|  |
| --- |
|  |
| Conteo de células neuronales |
| Proyecto Final – Procesamiento de Imágenes I |

|  |
| --- |
| Joaquín Bengochea - Camila Robles  Camila Robles |

## Introducción

El presente informe es parte del proyecto final de la materia Procesamiento de Imágenes I. En el mismo se presentarán detalles acerca del problema inicial y la forma en que fue desarrollada su solución, como también los inconvenientes que surgieron a la hora de resolverlo. A su vez se mostrarán ejemplos representativos de los distintos casos que se debieron tener en cuenta para lograrlo.

Se utilizó Fiji como herramienta principal para implementar la solución. Es un paquete de procesamiento de imágenes que fue creado como una extensión de ImageJ con plugins orientados al uso científico. Hay una amplia variedad de opciones para utilizarlo como medio de desarrollo que incluye el uso de macros, scripts o la implementación de plugins en Java. Se optó por ésta última ya que permitía un mayor acceso a los datos obtenidos a partir de las imágenes analizadas y mejores resultados en comparación a las demás.

El objetivo del proyecto era la creación de un plugin para Fiji que realice el conteo y la clasificación de células neuronales fluorescentes a partir de imágenes capturadas microscópicamente. Para lograrlo fue necesario tener un buen entendimiento de la problemática y los conceptos que esta incluye, los cuales se detallarán a continuación.

## Descripción del problema

El problema consistía en el conteo y la clasificación de células neuronales fluorescentes a partir de una imagen en formato czi dividida en cuatro canales, dos de los cuales fueron utilizados para obtener la información necesaria. Uno de los canales muestra los núcleos de las neuronas, diferenciándolos del citoplasma; y el otro exhibe la fluorescencia que representa la reacción de las células ante cierta proteína.

Para poder clasificarlas como positivas o negativas es esencial conocer ciertos conceptos: una célula es positiva cuando la fluorescencia coincide posicionalmente en ambos canales. Esto significa que el núcleo de la neurona en cierta posición ha tenido una reacción positiva a la proteína. Al mismo tiempo, para poder clasificarlas como positivas es necesario tener en cuenta parámetros como la intensidad en el canal de la proteína, índice de circularidad y porcentaje de recubrimiento del núcleo.

En las siguientes imágenes se pueden ver los casos particulares, algunos de los cuales requieren especial atención.

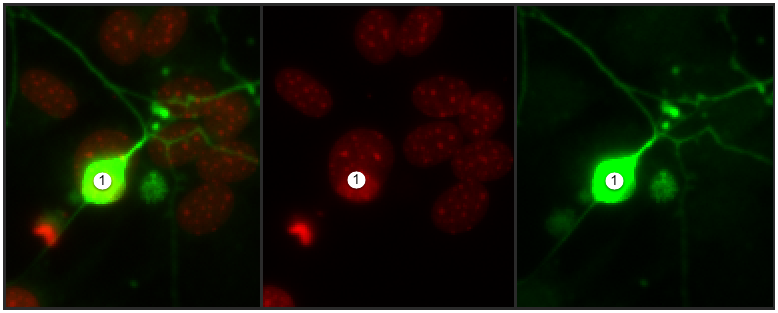


Ilustración núcleo positivo

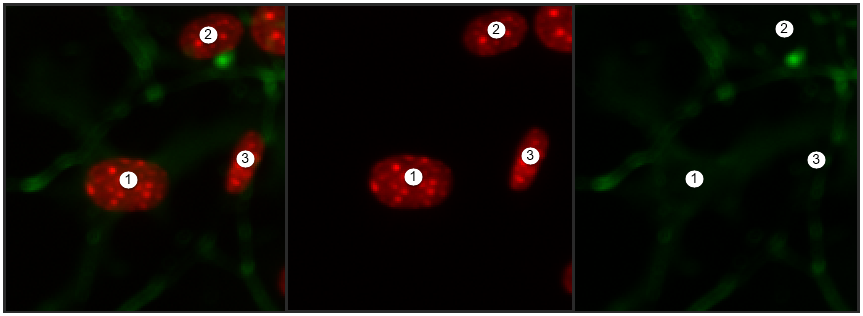


Ilustración núcleo negativo

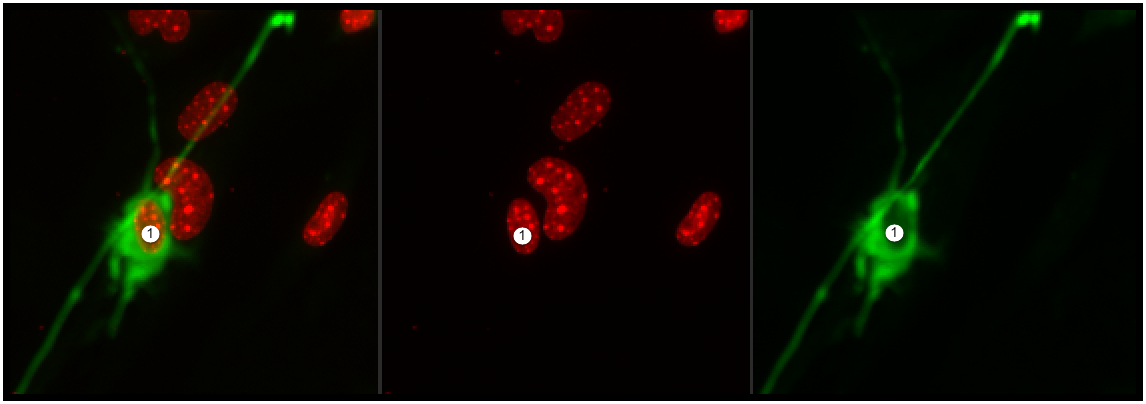


Ilustración citoplasma positivo - núcleo negativo

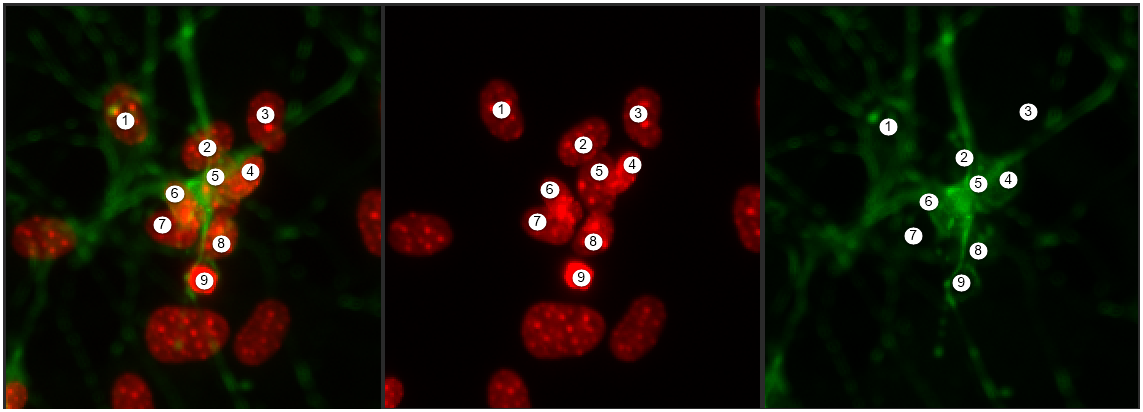


Ilustración núcleos agrupados negativos

## Detalles de la solución

Para implementar la solución se debieron tener en cuenta los distintos casos y la correcta configuración de los parámetros mencionados anteriormente.

En principio, se utilizó el método *run* de IJ con los parámetros necesarios para que se ejecuten en Fiji las funcionalidades que llevarían a cabo el procesamiento de las imágenes. La desventaja era que no se podía trabajar con las imágenes a un bajo nivel, aparejando ciertos problemas de performance. Para solucionarlo se modificó el código de manera de trabajar con los objetos directamente.

Se comienza con el preprocesamiento de las dos imágenes pertenecientes a los canales a utilizar: el de los núcleos y el de DAPI (proteína). Para hacerlo se obtienen los valores de los parámetros a partir de un diálogo que permite al usuario configurar el plugin según sus necesidades particulares. Estos valores incluyen: nuclei threshold min&max, nucleic circularity min, nucleic size min&max y protein size min&max. Por otro lado, los parámetros enhance contrast saturation, substract background rolling y protein threshold min&max tienen valores preestablecidos.

Una vez obtenidos los valores, se procesan las imágenes: por un lado la del canal que contiene los núcleos y por el otro la de la proteína. En principio, ambas son pasadas a 8 bits y procesadas según los parámetros mencionados anteriormente. Por su parte, a la primera imagen se le aplica además un watershed para distinguir núcleos que se encuentran agrupados, y se crea una máscara binaria a partir de ella que representa la morfología de cada núcleo.

A partir de estas imágenes se obtienen los resultados deseados con una de las funcionalidades incluidas en Fiji: *Analyze Particles*. Estos resultados se almacenan en dos tablas de resultados (objetos de la clase *ResultsTable*), que incluyen las siguientes columnas a partir de ambos canales: área, posición del centro de masa (X, Y) e índice de circularidad.

Una vez que el preprocesamiento es terminado y se obtienen las tablas, se procede con el conteo de las células cuyo núcleo es positivo en el canal de la proteína. El método *contarpositivos* se encarga de realizar el conteo mediante un algoritmo cuyo pseudocódigo se presenta a continuación:

contarpositivos (imgDAPI)

CantidadPositivos = 0;

por cada fila de la tabla de resultados //obtener regiones de interés de los núcleos

{

obtener el área, ancho y alto;

obtener valor *BX y BY;*

CantidadMayorAThreshold = 0;

por cada (k = x; k < x + ancho)

por cada (l = y; l < y + alto)

valor = imgDAPI.obtenerPixel(k,l);

si (valor > ProteinThreshold)

CantidadMayorAThreshold++;

si (CantidadMayorAThreshold / área > 0.5)

CantidadPositivos++;

}

devolver CantidadPositivos;

Si la opción *Protein Counting* fue seleccionada, se procede a crear la imagen *outlines* que presenta los núcleos con resultado positivo. A continuación se muestra el pseudocódigo del algoritmo que lo permite:

countGreenMatches(ImagePlus imp) {

IJ.run("Set Measurements...", "area centroid center shape decimal=3");

IJ.run(imp, "Analyze Particles...", "size="+proteinSizeMin+"-"+proteinSizeMax+" circularity=0.00-1.00 show=Outlines display clear record in\_situ");

verdes = (ResultsTable) ResultsTable.getResultsTable().clone();

// Return imp modified

nucleosX = nucleos.getColumnAsDoubles(ResultsTable.X\_CENTER\_OF\_MASS);

nucleosY = nucleos.getColumnAsDoubles(ResultsTable.Y\_CENTER\_OF\_MASS);

areas = nucleos.getColumnAsDoubles(ResultsTable.AREA);

CIs = nucleos.getColumnAsDoubles(ResultsTable.ROUNDNESS);

verdesX = verdes.getColumnAsDoubles(ResultsTable.X\_CENTER\_OF\_MASS);

verdesY = verdes.getColumnAsDoubles(ResultsTable.Y\_CENTER\_OF\_MASS);

Vector<Vector<Integer>> result = new Vector<Vector<Integer>>();

for (int i=0; i<verdesX.length; i++){

boolean found = false;

int j=0;

while(j<nucleosX.length && !found){

if(Math.abs(nucleosX[j]-verdesX[i])<2\*nucleicSizeMax && Math.abs(nucleosY[j]-verdesY[i])<2\*nucleicSizeMax){

double radio = areas[j]/Math.PI;

double dist = Math.pow(nucleosX[j]-verdesX[i], 2) + Math.pow(nucleosY[j]-verdesY[i], 2);

if(dist<= radio+ 2\*radio\*CIs[j]+CIs[j]\*CIs[j]){

Vector<Integer> v = new Vector<Integer>();

v.add((int) nucleosX[j]);

v.add((int) nucleosY[j]);

result.add(v);

found = true;

}

}

j++;

}

if(!found) {

ResultsTable.getResultsTable().deleteRow(i);

ResultsTable.getResultsTable().updateResults();

}

}

String resultXML = createXMLFromPoints(result);

IJ.saveString(resultXML, resultPath);

ResultsTable.getResultsTable().show("Results");

imp.show();

return imp;

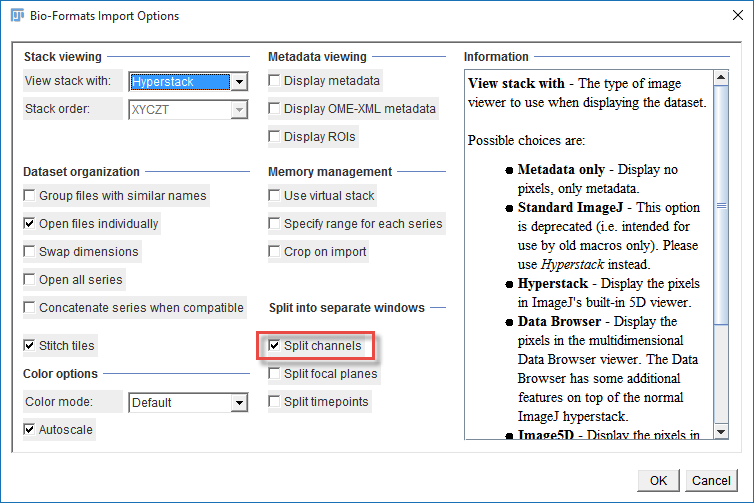
}

## Instalación

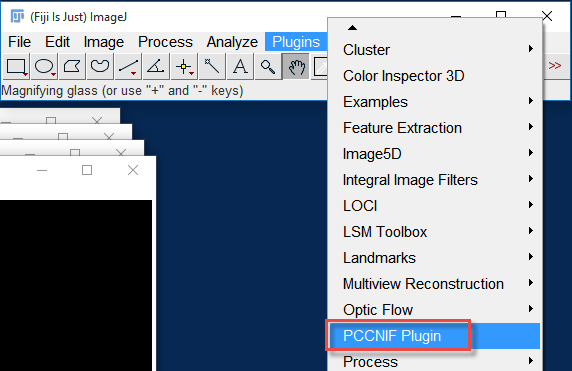
1. Cerrar Fiji;
2. Copiar el archivo "PCCNIF\_Plugin.**jar**" a la subcarpeta "Plugins" de la carpeta donde está funcionando el Fiji (por ejemplo: C:\Fiji.app\plugins);
3. Abrir Fiji.

## Instrucciones de uso

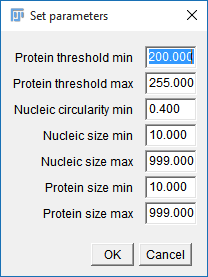
1. Abrir una imagen con formato CZI;
2. En Bio-Formats Import Options, tildar la opción "Split channels";



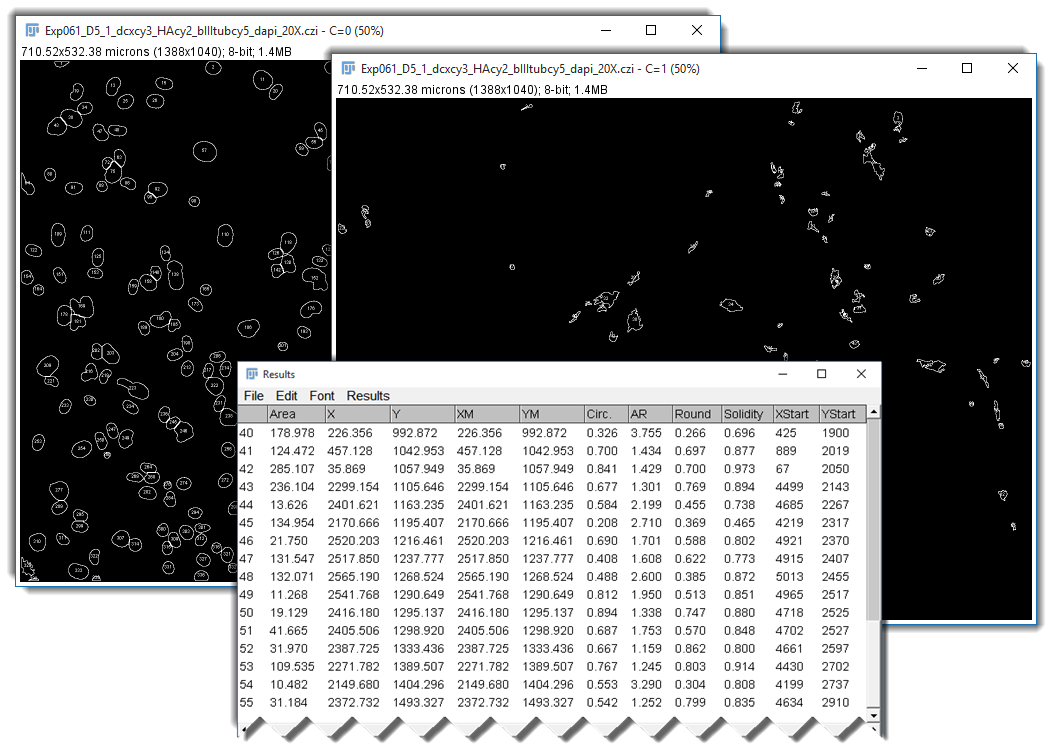
1. Ejecutar el plugin "PCCNIF\_Plugin" desde el menú "Plugins";



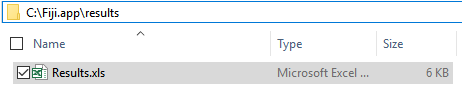
1. Establecer parámetros y click en el botón "OK".



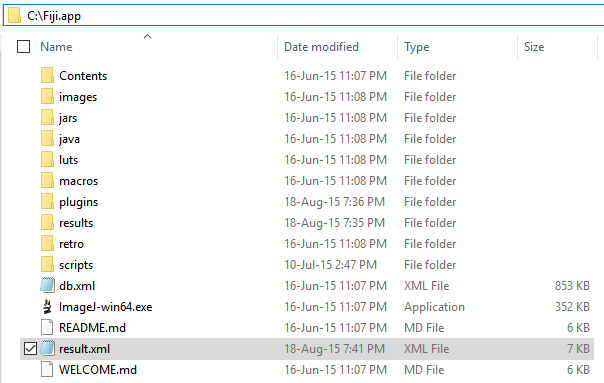
* El plugin se ejecutará y terminará con dos imágenes abiertas correspondientes a los canales en los que se ven los núcleos y la proteína con los contornos marcados, además de la tabla de resultados con los núcleos marcados como positivos para la proteína.



* El plugin grabará en la carpeta de instalación del Fiji (por ejemplo: C:\Fiji.app) un archivo denominado "result.xml " y una carpeta denominada "Results" con un archivo dentro llamado "results.xls".



* El archivo "result.xml" mencionado en el punto anterior se utiliza abriendo la imagen original (sin tildar la opción split channels), ejecutando el plugin "Analyze->Cell Counter", haciendo click en botón "Load Markers" y cargando el archivo "result.xml". En la imagen se visualizarán los núcleos marcados como positivos.



* El archivo "results.xls" es un Excel que contiene los detalles de coordenadas, área, etc. para cada una de los núcleos marcados como positivos.

