

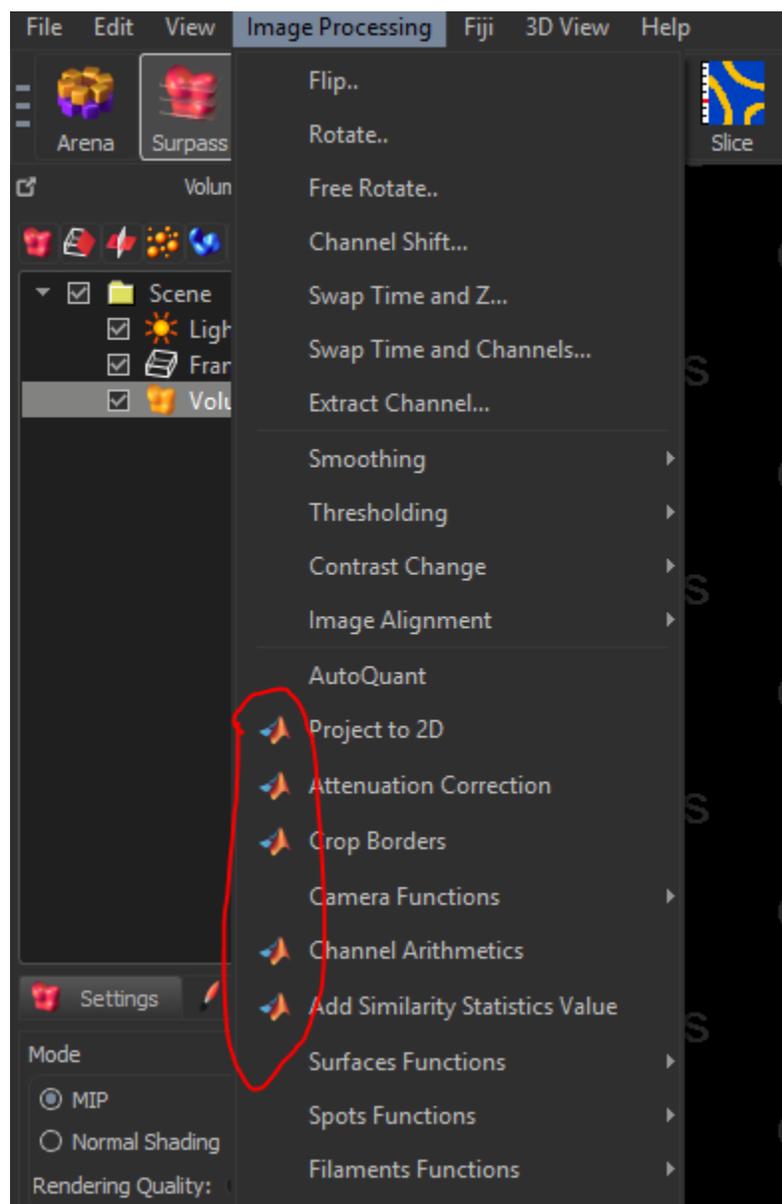
1 What is an XTension?

XTensions are plug-ins for Imaris. They allow you to run functions that Imaris does not have built into it already as well as create your own functionality within Imaris.

2 Running Imaris XTensions

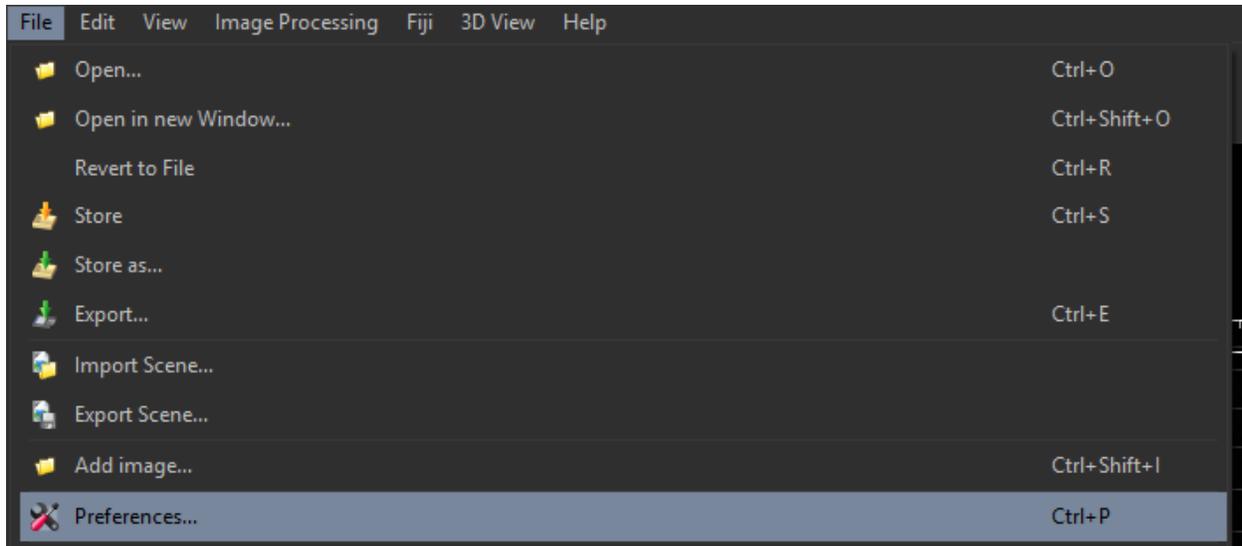
2.1 Using XTensions

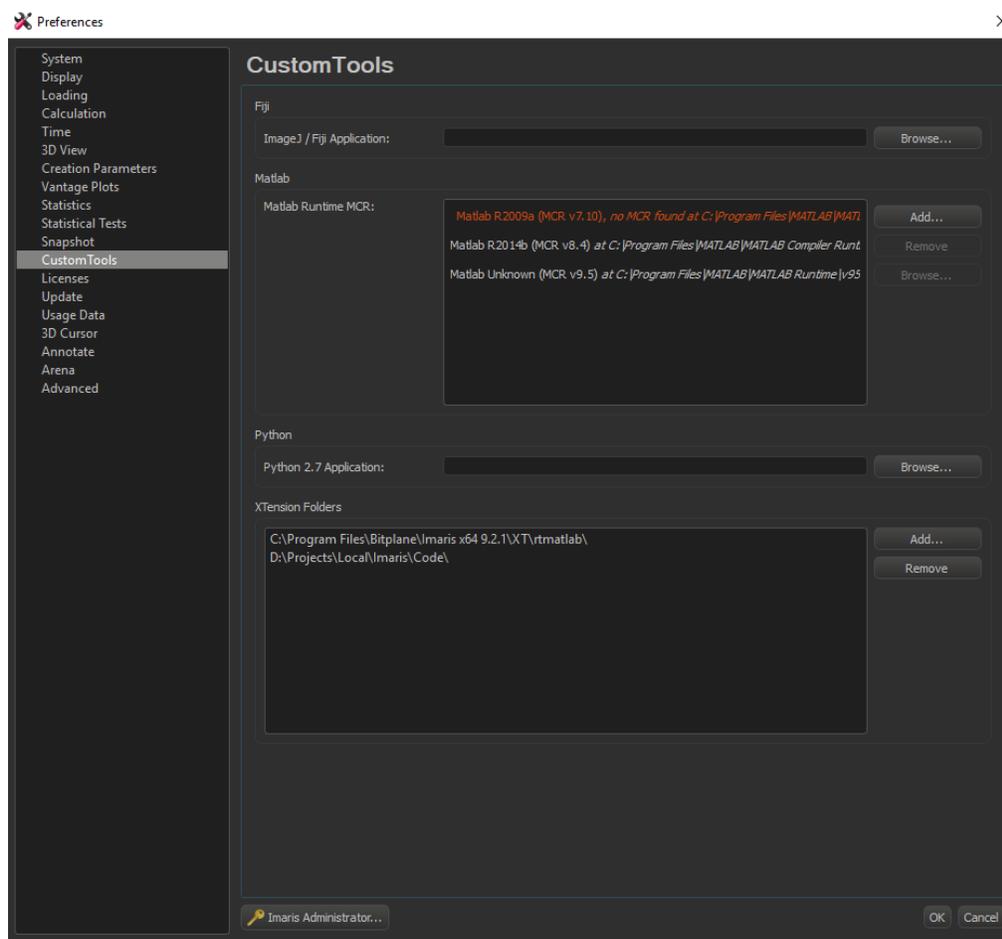
To view available XTensions in Imaris, go into Surpass and click the *Image Processing* dropdown. MATLAB XTensions will be visible with a MATLAB logo next to them. Imaris will come with certain XTensions already installed and available to you.



2.2 Adding Custom XTensions

When you have a folder that includes your Matlab XTension's .exe and .xml files, you can add it to the XTensions Imaris can see. To see where Imaris looks for XTensions, go to *File -> Preferences -> Custom Tools*.





Where it says “XTension Folders”, it will show the path to the folders it looks in when looking for XTensions. You can add the folder your files are in to where Imaris looks by using the “Add” button. The folder that includes your .exe and .xml files must also have the file *ImarisLib.jar* in it, as it will be referenced when the XTension runs. If it is not already present, you can copy in the version included with Imaris. This file is usually available at a path like this: *C:\Program Files\Bitplane\Imaris x64 X.X.X\XT\rtmatlab\ImarisLib.jar*

Once you have this path pointing to the folder with the Imaris XTension(s) you want to use, restart Imaris and the custom XTension(s) will appear in the dropdown menu with the other XTensions.

3 Spectral Unmixing XTension

3.1 XTension code

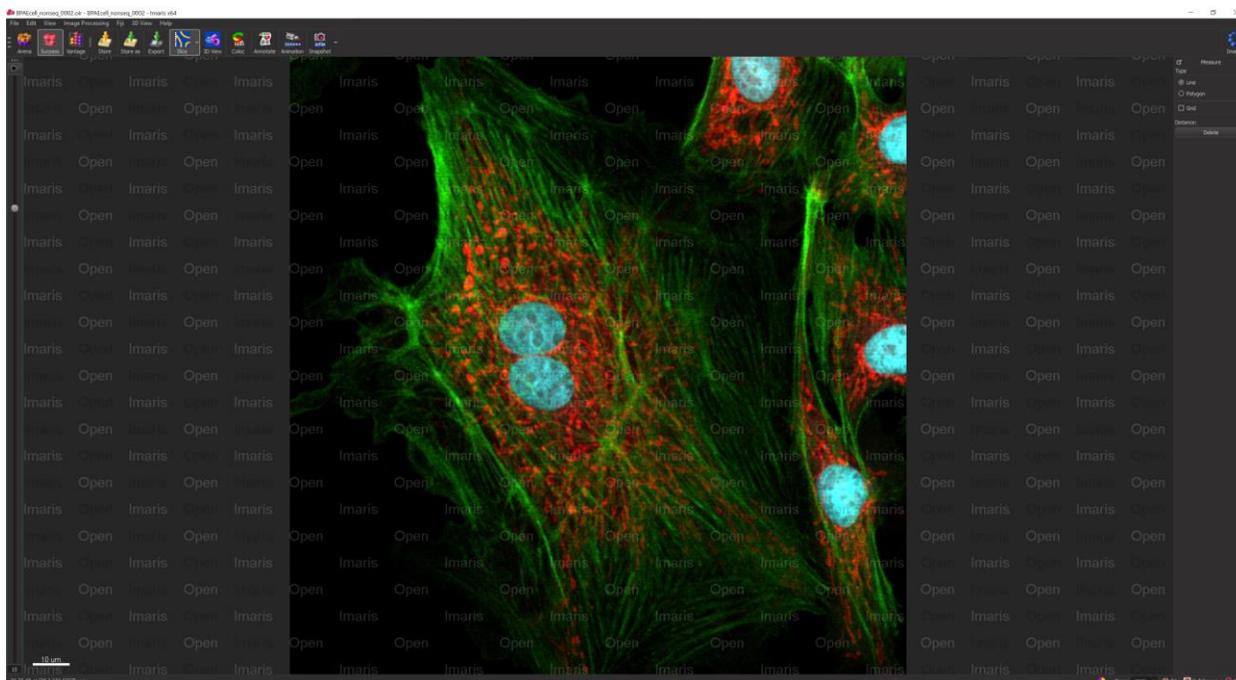
The Spectral Unmixing XTension written and maintained by the Multiphoton Research Core is available at <https://www.urmc.rochester.edu/research/multiphoton/image-analysis/spectral-unmixing.aspx>

Download the Spectral-Unmixing-Imaris-XTension-Files.zip file under the “XTension” link on the above page and unzip it on your computer.

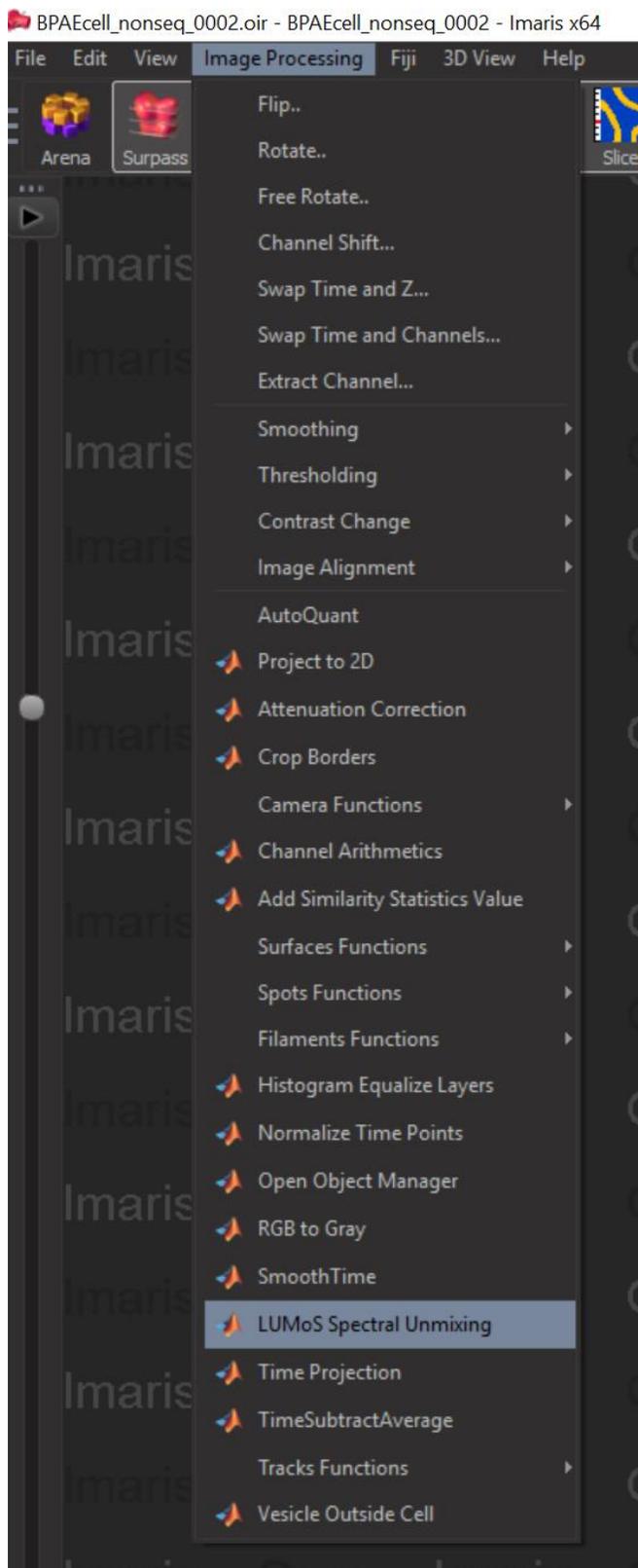
Using the instructions from section 2.4, point Imaris to the location where you saved the XTension.

3.2 Running the XTension

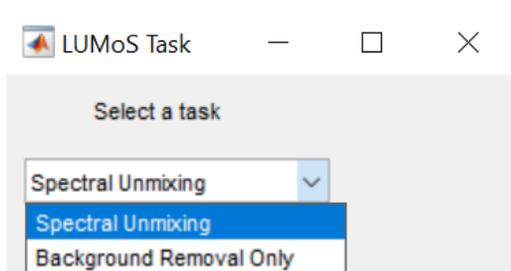
Open the image you wish to unmix in Imaris



The Spectral Unmixing XTension can be found under *Image Processing* -> *LUMoS Spectral Unmixing*



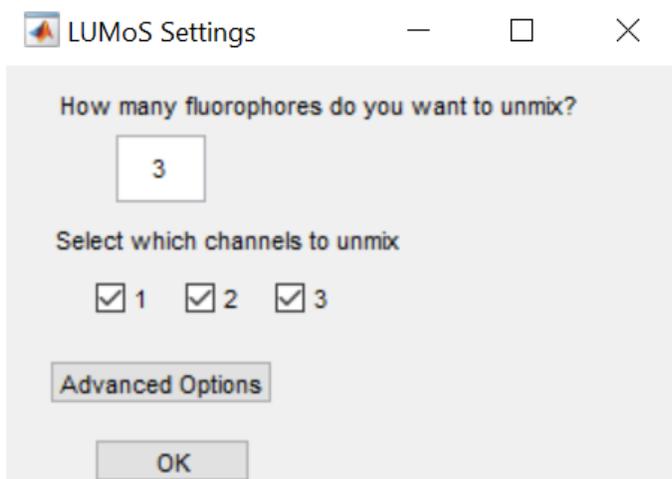
After you select LUMoS Spectral Unmixing, you will be prompted to choose your task. You have the choice between *Spectral Unmixing* and *Background Removal Only*. *Spectral Unmixing* will separate each fluorophore into a single channel while at the same time separating the background of the image into its own channel. *Background Removal Only* will remove the background from an image while leaving the rest of the image untouched.



3.3 Spectral Unmixing

The *Spectral Unmixing* option uses *k*-means to cluster each pixel into a channel corresponding to either a fluorophore or the background.

After selecting *Spectral Unmixing*, you will be asked about your image:



How many fluorophores do you want to unmix – The number of channels the algorithm will create for your fluorophores. An additional channel is added behind the scenes. This is the “*k*” in *k*-means. A good starting point is to select every channel for unmixing and ask it to unmix however many fluorophores were present during imaging. If there are some channels that you already know only include one fluorophore, you can reduce the number of fluorophores you want to unmix by one. If, after unmixing, it seems like there are still multiple fluorophores contained in some of the output channels, you can increase the number of fluorophores you ask it to unmix. This will have the effect of adding more clusters to *k*-means. If successful, this will result in two or more clusters that represent the same fluorophore but no clusters that represent multiple fluorophores.

Select which channels to unmix - Any channel whose box you check will be included in the unmixing and any unchecked channel will simply be preserved in the final image. If there are some channels that you know already only include one fluorophore, you can uncheck their box.

Advanced Options – Selecting this will give you a new pop-up with more choices. If this is not selected, default values will be used for all advanced options.

LUMoS Settings

How many fluorophores do you want to unmix?
3

How many instances of colocalization do you want to separate?
0

Select which channels to unmix
 1 2 3

Number of replicates (runs k-means multiple times and uses best result)
10

Representative timestep
1

Representative slice
14

Maximum number of iterations
100

Unmix based on pixel intensity ratio (instead of absolute channel values)
 Allow multiple fluorophores to occupy a single pixel
 Infer channel names
 Apply median filter

OK

How instances of colocalization do you want to separate? – The number of channels the algorithm will create for colocalized fluorophores. Each instance of colocalization adds an additional output channel to the image. For example, if you have two channels, - CH1 and CH2 - and wanted to separate pixels that are purely CH1, purely CH2 and a mix of CH1 and CH2, you would tell the algorithm to unmix 2 fluorophores with 1 instance of colocalization. The co-localization of CH1 and CH2 will have its own spectral signature and be given a unique output channel. This will only work if there are instances in your image where CH1 and CH2 both exist apart from one another.

Number of replicates – Runs k-means multiple times and uses the best result. More replicates will improve clustering quality at the expense of taking longer. There tends to be diminishing returns with number of replicates. Default is 10.

Representative timestep– In order to speed up clustering, clusters are defined based one 2D slice of your data and then applied to the rest of your data afterwards. Choose a timestep where you can see all of the different structures and colors. This number indexes from 1. If you only have one timestep, put 1 here. Default is the middle timestep.

Representative slice– Same as representative timestep but for an axial slice. Default is the middle slice.

Maximum number of Iterations – K-means is an iterative algorithm and this option can cap the number of iterations in each replicate of k-means. A lower number cuts off the algorithm earlier which could potentially speed the algorithm up at the expense of quality. Setting this to -1 tells the algorithm to run until it converges. Default is 100.

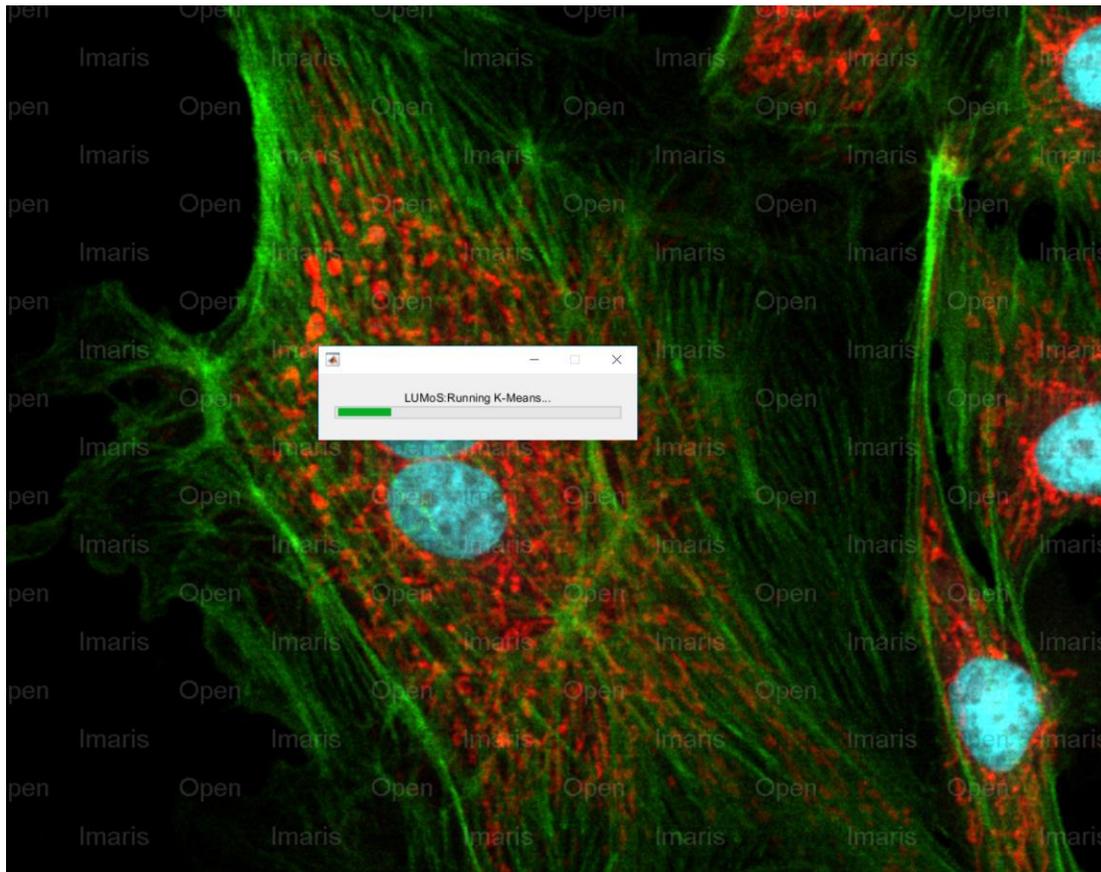
Unmix based on pixel intensity ratios – If checked, the algorithm will treat two pixels with the same ratio of channel intensities as being similar to one another. This is useful when overall intensity drops off with depth but the color stays the same. If pixels of different absolute intensity can represent different things in the image, leave this box unchecked. Default is true.

Allow multiple fluorophores per pixel – Select if you want each pixel to be able to belong to multiple output clusters. Implements fuzzy c-means instead of k-means. Default is false.

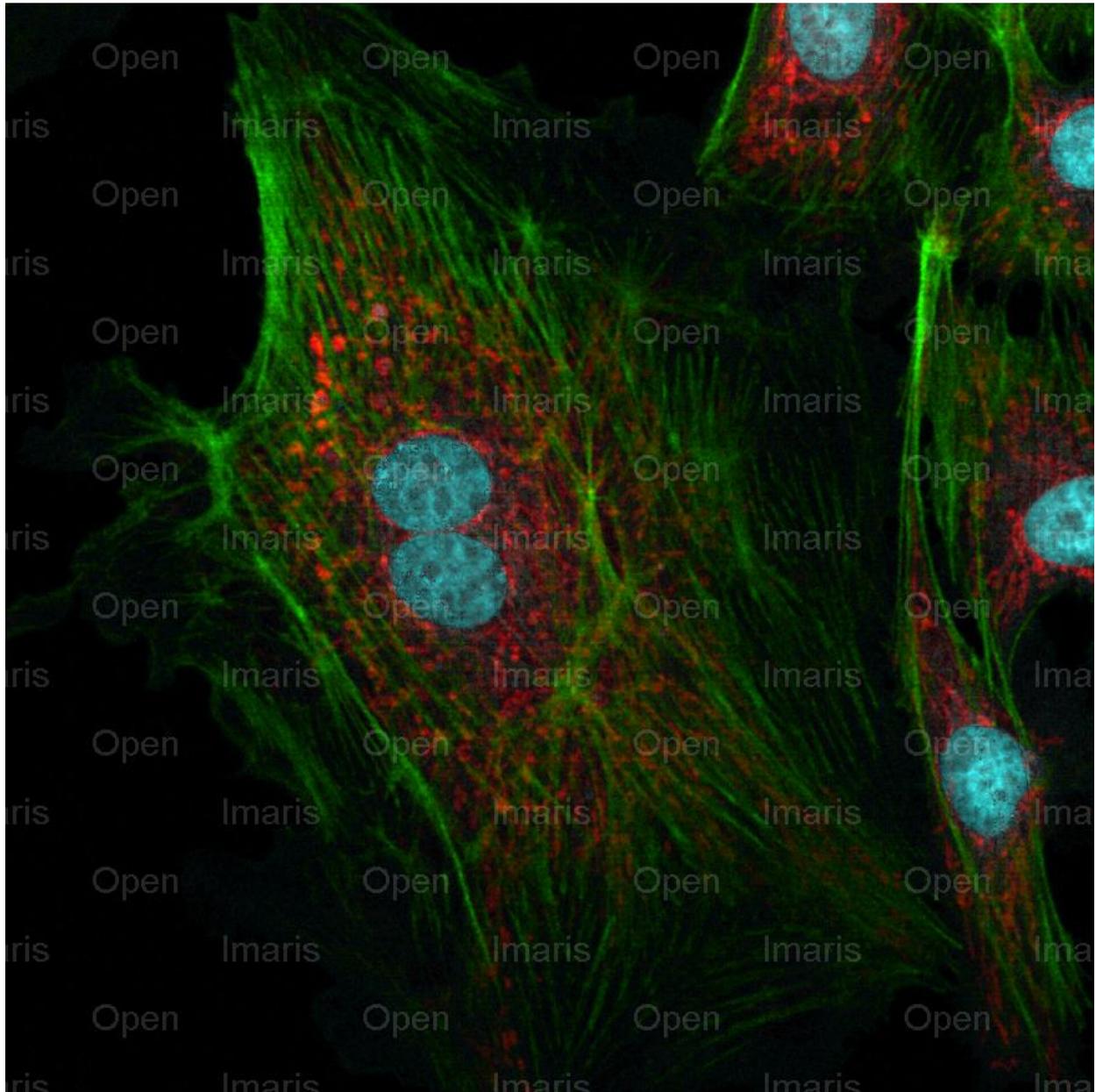
Infer channel names – Select if you want the output clusters to be named and colored according to the original names their channels had in Imaris. This will not necessarily work in cases where fluorophores overlap and double checking the names it gives channels is recommended. This will also label background and colocalization channels if applicable. Default is true.

Apply median filter – Select if you want to have a median filter applied to the input channels before unmixing happens. The median filter makes neighboring pixels more likely to represent the same fluorophores. Applies a 5x5x5 median filter. Not recommended when structures are <5 pixels thick. Filtered intensities are not used for the final output intensities, only classification. Default is true.

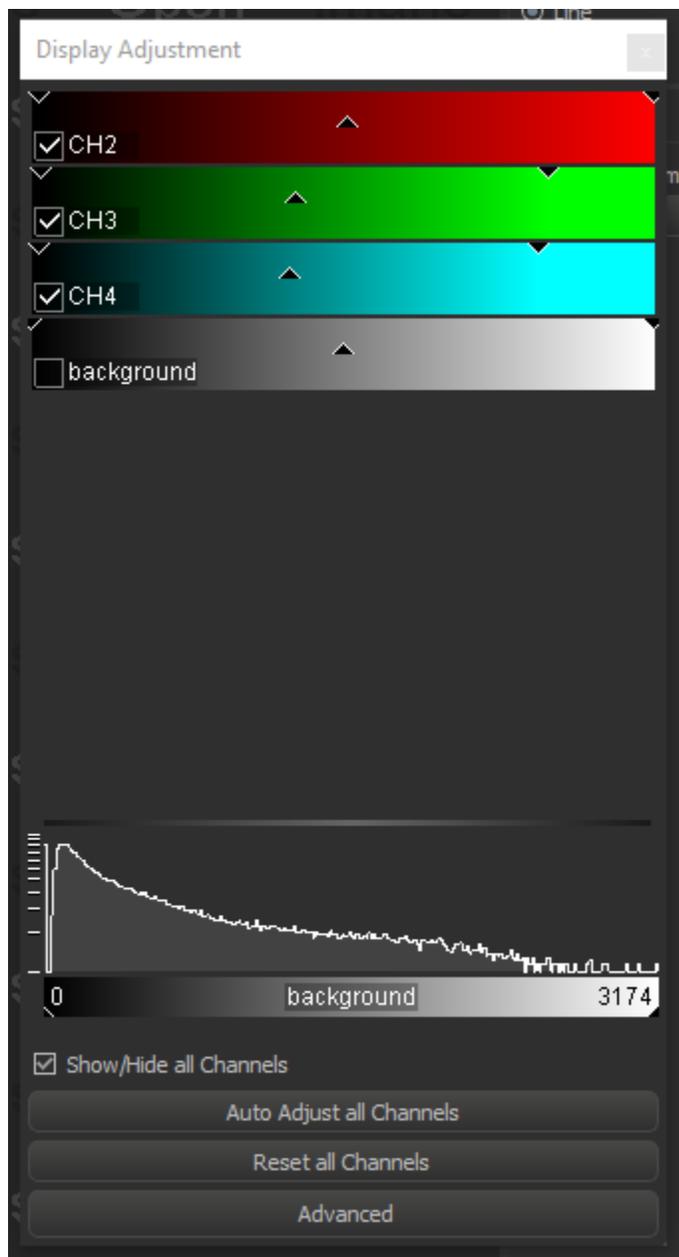
The progress of k-means will be displayed in a window which will pop up automatically after a minute or so when you start the XTension.



When finished, the XTension will write a new image to your Imaris workstation with the separated channels. (Note how the cell nuclei no longer show up in the green channel in this example.)



You can manually change the appearance of the channels with Imaris's Display Adjustment tool which can be brought up with ctrl+d on Windows or with *Edit -> Show Display Adjustment*. Here you can adjust channel intensities and turn off the background channel by unchecking it's box. It won't change the data, just its display in Imaris.



3.4 Background Removal Only

The *Background Removal* option uses *k*-means to label each pixel as being background or not. It then removes the background pixels and leaves the rest of the pixels untouched. It does not require any selection of options.

3.5 Autofluorescence removal

This spectral unmixing method can also be used for detecting and separating autofluorescence from an image. If you wish to use the spectral unmixing algorithm to create a separate channel for pixels where autofluorescence is occurring, add an extra “fluorophore” when asked how many fluorophores you want to unmix. For example, if you have two channels, - CH1 and CH2 - and wanted to separate pixels that are CH1, CH2 and autofluorescence, you would tell the algorithm to unmix three fluorophores. The autofluorescence will have its own spectral signature and be treated as a unique fluorophore. This will work best when autofluorescence is detected in multiple channels and/or has a significantly different intensity than the fluorophores with which it shares channels.

4 Other useful XTensions

4.1 Imaris File Exchange

An online community where user-created XTensions are created and discussed can be found at <http://open.bitplane.com> If you are looking for functionality that Imaris doesn't already have, this is a good place to check to see if there is a solution.

4.2 Additional XTension types

You can also integrate XTensions from Python or Fiji / ImageJ into Imaris. They are generally less popular and worse documented than MATLAB XTensions but they are a possibility.

References

1. McRae TD, Oleksyn D, Miller J, Gao Y-R. Robust blind spectral unmixing for fluorescence microscopy using unsupervised learning. *bioRxiv.* 2019; 797993. doi:10.1101/797993