

# Protocolo para Deconvolución local

---

Existen dos métodos para la deconvolución de imágenes en el ordenador del laboratorio: por CPU y por GPU. La deconvolución por CPU (dCPU) es más lenta, aunque tiene integración con Fiji, por lo que se recomienda en el caso en que se pretendan realizar más ajustes a la imagen, de color, recorte, z-stacking... La deconvolución por GPU (dGPU) es mucho más rápida, aunque no se encuentra integrada en Fiji. En ambos casos, es posible trabajar con archivos .dv directamente.

## Antes de comenzar

En la carpeta "Deconvolución" se encuentran las subcarpetas y archivos siguientes:

- PSFs, y las subcarpetas para cada tipo de microscopio utilizado, que contienen los archivos **Point Spread Function** que funcionan como **kernel** del filtro de deconvolución. Los archivos están en formato .tif, con la nomenclatura **psf\_{big}\_{nm}**, donde **big**, de estar presente, indica el filtro de mayor calidad y mayor tiempo de procesamiento, y **nm** la longitud de onda de excitación para cada PSF.
- mrc, que contiene los archivos necesarios para el funcionamiento del programa de dGPU en cuanto a la apertura de ficheros .dv o .mrc
- deconvolve.py, la aplicación para dGPU, ejecutable desde un terminal de comandos en el entorno Conda adecuado
- instrucciones.md, este mismo archivo

## Deconvolución por CPU (dCPU)

La dCPU se puede realizar en Fiji (o ImageJ) con el Plugin DeconvolutionLab2.

1. Abrir en Fiji la imagen a procesar, cualquier formato es válido. Es importante que se seleccione la opción "Split Channels".
2. Abre el fichero PSF para la longitud de onda de excitación del canal de la imagen anterior a deconvolucionar.
3. Navega a Plugins > DeconvolutionLab2 > DeconvolutionLab2 Lab
4. Dentro del apartado **Image**, navega a Choose > Get platform: image from the platform, y selecciona de la lista el stack para el canal a deconvolucionar
5. Dentro del apartado **PSF**, navega a Choose > Get platform: image from the platform, y selecciona de la lista el PSF abierto anteriormente para la longitud de onda adecuada
6. Dentro del apartado **Algorithm**, selecciona el algoritmo de Richardson-Lucy, y ajusta el número de iteraciones en el rango  $10 \leq N \leq 15$ . Cuantas más iteraciones, se puede suponer una mejor calidad del procesamiento, aunque la duración también aumenta.
7. (Opcional) Puedes configurar varios trabajos en serie en la opción **Batch > Add job**, realizando todo el proceso anterior para cada imagen a procesar.
8. Una vez configuradas todas las opciones, incluso si se ha programado una rutina **Batch**, iniciamos la tarea en **Launch > Run**. Deben aparecer dos ventanas en las que se vaya indicando el progreso de la tarea, así como la carga de CPU/RAM y el estado de cada uno de los archivos de imagen que se estén cargando.

## Deconvolución por GPU (dGPU)

**IMPORTANTE:** La dGPU sólo es válida para imágenes con z-stack, no para proyecciones. No obstante, sí son válidos archivos con un solo punto temporal. La dGPU se realiza a través de un **wrapper** para la librería CUDADecon, que permite la deconvolución utilizando los núcleos CUDA de una GPU Nvidia (AMD no compatible). Al igual que la dCPU es posible en cualquier equipo, instalando el plugin DeconvolutionLab2 en Fiji, la dGPU sólo es posible en equipos con una GPU compatible. Se recomienda una GPU igual o superior a una Nvidia GT 1030. Se generarán, como resultado, los archivos con nomenclatura {nombre\_original}{longitud}{D3D}.[dv|tiff] y {nombre\_original}.[dv|tiff].log, con el registro de toda la actividad del proceso de deconvolución.

## Deconvolución de archivos .dv (para DeltaVision)

1. Ejecutar **Windows PowerShell**. El terminal está en el entorno **base** de Conda.
2. Cambiar el entorno Conda a **decon\_env**

```
(base) PS C:\Users\admincabd> conda activate decon_env  
(decon_env) PS C:\Users\admincabd>
```

3. Navega a la carpeta donde se encuentre **deconvolve.py**

```
(decon_env) PS C:\Users\admincabd> cd .\Desktop\Deconvolución\
```

4. El comando para ejecutar la aplicación de deconvolución es el siguiente:

```
(decon_env) PS C:\Users\admincabd\Desktop\Deconvolución> python .\deconvolve.py --  
source [RUTA/ARCHIVO.dv] --psf [RUTA/PSF.tif]
```

Los archivos (entre []), deben ser provistos con ruta relativa o absoluta, separados por espacio. Para seleccionar varios archivos, el comando quedaría como:

```
(decon_env) PS C:\Users\admincabd\Desktop\Deconvolución> python .\deconvolve.py --  
source (get-item [RUTA/*.dv]) --psf [RUTA/PSF.tif]
```

Cuando se especifica el **cmdlet get-item**, se está especificando que la ruta vendrá dada por un modificador con **wildcards**. Si indicamos como \*.dv, estaremos pasando el comando para leer todos los archivos con extensión .dv de una carpeta. El orden de los canales debe ser respetado a la hora de indicar los PSF. Si un archivo tiene 2 canales (460nm y 618nm), hay que indicar los PSF, separados por espacio y en su ruta absoluta o relativa, en ese mismo orden. Para omitir uno o varios canales, hay que indicar la posición del correspondiente archivo PSF con barra baja (\_) en el terminal. Pueden seleccionarse cualquiera de los PSF, estén marcados o no como **big**. 5. Una vez ejecutemos el comando, a la finalización del programa deben haberse generado los ficheros en el directorio donde nos encontremos (en este caso, el directorio **Deconvolución**). **Asegúrese de seleccionar los PSF correctos para el tipo de microscopio**. 6. Es posible consultar una lista detallada de los comandos para la aplicación ejecutando

```
(decon_env) PS C:\Users\admincabd\Desktop\Deconvolución> python .\deconvolve.py --help
```

## Deconvolución de archivos tiff (para SpinningDisk u otros .tiff)

### Para cualquier .tiff

Con los mismos pasos 1-3 anteriores, es posible realizar la deconvolución de archivos .tif/.tiff, con un orden de stacks T-Z-W-XY (tipo DeltaVision). 4. A la hora de introducir el comando para la ejecución del programa, hay que tener en cuenta algunas especificaciones:

```
(decon_env) PS C:\Users\admincabd\Desktop\Deconvolución> python .\deconvolve.py --source [RUTA/ARCHIVO.tif(f)] --psf [RUTA/PSF.tif] --xyimage N --zimage N -w N1 N2 N3
```

El argumento `--xyimage` sirve para especificar el tamaño de pixel en micras de la imagen a deconvolucionar. El argumento `--zimage` indica la separación entre los Z-stacks, también en micras. El argumento `-w` viene seguido de tantos números como canales haya en la imagen, representando la longitud de onda de emisión para cada uno de ellos. El resto de pasos (5-6) y de posibles modificaciones a parámetros son equivalentes en el caso DV y TIFF.

### Para SpinningDisk (estructura por carpetas)

Con los mismos pasos 1-3 anteriores, es posible realizar la deconvolución de datos obtenidos del SpinningDisk, con una estructura de archivos Z-stack individuales por tiempo, y en carpetas individuales para cada canal. 4. A la hora de introducir el comando para la ejecución del programa, hay que tener en cuenta algunas especificaciones:

```
(decon_env) PS C:\Users\admincabd\Desktop\Deconvolución> python .\deconvolve.py --source [RUTA_CARPETA] --psf [RUTA/PSF.tif] --xyimage N --zimage N -w N1 N2 ... Nn --spinning M1 M2 ... Mn
```

Hay que tener en cuenta que, en este caso, `--source` no será un fichero de imagen, sino una carpeta o carpetas que contendrán dentro la estructura de datos típica para la microscopía por SpinningDisk. El argumento `--xyimage` sirve para especificar el tamaño de pixel en micras de la imagen a deconvolucionar. El argumento `--zimage` indica la separación entre los Z-stacks, también en micras. El argumento `-w` viene seguido de tantos números como canales haya en la imagen, representando la longitud de onda de emisión para cada uno de ellos. El argumento `--spinning` viene seguido de los nombres de los canales (coincidencia con el nombre de las carpetas) para el tipo de datos a tratar. **Asegúrese de seleccionar los PSF correctos para el tipo de microscopio.** El resto de pasos (5-6) y de posibles modificaciones a parámetros son equivalentes en el caso DV y TIFF.

### Otros parámetros modificables (DV y TIFF)

- *--project*, realiza la proyección máxima de un Z-Stack, para todos los planos. Si se especifica el argumento *--planes min MAX*, donde min y MAX son planos N-1 (empezando por el 0), se indicará un rango de Z-stacks a proyectar
- *--na*, indica la apertura numérica del objetivo, que debe coincidir con la del PSF.
- *--refractive*, indica el índice de refracción del aceite o medio de inmersión utilizado.
- *-i N*, donde N es el número de iteraciones para el algoritmo de Richardson-Lucy implementado.
- *--zfilter* y *--xyfilter*, los tamaños o separación en micras del stack Z y de los píxeles XY, para los archivos PSF indicados.