

Conectando CellProfiler con otras herramientas de software a través de archivos y complementos (*plugins*)

Beth Cimini

Broad Institute del MIT y Harvard, Cambridge, MA.

Información de contexto

Información general

Este ejercicio está orientado a ampliar y desarrollar el ejercicio de Segmentación para Principiantes disponible en [la página de tutoriales de CellProfiler] (tutorials.cellprofiler.org). Consulta ese tutorial para obtener información general sobre cómo configurar CellProfiler así como también información sobre las imágenes. Generalmente se asumirá que comprendes los módulos cubiertos en ese tutorial, incluidos los módulos de entrada, la creación de objetos, la superposición de objetos sobre imágenes y el guardado.

¿En qué consiste este ejercicio?

Hay una serie de excelentes herramientas de software disponibles para los científicos que desean analizar imágenes en estos días: [isolo](#) [forum.image.sc](#) tiene más de 60 herramientas de código abierto! A veces, sin embargo, ayuda tener un flujo de trabajo de múltiples herramientas: realizar un paso en la Herramienta A y luego otro en la Herramienta B (y luego posiblemente C, D, etc., ipero con suerte no!). En este tutorial, probarás 3 formas diferentes de acceder al trabajo que realizaste con 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*)

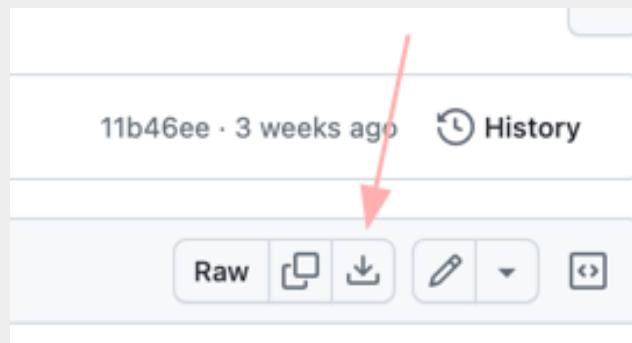
Si bien CellProfiler no tiene tantos complementos como, por ejemplo, Fiji, itiene muchos que puedes probar! Puede visitar plugins.cellprofiler.org para obtener más información. Citando ese sitio:

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 cierta versión 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 tiene instrucciones para [instalar complementos] (https://plugins.cellprofiler.org/using_plugins.html#installing-plugins-without-dependencies), en el paso 2 sugiere descargar todos los complementos de CellProfiler; Esto no es malo, pero también (o en lugar de ello) puedes descargar complementos individuales desde GitHub con el botón del sitio web a continuación.



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).

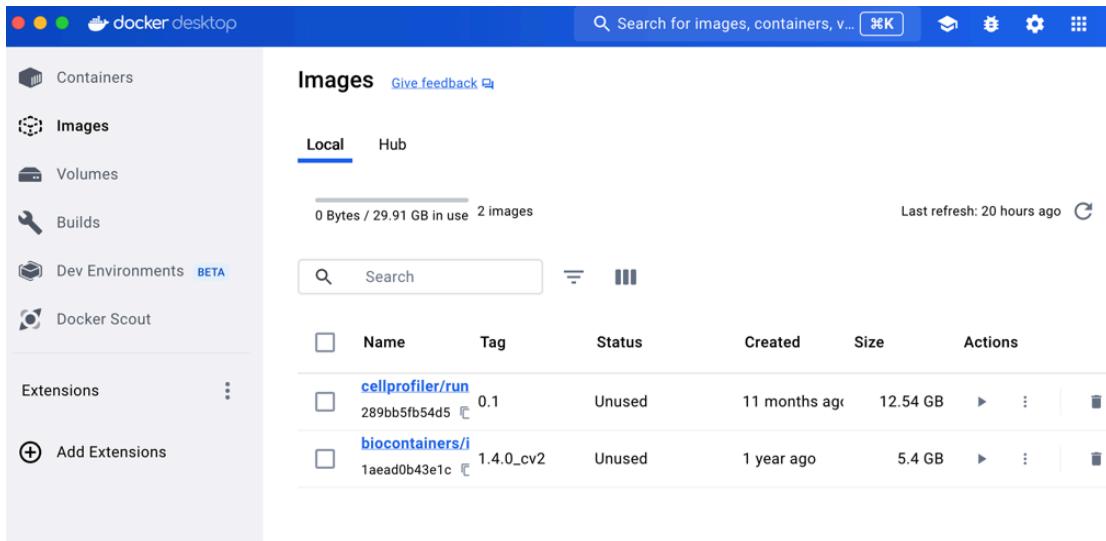
Docker Desktop

En el Ejercicio 3 (y para usuarios de Windows, opcionalmente también en el Ejercicio 2), queremos acceder a software que, o bien no queremos instalar localmente porque es complicado si uno no está cómodo usando Python (Cellpose), o bien que PODEMOS instalar localmente pero probablemente no funcione bien con la configuración de multiprocesamiento de otras herramientas (Ilastik).

Los científicos informáticos suelen utilizar **contenedores** de software para enviar herramientas o datos que son difíciles de instalar. [Aquí](#) puedes encontrar una buena introducción y descripción general para no expertos. Puedes pensar en los contenedores como «un sistema operativo preconfigurado en una caja». Debido a que vienen preconfigurados, la instalación de cualquier software se realiza una sola vez (por parte del creador del contenedor, no por ti) y debería seguir funcionando por mucho tiempo después de que, por ejemplo, la última actualización de Mac rompa una cierta instalación previa del software. Grupos como [biocontainers](#) ya han incluido en contenedores muchas de las herramientas que ya conoces y amas. Hay varios tipos de contenedores de software, pero uno de los más son los llamados contenedores **Docker**.

La mayoría de las personas que trabajan en Ciencias de la Vida no conocen los contenedores o no los utilizan, especialmente porque el uso típicamente implica acceder a ellos a través de una terminal. ¡Pero Docker no necesariamente implica usar la terminal! Hay [contenedores que generan sitios web interactivos para que los uses] (https://github.com/COBA-NIH/docker_gradio_demo), y CellProfiler tiene complementos que están específicamente configurados para llamar a otras herramientas que viven (y se ejecutan) dentro de contenedores Docker.

Para hacer esto, CellProfiler necesita tener la infraestructura para contenedores Docker en funcionamiento, lo cual puedes brindarle instalando un programa gratuito llamado [Docker Desktop](#). No necesitas crear una cuenta para usarlo, pero probablemente necesitarás reiniciar tu computadora una vez completa- da la instalación. Luego, siempre que deseas utilizar un complemento que llame a Docker en CellProfiler, simplemente debes asegurarse de que Docker Desktop esté abierto y ejecutándose, ¡CellProfiler se encar- gará 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

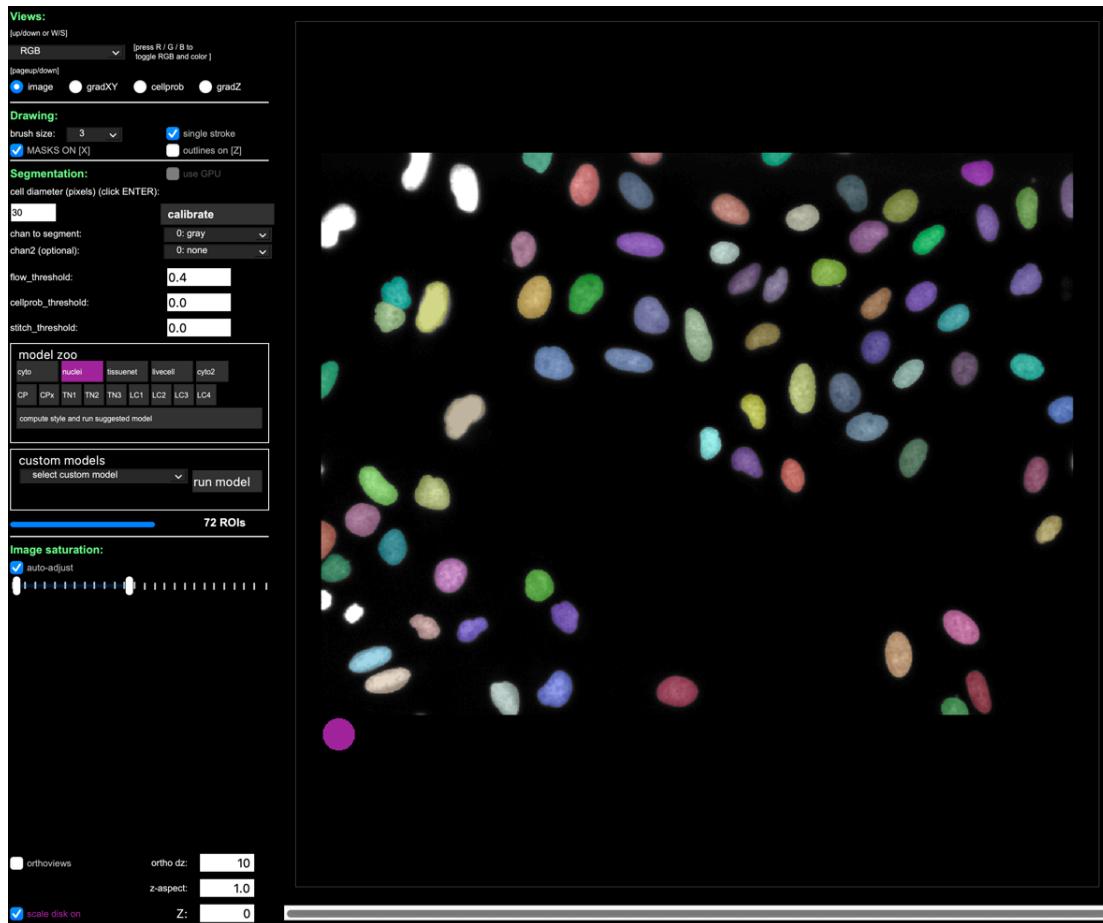
En este ejercicio, cargaremos *máscaras etiquetadas* o *label masks* (a veces también llamadas *matrices de etiquetas*, *label matrices*) generadas en Cellpose. Brevemente, una máscara marca es una imagen en la que todos los píxeles del fondo tienen un valor de 0, todos los píxeles del objeto 1 tienen un valor de 1, todos los píxeles del objeto 2 tener un valor de 2, etc. Es un formato comúnmente utilizado por muchas herramientas que realizan segmentación de objetos (aunque no es adecuado para aplicaciones donde hay objetos superpuestos). Dado que muchas herramientas (incluida CellProfiler) pueden crear estas máscaras, CellPro- filer tiene la capacidad de leerlas y detectarlas automáticamente como objetos.

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

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

Cellpose se instaló en un entorno de conda de acuerdo con las [instrucciones oficiales] (<https://github.com/MouseLand/cellpose?tab=readme-ov-file#installation>) 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

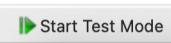
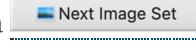
Las máscaras celulares se crearon navegando a la carpeta de imágenes y luego ejecutando el comando de abajo; esto requiere cierto conocimiento de la [sintaxis de la línea de comandos de Cellpose](#) pero es mucho más fácil para procesar una gran cantidad de imágenes.

```
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: 2.0.1
2024-04-30 10:59:04,196 [INFO] >>> using CPU
2024-04-30 10:59:04,198 [INFO] >>> running cellpose on 10 images using chan_to_seg GRAY and chan (opt) NONE
2024-04-30 10:59:04,198 [INFO] >>> using CPU
2024-04-30 10:59:04,198 [INFO] >> cyto2 << model set to be used
2024-04-30 10:59:04,198 [INFO] Downloading: "https://www.cellpose.org/models/cyto2torch_1" to /Users/bcimini/.cellpose/models/cyto2torch_1
100%|██████████| 10/10 [00:00<00:00, ?it/s]
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, ?it/s]
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, ?it/s]
2024-04-30 11:00:04,675 [INFO] WARNING: MKL version on torch not working/installed - 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 = 30.00 (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, ?it/s]
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<?, ?it/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] 10%# | 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 11:00:31,948 [INFO] 30%## | 3/10 [00:26<01:02, 8.96s/it]
2024-04-30 11:00:31,948 [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 11:00:48,542 [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 11:01:05,502 [INFO] >>> TOTAL TIME 8.38 sec
2024-04-30 11:01:05,546 [INFO] 70%##### | 7/10 [01:00<00:25, 8.54s/it]
2024-04-30 11:01:05,542 [INFO] >>> FINDING MASKS ~~~
2024-04-30 11:01:13,788 [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 [01:30<00:00, 10.00s/it]
2024-04-30 11:01:35,897 [INFO] 100%##### | 10/10 [01:30<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 %
```

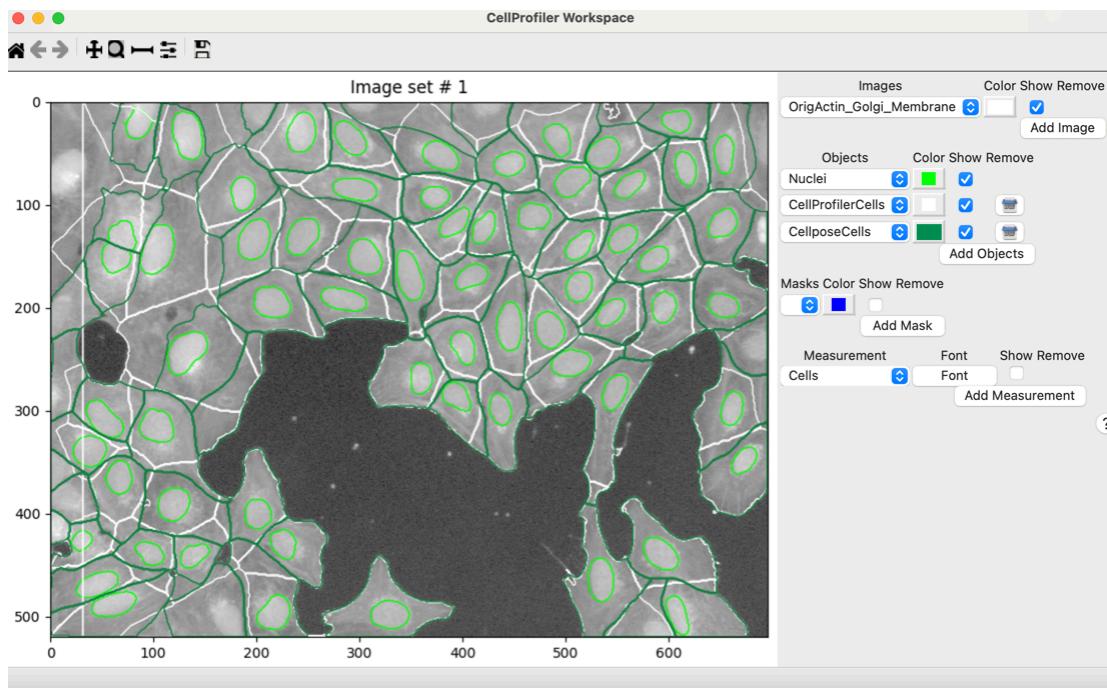
Ejecución de Cellpose desde la línea de comando

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. Coloca CellProfiler en Modo de Prueba (TestMode)  , abre el ícono del ojo  junto a OverlayOutlines y luego presiona  3 veces para crear una segmentación clásica y compararla con la versión generada por Cellpose. Puedes verificar la configuración en el módulo OverlayOutlines para ver qué color de contorno corresponde a qué segmentación en la salida.
4. Opcionalmente, abre el Visor de Espacio de Trabajo (Workspace Viewer) usando  para crear superposiciones fácilmente personalizables sobre la marcha.
5. Presiona  y repite 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 duplicado 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?

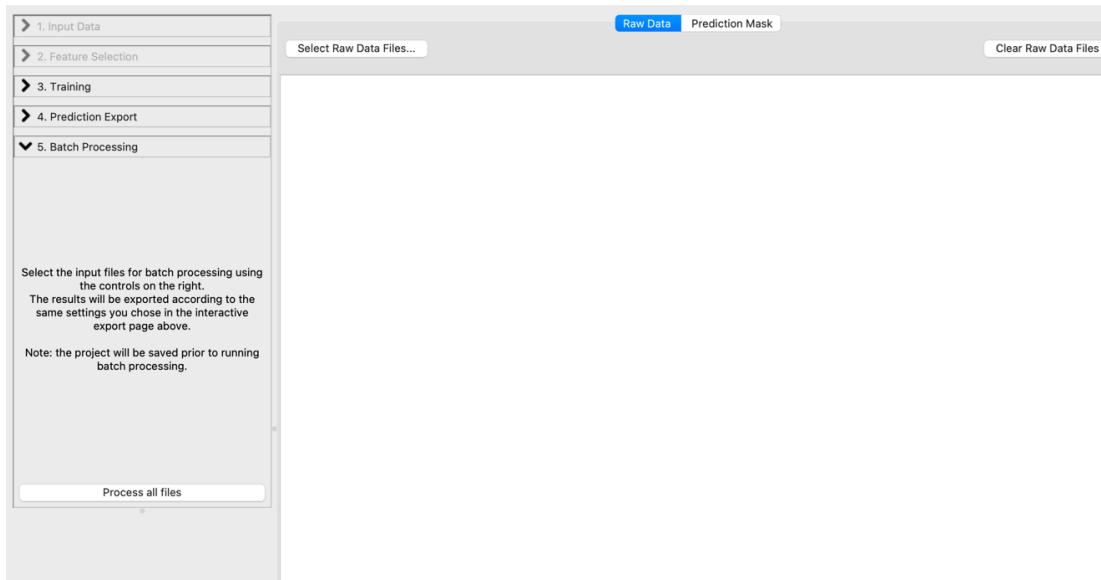
Ejercicio 2: Ejecutar ilastik localmente o desde un contenedor Docker

La segmentación clásica requiere que lo que nos interesa en una imagen sea brillante y que todos los demás píxeles sean (más) oscuros. Si tienes una buena señal fluorescente para lo que te importa, ¡genial! Si no es así, es posible que tengas que recurrir a algunos trucos.

Uno de esos trucos consiste en entrenar un clasificador píxel por píxel para que diga «esta es la probabilidad que estimo de que este sea un píxel que deseas que termine incluido en tu segmentación»; luego, este clasificador termina creando para cada píxel un *valor de probabilidad* que corresponde a la probabilidad

de que desee segmentar ese píxel. Si tu clasificador es bueno, obtendrás una imagen en la que los píxeles que te interesan tienen una alta probabilidad (brillantes) y el resto tienen una baja probabilidad (oscuros). ¡Eso es exactamente lo que queremos! Los científicos biológicos tienden a llamar a esto «**Clasificación de píxeles**»; Los informáticos a veces se refieren a ella como «**Segmentación semántica**».

Hay algunos complementos populares de Fiji para hacer esto, incluidos Weka Trainable Segmentation y Labkit, y definitivamente deberías usarlos si funcionan mejor para ti! Solemos usar ilastik porque facilita la automatización de la creación de clasificadores a partir de una cantidad muy pequeña de imágenes y luego su aplicación masiva a muchas otras en el modo «Batch Processing» o «Procesamiento en Lotes». Puedes consultar un tutorial que hemos escrito para ejecutar ilastik y luego CellProfiler en tutorials.cellprofiler.org (busca «Clasificación basada en píxeles»).



Modo de procesamiento por 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.

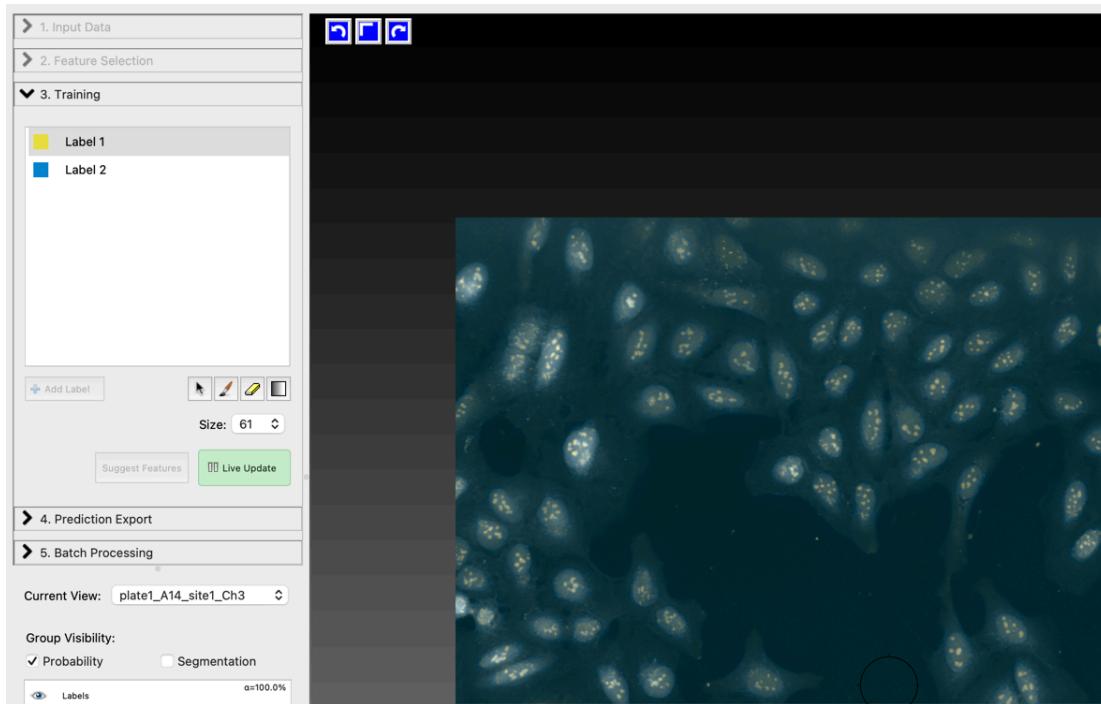
Necesitarás ilastik o Docker Desktop instalado en tu computadora para este ejercicio. Si usas ilastik, asegúrate de que esté CERRADO, si es Docker Desktop, asegúrate de que esté ABIERTO al momento de ejecutar el pipeline. Recomendamos instalar ilastik porque es una herramienta buena y útil y te permitirá 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 entrenando en 4 imágenes (A14_site1, E18_site1, D16_site1 y C12_site1) en las que etiquetamos solo dos clases: una clase para nucléolos (amarilla en la figura siguiente) y una clase para el resto de la imagen (azul en la imagen de abajo). PODRÍAMOS haber hecho más clases, cierto, pero esta manera funciona bien para nuestro objetivo.



Captura de pantalla de nuestro clasificador de Ilastik.

Cuando utilices ilastik para microscopía de fluorescencia, probablemente obtendrás el mejor rendimiento **si mantienes tus anotaciones al mínimo**: el clasificador que va a utilizar se entrenó utilizando un total de 31 píxeles de anotaciones 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. Te recomendamos encarecidamente que crees tus clasificadores un píxel a la vez, haciendo anotaciones puntuales. ¡Resiste la tentación de dibujar líneas onduladas por toda la imagen! Esto parece contradictorio, pero prometemos que es cierto.

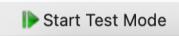
Habilitar el complemento Runilastik

1. Descarga [el complemento] (https://github.com/CellProfiler/CellProfiler-plugins/blob/master/active_plugins/runilastik.py) en una carpeta en tu computadora local. Como se indicó anteriormente, recomendamos fuertemente que lo guardes una carpeta que contenga SÓLO complementos.
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. Cierra y vuelve a abrir CellProfiler para cargar el complemento.

Cargar el clasificador y evaluarlo.

1. Arrastra el archivo `bonus_2_ilastik.cppipe` al panel de pipeline.
2. Arrastra la subcarpeta `images_Illum-corrected` desde el ejercicio principal al panel Imágenes.
3. Abre el módulo RunIlastik, establece la ruta a su archivo de proyecto (`NucleoliDetection.ilp`) y luego también:

- ilastik-instalado (recomendado): establece la ruta a tu instalación local de ilastik.
- Instalado en Docker: cambie la configuración superior de Local a Docker.

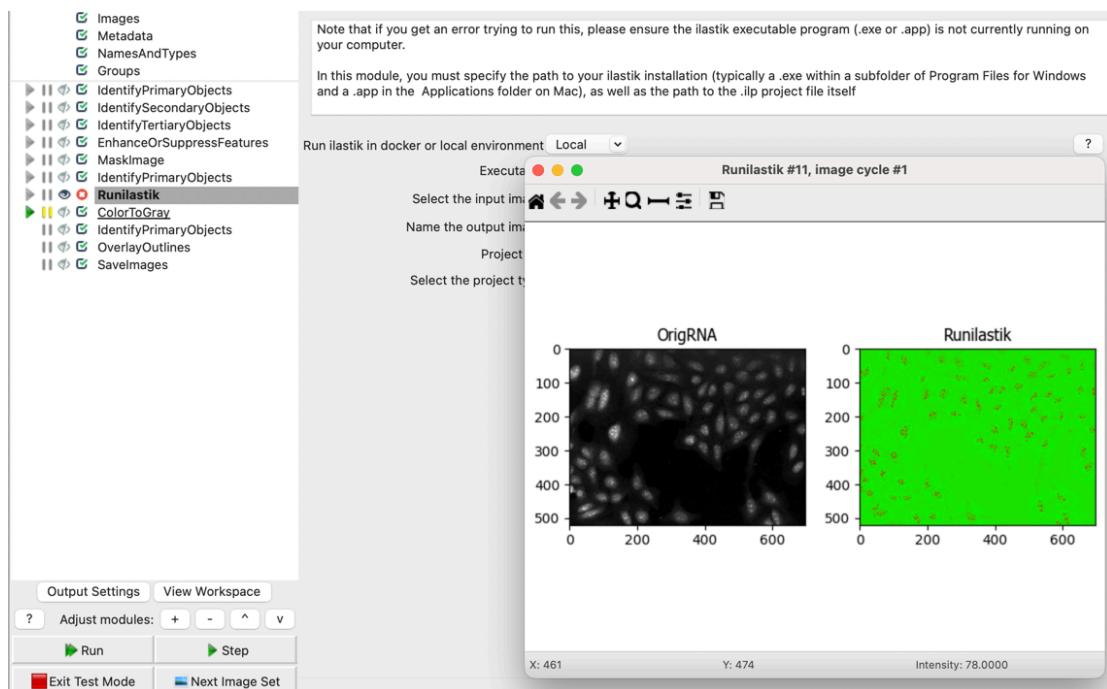
3. Coloca CellProfiler en Modo de Prueba (TestMode)  , abre los íconos de ojo  junto a **Runilastik** y **OverlayOutlines**, y luego presiona  . Puedes verificar la configuración en el módulo **OverlayOutlines** para ver qué color de contorno corresponde a qué segmentación en la salida.

- Si lo deseas, puedes poner una pausa  junto a **SaveImages**, o desmarcarlo  , para evitar que guarde imágenes.

Nota:

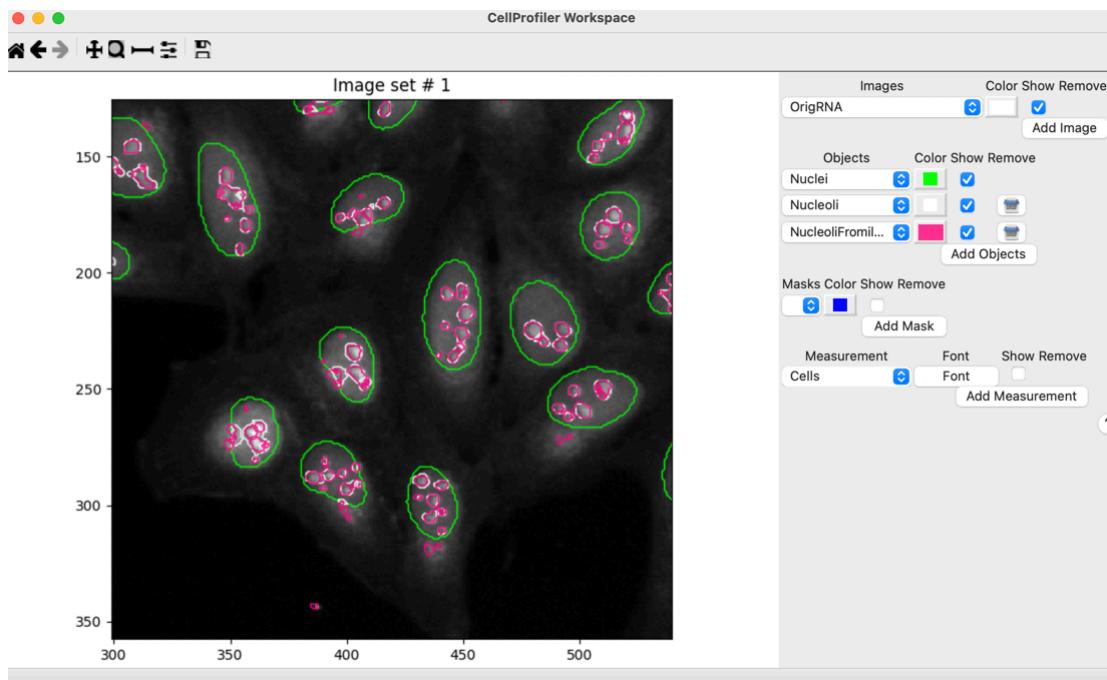
Si usas Docker, la primera vez que ejecutes el módulo **Runilastik**, deberás descargar un archivo de ~5 GB, que puede ser lento dependiendo de tu conexión a Internet. ¡Sin embargo, solo necesitas realizar este paso una vez!

4. Evalúa tu predicción en **Runilastik** a través de algunos conjuntos de imágenes: ¿cómo tan bien funciona? ¿Funciona peor en imágenes en las que no fue entrenado?



Captura de pantalla del módulo **Runilastik**

5. Evalúa tus segmentaciones en algunos conjuntos de imágenes utilizando **OverlayOutlines** y/o la herramienta **WorkspaceViewer**. ¿Hay casos en los que crees que la segmentación basada en modelos de ilastik funciona mejor? ¿Y la segmentación de filtrado y enmascaramiento?



Evaluación de segmentaciones en el `WorkspaceViewer`

Bonus del bonus: entrena tu propio modelo ilastik!

Según tus evaluaciones anteriores, ¿puedes identificar algunos lugares donde entrenamiento adicional podría ayudar a solucionar algunos problemas en el modelo ilastik?

1. Abre el archivo `NucleoliDetection.ilp` en ilastik.
2. Ve a la pestaña «Training».
3. Activa «LiveUpdate».
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 demostrarte 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, con mucho, nuestro complemento más popular, simplemente porque a) iCellpose es genial! y b) instalar software usando `conda` cuando no te sientes muy cómodo computacionalmente no lo es. Puede usar el complemento en de dos modos diferentes: usando una instalación local `conda` o `python` que contenga tanto CellProfiler como Cellpose o usando Docker. El tiempo de ejecución con Docker es sustancialmente más lento (alrededor de un minuto más por imagen, en nuestras pruebas), pero si la instalación te llevaría mucho tiempo y sería frustrante, en este sentido puedes «intercambiar» tu tiempo de frustración personal tiempo de ejecución de CellProfiler en tu computadora (mientras, tú puedes hacer otra cosa!). ¡Para muchas personas, este es un buen intercambio!

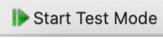
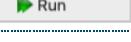
Iniciar el Docker Desktop

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 que se abra Docker Desktop, usa la función de búsqueda para buscar **Cellpose** y busca el modelo **cellprofiler/runcellpose_with_retrained** y extráelo. Si por alguna razón esto no funciona, sigue adelante con el ejercicio, pero esté te ahorrará tiempo más adelante.

Toma el complemento RunCellpose

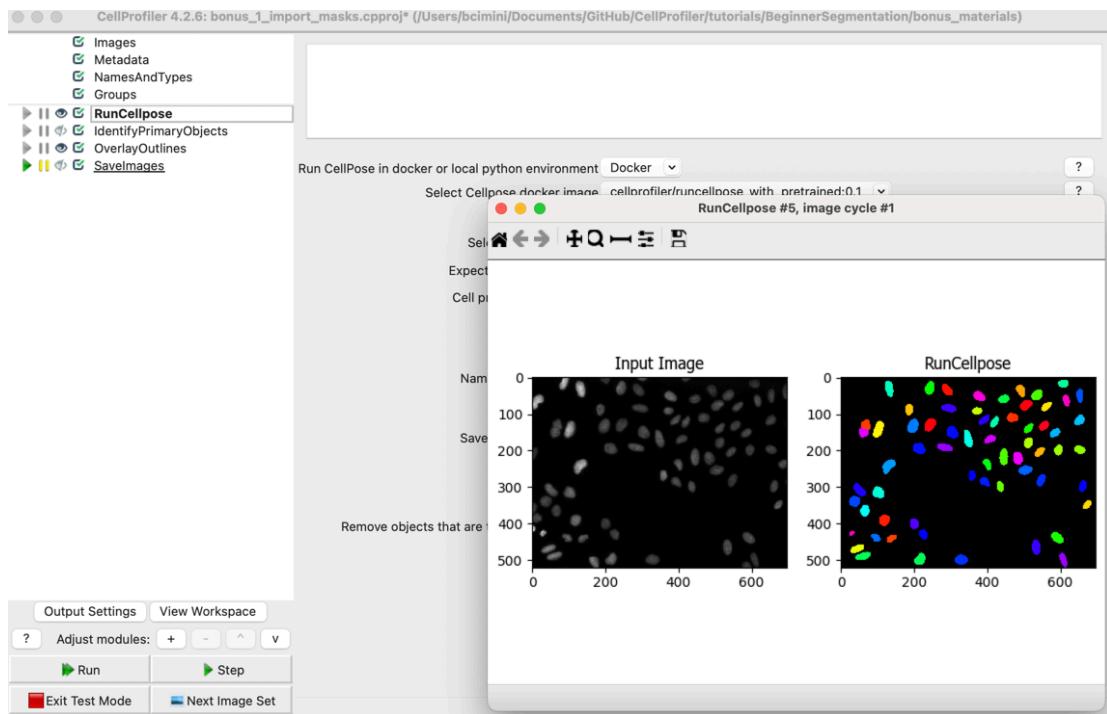
1. Descarga [el complemento] (https://github.com/CellProfiler/CellProfiler-plugins/blob/11b46ee7f6eb78f97f784a731c5a1931b66c90d4/active_plugins/runcellpose.py) en una carpeta en tu computadora. Como se indicó anteriormente, recomendamos fuertemente guardarla en una carpeta que contenga SÓLO complementos.
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

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. Coloca CellProfiler en Modo de Prueba  , abre los íconos de ojos  junto a **RunCellpose** y **OverlayOutlines**, y luego presiona 
- Si lo deseas, puedes poner una pausa junto a **SaveImages**, o desmarcarlo, para evitar que guarde imágenes.

Nota:

Si aún no extrajiste el contenedor en la sección anterior (Docker Desktop), la primera vez que ejecutes el módulo **RunCellpose**, deberás descargar un archivo de ~13 GB, lo cual puede ser lento dependiendo de tu conexión a internet. ¡Sin embargo, solo necesitas realizar este paso una vez!

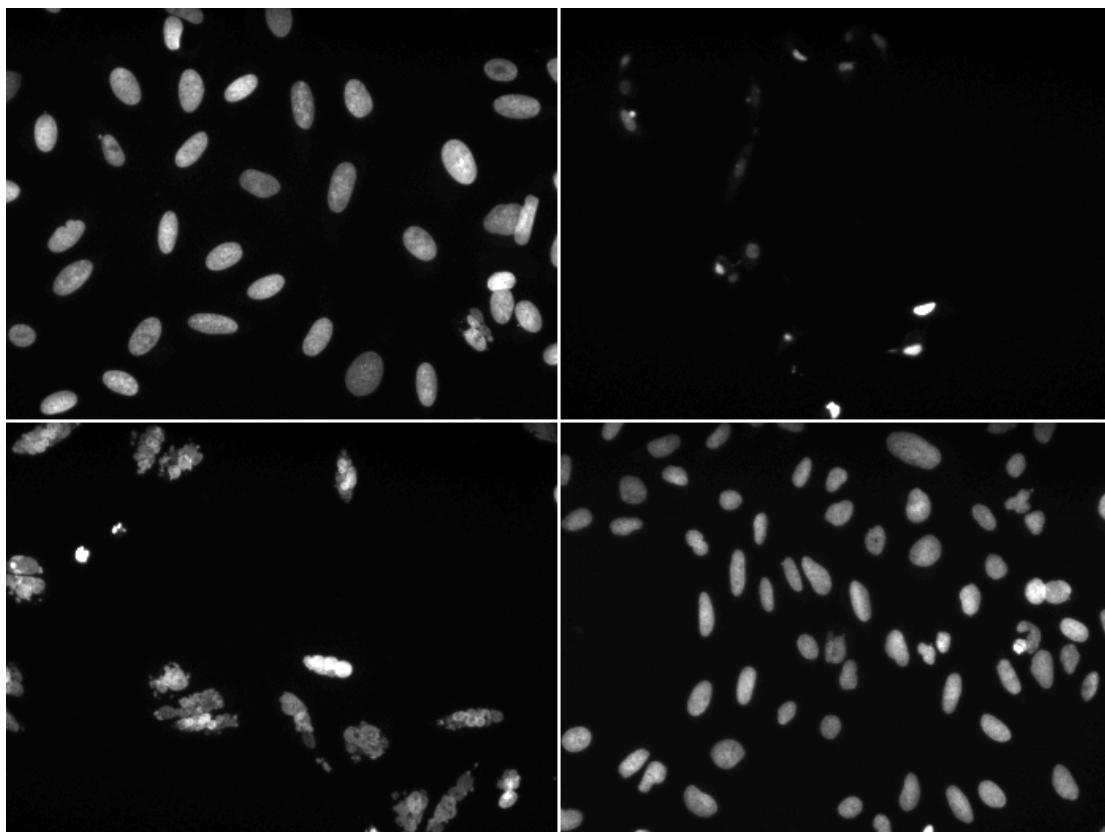


La salida del módulo RunCellpose

4. 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?
5. Utiliza el botón `[?]` para obtener más información sobre los diferentes parámetros que puedes pasarte a Cellpose (no los ofrecemos todos, pero sí muchos!). ¿Cómo afecta cambiar estos parámetros al resultado? ¿Qué tal cambiar el modelo que estás usando y/o la imagen que estás segmentando?

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 lista de pocillos: A08, A12, B12, B18, C7, D6, D19, D22, E3. Puedes seleccionarlos con **Test -> Choose Image Set**

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 complementos y módulos de CellProfiler?

1. Lee el [paper de complementos de CellProfiler](#)
2. Mira este [video](#) para aprender a escribir tu propio complemento!