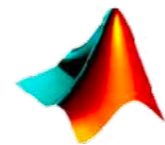


ImageJ/FIJI: “Amasijando los pixeles”



ImageJ
Image Processing & Analysis in Java



MATLAB



- Una aplicación de software (open source) para público general



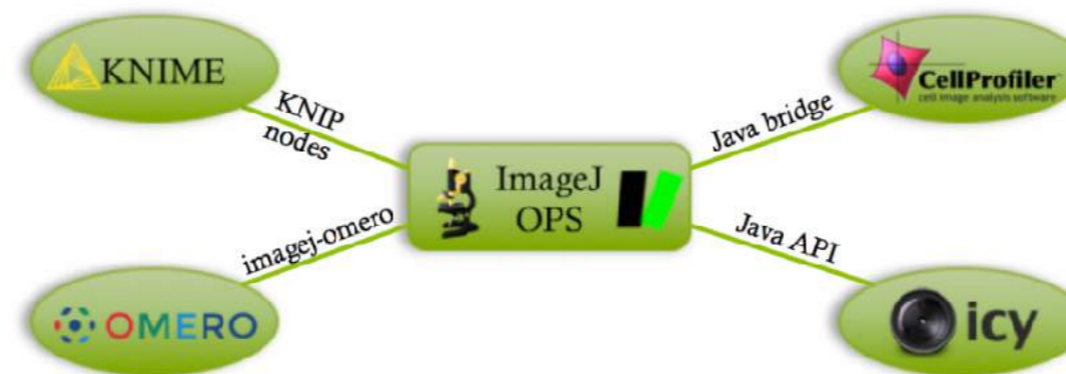
- Una biblioteca de software

```
1 public void loadAndDisplay(File file) {  
2     ImageJ ij = new ImageJ();  
3     Object data = ij.io().open(file);  
4     ij.ui().show(data);  
5 }
```

- Una colección de plugins y servicios



- “Write once, run anywhere” rutinas de procesamiento de imágenes



WELCOME



IMAGEJ

An open platform for scientific image analysis

[Download](#)[Learn](#)[Develop](#)

ImageJ is an [open source](#) image processing program designed for scientific multidimensional images.

ImageJ is highly [extensible](#), with thousands of [plugins](#) and [scripts](#) for performing a wide variety of tasks, and a [large user community](#).

Welcome to the Wiki!

This wiki documents all aspects of the **ImageJ ecosystem**, including:



ImageJ



Fiji



Plugins



Related software

Tools & Features

Open source ImageJ is a tool for the [scientific community](#). To maintain transparency, the [ImageJ application](#) and its [source code](#) will always be freely available.

Reproducible Powerful tools such as the [Script Editor](#) and [personal update sites](#) help you develop and share reproducible analysis workflows.

Interoperable ImageJ is [not an island](#). Use the best tool for the job, including [KNIME](#), [ITK](#), [MATLAB](#), and a multitude of [scripting](#) languages.

Join the Community

Image.sc Forum A great place to ask and answer questions, and become part of the community that has driven ImageJ's success.

GitHub Issues The [ImageJ team](#) uses [GitHub](#) for bug reports, technical suggestions and feature requests.

Contact and Help For more ways to communicate: [mailing lists](#), [chat](#) services and more.

ImageJ is developed by [contributors worldwide](#). This web site is hosted by [LOCI](#) at the University of Wisconsin-Madison.

Fiji Home Wiki Source Forum

Search



Fiji is an image processing package—a "batteries-included" distribution of [ImageJ](#), bundling a lot of plugins which facilitate scientific image analysis.

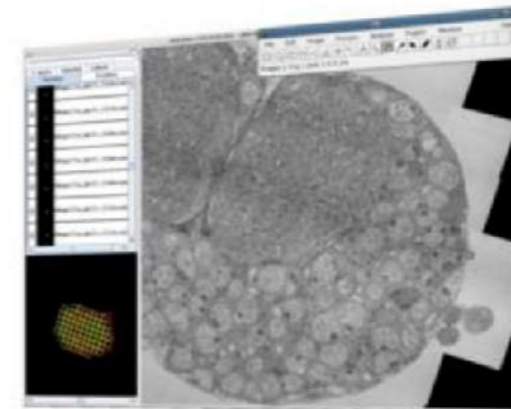
[Download »](#)

[Cite »](#)

[Contribute »](#)

TrakEM2

Perform morphological data mining, three-dimensional modeling, image stitching, registration, editing and annotation.



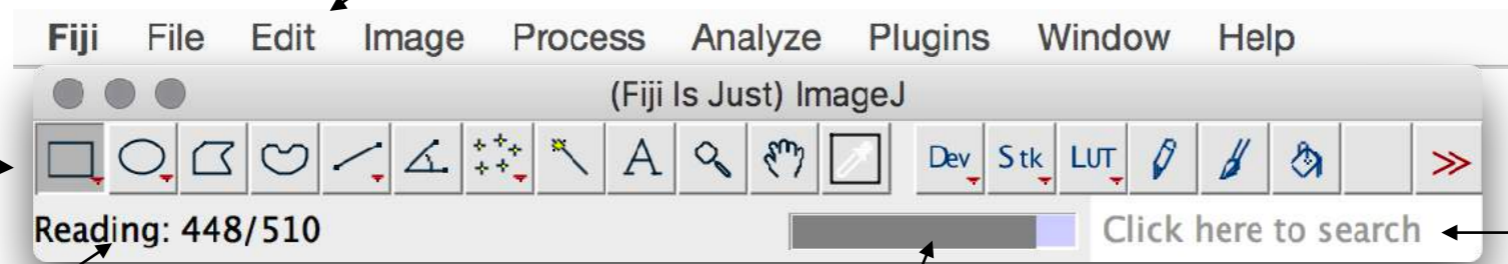
Fiji is [easy to install](#) and has an automatic update function, bundles a lot of plugins and offers comprehensive documentation.

Like ImageJ itself, Fiji is an [open source](#) project hosted in [Git](#) version control [repositories](#), with access to the source code of all internals, libraries and plugins, easing the [development](#) and [scripting](#) of plugins.

Fiji is licensed under the [GNU General Public License](#). It builds on top of the [ImageJ2](#) core, which is licensed under the permissive [BSD 2-clause license](#). Plugins and other components have [their own licenses](#).

Barra de herramientas

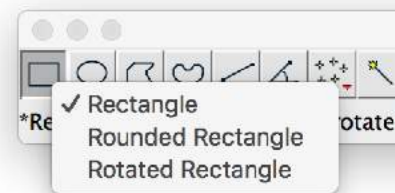
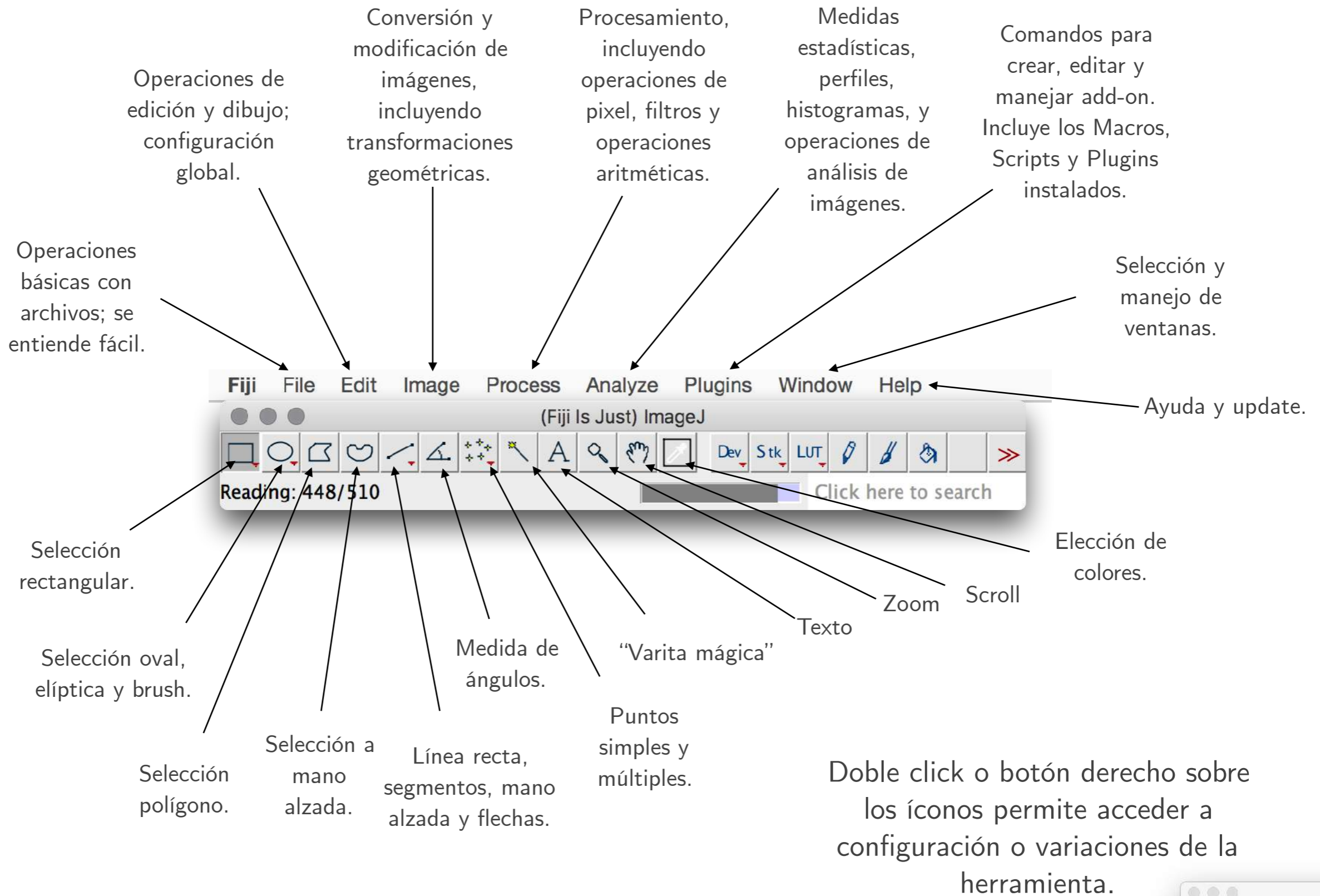
Barra de menu

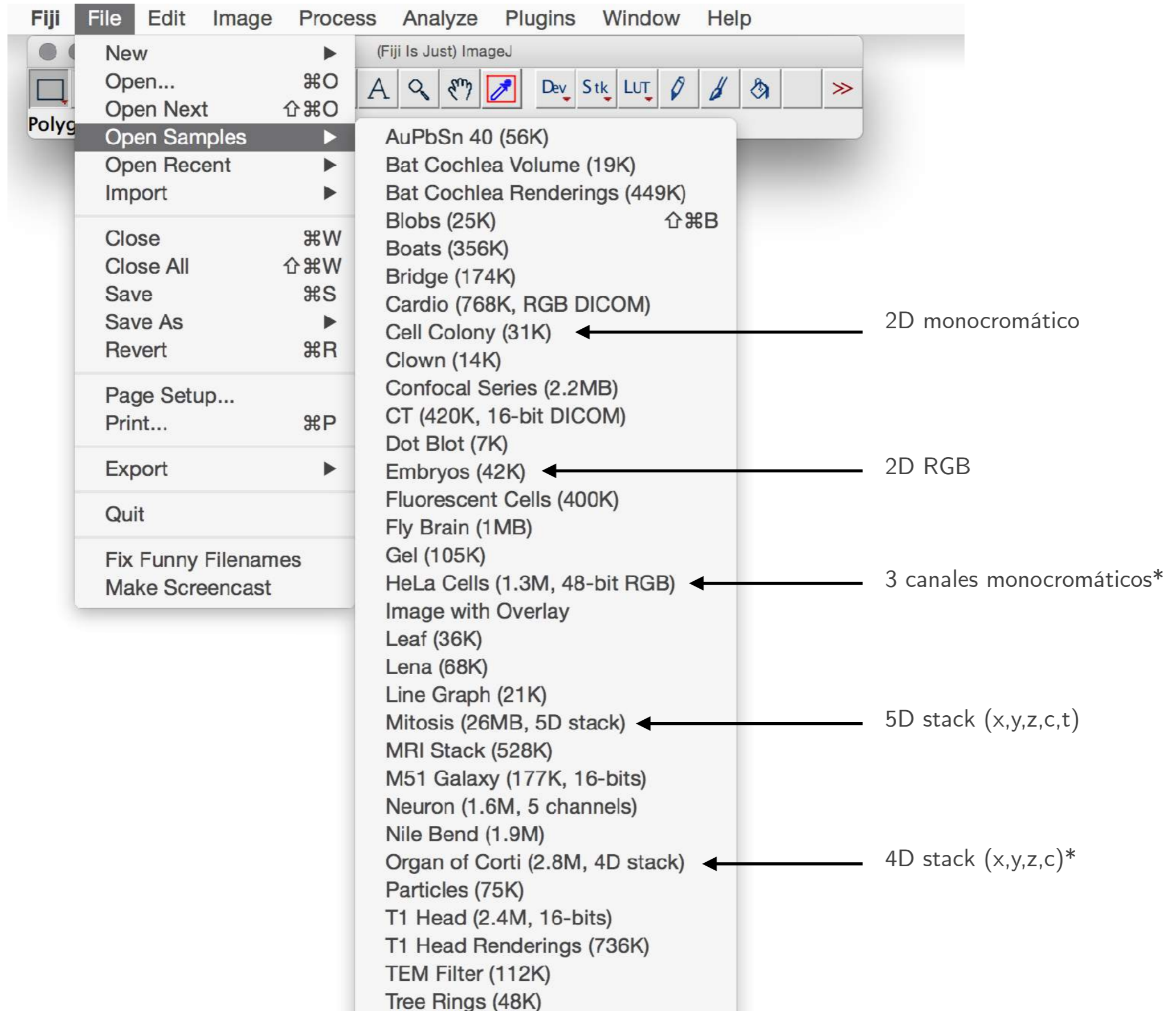


Barra de búsqueda
(Control+L)

Barra de estado

Barra de progreso





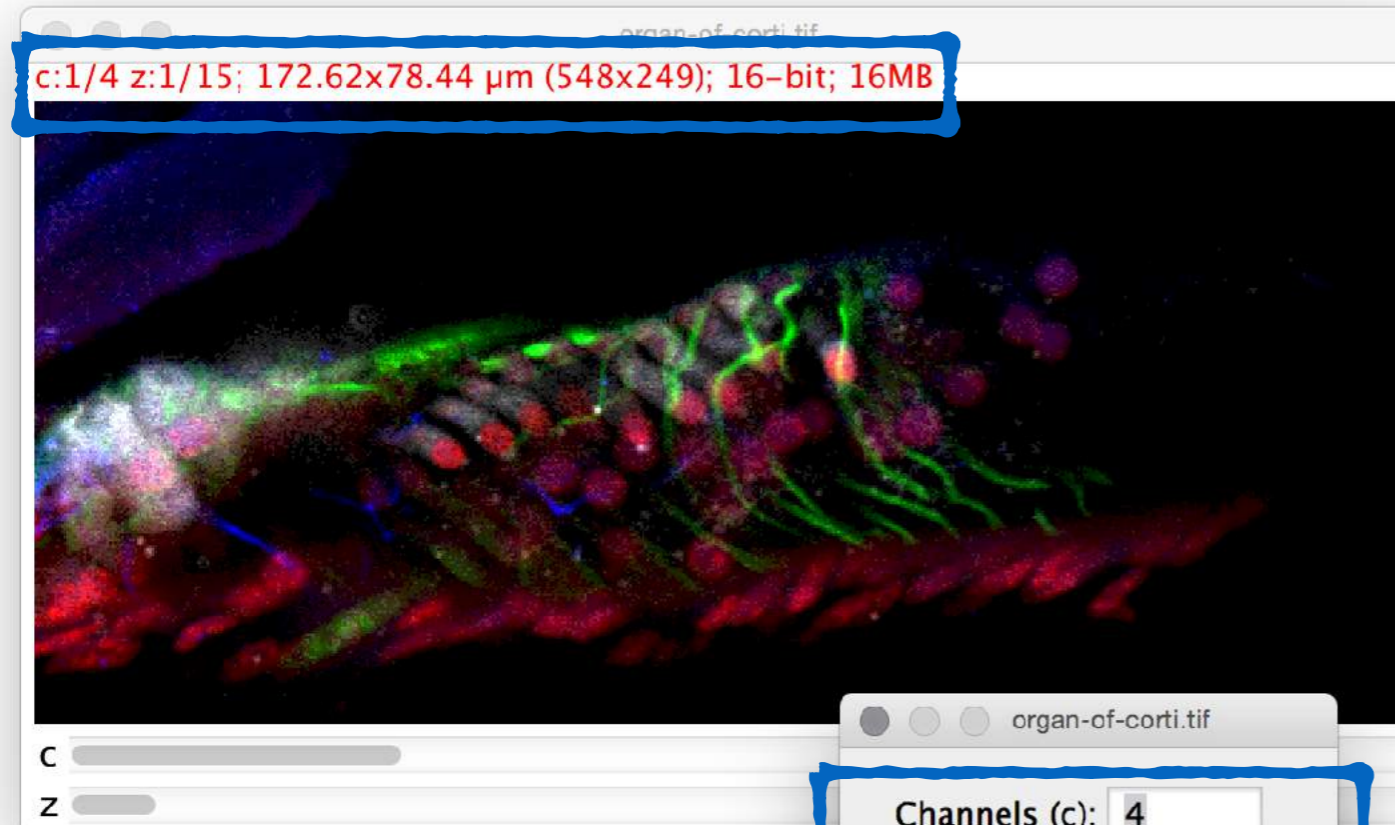
The image shows the 'File' menu of the Fiji software. The menu items are as follows:

- New
- Open... ⌘O
- Open Next ⇧⌘O
- Open Samples ▶
- Open Recent ▶
- Import ▶
- Close ⌘W
- Close All ⇧⌘W
- Save ⌘S
- Save As ▶
- Revert ⌘R
- Page Setup...
- Print... ⌘P
- Export ▶
- Quit
- Fix Funny Filenames
- Make Screenshot

The 'Open Samples' submenu is open, listing various test images with their sizes and formats. Annotations with arrows point to specific items:

- Cell Colony (31K) → 2D monocromático
- Embryos (42K) → 2D RGB
- HeLa Cells (1.3M, 48-bit RGB) → 3 canales monocromáticos*
- Mitosis (26MB, 5D stack) → 5D stack (x,y,z,c,t)
- Organ of Corti (2.8M, 4D stack) → 4D stack (x,y,z,c)*

The Fiji window title is '(Fiji Is Just) ImageJ'. The toolbar shows various tools, with the 'Open' tool (represented by a blue square with a white arrow) highlighted by a red box.



[Fiji]

organ-of-corti.tif

Channels (c): 4
Slices (z): 15
Frames (t): 1
Note: c*z*t must equal 60
Unit of length: μm
Pixel width: 0.3150001
Pixel height: 0.3150001
Voxel depth: 0.3150001
Frame interval: 0 sec
Origin (pixels): 0,0

Global

Cancel OK

Image > Show Info (Control + I)

Info for organ-of-corti.tif

I'm attaching a 4 channel Hyperstack. Hope it gets through at 10.2 M. I cropped the file significantly. The sample is the organ of Corti, the auditory sensory organ, within the cochlea of a young mouse. The cochlea was cleared in a mixture of methyl salicylate and benzyl benzoate after completion of immunolabeling to allow focusing through the tissue with minimal spherical aberration. Colors: gray is parvalbumin labeled with Cy5 in sensory cells (inner and outer hair cells) and some small fibers; red is DNA labeled by DAPI; green is acetylated tubulin labeled with Alexa488 in some support cells (pillar cells and the phalangeal processes of Dieter's cells supporting the outer hair cells) and blue is kD neurofilament protein labeled with Alexa568 in afferent nerve fibers. Both types of nerve fibers enter the organ from the left, most innervate the inner hair cells (gray blobs on the left) and many pass across through the pillar and Dieter's cells processes to innervate the outer hair cells. Olympus FV-1000 confocal, 20/.75 Plan-Fluor, .315 μm/pixel, .3 μm z-step. This is the first 4.5 μm from a volume 60 μm thick.

FYI, a companion image from a different cochlea took 2nd in the Olympus Bioscapes competition. More of these images were in the back two pages of October, 2007 Biophotonics International.

Glen MacDonald, University of Washington

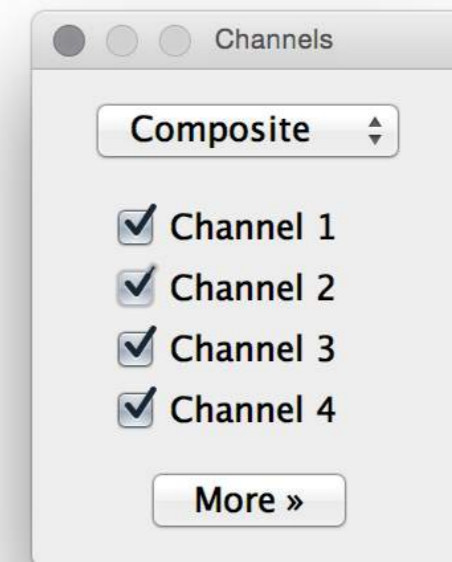
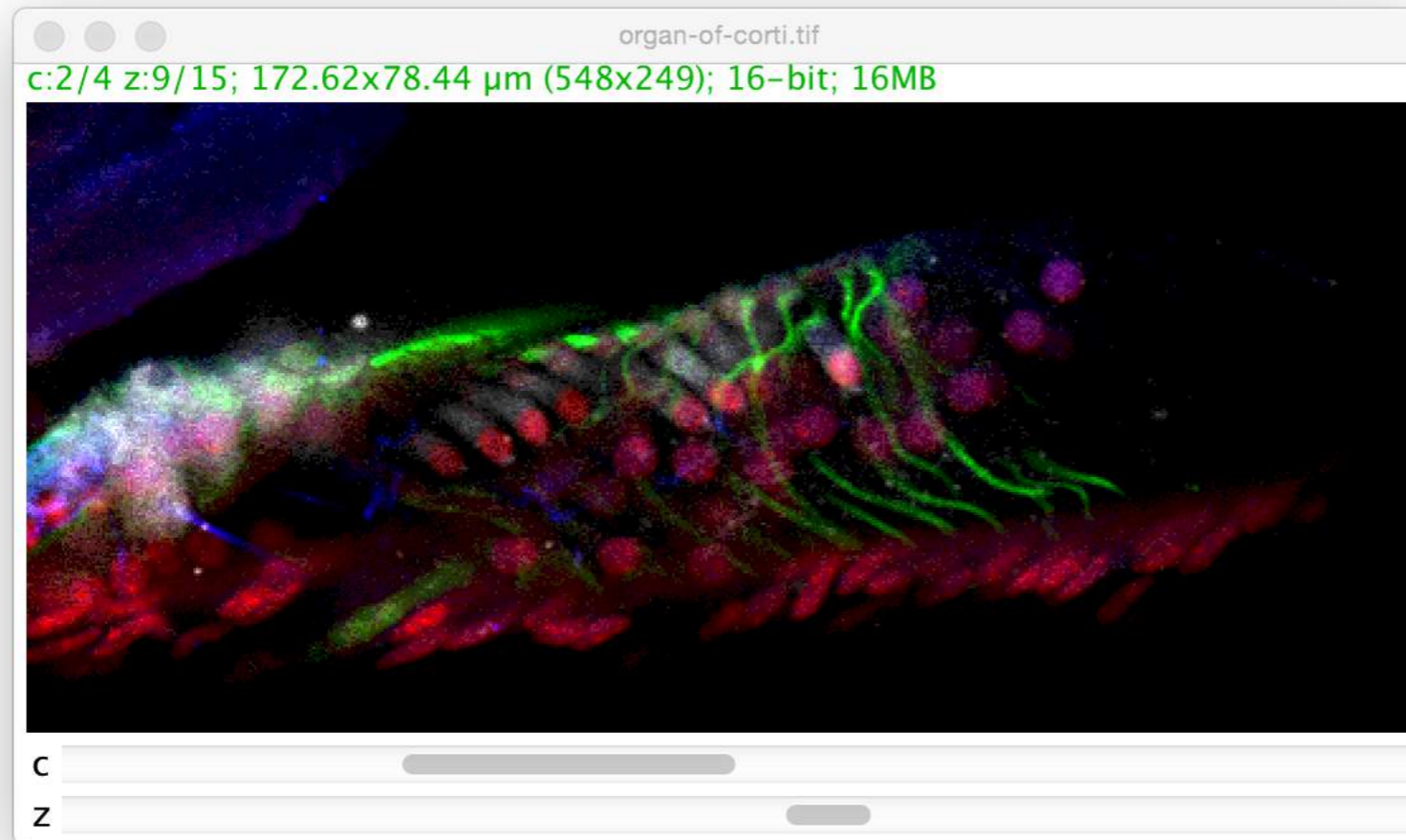
(Fiji Is Just) ImageJ 2.0.0-rc-49/1.51d; Java 1.8.0_66 [64-bit]; Mac OS X

Title: organ-of-corti.tif
Width: 172.6201 μm (548)
Height: 78.4350 μm (249)
Depth: 4.7250 μm (15)
Size: 16MB
Resolution: 3.1746 pixels per μm
Voxel size: 0.3150x0.3150x0.3150 μm³
ID: -91
Bits per pixel: 16 (unsigned)

Metadatos

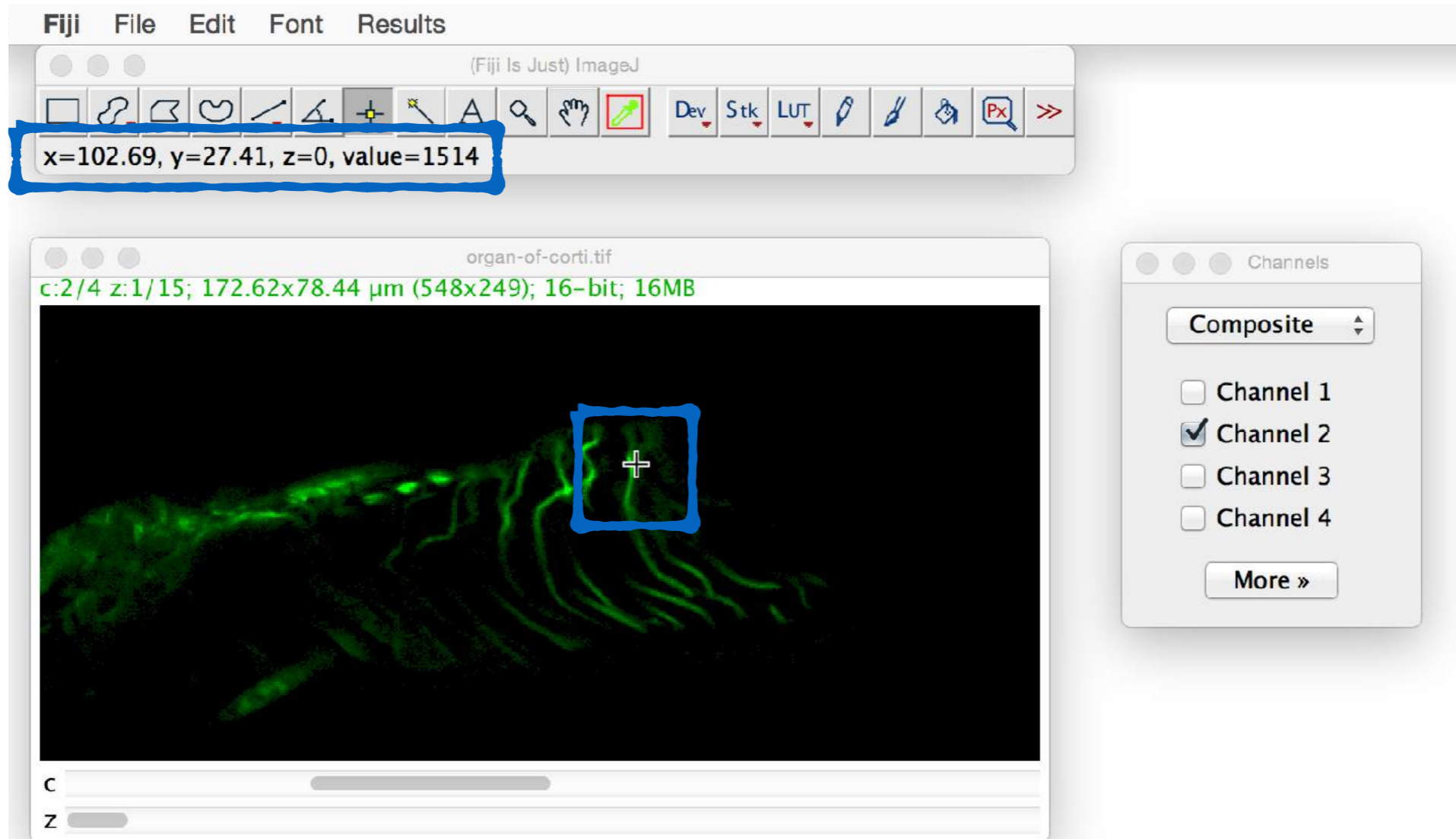
Image > Properties
(Control + Shift + P)

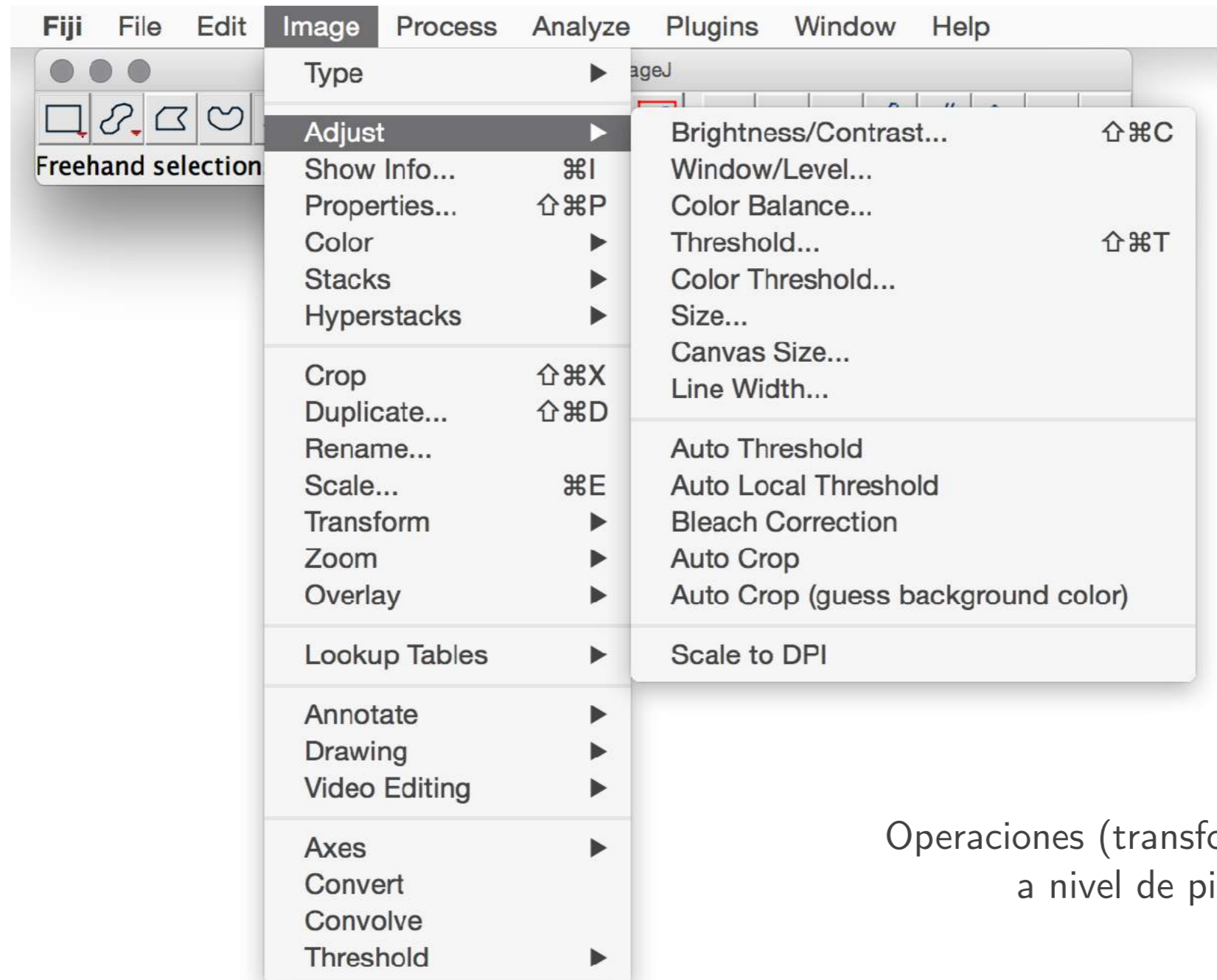
Image > Color > Channel Tools
(Shift + Z)



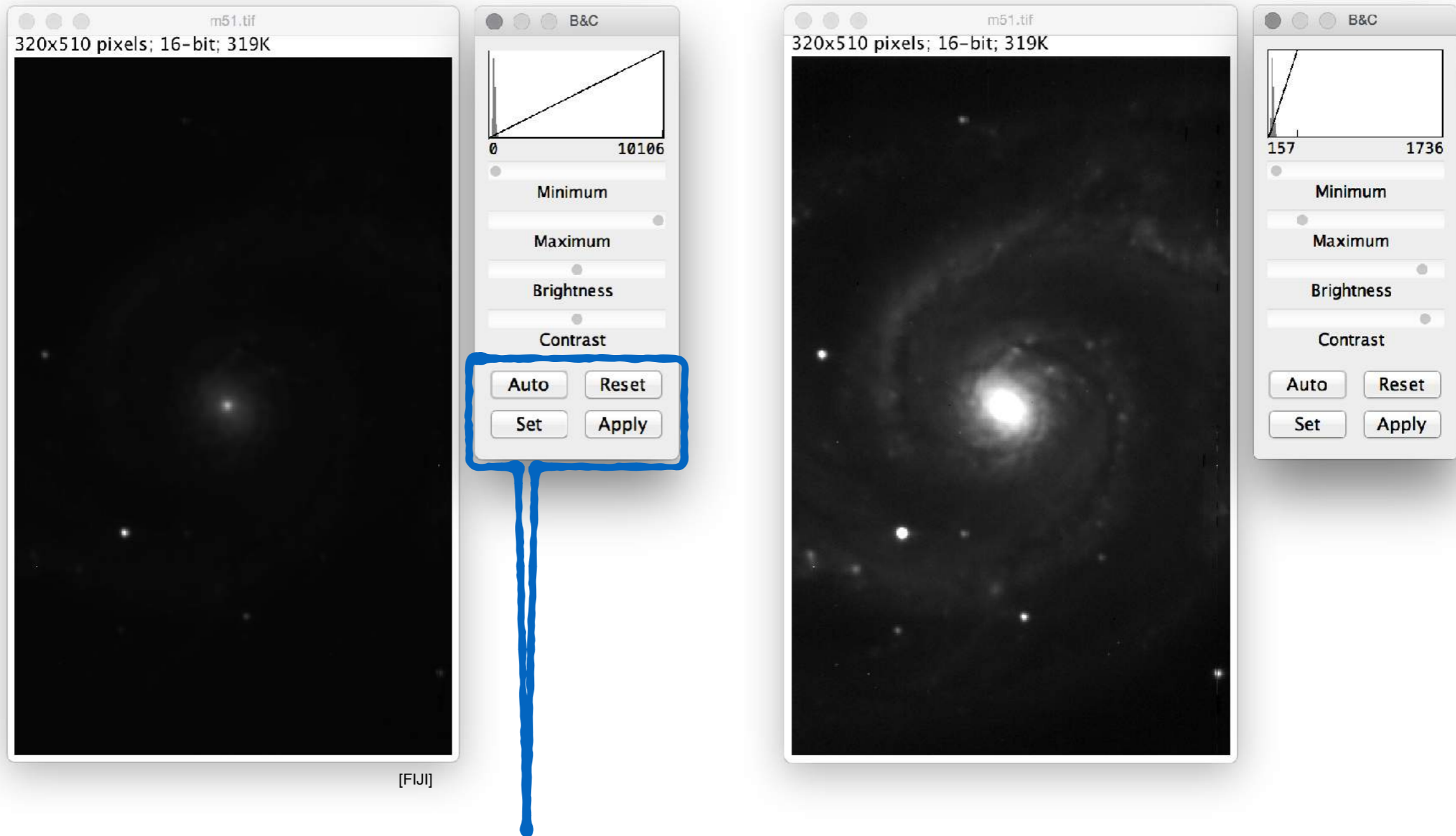
[Fiji]

En μm .





Operaciones (transformaciones)
a nivel de pixel.



Auto: define valores automáticamente.

Set: permite definir valores sin usar las barras de desplazamiento.

Reset: vuelve los valores a los predefinidos.

Apply: aplica la transformación modificando los datos en la imagen.

Image > Adjust > Auto Threshold

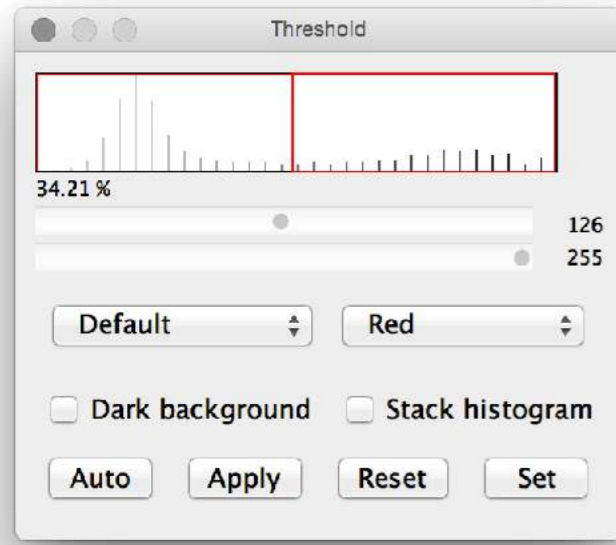
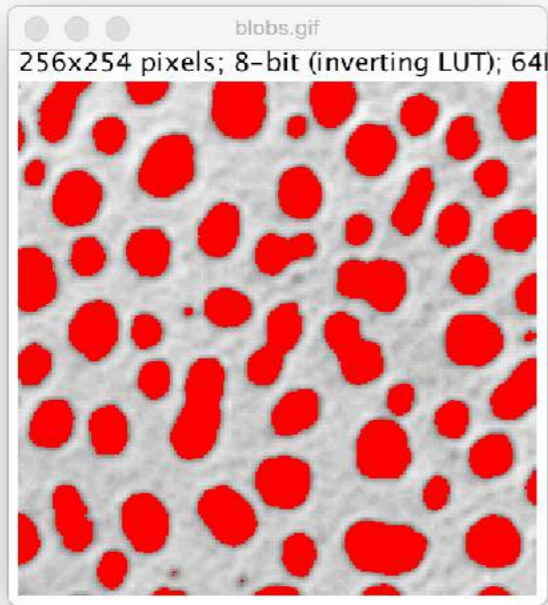
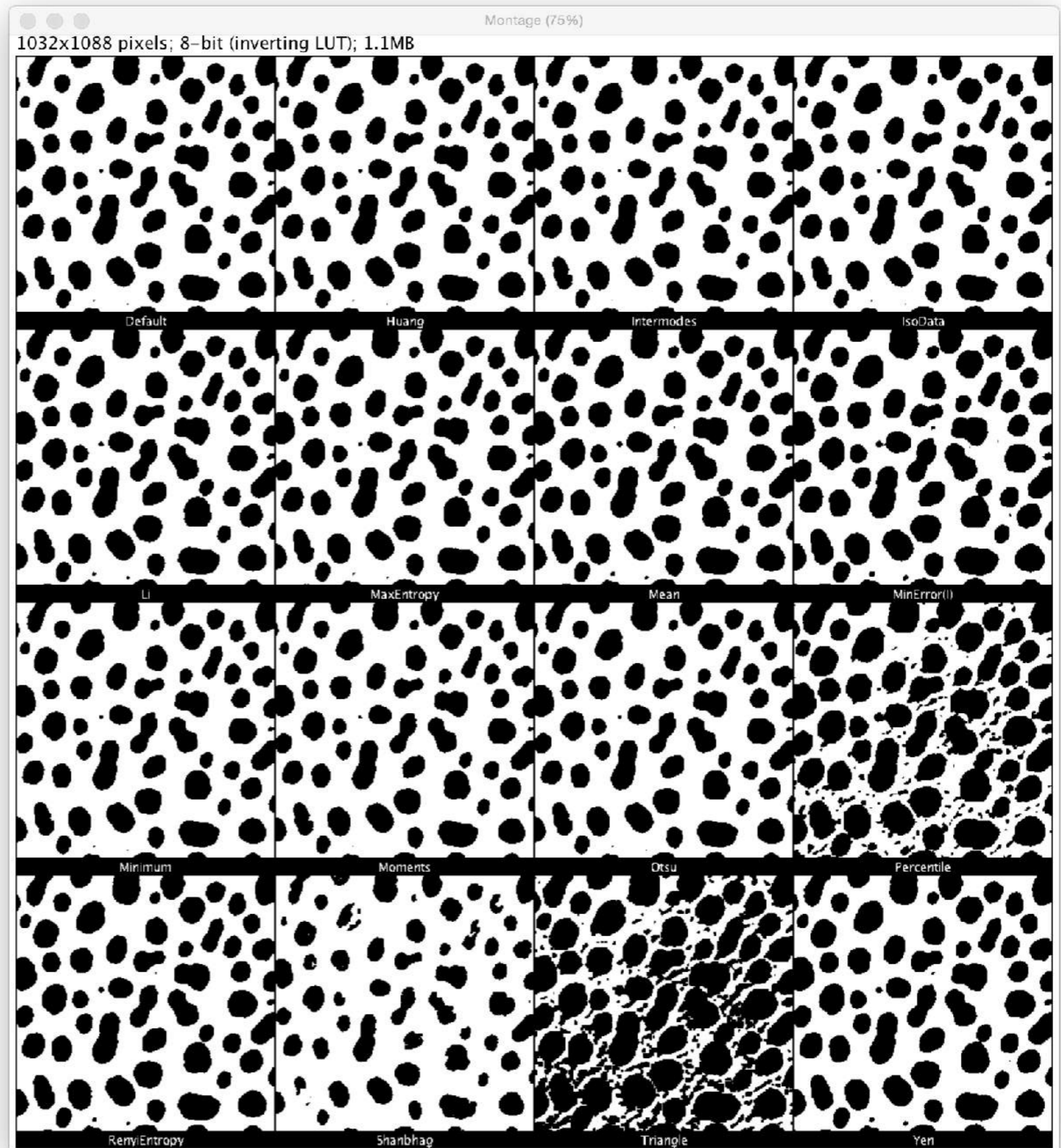
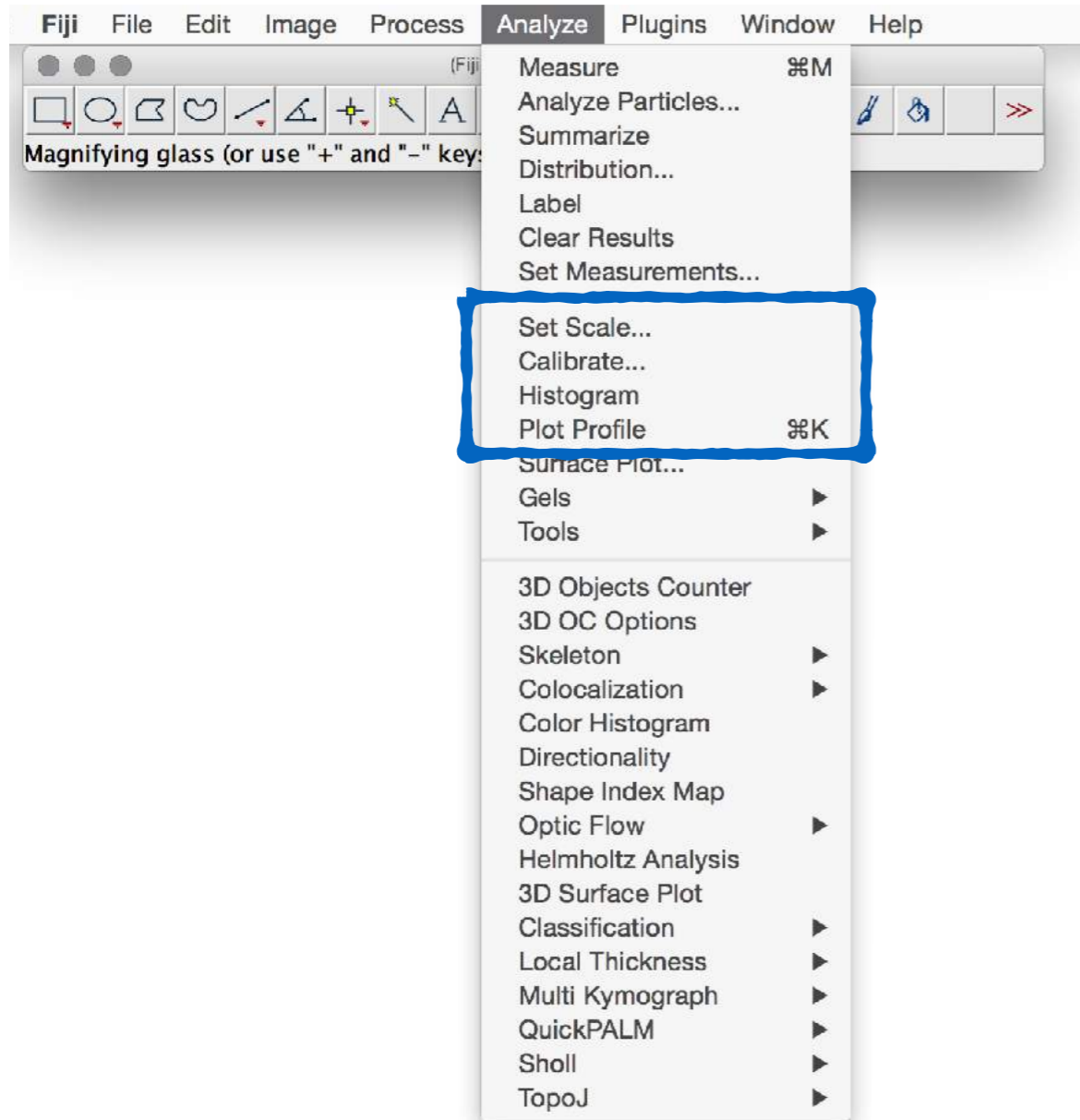
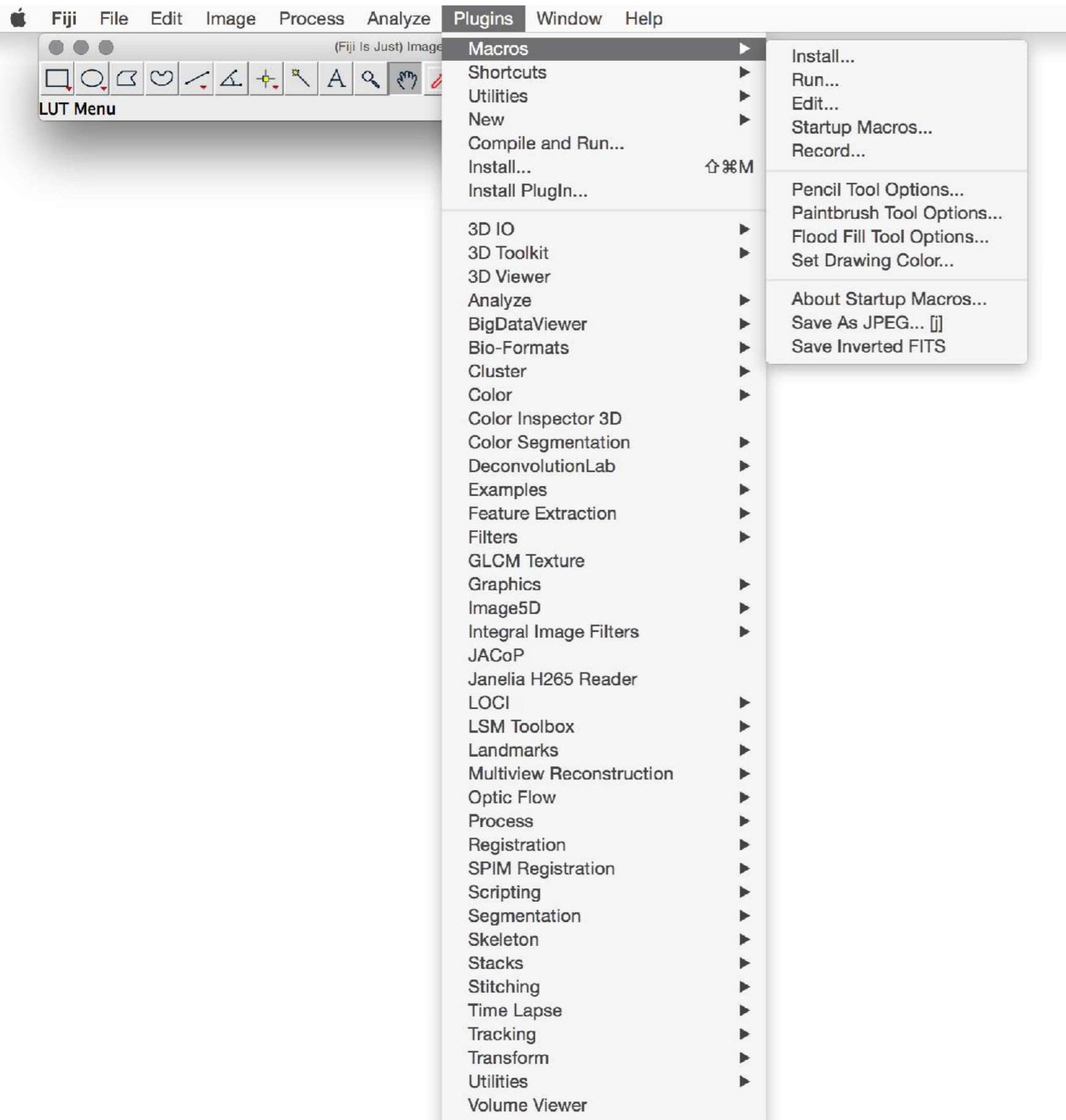


Image > Adjust > Threshold
(Shift + T)

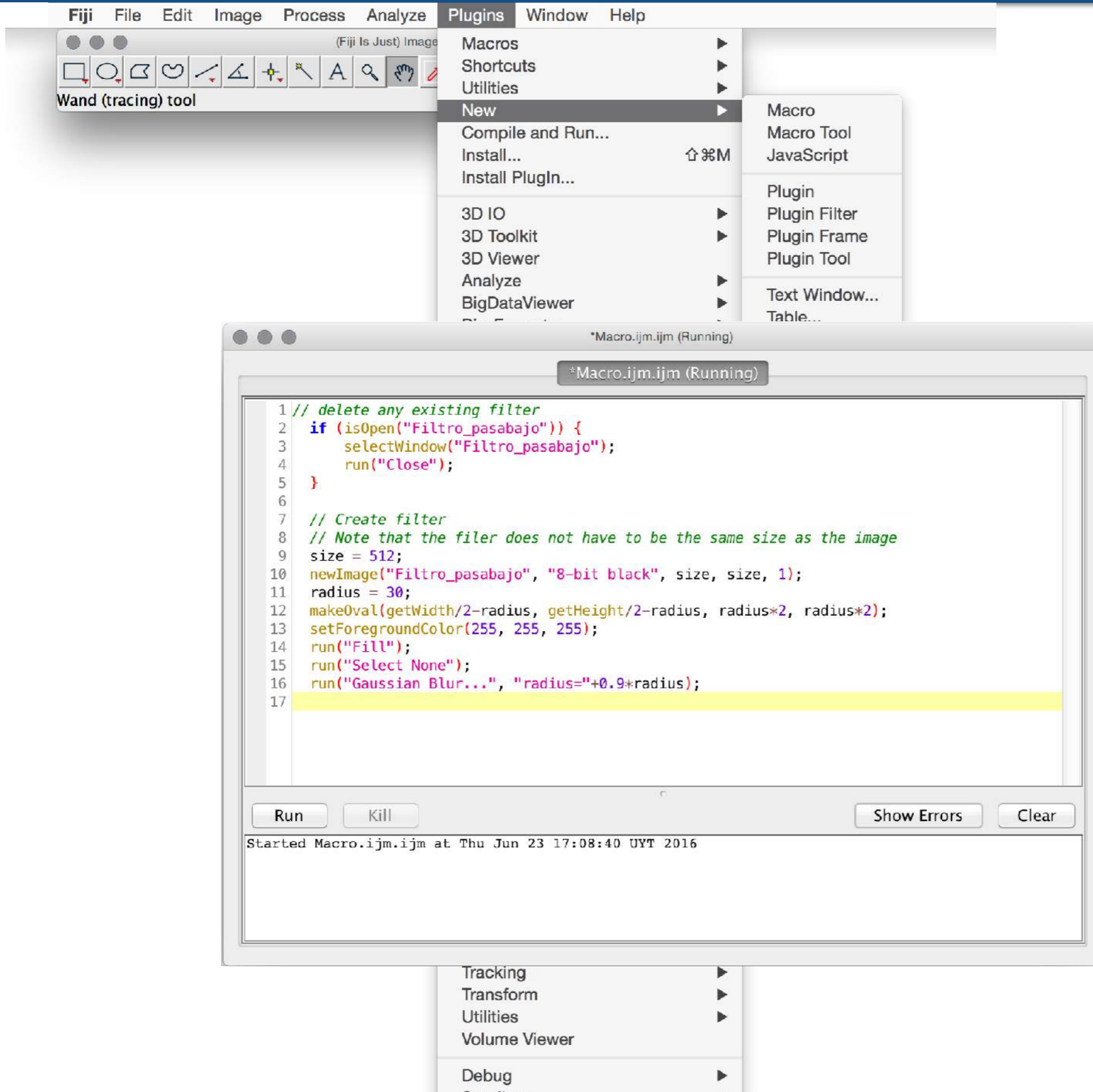




Volvemos en un rato...



Macros y Plugins

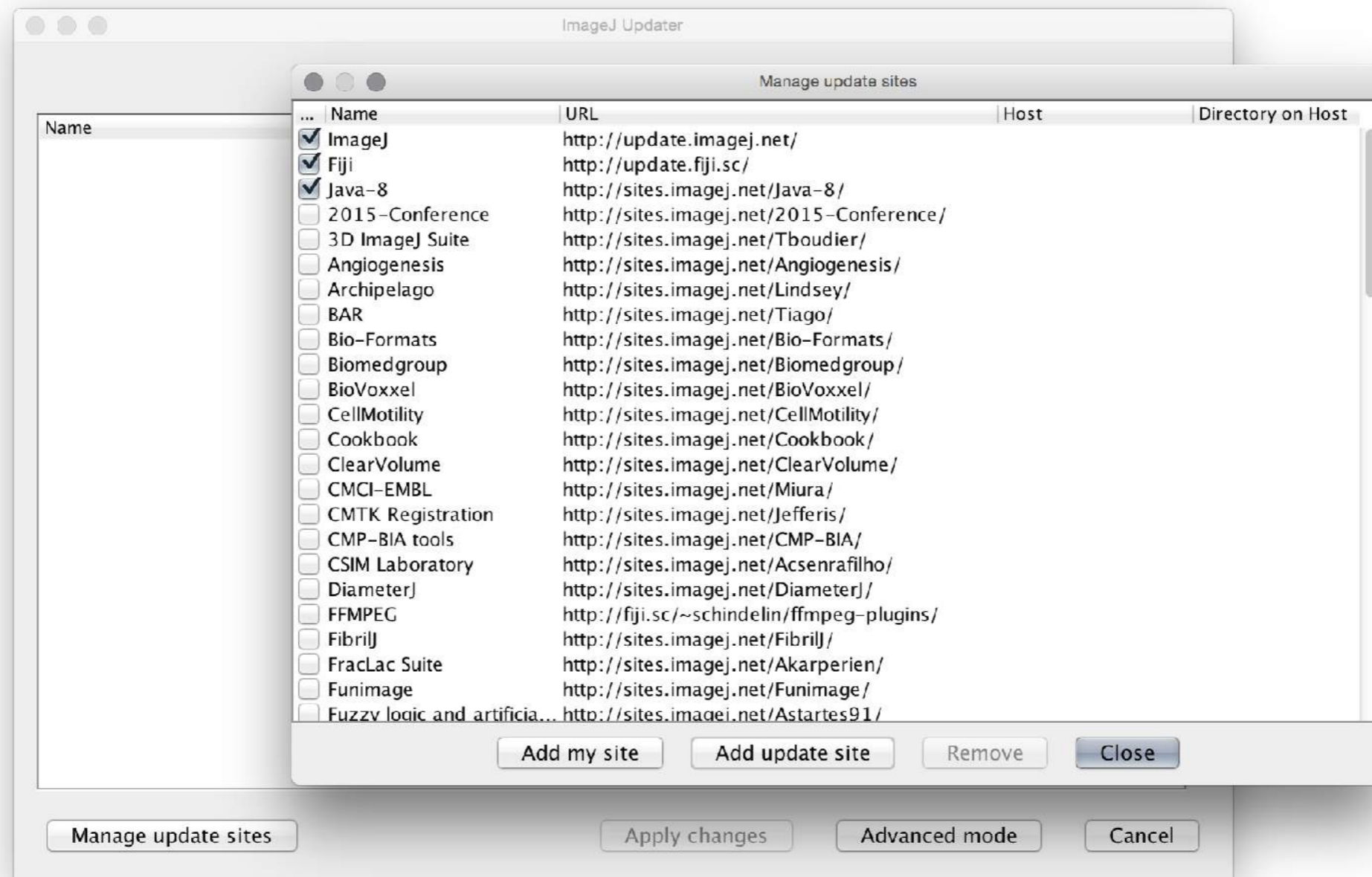


The image shows the Fiji software interface. The top menu bar includes 'Fiji', 'File', 'Edit', 'Image', 'Process', 'Analyze', 'Plugins', 'Window', and 'Help'. The 'Plugins' menu is open, showing options like 'Macros', 'Shortcuts', 'Utilities', 'New', 'Compile and Run...', 'Install...', 'Install PlugIn...', '3D IO', '3D Toolkit', '3D Viewer', 'Analyze', and 'BigDataViewer'. The 'New' submenu is also open, listing 'Macro', 'Macro Tool', 'JavaScript', 'Plugin', 'Plugin Filter', 'Plugin Frame', 'Plugin Tool', 'Text Window...', and 'Table...'. Below the menu, a 'Wand (tracing) tool' is visible. In the foreground, a window titled '*Macro.ijm.ijm (Running)' is open, displaying a macro script:

```
1 // delete any existing filter
2 if (isOpen("Filtro_pasabajo")) {
3     selectWindow("Filtro_pasabajo");
4     run("Close");
5 }
6
7 // Create filter
8 // Note that the filter does not have to be the same size as the image
9 size = 512;
10 newImage("Filtro_pasabajo", "8-bit black", size, size, 1);
11 radius = 30;
12 makeOval(getWidth/2-radius, getHeight/2-radius, radius*2, radius*2);
13 setForegroundColor(255, 255, 255);
14 run("Fill");
15 run("Select None");
16 run("Gaussian Blur...", "radius="+0.9*radius);
17
```

At the bottom of the macro editor window, there are buttons for 'Run', 'Kill', 'Show Errors', and 'Clear'. Below these buttons, a status bar indicates: 'Started Macro.ijm.ijm at Thu Jun 23 17:08:40 UYT 2016'. The bottom of the image shows the 'Tracking', 'Transform', 'Utilities', 'Volume Viewer', and 'Debug' menus.

- Instalación:
 - “Drag and drop”
 - Copiar a la carpeta FIJI/plugins
 - Seleccionar/agregar en el “Manager update site”



Histogramas

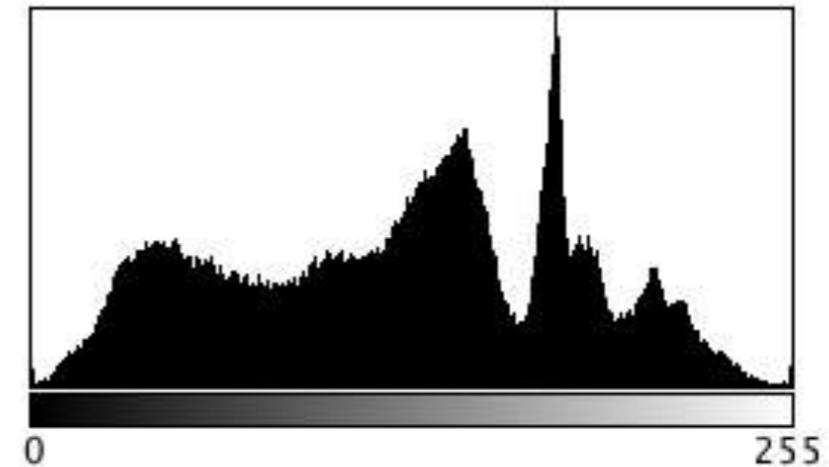
- El histograma es una **conteo** de los posibles valores que toma una señal (imagen).

$$I(u, v)$$



[Burguer & Burge]

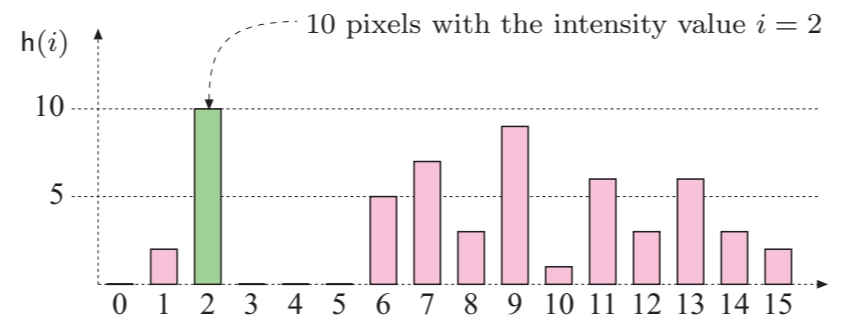
$$h(i)$$



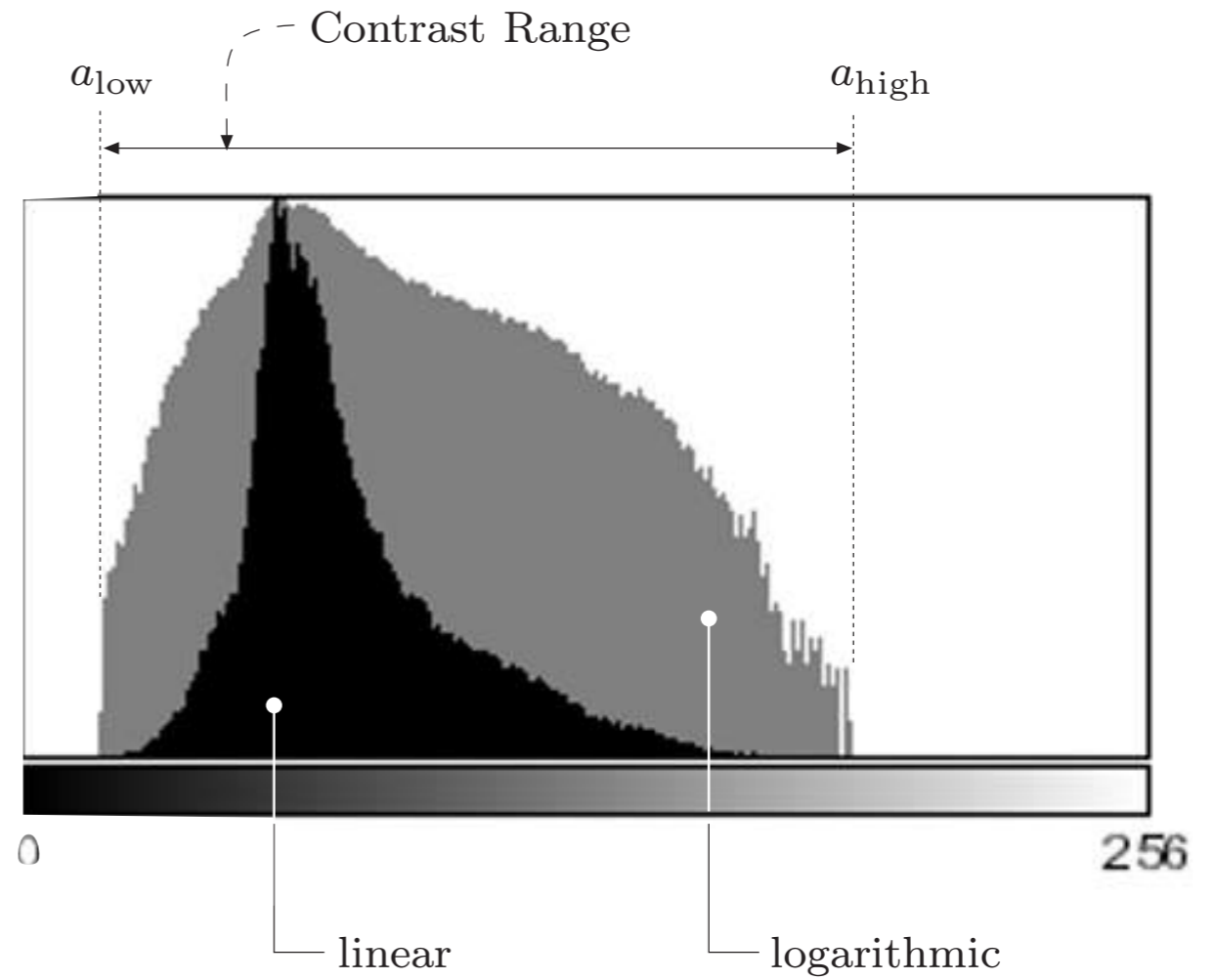
Count: 230464 Min: 0
 Mean: 123.536 Max: 255
 StdDev: 57.639 Mode: 176 (2998)

FIJI: Analyze > Histogram

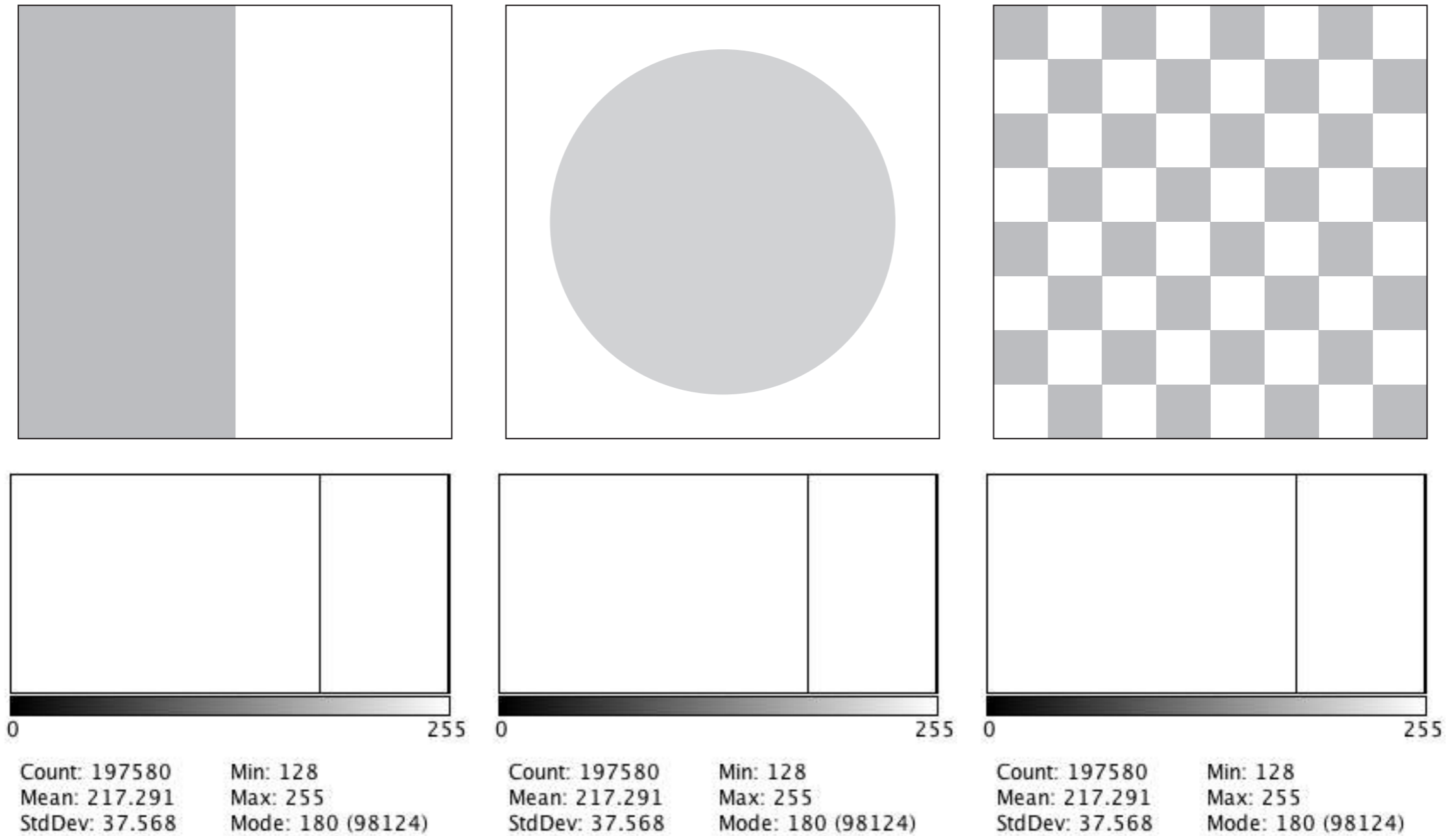
$$h(i) = \text{cardinal}\{(u, v) | I(u, v) = i\} \text{ para todos los } 0 \leq i \leq K$$



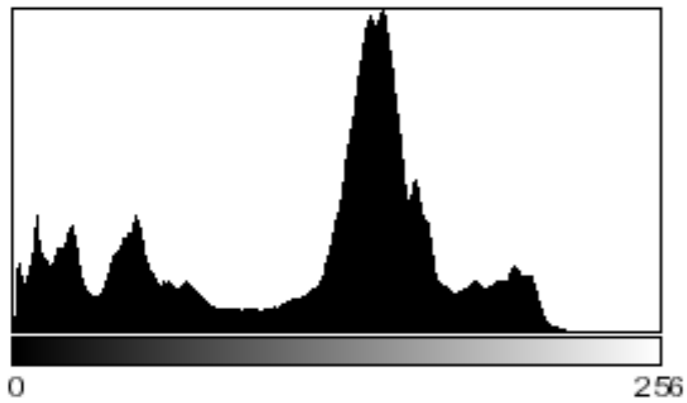
$h(i)$	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



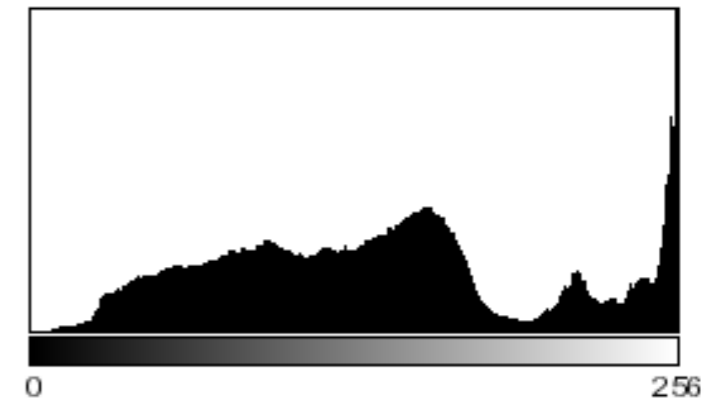
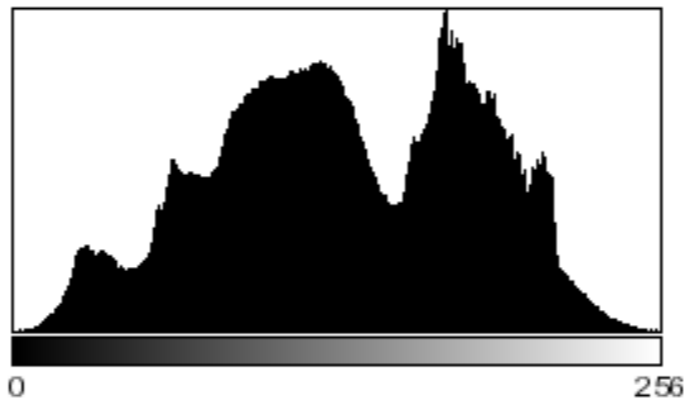
- Información global



- Exposición

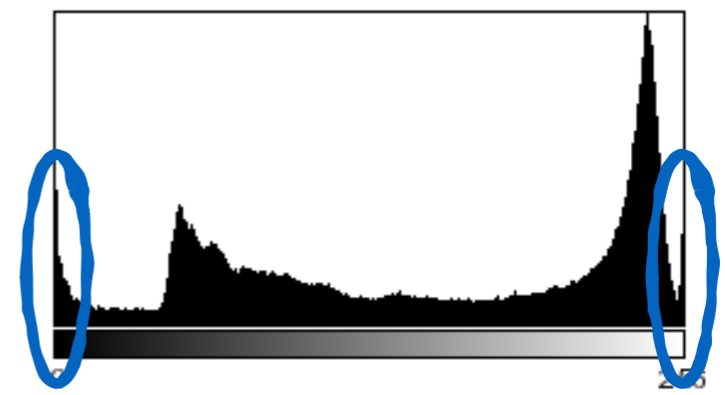
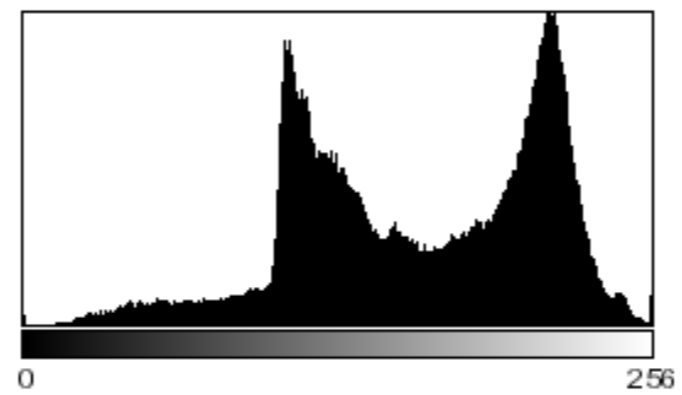
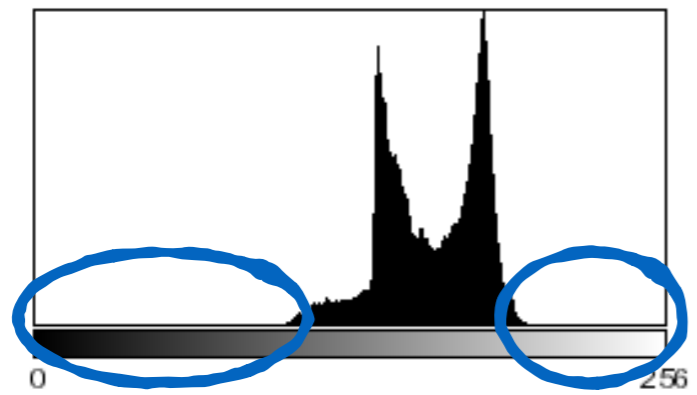


Sub-exposición

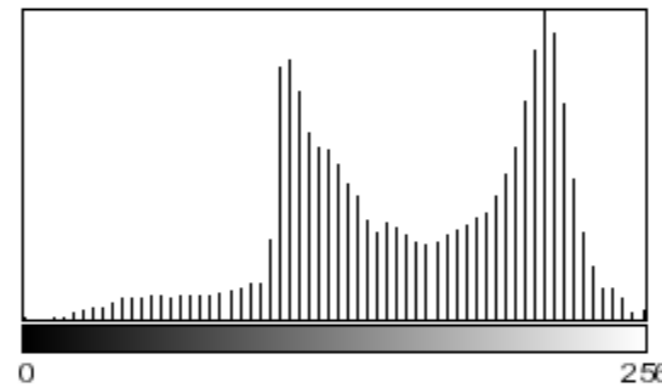
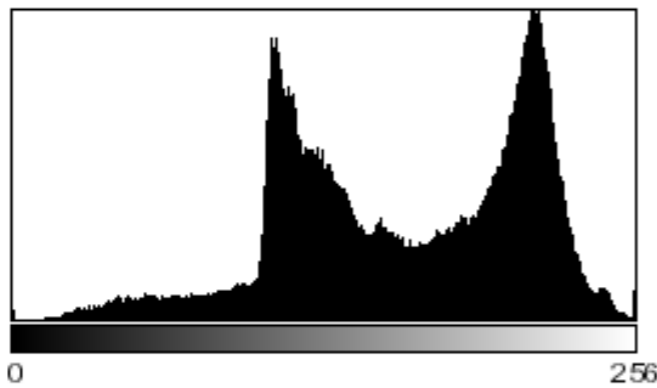
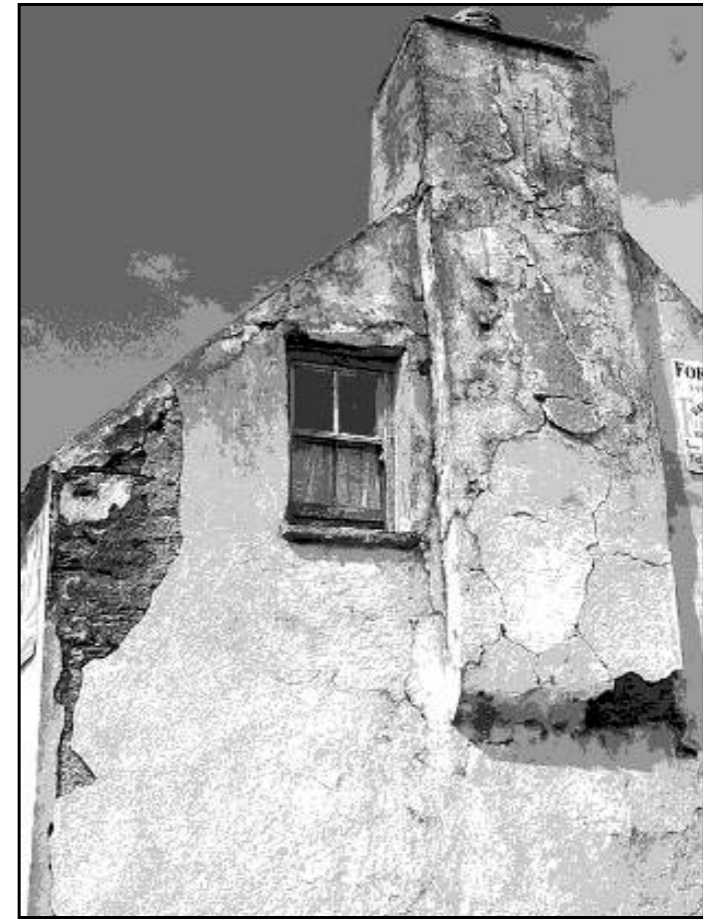


Sobre-exposición

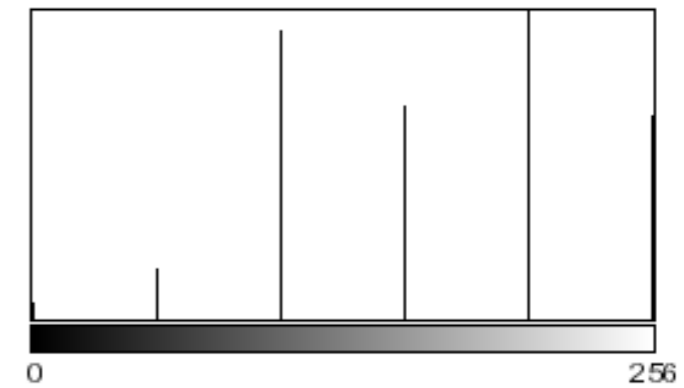
- Rango dinámico



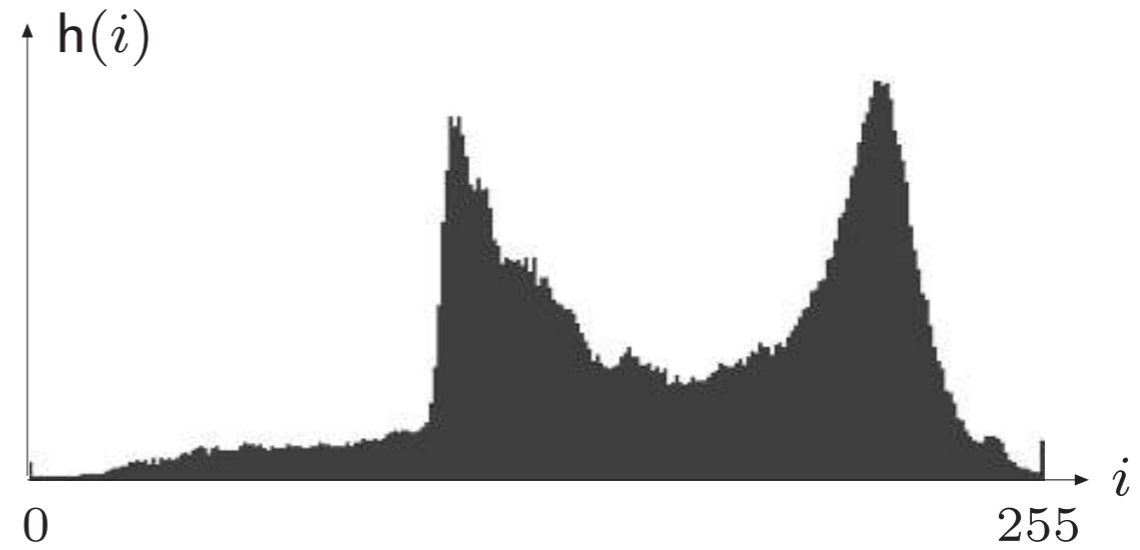
- Cuantización de intensidades



64 intensidades

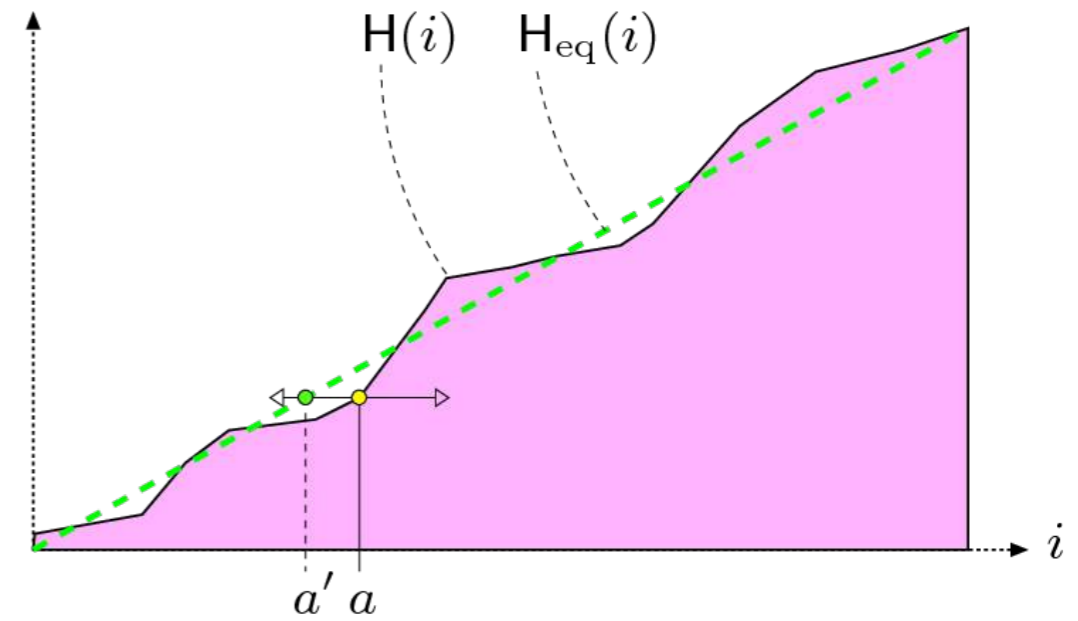
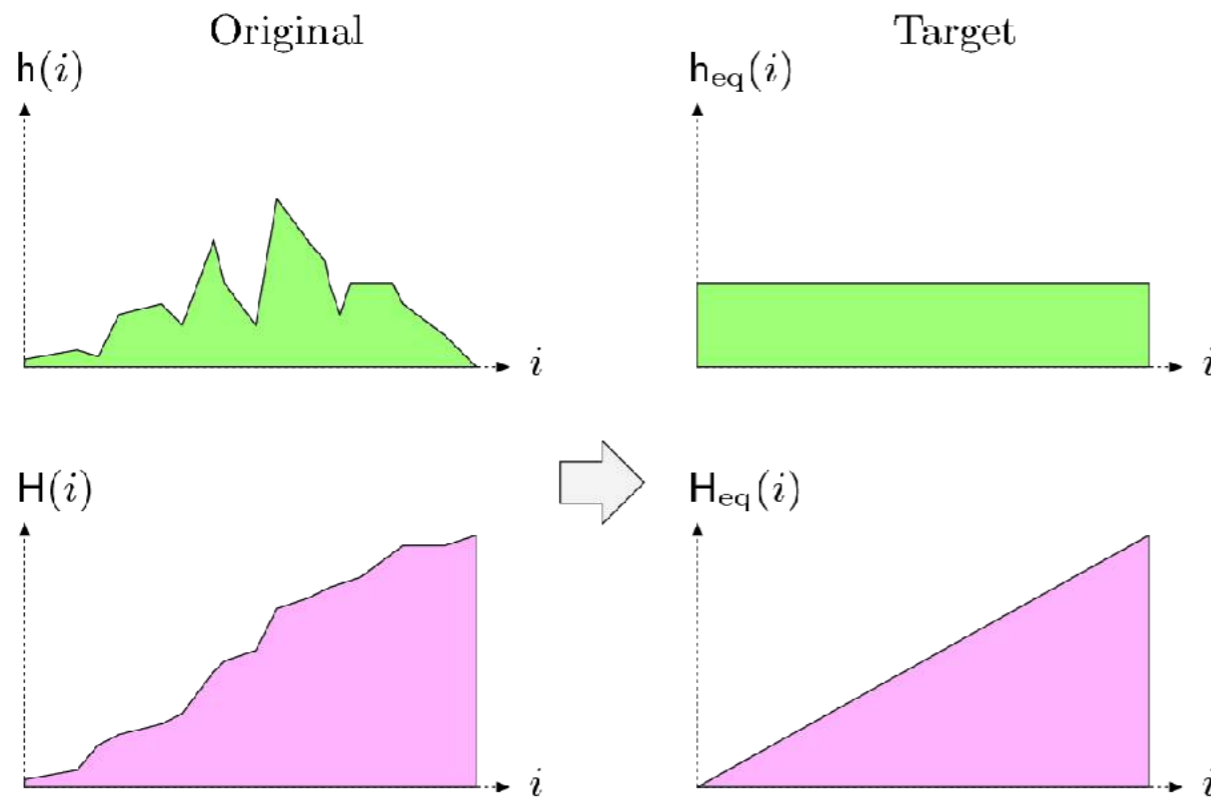


6 intensidades



$$H(i) = \sum_{j=0}^i h(j) \text{ para todos los } 0 \leq i \leq K$$

$$H(i) = \begin{cases} h(0) & \text{para } i = 0 \\ H(i-1) + h(i) & \text{para } 0 < i \leq K \end{cases}$$



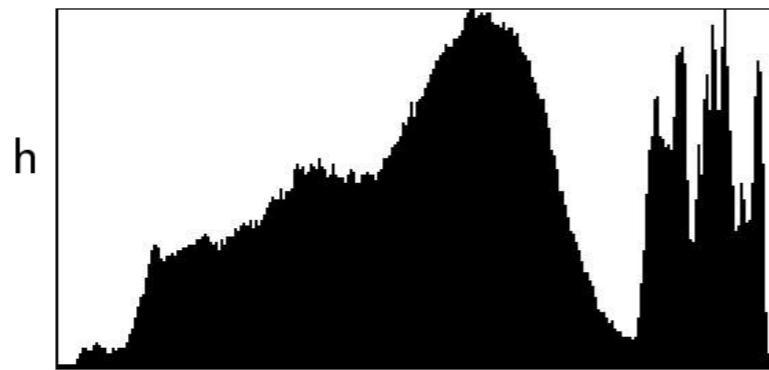
(Process > Enhance Contrast...)



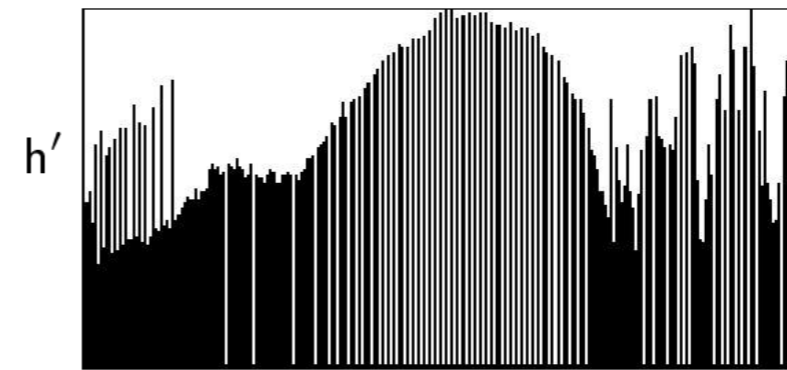
(a)



(b)



(c)



(d)

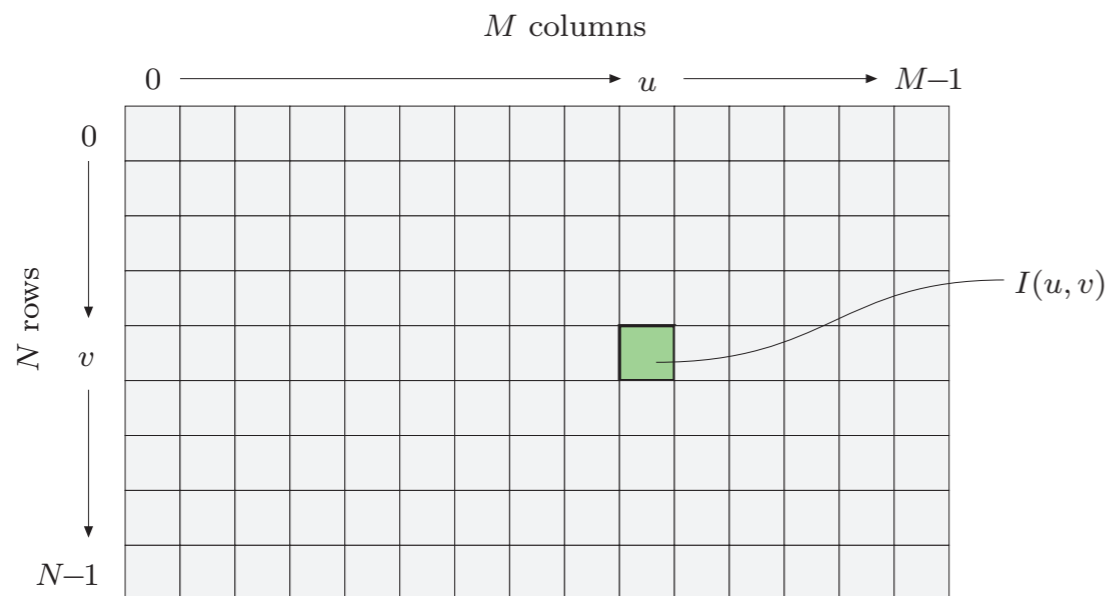
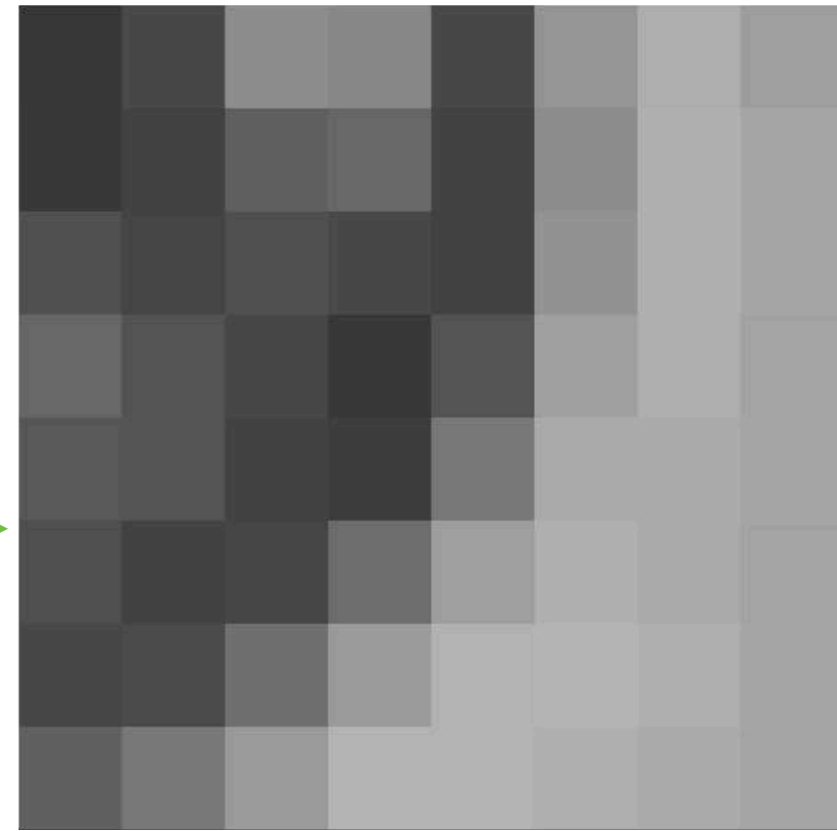


(e)



(f)

Operaciones de pixeles



57	75	143	137	75	150	175	159
59	68	98	105	69	142	179	167
83	70	82	75	68	149	178	168
104	85	74	58	86	162	176	165
89	86	69	64	122	173	173	167
81	68	71	110	160	178	172	165
74	76	114	157	180	182	177	167
99	122	156	182	181	177	172	166

Valores de los pixeles.

- Modificaciones del valor de un pixel
- No hay modificaciones de tamaño, geometría o estructura
- El nuevo valor de un pixel depende solamente del valor anterior en esa misma posición mediante una función

$$I'(u, v) = f(I(u, v))$$

- La función puede ser independiente de la posición (homogénea) o depender de la posición (no homogénea)

$$I'(u, v) = g(I(u, v), u, v)$$

- Si el resultado depende de más de una posición será el *filtrado* de la imagen.

- Operaciones globales (homogéneas)
 - modificar brillo y contraste
 - aplicar transformaciones arbitrarias mediante curvas
 - cuantificación (“posterizing”)
 - umbralización global
 - corrección de gamma
 - transformaciones de color
- Operaciones locales (no homogéneas)
 - cambio contraste o brillo local
 - umbralización (umbral adaptivo)



UNIVERSIDAD DE LA REPÚBLICA URUGUAY

Buscar cursos

ProEVA Mis cursos Curso actual + EVA Facultad Ingeniería

Activar edición Ocultar bloques Pantalla completa

Mis cursos Institutos Ingeniería Eléctrica Cursos del 2do semestre de 2018 TimagIEEE

Administración

Procesamiento de Imágenes y Visión Artificial (con un toque de Aprendizaje Automático)

Administración del curso

- Editar
- Activar
- Usuarios
- Dar de baja
- Filtros
- Información
- Configuración
- Copiar
- Reservar
- Importar
- Reiniciar
- Banear
- Papelera

Adquisición, representación y visualización imágenes. ImageJ/Fiji.

1ro. de octubre

Se puede descargar Fiji para los diferentes OS desde el sitio <http://fiji.sc/>

- Práctico: Representación y visualización de imágenes.
- Imágenes para trabajar en los prácticos.

Histogramas y operaciones de pixel

1ro. de octubre

- Práctico: Histogramas y operaciones con pixeles.

Navegación

- Página Principal
- Área personal
- Páginas del sitio
- Mis cursos
 - Institutos
 - Computación

...nfasis en ...do. Al ...cercado al ...os temas y ...presentadas

...normalmente, y para los prácticos en computadora. La parte práctica implementará ejemplos de aplicación de los normalizados presentadas con diferentes tipos de imágenes. La práctica se basará en el uso de una aplicación en software libre para el procesamiento de imágenes (FIJI, <http://fiji.sc/>). Además se presentarán ejemplos de implementación de diferentes métodos en lenguaje Python y diferentes librerías especializadas en procesamiento de imágenes y aprendizaje automático. No es necesario el manejo avanzado de Python para el seguimiento del tutorial. Para la comprensión

- Configurar el proxy de FIng para acceder a internet
 - [Edit > Options > Proxy Settings...]
 - Proxy server: httpproxy.fing.edu.uy
 - Port: 3128