Introduction to **Image Processing** and **Object Segmentation** using **Fiji/ImageJ**

Verónica Labrador Cantarero (veronica.labrador@cnic.es)
Servicio de Microscopía Óptica y Confocal (SMOC)
Centro de Biología Molecular Severo Ochoa (CSIC-UAM)

Image Processing

Image Restoration

Deconvolution

Image Enhancement

Look up tables
Brightness&Contrast
Histogram stretching/equalization
Arithmetic Operations
Background /Shading Corrections
Filtering

- Convolution
- Noise Reduction and Smoothing
- Feature Enhancement
- Frequency Domain

Other Processing

Segmentation, Stitching, Color Registration, Tracking, Particle Analysis

Image Enhancement

1. Image Enhancement

What for?

Enhances visual appearance of image features

Helps to reduce noise

Sometimes is needed before image segmentation

Attention!

It modifies pixels values, use with caution!

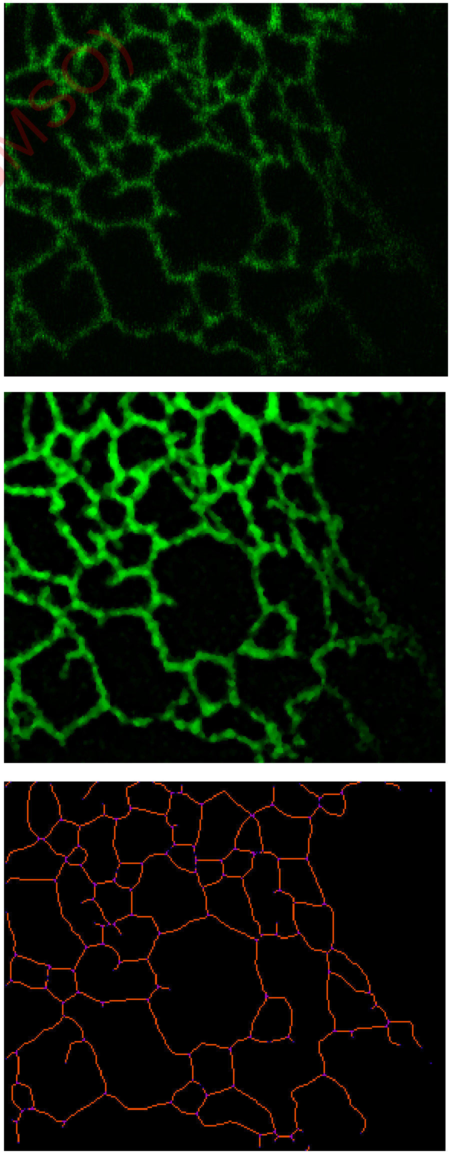
Final image must represent the original information

Always use a copy of the original file

For publication, manipulations should be indicated

Check journal guidelines regarding image processing

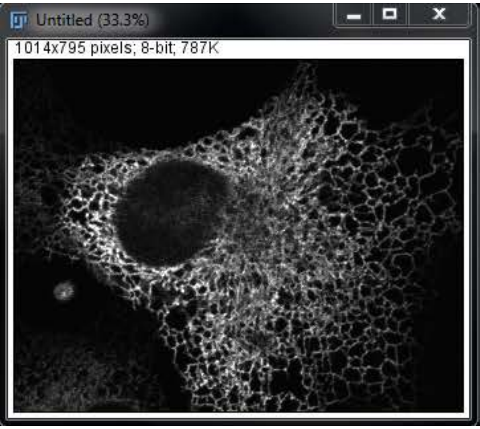
Ethics in image processing!

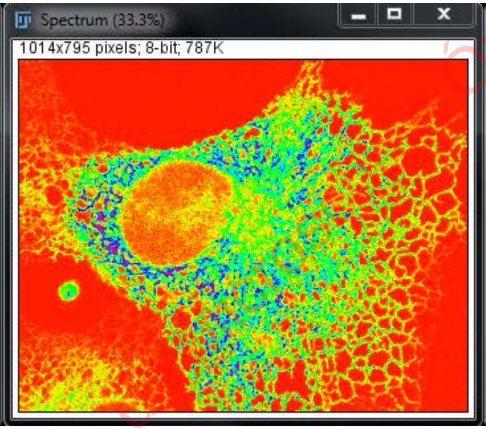


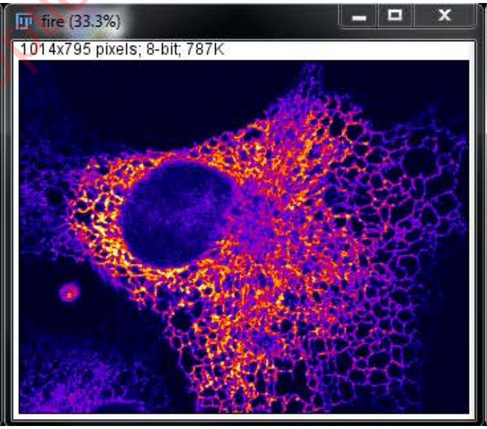
1.1 Lookup Tables

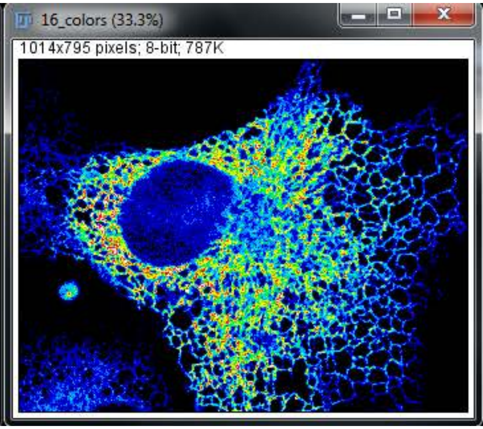
Image>Lookup tables

Easiest way to emphasize differences in intensity
Only modifies image appearance, not pixel values


Untitled (33.3%)
1014x795 pixels; 8-bit; 787K


Spectrum (33.3%)
1014x795 pixels; 8-bit; 787K


fire (33.3%)
1014x795 pixels; 8-bit; 787K


16_colors (33.3%)
1014x795 pixels; 8-bit; 787K

Have fun and create your own Lookup Table! **Image>Color>Edit LUT**

1.2 Brightness&Contrast. Levels. Gamma

Linear Adjustments

Image>Adjust>Window/Level

Image>Adjust>Brightness&Contrast

Linear relationship between image intensity and display intensity

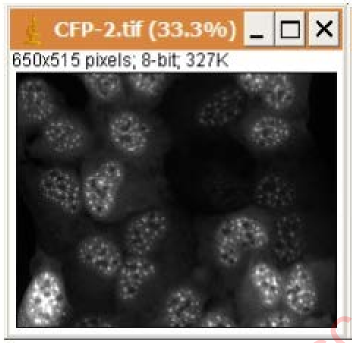


Illustration 11: The low intensities are invisible in the original image.

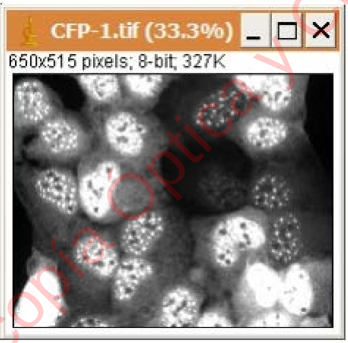


Illustration 12: A linear display adjustment makes low intensities visible but saturates high intensities.

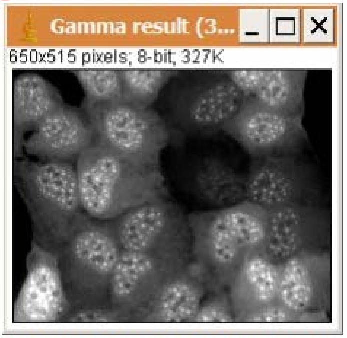


Illustration 16: The gamma-correction has been applied to the image.

Non-Linear Adjustments: Gamma

Process>Math>Gamma…

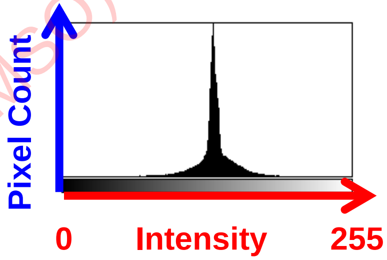
Changes in pixel values depend on a non-linear function.

Useful to enhance low pixel values without saturating high ones

1.3 Histogram Adjustment I

Histogram: Number of pixels at each intensity value

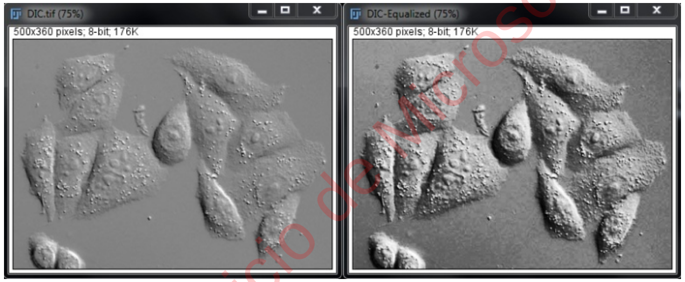
Analyze>Histogram

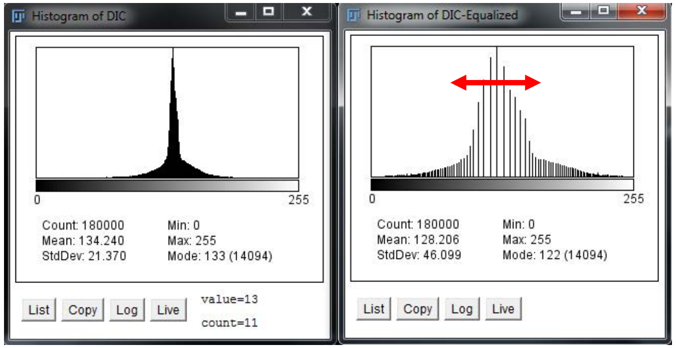


(Global) Histogram Equalization

Process>Enhance Contrast>Equalize Histogram

Attempts to perform an equally redistribution of pixel values along the image range





```
Count: 180000      Min: 0
Mean: 134.240      Max: 255
StdDev: 21.370     Mode: 133 (14094)
                   value=13
                   count=11
```

```
Count: 180000      Min: 0
Mean: 128.206      Max: 255
StdDev: 46.099     Mode: 122 (14094)
```

Spreads out the most frequent intensity values

Regions of Interest (ROIs) can be used to select image areas before equalization

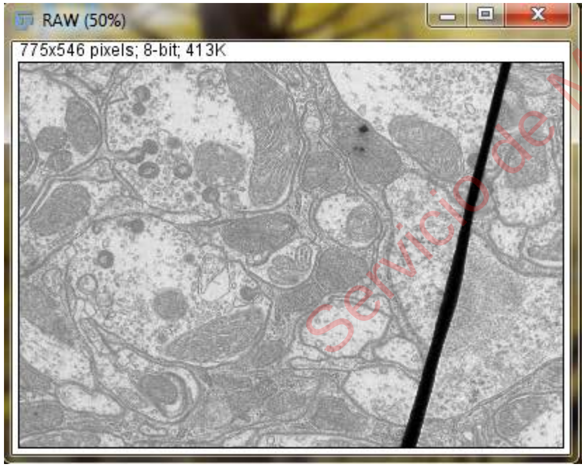
1.3 Histogram Adjustment II

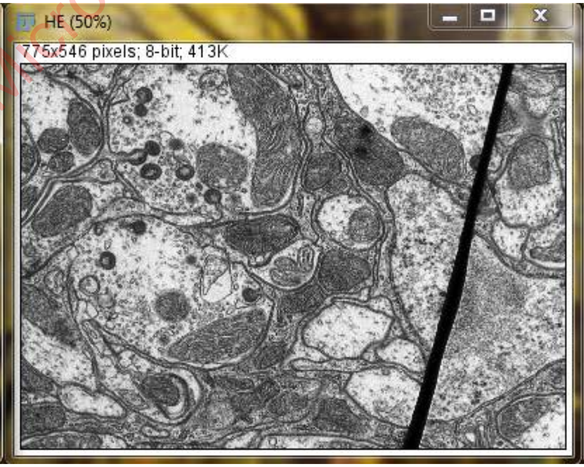
CLAHE-Contrast Limited Adaptive Histogram Equalization

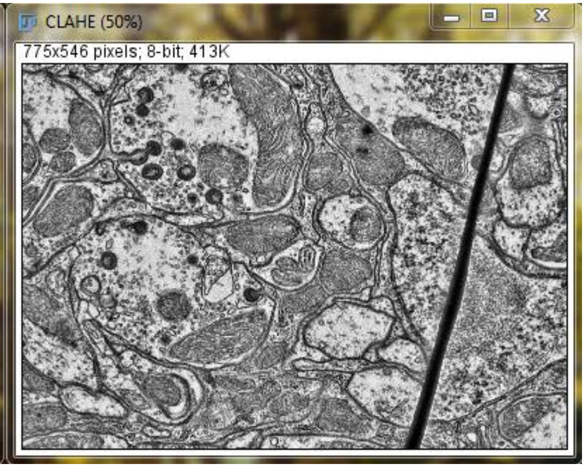
Process>Enhance Local Contrast (CLAHE)

Local Histogram Equalization

Does not apply histogram equalization on the whole image but on image blocks







1.3 Histogram Adjustment III

Histogram Normalization (Stretching)

Process>Enhance Contrast>Normalize

Normalizes histogram to maximum and minimum intensity values

Helps to establish the same intensity threshold between images with different intensity levels (for example, due to photobleaching).

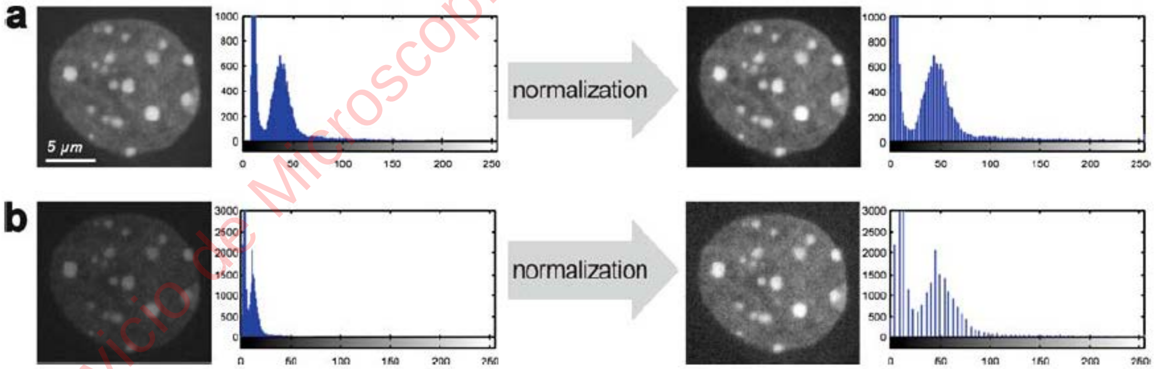


Figure 2. Intensity normalization. (a) Confocal image (8-bit format) of a mouse fibroblast nucleus stained with T0-PRO3 to show chromocenters. (b) The same image after simulated bleaching. The blue graphs show intensity histograms. Owing to nearly proportional fluorophore intensities in these images (as is also the case for natural bleaching e.g. due to image acquisition), intensity normalization improves both images. After this normalization, corresponding structures have similar gray values. This allows, for example, the same threshold to be applied to all images of the series.

Chromosome Research (2008) 16:523–562

1.4 Arithmetic Operations I

Work on a pixel by pixel manner

With a Constant Value

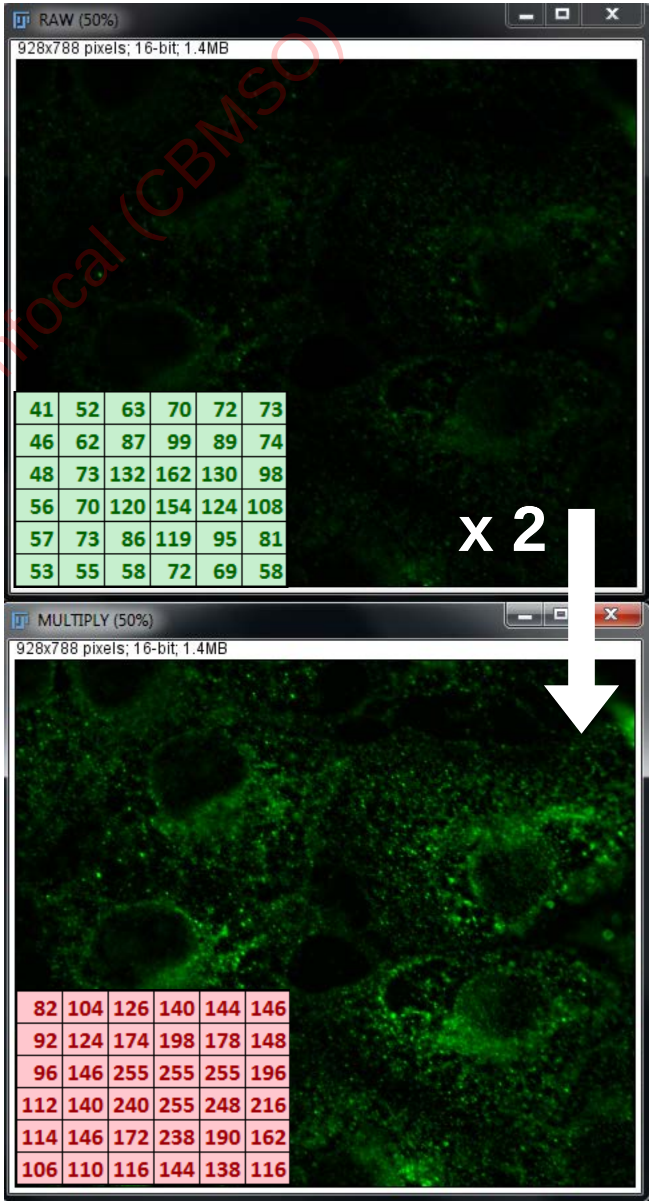
Process>Math

Add, Multiply, Divide

Use to adjust image brightness

Subtract

Use to remove background

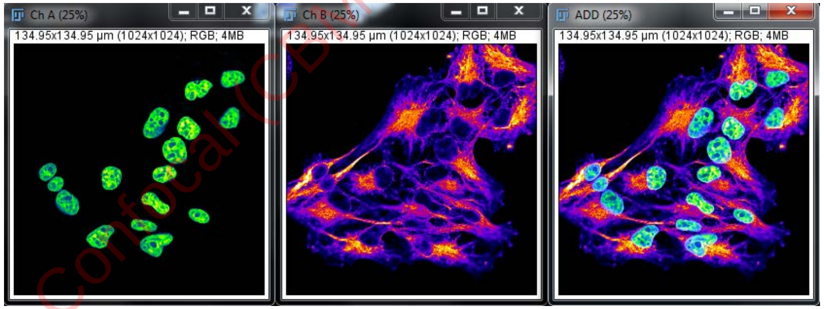


41	52	63	70	72	73
46	62	87	99	89	74
48	73	132	162	130	98
56	70	120	154	124	108
57	73	86	119	95	81
53	55	58	72	69	58

82	104	126	140	144	146
92	124	174	198	178	148
96	146	255	255	255	196
112	140	240	255	248	216
114	146	172	238	190	162
106	110	116	144	138	116

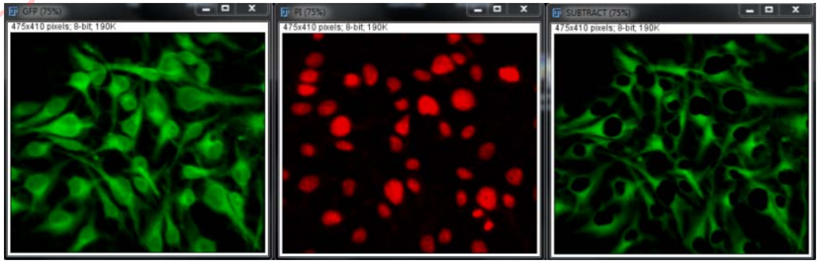
1.4 Arithmetic Operations II

Between two images


Ch A (25%)
134.95x134.95 µm (1024x1024); RGB; 4MB
Ch B (25%)
134.95x134.95 µm (1024x1024); RGB; 4MB
ADD (25%)
134.95x134.95 µm (1024x1024); RGB; 4MB

Process>Image Calculator

Plugin **Calculator Plus**


GFP (75%)
475x410 pixels; 8-bit; 190K
PI (75%)
475x410 pixels; 8-bit; 190K
SUBTRACT (75%)
475x410 pixels; 8-bit; 190K

Add to combine information

Subtract to extract certain image areas

Division to apply shading correction

Average to reduce noise (stack needed)

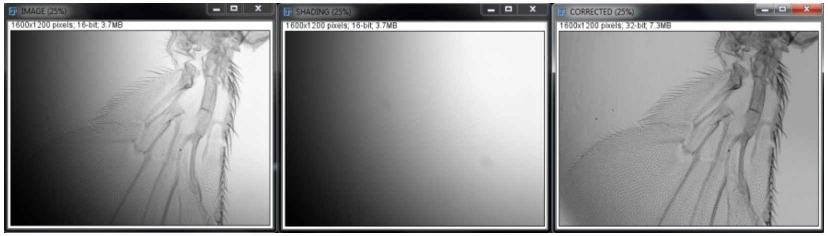


IMAGE (25%)
1600x1200 pixels; 16-bit; 3.7MB
SHADING (25%)
1600x1200 pixels; 16-bit; 3.7MB
CORRECTED (25%)
1600x1200 pixels; 32-bit; 7.3MB

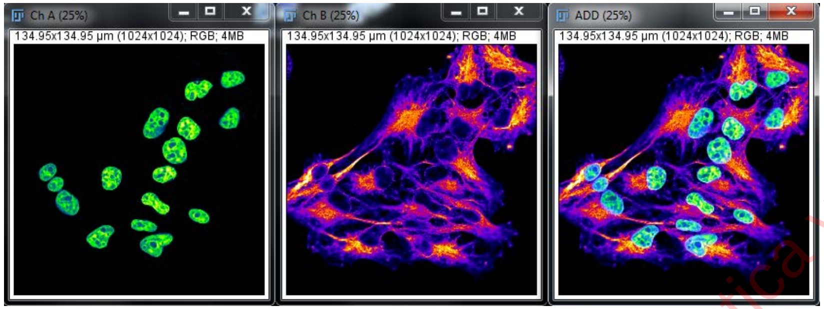
Arithmetic Operations Between 2 images Examples

‘Add’ to combine not usual look up tables

1. Open both files and change Look up tables: **Image>Lookup tables**

2. Go to **Process>Image Calculator** and select **Add** as the operation

3. Numerator image must be RGB. If not, first change it going to **Image>Type> RGB Color**


Ch A (25%)
134.95x134.95 µm (1024x1024); RGB; 4MB
Ch B (25%)
134.95x134.95 µm (1024x1024); RGB; 4MB
ADD (25%)
134.95x134.95 µm (1024x1024); RGB; 4MB

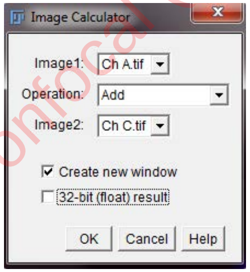


Image1: Ch A.tif

Operation: Add

Image2: Ch C.tif

☑ Create new window

□ 32-bit (float) result

‘Division’ to apply shading correction

1. Open imagen and Shading image (acquire an area of the coverslip without sample with exactly the same illumination settings).

2. Go to **Process>Image Calculator** and select **Divide** as the operation. Use Shading image as Denominator.

3. For proper visual display use the **32-bit(float) result** checkbox

4. Change to 8Bit to reduce image data: **Image>Type> 8-bit**

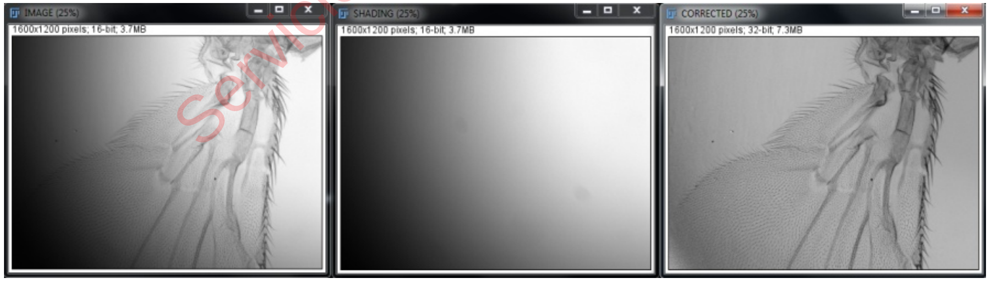


IMAGE (25%)
1600x1200 pixels; 16-bit; 3.7MB
SHADING (25%)
1600x1200 pixels; 16-bit; 3.7MB
CORRECTED (25%)
1600x1200 pixels; 32-bit; 7.3MB

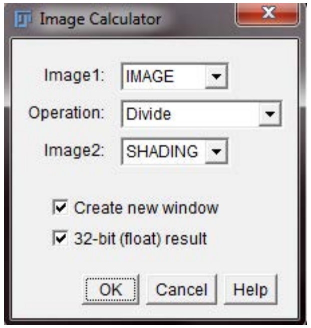


Image1: IMAGE

Operation: Divide

Image2: SHADING

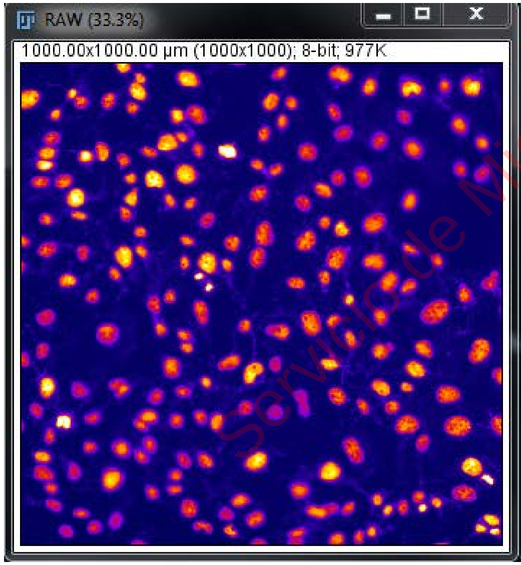
☑ Create new window

☑ 32-bit (float) result

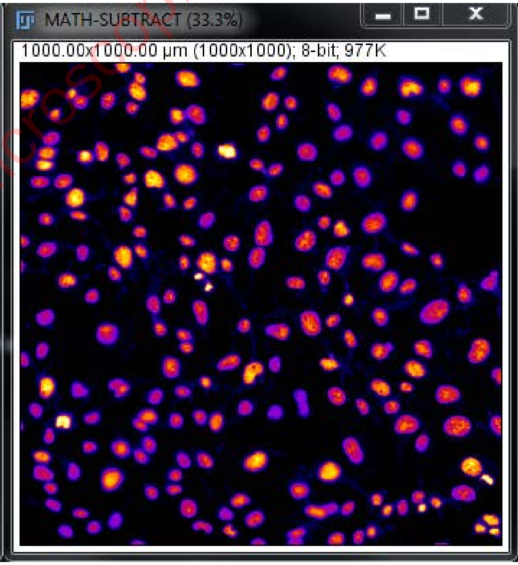
1.5 Background/Shading Corrections I

Even Background

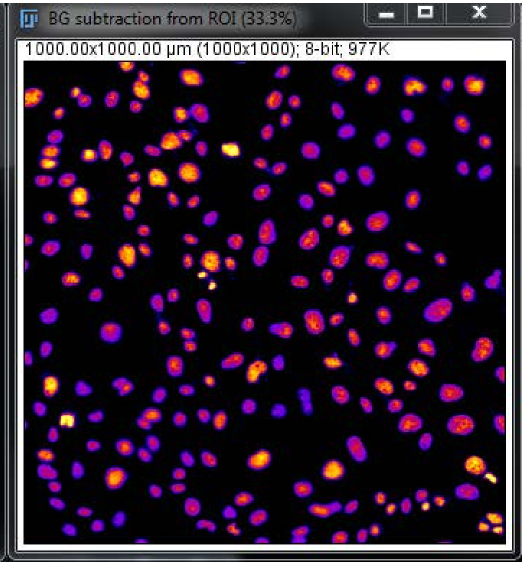
- Subtract average background from image **(Process>Math>Subtract)**
- Macro **BG Subtraction from ROI**



1000.00x1000.00 µm (1000x1000); 8-bit; 977K



1000.00x1000.00 µm (1000x1000); 8-bit; 977K



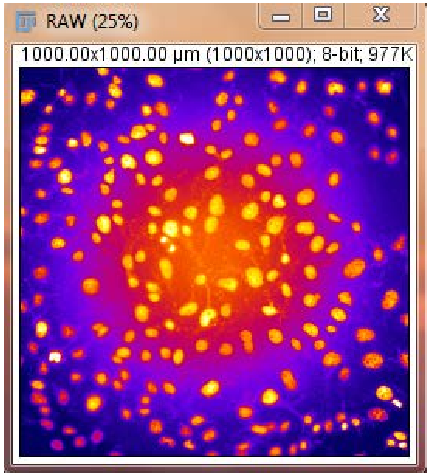
1000.00x1000.00 µm (1000x1000); 8-bit; 977K

1.5 Background/Shading Corrections II

Uneven Background/Shading Correction

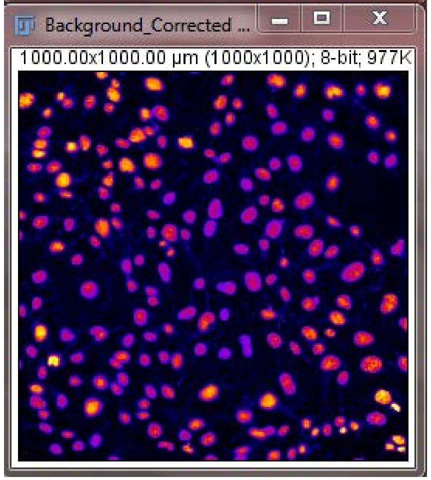
- Rolling ball algorithm **(Process>Subtract Background)**
- Plugin **Fit Polynomial**
- Plugin **Nonuniform Backgroun Removal**
- BandPass filter **(Process>FTT>BandPass filter)**
- Plugin **Shading Corrector** (a flat-field image is needed)
- Divide original image and 'shading image'

Additional information



RAW (25%)

1000.00x1000.00 µm (1000x1000); 8-bit; 977K



Background_Corrected ...

1000.00x1000.00 µm (1000x1000); 8-bit; 977K

Background/Shading Corrections Examples

Rolling Ball algorithm

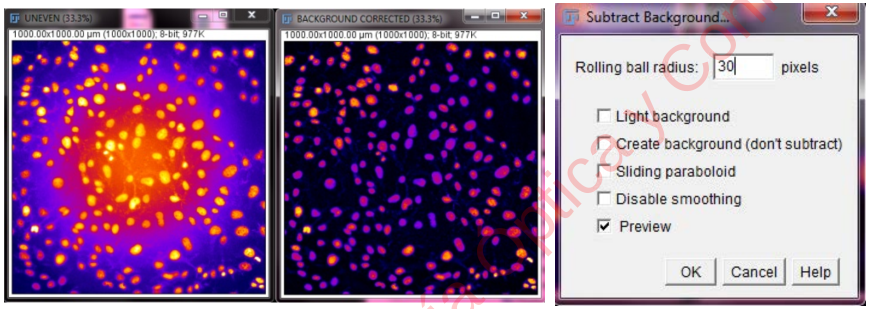
1. Open your file and change appearance for better visualization if necessary: **Image>Lookup tables**

2. Duplicate your original image: **Image>Duplicate…**

3. Apply ImageJ/Fiji background subtraction tool on you duplicate: Go to **Process>Subtract Background**

Information on this tool can be found here: http://rsb.info.nih.gov/ij/docs/guide/146-29.html#toc-Subsection-29.14

For this example, a Radius of 30 has been used.


UNEVEN (33.3%)
1000.00x1000.00 µm (1000x1000); 8-bit; 977K
BACKGROUND CORRECTED (33.3%)
1000.00x1000.00 µm (1000x1000); 8-bit; 977K
Subtract Background...

Rolling ball radius: 30 pixels

□ Light background

□ Create background (don't subtract)

□ Sliding paraboloid

□ Disable smoothing

☑ Preview

Fit Polynomial Plugin

1. Download and install **Fit Polynomial** plugin.

2. If necessary, open your file and change appearance for better visualization: **Image>Lookup tables**

3. Duplicate your original image: **Image>Duplicate…**

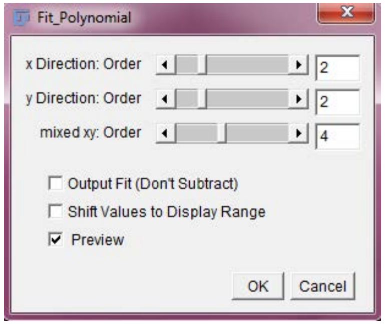
4. Open **Fit Polynomial** tool: **Plugins>Fit Polynomial**

Information on this tool can be found here:

http://imagejdocu.tudor.lu/doku.php?id=plugin:filter:fit_polynomial:start

Move the sliders until a good correction is got

This plugin works well for transmitted and DIC images correction



x Direction: Order 2

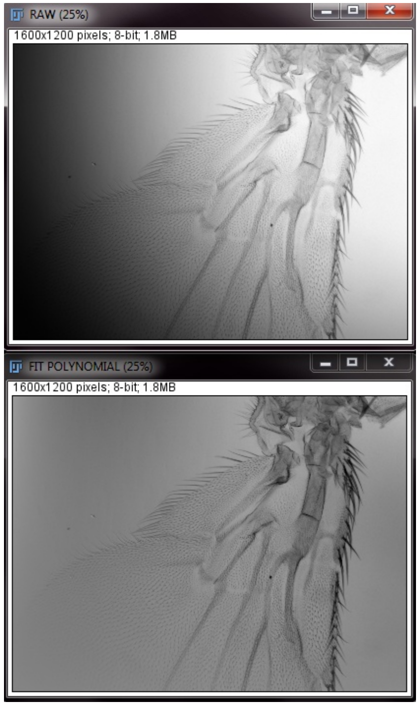
y Direction: Order 2

mixed xy: Order 4

□ Output Fit (Don't Subtract)

□ Shift Values to Display Range

☑ Preview

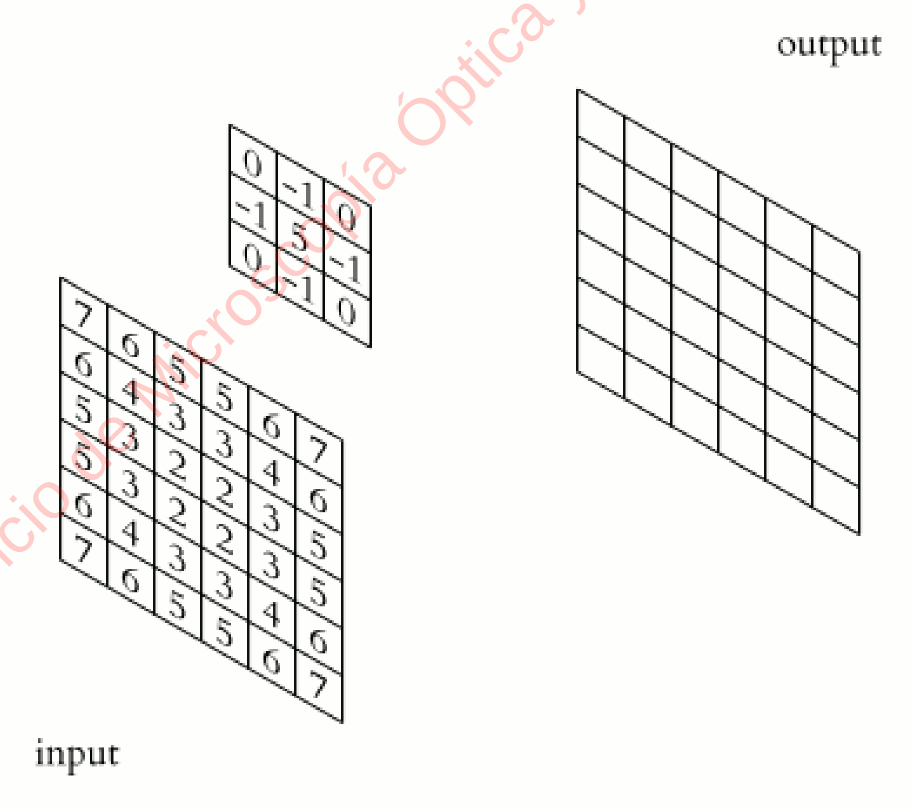

RAW (25%)
1600x1200 pixels; 8-bit; 1.8MB
FIT POLYNOMIAL (25%)
1600x1200 pixels; 8-bit; 1.8MB

1.7 Filtering

Involves a group of neigbour pixels to create a new pixel value

Used to smooth the image, remove noise or feature/edges enhancement

Filters apply an algorithm or a convolution kernel



0	-1	0
-1	5	-1
0	-1	0

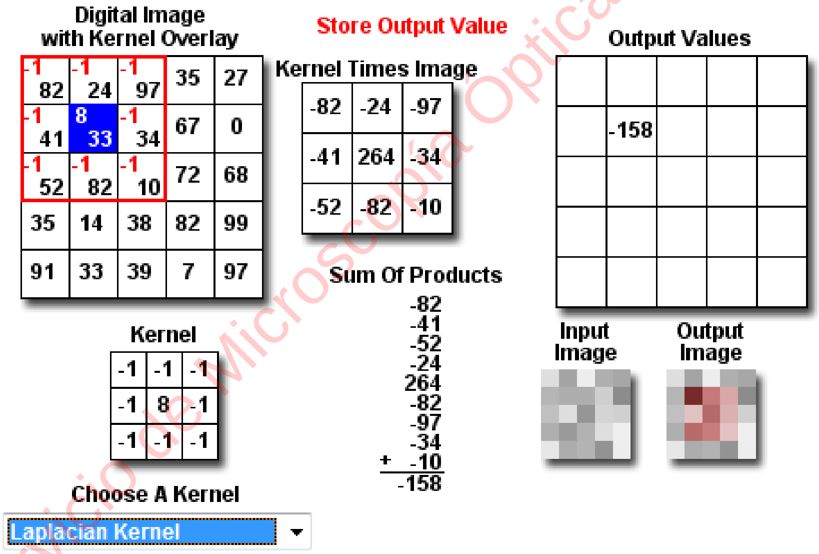
7	6	5	5	6	7
6	4	3	3	4	6
5	3	2	2	3	5
5	3	2	2	3	5
6	4	3	3	4	6
7	6	5	5	6	7

1.7.1 Convolution

Mathematical operation applied to a group of pixels

Intensity of each output pixel is a function of the intensity values of its neighbours

Size of the group of pixels involved depends on size of the **Convolution Kernel**, a square array, usually 3x3, 5x5, 7x7, with different numerical values



Digital Image with Kernel Overlay

-1 82	-1 24	-1 97	35	27
-1 41	8 33	-1 34	67	0
-1 52	-1 82	-1 10	72	68
35	14	38	82	99
91	33	39	7	97

Store Output Value

Kernel Times Image

-82	-24	-97
-41	264	-34
-52	-82	-10

Sum Of Products

```
  -82
  -41
  -52
  -24
  264
  -82
  -97
  -34
+ -10
 -158
```

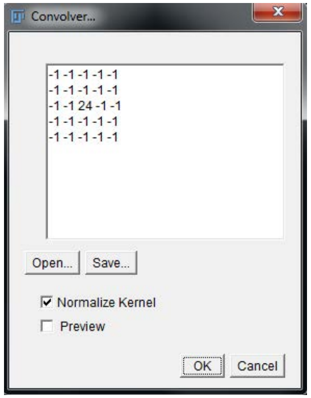
Output Values

	-158			

Kernel

-1	-1	-1
-1	8	-1
-1	-1	-1

Choose A Kernel: Laplacian Kernel



Convolver...

```
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 24 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
```

☑ Normalize Kernel
□ Preview

Create your own convolution kernel: **Process>Filters>Convolve**

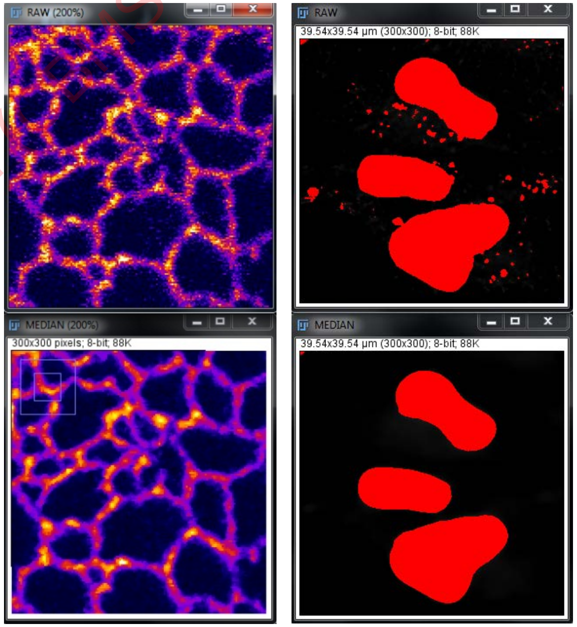
1.7.2 Denoising and Smoothing

Filters….What for?

To reduce noise to improve image quality

To facilitate other processing such as thresholding

To remove small isolated objects


RAW (200%)
RAW
39.54x39.54 µm (300x300); 8-bit; 88K
MEDIAN (200%)
300x300 pixels; 8-bit; 88K
MEDIAN
39.54x39.54 µm (300x300); 8-bit; 88K

Gaussian Blur filter

Process>Filters>Gaussian Blur…/Gaussian Blur 3D…

Mean Filter

Process>Filters>Mean…/Mean 3D…

Median Filter

Process>Filters>Median…/Median 3D…

Does not smooth object edges as much as Gaussian filter or Mean Filter

Process>Noise>Despeckle (Median 3x3)

1.7.2 …Some other denoising tools

Anisotropic Diffusion Filter

http://fiji.sc/Anisotropic_Diffusion_2D

Sigma Filter Plus

http://rsbweb.nih.gov/ij/plugins/sigma-filter.html

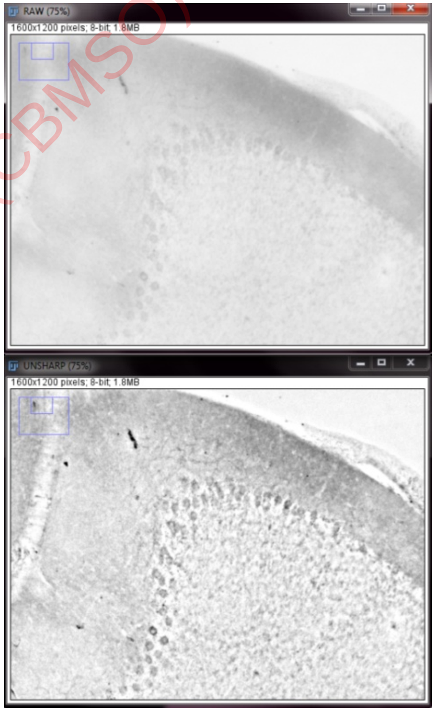
Non-Local Means Denoising

https://code.google.com/p/ij-non-local-means/

Kuwahara Filter

http://rsbweb.nih.gov/ij/plugins/kuwahara.html

1.7.3 Feature Enhancement


RAW (75%)
1600x1200 pixels; 8-bit; 1.8MB
UNSHARP (75%)
1600x1200 pixels; 8-bit; 1.8MB

Filters…What for?

To highlight edges or fine details from images

Spot detection

Sharpen Filter

Process>Sharpen

Unsharp Mask

Process>Filters>Unsharp Mask…

Laplacian Filter

Plugins>Feature Extraction>FeatureJ>FeatureJ Laplacian

LoG (Laplacian of Gaussian)

Plugin LoG3D

More information about Edge Detection in **Segmentation** section

Spot Detection example I

1- Download and install **LoG3D** plugin.

http://bigwww.epfl.ch/sage/soft/LoG3D/

2- Open your file and change appearance for better visualization if necessary: **Image>Lookup tables**

3- Duplicate your original image: **Image>Duplicate…**

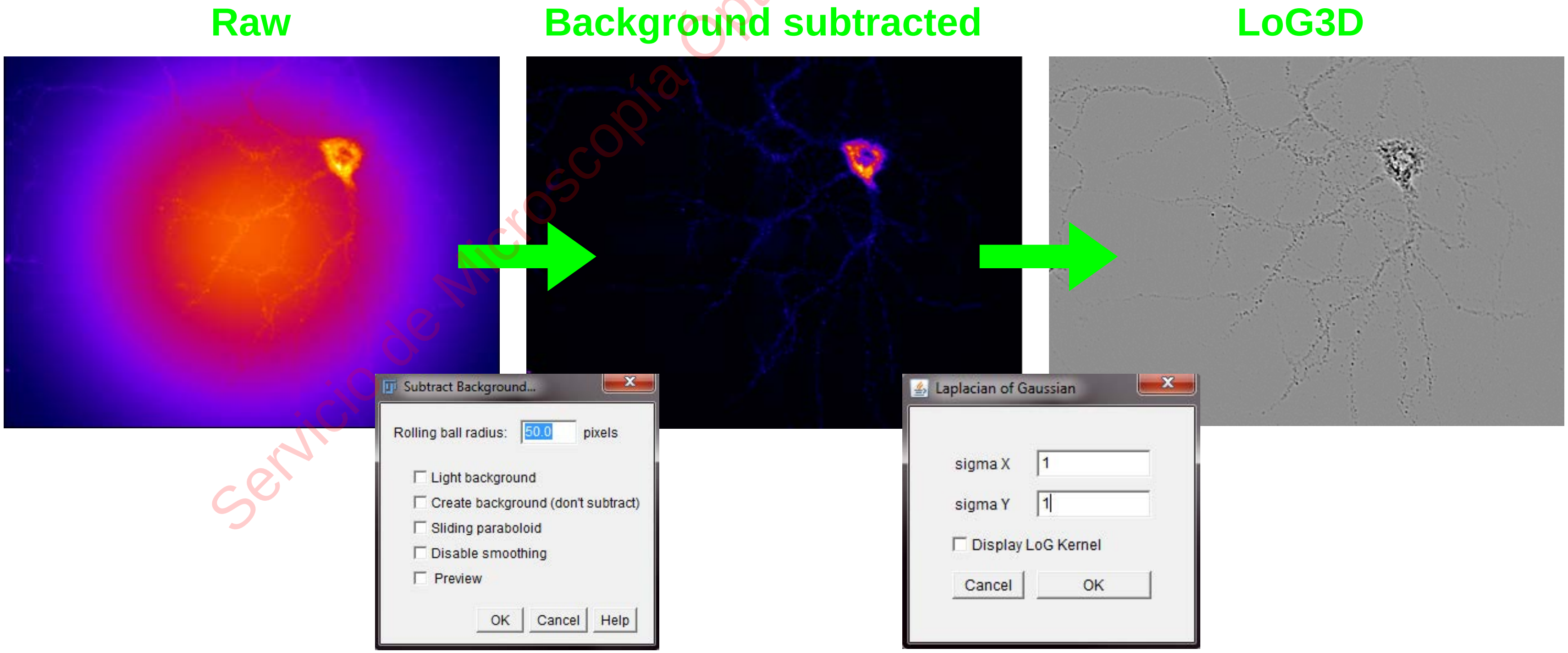
4- Apply ImageJ/Fiji background subtraction tool on your replica: Go to **Process>Subtract Background**

Information on this tool can be found here: http://rsb.info.nih.gov/ij/docs/guide/146-29.html#toc-Subsection-29.14

For this example, a Radius of 50 has been used.

5- Apply **LoG3D plugin** on your background-corrected image for spot detection: **Plugins>LoG3D**

For this example, a sigma X and Y of 1 was used.



Rolling ball radius: 50.0 pixels

☐ Light background
☐ Create background (don't subtract)
☐ Sliding paraboloid
☐ Disable smoothing
☐ Preview

sigma X 1

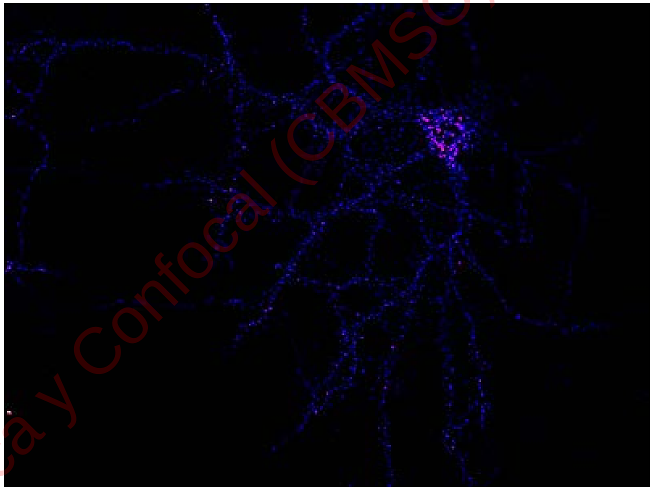
sigma Y 1

☐ Display LoG Kernel

Spot Detection example II

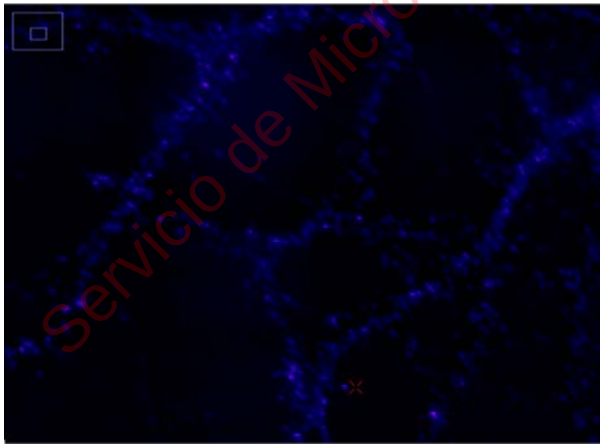
6- For better visualization of result image change lookup table , Invert lookup table and modify Brightness&Contrast:

Image>Lookup tables
Image>Lookup tables>Invert LUT
Image>Adjust>Brightness/Contrast

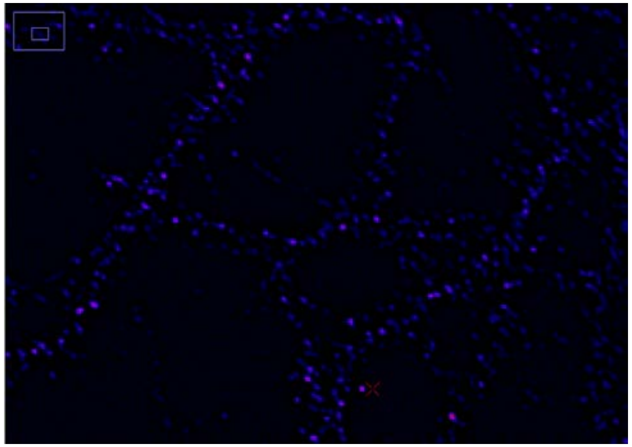


7- To compare both images (filtered and original), use the **Synchronize windows** tool (**Analyze>Tools**)

Raw



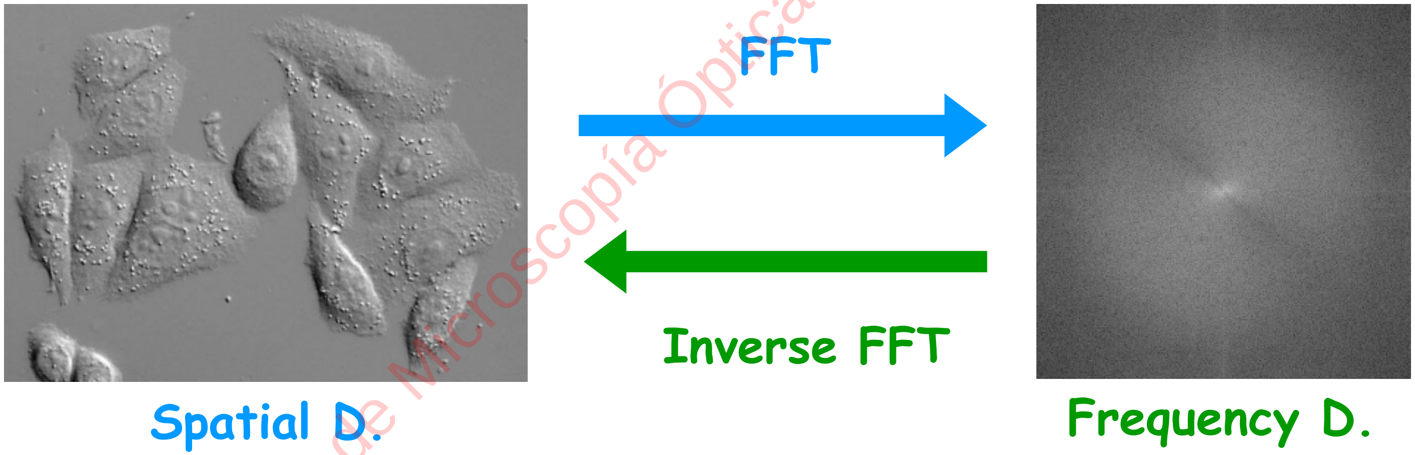
Filtered



1.7.4 Filtering in the Frequency Domain

Microscopy images are in the Spatial Domain (1pixel=1intensity value)

FFT algorithm (Fast Fourier Transform) is used to transform images from Spatial to Frequency Domain (amplitudes and frequencies sines/cosines that added up will give the image)



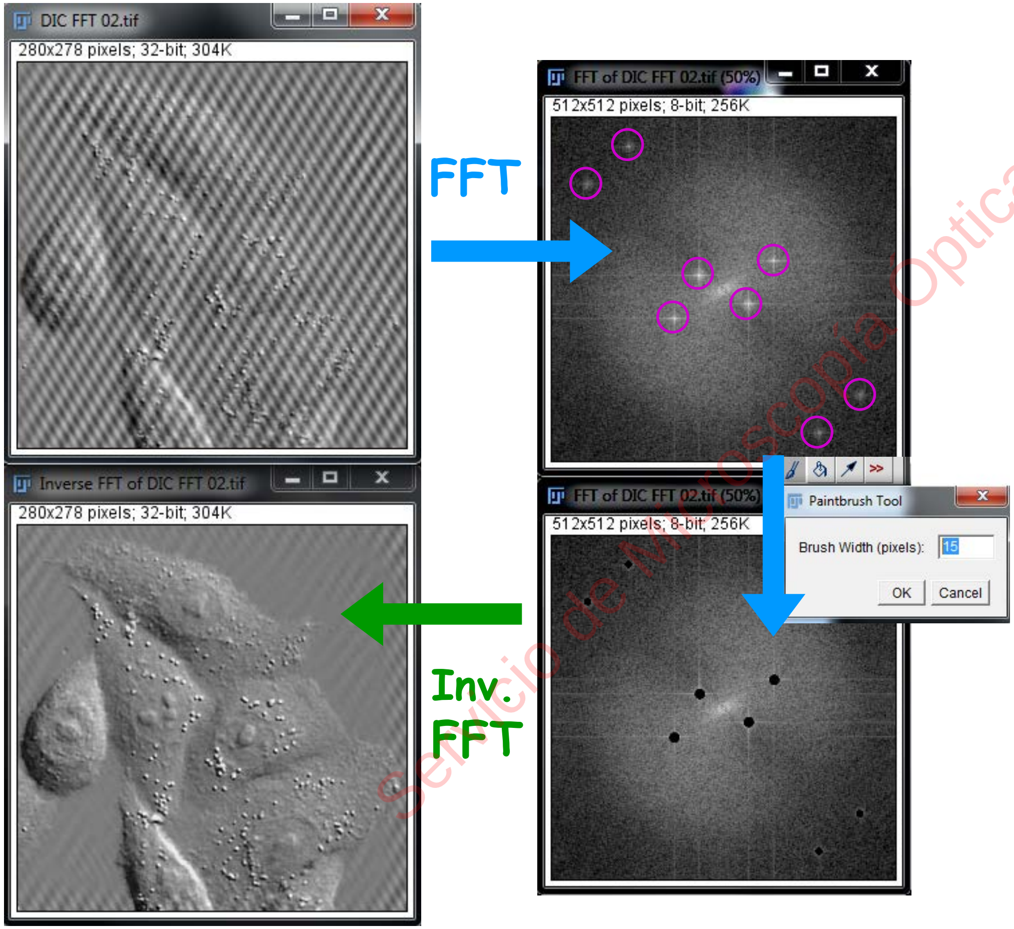
Process>FFT>FFT (Spatial domain to Frequency domain)
Process>FFT>Inverse FFT(Frequency domain to Spatial domain)

1.7.4 Filtering in the Frequency Domain

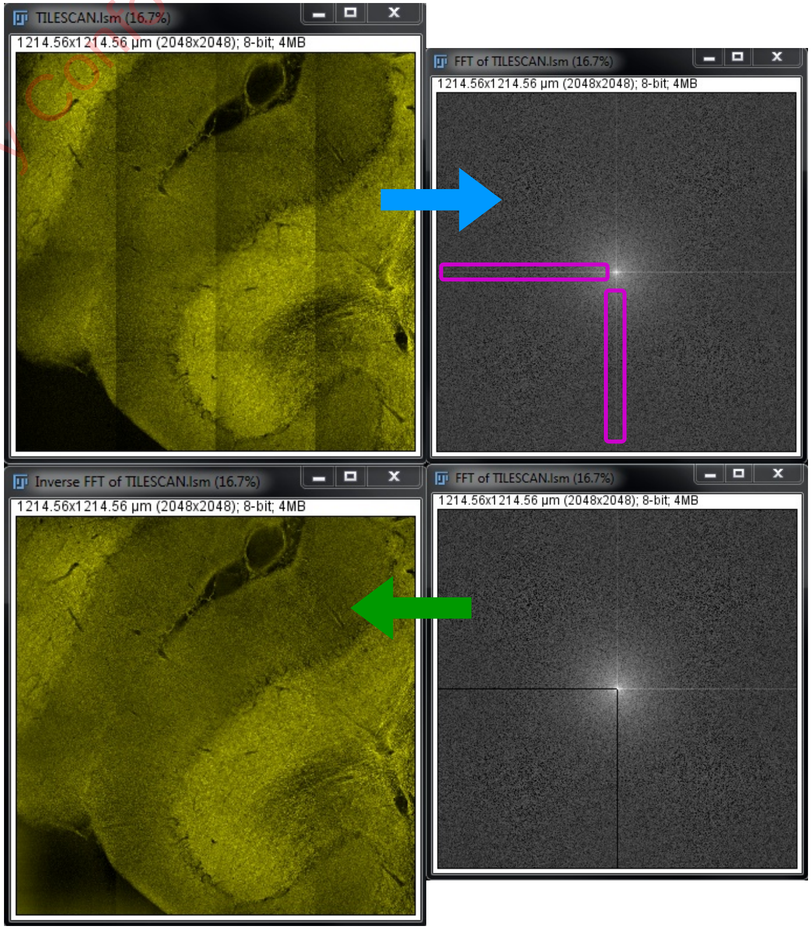
What for?

Correction of Periodic Artifacts

Each image requires a different approach (Sometimes is not easy to find)


DIC FFT 02.tif
280x278 pixels; 32-bit; 304K
FFT
FFT of DIC FFT 02.tif (50%)
512x512 pixels; 8-bit; 256K
FFT of DIC FFT 02.tif (50%)
512x512 pixels; 8-bit; 256K
Paintbrush Tool
OK
Cancel
Inverse FFT of DIC FFT 02.tif
280x278 pixels; 32-bit; 304K
Inv. FFT

Brush Width (pixels): 15

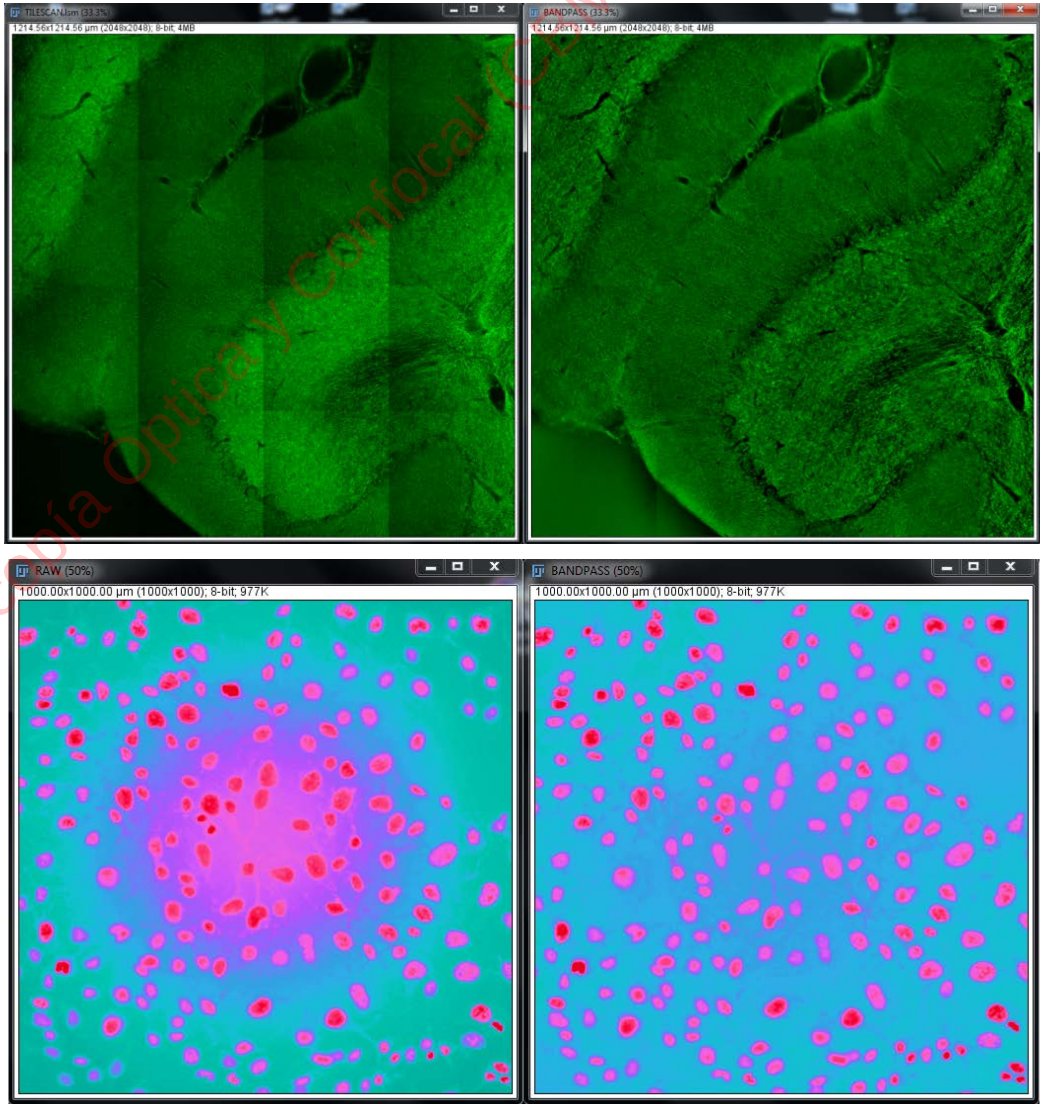

TILESCAN.lsm (16.7%)
1214.56x1214.56 µm (2048x2048); 8-bit; 4MB
FFT of TILESCAN.lsm (16.7%)
1214.56x1214.56 µm (2048x2048); 8-bit; 4MB
Inverse FFT of TILESCAN.lsm (16.7%)
1214.56x1214.56 µm (2048x2048); 8-bit; 4MB
FFT of TILESCAN.lsm (16.7%)
1214.56x1214.56 µm (2048x2048); 8-bit; 4MB

1.7.4 Filtering in the Frequency Domain II

BandPass Filter

Process>FFT>Bandpass Filter

Works in the Fourier domain


TILESCAN.lsm (33.3%)
1214.56x1214.56 µm (2048x2048); 8-bit; 4MB
BANDPASS (33.3%)
1214.56x1214.56 µm (2048x2048); 8-bit; 4MB
RAW (50%)
1000.00x1000.00 µm (1000x1000); 8-bit; 977K
BANDPASS (50%)
1000.00x1000.00 µm (1000x1000); 8-bit; 977K

What for?

Suppress Stripes

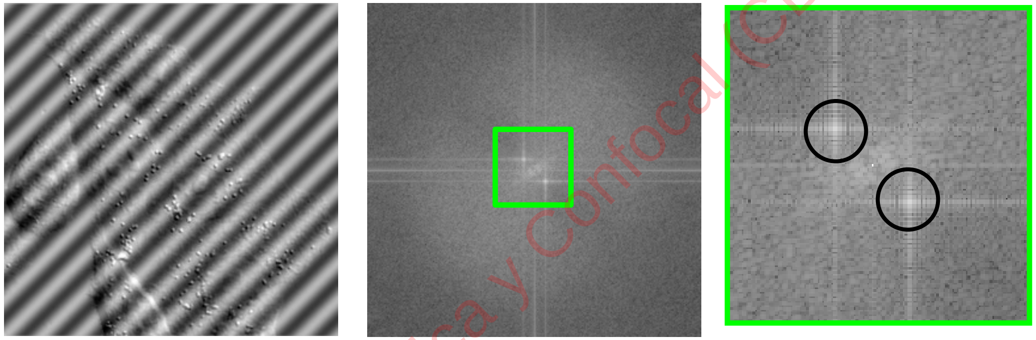
Shading Correction

FFT example

Correction of Periodic Artifacts

1. Open your file and change it to its Frequency Domain: **Process>FFT>FFT**

In the frequency domain image, periodic artifacts will show up as a pattern of a few bright star-like dots. When the artifacts are horizontally or vertically oriented, you will see horizontal and vertical bands along the center lines but not in the actual center of the frequency domain image.

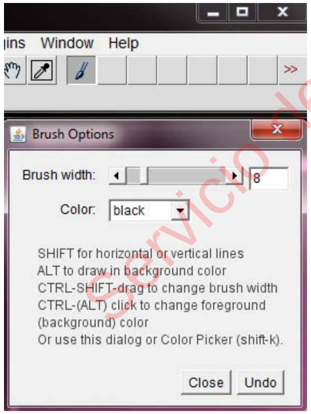


2. Double click on the **Paintbrush** tool: select Black Color and an appropriate brush width.

In the frequency domain image, paint black the star-like dots.

3. Return to the spatial domain: select your frequency domain image and go to **Process>FFT>Inverse FFT.**

Periodic artifacts should appear attenuated.



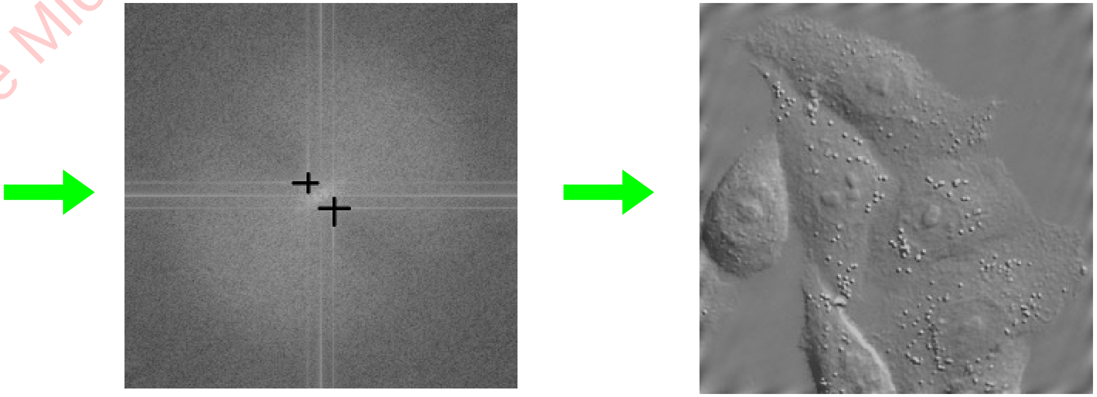
Brush Options

Brush width: 8

Color: black

SHIFT for horizontal or vertical lines
ALT to draw in background color
CTRL-SHIFT-drag to change brush width
CTRL-(ALT) click to change foreground (background) color
Or use this dialog or Color Picker (shift-k).

Close Undo



2. Segmentation

2. Segmentation

"Division of the image in different segments or groups of pixels that share certain features"

What for?

- Extract meaningful information (getting rid of unwanted image elements)
- Simplify image data for later analysis

Segmentation algorithms work considering 2 main Image Properties:

- **Discontinuity**: algorithm looks for abrupt intensity changes. Example: Edge-detection algorithms.
- **Similarity**: algorithm looks for image regions that have certain properties in common, such as intensity values. Example: thresholding and region growing.

Segmentation is a complex procedure!! Great variety of algoritms
It will affect quality of analysis: Bad Segmentation=Bad Results!!

2. Segmentation Methods…a simplyfied classification

Thresholding

Edge Detection

Trainable Segmentation

Region Based

Active Contours

Clustering

3D Segmentation

Color Segmentation

Watershed

2.1 Thresholding

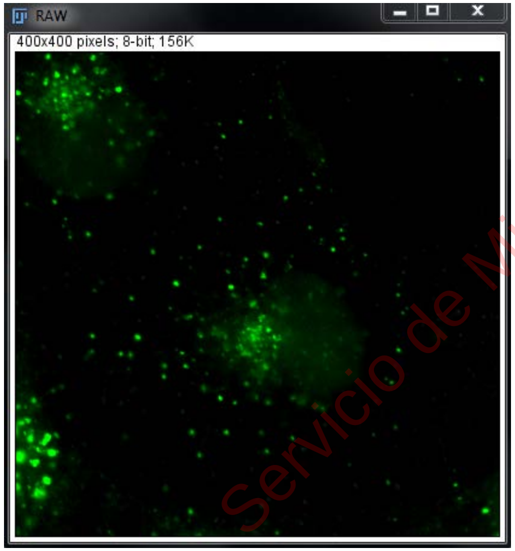
Simplest and most commonly used

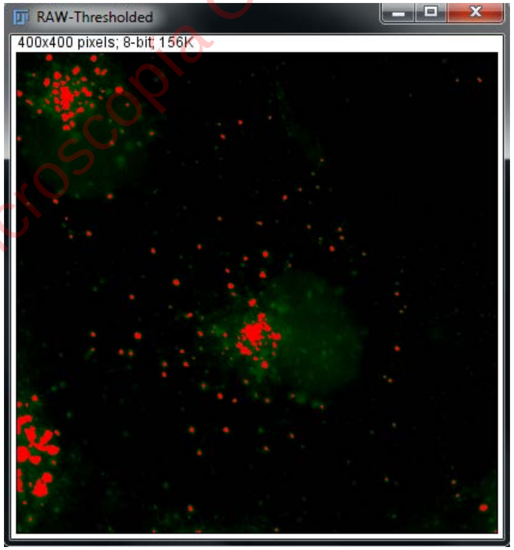
Pixels are grouped in different categories depending on its intensity value

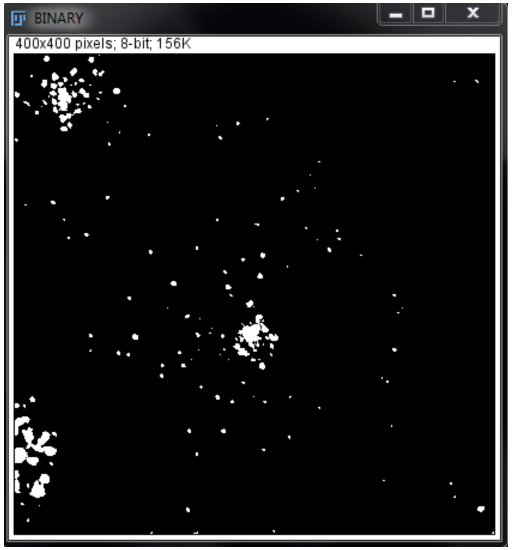
Does not take into account spatial characteristics of the image

Thresholding output = Binary Image

Binary image: a pixel can have only 2 values: Black (0) or White (255)







Raw Image

Thresholded

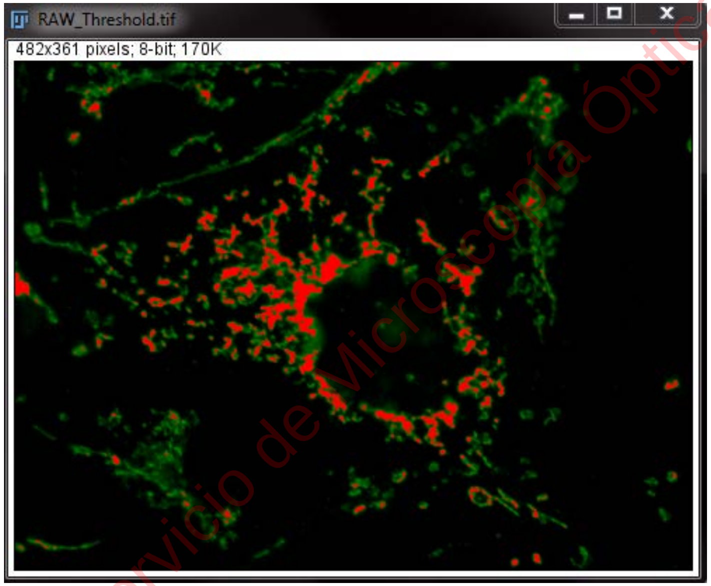
Binary Image

2.1 Thresholding…How to stablish the Threshold Value?

Important!! Threshold value has a strong influence in analysis

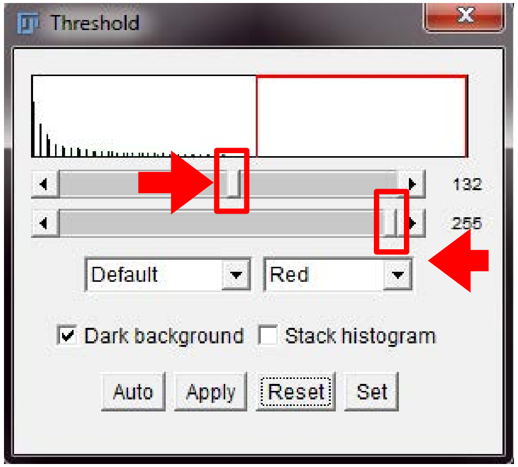
Interactively (Visual Inspection)

Image>Adjust/Threshold



RAW_Threshold.tif

482x361 pixels; 8-bit; 170K



Threshold

132

255

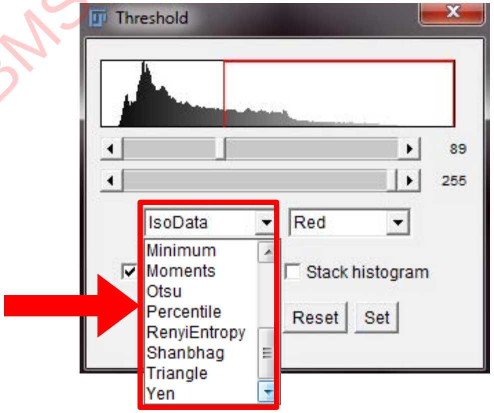
Default | Red

☑ Dark background □ Stack histogram

Caution!! Setting a threshold this way is subjective!

2.1 Thresholding…How to stablish the Threshold Value? II

Algorithms for automatic selection



Threshold

89

255

IsoData

Red

☑

□ Stack histogram

Global Threshold

Image>Adjust>Auto Threshold

Image>Adjust>Threshold

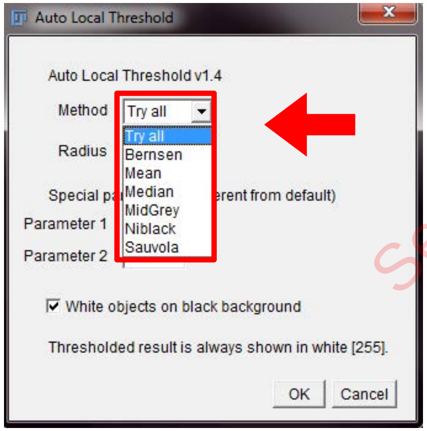
But…gives bad results when image background is not even

Local Threshold

Image>Adjust>Auto Local Threshold

Adapts the threshold value on each pixel to the local image characteristics

Try if background, illumination or intensity pattern are uneven.



Auto Local Threshold

Auto Local Threshold v1.4

Method Try all

Radius

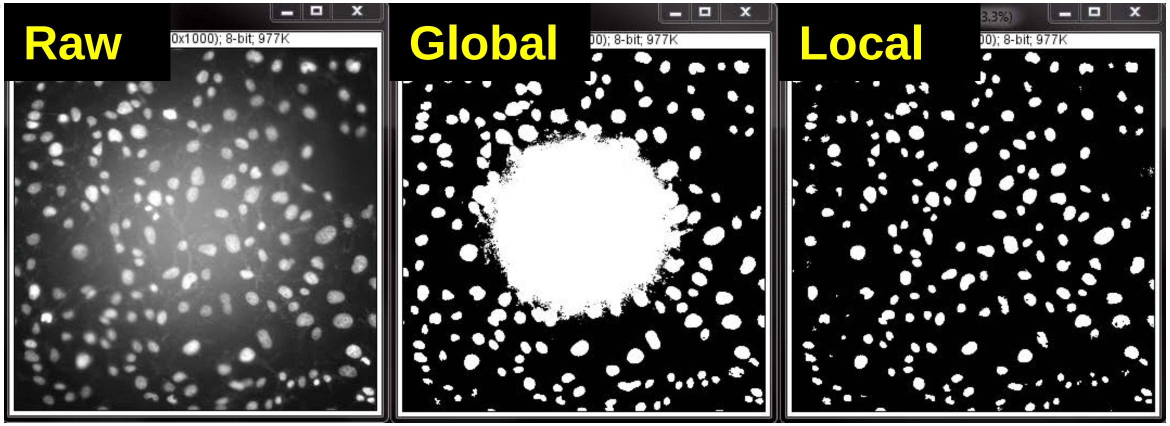
Special pa… …erent from default)

Parameter 1

Parameter 2

☑ White objects on black background

Thresholded result is always shown in white [255].



2.1.1 Morphological Operators on Binary Images I

These operators modify the morphology of an object in a binary image

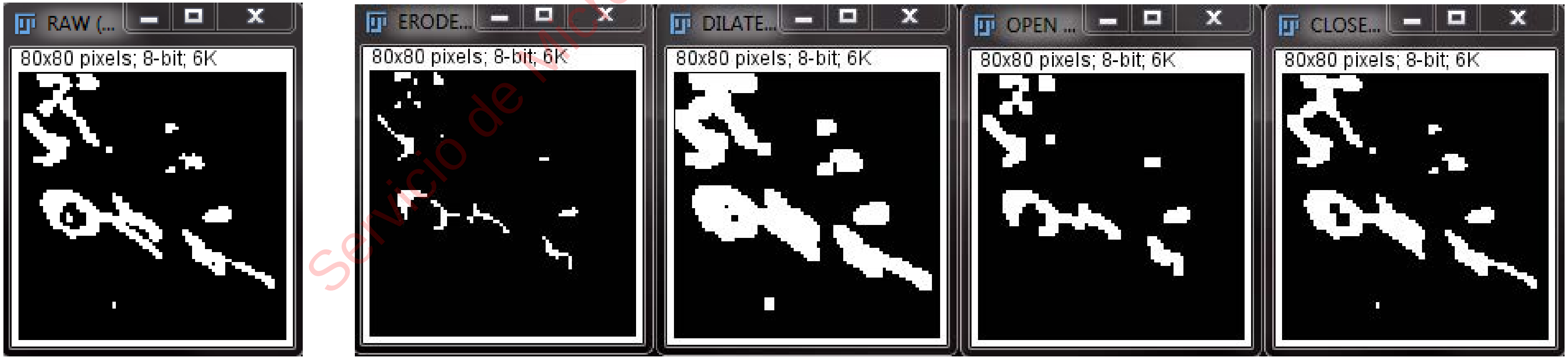
Process>Binary>

Erode: Shrinks the image. Holes became larger. Deletes small details

Dilate: Enlarges object borders. Holes became smaller

Open: Erode + Dilate. Smooths objects contourns, removes isolated elements, breaks thin connections

Close: Dilate + Erode. Smooths objects contourns, fill small holes, joins breaks



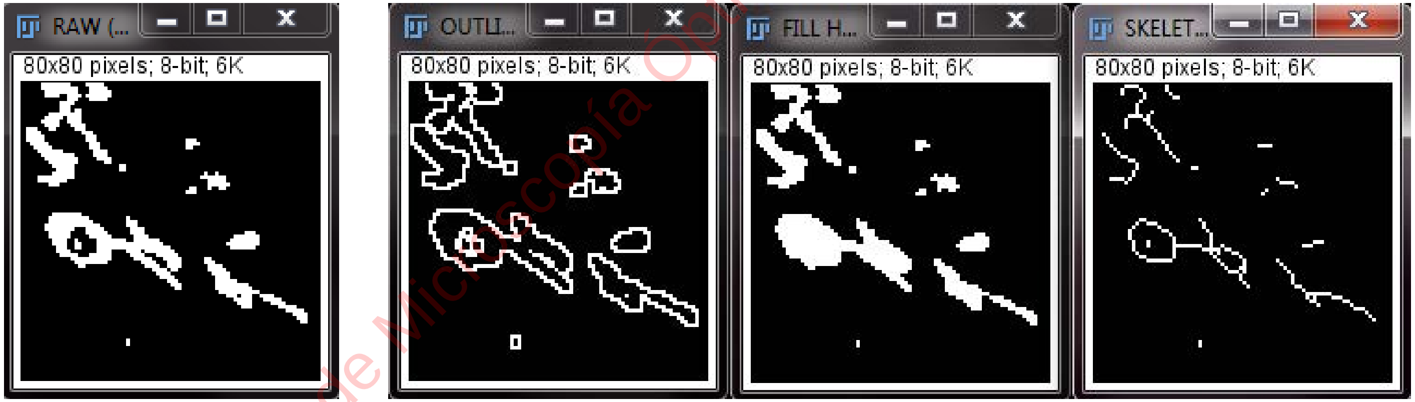
2.1.1 Morphological Operators on Binary Images II

Process>Binary>

Outline: Generates a one pixel wide outline of the objects

Fill Holes: Fills holes in objects

Skeletonize: Removes pixels from the edges of objects until they are 1pixel-wide

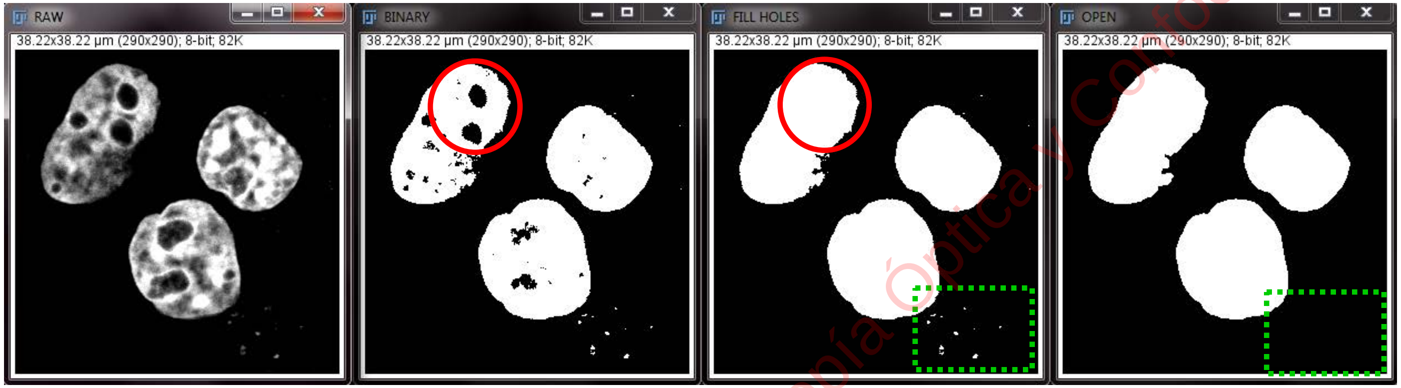


Process>Binary>Options Advance options to apply these morphological Operators

2.1.1 Morphological Operators on Binary Images. Applications?

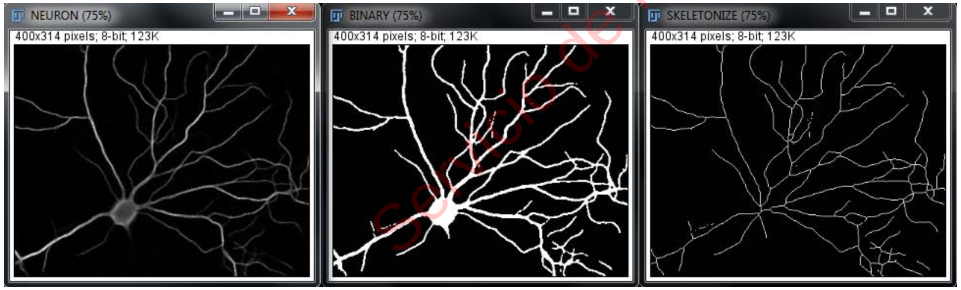
Can be used to improve segmentation after thresholding or edge detection

Example 1: Improve image before Nuclei Counting


RAW
38.22x38.22 µm (290x290); 8-bit; 82K
BINARY
38.22x38.22 µm (290x290); 8-bit; 82K
FILL HOLES
38.22x38.22 µm (290x290); 8-bit; 82K
OPEN
38.22x38.22 µm (290x290); 8-bit; 82K

1. Threshold→Binary
2. Fill Holes
3. Open

Example 2: Analyze Neuron Morphology


NEURON (75%)
400x314 pixels; 8-bit; 123K
BINARY (75%)
400x314 pixels; 8-bit; 123K
SKELETONIZE (75%)
400x314 pixels; 8-bit; 123K

Neuron image: http://labs.biology.ucsd.edu/halpain/images.html

Skeletonize is used in many Fiji/ImageJ neuron morphology/sholl analysis plugins

1. Threshold→Binary
2. Skeletonize

2.1.1 Morphological Operators on Grayscale images?

These operators modify the morphology of an object in a grayscale image

Process>Filters>Minimum... Does grayscale Erosion

Process>Filters>Maximum... Does grayscale Dilation

Also…

Grayscale Morphology plugin

Morphological Filters in Fast Morphology plugin

2.1.2 Logical/Boolean Operations Between two images

Process>Image Calculator

Plugin **Calculator Plus**

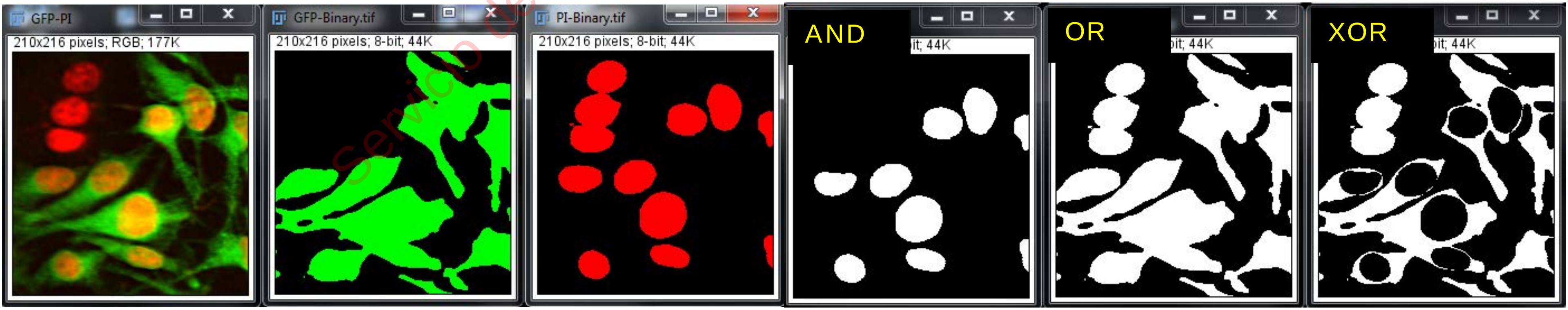
Work on a pixel by pixel manner

Usually done between 2 binary images or a binary and a grayscale image

Use to isolate specific structures of interest before image analysis

Result image is created based on these criteria:

AND: Pixels that are on in both images (common elements)
OR: Pixels that are in either image (all elements)
XOR: Pixels that are in one or the other image but not in both (exclusive elements)


GFP-PI
210x216 pixels; RGB; 177K
GFP-Binary.tif
210x216 pixels; 8-bit; 44K
PI-Binary.tif
210x216 pixels; 8-bit; 44K
AND
it; 44K
OR
bit; 44K
XOR
bit; 44K

2.2 Edge Detection

Works finding the edges or boundaries of objects in the image

Edges are image areas where a sharp change in intensity can be found

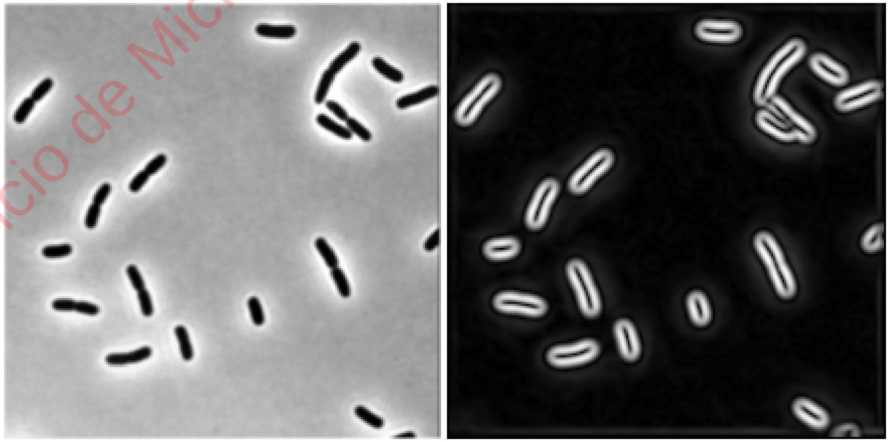
There are many filters and algorithms for edge detection:

Process>Find Edges

Plugin **Canny Edge Detector**

Plugin **Canny-Deriche filtering**

Plugin **FeatureJ>FeatureJ Edges**

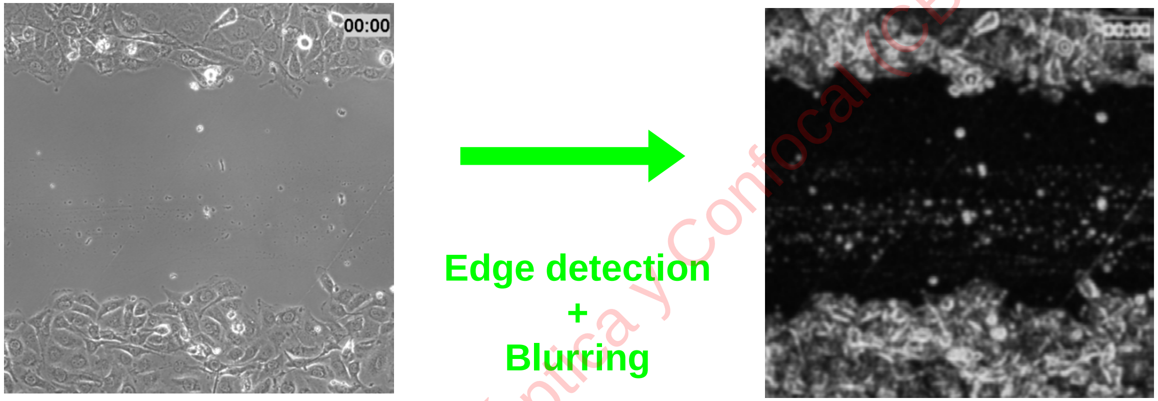


Wound Healing segmentation example I

1. Open and duplicate your original image: **Image>Duplicate…**

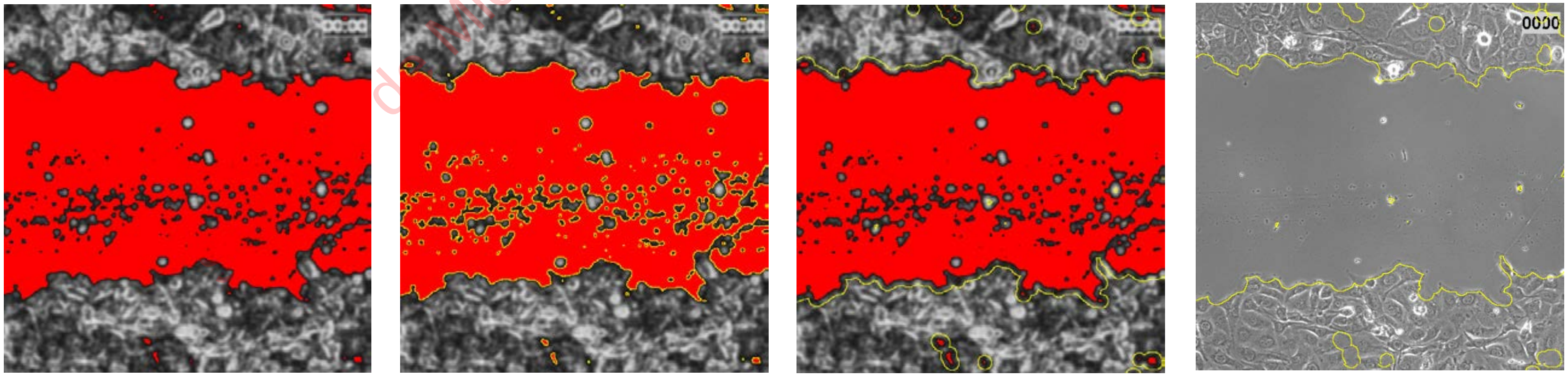
2. On your replica, apply any edge detection tool available. For this example **Find Edges** tool was used : **Process>Find Edges**

3. Apply a gaussian blur filter to obtain a blurred-edge image. For this example a **Radius** of 2 was used. **Process>Filters>Gaussian Blur…**



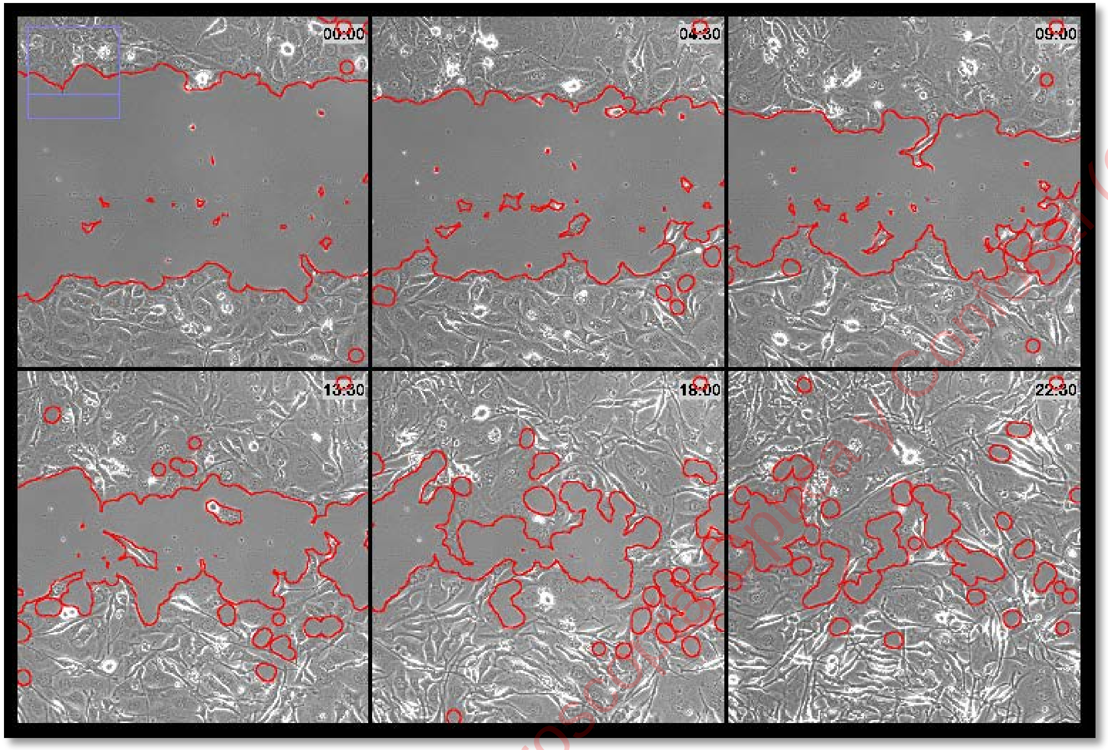
4. Apply an intensity threshold (**Image>Adjust>Threshold**) to detect the wound area and create a selection around the thresholded surface: **Edit>Selection>Create Selection**

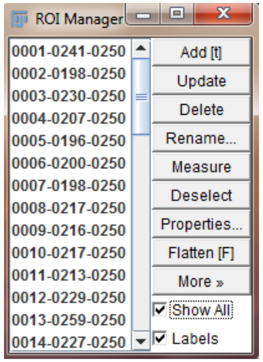
5. The selection was enlarged 8 pixels to fit selection borders to wound edge contours: **Edit>Selection>Enlarge**



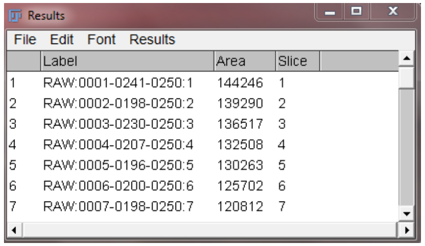
Wound Healing segmentation example II

6. Add selection to ROI Manager for area measurement: **Analyze>Tools>ROI Manager**.



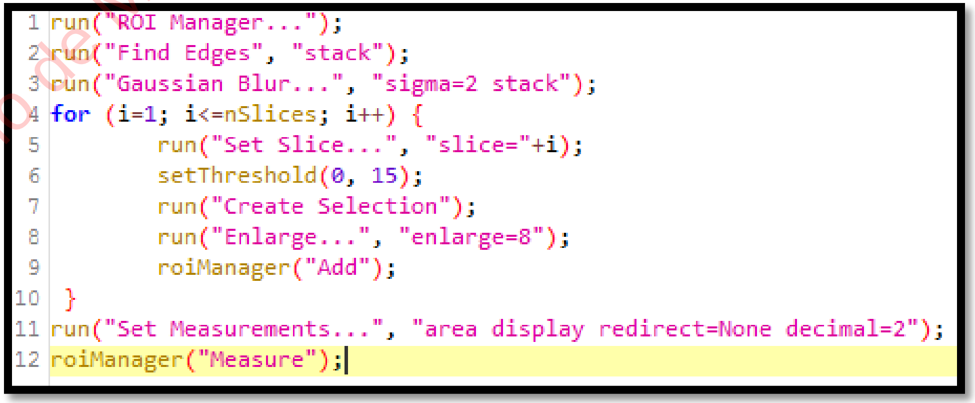


0001-0241-0250
0002-0198-0250
0003-0230-0250
0004-0207-0250
0005-0196-0250
0006-0200-0250
0007-0198-0250
0008-0217-0250
0009-0216-0250
0010-0217-0250
0011-0213-0250
0012-0229-0250
0013-0259-0250
0014-0227-0250

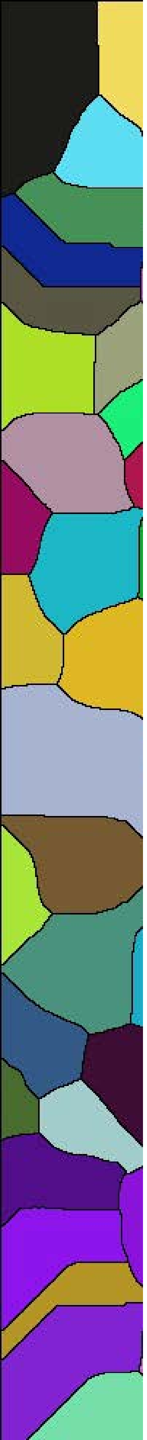


	Label	Area	Slice
1	RAW:0001-0241-0250:1	144246	1
2	RAW:0002-0198-0250:2	139290	2
3	RAW:0003-0230-0250:3	136517	3
4	RAW:0004-0207-0250:4	132508	4
5	RAW:0005-0196-0250:5	130263	5
6	RAW:0006-0200-0250:6	125702	6
7	RAW:0007-0198-0250:7	120812	7

For this example, a simple macro was created to analyze each timepoint automatically:



```
run("ROI Manager...");
run("Find Edges", "stack");
run("Gaussian Blur...", "sigma=2 stack");
for (i=1; i<=nSlices; i++) {
        run("Set Slice...", "slice="+i);
        setThreshold(0, 15);
        run("Create Selection");
        run("Enlarge...", "enlarge=8");
        roiManager("Add");
 }
run("Set Measurements...", "area display redirect=None decimal=2");
roiManager("Measure");
```



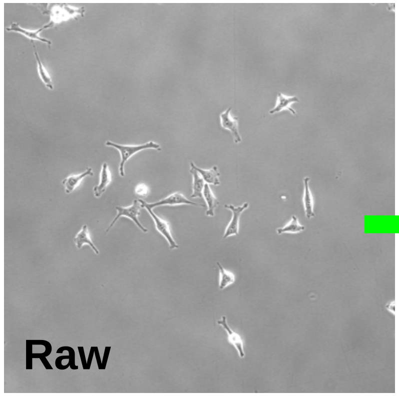
Phase Contrast Cells segmentation example I

PHANTAST: a plugin for automatic segmentation of phase contrast images

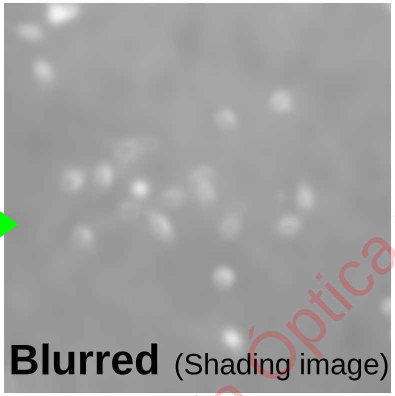
1. Remember to work always with a copy of your original images.

2. For this example, a shading correction was applied:

2.1 Open, duplicate your image and rename it as Shading: **Image>Duplicate…**
2.2 Apply a gaussian blur filter to blur out cells (Radius 5):
2.3 Use Image Calculator to apply the shading correction: Divide original image by shading image (new,32bits)
2.4 Scale result image from 32 Bit to 8 Bit: **Image>Type>8-bit**



Raw



Blurred (Shading image)

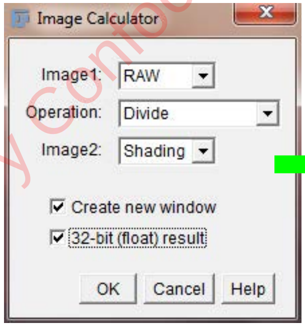
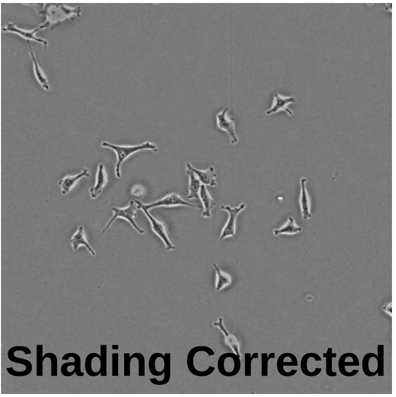


Image Calculator

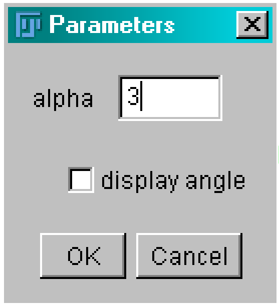
Image1: RAW
Operation: Divide
Image2: Shading
☑ Create new window
☑ 32-bit (float) result



Shading Corrected

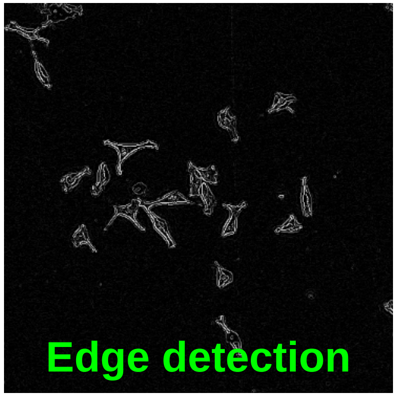
3. Because thresholding segmentation is difficult for phase contrast images, an edge detection segmentation plugin was used: **Image Edge plugin**:

3.1 Download **Image Edge** plugin. Information on this tool can be found here:
http://imagejdocu.tudor.lu/doku.php?id=plugin:filter:edge_detection:start
3.2 Select your shading corrected image and go to **Plugins>Image Edge>Deriche** .Use an **alpha** value of 3
3.3 Keep the result image with sharper edges: Canny-Deriche suppr 3.0

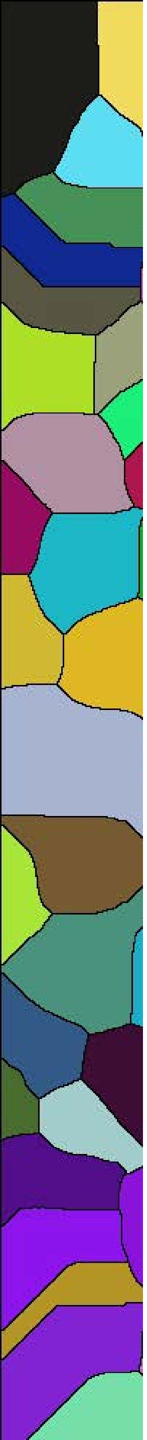


Parameters

alpha 3
□ display angle



Edge detection



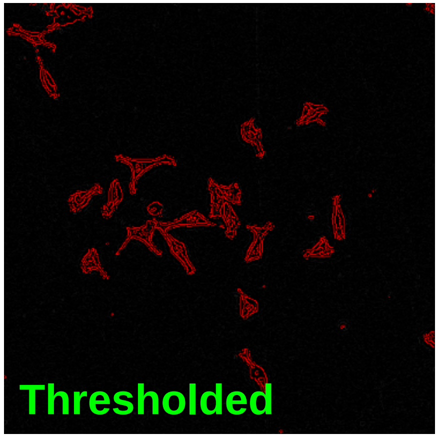
Phase Contrast Cells segmentation example II

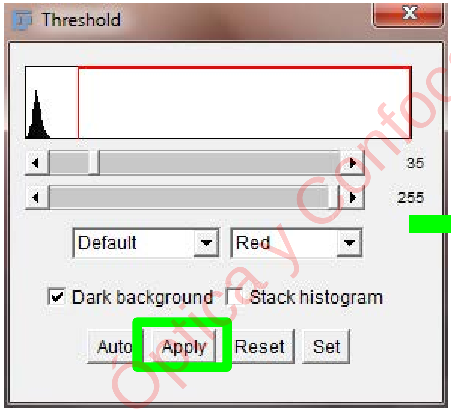
4. Scale result image from 32 Bit to 8 Bit: **Image>Type>8-bit**

5. Now that the image is simplified, apply an intensity threshold to obtain a binary image (white edges and black background):

5.1 **Image>Adjust>Threshold**

5.2 Once the threshold range has been selected, click on **Apply**, to obtain a binary image:




Threshold

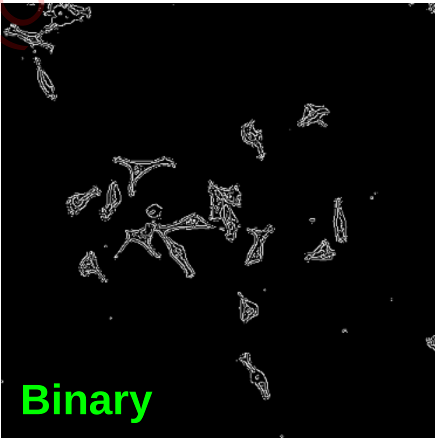
35

255

Default ▾ Red ▾

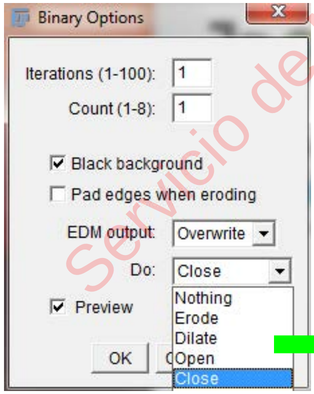
☑ Dark background □ Stack histogram

Auto Apply Reset Set



6. Use morphological operations to improve cell shape: **Process>Binary>Options**

Use **Close** and **Fill Holes** operations to join cells borders and fill holes:


Binary Options

Iterations (1-100): 1

Count (1-8): 1

☑ Black background

□ Pad edges when eroding

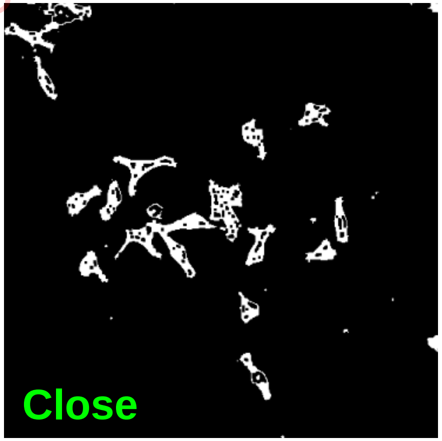
EDM output: Overwrite ▾

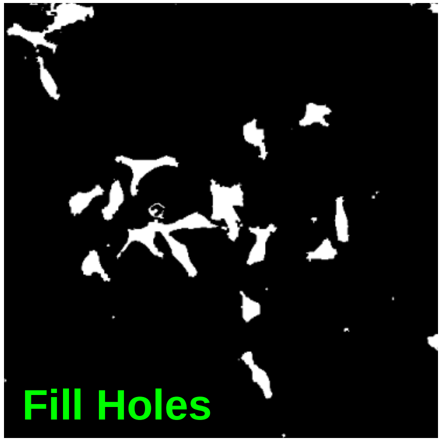
Do: Close ▾

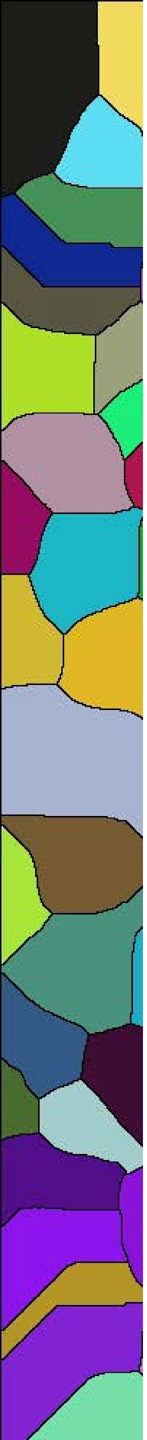
Nothing
Erode
Dilate
Open
Close

☑ Preview

OK

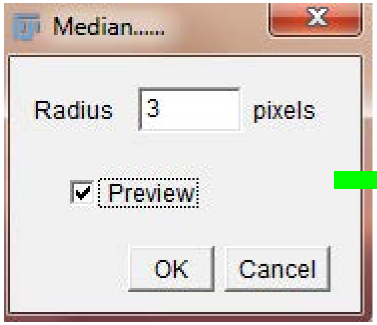






Phase Contrast Cells segmentation example III

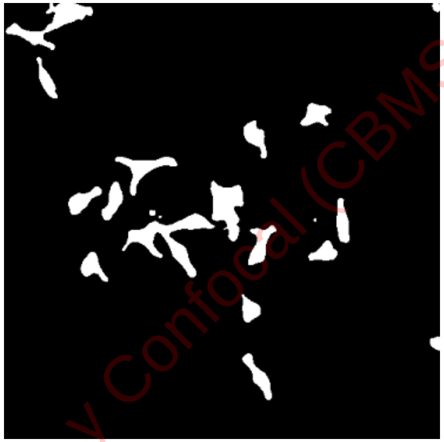
7. Apply a Median filter (Radius=3) to get rid of small pixels and to smooth contours: **Process>Filters>Median**…



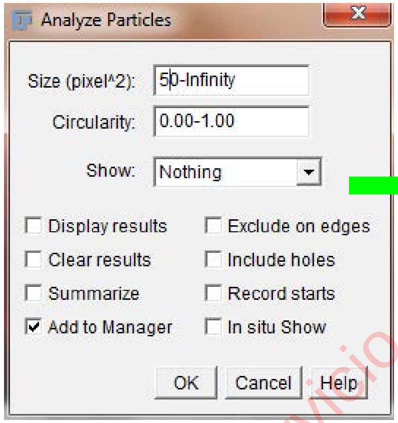
Median......

Radius 3 pixels

☑ Preview



8. Use **Analyze Particles** tool to detect individual cells regions: **Analyze> Analyze Particles**



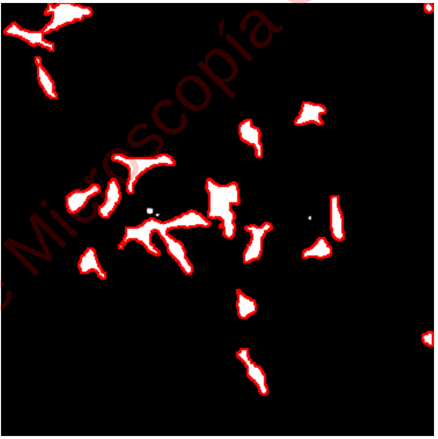
Analyze Particles

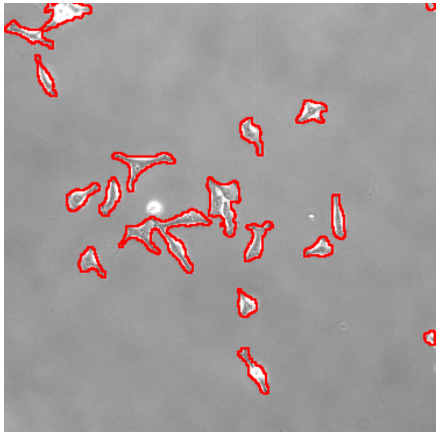
Size (pixel^2): 50-Infinity

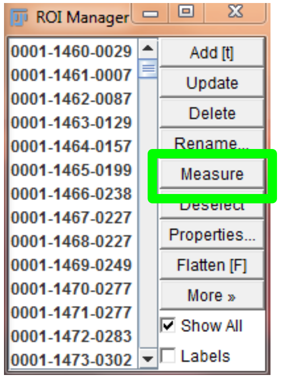
Circularity: 0.00-1.00

Show: Nothing

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







ROI Manager

0001-1460-0029
0001-1461-0007
0001-1462-0087
0001-1463-0129
0001-1464-0157
0001-1465-0199
0001-1466-0238
0001-1467-0227
0001-1468-0227
0001-1469-0249
0001-1470-0277
0001-1471-0277
0001-1472-0283
0001-1473-0302

☑ Show All
□ Labels

9. You can measure different parameters from cells regions using the **Measure** button:

Use **Analyze>Set Measurements** to define the parameters to measure

Take into account that some of the cells might appear fused

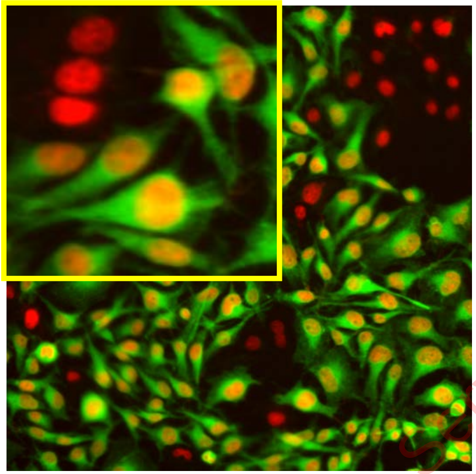
2.3 K-clustering

Segments the image in ‘K’ number of clusters or regions with similar intensity

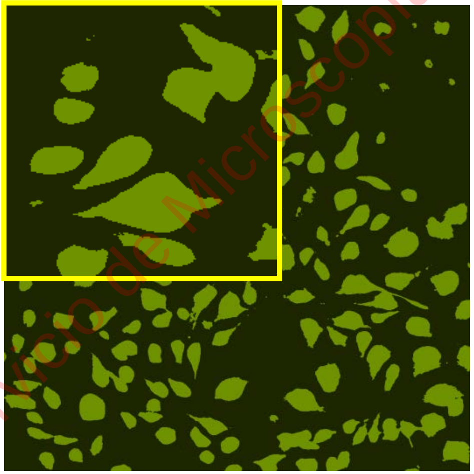
Number of clusters (K) must be defined by user

Can be used for colour images

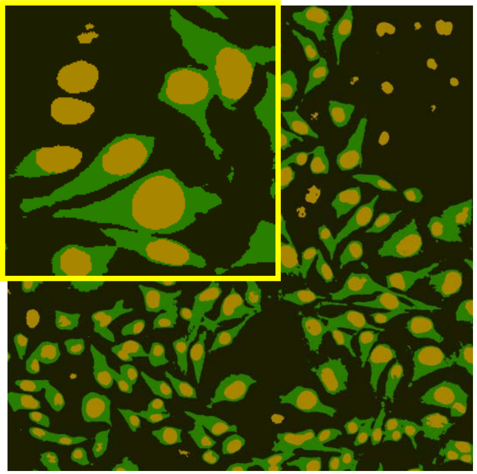
Plugin **k-means Clustering**



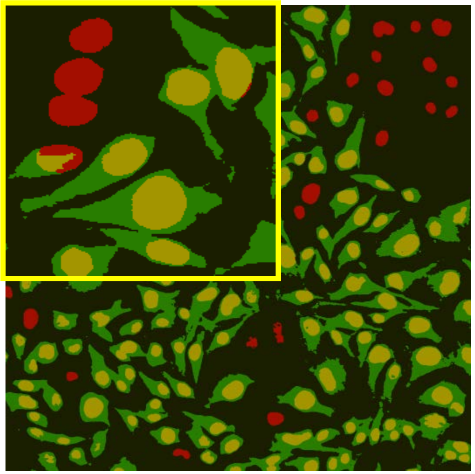
Input Image



K=2



K=3



K=4

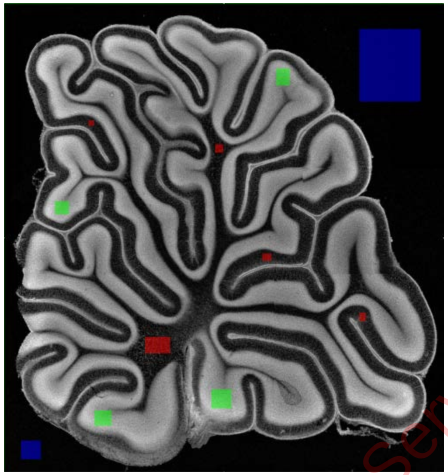
2.4 Region Based

Classifies pixels based on its similarity

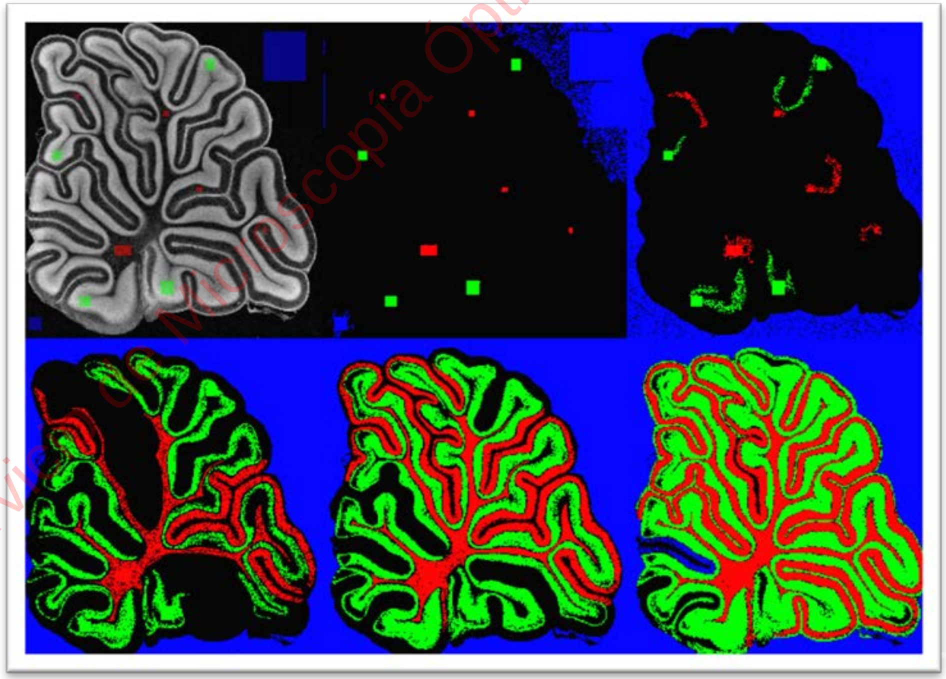
A certain number of Seeds or reference pixels must be defined by user

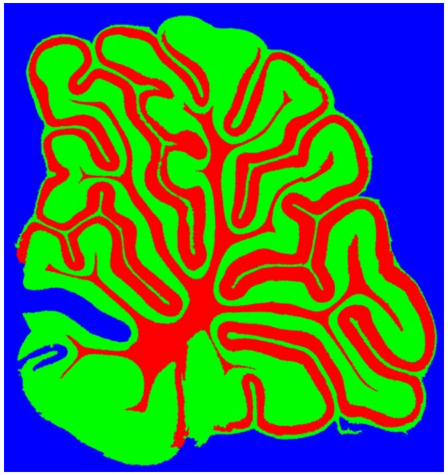
Algorithm increases iteratively seed area

Plugin **Seeded Region Growing Tool**



Raw Image + Seeds





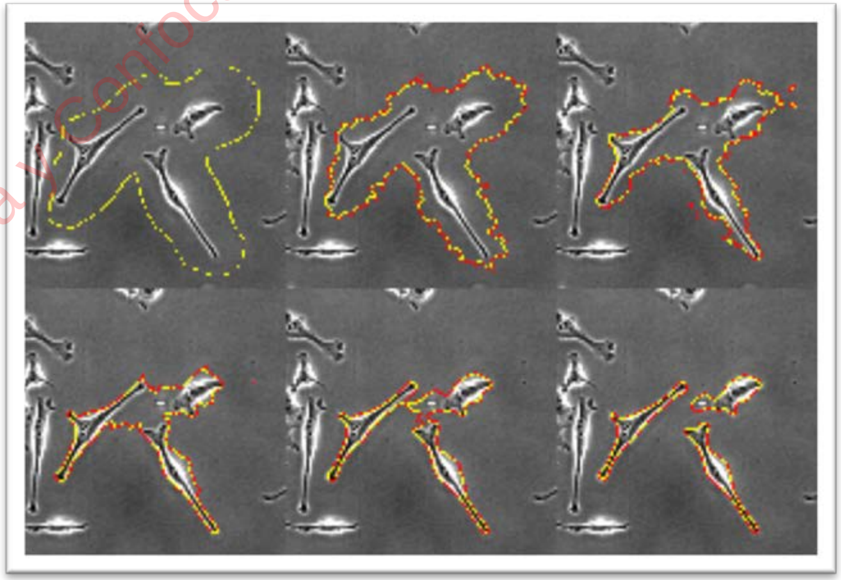
Result

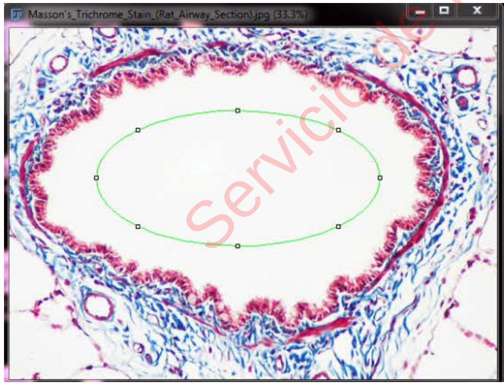
2.5 Active Contours

User specifies an initial contour around the object

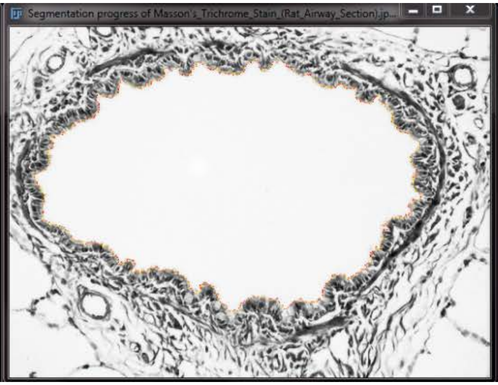
The curve is modified (it grows inwards or outwards) until it reaches object boundaries

Plugin **Level Sets**
Plugin **ABSnake**
Plugin **E-Snake**

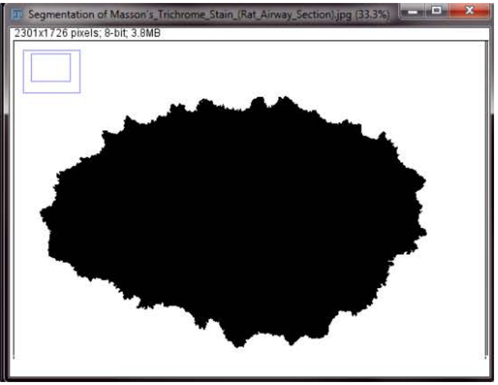




Masson's_Trichrome_Stain_(Rat_Airway_Section).jpg (33.3%)



Segmentation progress of Masson's_Trichrome_Stain_(Rat_Airway_Section).jp...



Segmentation of Masson's_Trichrome_Stain_(Rat_Airway_Section).jpg (33.3%)

2301x1726 pixels; 8-bit; 3.8MB

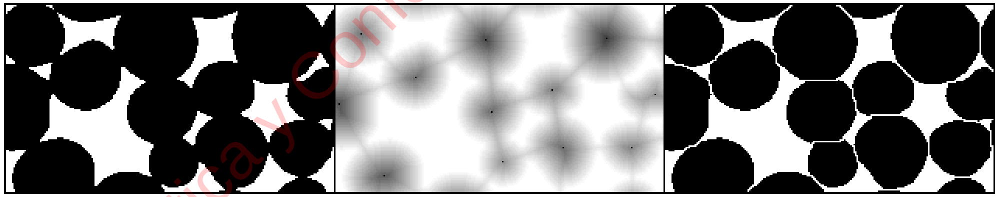
2.6 Watershed

Use to separate or cut apart particles that touch

Binary Watershed

Dilates as far as possible the point in the object located furthest from background

Process>Binary>Watershed



Thresholded Cells

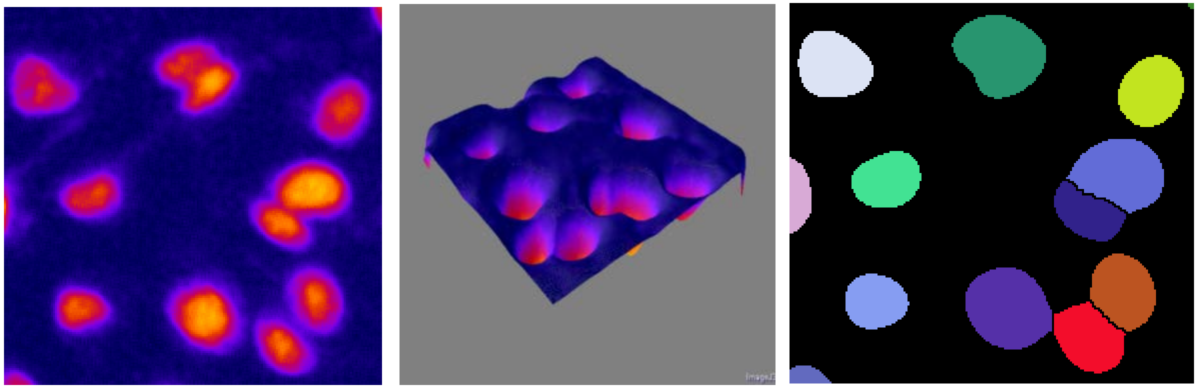
EDM and UEPs

After Watershed Segmentation

Grayscale Watershed

Image seen as a topographic representation. ‘Flooded’ from below to above. When ‘rising water’ in adjacent basins is about to merge, a dam (Watershed line) is built to prevent merging.

Plugin [Graylevel Watershed]

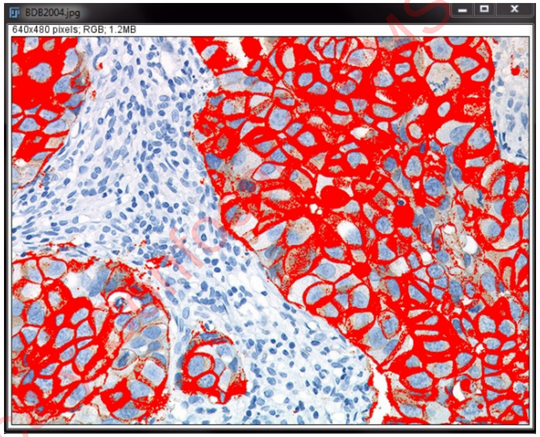


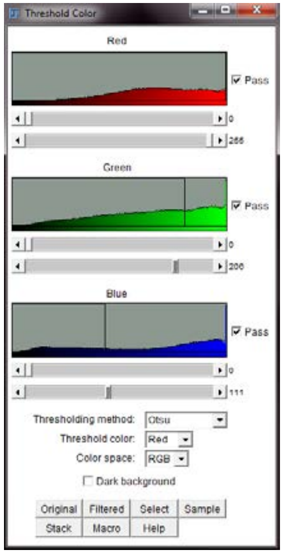
2.7 Colour Image Segmentation

2.7.1 Color Threshold

Color based threshold segmentation

Image>Adjust>Color Threshold


BDB2004.jpg
640x480 pixels; RGB; 1.2MB


Threshold Color

Red ☑ Pass 0 255
Green ☑ Pass 0 200
Blue ☑ Pass 0 111

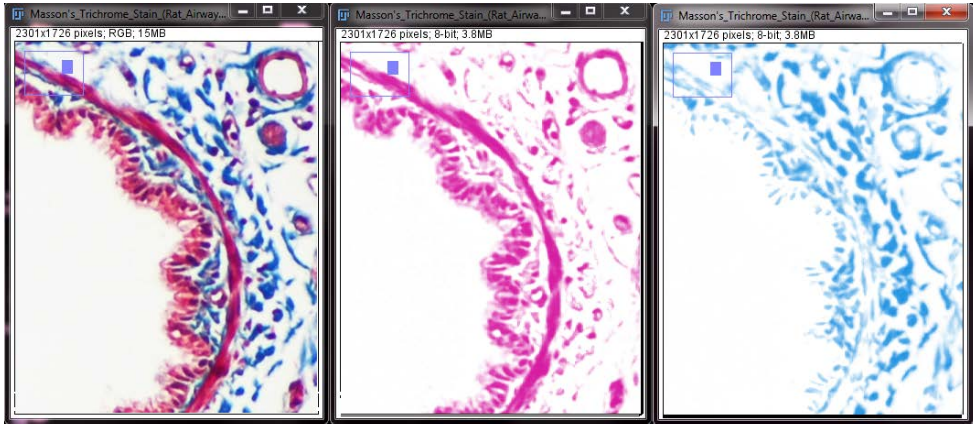
Thresholding method: Otsu
Threshold color: Red
Color space: RGB
☐ Dark background

2.7.2 Colour Deconvolution

Splits color image into separate channels based on 3 determined colors

Some histological stains already defined (e.g., DAB, H&E, Masson trichrome)

Plugin Colour Deconvolution


Masson's_Trichrome_Stain_(Rat_Airway...
2301x1726 pixels; RGB; 15MB
Masson's_Trichrome_Stain_(Rat_Airwa...
2301x1726 pixels; 8-bit; 3.8MB
Masson's_Trichrome_Stain_(Rat_Airwa...
2301x1726 pixels; 8-bit; 3.8MB

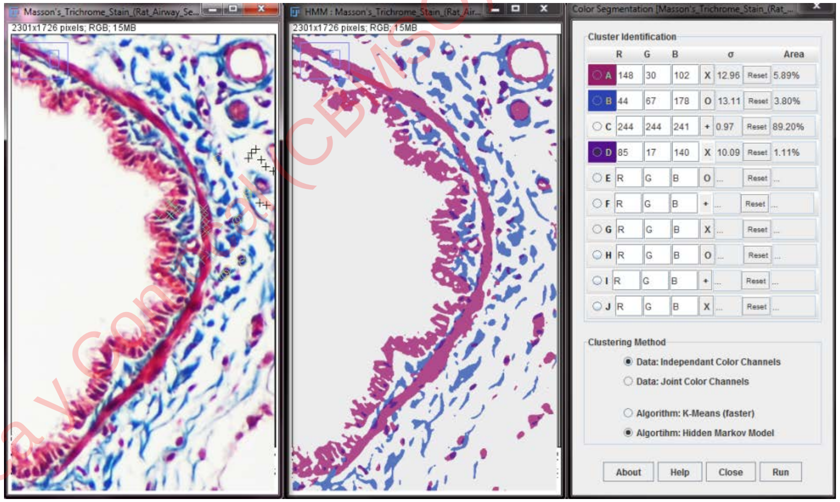
2.7 Colour Image Segmentation II

2.7.3 Color Segmentation

Groups pixels with similar color properties

Clusters are defined by user

Plugin Color Segmentation


Masson's_Trichrome_Stain_(Rat_Airway_Se...
2301x1726 pixels; RGB; 15MB
HMM : Masson's_Trichrome_Stain_(Rat_Air...
2301x1726 pixels; RGB; 15MB
Color Segmentation [Masson's_Trichrome_Stain_(Rat_...

Cluster Identification

	R	G	B		σ		Area
○ A	148	30	102	X	12.96	Reset	5.89%
○ B	44	67	178	O	13.11	Reset	3.80%
○ C	244	244	241	+	0.97	Reset	89.20%
● D	85	17	140	X	10.09	Reset	1.11%
○ E	R	G	B	O	...	Reset	...
○ F	R	G	B	+	...	Reset	...
○ G	R	G	B	X	...	Reset	...
○ H	R	G	B	O	...	Reset	...
○ I	R	G	B	+	...	Reset	...
○ J	R	G	B	X	...	Reset	...

Clustering Method

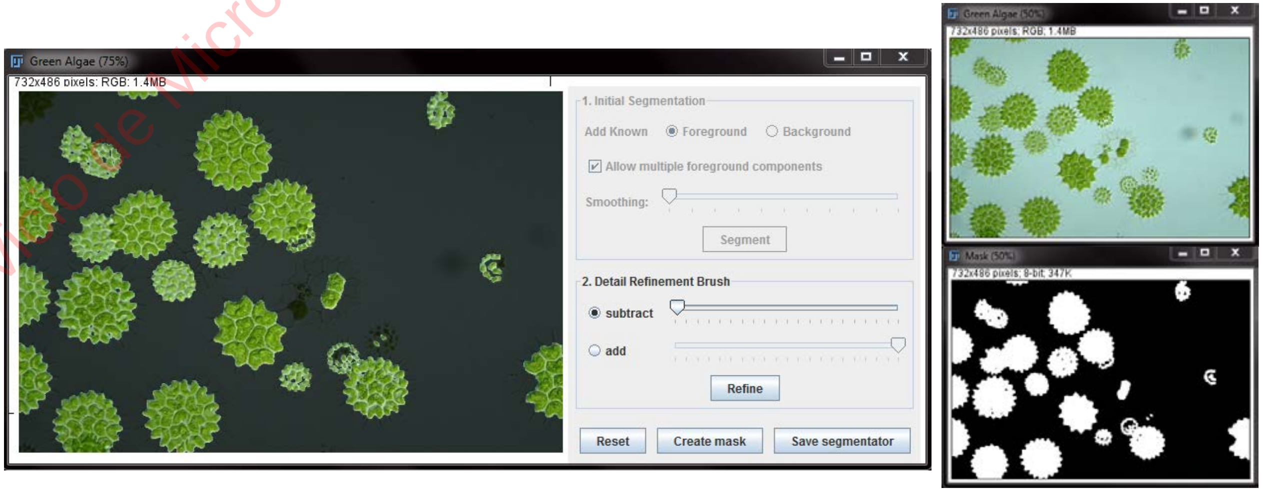
● Data: Independant Color Channels
○ Data: Joint Color Channels

○ Algorithm: K-Means (faster)
● Algortihm: Hidden Markov Model

2.7.4 SIOX (Simple Interactive Object Extraction)

Extracts foreground pixels (objects) from background pixels based on used defined regions

Plugin SIOX


Green Algae (75%)
732x486 pixels; RGB; 1.4MB
Green Algae (50%)
732x486 pixels; RGB; 1.4MB
Mask (50%)
732x486 pixels; 8-bit; 347K

1. Initial Segmentation

Add Known ● Foreground ○ Background

☑ Allow multiple foreground components

Smoothing:

2. Detail Refinement Brush

● subtract

○ add

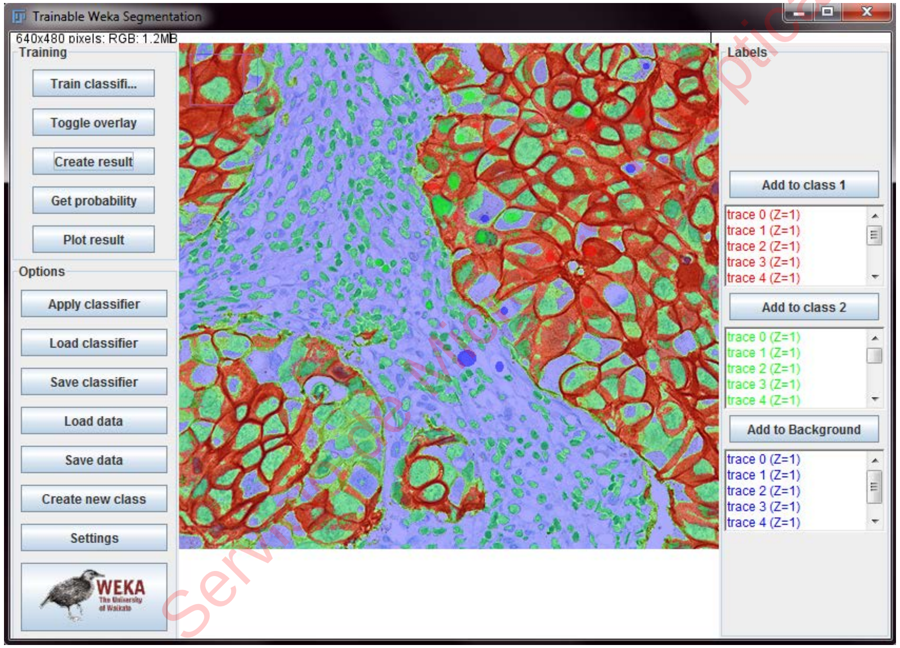
2.8 Trainable Segmentation

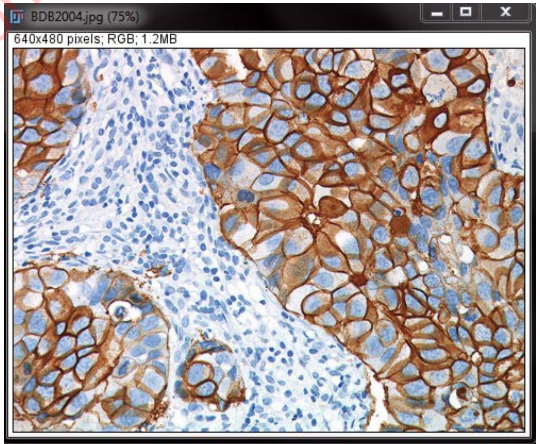
2.8.1 WEKA

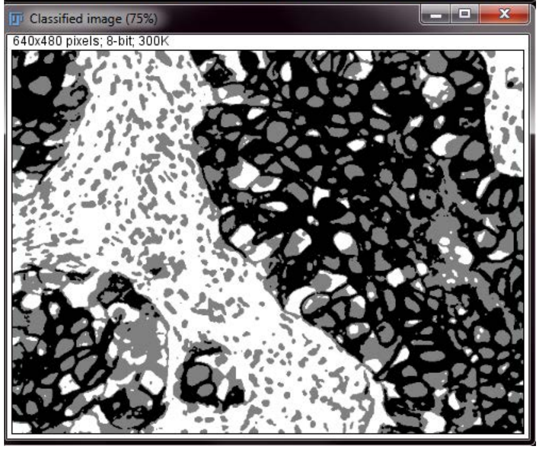
It can be trained from user defined regions

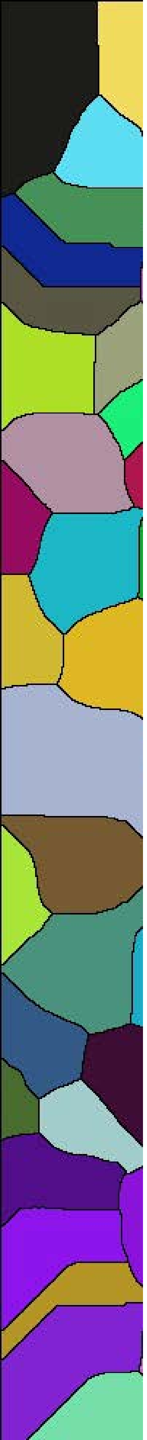
Creates a classifier to perform the same segmentation process in different images

Plugin **Trainable WEKA Segmentation**









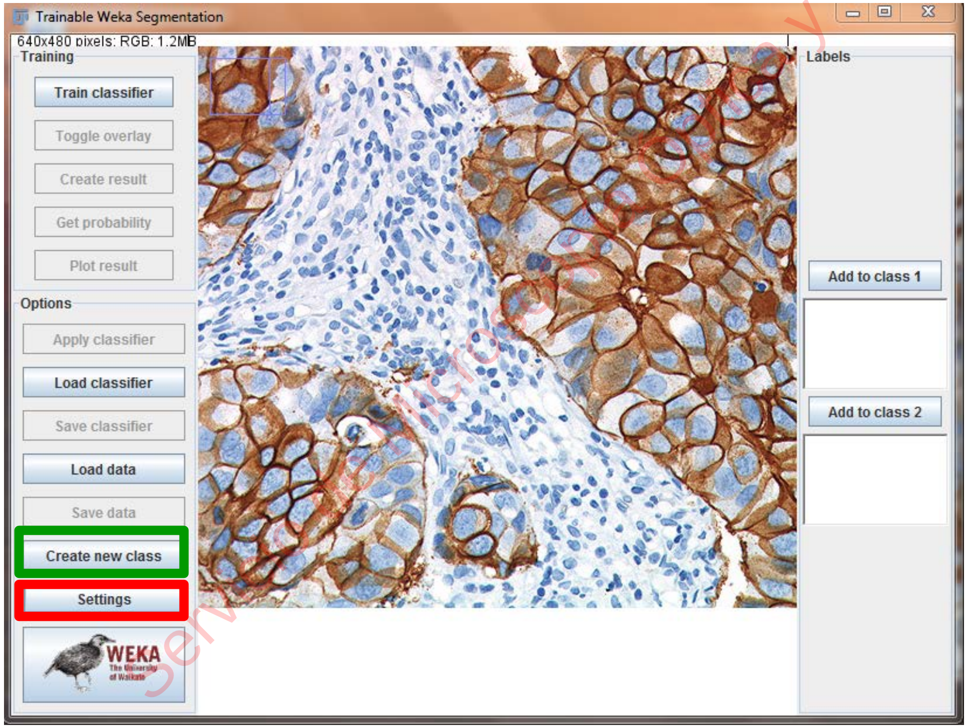
Trainable WEKA segmentation example I

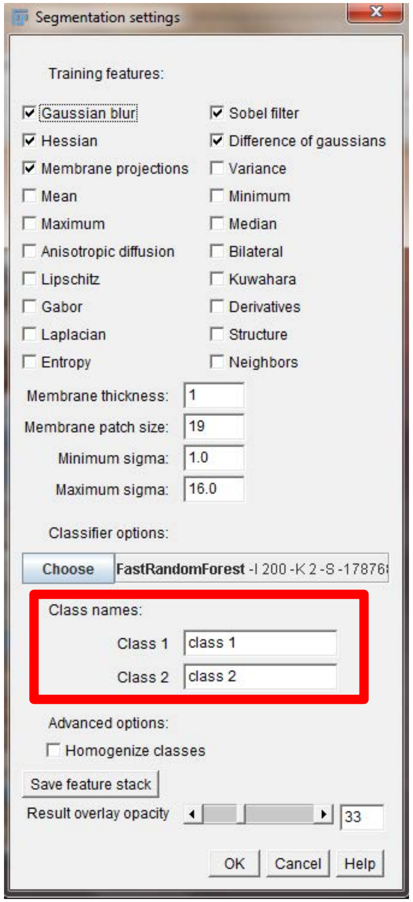
1. If necessary, download and install **Trainable WEKA Segmentation** plugin.

http://fiji.sc/Trainable_Weka_Segmentation

2. Open the image or stack of images and WEKA plugin

3. By default, classification is based only on 2 classes, but if needed more classification classes can be created clicking on the **Create new class** button. They can also be renamed going to **Settings**.





Training features:

☑ Gaussian blur	☑ Sobel filter
☑ Hessian	☑ Difference of gaussians
☑ Membrane projections	□ Variance
□ Mean	□ Minimum
□ Maximum	□ Median
□ Anisotropic diffusion	□ Bilateral
□ Lipschitz	□ Kuwahara
□ Gabor	□ Derivatives
□ Laplacian	□ Structure
□ Entropy	□ Neighbors

Membrane thickness: 1

Membrane patch size: 19

Minimum sigma: 1.0

Maximum sigma: 16.0

Classifier options:

Choose FastRandomForest -I 200 -K 2 -S -17876

Class names:

Class 1 class 1

Class 2 class 2

Advanced options:

□ Homogenize classes

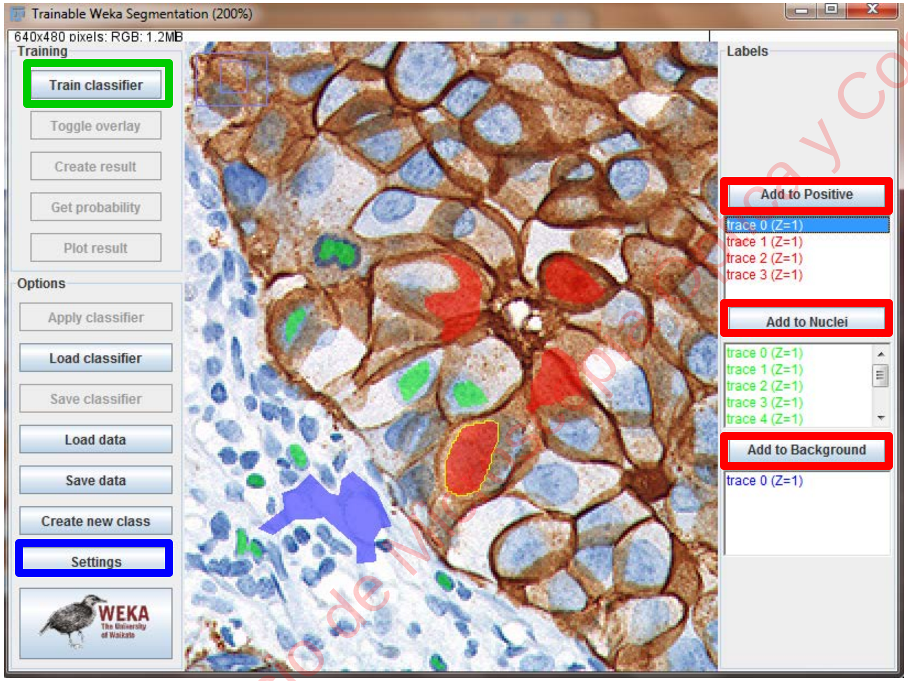
Save feature stack

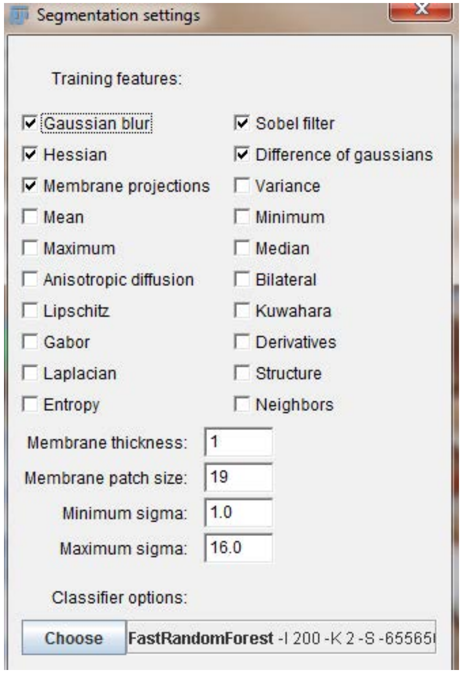
Result overlay opacity 33

Trainable WEKA segmentation example II

4. Using any of the drawing selection tools available, select image areas of the desired colour and assign them to a certain class (**Add to Class**)

5. There are different training features in **Settings** that will be taken into account for pixel classification. Increasing its number will increase the time needed for processing.




Segmentation settings

Training features:

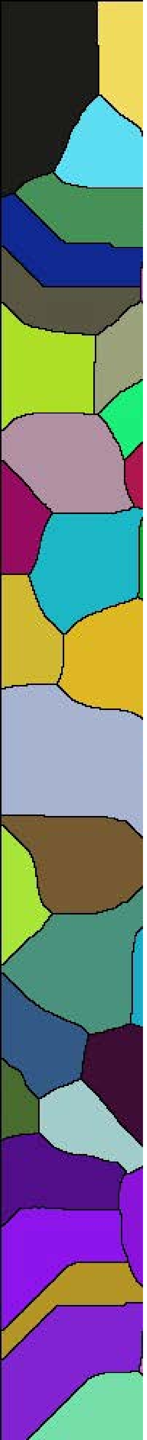
☑ Gaussian blur ☑ Sobel filter
☑ Hessian ☑ Difference of gaussians
☑ Membrane projections □ Variance
□ Mean □ Minimum
□ Maximum □ Median
□ Anisotropic diffusion □ Bilateral
□ Lipschitz □ Kuwahara
□ Gabor □ Derivatives
□ Laplacian □ Structure
□ Entropy □ Neighbors

Membrane thickness: 1
Membrane patch size: 19
Minimum sigma: 1.0
Maximum sigma: 16.0

Classifier options:

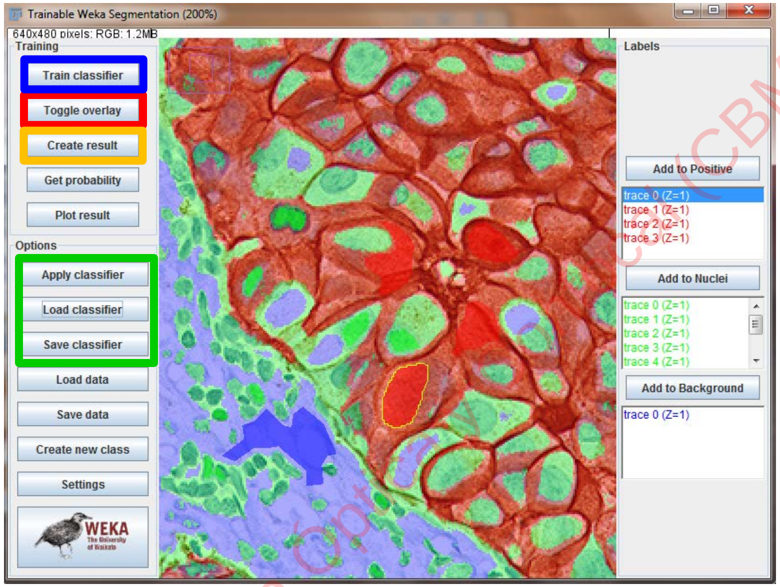
Choose FastRandomForest -I 200 -K 2 -S -65565

6. Click **Train classifier** to start pixel classification.



Trainable WEKA segmentation example III

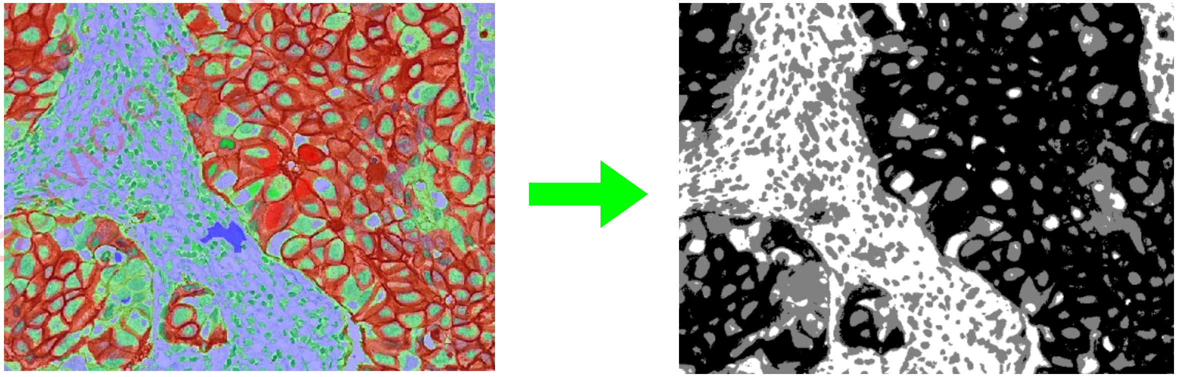
7. Use the **Toogle overlay** button to check segmentation accuracy



8. If necessary, assign new image areas to the classes and repeat **Train classifier** for colour detection improvement

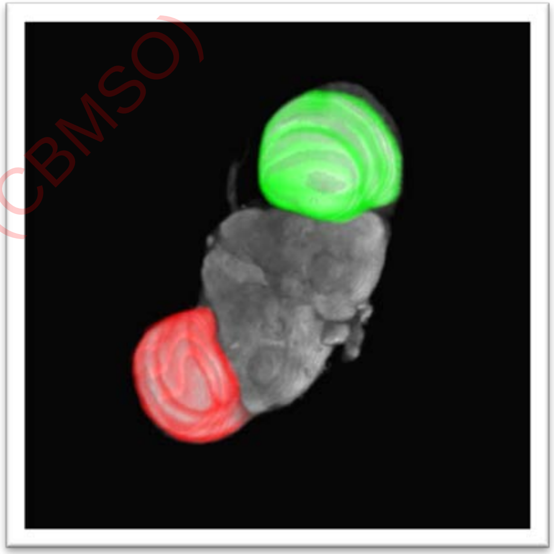
9. Once an acceptable result is obtained, save your classifier (**Save classifier**) or apply it to other images (**Apply classifier**)

10. Use **Create result** to produce the segmented image



2.9 3D Segmentation

2.9.1 Segmentation Editor



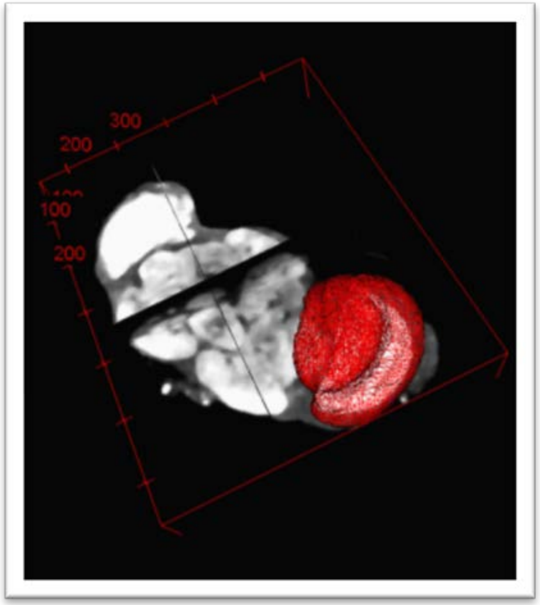
3D manual segmentation

Interpolation between ROIs

Labels can be combined with original image for 3D display

Plugin **Segmentation Editor**

2.9.2 Interactive 3D Segmentation



Semi-automated 3D object surface segmentation

Surface generation process from user defined seeds

3D viewer plugin needed

Plugin **Interactive 3D Segmentation**

2.9 3D Segmentation II

2.9.3 3D ROI Manager

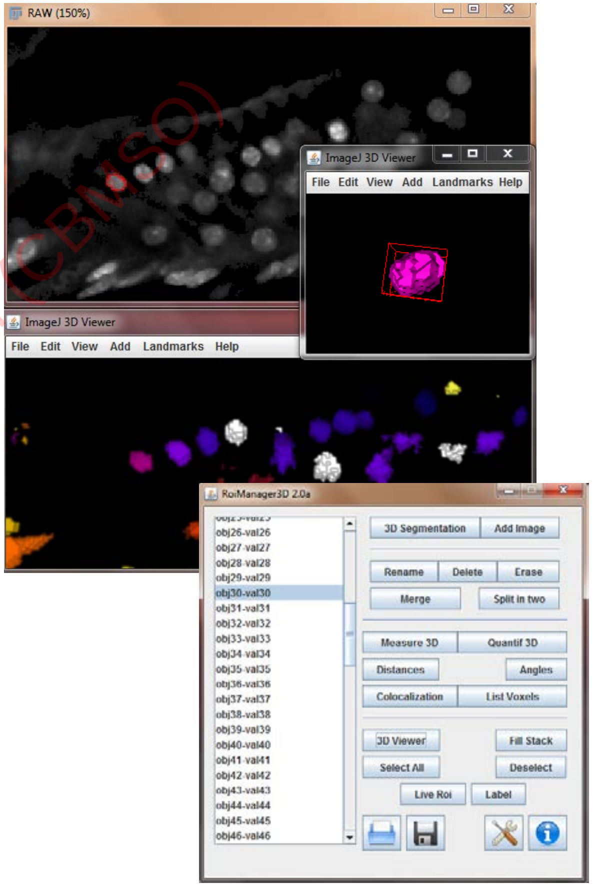
3D threshold segmentation

Detects 1 object even if it appears in different 2D slices

3D measurements (intensity, volume, etc)

3D Viewer plugin needed

Plugin **3D ROI Manager**



2.9.4 Simple Neurite Tracer

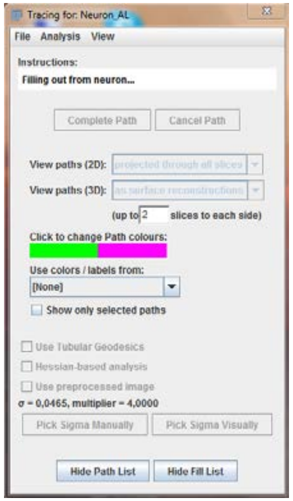
Semi-automatic 3D tracing and segmentation

Optimal for neurons/tubular structures

3D measurements (length, volume)

3D Viewer plugin needed

Plugin **Simple Neurite Tracer**



Instructions:
Filling out from neuron...

View paths (2D): projected through all slices
View paths (3D): as surface reconstructions
(up to 2 slices to each side)

Click to change Path colours: 🟩🟪

Use colors / labels from: [None]

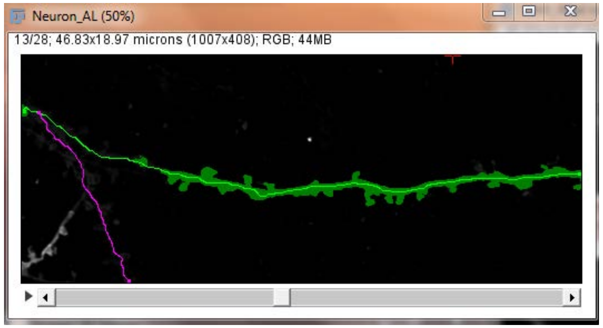
☐ Show only selected paths

☐ Use Tubular Geodesics

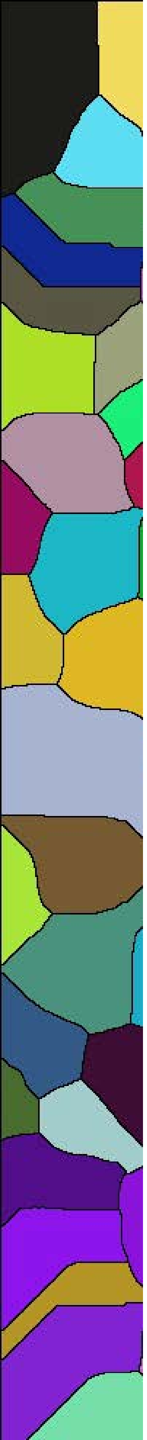
☐ Hessian-based analysis

☐ Use preprocessed image

σ = 0,0465, multiplier = 4,0000



13/28; 46.83x18.97 microns (1007x408); RGB; 44MB



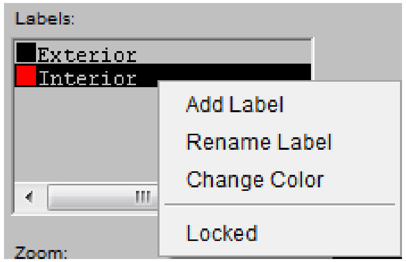
Segmentation Editor example I

1. If necessary, download and install **Trainable WEKA Segmentation** plugin.

http://fiji.sc/Segmentation_Editor

2. Open the Z stack and Segmentation Editor plugin

3. New labels or segmented areas can be added if necessary (right clic of the mouse over a previous label)

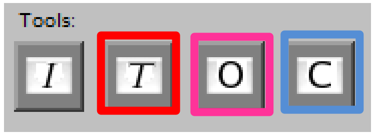


Labels:

Exterior

Interior

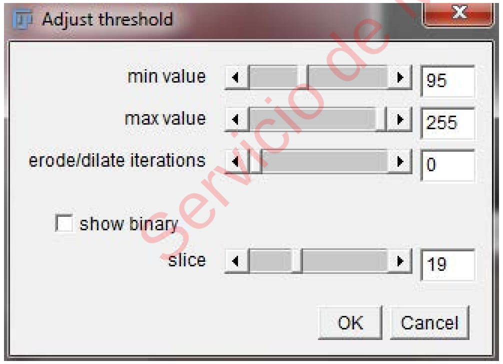
4. Segmented areas can be created by intensity thresholding:



4.1. Clic 'T' tool

4.2. Adjust min and max intensity value. If necessary, use the erode/dilate iterations slider to modify threshold result

4.3. A selection will be created around the thresholded pixels. Selection morphology can be modified with 'O' (Open) and 'C' (Close) .



Adjust threshold

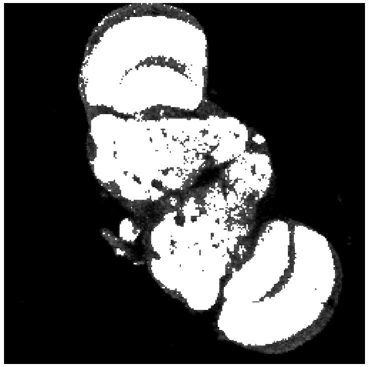
min value 95

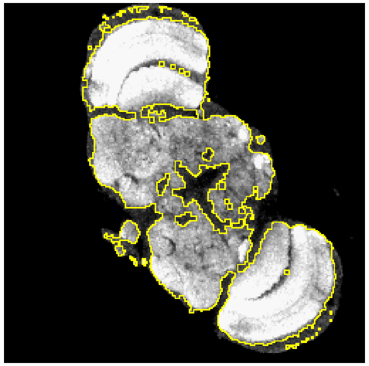
max value 255

erode/dilate iterations 0

□ show binary

slice 19

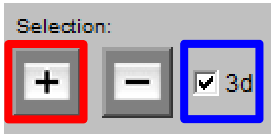


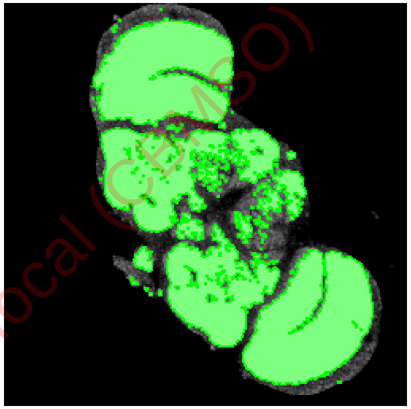


Segmentation Editor example II

5. Clic the **3d box** if selection includes more than one slice of the stack.

6. To assign the selection to one of the labels, clic ‘**+**’ button.





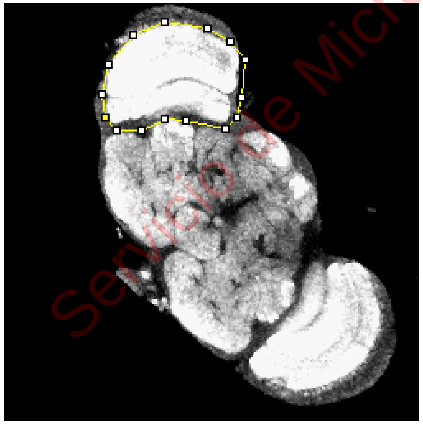
7. Segmented areas can also be created manually:

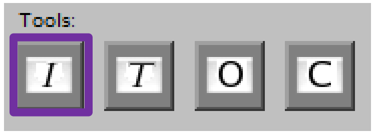
7.1. Use any of the drawing tools to create a selection

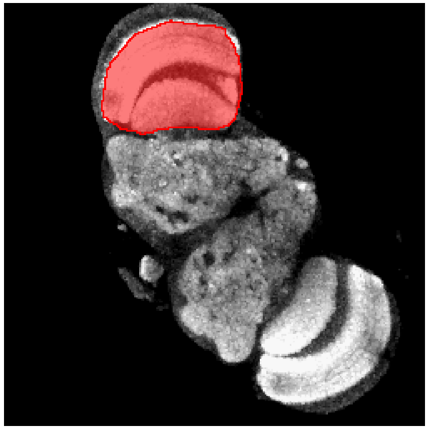
7.2. Move to a different slice of the Zstack and draw another

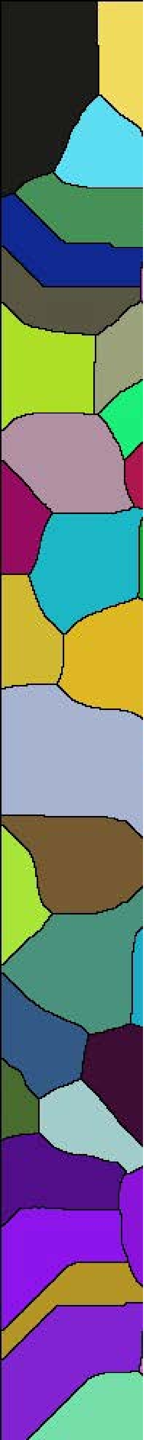
7.3. Use the ‘I’ tool to interpolate the selection shape between the slices.

7.4. Clic the **3d box** if selection includes more than one slice of the stack. To assign the selection to one of the labels, clic ‘**+**’ button.



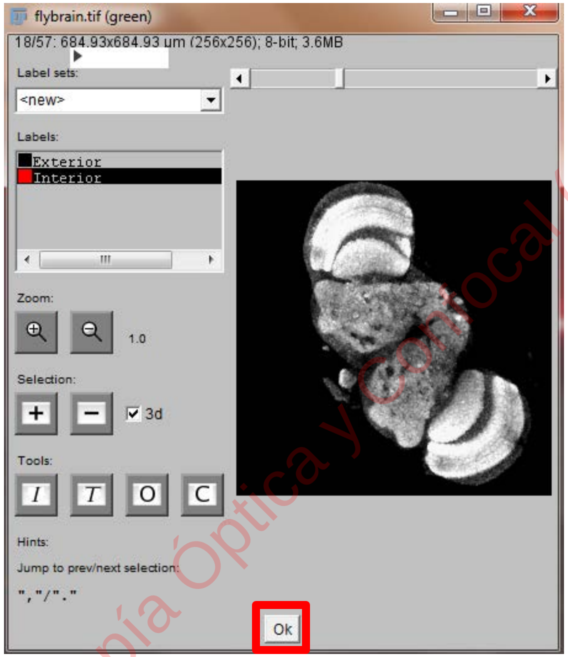






Segmentation Editor example III

8. Clic **OK** to finish the segmentation.



flybrain.tif (green)

18/57: 684.93x684.93 µm (256x256); 8-bit; 3.6MB

Label sets:

<new>

Labels:

Exterior

Interior

Zoom:

1.0

Selection:

☑ 3d

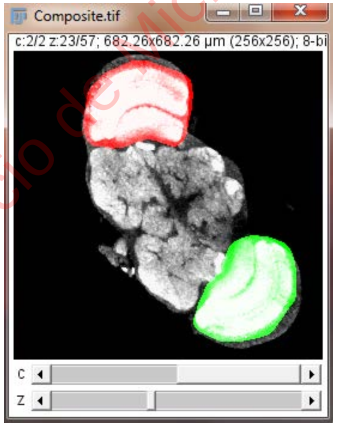
Tools:

Hints:

Jump to prev/next selection:

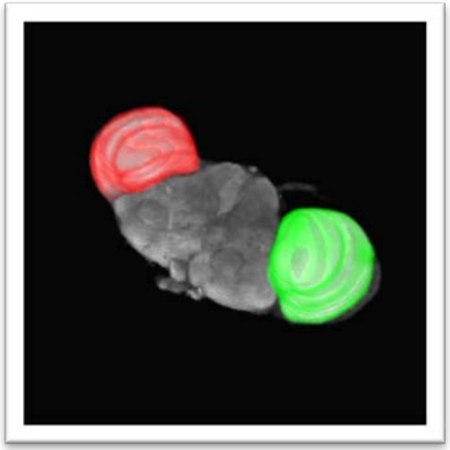
","/"."

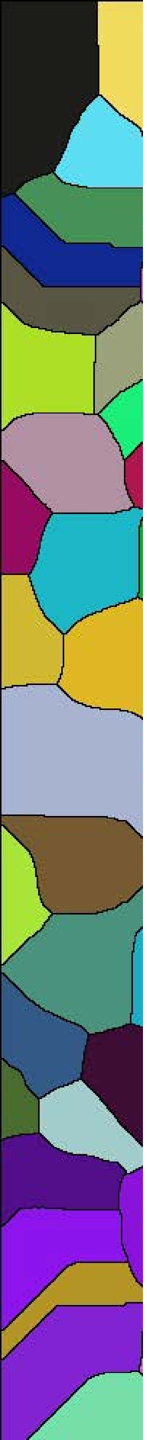
9. A composite image can be created with the labels and the original image. 3D Viewer plugin can be used for 3D rendering.



Composite.tif

c:2/2 z:23/57; 682.26x682.26 µm (256x256); 8-bi





Conclusions

Image enhancement is a powerful tool for a better detection of image features but… use with caution!:

Final image must represent the original information

Check journal guidelines regarding image processing

Great variety of segmentation algorithms…

Which to use?

Depends on your image

Review of free software tools for image analysis of fluorescence cell micrographs (J. Microscopy 2015)

Plugins ImageJ SMOC Web page