

TRABAJO FIN DE GRADO

Métodos de software libre para el análisis de imágenes biológicas



Julián Román Camacho

Tutor: Luis María Escudero Cuadrado

Facultad de Biología

Departamento de Biología Celular

Julio de 2023

Índice

0.	Resumen	1
1.	Introducción	1
1.1.	Historia y evolución del análisis de imágenes biológicas	1
1.2.	Concepto del análisis de imágenes digitales en biología	2
1.3.	Concepto de software para el análisis de imágenes	2
2.	Análisis de imágenes en biología	3
2.1.	Características del análisis de imágenes digitales	3
2.2.	Usos del análisis de imágenes en biología	6
2.3.	Pasos para el análisis de una imagen biológica	8
2.3.1.	Adquisición	8
2.3.2.	Procesamiento	8
2.3.3.	Segmentación	9
2.3.4.	Cuantificación	10
3.	Software libre para el análisis de imágenes biológicas	10
3.1.	<i>Fiji</i>	10
3.2.	Funcionamiento de <i>Fiji</i>	12
3.2.1.	Flujo de trabajo de un análisis en <i>Fiji</i>	16
3.3.	<i>CellProfiler</i>	17
3.4.	Funcionamiento de <i>CellProfiler</i>	17
3.4.1.	Flujo de trabajo de una <i>pipeline</i> en <i>CellProfiler</i>	20
3.5.	Comparativa entre <i>Fiji</i> y <i>CellProfiler</i>	21
4.	Ventajas y desventajas del software de uso libre respecto al software de propietarios ...	25
5.	Conclusión.....	27
6.	Referencias bibliográficas	28

MÉTODOS DE SOFTWARE LIBRE PARA EL ANÁLISIS DE IMÁGENES BIOLÓGICAS

0. Resumen

Hoy en día, los métodos computacionales son indispensables para el análisis de imágenes biológicas. Gracias a ellos, es posible estudiar la información derivada de las imágenes de microscopía de muestras biológicas de una forma objetiva. En las últimas décadas, han surgido diferentes tecnologías que enriquecen y fomentan el desarrollo de este campo de investigación. En esta memoria, se presentarán algunos de los procesos clave que conforman el análisis de imagen en biología, sus características y su finalidad, centrándose en dos de los programas de libre acceso con mayor prestigio, *Fiji* y *CellProfiler*. Así, este trabajo podrá contribuir a obtener un mayor conocimiento tanto del funcionamiento general del análisis de imagen, como del manejo y la aplicabilidad que presentan ambos programas, justificando así el reconocimiento que han obtenido este tipo de programas por parte de la comunidad científica.

1. Introducción

1.1. Historia y evolución del análisis de imágenes biológicas

Desde tiempos inmemoriales, el ser humano ha sentido la necesidad de explorar y descubrir cómo funciona su alrededor, es decir, cómo funciona la vida. El entorno visto a simple vista fue descrito para tratar de entender el complejo sistema donde vivimos, pero el ojo humano no tenía la capacidad suficiente para poder visualizar todo lo que le rodeaba. Debido a esto, surgieron diferentes instrumentos de observación, tanto a larga distancia como el telescopio y a corta distancia como el microscopio, convirtiéndose este último en una herramienta fundamental desde el siglo XVII para el descubrimiento y la comprensión de la biología, por su virtud para observar estructuras y organismos a tamaños muy por debajo de los 0,2 mm de resolución del ojo humano ([Hogg, 1883](#); [Swedlow et al., 2009](#)). Como menciona en su artículo [Lane \(2015\)](#), Antonie van Leeuwenhoek es considerado uno de los primeros pioneros en investigación microscópica, ya que con su microscopio simple de una lente determinó manualmente muchos microorganismos, como bacterias y protozoos nunca antes vistos ([Hogg, 1883](#); [Lane, 2015](#)). Las técnicas de óptica siguieron evolucionando dando lugar a nuevos tipos de microscopios como el compuesto, el electrónico, el de fluorescencia o el confocal, sin embargo, la capacidad de los científicos de interpretar y analizar el volumen y la complejidad

de los datos era limitada. Por esta razón, alrededor de la década de los 60-70 surgen los primeros programas específicos para analizar imágenes ([Gonzalez & Woods, 2018](#); [Mansfield & Pykett, 1978](#)), unas herramientas digitales que resultaron en una mejora de la automatización, velocidad y precisión de los procesos biológicos estudiados ([Eliceiri et al., 2012](#)).

1.2. Concepto del análisis de imágenes digitales en biología

El análisis de imágenes digitales tiene un significado amplio, por lo que se ha adaptado la definición de [González & Woods \(2018\)](#), esta expone que es un proceso de descubrimiento, identificación y comprensión de patrones que son relevantes para obtener datos para un estudio a partir de imágenes. El objetivo de este análisis en biología es extraer información cuantitativa y cualitativa de las imágenes obtenidas, para conocer mejor los procesos biológicos y conseguir descubrimientos científicos significativos ([Campa Fernández, 2016](#); [González & Woods, 2018](#)).

1.3. Concepto de software para el análisis de imágenes

Un software no es más que un conjunto de programas de ordenador que tiene asociado una información que contrasta para realizar un trabajo, en este caso, sirve para el análisis de imágenes ([Real Academia Española, s.f.](#)). Normalmente, existen dos tipos de software según el tipo de licencia, que es un acuerdo legal entre el titular de los derechos de autor del software y el usuario, que contiene información sobre los permisos y condiciones de uso del programa. Hay softwares de propietarios o privados y softwares de uso libre. Los softwares de propietarios, son desarrollados con el fin de obtener ganancias y poseen un código fuente protegido para evitar su distribución de manera ilegal. Por otro lado, los softwares de uso libre, son aquellos que se distribuyen de manera gratuita para los usuarios en la red puesto que su código no tiene restricciones ([Eliceiri et al., 2012](#); [Solomon, 2009](#)). Las libertades básicas para considerar un software libre son: la capacidad de ejecutar, copiar, distribuir y modificar dicho software ([Solomon, 2009](#)). Estos programas se basan en el uso de *scripts* y *macros*, los *scripts* son los diferentes lenguajes de programación que existen como *Java*, *Python* y *R*. Los *macros* por otra parte, son la secuencia de comandos que se escriben para automatizar tareas repetitivas ([Dobson et al., 2021](#)). Muchos de los programas de uso libre tienen un añadido que es el uso de complementos o *plugins*, son una serie de modificaciones al programa para ampliar su función sin necesidad de modificar el código original del programa ([Schindelin et al., 2012](#)).

2. Análisis de imágenes en biología

Aunque este trabajo se centra en los softwares específicos para analizar imágenes, se abordarán brevemente algunas características importantes del análisis de imagen, la utilidad que tienen hoy en día y se explicarán cuatro pasos generales para analizar una imagen digital.

Antes que nada, es importante conocer el término imagen digital tal y como lo definen [González & Woods \(2018\)](#). Una imagen digital es una representación visual de una función bidimensional, $f(x,y)$, donde x e y son coordenadas espaciales y la amplitud de f son los distintos valores de intensidad de la imagen. Estos valores deben ser finitos y discretos. ([González & Woods, 2018](#)).

2.1. Características del análisis de imágenes digitales

Formato de la imagen

El formato es el tipo de archivo digital que se utiliza para guardar una imagen, los formatos más recomendables en ciencia son aquellos que tienen compresión sin pérdida, esto significa que el tamaño del archivo no se reduce y los datos almacenados permanecen intactos, algunos formatos usuales son TIFF, PNG y BMP, el único inconveniente es que ocupan más espacio en el ordenador. Por otro lado, el formato compresión con pérdida, como los archivos JPEG o GIF, descartan información de la imagen para reducir su tamaño, esto puede resultar perjudicial debido a que se podría perder alguna información valiosa ([Campa Fernández, 2016](#); [Roeder et al., 2012](#)).

Píxeles y vóxeles

Las imágenes digitales están constituidas principalmente por píxeles, un píxel, según la definición adaptada de [Roeder et al. \(2012\)](#), es la menor unidad homogénea en dos dimensiones (2D) que forma parte de una imagen digital, mientras que un vóxel, es la unidad tridimensional (3D) de la que se compone un volumen digital. Ambos tienen una posición definida en el espacio y valores de intensidad de color asociados ([Roeder et al., 2012](#)).

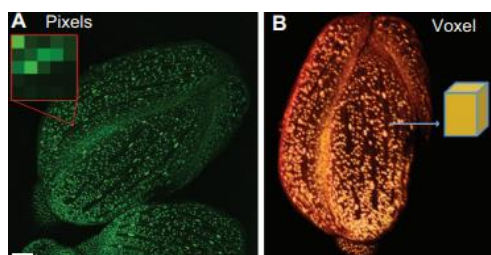


Figura 1: en la imagen A se muestran los píxeles de una imagen del capullo de una flor, mientras que en la imagen B, es una representación 3D de la imagen A, donde se muestra un esquema de un vóxel. **Fuente:** adaptada de [Roeder et al. \(2012\)](#).

Stack

Un *stack* es un conjunto de imágenes o fragmentos en dos dimensiones, normalmente ancho (x) y alto (y) de una única muestra biológica que se unen para formar una imagen tridimensional, añadiendo una nueva dimensión que puede ser la profundidad (z), el tiempo (t) o el color (ch) (Roeder et al., 2012). Esta construcción tridimensional permite mejorar la precisión del análisis visualizando características que en dos dimensiones no se aprecian. El *stack* permite trabajar individualmente con los diferentes planos en el espacio en un intervalo preciso con un valor z de profundidad asignado a cada imagen (Portiansky, 2018).

Tipos de imágenes y profundidad de bits

Existen tanto imágenes en escala de grises como a color. En la escala de grises, cada píxel representa un valor numérico asociado a la intensidad del color gris correspondiente. Frecuentemente, se utiliza el sistema de 8 bits, que sigue la regla matemática 2^n , con 8 bits hay $2^8=256$ intensidades de gris que se traducen en el valor 0 negro (sin intensidad), el valor 255 blanco (máxima intensidad) y los valores intermedios diferentes tonalidades de grises. Mientras más bits tenga la imagen se podrá distinguir una mejor definición de los colores, a esto se llama profundidad de bits (Jiang et al., 2015; Portiansky, 2018; Roeder et al., 2012).

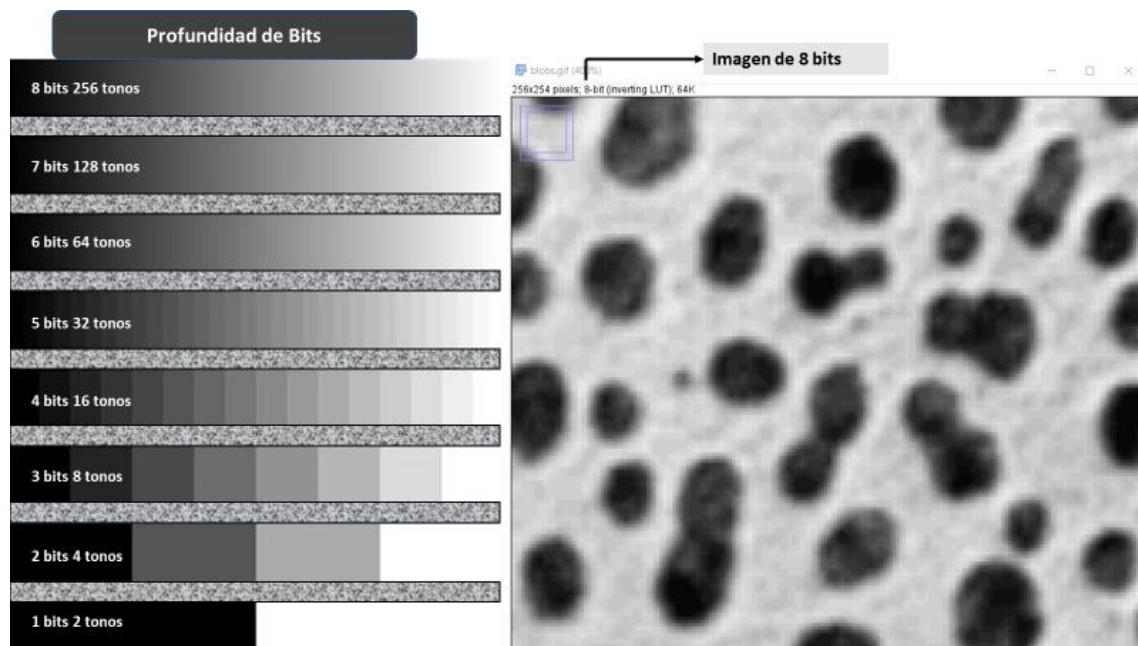


Figura 2: a la izquierda se muestra un esquema que muestra las diferencias de intensidad según la profundidad de bits y a la derecha una imagen de 8 bits donde se aprecian diferentes intensidades de gris. **Fuente:** elaboración propia a partir de [Fiji](#).

En el manejo de imágenes a color predomina el sistema RGB (*Red, Green, Blue*), que permite obtener cualquier color por la combinación de los tres colores primarios rojo, verde y

azul. Los canales son las diferentes capas de información que conforman una imagen. Mientras que en la escala de grises solo hay un canal que representa la intensidad de luz, las imágenes RGB utilizan tres canales asociados a un color primario cada uno y utilizan una codificación de al menos 24 bits, lo que significa que se asignan 8 bits a cada canal y esto se traduce en 256 tonalidades de colores diferentes por la mezcla de los tres primarios (**Figura 3-A**) ([Portiansky, 2018](#)). Al separar matemáticamente extrayendo el valor expresado en cada uno de los píxeles se obtienen tres imágenes monocromáticas correspondientes a cada canal (**Figura 3-B**). El sistema RGB es muy utilizado en imágenes multicanales, para poder etiquetar cada canal y estudiarlo por separado o tenerlo organizado y visualizarlo todo en una misma imagen, además el propio color sirve para destacar y diferenciar zonas de interés ([Tsai et al., 2011](#)). También existen otros sistemas de color como el CYMK (*cyan, yellow, magenta, key*) o el HSL (*hue, saturation, lightness*) ([Jiang et al., 2015](#); [Campa Fernández, 2016](#)).

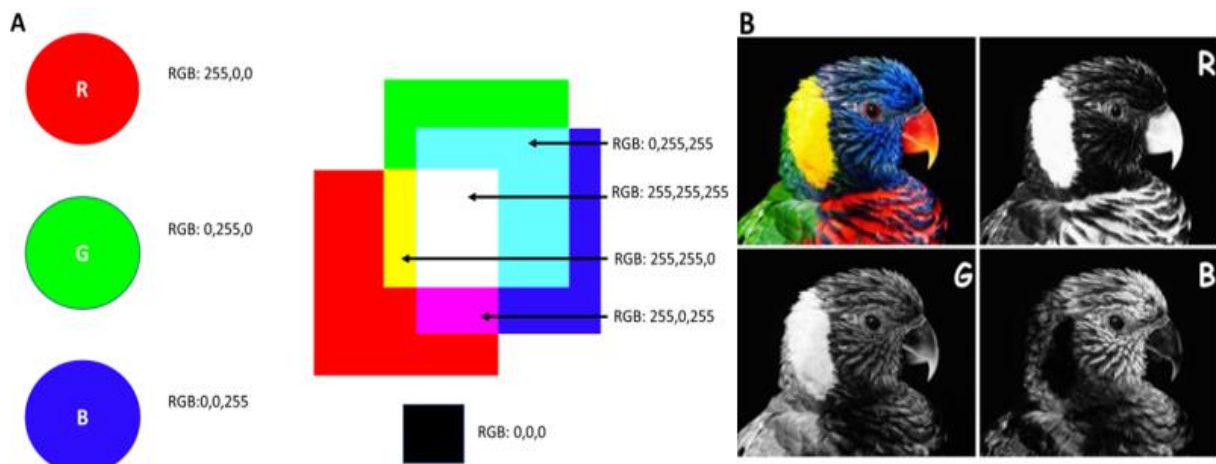


Figura 3: la imagen A es una representación del color según el sistema RGB de 8 bits. La imagen B muestra una separación de cada uno de los canales RGB. R: rojo; G: verde; B: azul. **Fuente:** imagen 3-A elaboración propia. Imagen 3-B adaptada de [Portiansky \(2018\)](#).

Resolución

La resolución en el área del análisis digital se refiere a la cantidad de píxeles por unidad de longitud, es decir, la relación entre el tamaño de píxeles de una imagen y sus dimensiones físicas. Mientras más cantidad de píxeles tenga una imagen mayor será su resolución y por ende estará más definida. La resolución puede determinar la distancia mínima en la que se pueden diferenciar dos puntos, fundamental a la hora de identificar y delimitar objetos de interés en una muestra ([Portiansky, 2018](#)).

2.2. Usos del análisis de imágenes en biología

En ciencia se cuenta con una amplia variedad de programas de análisis de imagen que brindan soluciones especializadas a una gran diversidad de disciplinas. En este caso, el enfoque se centrará en las aplicaciones generales en biología.

Los programas de análisis de imagen normalmente incluyen una serie de herramientas de edición y procesamiento de imágenes, que serán explicadas en el siguiente apartado. Pero resumidamente, el propósito de ambos tipos de métodos es mejorar la calidad de la imagen y/o facilitar el estudio de acorde a lo que se quiere ([Campa Fernández, 2016](#); [Roeder et al., 2012](#)).

En biología, estos análisis sirven para estudiar muchos elementos y estructuras que van según su complejidad desde moléculas como proteínas y ácidos nucleicos hasta células, tejidos, órganos y organismos. Se pueden realizar análisis cuantitativos individuales o a gran escala, en los que se estudian diversos fenómenos como la localización, forma, tamaño, patrones, densidad, complejidad, organización, conteo, intensidad de la señal ([Peng, 2008](#)), es decir, una gran cantidad de procesos que se basan en comprender la morfología, detectar cambios en esta y ver cómo afecta a un proceso ([Carpenter et al., 2012](#); [González Bardeci et al., 2017](#)). Además, permiten estudiar los procesos biológicos celulares en diferentes contextos experimentales ya sea *in vivo*, *in vitro*, *in situ* o *ex vivo*. Uno de los objetivos principales de estos análisis es examinar cualquier característica o rasgo observable de un organismo para determinar la función de los genes, señalar vías celulares y proporcionar información detallada de las características celulares para ver su estado, y si presentan alguna enfermedad o se puede mejorar su función desarrollando así tratamientos más efectivos ([Peng, 2008](#); [Swedlow et al., 2009](#)). Saber cómo interaccionan los distintos elementos biológicos entre sí, es una parte crucial del estudio, por ejemplo, células y moléculas entre sí, o proteínas con otras proteínas, esto muestra cómo funcionan las relaciones internas de los organismos y cómo se coordinan sus funciones ([Eliceiri et al., 2012](#)). Las células, tejidos y órganos de un organismo están en continuo movimiento y cambio, a esto se le conoce como dinámica celular y es muy útil para estudiar procesos vitales como la embriogénesis, la respuesta inmune, la migración, la proliferación celular o la regeneración de tejidos entre otros muchos procesos ([González Bardeci et al., 2017](#); [Peng, 2008](#)). Un gran número de estos procedimientos dinámicos se pueden seguir en tiempo real mediante diferentes técnicas de microscopía, como el uso de marcadores con sondas fluorescentes específicas en nuestra región de interés para hacer un seguimiento de la vida de una célula, viendo como interacciona con el medio que le rodea, su crecimiento, su división y

su muerte ([González Bardeci et al., 2017](#)). Las imágenes normalmente están en una superficie plana (2D), pero para un mejor entendimiento estas pueden pasar a tener una dimensión tridimensional (3D) a la que se le añade profundidad, una dimensión temporal (4D) para ver los cambios en el tiempo, e incluso ampliar a una quinta dimensión (5D) que representa información adicional, como propiedades espectrales, características morfológicas, información contextual o información multimodal ([Peng et al., 2014](#)). La comprensión de las estructuras biológicas mejora mucho al tener más dimensiones y permite realizar análisis más sofisticados y precisos de patrones espaciales, que, en dos dimensiones no se pueden apreciar. Por otra parte, estos programas son fundamentales para el estudio y creación de atlas de organismos modelos, donde se encuentra una gran cantidad de información biológica de un organismo, un ejemplo de esto es el *Allen Brain Atlas*, que se centra en la organización y actividad del cerebro de humano y del ratón de laboratorio (*Mus musculus*) ([Lein et al., 2007](#)). En cuanto a los procesos de cuantificación y medición, estos utilizan diferentes algoritmos estadísticos y matemáticos incluidos en los programas para obtener información precisa y detallada de las muestras biológicas, que, sumado a herramientas de representación gráfica, permiten visualizar los datos adquiridos de diversas formas como histogramas, gráficos de barras y de líneas ([Schroeder et al., 2021](#)). Los datos se guardan para su posterior interpretación, comparación y resolución. Para finalizar, es conveniente destacar una disciplina recientemente emergente en este ámbito llamada *machine learning*, es un enfoque de la inteligencia artificial en el que se desarrollan algoritmos y modelos que pueden aprender y tomar decisiones usando datos y experiencias pasadas de otros análisis sin estar explícitamente programados. Ya se ha aplicado exitosamente a muchas tareas como la clasificación, segmentación, detección, reconocimiento de patrones y se sigue desarrollando para poder llevar a cabo análisis más complejos. El *deep learning* va un paso más allá, es un subcampo del *machine learning* basado en crear redes neuronales, similares a las de un cerebro humano, con capas de nodos interconectadas denominadas como neuronas artificiales para extraer características directamente de los datos, resultando en un rendimiento y capacidad aprendizaje mucho mayor que con los métodos de programación tradicionales ([Rabbi et al., 2022](#)). Toda esta plétora de opciones y muchas más, complementan la investigación de los biólogos ([Carpenter et al., 2012](#); [Peng, 2008](#); [Roeder et al., 2012](#); [Schneider et al., 2012](#); [Schroeder et al., 2021](#); [Swedlow et al., 2009](#)).

2.3. Pasos para el análisis de una imagen biológica

El análisis general de una imagen biológica se divide en cuatro pasos diferentes desde que está la muestra preparada en el microscopio. Los pasos son: la adquisición, el procesamiento, la segmentación y la cuantificación.

2.3.1. Adquisición

El primer paso es tomar la imagen, esto se lleva a cabo por medio de un microscopio o mediante una cámara digital. Este paso se tiene que hacer con sumo cuidado, puesto que una toma de calidad de los parámetros facilitará el trabajo posterior. Si se toma la imagen de forma imprecisa puede introducir errores y distorsiones que afecten a la calidad de imagen y por tanto a las zonas de interés ([Roeder et al., 2012](#)).

2.3.2. Procesamiento

El procesamiento consiste en una serie de técnicas que implican modificar los valores de intensidad de los píxeles para incrementar el contraste entre los objetos de interés que serán cuantificados o medidos del resto de objetos conocidos normalmente como el fondo, con la finalidad global de mejorar la relación señal-ruido, consiguiendo que aumente la señal (intensidad del objeto) y disminuya el ruido ([Portiansky, 2018](#)). El ruido son todas aquellas perturbaciones aleatorias en la intensidad de una imagen debido normalmente a condiciones de poca luz, fallos en la cámara o movimientos de la muestra. Que desaparezca la mayor cantidad de ruido es primordial para la fiabilidad del análisis, consiguiendo una mayor nitidez y menor distorsión de la imagen ([Campa Fernández, 2016](#)).

Se van a presentar algunas de las técnicas más usadas en el procesamiento. La primera de ellas es la compresión, es una técnica bastante usual que reduce el tamaño original de la imagen para centralizarla y eliminar información redundante. Otra herramienta muy común en el procesamiento es la deconvolución, que se utiliza para eliminar distorsiones en la imagen como borrones, fallos en la iluminación o artefactos. Por otro lado, la transformación, es otro complemento donde se aplican distintas normas a cada píxel para cambiar la imagen, por ejemplo, el algoritmo que usa la transformada de Fourier sirve para detectar los bordes o cuantificar patrones repetitivos en la imagen. Por último, los filtros, son un tipo de transformación basadas en operaciones matemáticas que modifican los valores de intensidad de cada píxel, sirven para realzar características de la imagen, algunas aplicaciones comunes son

eliminar el ruido, el suavizado, la nitidez o mejorar el contraste de la imagen ([Portiansky, 2018](#); [Roeder et al., 2012](#); [Schroeder et al., 2021](#)).

Los filtros son una de las herramientas más frecuentes en el procesamiento, por ello, hay muchos tipos y clasificaciones posibles. Una de sus clasificaciones se basa en si depende linealmente de los píxeles entorno, si lo hace son filtros espaciales lineales o de convolución y si no, son filtros espaciales no-lineales o de no convolución. Como ejemplo, el filtro Gauss o gaussiano, es considerado como filtro de convolución ya que suaviza la imagen eliminando información de alta frecuencia. Un ejemplo de filtro de no convolución, es el filtro de la mediana, que elimina ruidos provocados por píxeles corruptos o faltantes, sustituyendo el píxel central por la mediana de los píxeles ([Portiansky, 2018](#); [Swedlow et al., 2009](#)).

2.3.3. Segmentación

Después de procesar la imagen, es necesario separar los objetos que se quieren contar o medir de aquellos que forman el fondo. Según [Portiansky \(2018\)](#), la segmentación sirve para reconocer objetos a través de su superficie o de sus bordes. Los programas tratan de crear patrones de identificación similares a como lo haría un humano, mediante diferentes algoritmos se agrupan los píxeles con características similares, ya sea en intensidad color, textura o forma ([Portiansky, 2018](#)).

Existen diversos métodos comunes para segmentar una imagen. En primer lugar, la segmentación manual, que implica el uso de herramientas de selección incluidas en el programa. La segmentación por color, basada en umbrales sobre el color o colores de interés. La segmentación por bordes, como dice su nombre delimita los bordes de un objeto. Nuevas técnicas como la segmentación por *machine learning* donde aprende los patrones de detección automáticamente. Y, por último, la umbralización o *threshold* en inglés, un método muy frecuente que se explicará seguidamente por su posterior uso en diferentes análisis. El *threshold* se basa en asignar un valor umbral a la intensidad de píxeles de una imagen, dando lugar a una imagen binaria en blanco y negro, donde los píxeles por encima del umbral se convierten en un color y los píxeles por debajo del umbral en otro ([González & Woods, 2018](#)). Destaca zonas de interés respecto del fondo de la imagen, por ejemplo, en la figura inferior (**Figura 4**) se muestra una umbralización de los núcleos de las células de un sépalo de *Arabidopsis* marcado con fluorescencia, una baja umbralización genera mucho ruido en la imagen, pero demasiada umbralización puede hacer que se pierda información, en la figura 4 se pierde el núcleo pequeño de la izquierda (B), es primordial mantener un equilibrio en el umbral ([Roeder et al., 2012](#)).

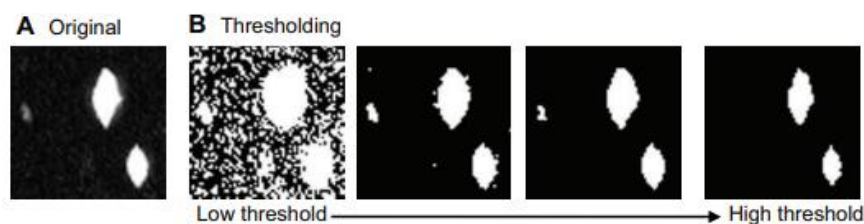


Figura 4: imagen en escala de grises de los núcleos de las células de un sépalo de *Arabidopsis* marcado con fluorescencia (A). Las diferentes imágenes (B) muestran el gradiente de una baja umbralización a una muy alta.

Fuente: adaptada de [Roeder et al. \(2012\)](#).

2.3.4. Cuantificación

Todos los pasos que se han estado describiendo convergen en un objetivo final, que es extraer información cuantitativa y objetiva a partir de imágenes. Con los datos conseguidos se podrán llevar a cabo muchos de los procedimientos mencionados en el apartado de usos del análisis de imagen cuyo propósito común es obtener información de las propiedades morfológicas y estructurales de las muestras biológicas.

Para cuantificar se llevan a cabo diversos métodos, como la calibración para establecer una escala entre los píxeles y las unidades físicas que representan, el conteo, determinar la orientación, la localización, la proximidad, la intensidad, el tamaño, la forma o colocalizar, para distinguir dos o más objetos próximos en un mismo ambiente ([Portiansky, 2018](#)). Al trabajar en estos análisis se busca la mayor precisión posible para que a la hora de interpretar todos los datos numéricos obtenidos estadísticamente se pueda sacar conclusiones fiables sobre las hipótesis previamente planteadas ([Campa Fernández, 2016](#)).

3. Software libre para el análisis de imágenes biológicas

Aunque existen multitud de programas para visualizar imágenes biológicas tanto de uso libre como de pago, esta memoria se va a enfocar en dos de los programas gratuitos más utilizados por la comunidad biológica, *Fiji* y *CellProfiler*.

3.1. *Fiji*

Fiji (“*Fiji is just ImageJ*”), es un paquete de procesamiento de imágenes digitales gratuito y de código abierto del programa *ImageJ*, creado en Estados Unidos por el Instituto Nacional de Salud (NIH) liderado por Wayne Rasband y sus colaboradores. Antes que nada, es necesario mencionar a los programas de análisis de imagen predecesores de *Fiji*, primeramente, surgió *NIH Image* en 1987 ([NIH Image, 2023](#)), que posteriormente pasó a llamarse *ImageJ* en

1997 ([ImageJ Wiki, 2023](#); [Schneider et al., 2012](#)). El gran auge de *ImageJ* en todo el mundo, derivó en algunos problemas de sostenibilidad, además, la plataforma se hizo para biólogos y no atraía tanto a científicos de otros campos de investigación. Así, en 2007 surgió *Fiji* ([Schindelin et al., 2012](#)), este software fue creado con la intención de dar soporte al *plugin* más complejo de *ImageJ* llamado “*TrakEM2*”, que se utiliza para el análisis, medición y extracción de datos morfológicos de imágenes de microscopía electrónica y además, *Fiji* sirvió como solución y mejora de muchos de los problemas técnicos de *ImageJ* ([Tinevez et al., 2017](#)).

Fiji es un programa muy amplio capaz de ejecutar todos los análisis mencionados en el apartado de los usos sobre el análisis de imagen y, por tanto, capaz de realizar los pasos de procesamiento, segmentación y cuantificación de una imagen convirtiéndose en un programa muy flexible. Esta capacidad de adaptación es la característica principal que presenta *Fiji*, y es tan adaptable gracias a la implementación automatizada de una gran variedad de *plugins*, concediendo al programa una capacidad ilimitada de progresión ampliando sus funcionalidades, también posee librerías de información, tutoriales y documentación útiles para hacer al programa familiar y sencillo de comprender para todos los usuarios, manteniendo al programa constantemente actualizado ([Schneider et al., 2012](#)). Tanto *ImageJ* como *Fiji* se desarrollaron en el lenguaje de programación *Java*, y aunque otros lenguajes como *C++* pueden ejecutar acciones más rápido, *Java* proporciona una facilidad, estabilidad y portabilidad con la que es difícil competir ([Schindelin et al., 2012](#)). Aunque esté desarrollado en *Java*, permite crear códigos en diferentes lenguajes de programación (*JavaScript, Python, R, Groovy*) ampliando el alcance del proyecto ([ImageJ Wiki, 2023](#)). Los investigadores con conocimiento en programación pueden contribuir a través de diferentes bibliotecas como “*ImgLib2*” o “*SciJava*”, donde se almacena información de las imágenes y donde se crean flujos de trabajos a partir de algoritmos complejos ([Schroeder et al., 2021](#)). A día de hoy, existen miles de *plugins* creados por la comunidad que se exponen en una wiki activa (<http://fiji.sc/>), donde viene documentación de todos los aspectos del programa. Además, existe un foro en línea llamado “*Scientific Community Image Forum*” que es usado diariamente por miles de usuarios de la plataforma y que sirve para informar y responder dudas relacionadas con el análisis de imagen ([Rueden et al., 2019](#)). Este proyecto ha ido evolucionando y ahora diversos programas como *KNIME, CellProfiler, Icy* y *MATLAB* se unieron al foro aumentando la relación entre la comunidad y los programas ([Dobson et al., 2021](#)). Cuando se crea un *plugin* y se quiere añadir al foro, se procede a una revisión de su código. Si pasa la revisión puede ser lanzado inmediatamente a través del “*Fiji Updater*” y cualquier persona podrá descargarlo. Los

desarrolladores mantienen el programa actualizado puesto que cada cierto tiempo organizan reuniones, donde trabajan en solucionar los errores e implementar mejoras ([ImageJ Wiki, 2023](#); [Schindelin et al., 2012](#); [Schneider et al., 2012](#); [Schroeder et al., 2021](#)).

3.2. Funcionamiento de Fiji

La interfaz de usuario, puede ser el principal factor por el que un científico se decante por un programa u otro, un diseño claro y una buena organización facilitará los análisis. A continuación, se explicará la interfaz de usuario y las capacidades que ofrece *Fiji*.

Fiji presenta la interfaz que se aprecia en la figura inferior (**Figura 5**), es una ventana donde en su parte superior aparecen diferentes menús que al seleccionarlos despliegan nuevas opciones. Si se sigue el orden de izquierda a derecha, cada menú se centra en: el manejo y apertura de los archivos, la edición, la imagen, su procesamiento, el análisis y, finalmente, la lista de *plugins* disponibles. Los dos últimos menús “*Window*” y “*Help*”, no son relevantes para un análisis puesto que son ventanas de información y ayuda. Justo debajo de cada menú se exponen las diferentes herramientas con las que se trabaja. Entre ellas se encuentran, herramientas de selección y medición como son las ocho primeras, que sirven para señalar ROIs (*Regions of Interest*) que son básicamente áreas delimitadas para su análisis, herramientas para dibujar y anotar datos en la imagen, un menú relacionado con datos del desarrollador, un menú para trabajar con *stacks* y un menú que enseña las LUT (*Look Up Table*) que son las diferentes tonalidades de color aplicables a una imagen. Abajo a la derecha hay una barra de búsqueda y abajo a la izquierda se encuentra lo que se conoce como barra de estado, que indica para qué sirve la herramienta seleccionada, en el caso de la figura inferior (**Figura 5**) muestra que está seleccionado el medidor de ángulos ([Campa Fernández, 2016](#); [ImageJ Wiki, 2023](#)).

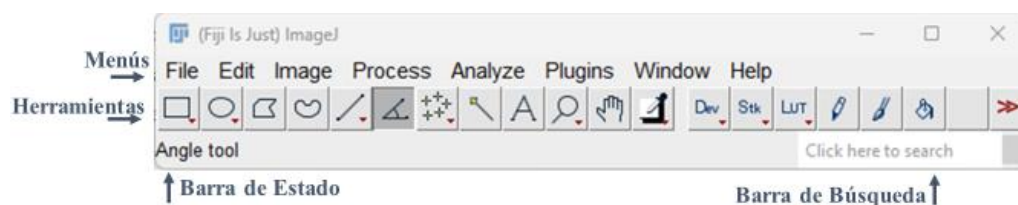


Figura 5: interfaz de *Fiji* para Windows. **Fuente:** elaboración propia a partir de [Fiji](#).

Siguiendo la figura superior (**Figura 5**), el primer menú es “*File*”, al abrirlo se muestran las diversas opciones que presenta, como abrir, guardar y cerrar la imagen con la que se esté trabajando. El segundo, es “*Edit*”, que no es demasiado relevante en este caso ya que son procesos de copiar, pegar y dibujar sobre una imagen. El tercer menú es “*Image*”, y está dividido

en diferentes submenús (**Figura 6**), el primero de ellos es “*Type*”, que sirve para consultar el tipo de imagen que se utiliza y permite convertirla a otro tipo de imagen (8-bit, 16-bit, RGB Color), el siguiente submenú es “*Adjust*”, posibilita realizar ajustes sobre diversas opciones como el brillo, el contraste, el color, el tamaño y proporciona diversas herramientas de umbralización como el ya mencionado “*Threshold*”. Muchas de estas opciones permiten ejecutar los ajustes automáticamente con el fin de mejorar la apariencia de la imagen en base a algoritmos numéricos, como, por ejemplo, aplicar umbralizaciones automáticas (“*Auto Threshold*”) donde con un solo clic umbraliza la imagen de la forma más óptima posible (Roeder et al., 2012). Otra de las funciones del menú “*Image*”, es el apartado “*Color*”, este submenú es exclusivo para imágenes a color, en él aparecen opciones como “*Split Channels*”, que se utiliza para separar canales RGB en tres imágenes monocromáticas independientes como se vio anteriormente para trabajar individualmente con cada canal, “*Merge Channels*”, hace la función inversa uniendo canales diferentes en una sola imagen. Asimismo, presenta diferentes funciones de las LUT en las que puedes asignar nuevos valores de intensidad para mejorar la apariencia visual de la imagen. El último submenú relevante es “*Stacks*” para crear, eliminar y modificar pilas de imágenes como ya se explicó en el apartado de características del análisis de imagen (ImageJ Wiki, 2023).

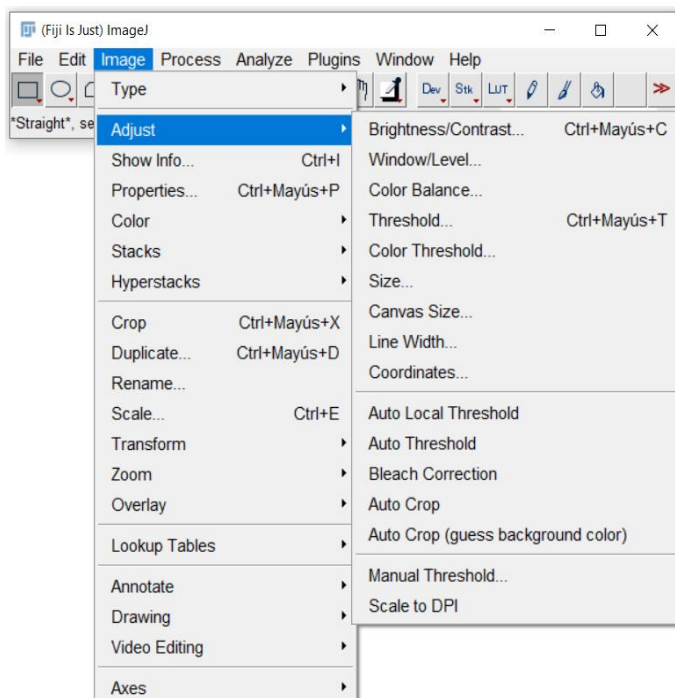


Figura 6: menú “*Image*” del programa *Fiji* que muestra el submenú “*Adjust*”. **Fuente:** elaboración propia a partir de *Fiji*.

El siguiente menú es “*Process*”, dedicado al procesamiento de una imagen, con el fin de mejorar la calidad de la imagen como ya se explicó en el apartado de procesamiento, aquí

están algunas de las herramientas ya mencionadas para modificar el ruido, el suavizado y múltiples filtros que cambian diversos aspectos de la imagen. El próximo menú y uno de los más importantes es “*Analyze*”, ya que contiene herramientas más dedicadas al análisis de imágenes, a continuación, se van a exponer algunas de sus opciones. La primera herramienta es “*Measure*”, su función es mostrar en una tabla de resultados las medidas de un área seleccionada o de la imagen completa. Otra opción disponible es “*Analyze Particle*”, permite contar y medir objetos de interés en imágenes binarias o umbralizadas y “*Summarize*”, calcula parámetros numéricos como la media, la desviación estándar, el máximo y mínimo del conjunto seleccionado. A partir de estos datos es posible generar gráficos como histogramas que representan la distribución de intensidad de píxeles en una imagen dándonos información sobre el brillo, el contraste, la saturación, la exposición y el rango dinámico de la imagen (Campa Fernández, 2016). Con “*Set Scale*” se hace un calibrado espacial de las imágenes relacionando los píxeles y las unidades de longitud, generando una barra de escala. Luego, la herramienta “*ROI Manager*”, es muy útil para agrupar regiones de interés, clasificarlas y posteriormente compararlas. Terminando con el menú “*Analyze*”, para cuantificar geles electroforéticos viene incluido un apartado llamado “*Gels*”, que permite hacer un cuantificado manual del área de cada banda electroforética (ImageJ Wiki, 2023).

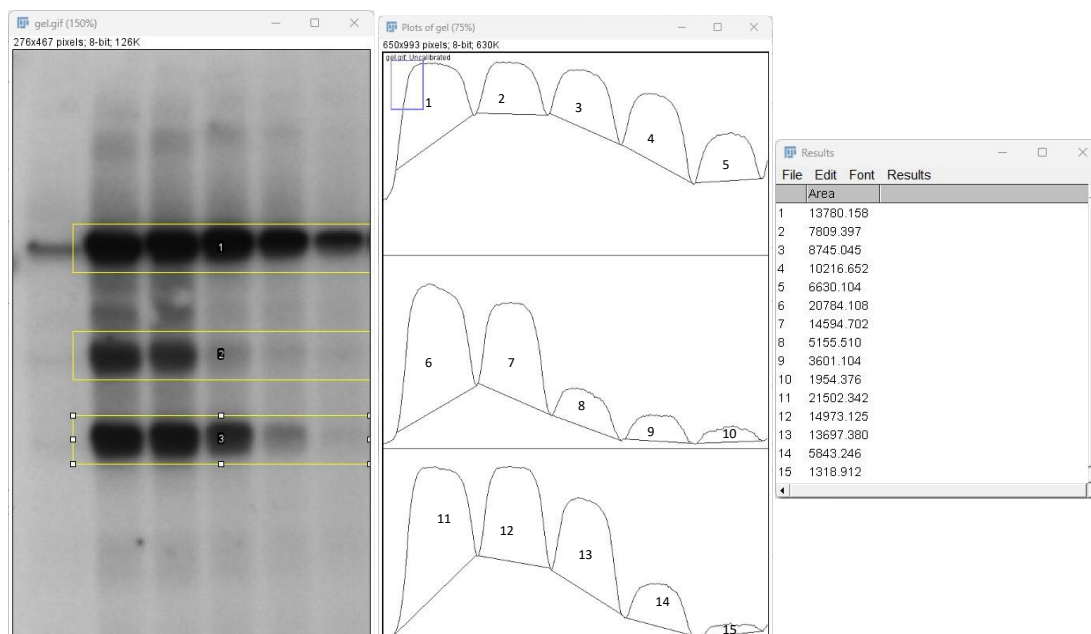


Figura 7: la primera imagen es un gel de electroforesis donde están seleccionadas las 3 zonas de estudio. La segunda imagen es el área gráfica de cada zona separada manualmente por las herramientas de selección. La tercera imagen corresponde a las medidas del área obtenida de cada banda. **Fuente:** elaboración propia a partir de Fiji.

Para finalizar, Fiji trae incluido el menú “*Plugins*”, que contiene de forma predeterminada muchos de los complementos utilizados para el análisis de imagen y un apartado dedicado a la

instalación de nuevos *plugins*. Se van a mencionar, como ejemplo, tres de ellos ([ImageJ Wiki, 2023](#)).

El primero es “*Bio-formats*”, es primordial a la hora de importar, exportar y procesar muchos de los formatos de archivo de imagen que *Fiji* en un principio no podría analizar, así, amplía la posibilidad de leer nuevos archivos ([Schroeder et al., 2021](#)). El segundo *plugin*, ha tenido un gran reconocimiento científico, es SNT (“*Simple Neurite Tracer*”), está dedicado a la morfometría neuronal y a la conectómica del cerebro, lleva a cabo el rastreo, edición, visualización, cuantificación y modelado de la neuroanatomía, es un gran recurso para la neurociencia ([Arshadi et al., 2021](#)). El último, es un *plugin* muy interactivo conocido como “*3D Viewer*”, para visualizar datos e imágenes en tres dimensiones. Con herramientas para rotar, seleccionar canales y cuantificar las imágenes ([Schmid et al., 2010](#)).

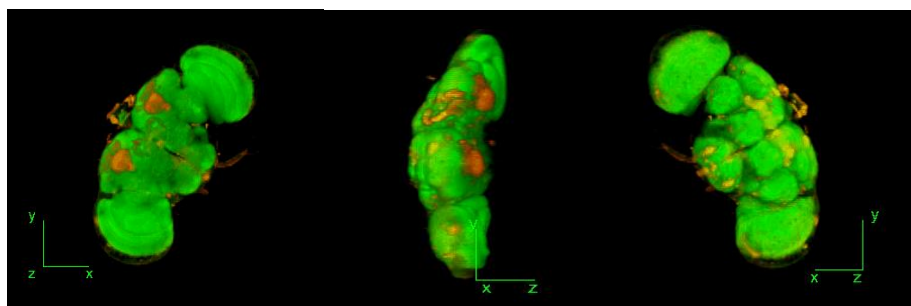


Figura 8: representación de un cerebro de mosca en *Fiji* usando el *plugin* “*3D Viewer*” para su análisis 3D. **Fuente:** adaptada de la página web [Fiji](#).

Una vez vista la interfaz, se van a explicar los parámetros más importantes que muestra *Fiji* al observar una imagen, utilizando de muestra la figura inferior (**Figura 9**) de un proceso de mitosis, algunos de estos parámetros pueden no aparecer ya que eso depende de la imagen a analizar. En la parte superior se muestra el logo del programa, el nombre del archivo y el tipo de archivo, en este caso es un archivo TIF y entre paréntesis el zoom que tiene aplicado la imagen (200%). Debajo del nombre, se dispone información sobre los canales, planos, fotogramas y formato de la imagen. La figura ejemplo muestra **c:2/2 z:5/5 t:24,51; 15.14x17.35 μm (171x196); 16-bit; 33MB** (**Figura 9**). Indica que se está viendo el segundo canal (verde), quinto plano, veinticuatroavo punto temporal de un stack de dos canales, cinco planos y cincuenta y un puntos temporales con un tamaño de imagen de 171x196 píxeles, que corresponden a 15.14x17.35 micras de la muestra, además, la imagen es de 16-bit y ocupa 33 megabytes ([Campa Fernández, 2016](#)). En la parte inferior de la imagen se puede cambiar de canal, de plano y visualizar en una animación cada punto temporal de la mitosis y pararlo en un punto concreto si es necesario ([ImageJ Wiki, 2023](#)).

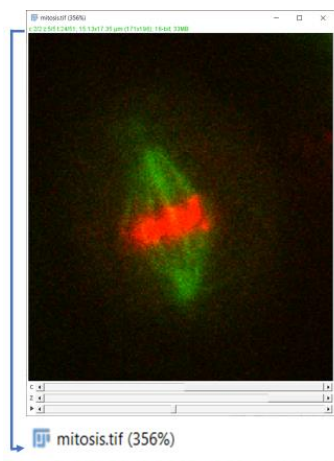


Figura 9: imagen de un proceso de mitosis analizado mediante el software *Fiji*. Justo debajo ampliación de la parte superior de la interfaz. **Fuente:** elaboración propia a partir de *Fiji*.

3.2.1. Flujo de trabajo de un análisis en *Fiji*.

Para ver un ejemplo de un análisis en *Fiji*, se ha realizado manualmente un dibujo de estructuras ramificadas simulando lo que podrían ser neuronas, vasos sanguíneos, glándulas, raíces de plantas y otras estructuras ramificadas filamentosas. Este dibujo original RGB se ha tenido que transformar en una imagen 8-bit para poder utilizar el plugin “*Skeletonize (2D/3D)*” que ha delimitado las estructuras ramificadas, posteriormente con la herramienta “*Analyze Skeleton (2D/3D)*” se ha llevado a cabo un análisis cuantitativo para ver entre otras medidas, el número de ramas, el número de uniones, la longitud de cada rama o la distancia euclídea.

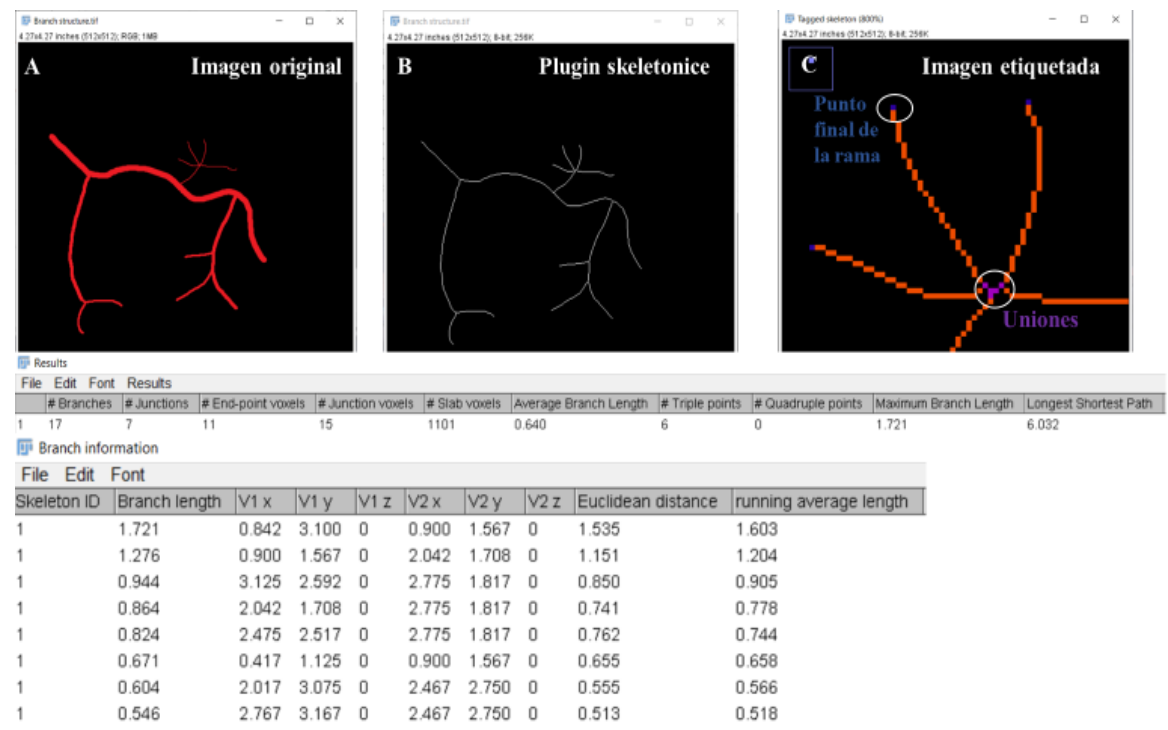


Figura 10: estructura ramificada en 2D (A) en la cual se ha aplicado el *plugin* “*Skeletonize (2D/3D)*” (B) y luego se ha analizado dicha figura con “*Analyze Skeleton (2D/3D)*” dando una imagen C que muestra las uniones y los vóxeles finales de cada rama. Justo debajo de las imágenes A, B y C se muestran dos tablas de resultados con datos obtenidos del análisis. **Fuente:** elaboración propia a través de *Fiji*.

3.3. CellProfiler

CellProfiler, es un programa de uso libre, versátil y de código abierto para cuantificar automáticamente grandes cantidades de información sobre imágenes biológicas, especialmente experimentos de alto rendimiento ([CellProfiler, 2023](#); [Haase et al., 2022](#)). Se lanzó en 2005 por el Instituto Whitehead de investigación biomédica en Cambridge y el Instituto Tecnológico de Massachusetts y actualmente lo mantiene el laboratorio Cimini, en el instituto Broad del MIT y Harvard ([Carpenter et al., 2006](#)). Fue desarrollado primeramente en el lenguaje de programación *MATLAB*, pero reescrito en *Python*. La propiedad principal de este programa es el uso de *pipelines* que son secuencias de módulos ordenadas. Cada uno de estos módulos está diseñado para realizar una tarea específica y en conjunto, forman una serie de pasos consecutivos que dan lugar a un flujo de trabajo para realizar un análisis continuo de una imagen, cada módulo es personalizable según lo que se busque analizar ([Lamprecht et al., 2007](#); [Stirling et al., 2021](#)). Desde que entras en su página web (<https://cellprofiler.org/>) muestra que es un programa que se basa en su sencillez. En la web destaca la posibilidad de descargar una gran lista de *pipelines*, que se han utilizado en diferentes estudios, con una descripción de cómo usarlas ([Carpenter et al., 2006](#); [CellProfiler, 2023](#)). Esto permite a los investigadores compartir sus datos fácilmente para que otros puedan usarlo libremente. Este software realiza los pasos comunes mencionados de un análisis de imagen, sirviendo así para identificar objetos de interés y luego medir sus propiedades, con métodos como la segmentación o la cuantificación en base a las *pipelines* diseñadas para ese estudio ([Dobson et al., 2021](#)). Con el tiempo han salido nuevas versiones del mismo con mejoras notables en velocidad de procesamiento y utilidad, la última gran actualización disponible es *CellProfiler 4.0* ([Stirling et al., 2021](#)). Se ha convertido en un programa muy consolidado que ha logrado establecer una gran comunidad al igual que *Fiji*, donde ambos cuentan con más de 10000 citas anuales en diversos artículos científicos ([Dobson et al., 2021](#); [Rueden et al., 2019](#)).

3.4. Funcionamiento de CellProfiler

En este apartado también se empieza describiendo la interfaz y a partir de ella, se exploran las funciones disponibles del programa. Tomando la figura 11 como referencia, *CellProfiler* presenta una serie de menús en la parte superior izquierda al igual que *Fiji*, que ayudarán a analizar el conjunto de datos, estos menús son “*File*”, “*Edit*”, “*Test*”, y dos menús que proporcionan información de ayuda del programa “*Windows*” y “*Help*”. El menú “*File*”, brinda distintas posibilidades destinadas a abrir y guardar *pipelines*, y a dar comienzo al análisis.

El menú “*Edit*” muestra una lista de opciones para administrar los módulos disponibles, donde se puede agregar nuevos módulos, reorganizar su posición, copiarlos o eliminarlos. A su vez, el programa da la opción de ejecutar una *pipeline* en una vista previa antes de realizar el análisis para ajustar la configuración del módulo sobre la marcha, esto se hace con el menú “*Test*”, y además, proporciona diferentes opciones de selección para trabajar con el set de imágenes y módulos ([CellProfiler, 2023](#); [Stirling et al., 2021](#)).

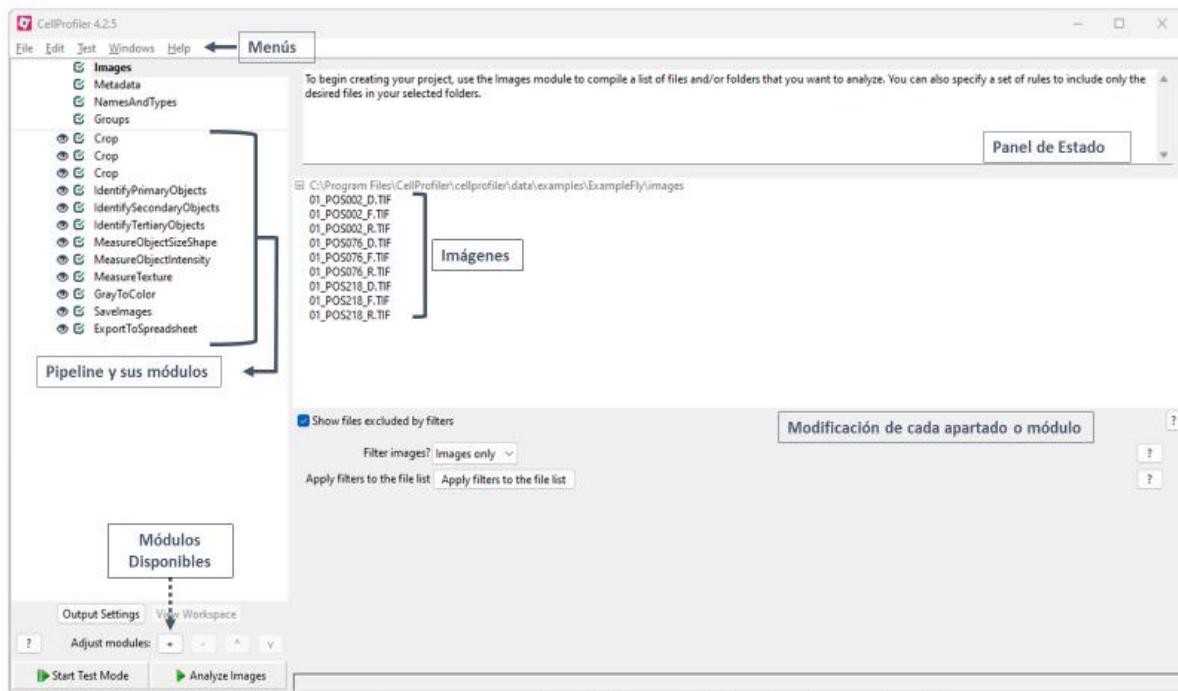


Figura 11: interfaz inicial de *CellProfiler*. **Fuente:** elaboración propia a partir de [CellProfiler](#).

A la izquierda está el panel de la *pipeline*, en la figura 11 se presenta una *pipeline* a modo de ejemplo, compuesta de módulos en los que cada uno tiene su tarea específica. En la parte inferior izquierda hay dos opciones, la primera es “*Start Test Mode*” que inicia el modo test o vista previa mencionado anteriormente donde es posible ver y modificar cada módulo individualmente. La otra opción es “*Analyze Images*”, que efectúa el análisis completo sin pausas. Encima de la *pipeline* se visualizan una serie de apartados empezando por “*Images*” que es donde se almacenarán el conjunto de imágenes que se importan para el análisis y estas aparecerán en el panel central, por ejemplo, en la figura 11 hay una serie de imágenes en formato TIF. La casilla “*Metadata*” extrae información adicional de las imágenes y la almacena, la casilla “*NamesandTypes*” permite nombrar a nuestras imágenes y, por último, “*Groups*” agrupa las imágenes en subconjuntos ([CellProfiler, 2023](#)).

Un apartado de especial importancia se localiza al clicar en la casilla “+” de “*Adjust modules*”, se abre una nueva ventana que presenta las categorías de módulos que existen en el

programa y los módulos ejecutables dentro de cada categoría, cada uno presenta unos ajustes propios y una serie de parámetros modificables (**Figura 11**) ([Carpenter et al., 2006](#)). La primera categoría es “*File Processing*”, al igual que menú “*File*” sirve para exportar y guardar datos de imágenes. La segunda categoría es “*Image Processing*”, donde cada módulo aplica una operación concreta sobre una imagen que la convierte en una nueva, entre sus distintas opciones se encuentran, por ejemplo, “*ColorToGray*”, para devolver a escala de grises una imagen a color, “*GrayToColor*” es su inverso, da color a una imagen. “*Crop*”, para recortar imágenes hasta obtener la zona que se desea, “*EnhanceEdges*” para identificar los bordes de una imagen, “*Smooth*” para suavizar una imagen o el ya nombrado “*Threshold*”. La siguiente categoría es “*Objetc Processing*”, se basa en módulos que se encargan de manipular objetos identificados en una imagen. Un objeto es cualquier característica identificable en una imagen, generalmente un organismo, célula o compartimento celular (núcleos, células, colonias). “*ClassifyObjects*” clasifica los objetos de acorde a unos valores numéricos que se elijan, “*CombineObjects*” une un set de dos objetos en un solo. También hay módulos de identificación de objetos, ya sea manualmente o automáticamente, por ejemplo, “*IdentifyPrimaryObjects*” detecta automáticamente los objetos biológicos de interés siempre que sean imágenes en escala de grises 2D con objetos brillantes sobre un fondo oscuro. La siguiente sección es “*Measurement*”, en ella hay módulos para medir diferentes procesos u objetos como la intensidad de píxeles de la imagen, medir el área que ocupa un objeto en una imagen, medir la forma y el tamaño de un objeto, determinar la colocalización entre dos objetos y otros procesos que aparecen en la figura 12 ([CellProfiler, 2023](#)).

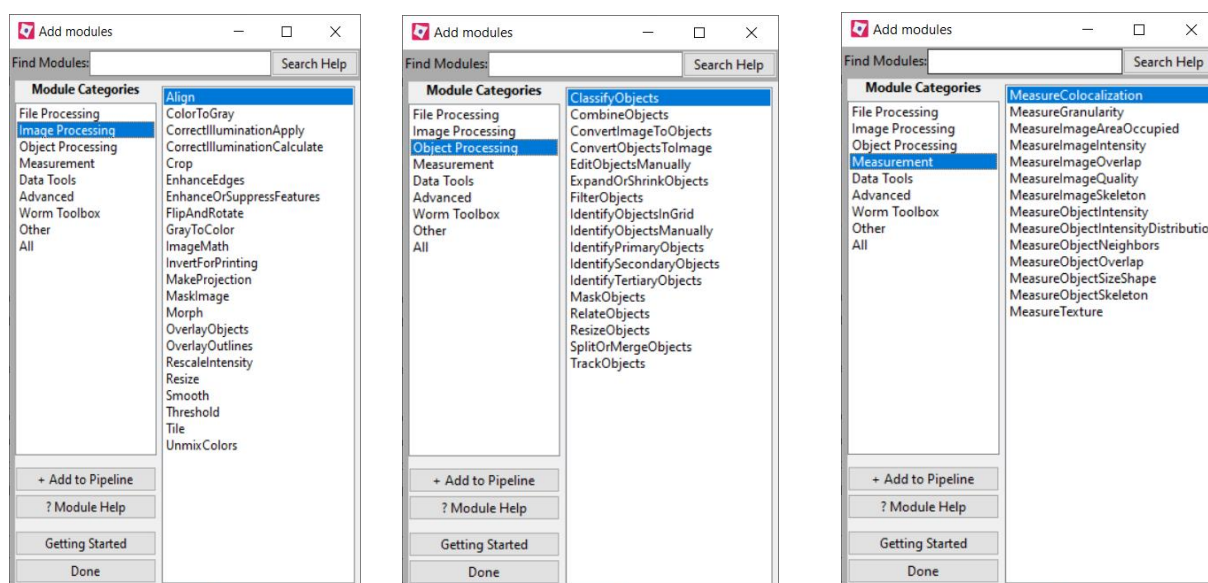


Figura 12: apartado de módulos de *CellProfiler*. La primera ventana muestra la categoría “*Image Processing*”, la segunda “*Object Processing*” y la tercera “*Measurement*”. **Fuente:** elaboración propia a partir de [CellProfiler](#).

En la quinta categoría se encuentra “*Data Tools*”, son herramientas dedicadas a la ejecución de operaciones matemáticas y estadísticas, a su vez, permite generar diferentes figuras de análisis cuantitativo como histogramas o gráficos de dispersión. La próxima categoría es “*Advanced*”, y está destinada a hacer muchas de las técnicas ya nombradas de procesamiento como aplicar distintos filtros. Finalmente, también existe una categoría exclusiva dedicada al análisis de gusanos ([CellProfiler, 2023](#)).

Cuando se analiza una imagen en *CellProfiler*, se encuentra la información que aparece en la figura inferior (**Figura 13**). Este programa, mide los ejes y la intensidad automáticamente, en la parte inferior de la ventana se aprecian unos valores concretos que son medidos por la posición en la que se mantiene el mouse en ese instante. Los valores de intensidad en *CellProfiler* a diferencia de otros programas que miden de 0 a 255, como *Fiji*, se miden de 0 a 1 porque la mayoría de algoritmos de umbralización trabajan mejor así, es la misma información, pero a escala ([Carpenter et al. 2006](#)). Hay herramientas como la casa, para volver a la imagen original, el zoom para ampliarla, las flechas para rehacer o deshacer un paso o la barra para medir longitudes y con clic derecho en la imagen se podrá ajustar el contraste u obtener un histograma de la intensidad de los píxeles ([CellProfiler, 2023](#)).

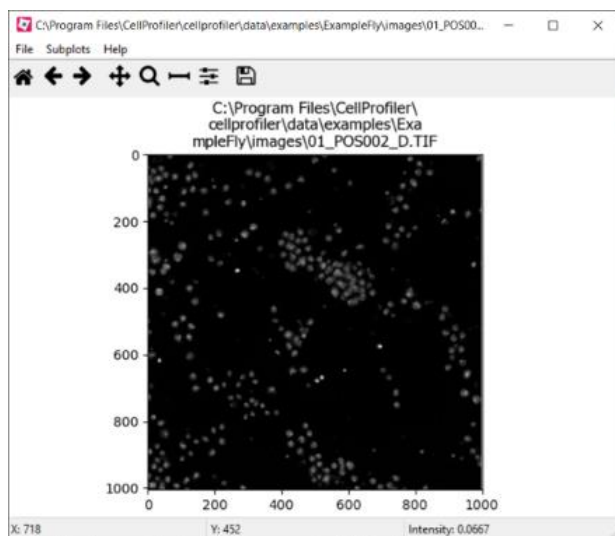


Figura 13: proceso de análisis de imagen en *CellProfiler* con un archivo que muestra las células Kc167 de *Drosophila melanogaster*. **Fuente:** elaboración propia a partir de [CellProfiler](#).

3.4.1. Flujo de trabajo de una *pipeline* en *CellProfiler*

En la siguiente figura (**Figura 14**) se expone un modelo simple de una *pipeline* por defecto de *CellProfiler* con células Kc167 de *Drosophila melanogaster*. En las imágenes 1 y 2 se produce un *crop* o recorte para ampliar la sección que se quiere visualizar, luego se utiliza los módulos de identificación de objetos primarios, secundarios y terciarios (células, núcleos y citoplasma) que incluye un ajuste del contraste de la imagen y, por último, se mide la

información sobre el tamaño de los objetos identificados ([CellProfiler, 2023](#); [Stirling et al., 2021](#)).

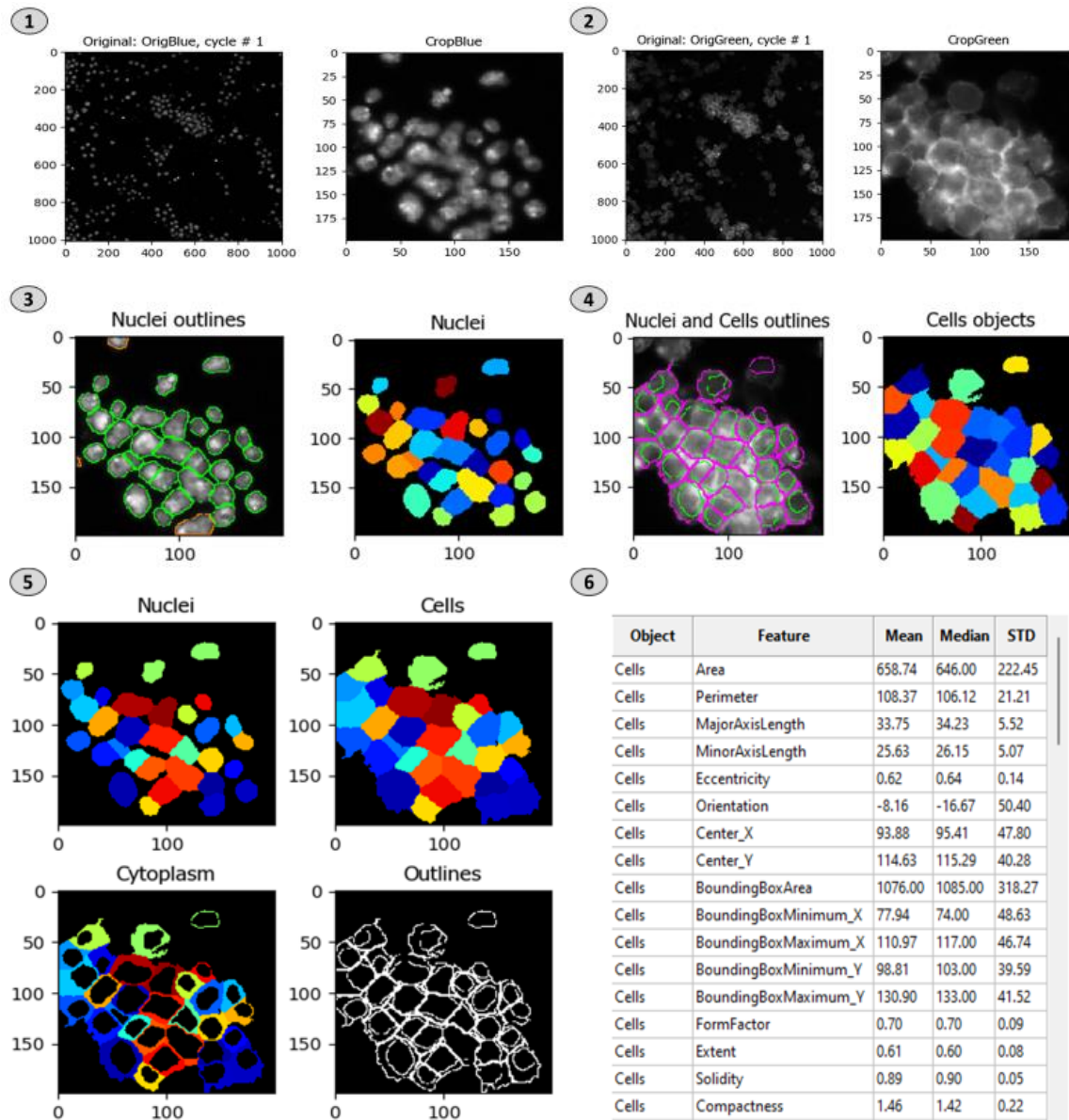


Figura 14: pipeline en células Kc167 de *Drosophila melanogaster* usando *CellProfiler*. **Fuente:** elaboración propia a partir de [CellProfiler](#).

3.5. Comparativa entre *Fiji* y *CellProfiler*

Como ya se ha visto, *Fiji* destaca por su capacidad de implementar *plugins* en su algoritmo y *CellProfiler* está diseñado para configurar *pipelines* automatizadas en análisis a gran escala ([Haase et al., 2022](#)). Aunque ambos programas pueden realizar tareas similares y depende mucho del contexto y la complejidad del análisis, es importante resaltar algunos de los puntos de comparación entre ellos.

Al crear un experimento en el mismo ordenador se ha observado que, en muchos casos *CellProfiler* es más rápido en procesamiento de grandes conjuntos de imágenes, debido a que su arquitectura de módulos está diseñada para optimizar la eficiencia y escalabilidad ([Dobson et al., 2021](#); [Laghi et al., 2022](#)). Los múltiples módulos ejecutan análisis en paralelo en los que se usan diferentes núcleos de CPU y GPU del ordenador para aprovechar al máximo la velocidad de procesamiento del ordenador. *Fiji*, sin embargo, no posee módulos y por eso es un poco menos eficiente, se basa en *macros* y en la capacidad de procesamiento del ordenador, aun así, tiene un rendimiento competitivo ([Eliceiri et al., 2012](#)).

Una vez se observan las interfaces de ambos programas (**Figura 5 y Figura 11**), se puede conseguir mucha información de ellos. *CellProfiler*, es más orientativo, trata de ayudar al investigador en cada paso que da marcando cada herramienta o apartado con un símbolo de interrogación, que informa sobre cómo usarlo. Además, posee el manual online que explica cada apartado y también muestra un ejemplo de cómo sería el análisis usando una *pipeline* concreta ([CellProfiler, 2023](#)). Por otro lado, *Fiji*, solo tiene un menú de ayuda que envía a la página web del programa, donde sí hay información suficiente para saber cómo manejarlo, asimismo, posee muchas imágenes de ejemplo sencillas para familiarizarse con el análisis. Ambos programas destacan por su clara organización basada en menús ordenados con cada proceso que se puede ejecutar y se fundamentan en el uso de ventanas que cuando ejecutan alguna operación abren más ventanas que reflejan las nuevas modificaciones ([ImageJ Wiki, 2023](#)). *Fiji*, posee muchas más herramientas personalizadas de análisis que sumada a la gran cantidad de *plugins*, más de 1000 actualmente, lo hacen un programa más personalizable y flexible en cualquier campo de estudio. *CellProfiler*, presenta módulos bastante personalizables, pero no llegan a ser tantos como *plugins* tiene *Fiji*, que al tener una mayor comunidad pues muchos más científicos se han dedicado a programar complementos para mejorarlo ([ImageJ Wiki, 2023](#); [Schroeder et al., 2021](#)).

Un estudio llevado a cabo por [Laghi et al. \(2022\)](#), empleaba *pipelines* que crearon específicamente en *CellProfiler* para cuantificar parámetros musculares en ratón y para verificar que el estudio se realizaba correctamente, comprobaron manualmente mediante *Fiji* los resultados ([Laghi et al., 2022](#)). Como se aprecia en la figura inferior (**Figura 15**) las diferencias en los resultados eran ínfimas, de pocos decimales o incluso centésimas, esto es un claro ejemplo, de que, aunque sean programas diferentes los resultados de cada uno son muy similares.

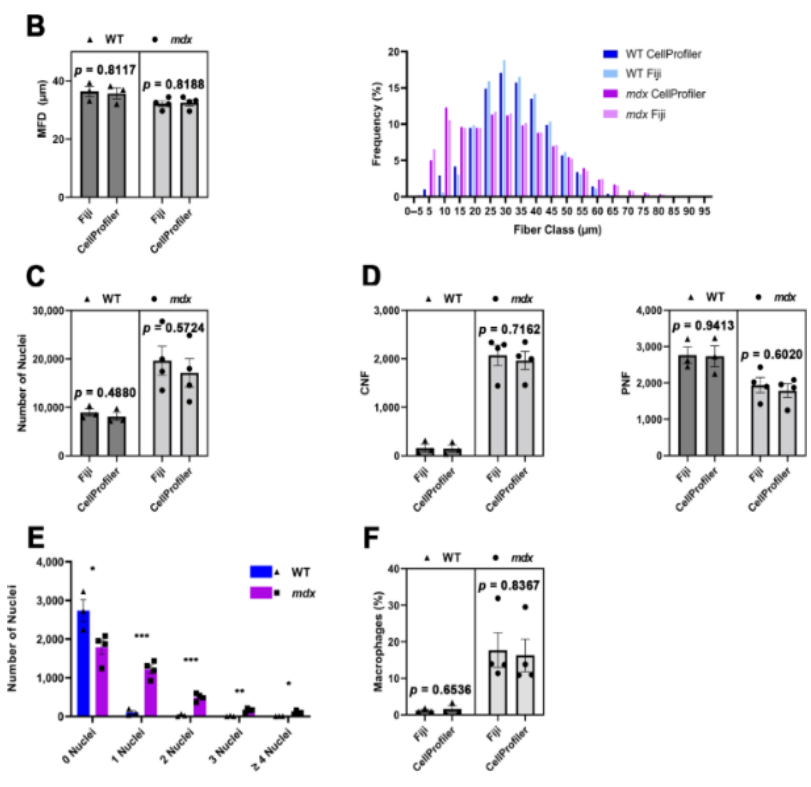


Figura 15: medidas cuantitativas automáticas con *CellProfiler* y comparadas con *Fiji*. B) Cuantificación del diámetro mínimo de Feret (MFD) y distribución del tamaño de fibra de las miofibras. C) Cuantificación del número de núcleos. D) Cuantificación de fibras nucleadas centralmente (CNF) y periféricamente (PNF). Cuantificación del número de núcleos por CNF realizada por *CellProfiler*. F) Cuantificación del porcentaje de macrófagos. **Fuente:** adaptada de [Laghi et al. \(2022\)](#).

Aparte de comparar ambos programas, es esencial hablar de ellos en conjunto puesto que así es cuando realmente le sacan partido al análisis, se retroalimentan tomando las ventajas de cada programa ([Rueden et al., 2019](#)). Para apoyar la cooperación entre ambos programas se han creado nuevos complementos como el módulo “*RunImageJ Macro*”, que permite a los usuarios intercambiar datos de una *pipeline* de *CellProfiler* a una *macro* de *Fiji* o *ImageJ* y al revés. También se ha desarrollado un nuevo lenguaje de programación mezcla de *Python* y *Java* llamado *PyImageJ* que ha resultado en una solución al gran obstáculo que tenían para crear flujos de trabajo entre ambos programas. En un artículo de [Dobson et al. \(2021\)](#), se utilizaron ambos programas para demostrar el beneficio que suponía la colaboración entre ambos. En el primer experimento, se utilizaba el *plugin* “*TrackMate*” de *Fiji* para el seguimiento de células, y *CellProfiler* para detectar y cuantificar la morfología celular. Y en el segundo protocolo, se hacía una segmentación cuantitativa de distintos fenotipos en *CellProfiler*, con ayuda de herramientas de manipulación de imágenes de *Fiji* para verificar los resultados. Concluyeron que, aunque a veces, haya que hacer comprobaciones manuales y estos programas funcionen perfectamente por separado, al hacerlo en colaboración han podido identificar fenotipos extraños que por separado en el análisis masivo no se detectaban, también resultó en una mejora en la calidad, la personalización de las imágenes y un ahorro de tiempo en cada proceso, ya que usaban la especialidad de cada programa ([Dobson et al., 2021](#)).

Finalmente, se ha llevado a cabo un análisis comparativo experimental entre *Fiji* y *CellProfiler* para este proyecto, de un mismo proceso de identificación en células U2OS de cáncer óseo en humano. Incluso realizando un análisis breve (**Figuras 16 y 17**) para contar el número células en una imagen y medir ciertos datos como el área media y la mediana, los valores y el desarrollo del análisis son muy similares.

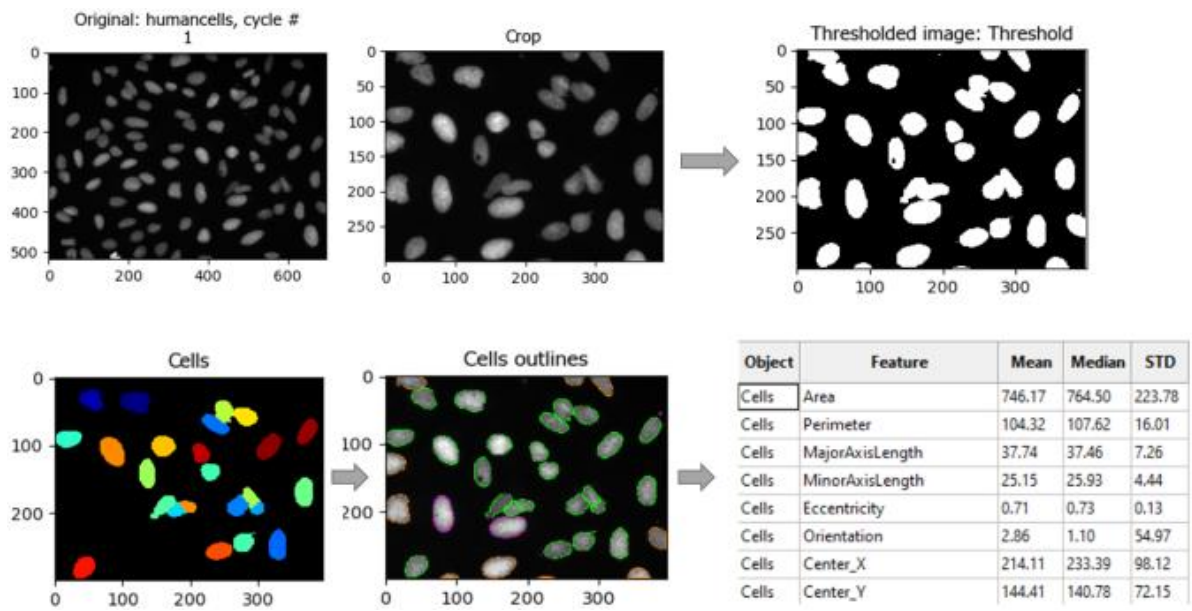


Figura 16: proceso de análisis de imagen de *CellProfiler* donde se observa secuencialmente, la imagen original, un recorte, una umbralización, la identificación de células y la medición distintos valores numéricos. **Fuente:** elaboración propia a partir de [CellProfiler](#).

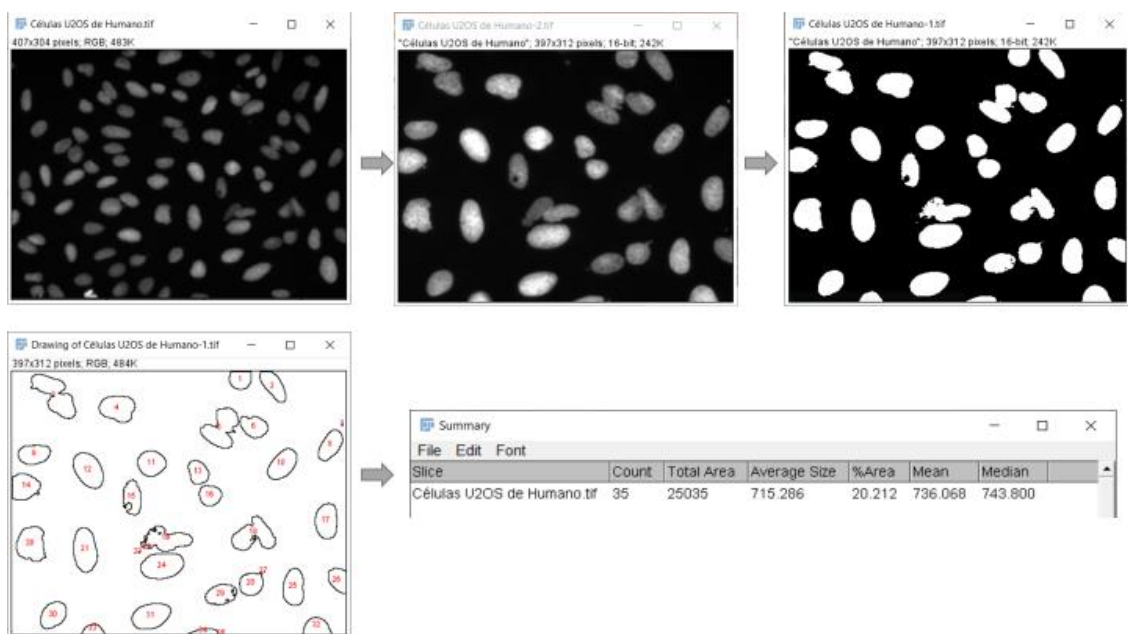


Figura 17: proceso de análisis de imagen de *Fiji*, donde se observa secuencialmente la imagen original, un recorte, una umbralización, la identificación de células y la medición de distintos valores numéricos. **Fuente:** elaboración propia a partir de [Fiji](#).

4. Ventajas y desventajas del software de uso libre respecto al software de propietarios

El objetivo de este apartado será mencionar algunos de los beneficios y limitaciones que presentan las herramientas informáticas de uso libre para el análisis de imágenes en comparación con las de propietarios. Primero, es necesario mencionar alguno de los programas de propietarios más conocidos en el análisis de imagen, *Huygens*, especializado en microscopía de fluorescencia y superresolución ([Huygens software, 2023](#)), *Volocity*, un software diseñado para el análisis 3D y 4D de imágenes de fluorescencia ([Volocity software, 2023](#)) y *Zen*, para análisis sencillos y directos con un amplio repertorio de técnicas de microscopía ([Software de microscopía ZEN, 2023](#)).

Hay que tener en cuenta que las ventajas y desventajas se proponen de manera general y cada una variará según los softwares que se comparen.

La primera ventaja de los programas de uso libre respecto a los de pago y de la cual proviene su nombre, es el coste, básicamente son programas gratuitos para todo el mundo, sin presupuestos adicionales por obtener la licencia ([Solomon, 2009](#)). A pesar de que se puede pensar que existe competencia entre programas libres y de propietarios, las interacciones entre ambos han sido de beneficio mutuo ya que el objetivo es crear soluciones a las necesidades del investigador y así, colaborar con el intercambio de información entre científicos ([Carpenter et al., 2012](#)). Otra gran importancia de que el código este abierto para todos los usuarios, es la flexibilidad y adaptabilidad, así cualquier investigador que sepa de programación podrá modificar el código añadiendo los ya nombrados complementos, para esa renovación y modernización continua de los programas ([Swedlow et al., 2009](#)). La interoperabilidad es otro factor importante, que se trabaje no solo usando los datos de su programa sino también transmitiendo los datos a otros, por ejemplo, crear unos datos en un software libre y pasarlos a un software específico para un mejor análisis, si alguno de los dos programas no es compatible o capaz de pasar los datos al otro puede convertirse en un gran cuello de botella si el algoritmo no se puede integrar, esto limitaría el rango de actuación ([Eliceiri et al., 2012](#); [Swedlow et al., 2009](#)). Gracias a la interoperabilidad, los programas se diseñan para funcionar en diversas plataformas y sistemas operativos como Windows, Linux y MacOS sin necesidad de adquirir versiones separadas para cada plataforma. Los programas de uso libre tienden a ser más interoperables que los de propietarios, creando puentes de intercambio de información ([Eliceiri et al., 2012](#); [Schneider et al., 2012](#)).

Como se explicó en el apartado anterior, los programas pueden colaborar unos con otros para construir flujos de trabajo comunes, donde los datos de una imagen se puedan traspasar a otro programa. *Fiji* y *CellProfiler* han colaborado juntos para crear el COBA (*Center for Open Bioimage Analysis*), un gran proyecto que pretende unificar todo el campo del análisis de imágenes biológico ([Dobson et al., 2021](#)). A lo largo del proyecto se ha mencionado la gran comunidad de usuarios que poseen estos programas gratuitos y es que una comunidad tan activa aporta reseñas, críticas, información y recursos constantemente convirtiéndose en el núcleo de todos estos programas, así surgieron algunos foros de análisis de imagen como el ya nombrado “*Scientific Community Image Forum*”, con cifras de más de 10000 personas y 15000 publicaciones por año ([Rueden et al., 2019](#)).

Entre las “desventajas” de los softwares de uso libre respecto a los softwares de propietarios se encuentra el soporte técnico, los desarrolladores de los softwares libres normalmente no tienen el mismo nivel de recursos económicos para proporcionar ayuda y actualizaciones constantes ([Eliceiri et al., 2012](#)). Los softwares de propietarios, presentan una mayor eficacia al tener personal implicado en resolver cualquier duda con la mayor brevedad posible. Sin embargo, como ya se ha visto, en los programas gratuitos, existen multitud de opciones donde los usuarios comparten su conocimiento y ayudan a otras personas a resolver sus problemas ([Dobson et al., 2021](#); [ImageJ Wiki, 2023](#)). Algunos programas de software libre tienen una curva de aprendizaje más difícil, no son tan estéticamente entendibles y suelen requerir mayor conocimiento informático sobre programación. La interfaz de usuario de los programas de propietarios suele ser bastante más intuitiva y comprensible para personas que no son expertas en informática, llamar la atención del comprador con una interfaz llamativa es importante para las ventas ([Roeder et al., 2012](#); [Schindelin et al., 2012](#)).

Realmente las desventajas pueden existir más bien entre programas concretos, ya sean de propietarios o libres, las desventajas recién mencionadas no tienen por qué aparecer siempre que se traten de softwares gratuitos, el gran alcance y desarrollo que han tenido los programas gratuitos ha conseguido que estos mejoren, convirtiéndolos en programas igual de competitivos que los programas de propietarios, y por ello, si se quieren comprobar las ventajas y desventajas de dos programas es más conveniente que sean simplemente programas similares, ya sean de pago o gratuitos. Los principales factores claves a la hora de decantarse por un programa serán, el propio gusto del consumidor de la herramienta, los factores estéticos de la interfaz, la velocidad de procesamiento o la comunidad que alberga ([Dobson et al., 2021](#); [Eliceiri et al., 2012](#); [Haase et al., 2022](#)).

5. Conclusión

A modo de cierre de esta memoria, podemos concluir que el análisis de imagen computarizado ha permitido llevar la investigación biológica a un nivel superior. Hemos podido ver que esta área del conocimiento tiene mucho que ofrecer, gracias a la amplia lista de opciones que plantea para tratar una imagen, aportándonos una cuantiosa cantidad de información sobre numerosos procesos biológicos. También, hemos comprobado que cada paso en el análisis es esencial y necesario para poder llevar a cabo a una investigación sólida.

Como se ha recalcado varias veces a lo largo del texto, la mayor parte del progreso proviene del vertiginoso crecimiento que ha tenido la comunidad que utiliza estas herramientas de análisis de imagen. Actualmente, los investigadores, están dejando de centrarse en los programas individualmente y están empezando a valorar todo como un proyecto único, siendo la comunidad incluso más importante que los propios programas. Por todo esto, se ha podido concluir que la idea principal de aquellos que forman parte de esta disciplina científica es llegar a la mayor cantidad de personas posibles mediante foros, publicaciones científicas, tutoriales y documentación. Así, los usuarios contribuyen al análisis resolviendo dudas, expandiendo el conocimiento o aportando recursos con herramientas como los *plugins* y módulos para aumentar la funcionalidad y automatización de los programas. Alcanzar esta conexión entre usuarios no hubiese sido posible si no fuese por los programas de uso libre, al ser gratuitos e interoperables han permitido que el intercambio de información alcance una gran extensibilidad, siendo programas como *Fiji* y *CellProfiler* pioneros en este ámbito.

Una vez expuesto el modo de trabajo de *Fiji* y *CellProfiler*, hemos determinado que, aunque existen pequeñas diferencias entre ambos, a la hora de interpretar los resultados de un estudio la variación en los datos es apenas perceptible y la mejor manera de aprovechar el rendimiento de ambos es usarlos en sinergia y esto es extrapolable a cualquier otro programa de análisis de imagen. Por otra parte, aunque los softwares de uso libre fueron el primer paso para aunar el conocimiento, los softwares de propietarios también han resultado ser muy útiles y competitivos, como se valoró en el apartado anterior, por lo que decantarse por un programa u otro, será como elegir entre dos ordenadores con los mismos componentes de marcas distintas, por lo que las propias necesidades y gustos del investigador marcarán la decisión.

De cara al futuro, estas herramientas continuarán con su adaptación constante al ambiente cambiante del análisis de imagen, convirtiéndose en herramientas aún más poderosas que harán de puente para conseguir descubrimientos que beneficien a la comunidad científica.

6. Referencias bibliográficas

Arshadi, C., Günther, U., Eddison, M., Harrington, K. I. S., & Ferreira, T. A. (2021). SNT: a unifying toolbox for quantification of neuronal anatomy. *Nature methods*, 18(4), 374–377. <https://doi.org/10.1038/s41592-021-01105-7>

Campa Fernández, V. M. (2016). *Análisis de imágenes de microscopía con ImageJ & Fiji*. Universidad Nacional Autónoma de México.

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Guertin, D. A., Chang, J. H., Lindquist, R. A., Moffat, J., Golland, P., & Sabatini, D. M. (2006). CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10), R100. <https://doi.org/10.1186/gb-2006-7-10-r100>

Carpenter, A. E., Kamentsky, L., & Eliceiri, K. W. (2012). A call for bioimaging software usability. *Nature methods*, 9(7), 666–670. <https://doi.org/10.1038/nmeth.2073>

CellProfiler. (2023). cellprofiler.org. Recuperado en 2023, de <https://cellprofiler.org/>

Dobson, E. T. A., Cimini, B., Klemm, A. H., Wählby, C., Carpenter, A. E., & Eliceiri, K. W. (2021). ImageJ and CellProfiler: Complements in Open-Source Bioimage Analysis. *Current protocols*, 1(5), e89. <https://doi.org/10.1002/cpz1.89>

Eliceiri, K. W., Berthold, M. R., Goldberg, I. G., Ibáñez, L., Manjunath, B. S., Martone, M. E., Murphy, R. F., Peng, H., Plant, A. L., Roysam, B., Stuurman, N., Swedlow, J. R., Tomancak, P., & Carpenter, A. E. (2012). Biological imaging software tools. *Nature methods*, 9(7), 697–710. <https://doi.org/10.1038/nmeth.2084>

González Bardeci, N., Angiolini, J. F., De Rossi, M. C., Bruno, L., & Levi, V. (2017). Dynamics of intracellular processes in live-cell systems unveiled by fluorescence correlation microscopy. *IUBMB life*, 69(1), 8–15. <https://doi.org/10.1002/iub.1589>

Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing (4th ed.)*. Pearson.

Haase, R., Fazeli, E., Legland, D., Doube, M., Culley, S., Belevich, I., Jokitalo, E., Schorb, M., Klemm, A., & Tischer, C. (2022). A Hitchhiker's guide through the bio-image analysis software universe. *FEBS letters*, 596(19), 2472–2485. <https://doi.org/10.1002/1873-3468.14451>

Hogg, J. (1883). *The Microscope. Its History, Construction, and Application: Being a Familiar Introduction to the Use of the Instrument and the Study of Microscopical Science*.

Huygens software. (2023). Scientific Volume Imaging. Recuperado en 2023, de <https://svi.nl/Huygens-Software>

ImageJ Wiki. (2023). ImageJ Wiki. Recuperado en 2023, de <https://imagej.net/>

- Jiang, K., Sun, S., Zhang, L., Lu, Y., Wu, A., Cai, C., & Lin, H. (2015). Red, green, and blue luminescence by carbon dots: full-color emission tuning and multicolor cellular imaging. *Angewandte Chemie (International ed. in English)*, 54(18), 5360–5363. <https://doi.org/10.1002/anie.201501193>
- Laghi, V., Ricci, V., De Santa, F., & Torcinaro, A. (2022). A User-Friendly Approach for Routine Histopathological and Morphometric Analysis of Skeletal Muscle Using CellProfiler Software. *Diagnostics (Basel, Switzerland)*, 12(3), 561. <https://doi.org/10.3390/diagnostics12030561>
- Lamprecht, M. R., Sabatini, D. M., & Carpenter, A. E. (2007). CellProfiler: free, versatile software for automated biological image analysis. *BioTechniques*, 42(1), 71–75. <https://doi.org/10.2144/000112257>
- Lane N. (2015). The unseen world: reflections on Leeuwenhoek (1677) 'Concerning little animals'. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 370(1666), 20140344. <https://doi.org/10.1098/rstb.2014.0344>
- Lein, E. S., Hawrylycz, M. J., Ao, N., Ayres, M., Bensinger, A., Bernard, A., Boe, A. F., Boguski, M. S., Brockway, K. S., Byrnes, E. J., Chen, L., Chen, L., Chen, T. M., Chin, M. C., Chong, J., Crook, B. E., Czaplinska, A., Dang, C. N., Datta, S., Dee, N. R., ... Jones, A. R. (2007). Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124), 168–176. <https://doi.org/10.1038/nature05453>
- Mansfield, P., & Pykett, I. L. (1978). Biological and medical imaging by NMR. *Journal of Magnetic Resonance*, 29(2), 355–373. [https://doi.org/10.1016/0022-2364\(78\)90159-2](https://doi.org/10.1016/0022-2364(78)90159-2)
- NIH Image. (2023). Nih.gov. Recuperado de 2023, de <https://imagej.nih.gov/nih-image/>
- Peng H. (2008). Bioimage informatics: a new area of engineering biology. *Bioinformatics (Oxford, England)*, 24(17), 1827–1836. <https://doi.org/10.1093/bioinformatics/btn346>
- Peng, H., Bria, A., Zhou, Z., Iannello, G., & Long, F. (2014). Extensible visualization and analysis for multidimensional images using Vaa3D. *Nature protocols*, 9(1), 193–208. <https://doi.org/10.1038/nprot.2014.011>
- Portiansky, E. L. (2018). *Análisis multidimensional de imágenes digitales. 2a edición*. Facultad de Ciencias Veterinarias (UNLP).
- Rabbi, F., Dabbagh, S. R., Angin, P., Yetisen, A. K., & Tasoglu, S. (2022). Deep Learning-Enabled Technologies for Bioimage Analysis. *Micromachines*, 13(2), 260. <https://doi.org/10.3390/mi13020260>
- Real Academia Española. (s.f.). Software. En *Diccionario de la lengua española*. Recuperado en 2023, de <https://dle.rae.es/software>
- Roeder, A. H., Cunha, A., Burl, M. C., & Meyerowitz, E. M. (2012). A computational image analysis glossary for biologists. *Development (Cambridge, England)*, 139(17), 3071–3080. <https://doi.org/10.1242/dev.076414>

Rueden, C. T., Ackerman, J., Arena, E. T., Eglinger, J., Cimini, B. A., Goodman, A., Carpenter, A. E., & Eliceiri, K. W. (2019). Scientific Community Image Forum: A discussion forum for scientific image software. *PLoS biology*, 17(6), e3000340. <https://doi.org/10.1371/journal.pbio.3000340>

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J. Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7), 676–682. <https://doi.org/10.1038/nmeth.2019>

Schmid, B., Schindelin, J., Cardona, A., Longair, M., & Heisenberg, M. (2010). A high-level 3D visualization API for Java and ImageJ. *BMC bioinformatics*, 11, 274. <https://doi.org/10.1186/1471-2105-11-274>

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature methods*, 9(7), 671–675. <https://doi.org/10.1038/nmeth.2089>

Schroeder, A. B., Dobson, E. T. A., Rueden, C. T., Tomancak, P., Jug, F., & Eliceiri, K. W. (2021). The ImageJ ecosystem: Open-source software for image visualization, processing, and analysis. *Protein science : a publication of the Protein Society*, 30(1), 234–249. <https://doi.org/10.1002/pro.3993>

Software de microscopía ZEN. (2023). Zeiss.com. Recuperado en 2023, de <https://www.zeiss.com/microscopy/es/productos/software/zeiss-zen.html>

Solomon R. W. (2009). Free and open source software for the manipulation of digital images. *AJR. American journal of roentgenology*, 192(6), W330–W334. <https://doi.org/10.2214/AJR.08.2190>

Stirling, D. R., Swain-Bowden, M. J., Lucas, A. M., Carpenter, A. E., Cimini, B. A., & Goodman, A. (2021). CellProfiler 4: improvements in speed, utility and usability. *BMC bioinformatics*, 22(1), 433. <https://doi.org/10.1186/s12859-021-04344-9>

Swedlow, J. R., Goldberg, I. G., Eliceiri, K. W., & OME Consortium (2009). Bioimage informatics for experimental biology. *Annual review of biophysics*, 38, 327–346. <https://doi.org/10.1146/annurev.biophys.050708.133641>

Tinevez, J. Y., Perry, N., Schindelin, J., Hoopes, G. M., Reynolds, G. D., Laplantine, E., Bednarek, S. Y., Shorte, S. L., & Eliceiri, K. W. (2017). TrackMate: An open and extensible platform for single-particle tracking. *Methods (San Diego, Calif.)*, 115, 80–90. <https://doi.org/10.1016/j.ymeth.2016.09.016>

Tsai, C. L., Lister, J. P., Bjornsson, C. S., Smith, K., Shain, W., Barnes, C. A., & Roysam, B. (2011). Robust, globally consistent and fully automatic multi-image registration and montage synthesis for 3-D multi-channel images. *Journal of microscopy*, 243(2), 154–171. <https://doi.org/10.1111/j.1365-2818.2011.03489.x>

Volocity software. (2023). Volocity Software. Recuperado en 2023, de <https://www.volocity4d.com/>