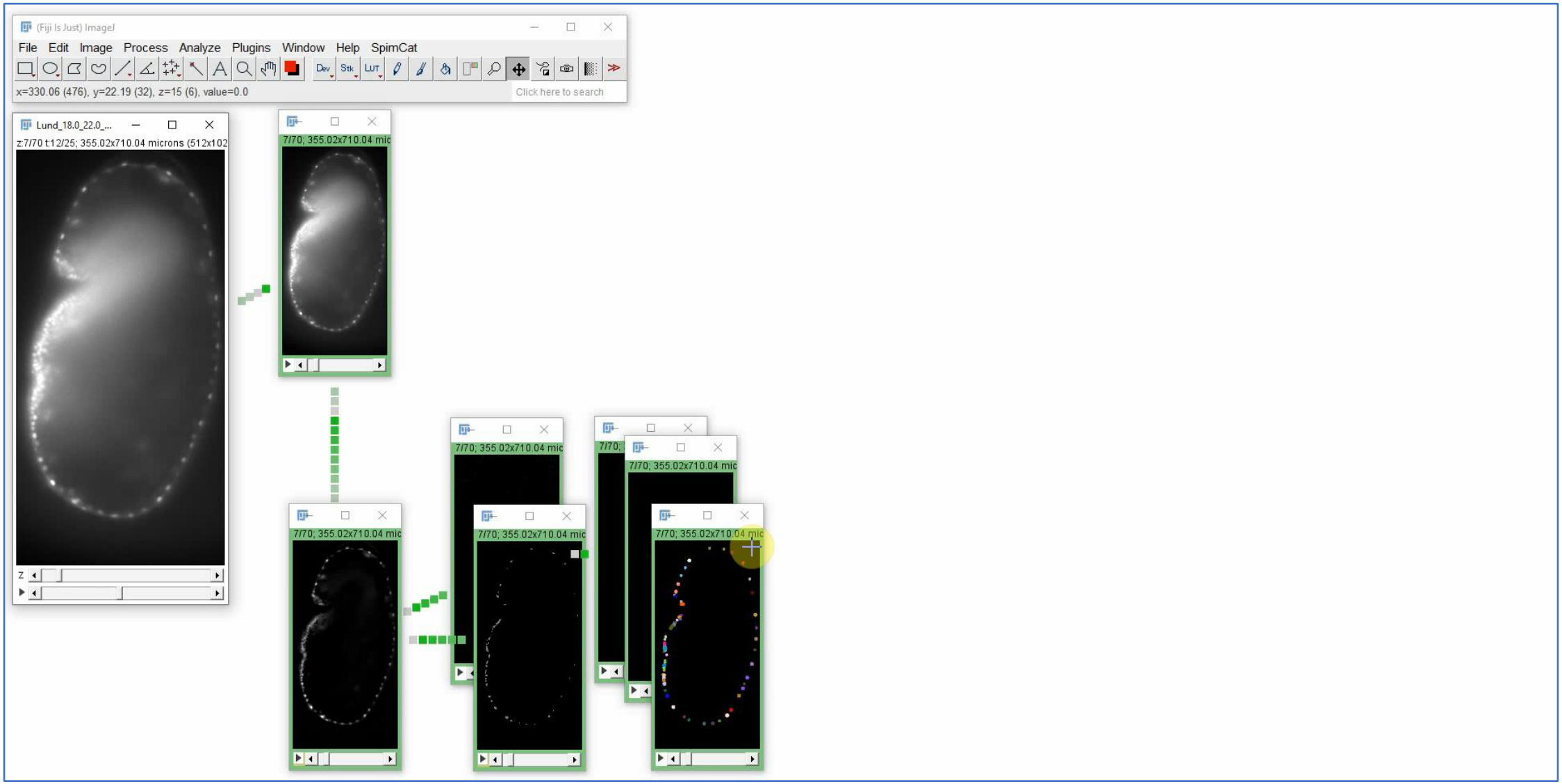
cle._

- Use the same command for Fiji, Python, Matlab, and Icy for image analysis!
- GPU- acceleration!



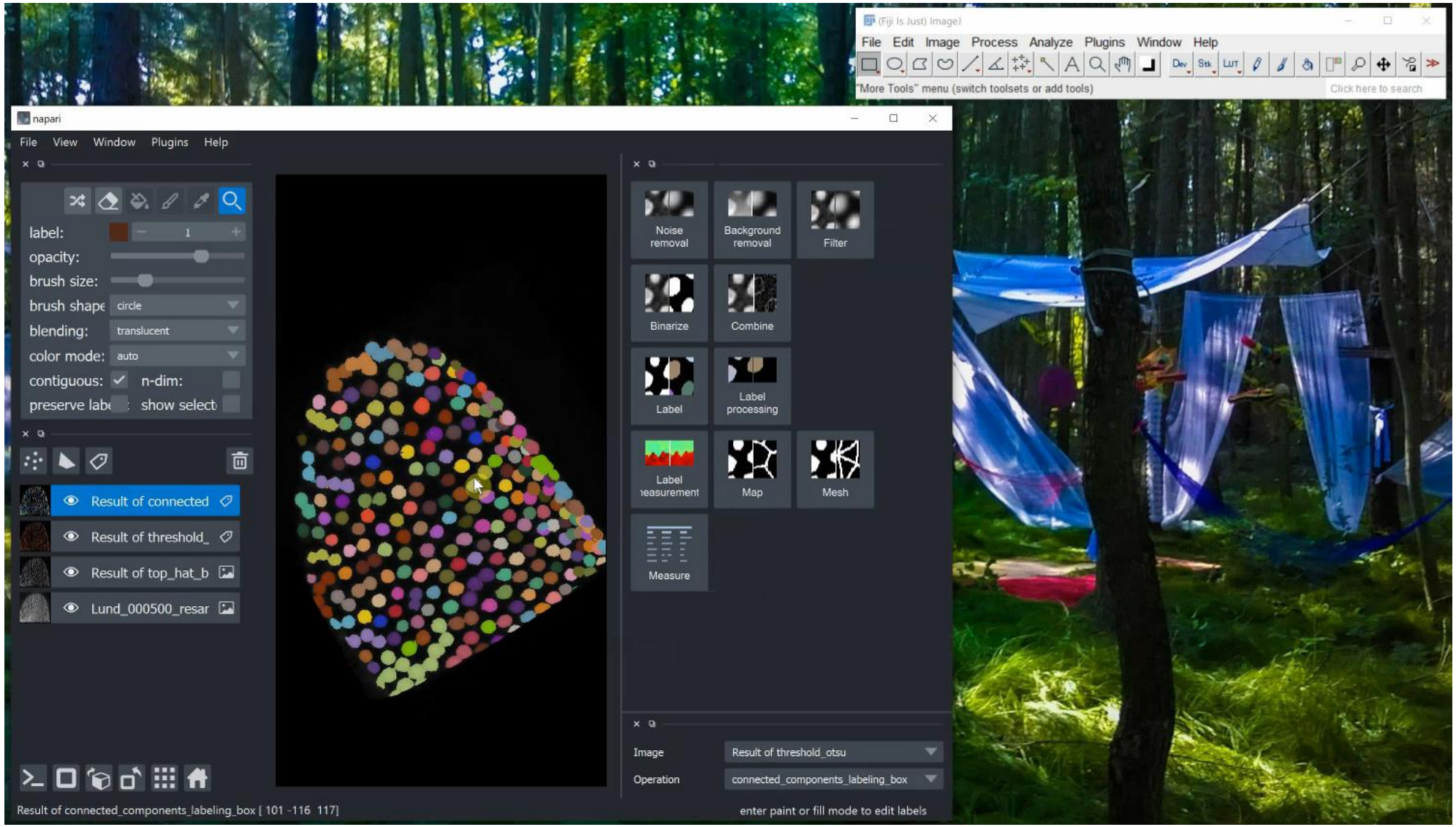
https://clesperanto.github.io/

From Fiji to napari



Adapted from the learning material of Dr. Robert Haase

From Napari to Fiji



Checklist

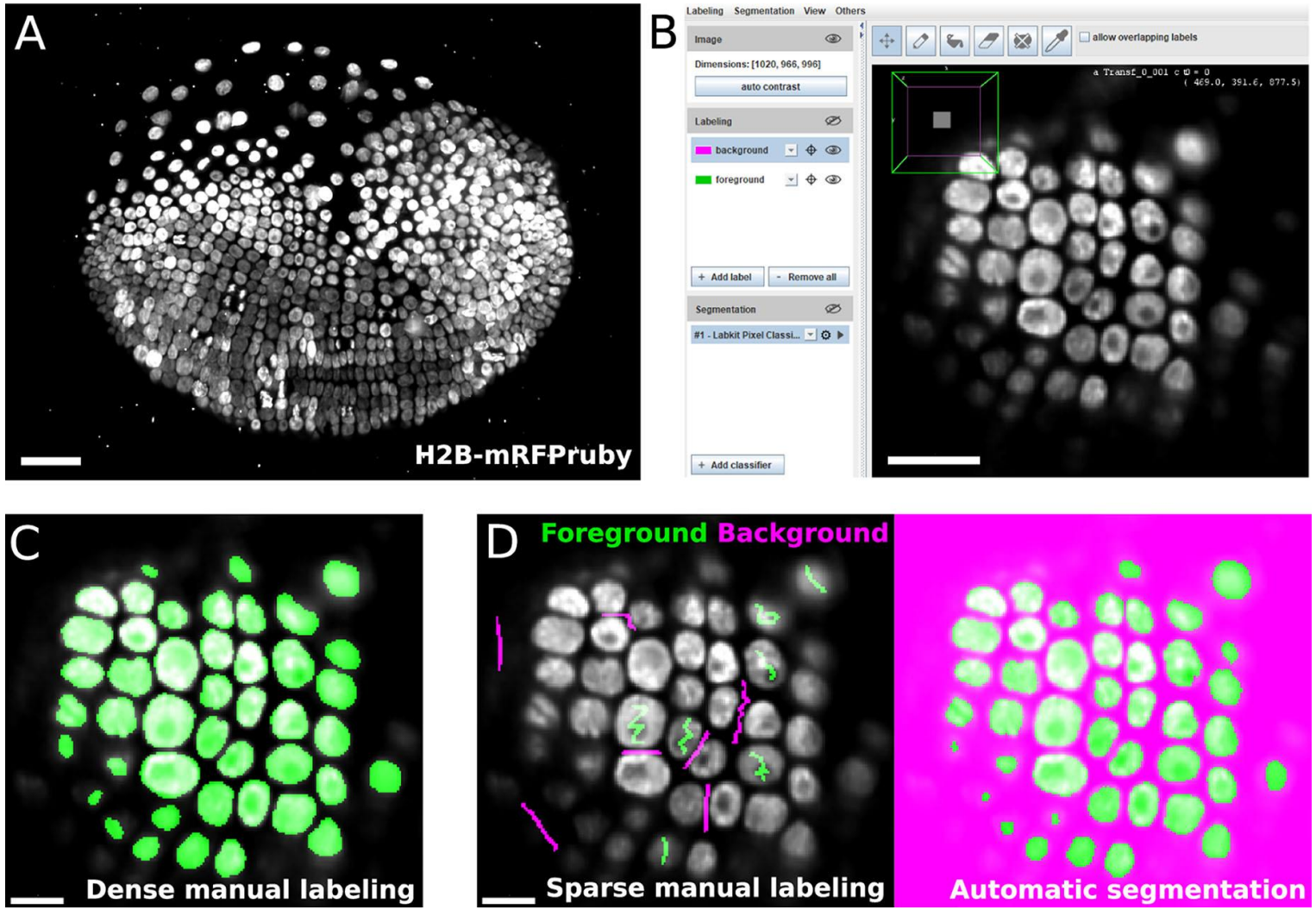
Continue using Fiji if, …

- **it does the job you need**.
- you don’t speak python (yet).
- you want to do [cell-]**tracking** or **image stitching**.
- work with **big data**
- you’re ok with just applying advanced deep learning models
- want to keep things **simple**. (especially installation)

Try out Napari if

- you work with **3D** (+time, channel) data
- your data fits in memory
- you want to use **machine learning** for analyzing image/object data (also: ilastik)
- tools you need are listed on the **napari hub**
- you speak python, or
- you want to **learn python**!

Pixel classification by LABKIT



- GPU-acceleration by CLIJ (OpenCL)
- Integrate with Imaris (Since Ver 9.9)

https://imaris.oxinst.com/learning/view/article/how-to-use-labkit-with-imaris

- Can handle with big data!

https://www.youtube.com/watch?v=AedreoUkrUU&t=19s

https://doi.org/10.3389/fcomp.2022.777728

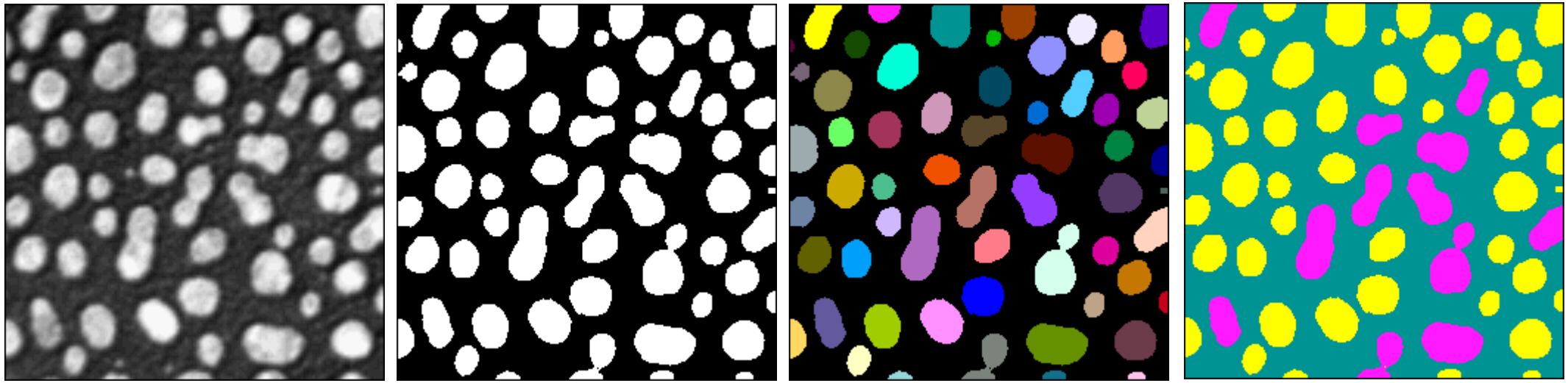
Pixel and Object classification by iLastik

- GPU-acceleration (Nvidia CUDA)
- Can handle with big data!

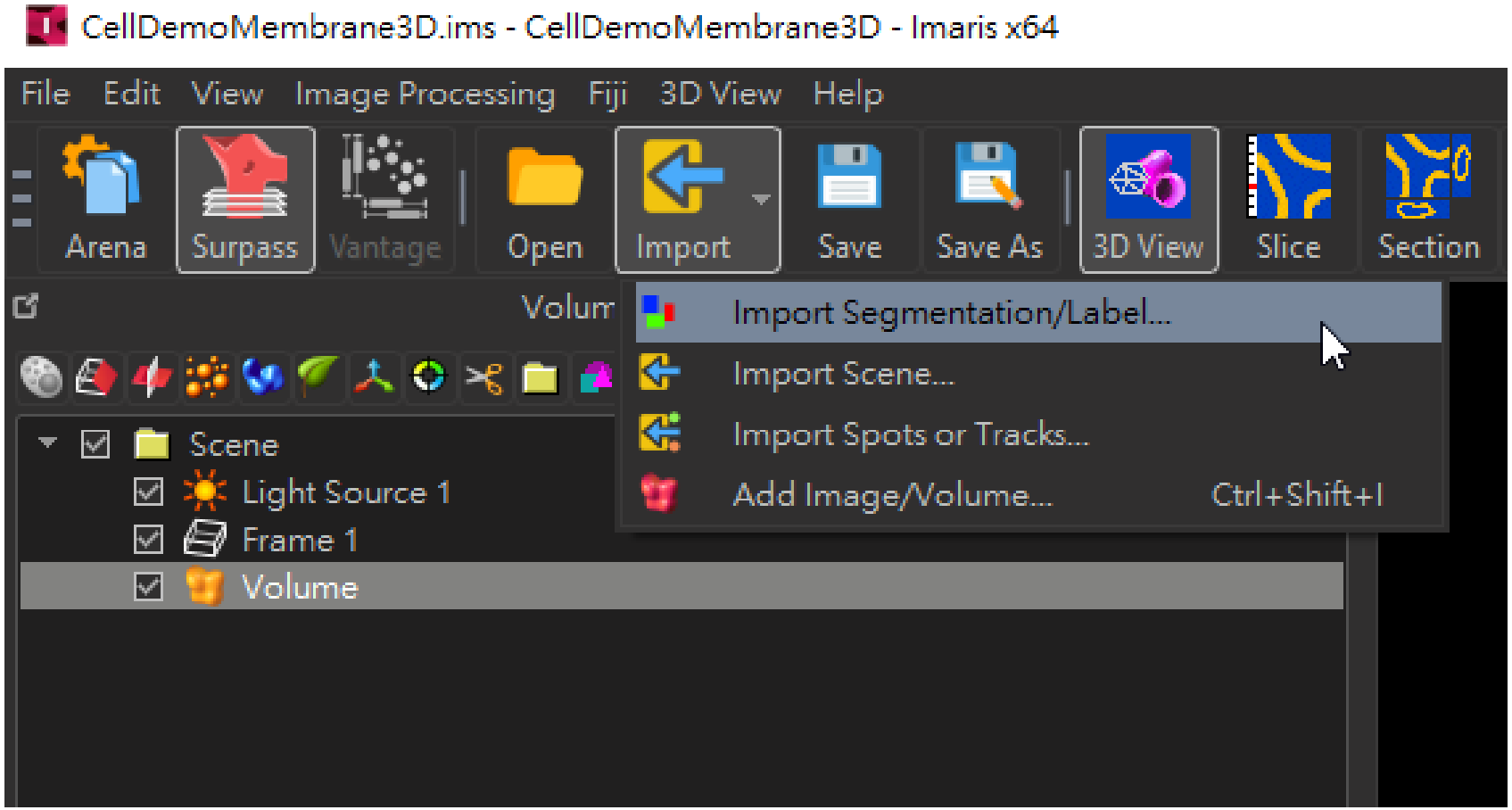
https://www.youtube.com/watch?v=AedreoUkrUU&t=19s

iLastik

Segmentation **Labeling** **Object classification**

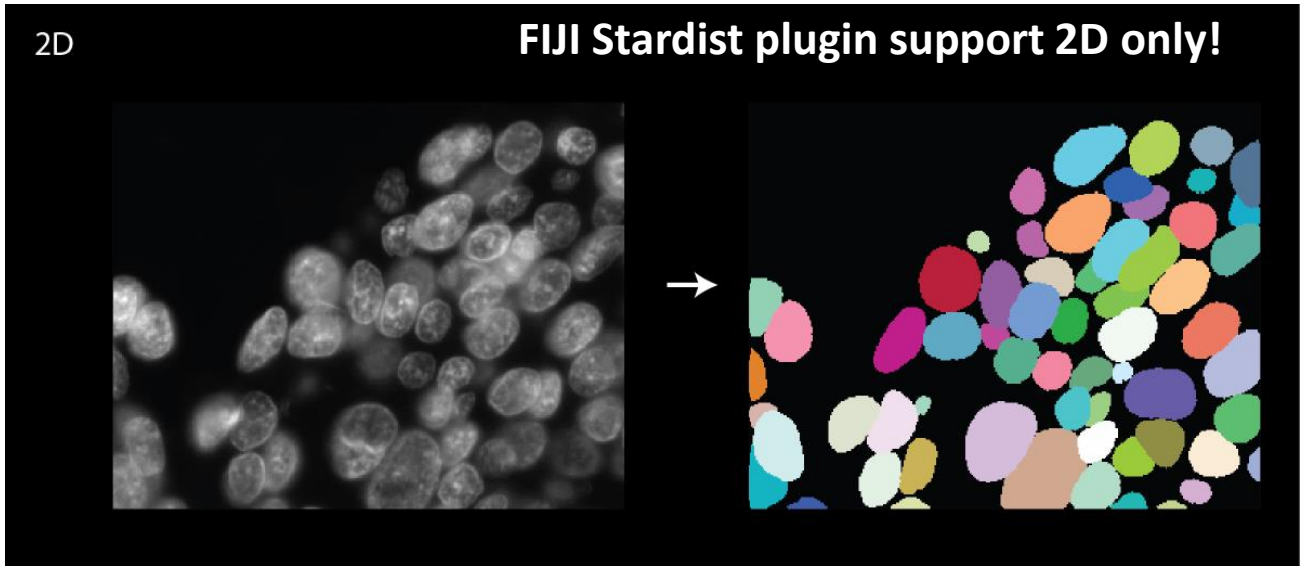


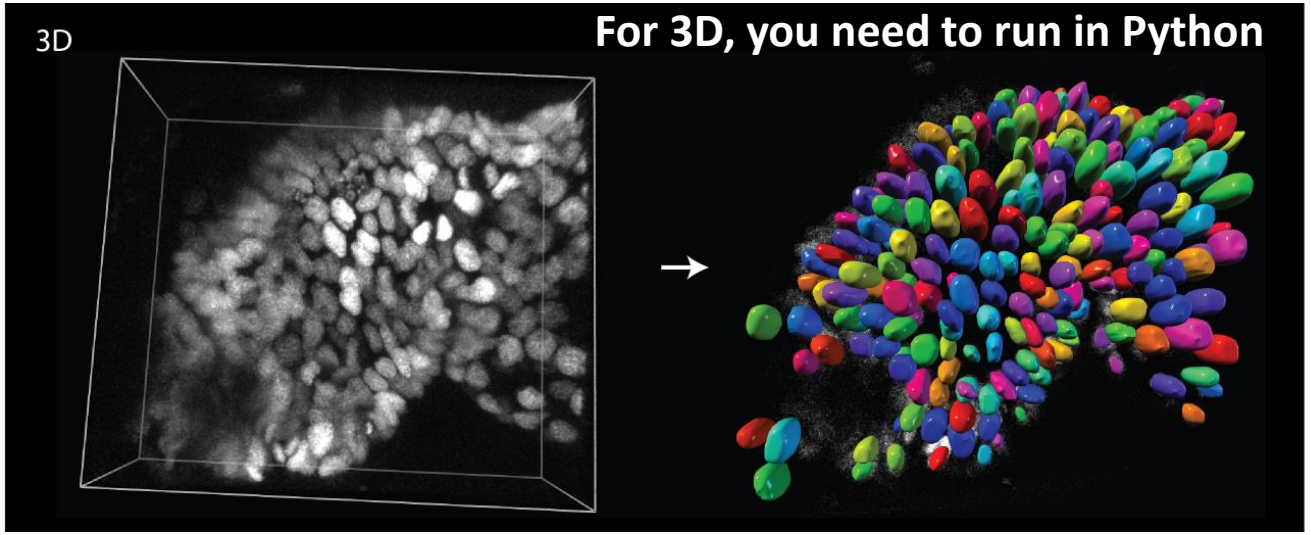
Mask /label can import to Imaris as a “Surface” (since Ver 9.9)



https://imaris.oxinst.com/learning/view/article/how-to-import-label-images-from-ilastik

Stardist





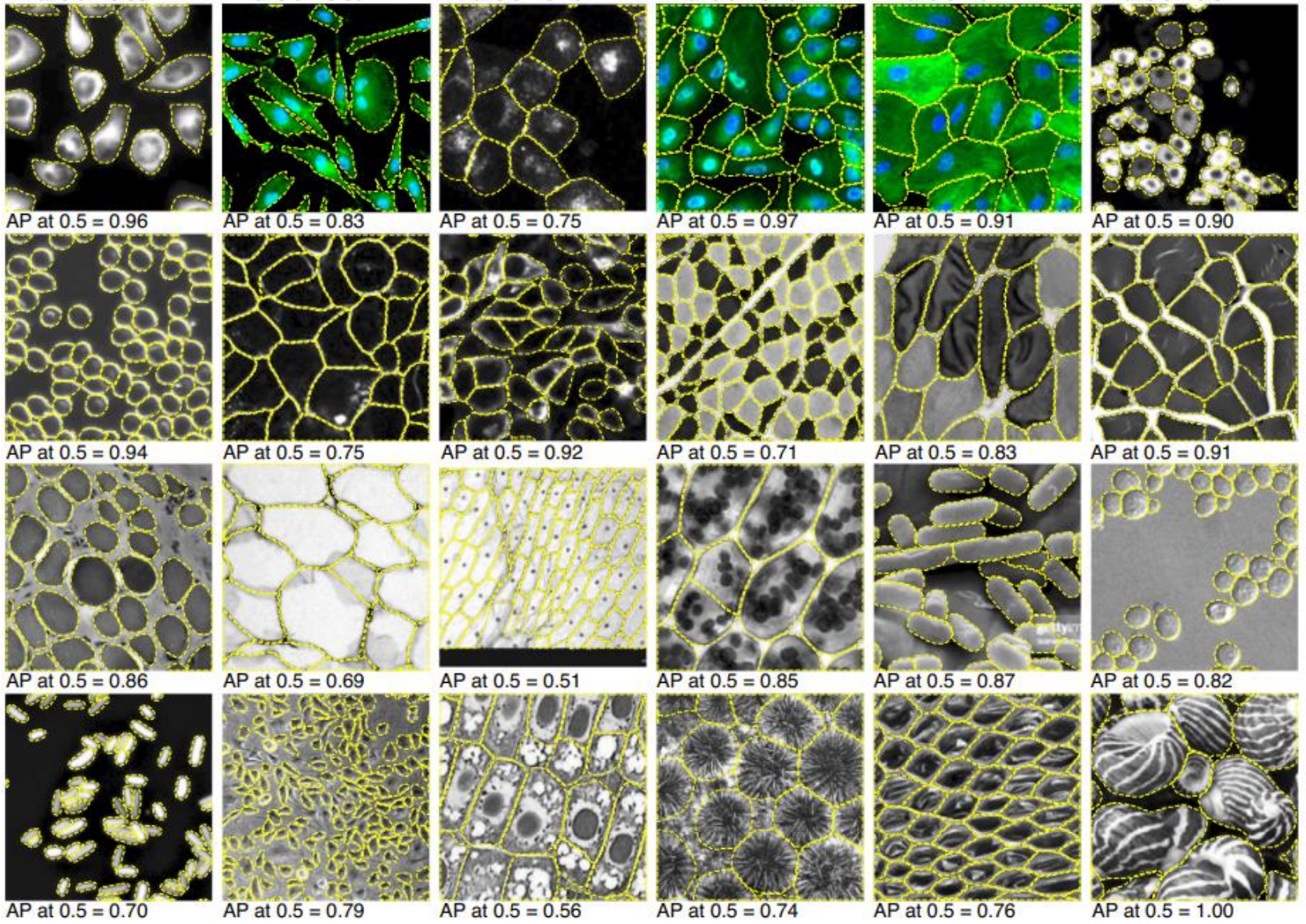
- Useful for nuclei segmentation (or something similar)
- GPU-acceleration (Nvidia CUDA)

Webinar:
https://www.youtube.com/watch?v=Amn_eHRGX5M

https://github.com/stardist/stardist

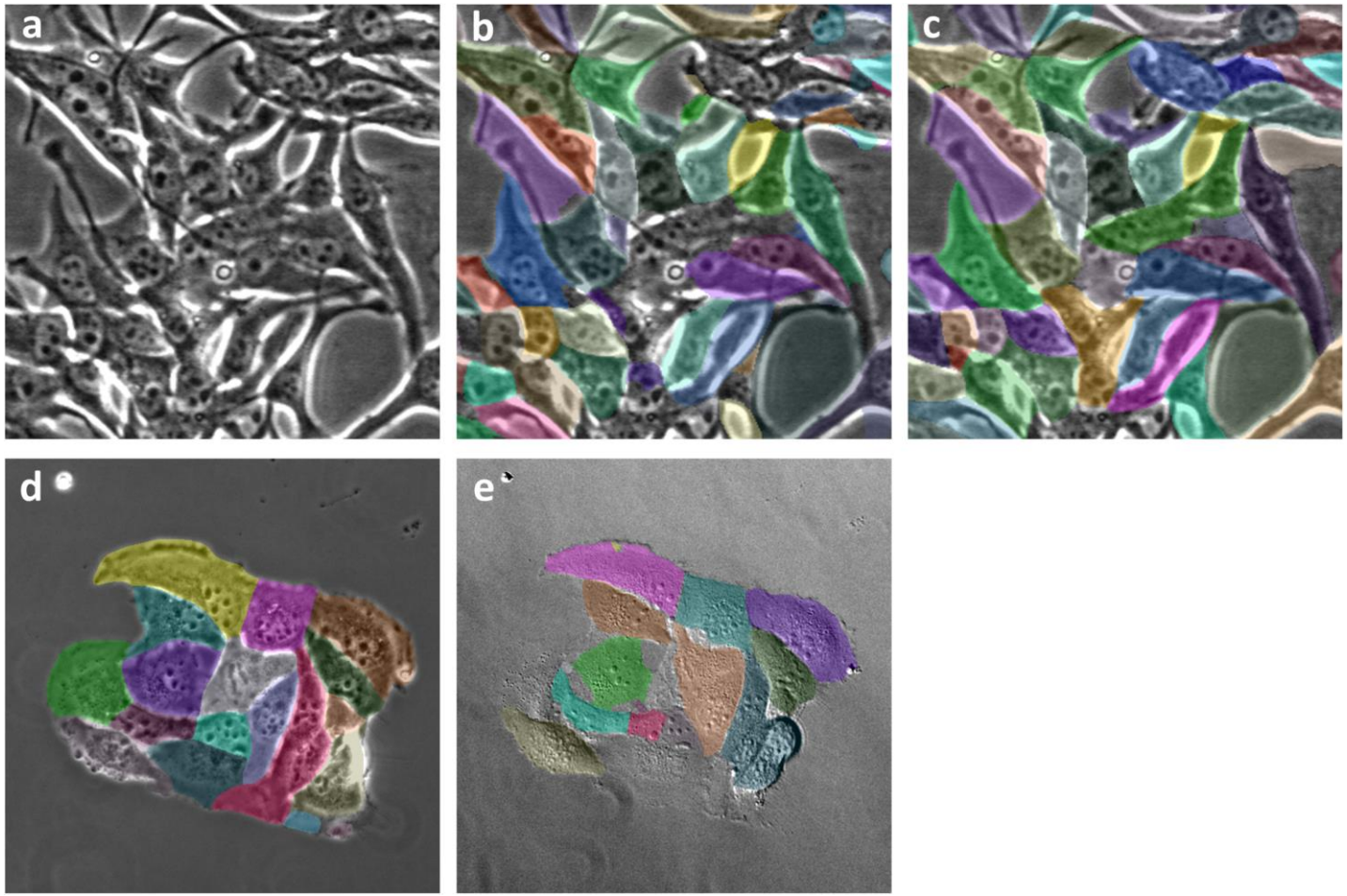
Cellpose/ Cellpose 2

- Deep Learning segmentation tools
- GPU-acceleration (Nvidia CUDA)



https://github.com/MouseLand/cellpose

Cellpose is a great tools for BF, Ph, DIC segmentation



My workflow:

Image of Ph/DIC channel

↓ Cellpose
(Train for your model is better)

Segmentation Image file

↓ pullLabelsToROIManager (CLIJ2)

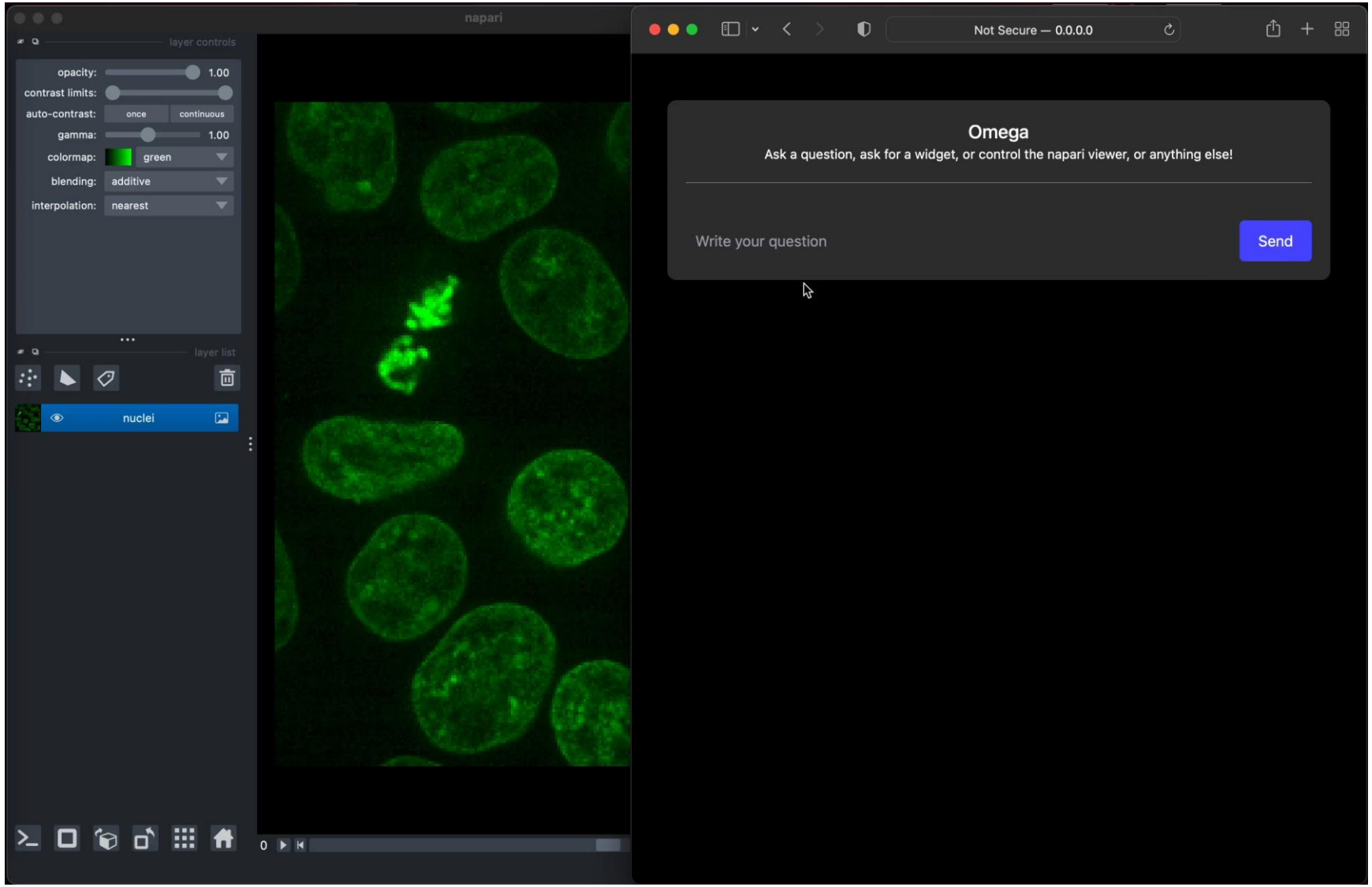
ROI List

↓

Measure on the fluorescence channel

https://github.com/MouseLand/cellpose

Napari-chatgpt (OMEGA)



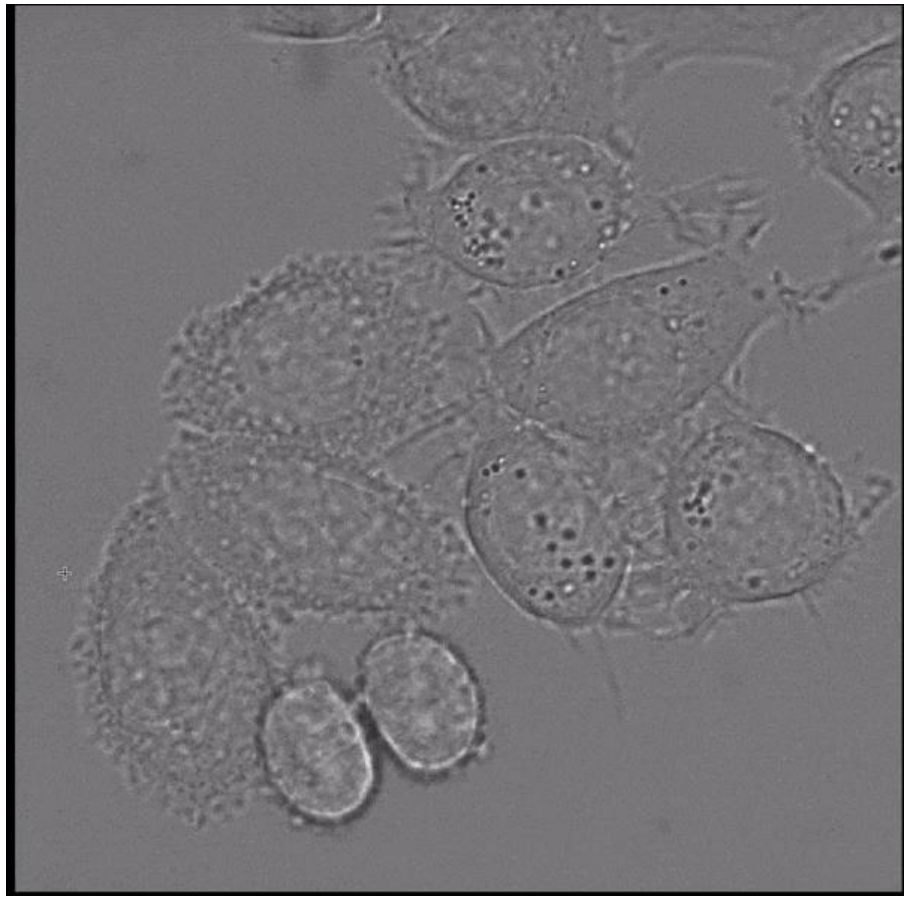
https://github.com/royerlab/napari-chatgpt

Segment Anything Model (SAM) for Microscopy

Meta AI

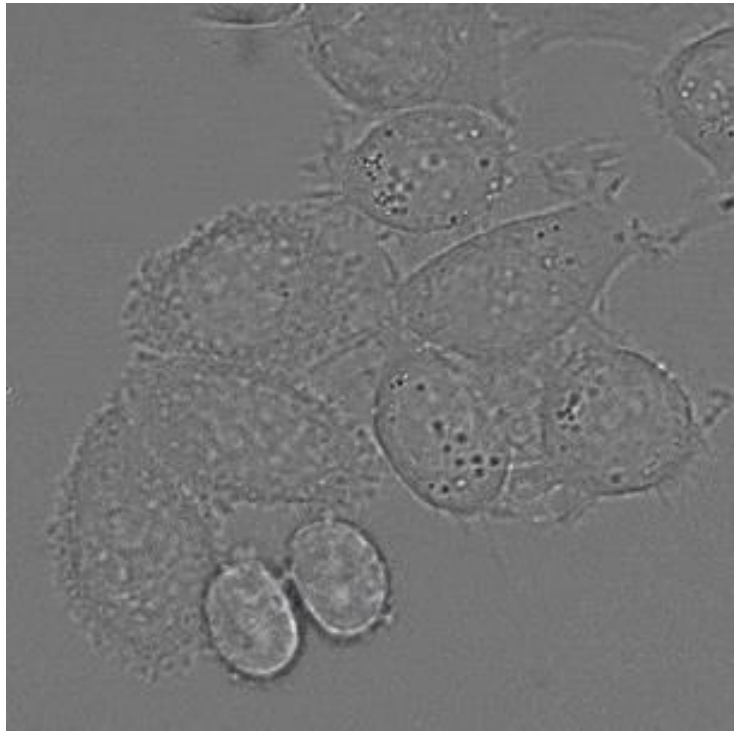
https://github.com/facebookresearch/segment-anything

https://arxiv.org/abs/2304.02643

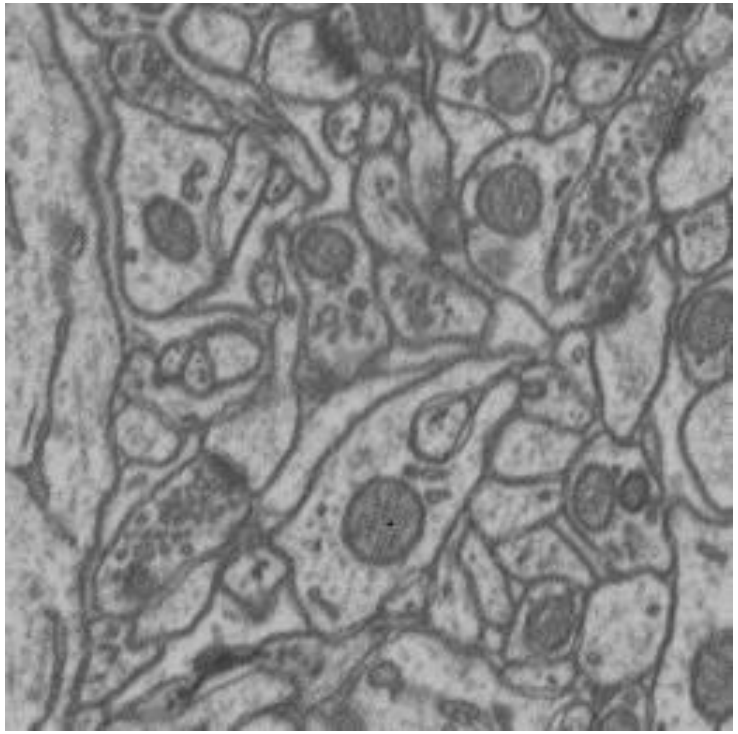


https://github.com/computational-cell-analytics/micro-sam

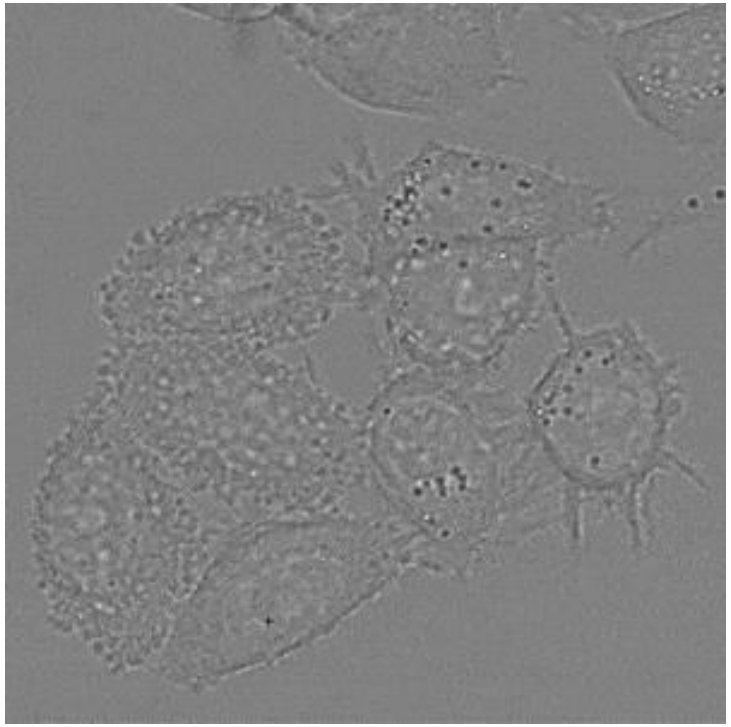
Segment Anything Model (SAM) for Microscopy



2D



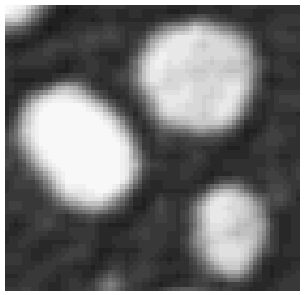
3D



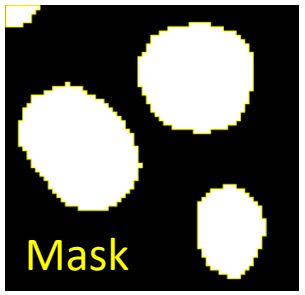
2D + Tracking

https://github.com/computational-cell-analytics/micro-sam

Segmentation with ROI list, Mask, and Label

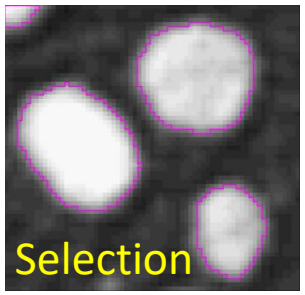


Thresholding



Create Selection
(Whole mask area selection)

Create Mask



Analysis Particle

Add to ROI Manager
Split

Select ROIs

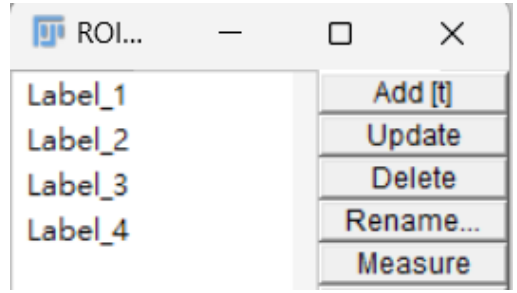
Analyze Particle
Connected Component
Labeling (CLIJ2)

Thresholding

AI-based segmentation tools

Label to ROI (BIOP)
pullLabelsToROIManager (CLIJ2)

ROI to Label (BIOP)



This slide is important!

Acknowledgements

Dr. Robert Haase
"Physic of Life" (PoL),
TU Dresden, Germany

Dr. Peter Bankhead
University of Edinburgh, UK

neubias
network of european
bioimage analysts

http://eubias.org/