

INICIACIÓN

Iniciación al uso de Macros en Fiji

confocal-cbm@listas.csic.es

www.cbm.uam.es/confocal



- ❖ Introducción
- ❖ Problema a resolver
- ❖ Herramienta Record
- ❖ Potenciar esta macro
 - ❖ Seleccionar directorios
 - ❖ Trabajar con Listas
 - ❖ Iterar listas
- ❖ Consejos e información

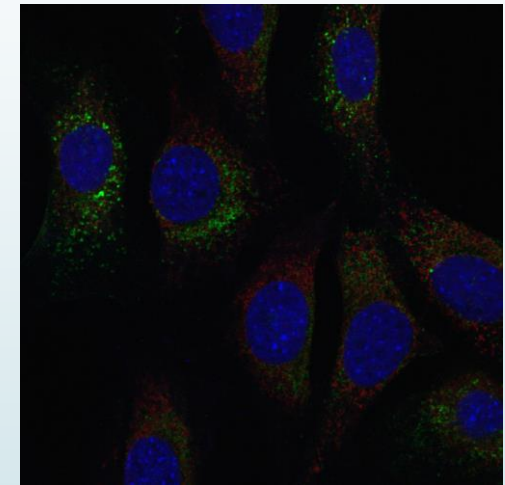
INTRODUCCION

- ❖ Una macro es un código que traduce cada acción a una línea de texto y permite repetir tareas de forma sencilla.
- ❖ Las macros os pueden ayudar a realizar trabajos largos y repetitivos en unos pocos “clicks” de forma rápida y fácil.

PROBLEMA A RESOLVER

❖ Pasos a seguir para esta macro:

1. Abrir la imagen
2. Separar los canales
3. Guardar todos los canales



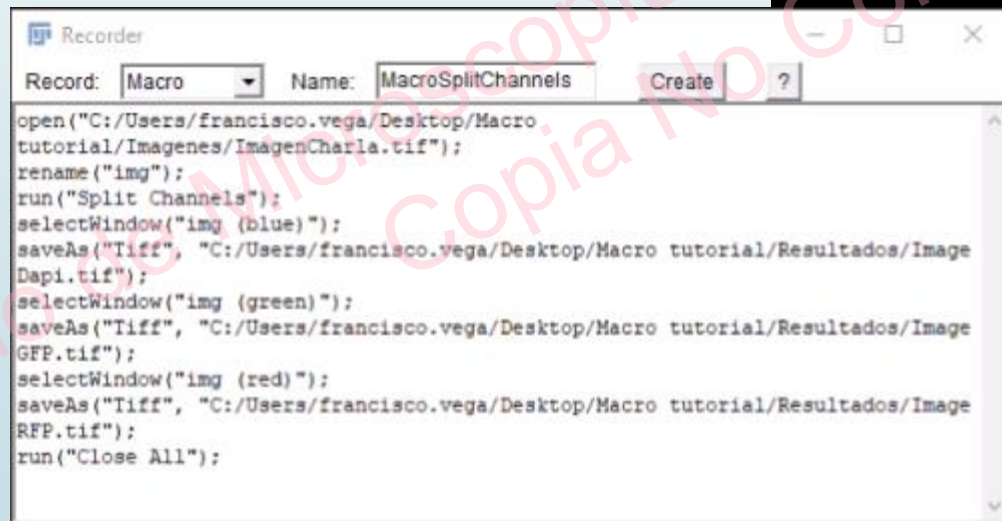
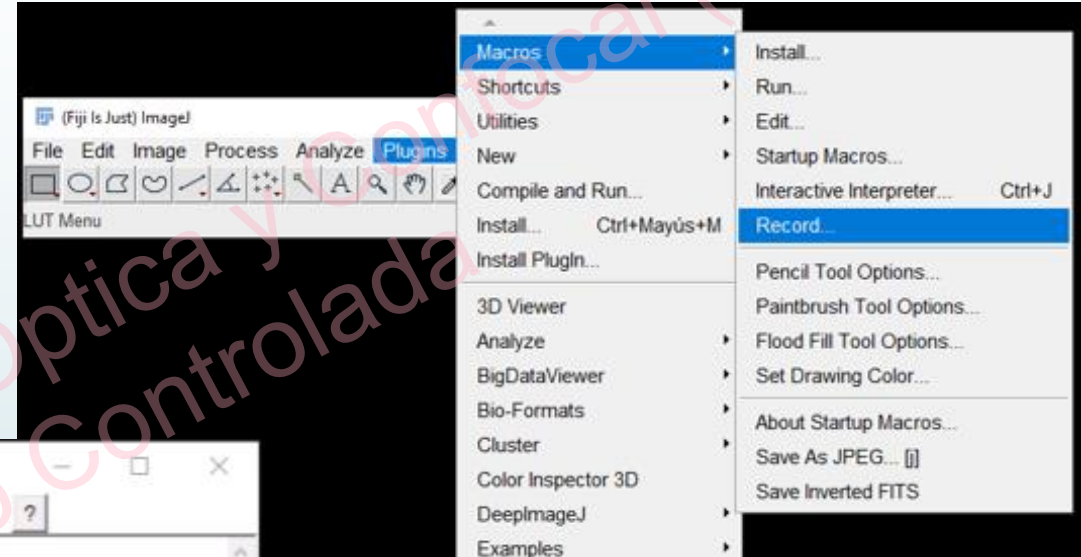


Herramienta Record

- ❖ Útil para convertir a texto todas las acciones
- ❖ Permite crear macros reutilizables fácilmente
- ❖ CUIDADO: Graba TODAS las acciones dentro de Fiji
- ❖ Lo emplearemos para definir la estructura de nuestras macros

Herramienta Record

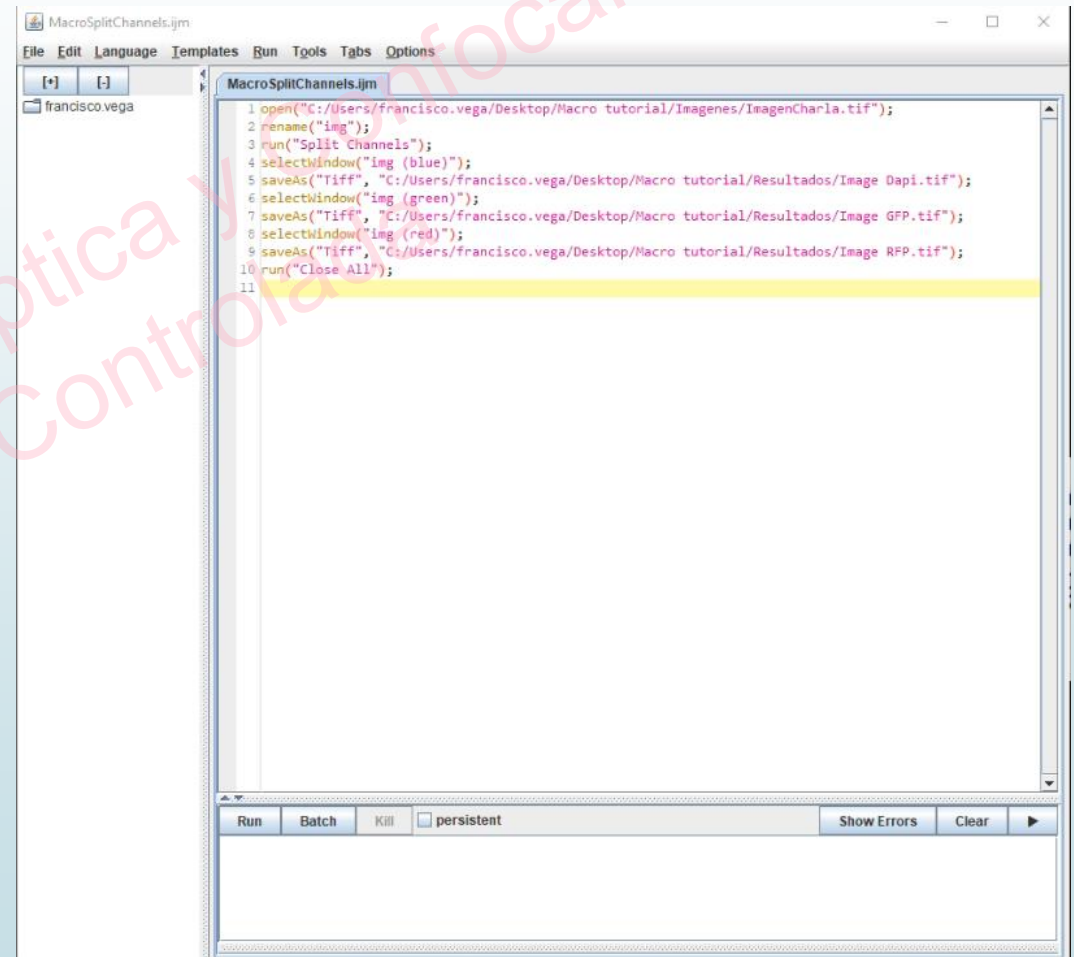
- ❖ Abrir la herramienta en la pestaña Plugins > Macros > Record...



- ❖ Tras realizar el tratamiento, el resultado de la grabación sería algo similar a esto.

Herramienta Record

- ❖ Tras dar al botón create aparecerá la siguiente pantalla y en ella comenzaremos a editar la macro.
- ❖ Es muy importante guardar la macro para que se ejecute correctamente tras los cambios que haremos.



POTENCIAR ESTA MACRO

- ❖ Seleccionar el directorio de múltiples imágenes
 - ❖ Uso de variables para almacenarlo
 - ❖ Selección de directorio
 - ❖ Listar los archivos de este
 - ❖ Recorrer la lista con los archivos
- ❖ Si algunos archivos no son imágenes ¿Qué hago?

Seleccionar un directorio

- ❖ Permite elegir el directorio donde estan las imágenes cada vez que ejecutemos la macro.
- ❖ Necesitamos una estructura donde almacenar esta dirección, esta estructura es una VARIABLE
- ❖ Las variables nos permiten guardar un dato cualquiera en su interior y utilizarlo múltiples veces



Seleccionar un directorio

❖ ¿Cómo aplicamos esto a una macro?

Usaremos la función `getDirectory()` para abrir un cuadro de selección de directorio

Todo esto lo almacenaremos en una variable llamada "origen"

Esta variable la usaremos para abrir las imágenes

```
1 origen = getDirectory("Elige el directorio de entrada de las imagenes.");  
2  
3  
4 open(origen);
```

Seleccionar un directorio

```
*MacroSplitChannels.ijm
1 origen = getDirectory("Elige el directorio de entrada de las imagenes.");
2
3
4 open(origen);
5 rename("img");
6 run("Split Channels");
7 selectWindow("img (blue)");
8 saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image Dapi.tif");
9 selectWindow("img (green)");
10 saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image GFP.tif");
11 selectWindow("img (red)");
12 saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image RFP.tif");
13 run("Close All");
14
```

PROBLEMA!! => No coge el nombre de las imágenes y solo me abre una y mal...

Listar las imágenes

❖ ¿Cómo conseguimos que abra las imágenes?

Usaremos la estructura lista, para ordenar todas las imágenes y así poder abrirlas.

❖ ¿Cómo creamos esa lista?

Usaremos la función `getFileList()` para crear una lista a partir del directorio seleccionado.

```
3 listaImagenes = getFileList(origen);
```

Listar las imágenes

```
*MacroSplitChannels.ijm
1 origen = getDirectory("Elige el directorio de entrada de las imágenes.");
2 listaImagenes = getFileList(origen);
3
4 open(origen);
5 rename("img");
6 run("Split Channels");
7 selectWindow("img (blue)");
8 saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image Dapi.tif");
9 selectWindow("img (green)");
10 saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image GFP.tif");
11 selectWindow("img (red)");
12 saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image RFP.tif");
13 run("Close All");
```

ALERTA SPOILER => No funciona haciendo `open()` de la lista

Trabajar con listas

❖ ¿Cómo recorremos esa lista?

Las listas son estructuras ordenadas con un valor ordinal (1,2,3...) y un dato.

Para acceder al dato solo tenemos que llamar a la lista e indicarle su posición en esta. EJ:

```
nombrelmg = listamagenes[1]
```

Si queremos saber la longitud de una lista hay que usar la siguiente función:

```
listamagenes.lenght
```

Para poder hacer esto necesitamos usar un bucle

Iterar en la lista

❖ ¿Cómo usamos el bucle FOR?

El bucle For necesita 3 datos para trabajar:

```
for(z=0; z<listamagenes.length; z++)
```

Dato 1: Desde que valor empiezo

Dato 2: Hasta que valor hay que iterar

Dato 3: Tamaño del salto (Habitualmente de 1 en 1)

Como habéis visto, iteramos en una variable llamada “**z**” que es la que contiene el valor actual de la iteración

Iterar en la lista

❖ Con todo esto ya podremos trabajar

Una vez puesto en orden todo esto, ya podremos ejecutar nuestra macro para múltiples imágenes.

```
*MacroSplitChannels.ijm
1 origen = getDirectory("Elige el directorio de entrada de las imagenes.");
2 listaImagenes = getFileList(origen);
3
4 for(z=0; z<listaImagenes.length; z++){
5
6     open(listaImagenes[z]);
7     rename("img");
8     run("Split Channels");
9     selectWindow("img (blue)");
10    saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image Dapi.tif");
11    selectWindow("img (green)");
12    saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image GFP.tif");
13    selectWindow("img (red)");
14    saveAs("Tiff", "C:/Users/francisco.vega/Desktop/Macro tutorial/Resultados/Image RFP.tif");
15 }
16
17 run("Close All");
```

Pero... ¿Os sale un error al guardar resultados? ¿Todos tienen el mismo nombre ?

¿Qué nos falta?

❖ ¿Dónde quiero almacenar los resultados?

Se hace igual que la selección de carpeta inicial.

```
3 destino = getDirectory("Elige el directorio donde guardar las imagenes.");
```

❖ ¿Cómo se llamará el resultado? ¿Cómo se llamaba la imagen original?

El nombre lo elegiremos nosotros. Para obtener cada nombre utilizamos la función `File.nameWithoutExtension`

```
13 name = File.nameWithoutExtension;
```

Para modificarlo y poner el nombre final, podemos añadir el nombre del canal al nombre inicial.

```
26 saveAs("Tiff", destino + name + "(green).tif");
```

Consejos

❖ Renombrar las imágenes

Para trabajar mas cómodamente se recomienda guardar el nombre (Como hicimos al grabar la macro) y renombrar la imagen con un nombre genérico. Esto permite generalizar todas las imágenes fácilmente.

```
9      open(listaImagenes[z]);  
10     name = File.nameWithoutExtension;  
11  
12     rename("img");  
13  
14     run("Split Channels");  
15     selectWindow("img (blue)");
```

Al renombrarlas, es mas sencillo trabajar con varias imágenes de forma objetiva y luego recuperar su nombre original para almacenar los resultados.

Consejos

❖ Donde obtener mas info.

En la pestaña Help de Fiji tenéis la documentación necesaria para consultar estructuras y conceptos (Documentation...) como funciones ya implementadas (Dev. Resources).

