An introduction to image analysis using ImageJ

Mark Willett, Imaging and Microscopy Centre,
Biological Sciences, University of Southampton.

Pete Johnson, Biophotonics lab,
Institute for Life Sciences University of Southampton.

“Raw Images, regardless of their aesthetics, are generally qualitative and therefore may have limited scientific use”.

“We may need to apply quantitative methods to extrapolate meaningful information from images”.

Examples of statistics that can be extracted from image sets

- **Intensities** (FRET, channel intensity ratios, target expression levels, phosphorylation etc).
- **Object counts** e.g. Number of cells or intracellular foci in an image.
- **Branch counts and orientations** in branching structures.
- **Polarisations and directionality**
- **Colocalisation of markers between channels** that may be suggestive of structure or multiple target interactions.
- **Object Clustering**
- **Object Tracking** in live imaging data.

Regardless of the image analysis software package or code that you use…..

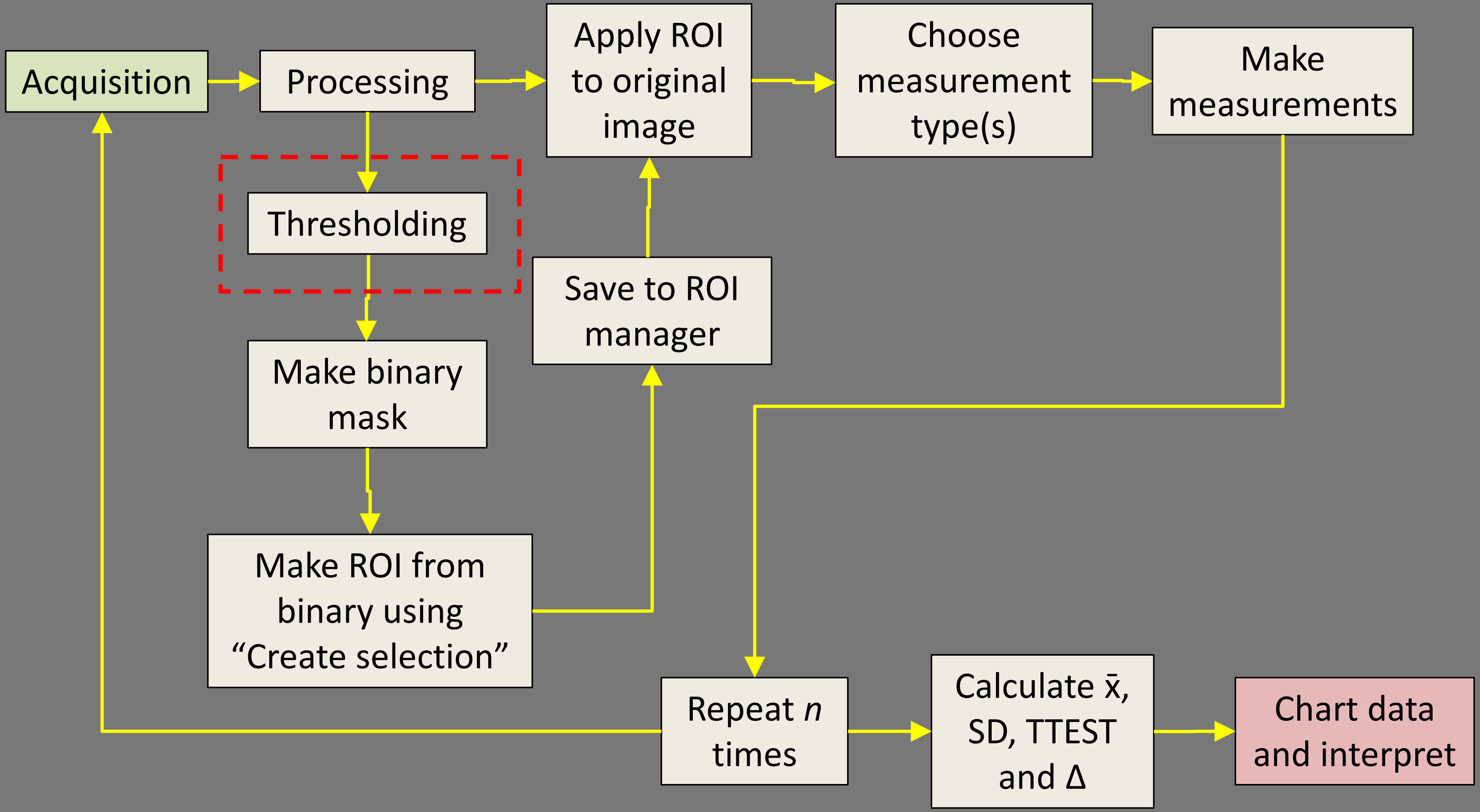
- ImageJ, Fiji, Matlab, Volocity and IMARIS apps.
- Java and Python coding languages.

….image analysis comprises of a workflow of predefined functions

which can be native, user programmed, downloaded as plugins or even used between apps.

This is much like a flow diagram or computer code.

Here’s one example of an image analysis workflow:


Acquisition
Processing
Apply ROI to original image
Choose measurement type(s)
Make measurements
Thresholding
Save to ROI manager
Make binary mask
Make ROI from binary using “Create selection”
Repeat *n* times
Calculate x̄, SD, TTEST and Δ
Chart data and interpret

A few example Functions that can inserted into an image analysis workflow. You can mix and match them to achieve the analysis that you want.

****If there is an ROI present on the image, Fiji will only execute the function on the part of the image inside the ROI****

- Automatic object detection
- Binarisation
- Image intensity Thresholding
- Mask generation
- Automatic generation of ROIs
- Skeletonization
- Vectorisation
- Object tracking
- Various colocalisation algorithms
- “Image math”

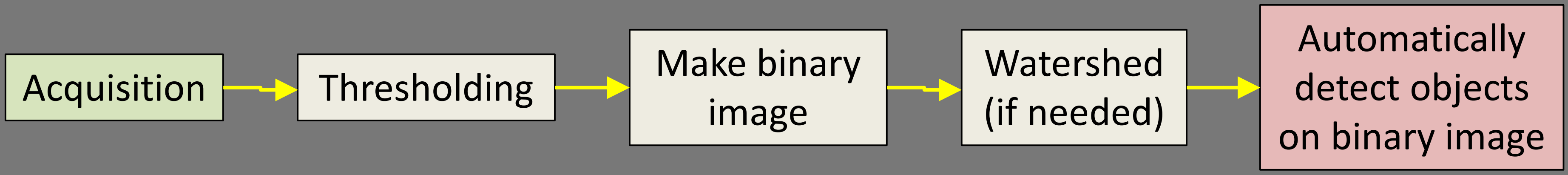
Particle selection

If you had multiple objects (cells, particles, nuclei etc) you could count them manually, but you can use imageJ/Fiji to do it for you and create masks and ROIs for measurement too.

This function enables the automatic detection of multiple objects in the image.

However, it needs the application of some other functions first. We need a binary (black and white only) image to help the particle analysis function detect the objects.

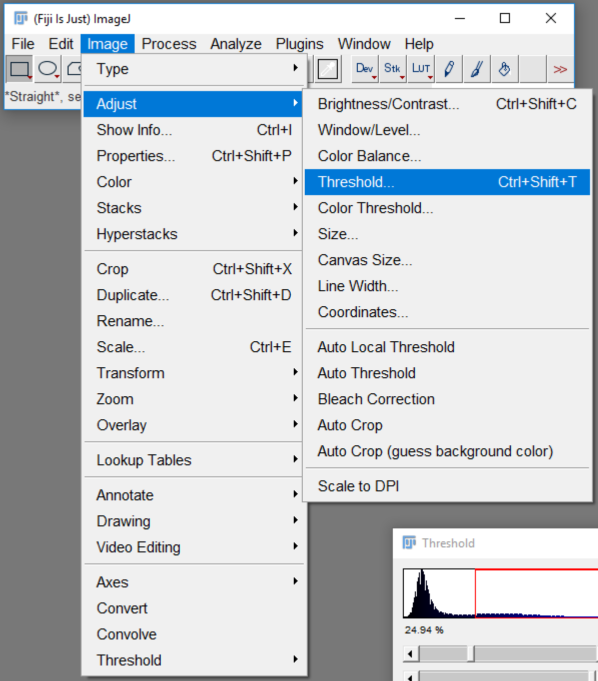
To tell Fiji which pixel grey values to make white and which ones black on the binary image, we need to threshold the image. Values above and below the selected threshold will be sent to either black or white.

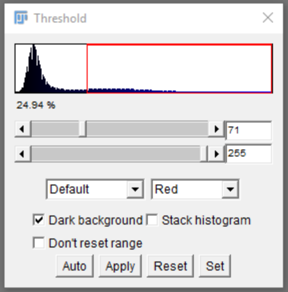


Particle selection

Open image “Nuclei”.
Select Image>Adjust>Threshold.

Tick “Dark background” and select “Default” and Red” from the pulldown menus. Using the top slider, threshold image until the nuclei are red against a dark background, then click “Apply” and then Process>binary>make binary to convert to a binary image.





24.94 %

71

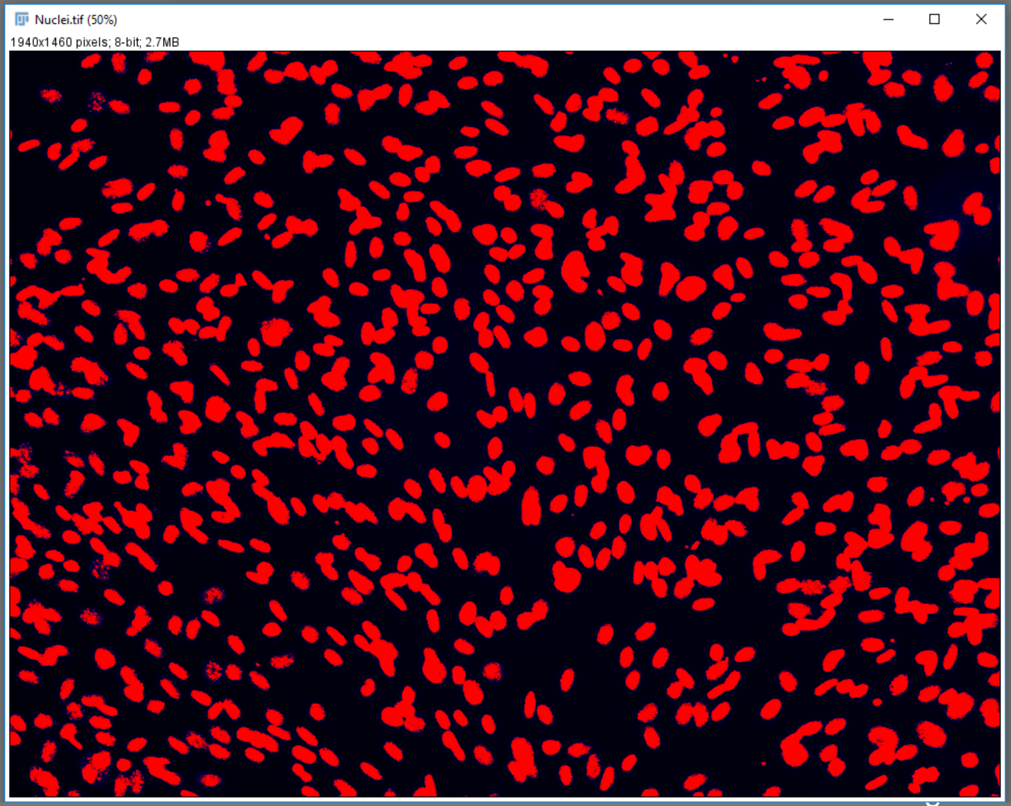
255

Default

Red

☑ Dark background ☐ Stack histogram

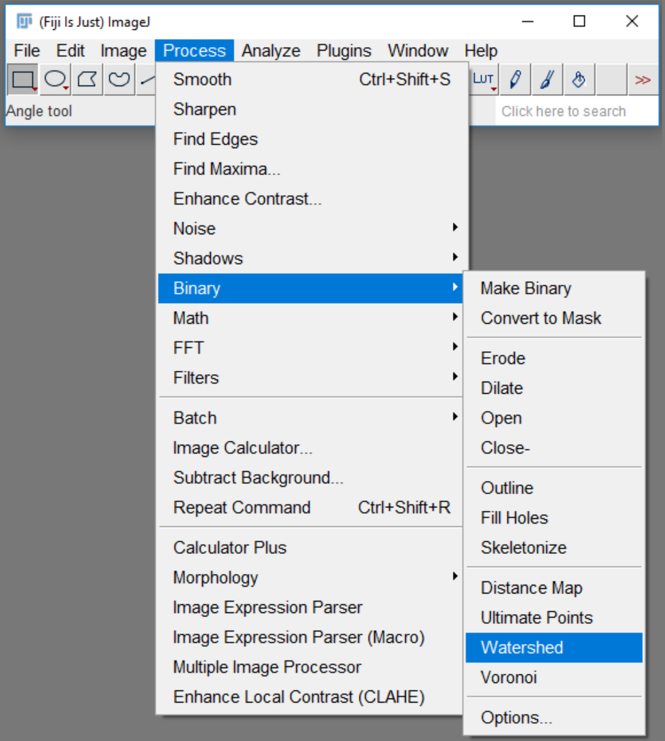
☐ Don't reset range

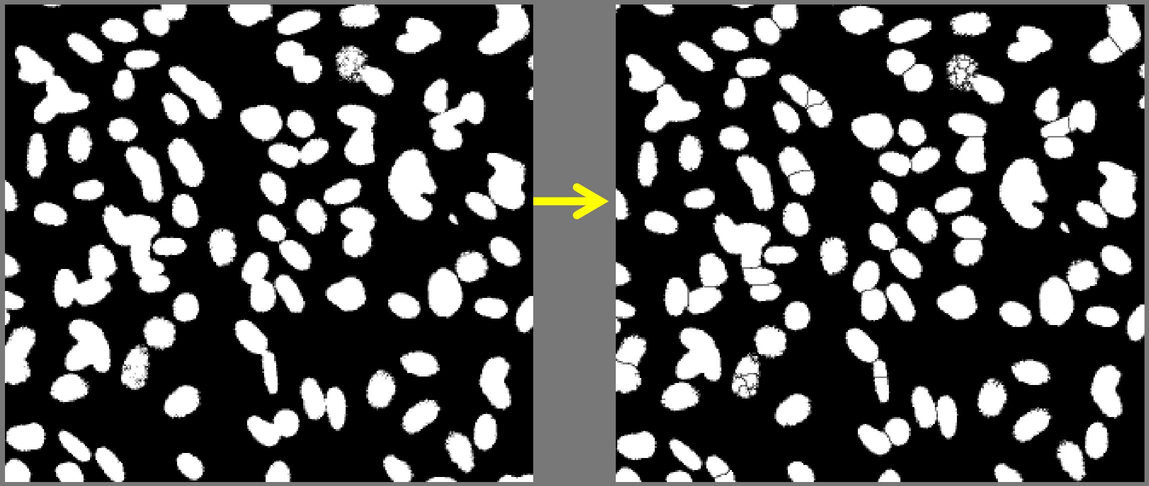


Particle selection

Select Process>Binary>Watershed.

This function attempts to separate objects based on their circularity.





Particle selection

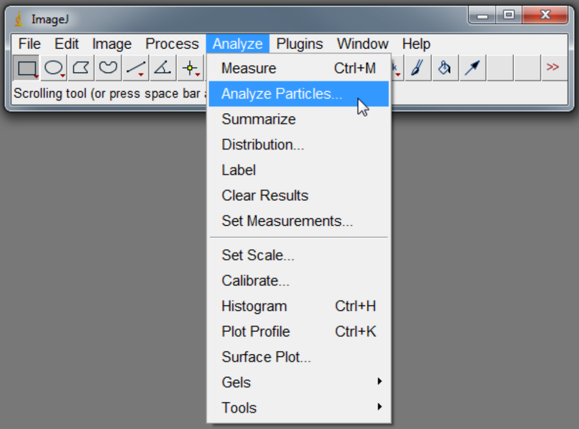
(3) Set the particle size and circularity discrimination and do the analysis

① Select Analyse>Analyse Particles.

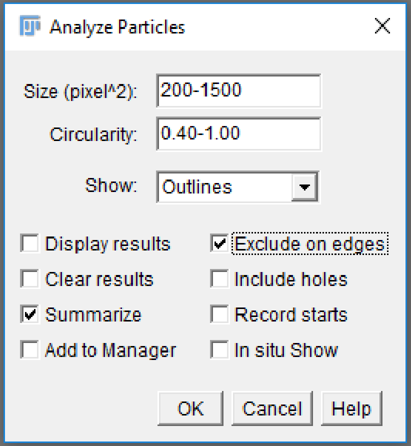
② Adjust Size to 200-1500 and Circularity discrimination to 0.40 - 1.00 and select "Show: Outlines". *Particles that are larger or less circular will be excluded*

③ Tick "Summarize" and "Exclude on edges" and click "OK", A results box should appear with a particle count and other information.

①



②


Analyze Particles

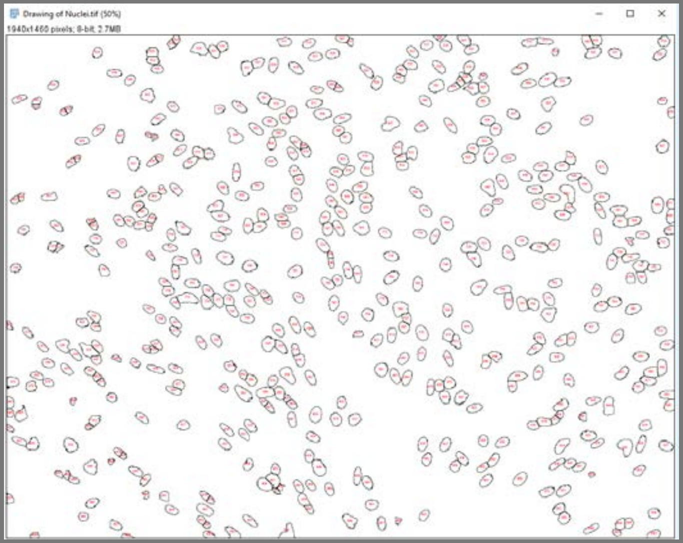
Size (pixel^2): 200-1500

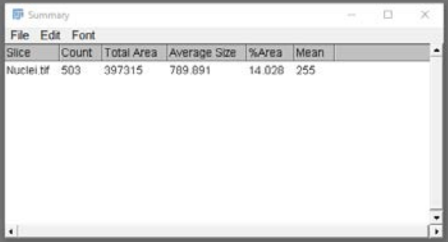
Circularity: 0.40-1.00

Show: Outlines

☐ Display results ☑ Exclude on edges
☐ Clear results ☐ Include holes
☑ Summarize ☐ Record starts
☐ Add to Manager ☐ In situ Show

③




Summary

Slice	Count	Total Area	Average Size	%Area	Mean
Nuclei.tif	503	397315	789.891	14.028	255

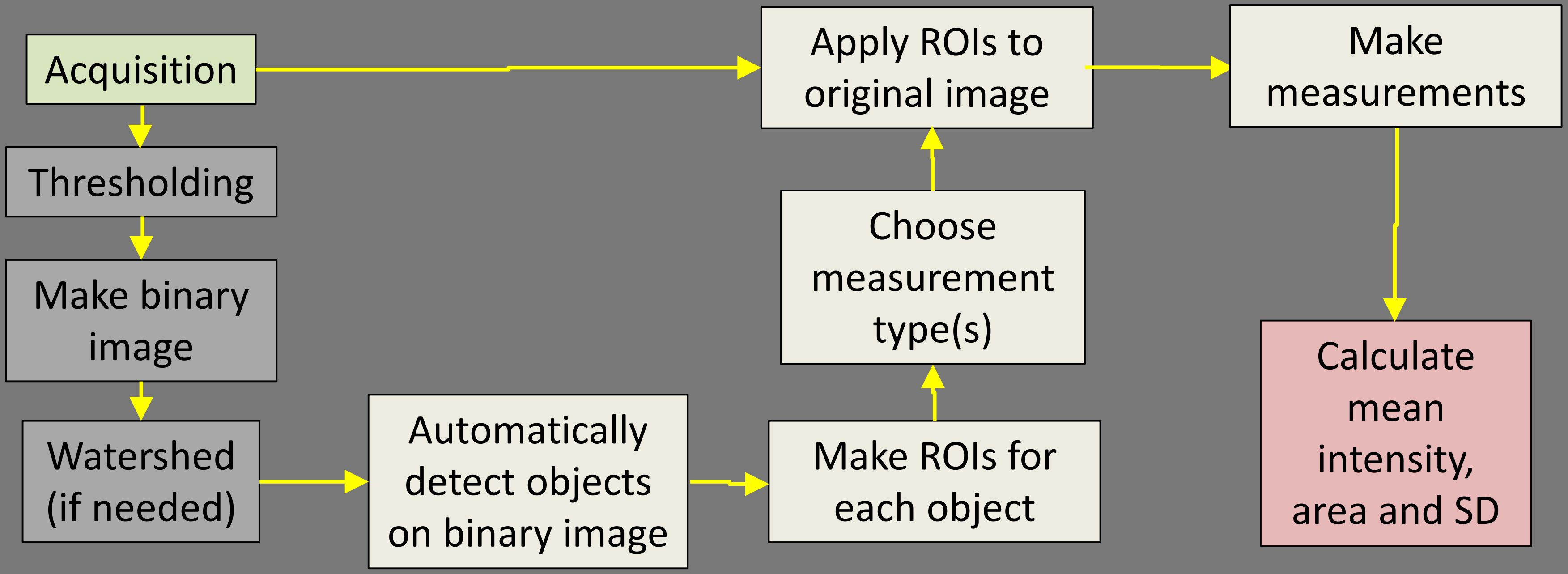
Keep the binary of the nuclei – you need it for the next bit!

Particle analysis – Area and Intensity

Now you know how to select particles, lets use the workflow to do an actual analysis.

This time we will use the binary image to make a separate ROI for each nucleus and then measure the area and mean intensity of each one.

Then we'll calculate mean and SD for both measurements for the whole population.


Acquisition
Thresholding
Make binary image
Watershed (if needed)
Automatically detect objects on binary image
Make ROIs for each object
Choose measurement type(s)
Apply ROIs to original image
Make measurements
Calculate mean intensity, area and SD

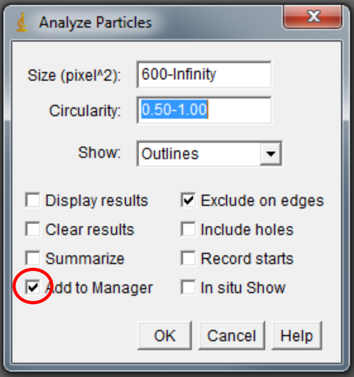
Particle analysis – Area and Intensity

1. Select the watershed binary image and select Analyze>Measure>Analyse Particles again
Repeat the particle analysis step tick "Add to manager". Click "OK".

The ROI manager should pop up with a list of separate ROIs (one for each nucleus), which will also be displayed as an overlay on the watershed binary image.

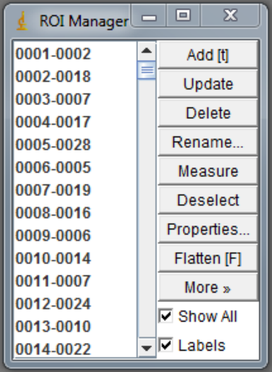
2. Select Analyze>Set Measurements. And select which measurements you want to make, for example tick "Area" and "Mean gray value".

1


Analyze Particles

Size (pixel^2): 600-Infinity
Circularity: 0.50-1.00
Show: Outlines

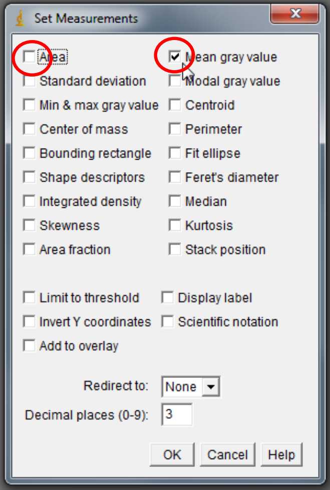
☐ Display results	☑ Exclude on edges
☐ Clear results	☐ Include holes
☐ Summarize	☐ Record starts
☑ Add to Manager	☐ In situ Show


ROI Manager

```
0001-0002
0002-0018
0003-0007
0004-0017
0005-0028
0006-0005
0007-0019
0008-0016
0009-0006
0010-0014
0011-0007
0012-0024
0013-0010
0014-0022
```

☑ Show All
☑ Labels

2

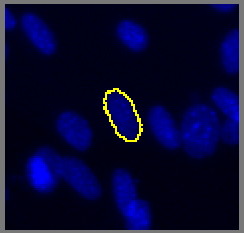

Set Measurements

☐ Area	☑ Mean gray value
☐ Standard deviation	☐ Modal gray value
☐ Min & max gray value	☐ Centroid
☐ Center of mass	☐ Perimeter
☐ Bounding rectangle	☐ Fit ellipse
☐ Shape descriptors	☐ Feret's diameter
☐ Integrated density	☐ Median
☐ Skewness	☐ Kurtosis
☐ Area fraction	☐ Stack position
☐ Limit to threshold	☐ Display label
☐ Invert Y coordinates	☐ Scientific notation
☐ Add to overlay	

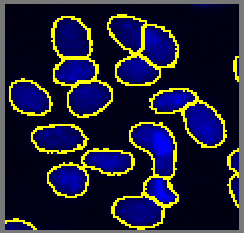
Redirect to: None
Decimal places (0-9): 3

Particle analysis – Area and Intensity

Open the original unthresholded image “Nuclei” again. Individual nuclei can be selected and measured using the ROI manager (you can click the top one and scroll through with the mouse wheel). Select “Measure” or “Ctrl + M” to make individual measurements.



tick “Show all” to display all of the particles in the analysis



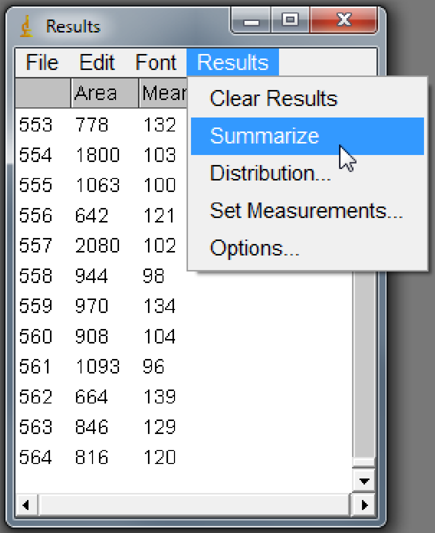
Make multiple selections by holding down [Ctrl] and Left-clicking.

Scroll to the first ROI in the ROI manager. Hold down [shift] and Left click. Scroll to the last ROI. Hold down [shift] and Left click again to select all of the ROIs.

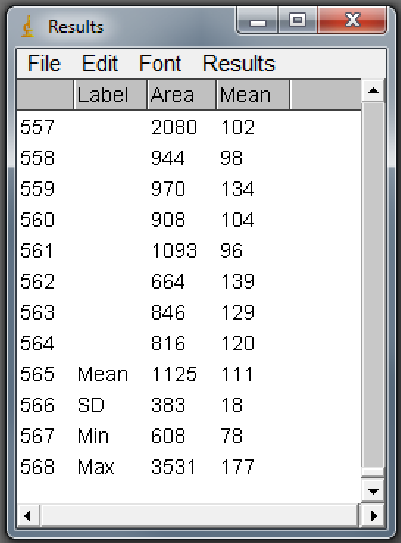
Select “Measure”.

Particle analysis – Area and Intensity

In the results window, select Results>Summarize to get mean and standard deviation data.
Data can be copied and pasted into Excel for further analysis.



	Area	Mear
553	778	132
554	1800	103
555	1063	100
556	642	121
557	2080	102
558	944	98
559	970	134
560	908	104
561	1093	96
562	664	139
563	846	129
564	816	120



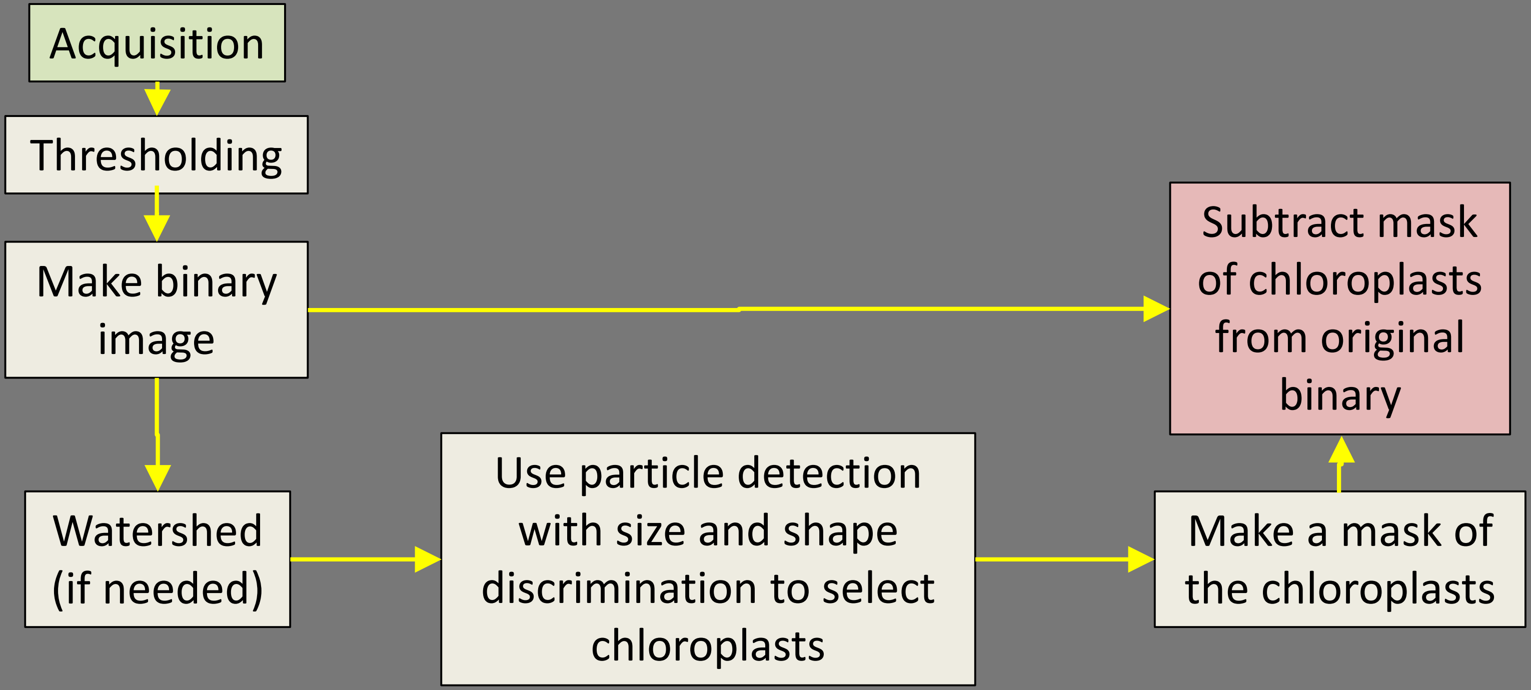
	Label	Area	Mean
557		2080	102
558		944	98
559		970	134
560		908	104
561		1093	96
562		664	139
563		846	129
564		816	120
565	Mean	1125	111
566	SD	383	18
567	Min	608	78
568	Max	3531	177

Mean grey value of particle
Area of particle
Particle number
If image is not calibrated, measurement results will be displayed in pixel units.

Segmentation

We can use segmentation to separate a single image into separate components based on shape, size or intensity to make image analysis of each of those components possible.

For this we use "masks". Masks are just a binary image being used for "Image calculation" (e.g. digital subtraction) to extract or remove specific structures. In this example we'll use a particle analysis to make the masks and remove the chloroplasts from an image of a plant section.

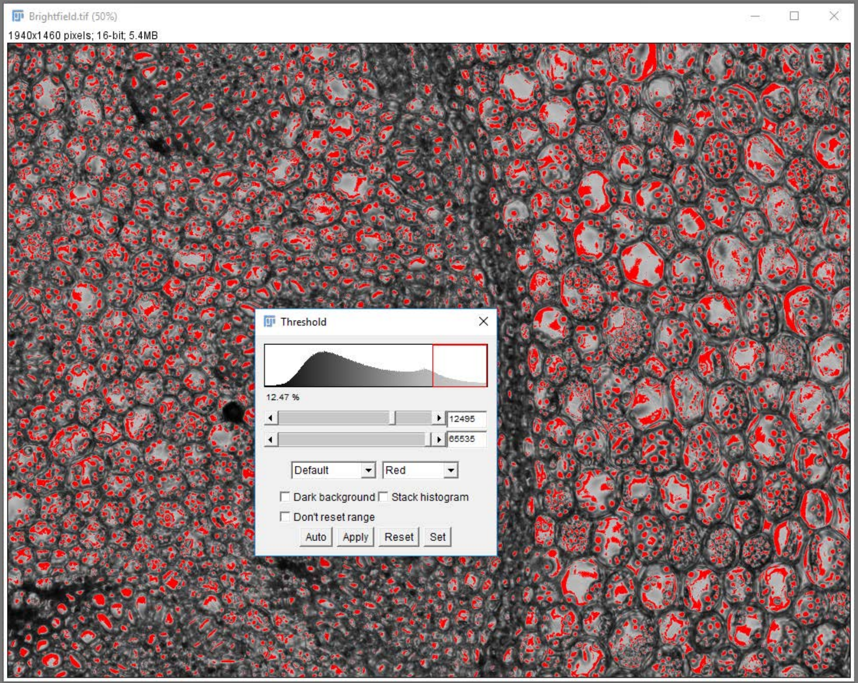

Acquisition
Thresholding
Make binary image
Watershed (if needed)
Use particle detection with size and shape discrimination to select chloroplasts
Make a mask of the chloroplasts
Subtract mask of chloroplasts from original binary

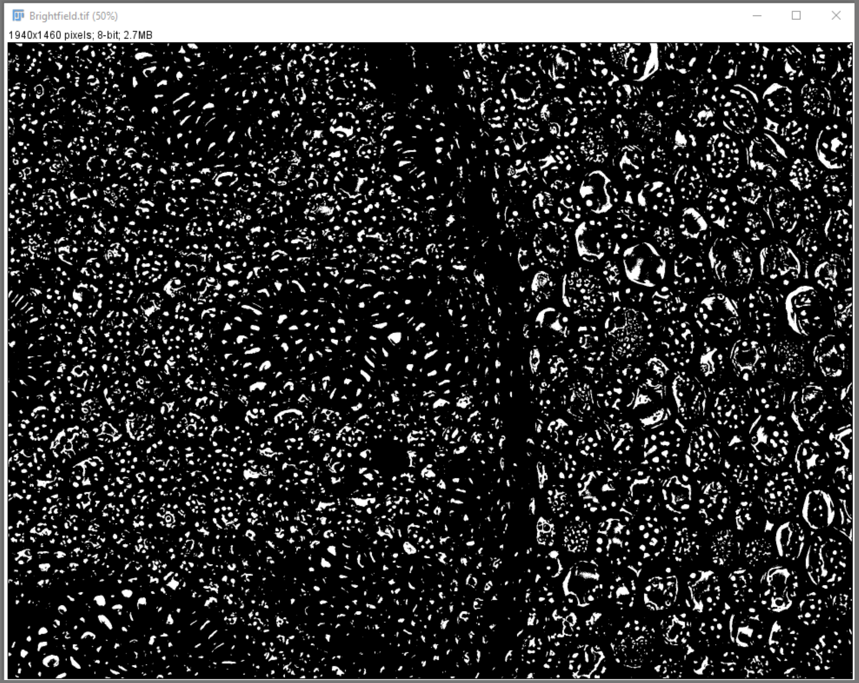
Segmentation

(1) Make a binary image to allow the software to identify the particles.

Open image “Brightfield” We just want to analyse chloroplasts (the small circular foci). Untick “Dark background”, Select “Red” in the pull down menu

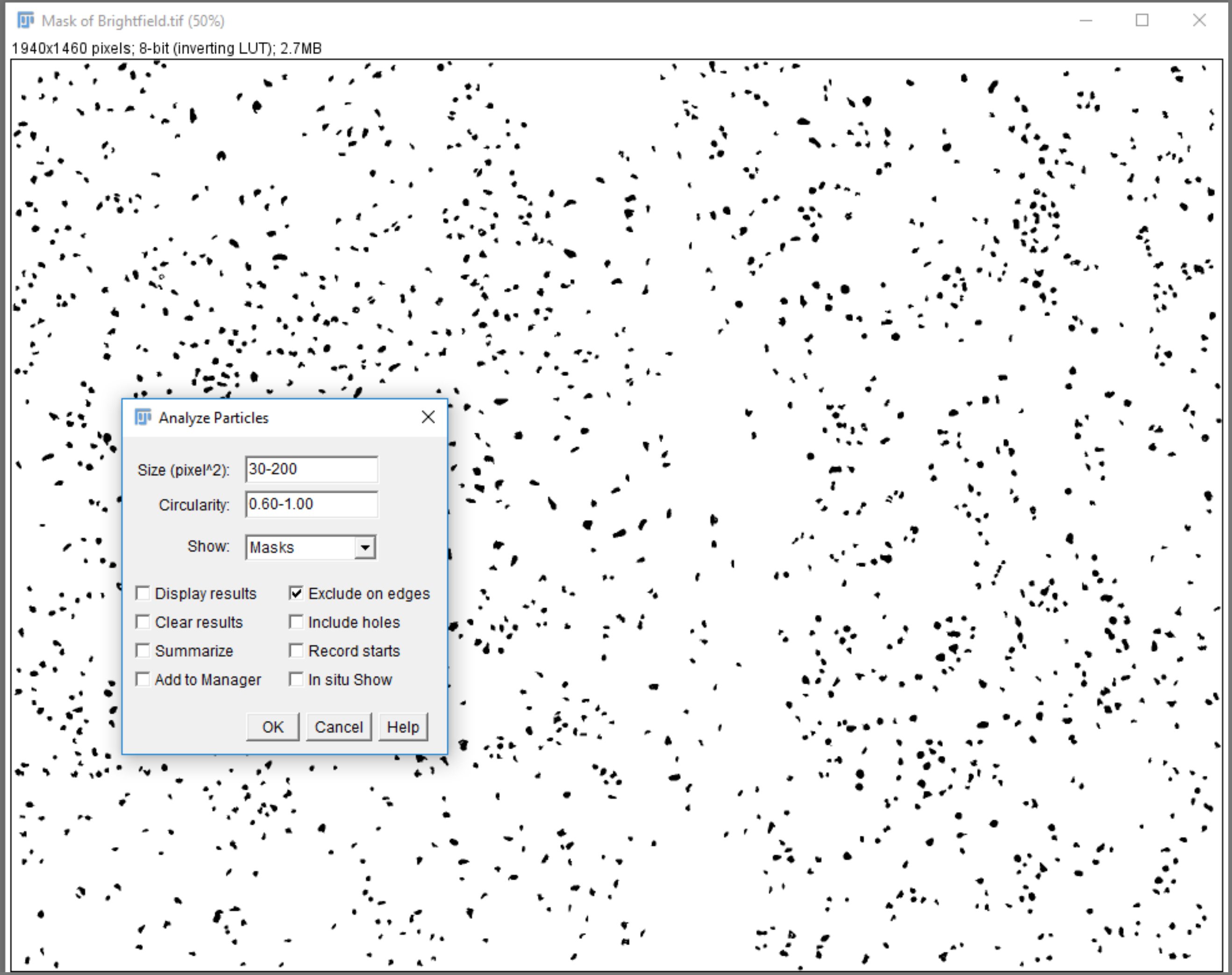
Image>Adjust>Threshold. Threshold using the “Minimum“ (top) slider to find the range of image intensities that excludes the cell walls while preserving other structures (I chose 12495), they should turn red. and click “Apply”. Then “Process>Binary>Make binary”





Segmentation

Analyze>Analyze Particles. Select "Show: Masks" and "Exclude on edges" Adjust the size (30-200) and circularity (0.6-1.00) discrimination of the particle analysis to select only the chloroplasts. Click "OK". A binary image of the chloroplasts appears.


Mask of Brightfield.tif (50%)
1940x1460 pixels; 8-bit (inverting LUT); 2.7MB
Analyze Particles

Size (pixel^2): 30-200
Circularity: 0.60-1.00
Show: Masks

- [ ] Display results
- [x] Exclude on edges
- [ ] Clear results
- [ ] Include holes
- [ ] Summarize
- [ ] Record starts
- [ ] Add to Manager
- [ ] In situ Show

Segmentation

Select Process>Image Calculator. Select your original thresholded binary image as "image 1" and the binary mask of the chloroplasts as "image 2".

Select "Subtract". A new image with the chloroplasts subtracted appears.

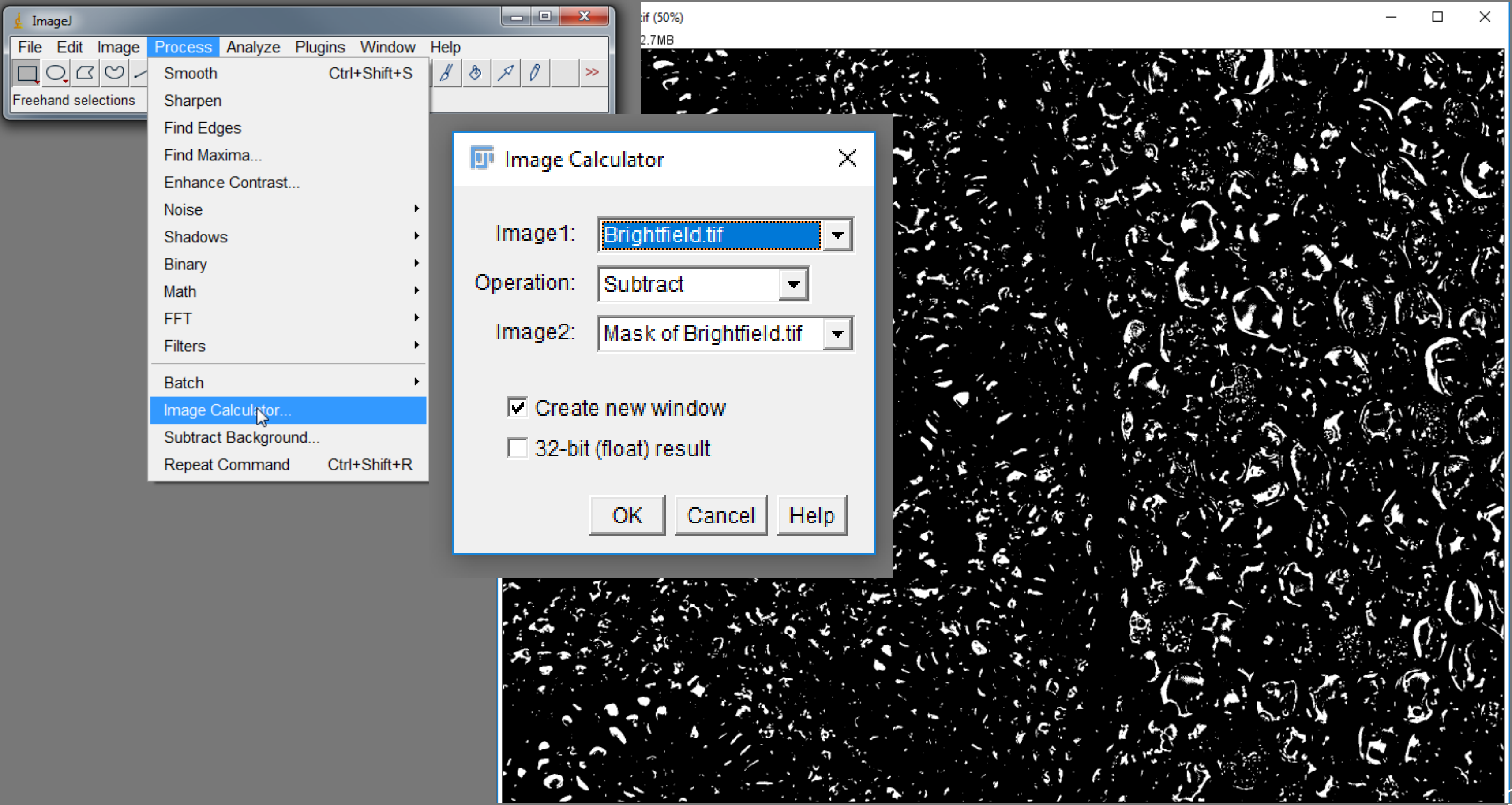


Image Calculator

Image1: Brightfield.tif

Operation: Subtract

Image2: Mask of Brightfield.tif

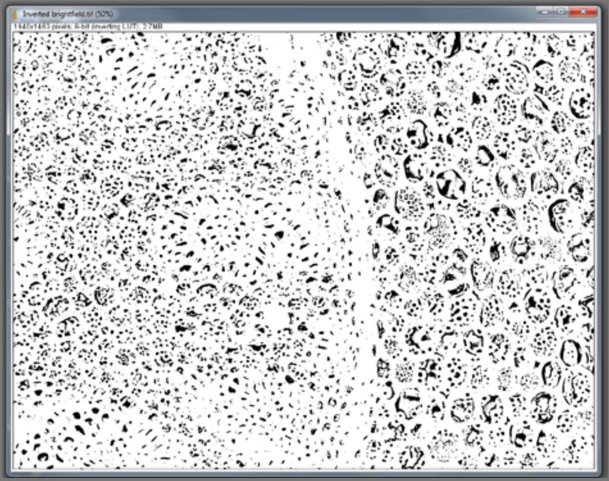
☑ Create new window

☐ 32-bit (float) result

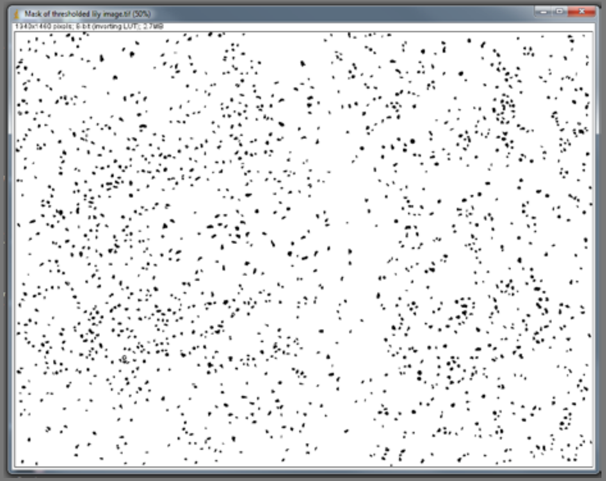
Segmentation

Images can now be used for analysis (e.g. particle count and area, create ROIs to analyse the original unthresholded images, as an image overlay or further segmentation).

Original binary

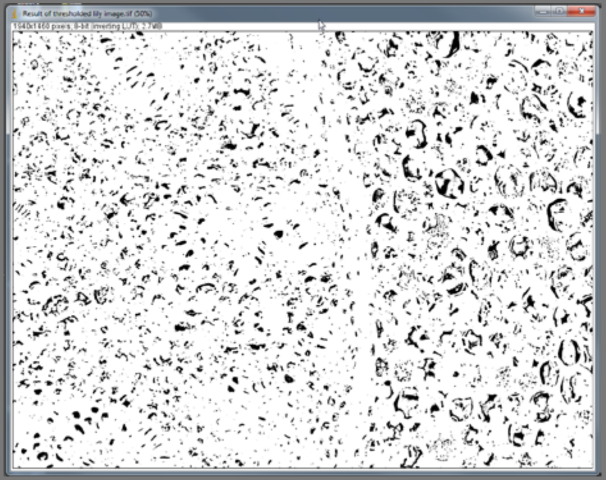


(I used Edit>Invert for clarity)



Binary of chloroplasts

- Particle analysis
- Overlay image
- Make into ROIs for further analysis
- Further segmentation



Binary with chloroplasts removed

- Overlay image
- Analyse stuff that isn't chloroplasts
- Further segmentation

Segmentation

Question:

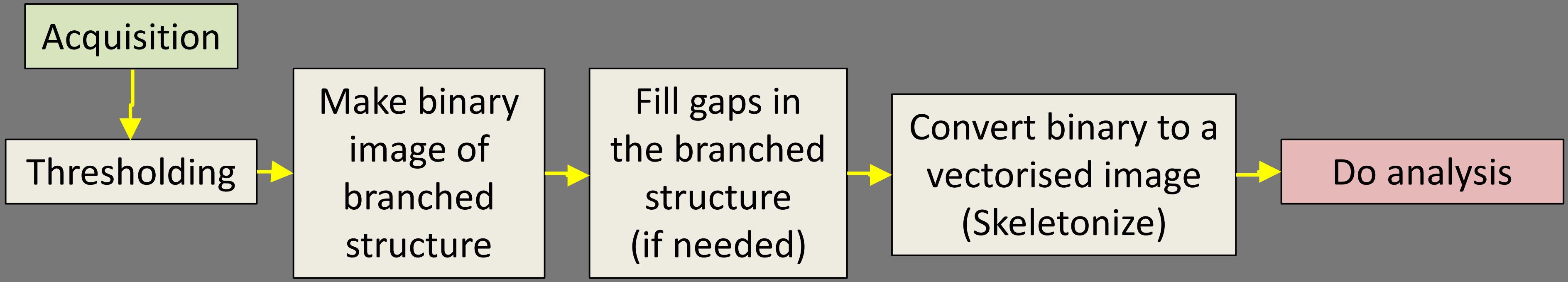
How could you create a grey scale image (rather than a binary) of the plant section without chloroplasts?

Analysing branched structures

We can analyse branched structures using "skeletonisation" of binary images

Branched structures could be neurons, blood vessels, lymph nodes or glands, root structures on plants or any other filamentous branching structure.

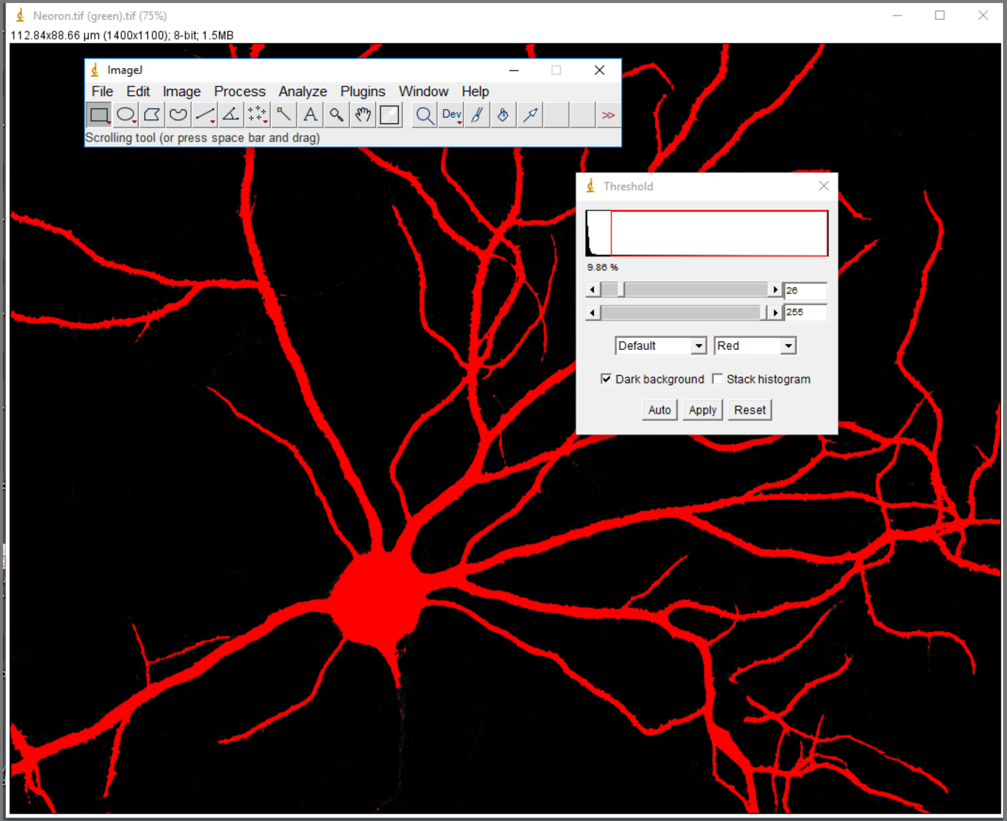
From these we might want to extract information as to the number or length of branches present, the number of branches per branch point, or the tortuosity of the structure (how "gnarly" it is).


Acquisition
Thresholding
Make binary image of branched structure
Fill gaps in the branched structure (if needed)
Convert binary to a vectorised image (Skeletonize)
Do analysis

Branch analysis

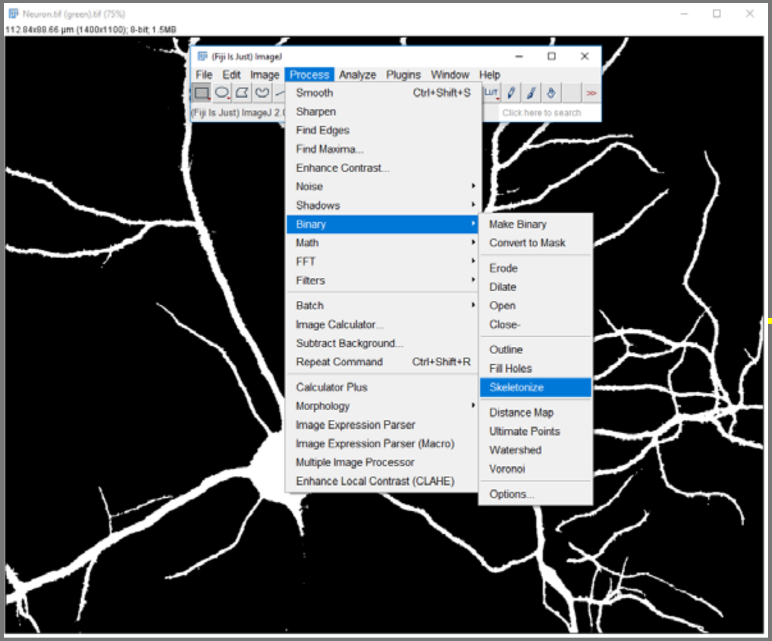
Open image: Neuron (green).tif

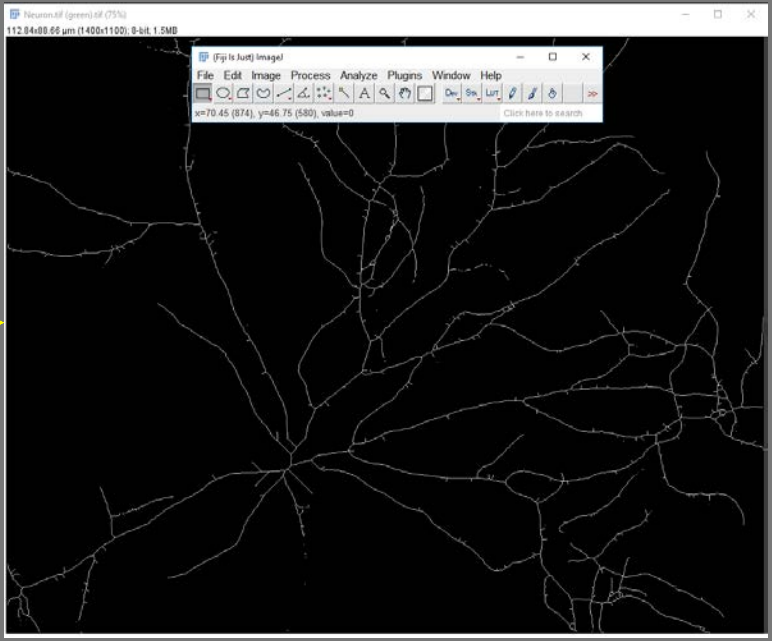
Threshold the image
Image>Adjust>Threshold. Choose "Default" and "Red" in the pulldown menus.
Tick "Dark Background"
To threshold the image, move the top slider until all of the foreground is red. Try to preserve the continuity of the axons as much as possible. When done click "Apply".


Neoron.tif (green).tif (75%)
112.84x88.66 µm (1400x1100); 8-bit; 1.5MB
ImageJ
File Edit Image Process Analyze Plugins Window Help
Scrolling tool (or press space bar and drag)
Threshold
9.86 %
26
255
Default
Red
Dark background
Stack histogram
Auto
Apply
Reset

Skeletonise the binary image
Process>Binary>Skeletonize.

This converts the binary image to vectors – discrete lines with a known start and finish position.

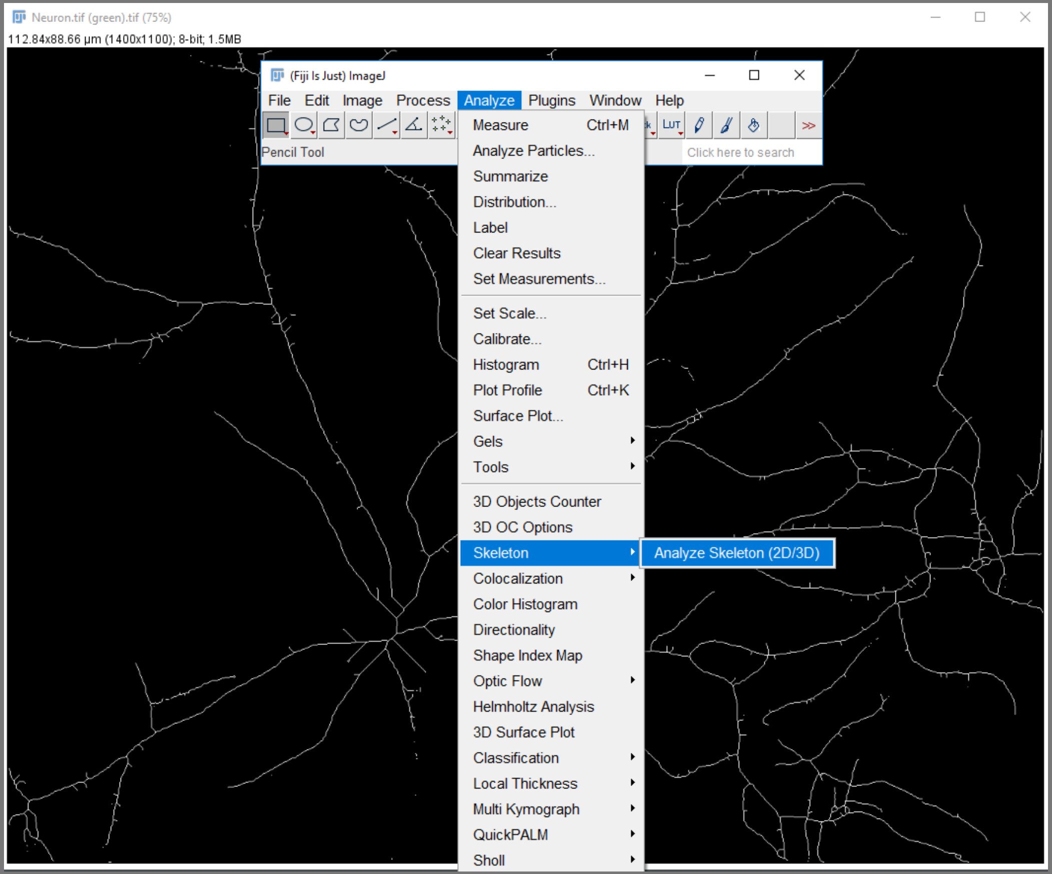




Neuron.tif (green).tif (75%)
112.84x88.66 µm (1400x1100); 8-bit; 1.5MB
(Fiji Is Just) ImageJ
File
Edit
Image
Process
Analyze
Plugins
Window
Help
x=70.45 (874), y=46.75 (580), value=0
Click here to search

Analyse the vectorised image

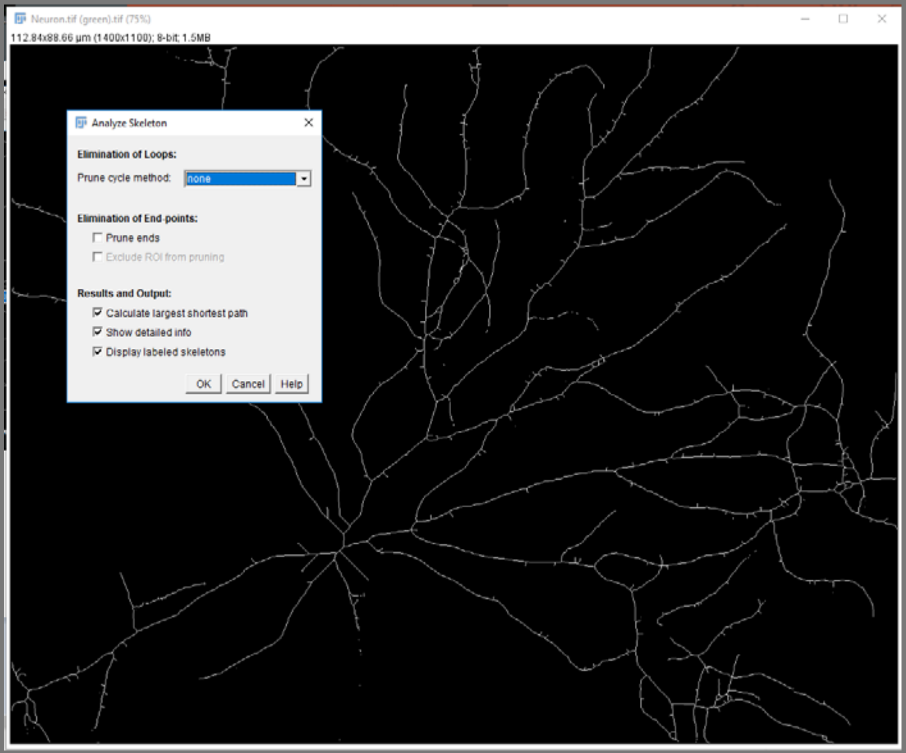
Analyse>Skeleton>Analyze Skeleton 2D/3D.



Tick:

Show detailed info and Display labelled skeletons

Select OK


Neuron.tif (green).tif (75%)
112.84x88.66 µm (1400x1100); 8-bit; 1.5MB

Analyze Skeleton

Elimination of Loops:

Prune cycle method: none

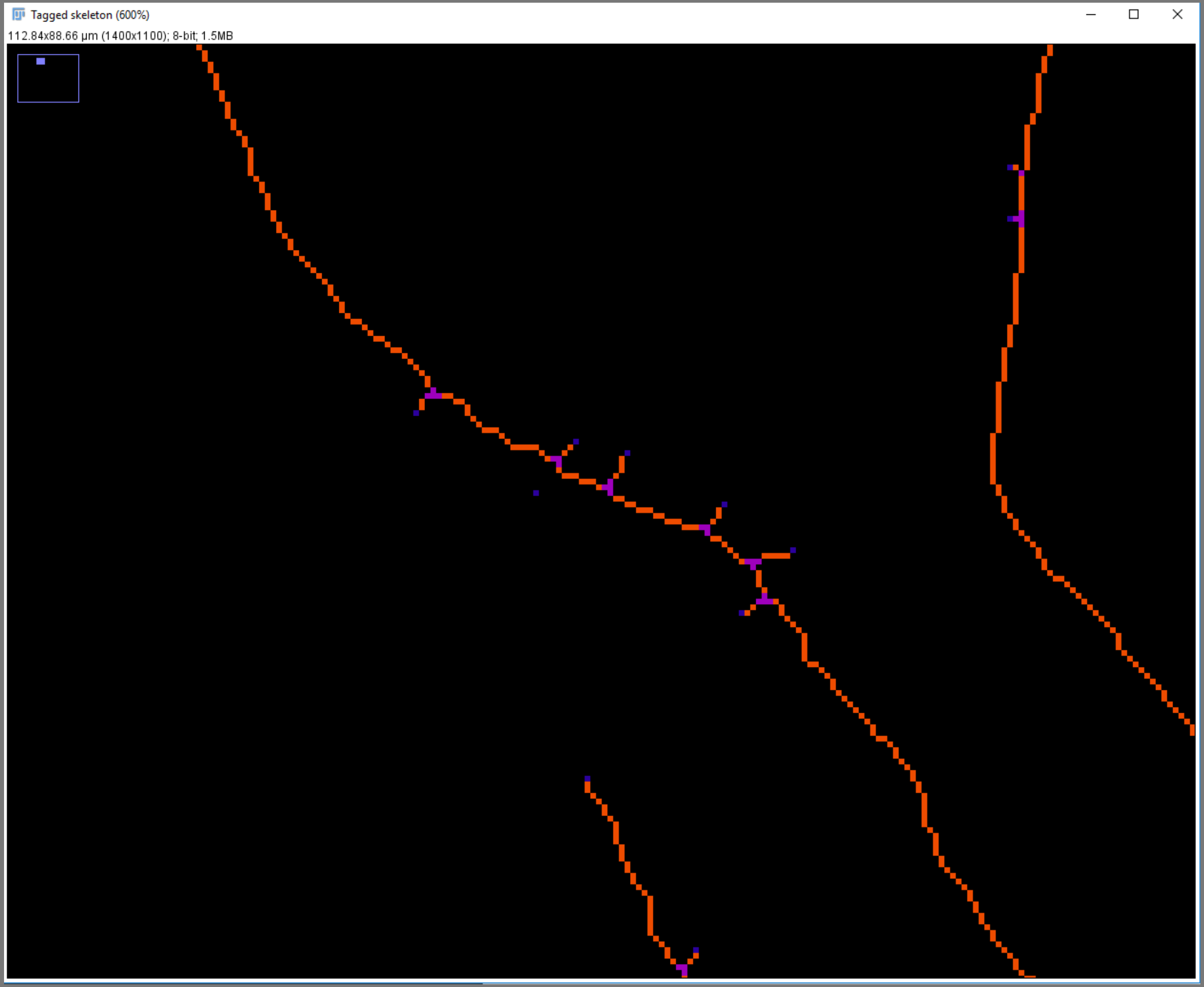
Elimination of End-points:

- [ ] Prune ends
- [ ] Exclude ROI from pruning

Results and Output:

- [✓] Calculate largest shortest path
- [✓] Show detailed info
- [✓] Display labeled skeletons

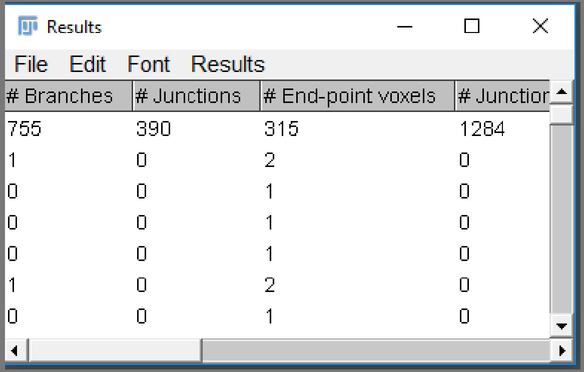
On the resulting image, branches are labelled in orange, junctions in magenta and end points in blue.



(Zoomed in with [Ctrl] + mouse wheel)

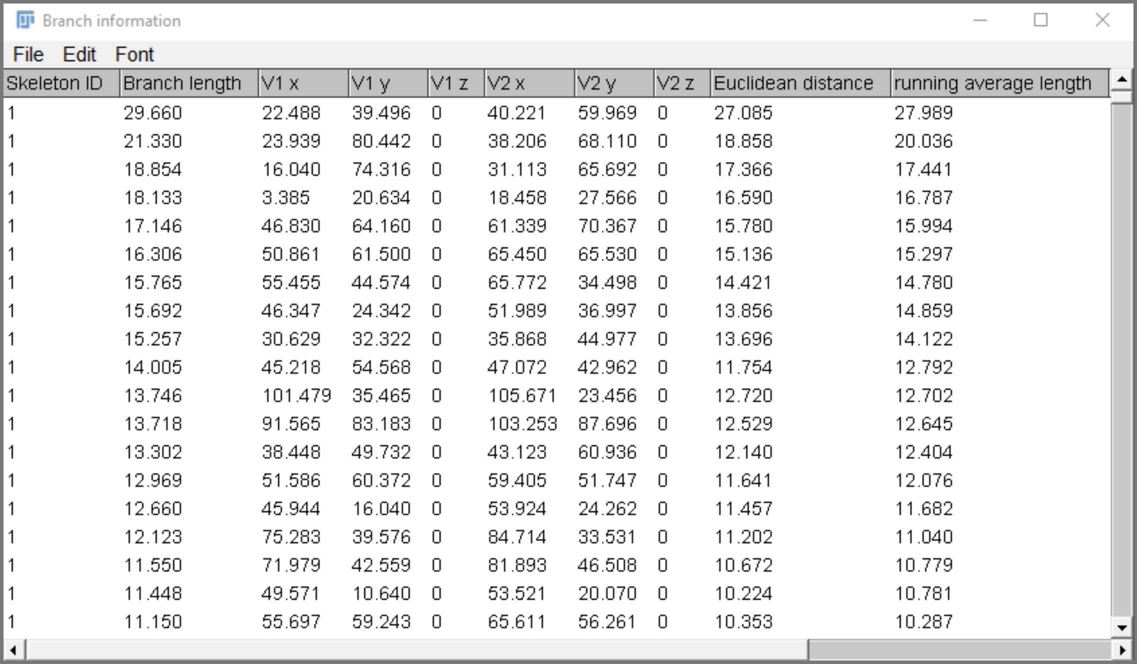
There are two results windows. One gives you general information about each complete skeleton in the image. E.g. how branched it is and how many double or triple junctions there are.

The other gives information about each branch in the skeleton. E.g. Branch length *vs* Euclidian distance will give you information about tortuosity. You could bin number of branches of each type depending on the branch length etc.


Results

File Edit Font Results

# Branches	# Junctions	# End-point voxels	# Junction
755	390	315	1284
1	0	2	0
0	0	1	0
0	0	1	0
0	0	1	0
1	0	2	0
0	0	1	0


Branch information

File Edit Font

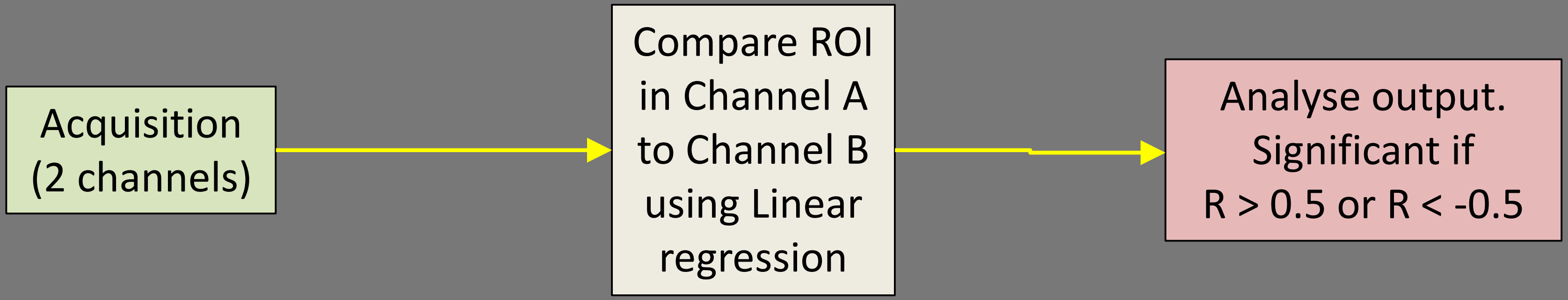
Skeleton ID	Branch length	V1 x	V1 y	V1 z	V2 x	V2 y	V2 z	Euclidean distance	running average length
1	29.660	22.488	39.496	0	40.221	59.969	0	27.085	27.989
1	21.330	23.939	80.442	0	38.206	68.110	0	18.858	20.036
1	18.854	16.040	74.316	0	31.113	65.692	0	17.366	17.441
1	18.133	3.385	20.634	0	18.458	27.566	0	16.590	16.787
1	17.146	46.830	64.160	0	61.339	70.367	0	15.780	15.994
1	16.306	50.861	61.500	0	65.450	65.530	0	15.136	15.297
1	15.765	55.455	44.574	0	65.772	34.498	0	14.421	14.780
1	15.692	46.347	24.342	0	51.989	36.997	0	13.856	14.859
1	15.257	30.629	32.322	0	35.868	44.977	0	13.696	14.122
1	14.005	45.218	54.568	0	47.072	42.962	0	11.754	12.792
1	13.746	101.479	35.465	0	105.671	23.456	0	12.720	12.702
1	13.718	91.565	83.183	0	103.253	87.696	0	12.529	12.645
1	13.302	38.448	49.732	0	43.123	60.936	0	12.140	12.404
1	12.969	51.586	60.372	0	59.405	51.747	0	11.641	12.076
1	12.660	45.944	16.040	0	53.924	24.262	0	11.457	11.682
1	12.123	75.283	39.576	0	84.714	33.531	0	11.202	11.040
1	11.550	71.979	42.559	0	81.893	46.508	0	10.672	10.779
1	11.448	49.571	10.640	0	53.521	20.070	0	10.224	10.781
1	11.150	55.697	59.243	0	65.611	56.261	0	10.353	10.287

Colocalisation analysis.

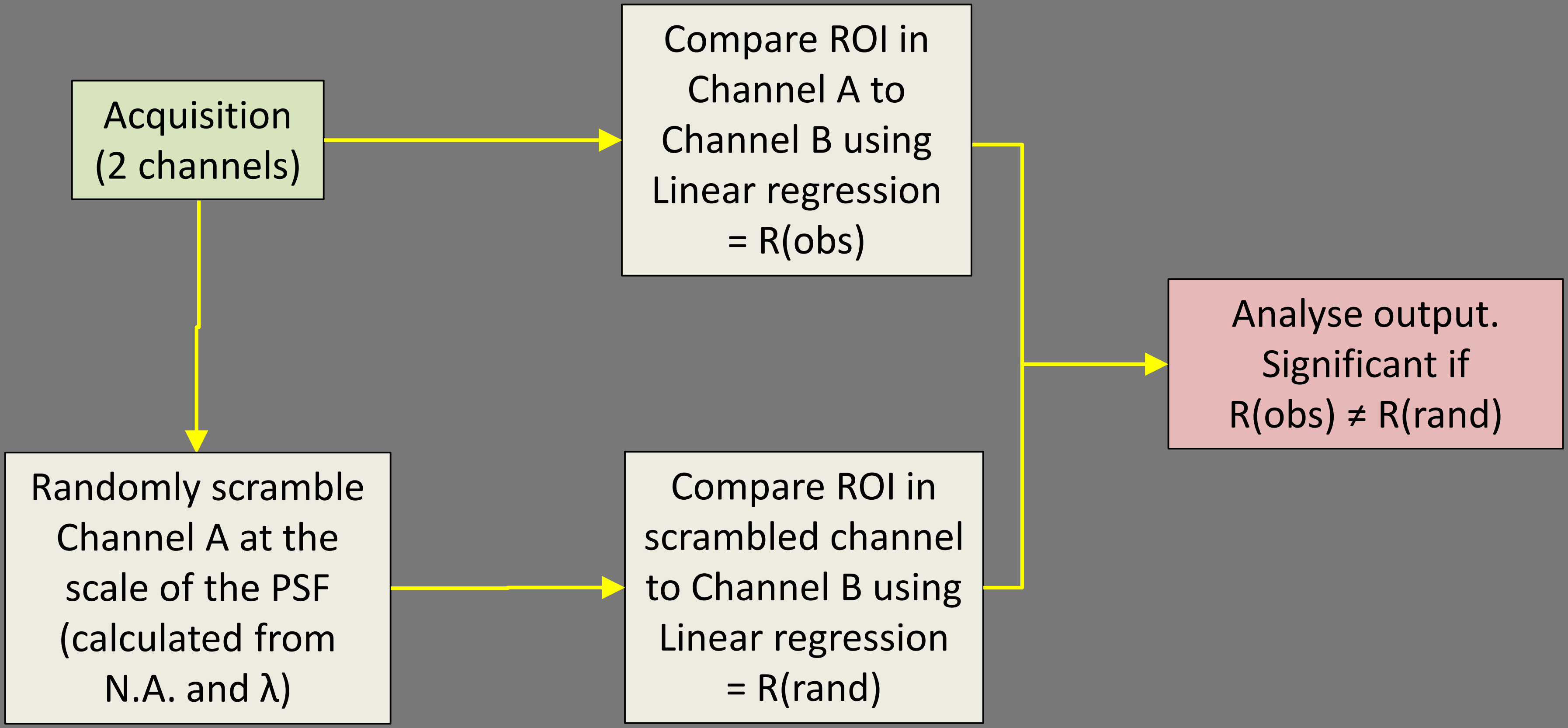
Colocalisation is a process where we attempt to quantify the relationship between markers from two different channels.

There are many methods, but they fall into two main categories – quantifying pixel overlap or quantifying co-dependency using regression. For this workshop we'll focus on the regression method, however here are some other workflows.

<u>Simple Pearson's R analysis (Regression, background insensitive)</u>

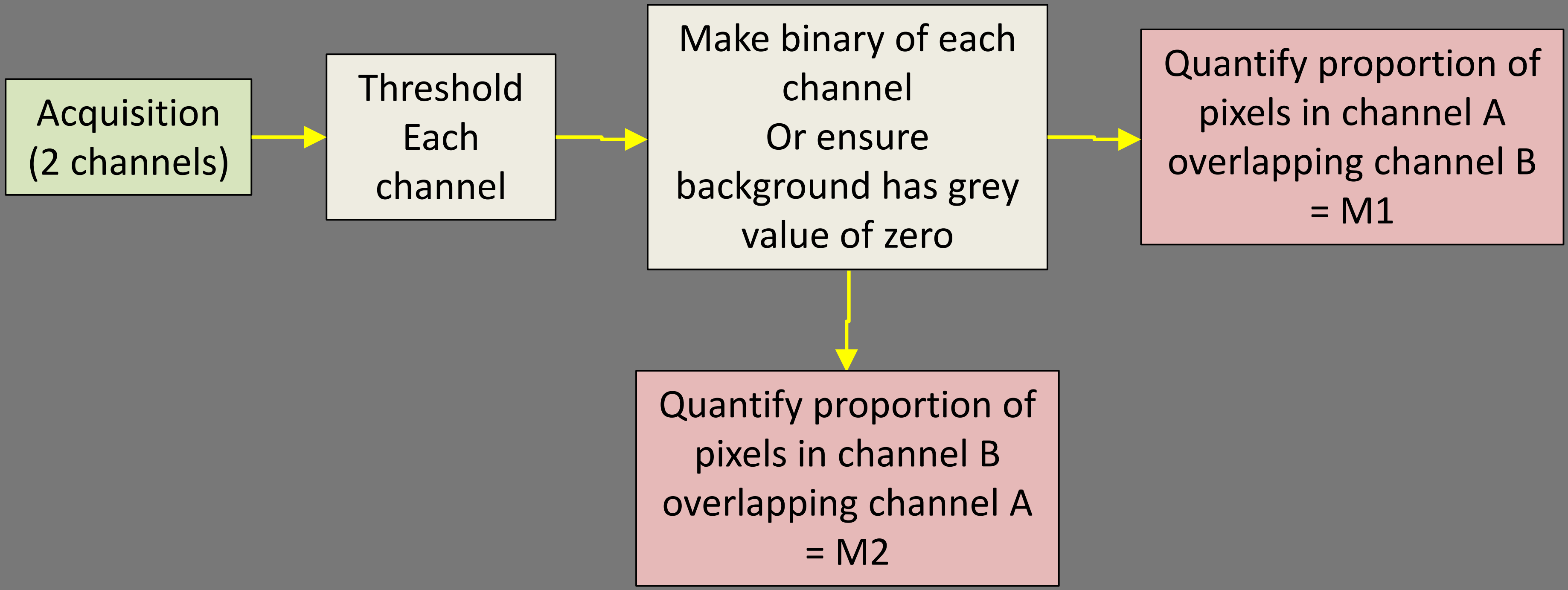

Acquisition (2 channels)
Compare ROI in Channel A to Channel B using Linear regression
Analyse output. Significant if R > 0.5 or R < -0.5

Costes' method (Regression, background insensitive)

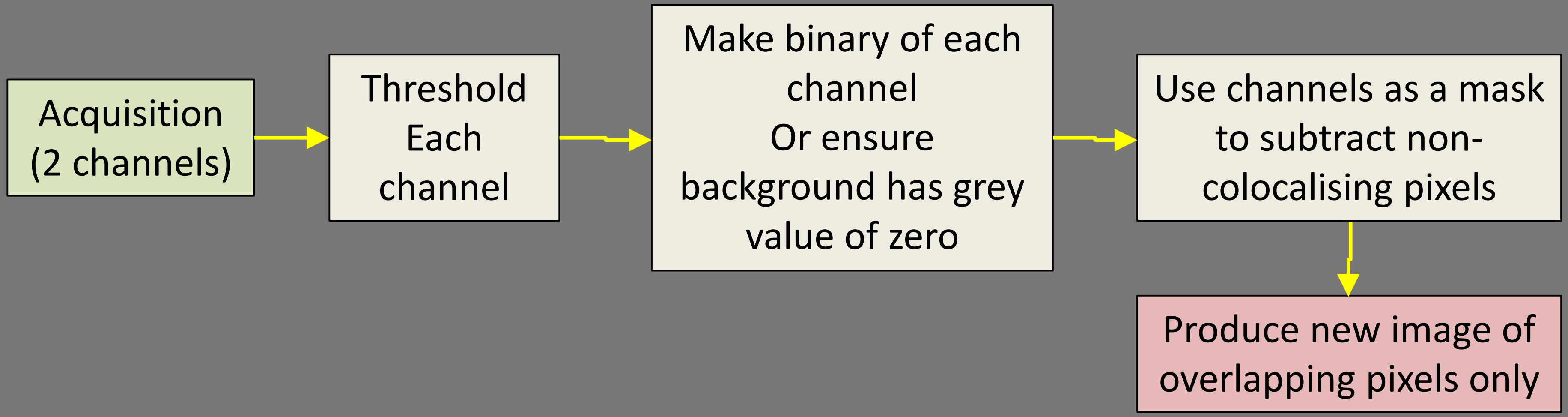

Acquisition (2 channels)
Compare ROI in Channel A to Channel B using Linear regression = R(obs)
Randomly scramble Channel A at the scale of the PSF (calculated from N.A. and λ)
Compare ROI in scrambled channel to Channel B using Linear regression = R(rand)
Analyse output. Significant if R(obs) ≠ R(rand)

Manders Coefficients (Overlap, background Sensitive)

We may know that two targets colocalise from their Pearson's R, but we can also quantify how much of each target is colocalising with the other channel.


Acquisition (2 channels)
Threshold Each channel
Make binary of each channel Or ensure background has grey value of zero
Quantify proportion of pixels in channel A overlapping channel B = M1
Quantify proportion of pixels in channel B overlapping channel A = M2

Digital subtraction (Overlap, background Sensitive, not quantitative)


Acquisition (2 channels)
Threshold Each channel
Make binary of each channel Or ensure background has grey value of zero
Use channels as a mask to subtract non-colocalising pixels
Produce new image of overlapping pixels only

Pearson’s colocalisation analysis

Colocalisation analysis provides a statistically testable numerical value relative to the degree of colocalisation between two image channels.

Pearson’s colocalisation coefficient uses linear regression to measure the co-dependency of the variations in grey intensity across two channel images and returns the value “R”.

The value for R can range from -1 to +1 and can be interpreted as follows:

R = -1	The two stains are absolutely mutually exclusive
R = 0	No significant co-dependency between the two stains – random distribution
R = 1	The two stains are absolutely dependent (A value very close to 1 may indicate experimental error or channel cross-talk)

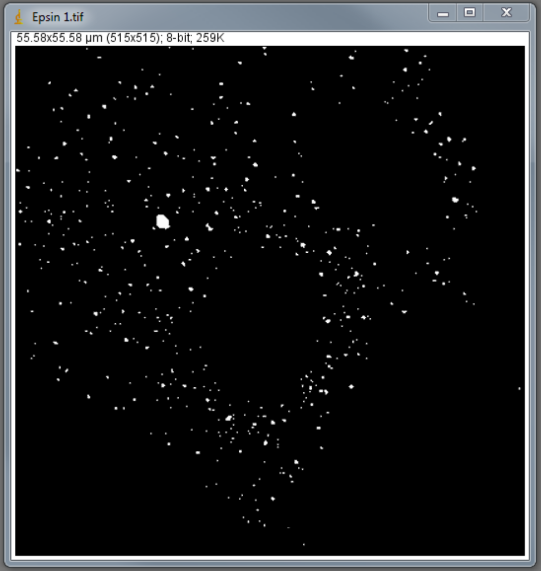
Pearson’s colocalisation analysis

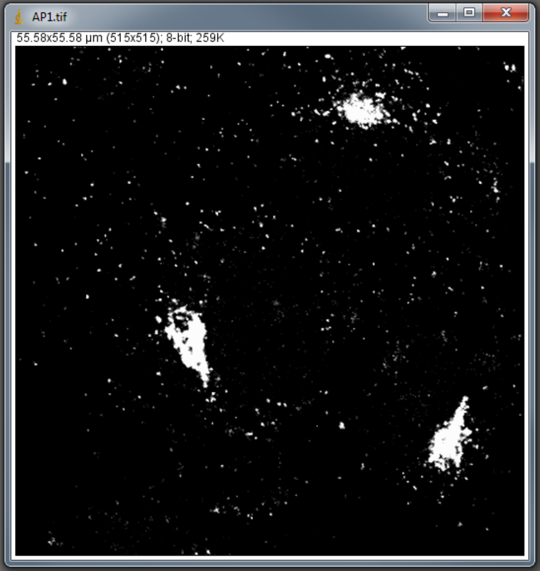
As R-values may be positive but still close to zero, a value closer to 0.5 (or above) is considered statistically significant. However, the significance of Pearson’s R can be further tested using Costes’ approximation:

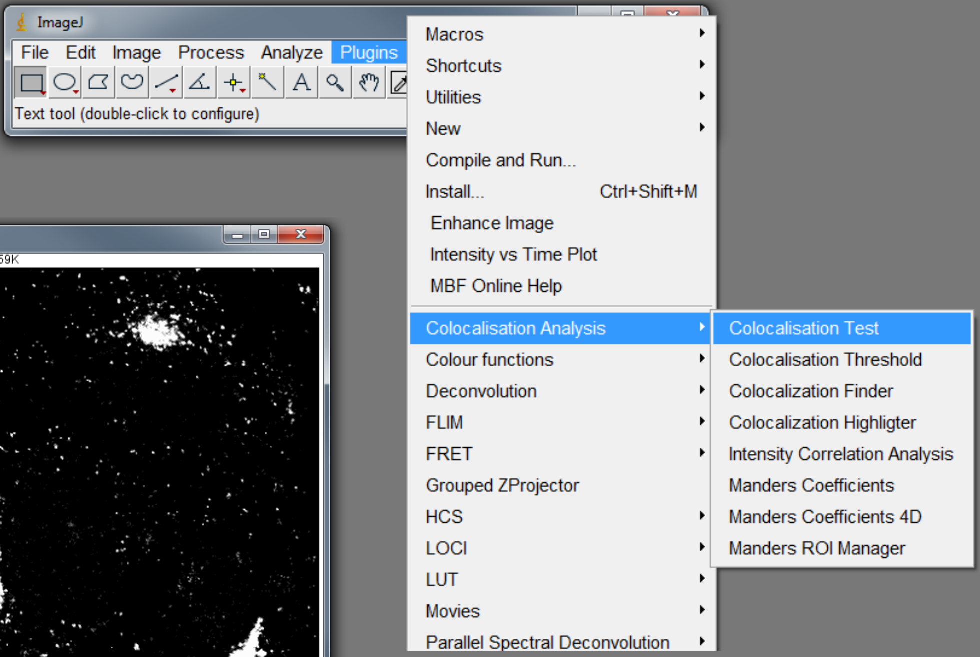
- The channel images are analysed to return the observed Pearson’s coefficient R(obs).
- One of the channel images is then randomised using a unit size determined by the optical resolution of the image determined by a PSF calculated from the N.A. of the objective lens used.
- The randomised image is then compared to the intact channel image using Pearson’s colocalisation coefficient.
- This process is repeated 200 times to return R(rand) with an expected outcome close to zero, and a standard deviation is calculated.

Pearson's colocalisation analysis – worked example

Open channel images "Epsin 1.tif" and ""AP1.tif"
Select Plugins>Colocalisation Analysis>Colocalisation Test.







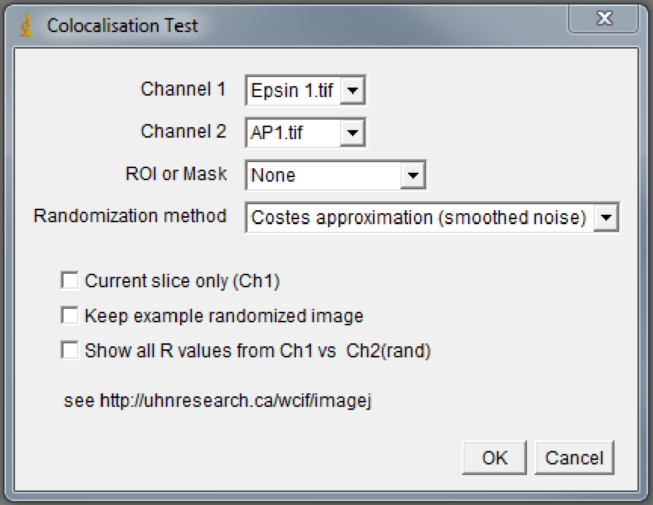
Pearson's colocalisation analysis – worked example

① Select Epsin 1 as Channel 1 and AP1 as Channel 2 and click "OK"

② Enter the channel 2 wavelength and the objective N.A. Iterations should be set to 200.

Click "OK".

①

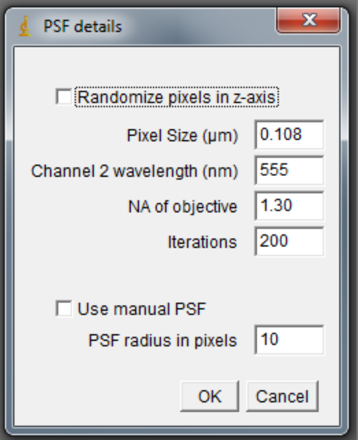

Colocalisation Test

Channel 1	Epsin 1.tif
Channel 2	AP1.tif
ROI or Mask	None
Randomization method	Costes approximation (smoothed noise)

☐ Current slice only (Ch1)
☐ Keep example randomized image
☐ Show all R values from Ch1 vs Ch2(rand)

see http://uhnresearch.ca/wcif/imagej

②


PSF details

☐ Randomize pixels in z-axis

Pixel Size (µm)	0.108
Channel 2 wavelength (nm)	555
NA of objective	1.30
Iterations	200

☐ Use manual PSF

PSF radius in pixels	10

Pearson’s colocalisation analysis – worked example

A results window pops up summarising the R(obs) and R(rand) for the two channels.

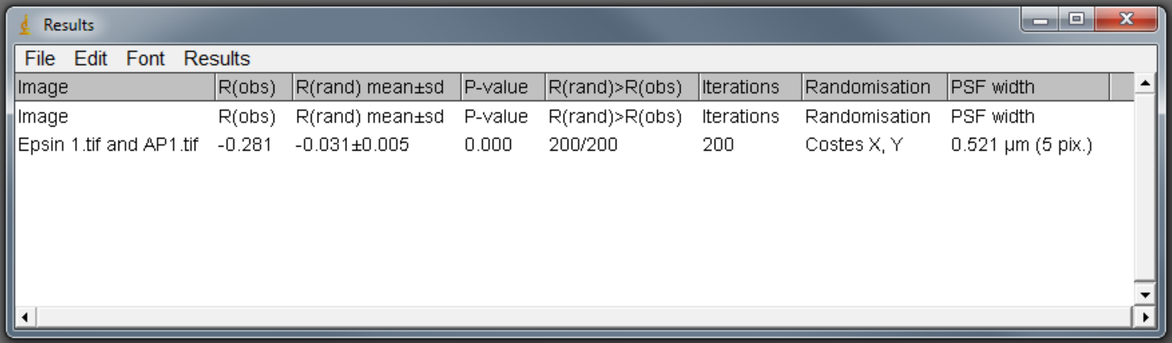
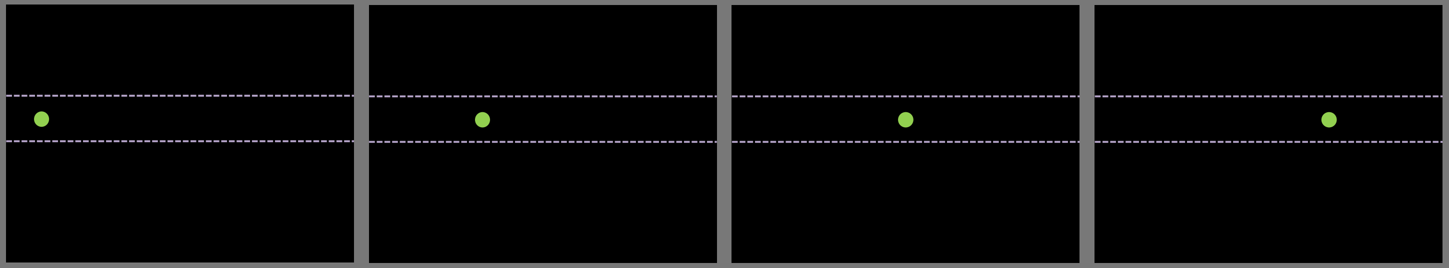


Image	R(obs)	R(rand) mean±sd	P-value	R(rand)>R(obs)	Iterations	Randomisation	PSF width
Image	R(obs)	R(rand) mean±sd	P-value	R(rand)>R(obs)	Iterations	Randomisation	PSF width
Epsin 1.tif and AP1.tif	-0.281	-0.031±0.005	0.000	200/200	200	Costes X, Y	0.521 µm (5 pix.)

Manual tracking



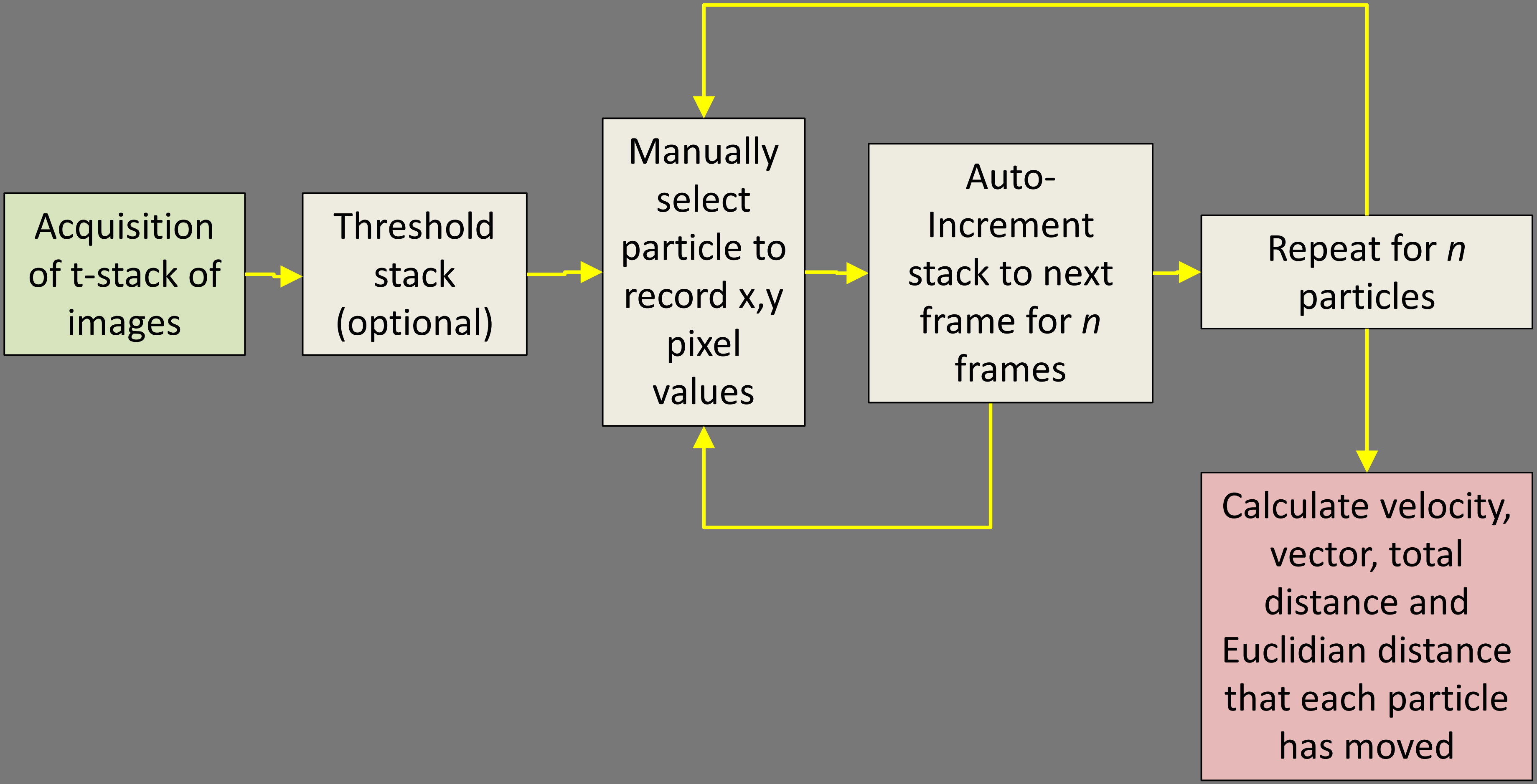
t=0 **+5 min** **+10 min** **+15 min**

The microscope is focussed the specimen (e.g. paricles being transported on an axon) and a sequence of images is collected at multiple time-points

Images are converted to binary (black and white) and combined into a single “t-stack” file (a scrollable time-lapse movie).

Manual tracking

Can be used to track intracellular particles/organelles or individuals within larger motile populations, e.g. cells.


Acquisition of t-stack of images
Threshold stack (optional)
Manually select particle to record x,y pixel values
Auto-Increment stack to next frame for *n* frames
Repeat for *n* particles
Calculate velocity, vector, total distance and Euclidian distance that each particle has moved

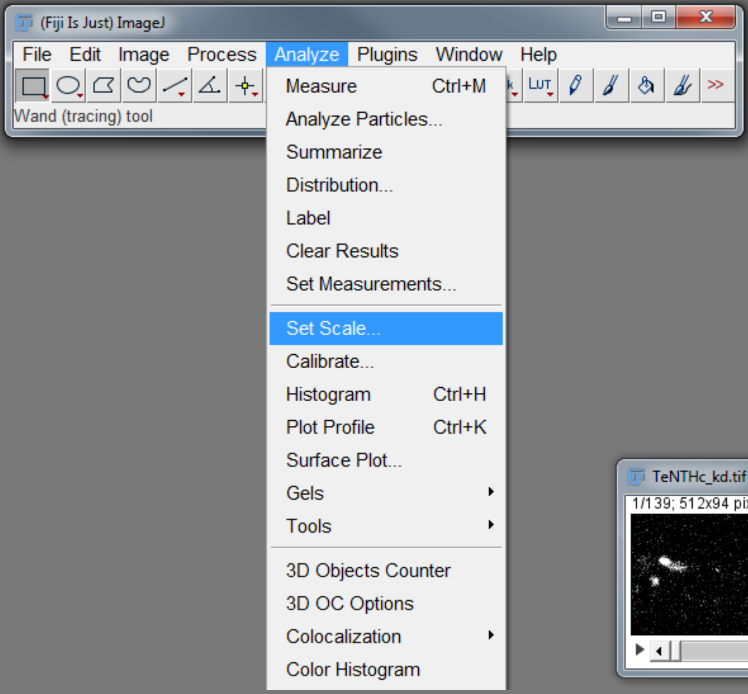
Measuring axonal transport

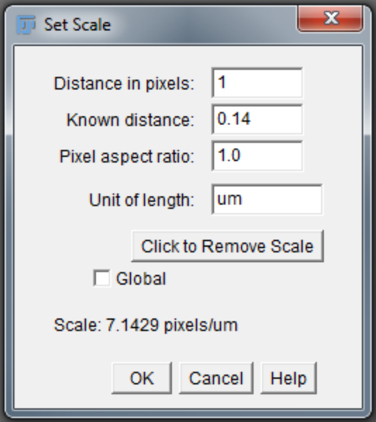
Calibrate the images:

Open image stack “TeNTHc_kd”.

Select Analyze>Set Scale...

Enter the microns/pixel and “um” as the unit of length and click “OK”




Set Scale

Distance in pixels: 1

Known distance: 0.14

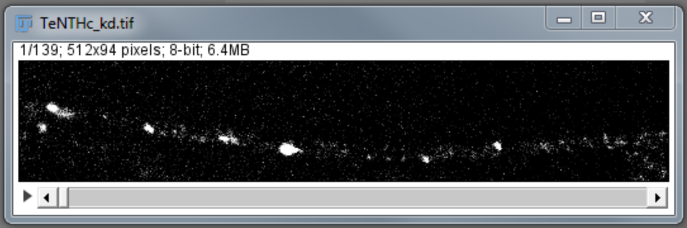
Pixel aspect ratio: 1.0

Unit of length: um

Click to Remove Scale

☐ Global

Scale: 7.1429 pixels/um

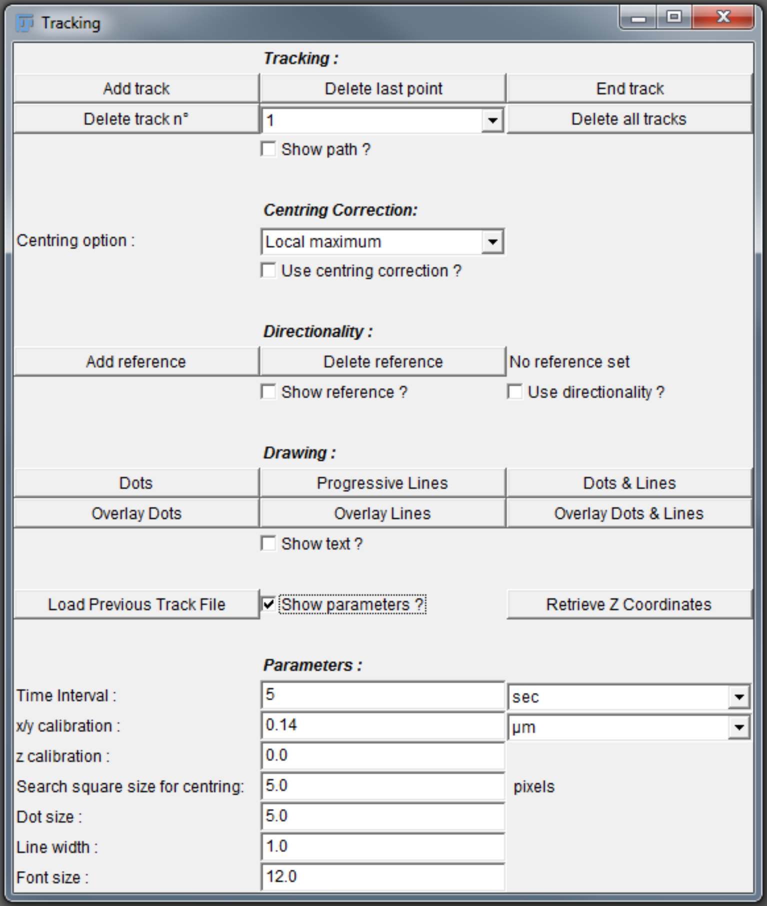

TeNTHc_kd.tif

1/139; 512x94 pixels; 8-bit; 6.4MB

Measuring axonal transport

Open the manual tracking plugin:

Plugins>Tracking>Manual Tracking : http://rsbweb.nih.gov/ij/plugins/track/track.html


Tracking

Tracking :

Add track	Delete last point	End track
Delete track n°	1	Delete all tracks

☐ Show path ?

Centring Correction:

Centring option : Local maximum

☐ Use centring correction ?

Directionality :

Add reference	Delete reference	No reference set

☐ Show reference ? ☐ Use directionality ?

Drawing :

Dots	Progressive Lines	Dots & Lines
Overlay Dots	Overlay Lines	Overlay Dots & Lines

☐ Show text ?

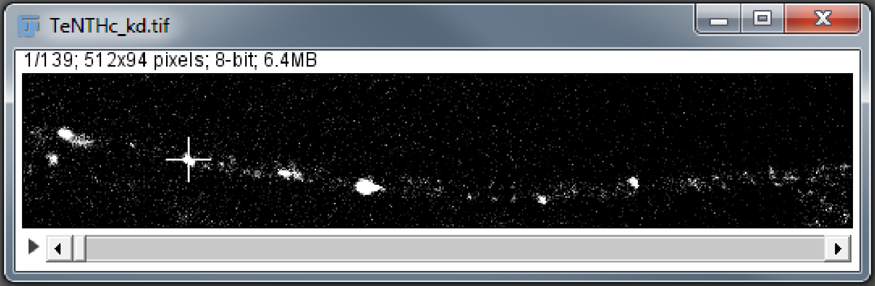
Load Previous Track File ✓ Show parameters ? Retrieve Z Coordinates

Parameters :

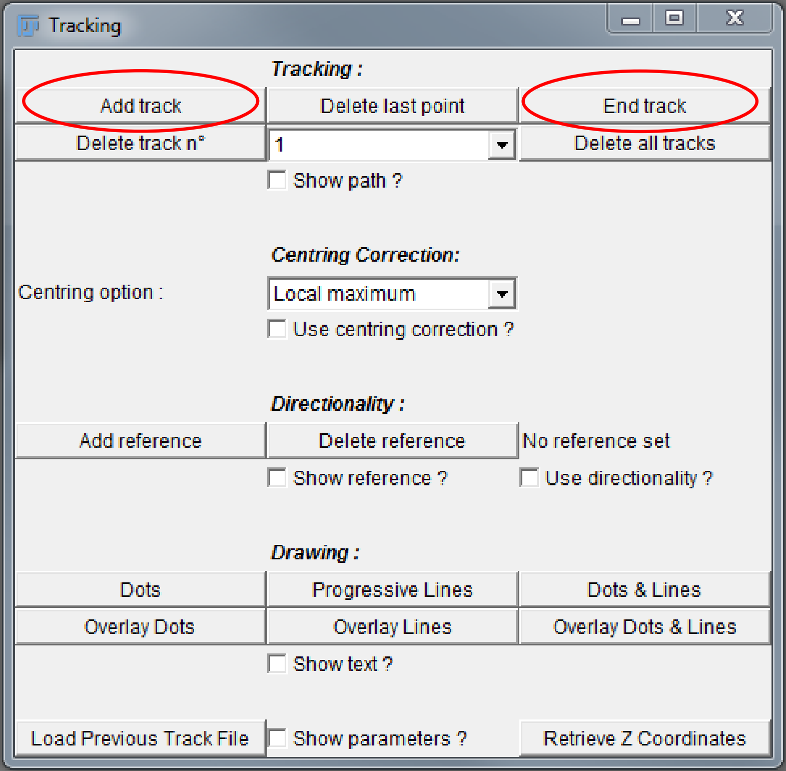
Time Interval :	5	sec
x/y calibration :	0.14	µm
z calibration :	0.0	
Search square size for centring:	5.0	pixels
Dot size :	5.0	
Line width :	1.0	
Font size :	12.0	

Enter the calibration parameters:
“Time interval” (seconds) and
“x/y calibration” (microns/pixel)

Measuring axonal transport


TeNTHc_kd.tif

1/139; 512x94 pixels; 8-bit; 6.4MB


Tracking

Tracking :

Add track | Delete last point | End track

Delete track n° | 1 | Delete all tracks

☐ Show path ?

Centring Correction:

Centring option : Local maximum

☐ Use centring correction ?

Directionality :

Add reference | Delete reference | No reference set

☐ Show reference ? ☐ Use directionality ?

Drawing :

Dots | Progressive Lines | Dots & Lines

Overlay Dots | Overlay Lines | Overlay Dots & Lines

☐ Show text ?

Load Previous Track File | ☐ Show parameters ? | Retrieve Z Coordinates

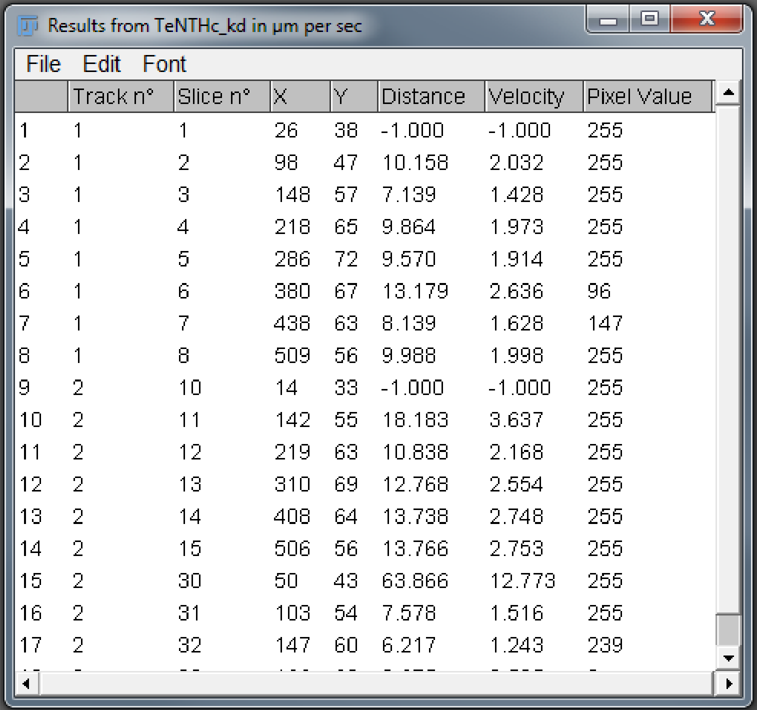
① To start the analysis of each particle, click "Add track"

② Using the mouse cursor, click on the particle that you want to track. The stack will increment by one frame. Continue clicking on the particle until you reach the end of its travel.

③ When you have finished tracking the particle, click "End track". Click "Add track" to start tracking a new one.

Measuring axonal transport

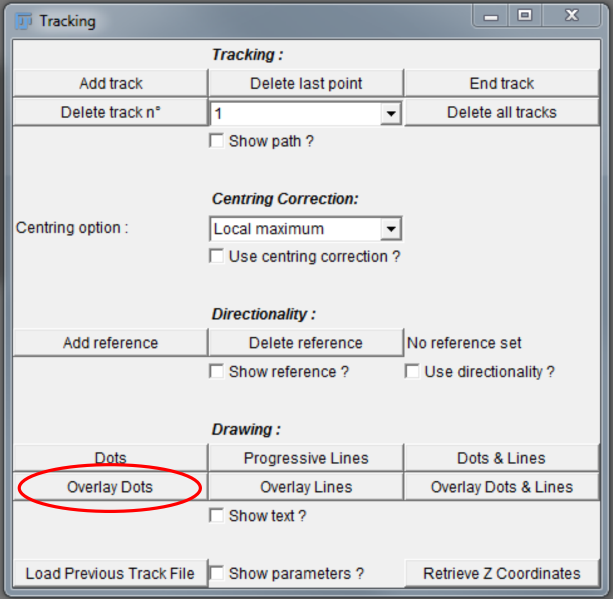
As you end each track, data showing the distance travelled per frame and velocity are added to the results window. This data can be copied and pasted into excel for further analysis



Results from TeNTHc_kd in µm per sec

File Edit Font

	Track n°	Slice n°	X	Y	Distance	Velocity	Pixel Value
1	1	1	26	38	-1.000	-1.000	255
2	1	2	98	47	10.158	2.032	255
3	1	3	148	57	7.139	1.428	255
4	1	4	218	65	9.864	1.973	255
5	1	5	286	72	9.570	1.914	255
6	1	6	380	67	13.179	2.636	96
7	1	7	438	63	8.139	1.628	147
8	1	8	509	56	9.988	1.998	255
9	2	10	14	33	-1.000	-1.000	255
10	2	11	142	55	18.183	3.637	255
11	2	12	219	63	10.838	2.168	255
12	2	13	310	69	12.768	2.554	255
13	2	14	408	64	13.738	2.748	255
14	2	15	506	56	13.766	2.753	255
15	2	30	50	43	63.866	12.773	255
16	2	31	103	54	7.578	1.516	255
17	2	32	147	60	6.217	1.243	239



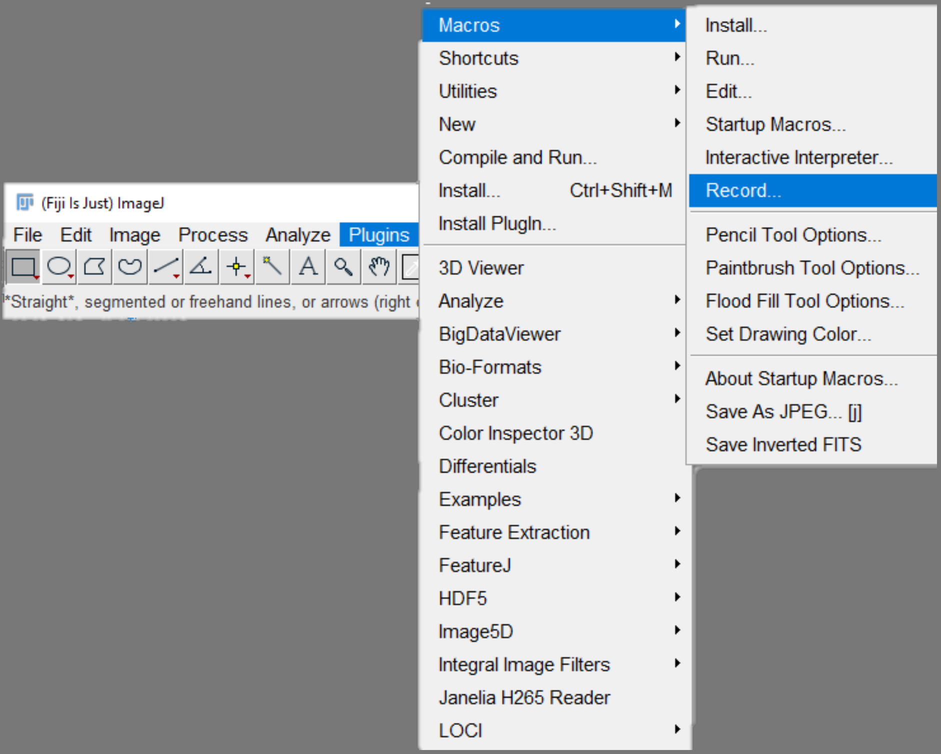
You can review the particles that you have tracked by clicking on the "Overlay Dots" function.

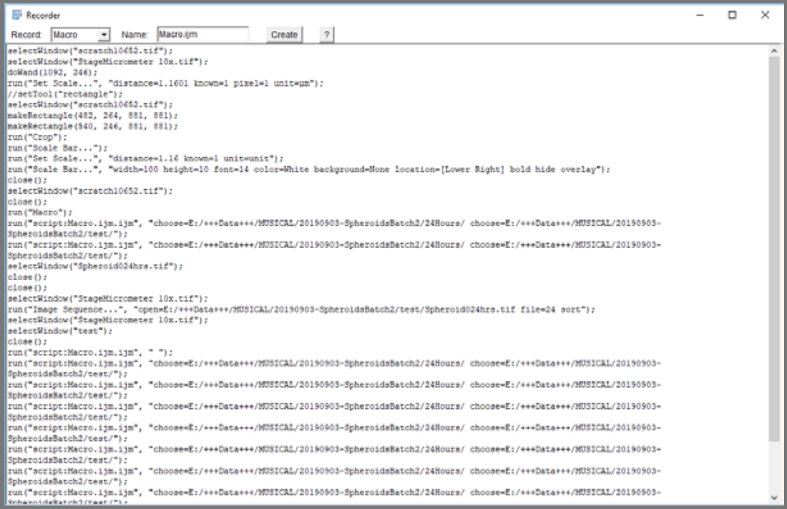
Macros

Macros are a series of code steps used to process images in a quick and repeatable way

ImageJ macros are based on java but you don't need to know java to create one

The macro recorder tool allows you to easily turn image analysis steps into a script





```
selectWindow("scratch10652.tif");
selectWindow("StageMicrometer 10x.tif");
doWand(1092, 246);
run("Set Scale...", "distance=1.1601 known=1 pixel=1 unit=µm");
//setTool("rectangle");
selectWindow("scratch10652.tif");
makeRectangle(482, 264, 881, 881);
makeRectangle(540, 246, 881, 881);
run("Crop");
run("Scale Bar...");
run("Set Scale...", "distance=1.16 known=1 unit=unit");
run("Scale Bar...", "width=100 height=10 font=14 color=White background=None location=[Lower Right] bold hide overlay");
close();
selectWindow("scratch10652.tif");
close();
run("Macro");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
selectWindow("Spheroid024hrs.tif");
close();
close();
selectWindow("StageMicrometer 10x.tif");
run("Image Sequence...", "open=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/test/Spheroid024hrs.tif file=24 sort");
selectWindow("StageMicrometer 10x.tif");
selectWindow("test");
close();
run("script:Macro.ijm.ijm", " ");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
run("script:Macro.ijm.ijm", "choose=E:/+++Data+++/MUSICAL/20190903-SpheroidsBatch2/24Hours/ choose=E:/+++Data+++/MUSICAL/20190903-
SpheroidsBatch2/test/");
```

Macros

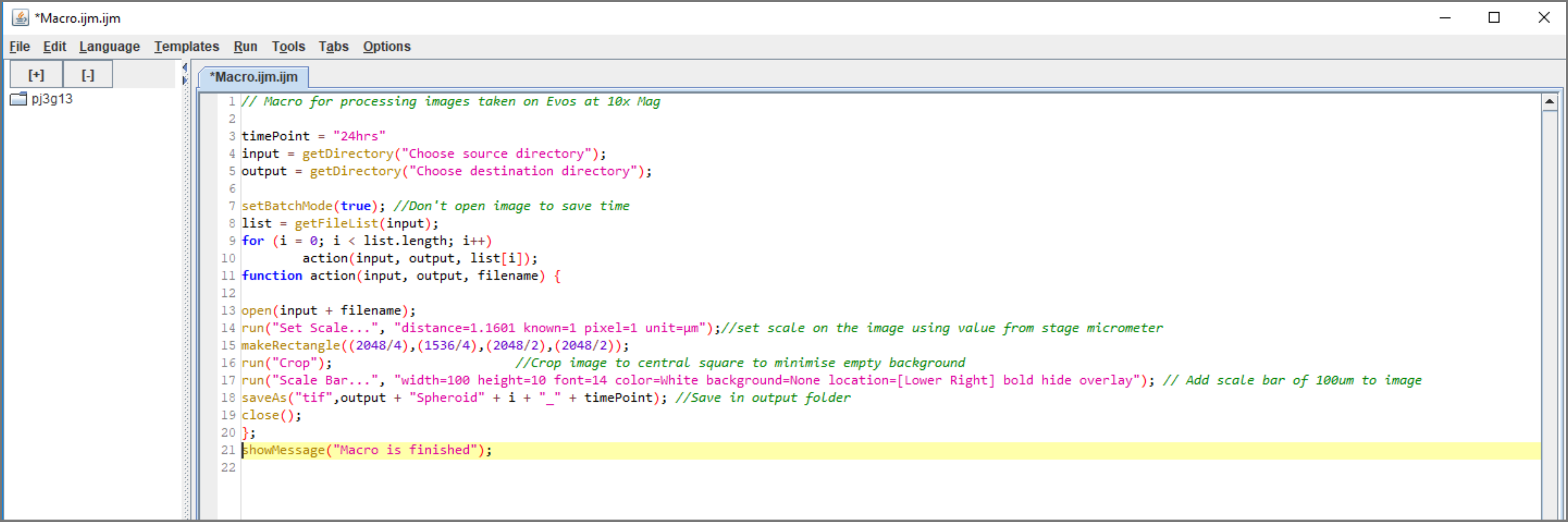
To create a new macro go to Plugins > New > Macro

You can copy and paste step from the recorder into the macro

They can be saved as macro files or as text documents

An example macro

This macro takes a folder of images, crops them, scales them, adds a scale bar and saves them into a new folder with meaningful names.



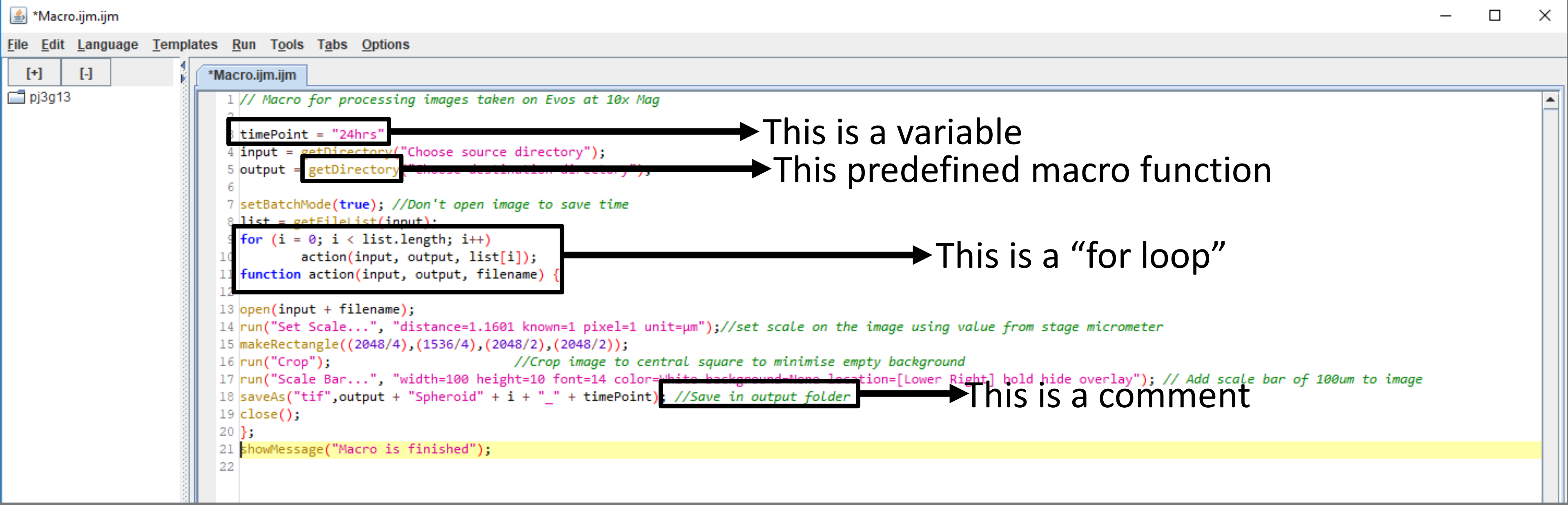
```
// Macro for processing images taken on Evos at 10x Mag

timePoint = "24hrs"
input = getDirectory("Choose source directory");
output = getDirectory("Choose destination directory");

setBatchMode(true); //Don't open image to save time
list = getFileList(input);
for (i = 0; i < list.length; i++)
        action(input, output, list[i]);
function action(input, output, filename) {

open(input + filename);
run("Set Scale...", "distance=1.1601 known=1 pixel=1 unit=µm");//set scale on the image using value from stage micrometer
makeRectangle((2048/4),(1536/4),(2048/2),(2048/2));
run("Crop");                            //Crop image to central square to minimise empty background
run("Scale Bar...", "width=100 height=10 font=14 color=White background=None location=[Lower Right] bold hide overlay"); // Add scale bar of 100um to image
saveAs("tif",output + "Spheroid" + i + "_" + timePoint); //Save in output folder
close();
};
showMessage("Macro is finished");

```



```
// Macro for processing images taken on Evos at 10x Mag

timePoint = "24hrs"
input = getDirectory("Choose source directory");
output = getDirectory("Choose destination directory");

setBatchMode(true); //Don't open image to save time
list = getFileList(input);
for (i = 0; i < list.length; i++)
        action(input, output, list[i]);
function action(input, output, filename) {

open(input + filename);
run("Set Scale...", "distance=1.1601 known=1 pixel=1 unit=µm");//set scale on the image using value from stage micrometer
makeRectangle((2048/4),(1536/4),(2048/2),(2048/2));
run("Crop");                    //Crop image to central square to minimise empty background
run("Scale Bar...", "width=100 height=10 font=14 color=White background=None location=[Lower Right] bold hide overlay"); // Add scale bar of 100um to image
saveAs("tif",output + "Spheroid" + i + "_" + timePoint); //Save in output folder
close();
};
showMessage("Macro is finished");
```

Variables can be numbers, strings, arrays etc their value can change which is useful

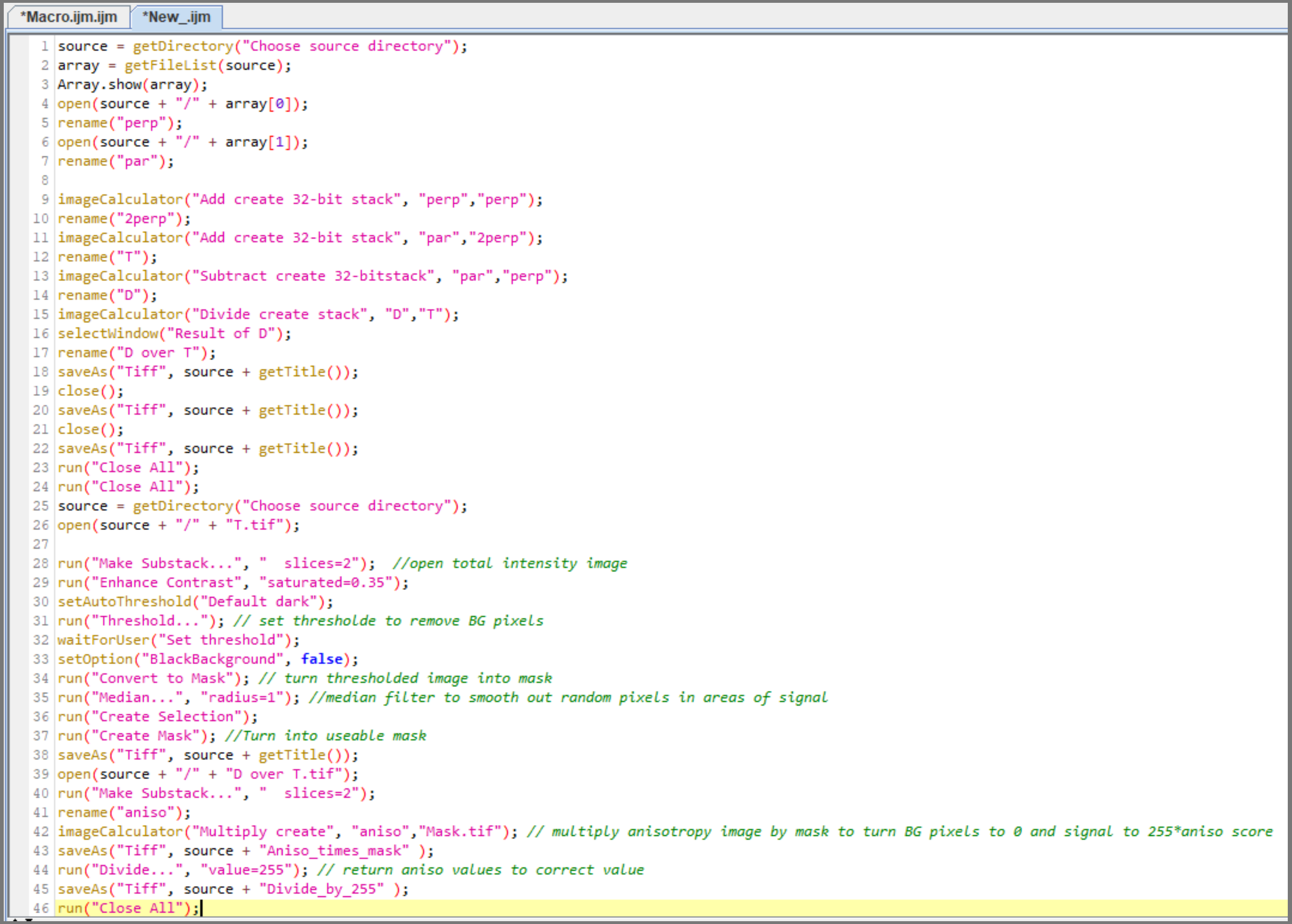
ImageJ has hundreds of predefined macro functions that streamline the process of image analysis, you can find a list Googling "ImageJ macro functions"

For Loops, allow you to repeat the same set of commands for multiple images without needing outside input

Comments are bits of code that the computer doesn't read and are marked with a "//", commenting macros is useful to remind yourself what each step is doing

Macros

An example macro for analysis of images



```
source = getDirectory("Choose source directory");
array = getFileList(source);
Array.show(array);
open(source + "/" + array[0]);
rename("perp");
open(source + "/" + array[1]);
rename("par");

imageCalculator("Add create 32-bit stack", "perp","perp");
rename("2perp");
imageCalculator("Add create 32-bit stack", "par","2perp");
rename("T");
imageCalculator("Subtract create 32-bitstack", "par","perp");
rename("D");
imageCalculator("Divide create stack", "D","T");
selectWindow("Result of D");
rename("D over T");
saveAs("Tiff", source + getTitle());
close();
saveAs("Tiff", source + getTitle());
close();
saveAs("Tiff", source + getTitle());
run("Close All");
run("Close All");
source = getDirectory("Choose source directory");
open(source + "/" + "T.tif");

run("Make Substack...", "  slices=2");  //open total intensity image
run("Enhance Contrast", "saturated=0.35");
setAutoThreshold("Default dark");
run("Threshold..."); // set thresholde to remove BG pixels
waitForUser("Set threshold");
setOption("BlackBackground", false);
run("Convert to Mask"); // turn thresholded image into mask
run("Median...", "radius=1"); //median filter to smooth out random pixels in areas of signal
run("Create Selection");
run("Create Mask"); //Turn into useable mask
saveAs("Tiff", source + getTitle());
open(source + "/" + "D over T.tif");
run("Make Substack...", "  slices=2");
rename("aniso");
imageCalculator("Multiply create", "aniso","Mask.tif"); // multiply anisotropy image by mask to turn BG pixels to 0 and signal to 255*aniso score
saveAs("Tiff", source + "Aniso_times_mask" );
run("Divide...", "value=255"); // return aniso values to correct value
saveAs("Tiff", source + "Divide_by_255" );
run("Close All");
```