

# Interfaz de CellProfiler con otras herramientas de software mediante archivos y complementos

Beth Cimini y Erin Weisbart\ Plataforma de imágenes, Instituto Broad del MIT y Harvard, Cambridge, MA.

## Preparación

1. Descargue e instale [CellProfiler](#).
2. Descargue e instale [Docker Desktop](#) (recomendado para usuarios con menos experiencia computacional) o [Podman Desktop](#) (una alternativa a Docker para usuarios con mayor experiencia computacional).
3. Descargue al menos un Docker de RunCellpose y un Docker de ilastik desde Dockerhub. Esto se realizará automáticamente la primera vez que llame a un Docker desde CellProfiler (es decir, ejecute un CellProfiler pipeline que use el Docker). Sin embargo, si está ejecutando este tutorial en un taller, le recomendamos encarecidamente que lo descargue con antelación, ya que se trata de archivos grandes (5-10 GB) y el ancho de banda suele ser limitado en un taller. En Docker Desktop o Podman Desktop, puede buscar contenedores en la barra de búsqueda superior (ver más abajo). Asegúrese de seleccionar una etiqueta (versión) compatible con el complemento que esté utilizando y luego seleccione «Extraer». Recomendamos `biocontainers/ilastik:1.4.0_cv2` para ilastik y `cellprofiler/runcellpose_with_pretrained:3.1.2.2` para Cellpose.
4. Descarga las imágenes de ejemplo del Tutorial de Segmentación para Principiantes. [Haz clic aquí para descargar](#). (Elimina o ignora las canalizaciones; solo usaremos las imágenes).
5. Descarga las canalizaciones y los materiales para este tutorial. [Haz clic aquí para descargar](#).
6. (Opcional pero recomendado) Descargue e instale [ilastik](#).
7. (Opcional, pero recomendado) Clona (es decir, descarga el repositorio completo) de complementos de Cellprofiler. En tu terminal, escribe `git clone https://github.com/CellProfiler/CellProfiler-plugins.git`. Como alternativa, descarga solo el complemento Runilastik y el complemento RunCellpose seleccionando el botón «Descargar archivo sin procesar».



Busque en Docker Desktop el contenedor deseado

## Información de fondo

### Información general

Este ejercicio pretende ampliar y desarrollar el ejercicio de Segmentación para Principiantes, disponible en la página de tutoriales de CellProfiler ([tutorials.cellprofiler.org](http://tutorials.cellprofiler.org)). Consulte dicho tutorial para obtener información general sobre cómo configurar CellProfiler como pocillo, así como información sobre las imágenes. Se asumirá que comprende los módulos que se tratan en dicho tutorial, incluyendo la entrada, la creación de objetos, las superposiciones y el guardado.

### ¿En qué consiste este ejercicio?

Si bien CellProfiler es una herramienta de análisis de imágenes, también funciona como un gestor de flujo de trabajo que puede interactuar con otras herramientas, intercambiando datos automáticamente para que pueda usarlas eficazmente dentro de un mismo CellProfiler pipeline. Esto significa que, en un hipotético pipeline, CellProfiler debe ejecutar los pasos 1 y 2, la Herramienta A el 3, la Herramienta B el 4 y CellProfiler el 5, todo ello mientras configura las Herramientas A y B dentro de CellProfiler y sin tener que gestionar la importación o exportación de datos desde otras herramientas. En este tutorial, explorará tres maneras diferentes de acceder al trabajo realizado en otras herramientas:

1. Cargando máscaras creadas por otras herramientas de segmentación (en este caso, Cellpose, pero la misma estrategia funciona para muchas herramientas que usan este formato).
2. Accediendo a ilastik para utilizar un modelo de clasificación de píxeles entrenado a través del sistema de complementos de CellProfiler.
3. Ejecutando Cellpose en CellProfiler a través del sistema de complementos de CellProfiler y un contenedor Docker.

## Complementos (*plugins*)

CellProfiler cuenta con varios complementos que actúan como módulos dentro de un pipeline. Los complementos están disponibles en Github (<https://github.com/CellProfiler/CellProfiler-plugins>). Para más información, consulte la documentación de los complementos de CellProfiler ([plugins.cellprofiler.org](http://plugins.cellprofiler.org)).

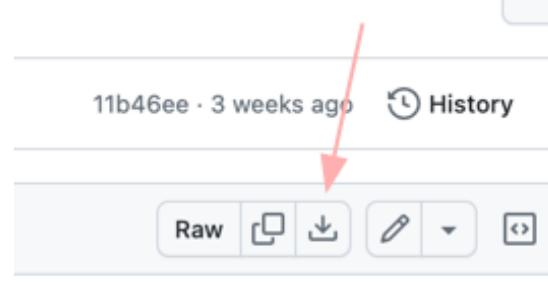
Para citar el campo:

Los complementos mejoran las capacidades de CellProfiler pero no son mantenidos oficialmente de la misma manera que los módulos. Un módulo puede estar en CellProfiler-plugins en lugar del propio CellProfiler porque:

- está en desarrollo activo
- tiene una audiencia de nicho
- no está documentado según los estándares de CellProfiler
- Sólo funciona con ciertas versiones de CellProfiler
- requiere bibliotecas adicionales u otras dependencias que no podemos o no queremos requerir para CellProfiler
- ha sido aportado por un miembro de la comunidad

### Truco:

Si bien esa documentación incluye instrucciones para instalar plugins, en el paso 2 se sugiere descargar todos los plugins de CellProfiler. Esto es recomendable, pero puedes descargar plugins individuales desde GitHub con el botón del sitio web a continuación: pocillo o usando el botón «Descargar archivos RAW» de GitHub.



Botón «Descargar archivos sin formato» de GitHub

### Truco:

Recomendamos fuertemente crear una carpeta dedicada para almacenar tus complementos de CellProfiler, ya que la carga puede ser lenta si hay muchos otros archivos en la misma ubicación (por ejemplo, si se encuentran en la carpeta Descargas).

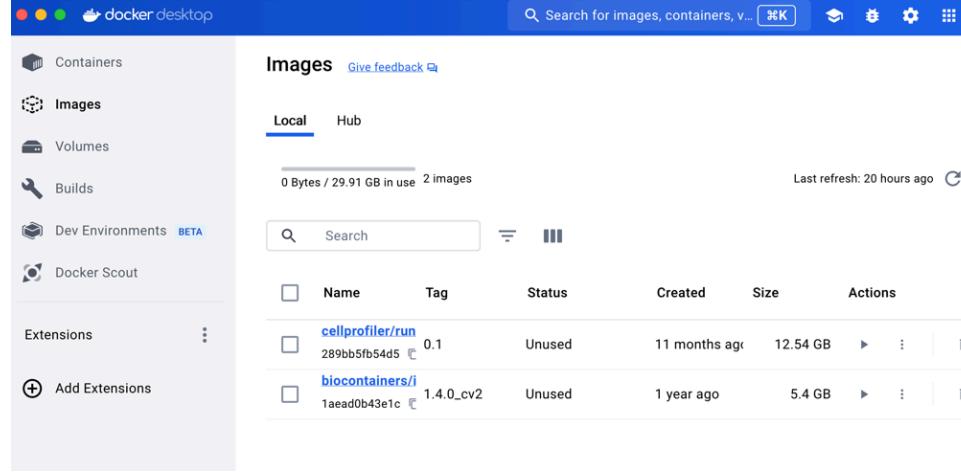
## Docker Desktop

En nuestro tutorial, queremos acceder a software que no podemos o no queremos instalar en nuestro ordenador local. (Las herramientas de aprendizaje profundo como Cellpose suelen ser muy difíciles de instalar y es posible que ilastik no funcione correctamente con la configuración de multiprocesamiento de otras herramientas). Para solucionar estos problemas, utilizamos **contenedores**. Los informáticos suelen utilizar **contenedores** de software para enviar herramientas o datos difíciles de instalar. Aquí hay una buena introducción y descripción general de los contenedores para quienes no son informáticos. Los contenedores son como un sistema operativo preconfigurado en una caja. Dado que vienen preconfigurados, la instalación de cualquier software se realiza una sola vez (por el creador del contenedor, no por ti) y debería funcionar durante mucho tiempo. (Por ejemplo, no tienes que preocuparte de que una actualización de tu por-

tátil dañe algún software antiguo que tengas instalado). Grupos como [biocontainers](#) ya han contenedorizado muchas de las herramientas que conoces y te encantan. Existen varios tipos de contenedores de software, pero uno de los más comunes se llama contenedor **Docker**.

La mayoría de los biólogos desconocen o no usan contenedores, sobre todo porque su uso habitual implica acceder a ellos mediante la terminal. ¡Pero puedes usar Docker sin usar una terminal! CellProfiler tiene plugins configurados específicamente para llamar a otras herramientas que residen (y se ejecutan) dentro de contenedores Docker.

Para ello, CellProfiler necesita tener la infraestructura para contenedores Docker en funcionamiento. Esto se puede conseguir instalando un programa llamado **Docker Desktop**. No es necesario crear una cuenta para usarlo, pero probablemente deba reiniciar el equipo una vez finalizada la instalación. [Podman Desktop](#) es una alternativa a Docker que CellProfiler también admite para usuarios con mayor experiencia computacional. Luego, siempre que desee usar un complemento que invoque Docker en CellProfiler, simplemente asegúrese de que Docker Desktop (o Podman Desktop) esté abierto y en ejecución, y CellProfiler se encargará de todo lo demás.



La interfaz de Docker Desktop para Mac. Los contenedores se pueden buscar en la barra de búsqueda superior.

### Advertencia:

Siempre que instales un programa que pueda ejecutar otros programas, por supuesto debes tener cuidado. Los contenedores Docker te brindan acceso a una gran variedad de herramientas que de otro modo podrían ser difíciles de instalar (o de instalar juntas), pero ten cuidado de ejecutar sólo contenedores que reconozcas o en los que confíes.

## Ejercicio 1: Importar máscaras desde otra herramienta de segmentación

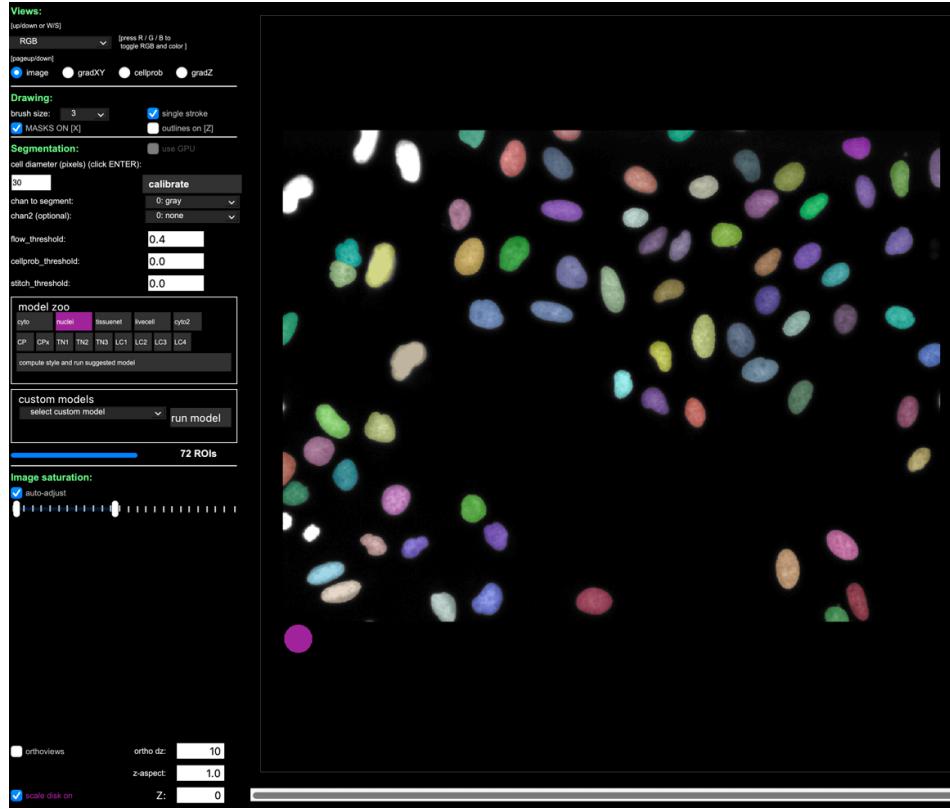
En este ejercicio, cargaremos máscaras marca (a veces también llamadas matrices marca) creadas en Cellpose. En resumen, una máscara marca es una imagen en la que todos los píxeles de fondo tienen un valor de 0, todos los píxeles del objeto 1 tienen un valor de 1, todos los píxeles del objeto 2 tienen un valor de 2, etc. Es un formato común en muchas herramientas de segmentación de objetos (aunque no es adecuado para aplicaciones con objetos superpuestos). Dado que muchas herramientas (incluido CellProfiler) pueden crear estas máscaras, CellProfiler tiene la capacidad de leerlas y detectar automáticamente los objetos que contienen.

Hemos utilizado Cellpose 2.2.2 para generar máscaras nucleares a partir de las imágenes del canal de ADN y máscaras de células a partir del canal ActinGolgi.

### Solo si tienes curiosidad: ¿cómo hicimos estas máscaras etiquetadas?

Cellpose se instaló en un entorno de conda de acuerdo con las [instrucciones oficiales](#) para la instalación con la interfaz gráfica (GUI). El software se inició en ese entorno conda usando el comando «cellpose».

Las máscaras nucleares se crearon arrastrando cada imagen de ADN a la ventana de Cellpose, seleccionando el modelo “nuclei” con la configuración predeterminada y luego guardándolas individualmente como PNG usando *Cmd+N* (Mac) *Ctrl+N* (Windows). Como solo había 10 imágenes y las teclas de acceso rápido estaban disponibles, esto no llevó demasiado tiempo.



Núcleos segmentados en la interfaz gráfica de Cellpose

Las máscaras celulares se crearon navegando a la carpeta de imágenes y luego ejecutando el comando de abajo; esto requiere cierto conocimiento de la [sintaxis de la línea de comandos de Cellpose](#) pero es mucho más fácil para procesar una gran cantidad de imágenes.

```
python -m pose celular --dir . --img_filter Ch4 --pretrained_model cyto2 --diameter 40 --save_png --verbose
```

```
(cellpose) bcimini@wm4f8-761 cellpose_masks_cells % python -m cellpose --dir . --pretrained_model cyto2 --diameter 40 --save_png --verbose
2024-04-30 10:59:04,196 [INFO] WRITING LOG OUTPUT TO /Users/bcimini/.cellpose/run.log
2024-04-30 10:59:04,196 [INFO]
cellpose version: 2.2.2
platform: darwin
python version: 3.9.16
torch version: 2.6.1
2024-04-30 10:59:04,196 [INFO] >>> using CPU
2024-04-30 10:59:04,196 [INFO] >>> running cellpose on 10 images using chan_to_seg GRAY and chan (opt) NONE
2024-04-30 10:59:04,196 [INFO] >>> using CPU
2024-04-30 10:59:04,196 [INFO] >> cyto2 <> model set to be used
2024-04-30 10:59:04,196 [INFO] Downloading: "https://www.cellpose.org/models/cyto2torch_1" to /Users/bcimini/.cellpose/models/cyto2torch_1
100%|██████████| 10/10 [00:00:00.00, 10.00s/it]
2024-04-30 10:59:37,787 [INFO] Downloading: "https://www.cellpose.org/models/cyto2torch_2" to /Users/bcimini/.cellpose/models/cyto2torch_2
100%|██████████| 10/10 [00:00:00.00, 10.00s/it]
2024-04-30 10:59:53,084 [INFO] Downloading: "https://www.cellpose.org/models/cyto2torch_3" to /Users/bcimini/.cellpose/models/cyto2torch_3
100%|██████████| 10/10 [00:00:00.00, 10.00s/it]
2024-04-30 11:00:04,675 [INFO] WARNING: MKL version on torch not working/installation - CPU version will be slightly slower.
2024-04-30 11:00:04,675 [INFO] see https://pytorch.org/docs/stable/backends.html?highlight=mkl
2024-04-30 11:00:04,877 [INFO] >>> model diam_mean = 30.000 (ROIs rescaled to this size during training)
2024-04-30 11:00:04,881 [INFO] Downloading: "https://www.cellpose.org/models/size_cyto2torch_0.npy" to /Users/bcimini/.cellpose/models/size_cyto2torch_0
100%|██████████| 10/10 [00:00:00.00, 10.00s/it]
2024-04-30 11:00:05,073 [INFO] >>> using diameter 40.000 for all images
2024-04-30 11:00:05,074 [INFO] 0%| | 0/10 [00:00<0, 0it/s]
2024-04-30 11:00:05,078 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:00:13,672 [INFO] >>> TOTAL TIME 8.59 sec
2024-04-30 11:00:13,719 [INFO] 100%# | 1/10 [00:08<01:17, 8.64s/it]
2024-04-30 11:00:13,731 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:00:23,115 [INFO] >>> TOTAL TIME 9.38 sec
2024-04-30 11:00:23,152 [INFO] 20%## | 2/10 [00:18<01:12, 9.11s/it]
2024-04-30 11:00:23,156 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:00:31,916 [INFO] >>> TOTAL TIME 8.76 sec
2024-04-30 11:00:31,940 [INFO] 30%### | 3/10 [00:26<01:02, 8.96s/it]
2024-04-30 11:00:31,945 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:00:39,665 [INFO] >>> TOTAL TIME 7.72 sec
2024-04-30 11:00:39,727 [INFO] 40%### | 4/10 [00:34<00:50, 8.50s/it]
2024-04-30 11:00:39,728 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:00:48,511 [INFO] >>> TOTAL TIME 8.78 sec
2024-04-30 11:00:48,541 [INFO] 50%#### | 5/10 [00:43<00:43, 8.61s/it]
2024-04-30 11:00:48,544 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:00:57,089 [INFO] >>> TOTAL TIME 8.54 sec
2024-04-30 11:00:57,122 [INFO] 60%##### | 6/10 [00:52<00:34, 8.60s/it]
2024-04-30 11:00:57,125 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:01:05,582 [INFO] >>> TOTAL TIME 8.38 sec
2024-04-30 11:01:05,609 [INFO] 70%##### | 7/10 [01:00<00:25, 8.54s/it]
2024-04-30 11:01:05,642 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:01:13,789 [INFO] >>> TOTAL TIME 8.25 sec
2024-04-30 11:01:13,826 [INFO] 80%##### | 8/10 [01:08<00:16, 8.46s/it]
2024-04-30 11:01:13,829 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:01:22,739 [INFO] >>> TOTAL TIME 8.91 sec
2024-04-30 11:01:22,802 [INFO] 90%##### | 9/10 [01:17<00:08, 8.62s/it]
2024-04-30 11:01:22,813 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:01:35,845 [INFO] >>> TOTAL TIME 13.03 sec
2024-04-30 11:01:35,897 [INFO] 100%##### | 10/10 [01:39<00:00, 10.00s/it]
2024-04-30 11:01:35,897 [INFO] 100%##### | 10/10 [01:39<00:00, 9.08s/it]
2024-04-30 11:01:35,897 [INFO] >>> completed in 151.701 sec
(cellpose) bcimini@wm4f8-761 cellpose_masks_cells %
```

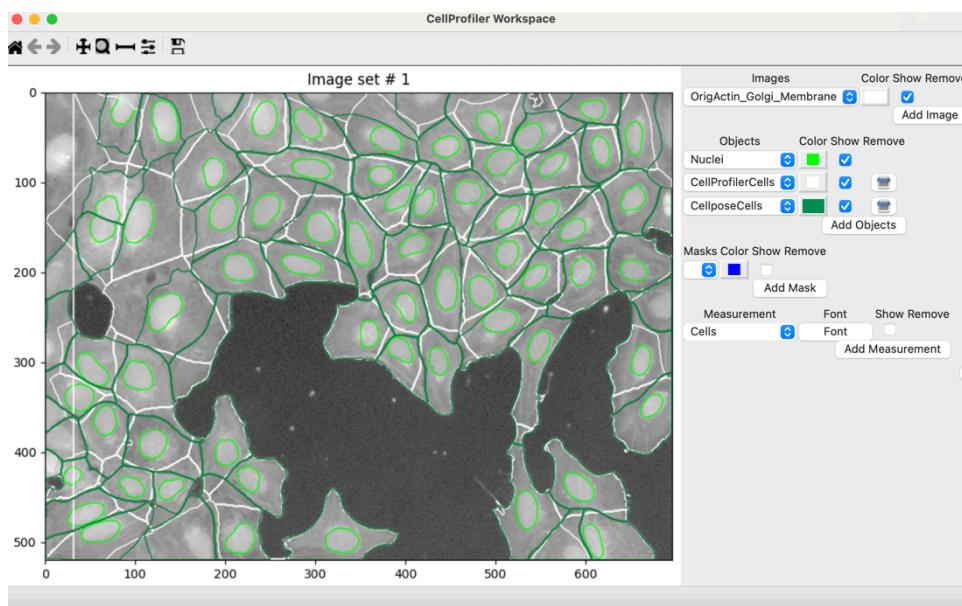
Ejecución de Cellpose desde la línea de comando

## Cargando las máscaras etiquetadas en CellProfiler

1. Abre una copia limpia de CellProfiler (o ejecuta File -> New Project) y arrastra `bonus_1_import_masks.cppipe` al panel de flujo de trabajo.
2. Arrastra la subcarpeta `images_Illum-corrected` del ejercicio principal de Segmentación para Principiantes y también la subcarpeta `cellpose_masks_cells` de este ejercicio. **No arrastres a la carpeta `cellpose_masks_nuclei`.**
3. Ponga CellProfiler en modo de prueba , abra el icono del ojo junto a OverlayOutlines y, a continuación, pulse el botón de paso tres veces para crear una segmentación clásica y compararla con la versión generada por Cellpose. Puede consultar la configuración en el módulo OverlayOutlines para ver qué color de contorno corresponde a cada segmentación en el resultado.
4. Opcionalmente, abra el Visor del espacio de trabajo usando el botón Ver espacio de trabajo para crear fácilmente superposiciones personalizables sobre la marcha.
5. Presione el botón Siguiente conjunto de imágenes y repita un par de veces para examinar las segmentaciones en más imágenes.

### Una pregunta para ti:

¿Hay casos en los que crees que la segmentación clásica suele funcionar mejor? ¿Dónde funciona mejor Cellpose?



Visualización de imágenes y objetos en el Workspace Viewer.

## Haz ingeniería reversa cómo funcionó esto

1. Abre el módulo NamesAndTypes.
2. Desplázate hacia abajo en NamesAndTypes para encontrar la entrada que agrega las máscaras. ¿Observas alguna configuración que sea diferente a la de las otras imágenes?
3. Busca la entrada en NamesAndTypes que tiene 2 reglas que la imagen debe pasar, no solo una. ¿Entiendes por qué?

Para créditos extra: agrega también las segmentaciones nucleares proporcionadas.

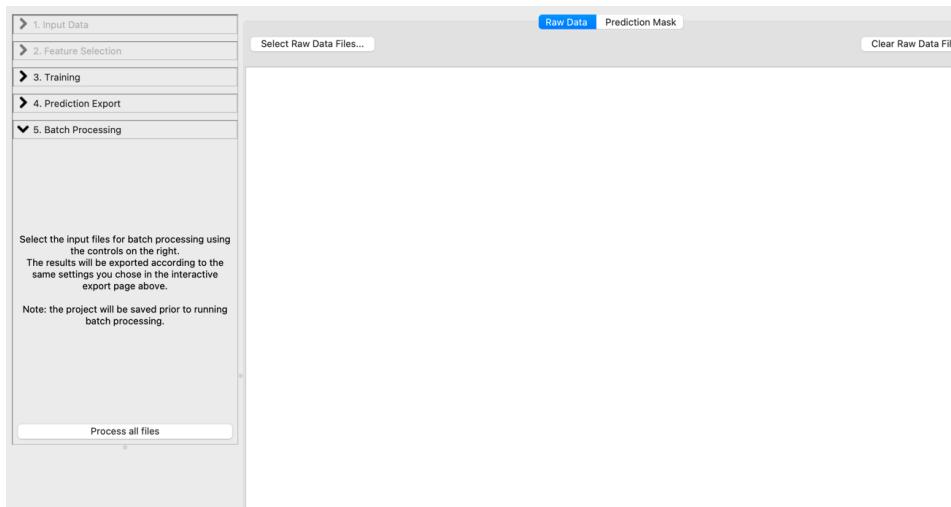
1. Arrastra y suelta la carpeta `cellpose_masks_nuclei` en el módulo Images para agregar las máscaras nucleares a la lista de imágenes.
2. Ve al módulo NamesAndTypes y configúralo para que se puedan cargar ambos tipos de máscaras (pista: hay un botón *Duplicate* que puedes usar para hacer una segunda copia de un canal).
- Probablemente tendrás que modificar la configuración existente para cargar las máscaras celulares cambiando su regla o agregando una segunda regla. ¿Entiendes cómo/por qué?

## Ejercicio 2: Ejecutar ilastik localmente o desde un contenedor Docker

La segmentación clásica requiere que el elemento que te interesa en una imagen sea brillante y que todos los demás píxeles sean más oscuros. Si tienes una señal fluorescente clara y nítida para el elemento que te interesa, ¡genial! Si no, quizás tengas que recurrir a algunos trucos o usar otra herramienta, como un clasificador de píxeles.

Un clasificador de píxeles se entrena píxel por píxel para indicar: «Esta es la probabilidad que creo que tiene este píxel en tu segmentación». Este clasificador crea para cada píxel un *valor de probabilidad* que corresponde a la probabilidad de segmentarlo. Si tu clasificador es bueno, obtendrás una imagen donde los píxeles que te interesan tienen alta probabilidad (brillantes) y el resto tiene baja probabilidad (oscuros). ¡Eso es justo lo que buscamos! Los biólogos suelen llamar a esto «**Clasificación de Píxeles**»; los informáticos a veces lo llaman «**Segmentación Semántica**».

Nos gusta usar ilastik para la clasificación de píxeles porque facilita la automatización de la creación de un clasificador a partir de un número muy reducido de imágenes y su posterior aplicación masiva a muchas otras en modo de procesamiento por lotes. Puedes consultar un tutorial que hemos escrito para ejecutar ilastik y, posteriormente, CellProfiler en [tutorials.cellprofiler.org](http://tutorials.cellprofiler.org) (busca «Clasificación basada en píxeles»). Fiji cuenta con algunos plugins populares para la clasificación de píxeles, como Weka Trainable Segmentation y Labkit, y sin duda deberías usarlos si te funcionan mejor.



### Modo de procesamiento en lotes de ilastik

¿Por qué hacer dos pasos cuando puedes hacer uno? En este tutorial, tomaremos un clasificador ilastik previamente entrenado y lo ejecutaremos dentro de nuestro pipeline de CellProfiler, para poder encontrar y medir objetos, todo en un solo paso.

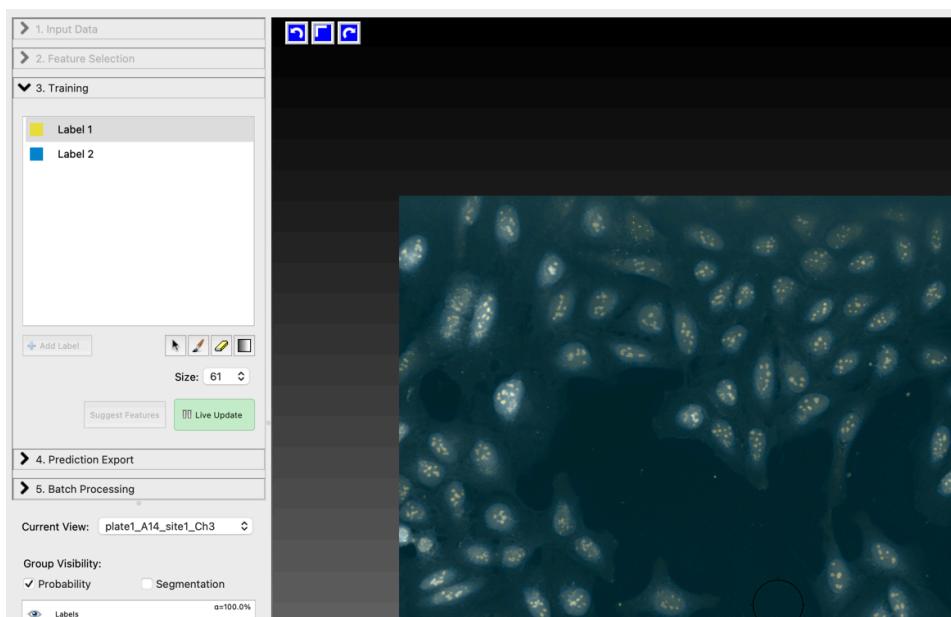
Necesitará tener ilastik o Docker Desktop (o Podman Desktop) instalado en su computadora para este ejercicio. Si usa ilastik, asegúrese de que esté CERRADO. Si usa Docker Desktop (o Podman Desktop), asegúrese de que esté ABIERTO. Recomendamos instalar ilastik, ya que es una herramienta útil y le permitirá realizar el desafío adicional de este ejercicio.

#### Advertencia:

Si estás en Windows, ejecuta el módulo `RunIlastik` en modo Local (trabajando en una copia instalada de ilastik, en lugar de una copia dentro de un archivo Docker). Este ejercicio funcionará en Modo de Prueba, pero no se ejecutará en el Modo de Análisis; estamos trabajando con los desarrolladores de ilastik para determinar por qué ocurre esto. Sin embargo, alcanza para el propósito de este ejercicio. Si tienes muchos datos propios que deseas ejecutar más adelante, aún puede usar ilastik en un proceso de dos pasos y/o usar ilastik Dockerizado.

### Sólo si tienes curiosidad: ¿cómo entrenamos al clasificador?

Este clasificador se creó en ilastik 1.4.1b5 mediante entrenamiento con 4 imágenes (A14\_sitio1, E18\_sitio1, D16\_sitio1 y C12\_sitio1), en las que etiquetamos solo dos clases: una para los核 (en amarillo en la figura inferior) y otra para cada parte de la imagen (en azul en la imagen inferior). Se podrían haber creado más clases, pero esto funcionó para nuestros propósitos.



Al usar ilastik para microscopía de fluorescencia, probablemente obtendrá el mejor rendimiento si **mantiene sus anotaciones extremadamente mínimas**: el clasificador que va a usar se entrenó utilizando 31 píxeles totales de anotación en las 4 imágenes (13 píxeles de anotación dentro de los nucléolos y 18 píxeles fuera de ellos). Ninguna imagen tenía más de 14 píxeles anotados en total. ¡Le recomendamos encarecidamente que haga sus clasificadores un píxel a la vez, haciendo anotaciones de puntos! ¡Resista la tentación de dibujar líneas onduladas por toda la imagen! Esto parece contradictorio, pero le prometemos que es cierto. Obtenga más información sobre el uso de ilastik en el podcast de video Ask Erin, Dear Beth [Introducción a la clasificación de píxeles en ilastik - Episodio 11](#) y [Cómo entrenar un clasificador de píxeles - Episodio 13](#).

## Habilitar el complemento Runilastik

1. Descarga el plugin Runilastik ([https://github.com/CellProfiler/CellProfiler-plugins/blob/master/active\\_plugins/runilastik.py](https://github.com/CellProfiler/CellProfiler-plugins/blob/master/active_plugins/runilastik.py)) en una carpeta de tu ordenador. Como se mencionó anteriormente, te recomendamos usar una carpeta que contenga únicamente plugins.
2. En el menú Archivo -> Preferencias de CellProfiler (CellProfiler -> Preferencias en Mac), configura el directorio de complementos de CellProfiler (CellProfiler Plugins Directory) en la carpeta que contiene el complemento.
3. Cierre y vuelve a abrir CellProfiler para poder cargar el complemento.

## Cargar el clasificador

### Nota:

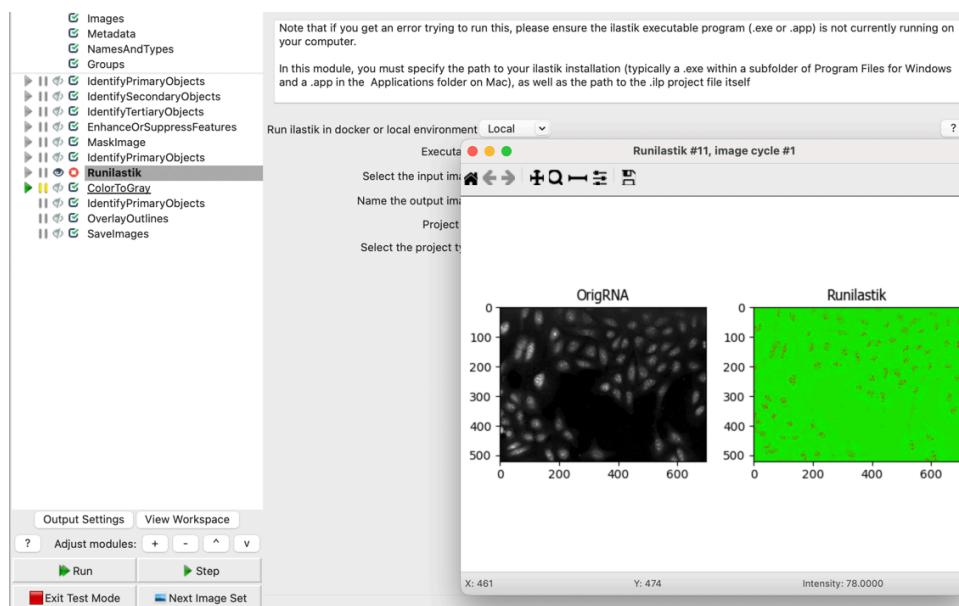
Si usa Docker, la primera vez que ejecute el módulo Runilastik, deberá descargar un archivo de aproximadamente 5 GB, lo cual puede ser lento dependiendo de su conexión a internet (a menos que lo haya descargado como preparación, como sugerimos anteriormente). Sin embargo, isolo necesita realizar este paso una vez!

1. Arrastre `bonus_2_ilastik.cppipe` al panel pipeline para cargar pipeline.
2. Arrastra la subcarpeta `images_Illum-corrected` desde el ejercicio principal al panel Imágenes.
3. Abra el módulo RunIlastik, establezca la ruta a su archivo de proyecto (`NucleoliDetection.ilp`).
4. Si tiene ilastik instalado en su equipo, puede configurar la ruta a su instalación local. De lo contrario, cambie «Runilastik en un contenedor» a «Docker».
5. Es posible que desees poner una pausa junto a SaveImages, o desmarcarla , para evitar que guarde imágenes mientras exploras este pipeline.
6. Active CellProfiler en modo de prueba pulsando el botón Iniciar modo de prueba , abra los iconos de ojo junto a Runilastik y OverlayOutlines, y luego pulse el botón Ejecutar . Puede consultar la configuración en el módulo OverlayOutlines para ver qué color de contorno corresponde a cada segmentación en la salida.

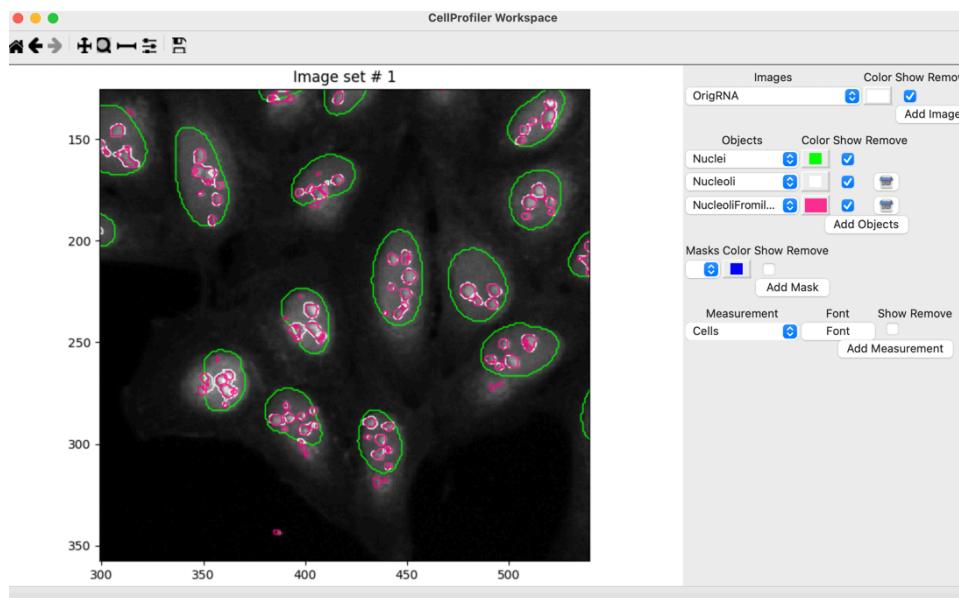
## Evaluuar el clasificador

Evalúe su predicción en Runilastik en algunos conjuntos de imágenes utilizando OverlayOutlines y/o WorkspaceViewer

- ¿Cómo funciona pocillo?
- ¿Tiene un peor rendimiento en imágenes en las que no fue entrenado?
- ¿Hay casos en los que cree que la segmentación basada en el modelo ilastik funciona mejor?
- ¿Hay casos en los que cree que la segmentación mediante filtrado y enmascaramiento funciona mejor?



Captura de pantalla del módulo Runilastik



Evaluación de segmentaciones en el WorkspaceViewer

## Bono a bono: entrena tu propio modelo ilastik

Con base en sus evaluaciones anteriores, ¿podría identificar algunos aspectos donde una capacitación adicional podría ayudar a solucionar algunos problemas del modelo ilastik? (Para ello, necesitará tener ilastik descargado en su computadora).

1. Abre el archivo **NucleoliDetection.ilp** en ilastik.
2. Ve a la pestaña «Training».
3. Activa «Live Update».
4. Elije una imagen que creas que necesita ayuda y agrega anotaciones (una cantidad muy pequeña de anotaciones de un solo píxel!). ¿Las cosas mejoraron o empeoraron?
5. Agrega algunas anotaciones muy grandes (una línea larga y ondulada, por ejemplo) a una imagen, luego cambia de imagen para demostrarle a tí mismo que las anotaciones grandes en realidad pueden dañar en algunos casos en lugar de ayudar (si deseas rescatar tu clasificador, regresa y usa la herramienta de borrador para eliminar estas anotaciones grandes!).
6. Si crees que has mejorado lo suficiente la predicción, guarda este nuevo modelo y regresa a CellProfiler. ¿Ayudó en los casos en los que pensabas que ayudaría?

# Ejercicio 3: Ejecutar Cellpose desde un contenedor Docker

RunCellpose es, sin duda, nuestro plugin más popular, simplemente porque a) Cellpose es genial y b) instalar software con `conda` cuando no se tiene mucha experiencia computacional no lo es. Se puede usar el plugin de dos maneras: con una instalación local de `conda` o `python` que contenga CellProfiler y Cellpose en el mismo entorno, o con Docker. El módulo es bastante más lento al ejecutar Cellpose en Docker, pero para muchos, es una buena compensación por el desafío y la frustración que supone la instalación.

## Iniciar el Docker Desktop

1. Si aún no has instalado Docker Desktop desde el enlace anterior, ¡hazlo ahora! Esto puede implicar reiniciar tu computadora.
2. Inicia Docker Desktop.
3. Opcional, pero muy recomendable: una vez abierto Docker Desktop, usa la función de búsqueda para buscar «Cellpose» y encontrar un modelo «cellprofiler/runCellpose\_with\_pretrained» y obténlo (si aún no lo has hecho). Si por alguna razón esto no funciona, continúa, pero te ahorrará tiempo más adelante.

## Toma el complemento RunCellpose

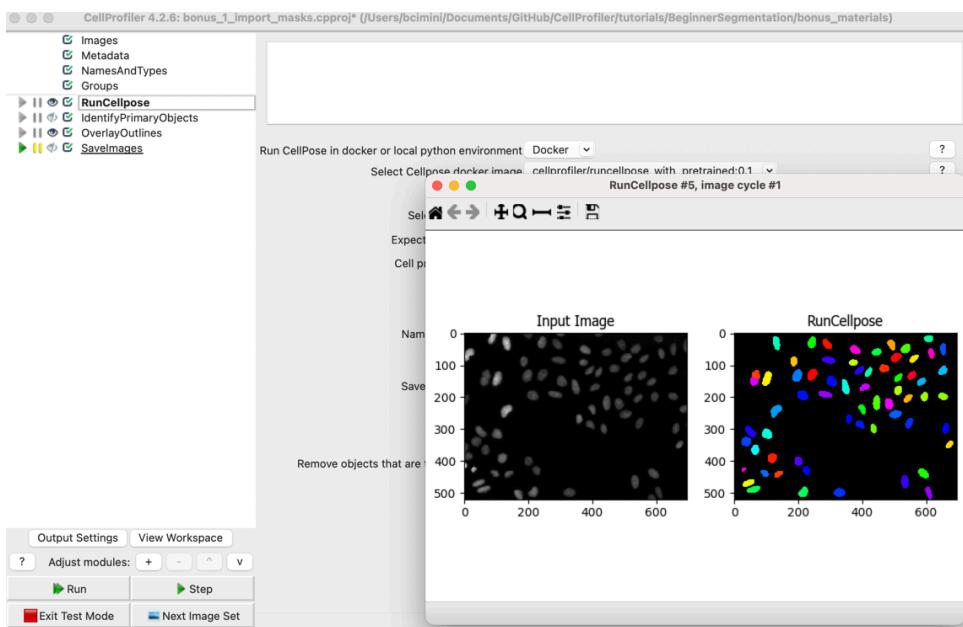
1. Descarga el plugin en una carpeta de tu ordenador. Como se mencionó anteriormente, te recomendamos usar una carpeta que contenga únicamente plugins.
2. En el menú File -> Preferences de CellProfiler, configura el Directorio de Complementos de CellProfiler (CellProfiler plugins directory) en la carpeta que contiene el complemento.
3. Cierra y vuelve a abrir CellProfiler para cargar el complemento.

## Carga pipeline y evalúa la segmentación

### Nota:

Si usa Docker/Podman, la primera vez que ejecute el módulo RunCellpose, deberá descargar un archivo de entre 5 y 10 GB (según la versión de Cellpose), lo cual puede ser lento según su conexión a internet (a menos que lo haya descargado como preparación, como sugerimos anteriormente). Sin embargo, solo necesita realizar este paso una vez!

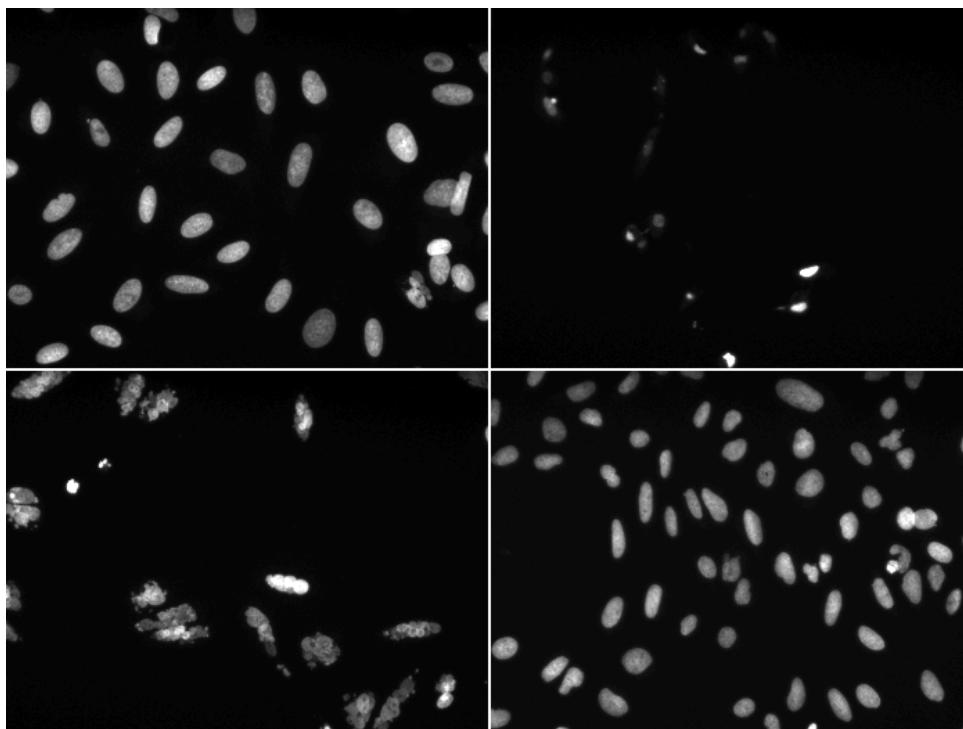
1. Arrastra `bonus_3_cellpose.cppipe` al panel de pipeline.
2. Arrastra la subcarpeta `images_Illum-corrected` desde el ejercicio principal al módulo `Images`.
3. Coloque CellProfiler en TestMode 
4. Si usa Docker/Podman y ya ha descargado previamente un contenedor, asegúrese de que la configuración «Seleccionar imagen Docker de Cellpose» en el módulo «RunCellpose» coincida con el contenedor que ya ha descargado.
5. Abra los íconos de ojos  junto a RunCellpose y OverlayOutlines, y luego presione Ejecutar 
6. Es posible que deseas poner una pausa  junto a SaveImages, o desmarcarla , para evitar que guarde imágenes mientras exploras este pipeline.
7. Como antes, utilizando `OverlayOutlines` y/o `WorkspaceViewer`, evalúa la segmentación en algunas imágenes. ¿Dónde le está yendo mejor a CellProfiler y dónde le está yendo mejor a Cellpose?
8. Usa el botón de información  para obtener más información sobre los diferentes parámetros que puedes pasar a Cellpose (no los ofrecemos todos, ipero sí muchos!). ¿Cómo afecta ajustarlos a tu resultado? ¿Qué tal si cambias el modelo que usas o la imagen que segmentas?



La salida del módulo RunCellpose

## Bonus del bonus 1: prueba con un conjunto de datos más variado

Los datos que te proporcionamos en este ejercicio estaban muy limpios y todos provenían de los pocillos de control negativo de este experimento. Esta carpeta contiene imágenes con una variedad más amplia de fenotipos. Prueba reemplazar las imágenes en el módulo «Images» con las de esta carpeta y fíjate cómo funcionan los dos tipos de segmentación en diferentes condiciones.



Una gama más amplia de fenotipos puede resultar más difícil de tener en cuenta de manera consistente al realizar la segmentación.

### Truco:

Te recomendamos explorar el rendimiento en algunas imágenes aleatorias por tu cuenta (puedes hacerlo desde el menú Test con la opción Random Image Set), pero si te das cuenta de que siempre terminas con imágenes que se parecen, puedes intentar examinar imágenes de la siguiente lis-

## Bonus del Bonus 2: mejora la segmentación de Cellpose

En función de las evaluaciones anteriores, ¿puedes identificar casos en los que entrenamiento adicional podría ayudar a solucionar algunos problemas en el modelo Cellpose? Si tienes Cellpose en tu computadora (o deseas intentar instalarlo - consulta el Ejercicio 1 para obtener un enlace a las instrucciones de instalación oficiales), abre Cellpose con algunas de las imágenes Ch1 y fíjate si puedes entrenar un modelo que funcione mejor.

## Bonus del Bonus 3: prueba tus habilidades de segmentación clásica

Los parámetros de segmentación clásicos que se proporcionan aquí se basan en el ajuste experto de una usuaria de CellProfiler con mucha experiencia. ¿Hasta dónde puede llegar? Agrega otro módulo `IdentifyPrimaryObjects` al pipeline, ajústalo tú mismo y luego observa el Workspace Viewer para ver cómo funciona, especialmente en el conjunto de imágenes más diverso de «Bonus del Bonus 1». Compara tus parámetros con los de la experta: ¿en qué se diferencian? ¿Entiendes por qué se pueden haber elegido algunas de estas configuraciones?

## ¿Qué sigue? ¿Quieres saber más sobre los plugins y módulos de CellProfiler?

1. Lea el [documento sobre los complementos de CellProfiler](#).
2. Lea la [documentación de complementos de CellProfiler] (<https://plugins.cellprofiler.org/>).
3. Mira este [video](#) para aprender a escribir un módulo.
4. Mire este [video podcast Pregúntele a Erin Dear Beth](#) para obtener más información sobre los complementos.