1 What is an XTension?

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

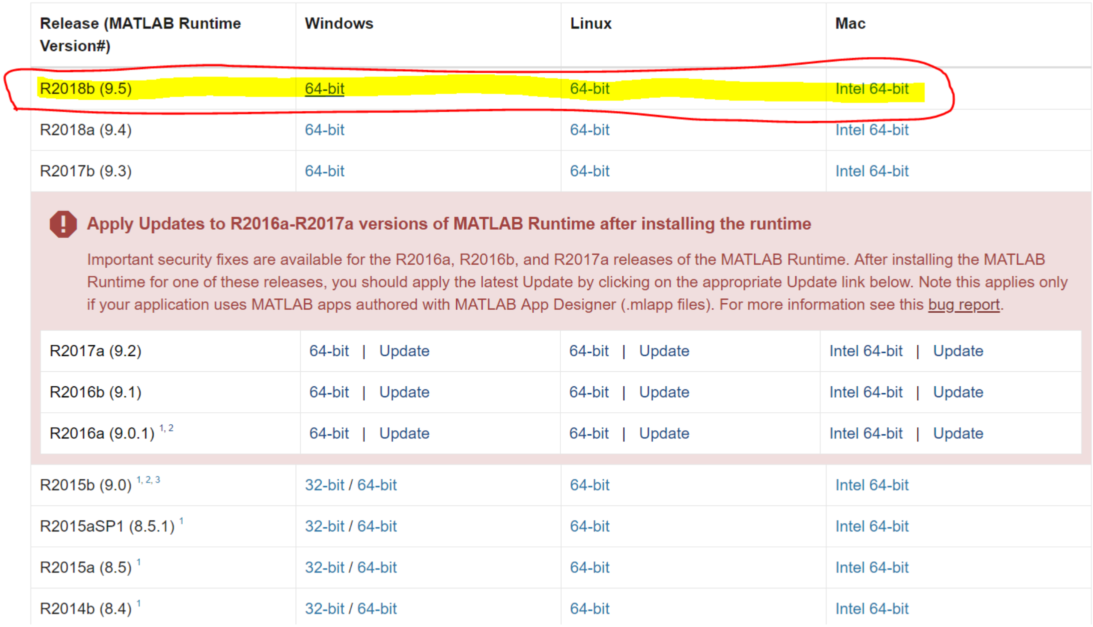
2 Running Imaris XTensions

2.1 Imaris License

In order to run Imaris XTensions, you will need a working Imaris license.

2.2 MATLAB Runtime Compiler

To run XTensions written in MATLAB (which is most of them) you will need a MATLAB Runtime Compiler. No MATLAB license is required to download or use the MATLAB Runtime Compiler. This can be downloaded from mathworks.com/products/compiler/matlab-runtime.html. Imaris XTensions require you to download and install the **R2018b (9.5)** version of the MATLAB Runtime Compiler.



Release (MATLAB Runtime Version#)	Windows	Linux	Mac
R2018b (9.5)	64-bit	64-bit	Intel 64-bit
R2018a (9.4)	64-bit	64-bit	Intel 64-bit
R2017b (9.3)	64-bit	64-bit	Intel 64-bit

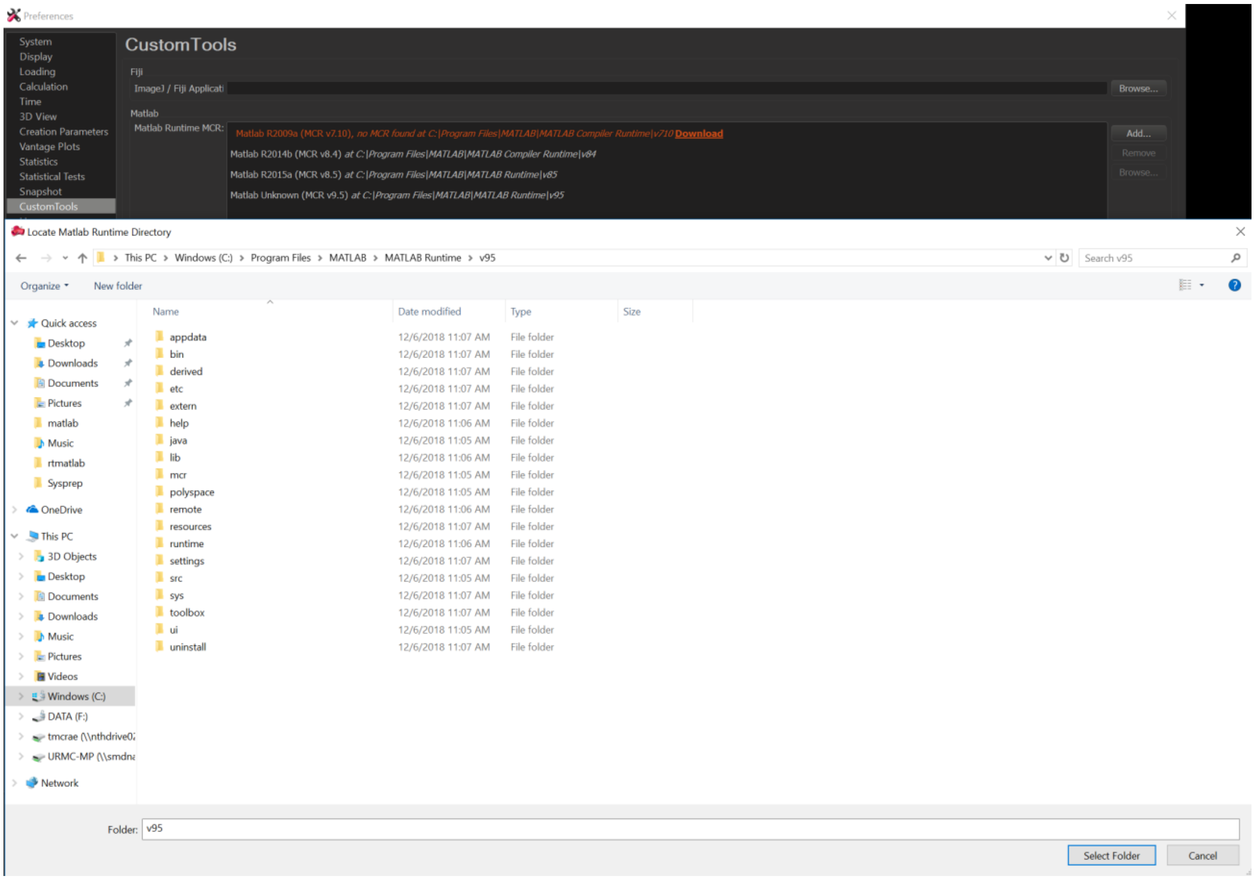
Apply Updates to R2016a-R2017a versions of MATLAB Runtime after installing the runtime

Important security fixes are available for the R2016a, R2016b, and R2017a releases of the MATLAB Runtime. After installing the MATLAB Runtime for one of these releases, you should apply the latest Update by clicking on the appropriate Update link below. Note this applies only if your application uses MATLAB apps authored with MATLAB App Designer (.mlapp files). For more information see this bug report.

R2017a (9.2)	64-bit | Update	64-bit | Update	Intel 64-bit | Update
R2016b (9.1)	64-bit | Update	64-bit | Update	Intel 64-bit | Update
R2016a (9.0.1) [1,2]	64-bit | Update	64-bit | Update	Intel 64-bit | Update
R2015b (9.0) [1,2,3]	32-bit / 64-bit	64-bit	Intel 64-bit
R2015aSP1 (8.5.1) [1]	32-bit / 64-bit	64-bit	Intel 64-bit
R2015a (8.5) [1]	32-bit / 64-bit	64-bit	Intel 64-bit
R2014b (8.4) [1]	32-bit / 64-bit	64-bit	Intel 64-bit

Once you have downloaded the MATLAB Runtime Compiler, you should run the installer with all of the default options.

After you have run the installer, you will have to point Imaris to where the runtime compiler is. To do this, in Imaris, go to *File -> Preferences -> Custom Tools.* Next to where it says “Matlab Runtime MCR” click “Add” to point to the path where you saved the MCR. When prompted, point it to *Program Files/MATLAB/MATLAB Runtime/v95* and click “Select Folder”.



CustomTools

Fiji

ImageJ / Fiji Applicati

Matlab

Matlab Runtime MCR:

```
Matlab R2009a (MCR v7.10), no MCR found at C:\Program Files\MATLAB\MATLAB Compiler Runtime\v710 Download
Matlab R2014b (MCR v8.4) at C:\Program Files\MATLAB\MATLAB Compiler Runtime\v84
Matlab R2015a (MCR v8.5) at C:\Program Files\MATLAB\MATLAB Runtime\v85
Matlab Unknown (MCR v9.5) at C:\Program Files\MATLAB\MATLAB Runtime\v95
```

Locate Matlab Runtime Directory

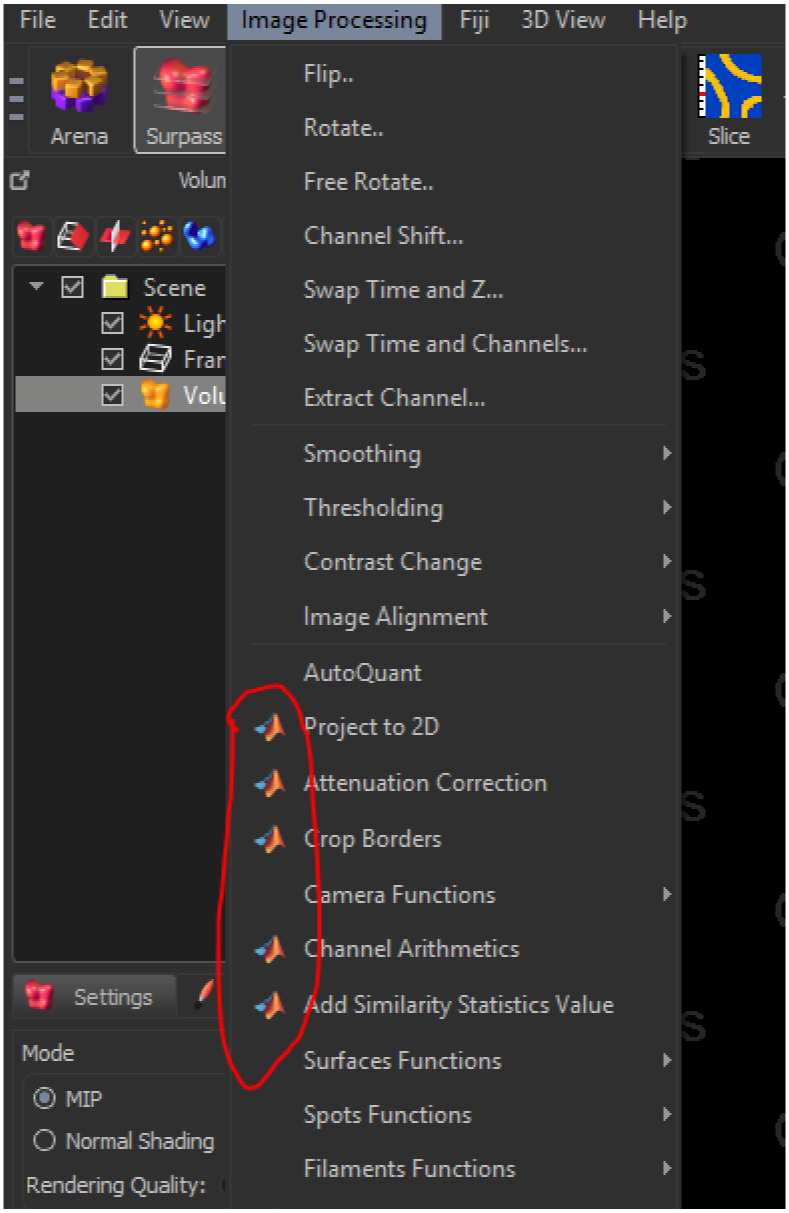
This PC > Windows (C:) > Program Files > MATLAB > MATLAB Runtime > v95

Name	Date modified	Type	Size
appdata	12/6/2018 11:07 AM	File folder	
bin	12/6/2018 11:07 AM	File folder	
derived	12/6/2018 11:07 AM	File folder	
etc	12/6/2018 11:07 AM	File folder	
extern	12/6/2018 11:07 AM	File folder	
help	12/6/2018 11:06 AM	File folder	
java	12/6/2018 11:05 AM	File folder	
lib	12/6/2018 11:06 AM	File folder	
mcr	12/6/2018 11:05 AM	File folder	
polyspace	12/6/2018 11:05 AM	File folder	
remote	12/6/2018 11:06 AM	File folder	
resources	12/6/2018 11:07 AM	File folder	
runtime	12/6/2018 11:06 AM	File folder	
settings	12/6/2018 11:07 AM	File folder	
src	12/6/2018 11:05 AM	File folder	
sys	12/6/2018 11:07 AM	File folder	
toolbox	12/6/2018 11:07 AM	File folder	
ui	12/6/2018 11:05 AM	File folder	
uninstall	12/6/2018 11:07 AM	File folder	

Folder: v95

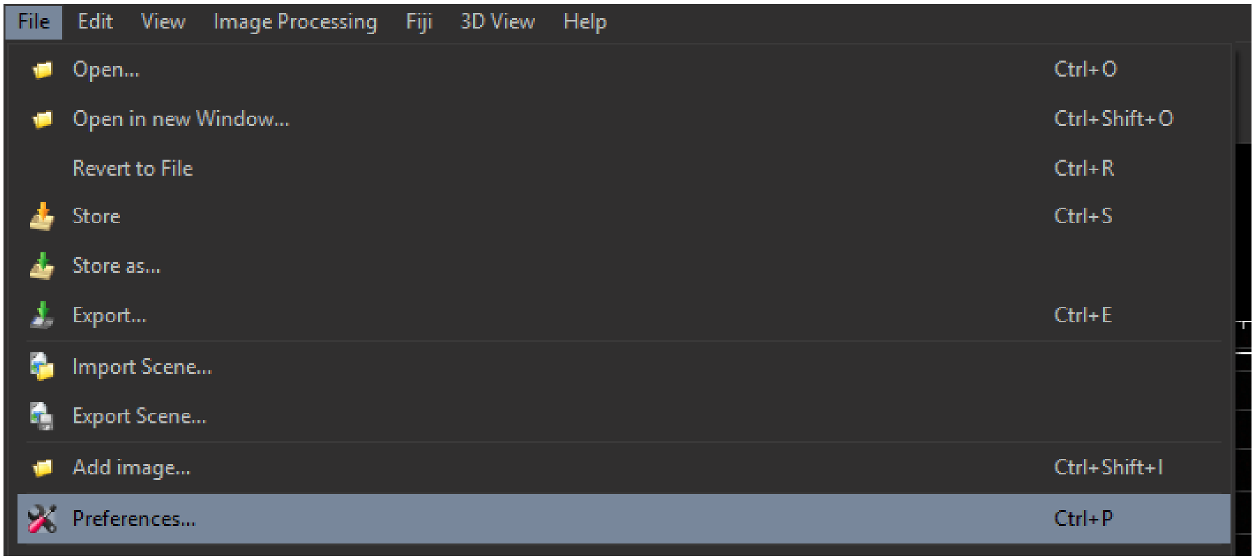
2.3 Using XTensions

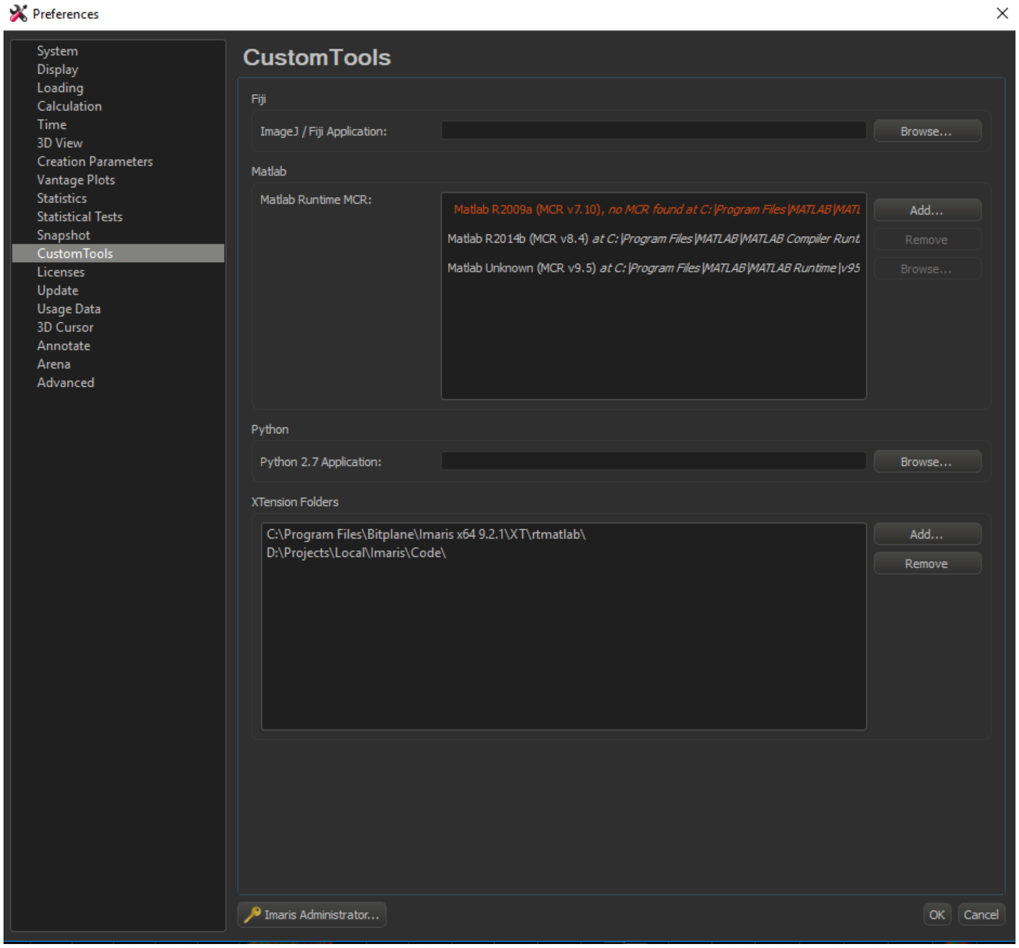
To view available XTensions in Imaris, go into Surpass and click the *Image Processing* dropdown. MATLAB XTensions will be visible with a MATLAB logo next to them. When Imaris is properly connected to MATLAB, this logo will be in color. Otherwise it will be grayed out. Imaris will come with certain XTensions already installed and available to you.



2.4 Adding Custom XTensions

When you have a MATLAB XTension in the form of a .exe file, you can add it to the XTensions Imaris can see. In order to do this, you will have to move the .m file into a folder where Imaris is looking for XTensions. To see where Imaris looks for XTensions, go to *File -> Preferences -> Custom Tools*.





CustomTools

Fiji

ImageJ / Fiji Application:

Matlab

Matlab Runtime MCR:

```
Matlab R2009a (MCR v7.10), no MCR found at C:\Program Files\MATLAB\MATL
Matlab R2014b (MCR v8.4) at C:\Program Files\MATLAB\MATLAB Compiler Runt
Matlab Unknown (MCR v9.5) at C:\Program Files\MATLAB\MATLAB Runtime\v95
```

Python

Python 2.7 Application:

XTension Folders

```
C:\Program Files\Bitplane\Imaris x64 9.2.1\XT\rtmatlab\
D:\Projects\Local\Imaris\Code\
```

Where it says “XTension Folders”, it will show the path to the folders it looks in when looking for XTensions. You can move your .m file into a folder it is already looking at. Modifying the folder it’s looking at can sometimes require administrative privileges. If this is the case and you do not have administrative privileges, you can add the folder your .m file is already in to where Imaris looks by using the “Add” button.

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 on Box here: https://rochester.app.box.com/file/363188048106 and on GitHub here: https://github.com/tristan-mcrae-rochester/Multiphoton-Image-Analysis/blob/master/Spectral%20Unmixing/Code/XTSpectralUnmixing.exe

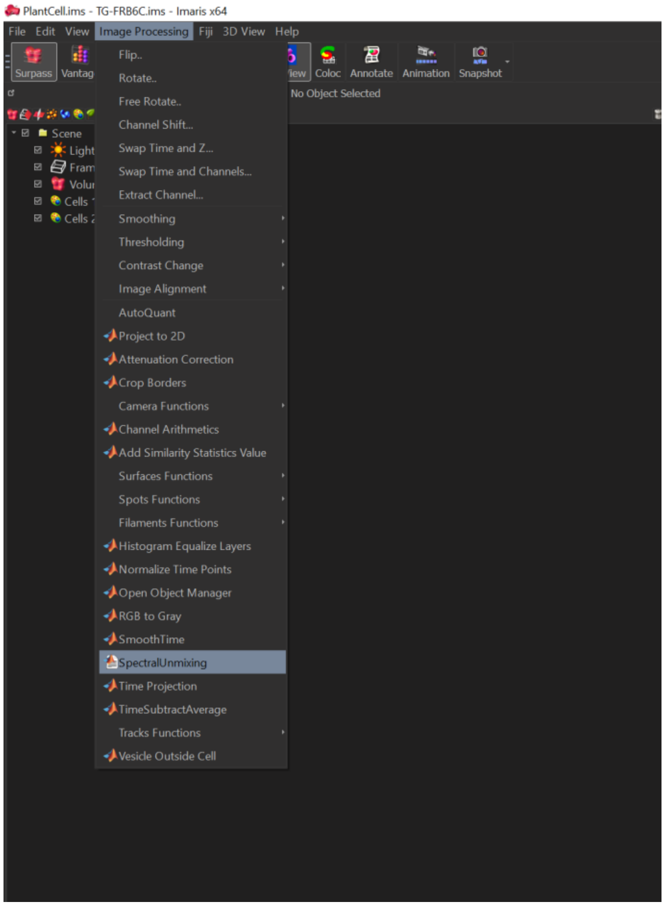
Download the XTSpectralUnmixing.exe file linked above from either source and save it to your computer.

The spectral unmixing application also requires a matching XML file to be saved in the same folder. The XML file for XTSpectralUnmixing is available for download in the box here: https://rochester.app.box.com/file/363271106451 and on github here: https://github.com/tristan-mcrae-rochester/Multiphoton-Image-Analysis/blob/master/Spectral%20Unmixing/Code/XTSpectralUnmixing.XML

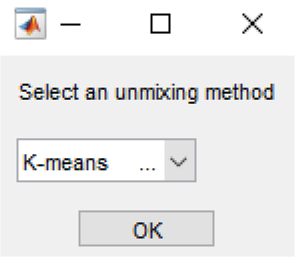
Using the instructions from section 2.4, point Imaris to the location where you saved the XTension.

3.2 Running the XTension

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



After you select Spectral Unmixing, you will be prompted to choose your unmixing method. Currently only k-means unmixing is supported in this XTension but other methods will be available in the future for different unmixing scenarios.

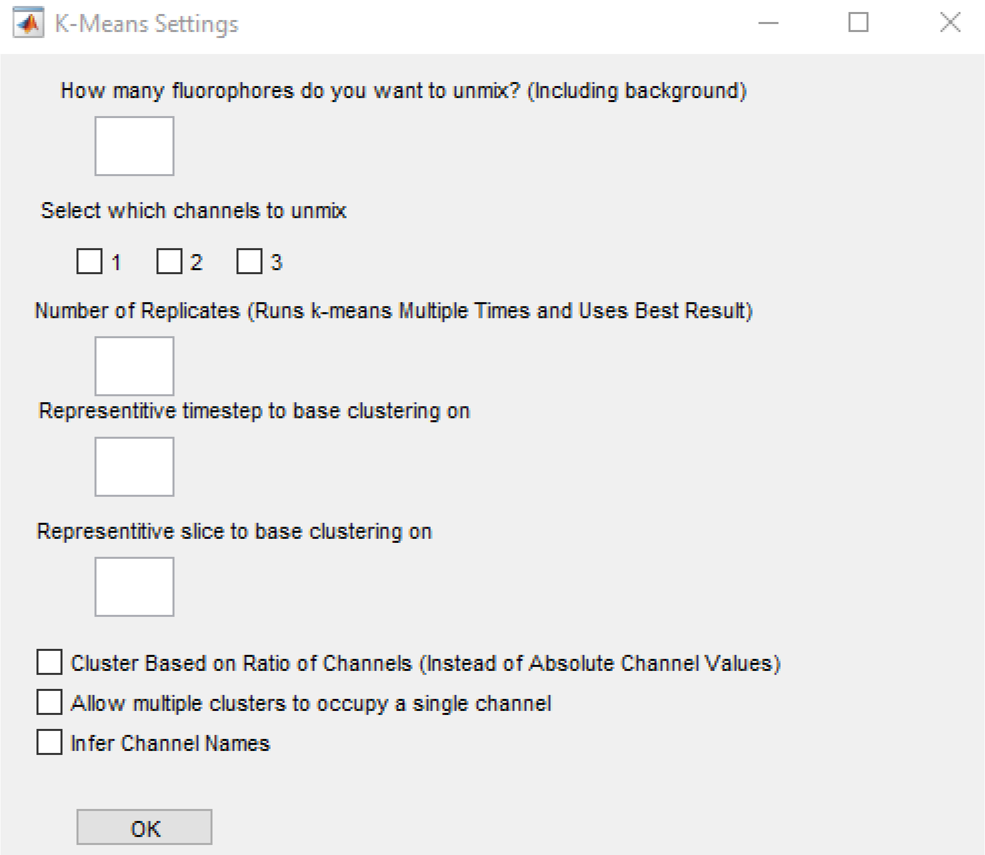


Select an unmixing method

K-means ...

OK

After selecting k-means, you will then be asked about a number of options:



K-Means Settings

How many fluorophores do you want to unmix? (Including background)

Select which channels to unmix

□ 1 □ 2 □ 3

Number of Replicates (Runs k-means Multiple Times and Uses Best Result)

Representitive timestep to base clustering on

Representitive slice to base clustering on

□ Cluster Based on Ratio of Channels (Instead of Absolute Channel Values)

□ Allow multiple clusters to occupy a single channel

□ Infer Channel Names

OK

How many fluorophores you want to unmix (including background) – The number of clusters the algorithm will create. 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 +1 for a background cluster. 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. Note! This is different than previous versions where a background cluster was automatically added behind the scenes. 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 done successfully, 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.

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. I recommend starting with something around 25 and changing it as needed for faster or higher quality results.

Representative timestep to base clustering on – 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.

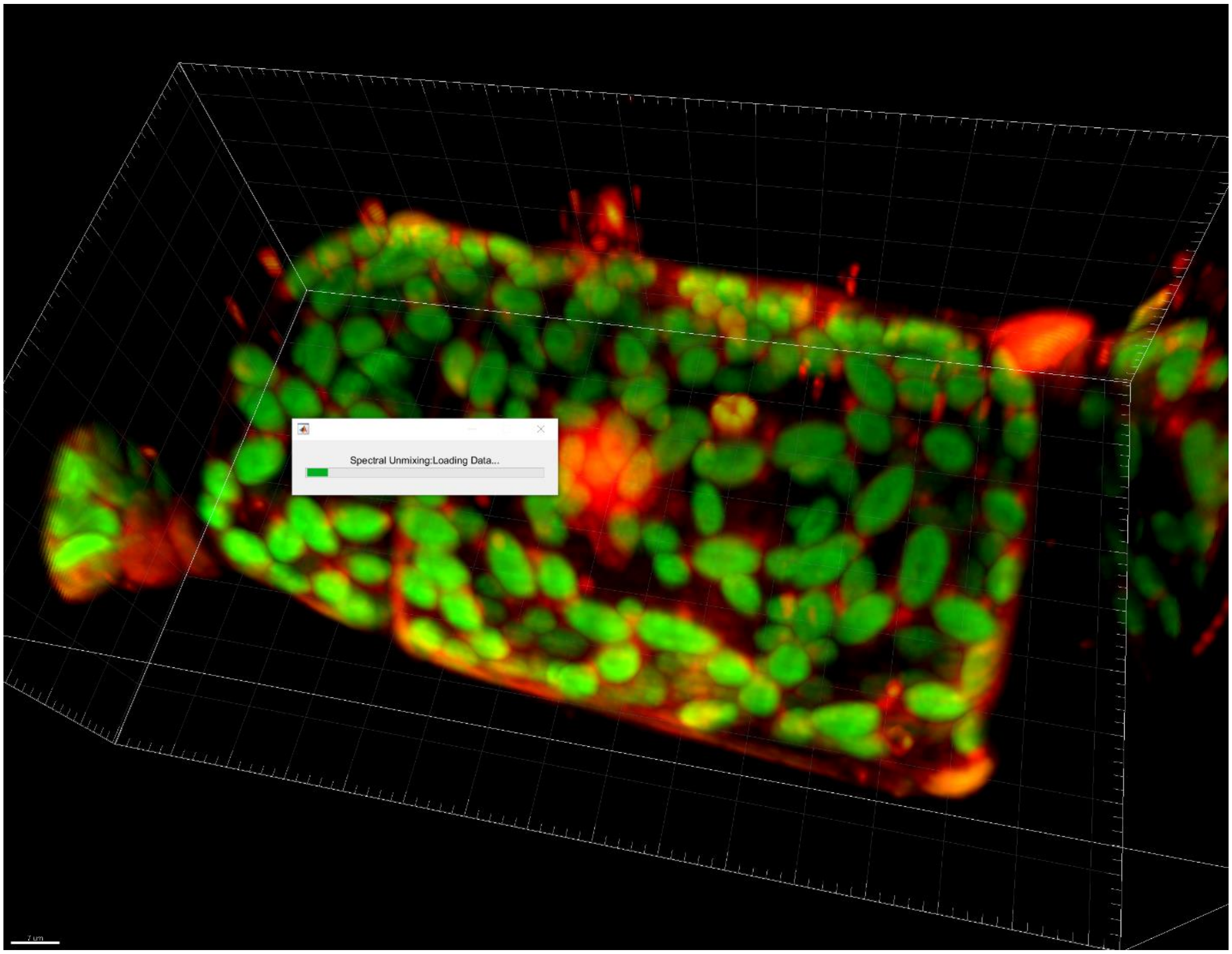
Representative slice to base clustering on – Same as representative timestep but for an axial slice.

Cluster 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.

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.

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.

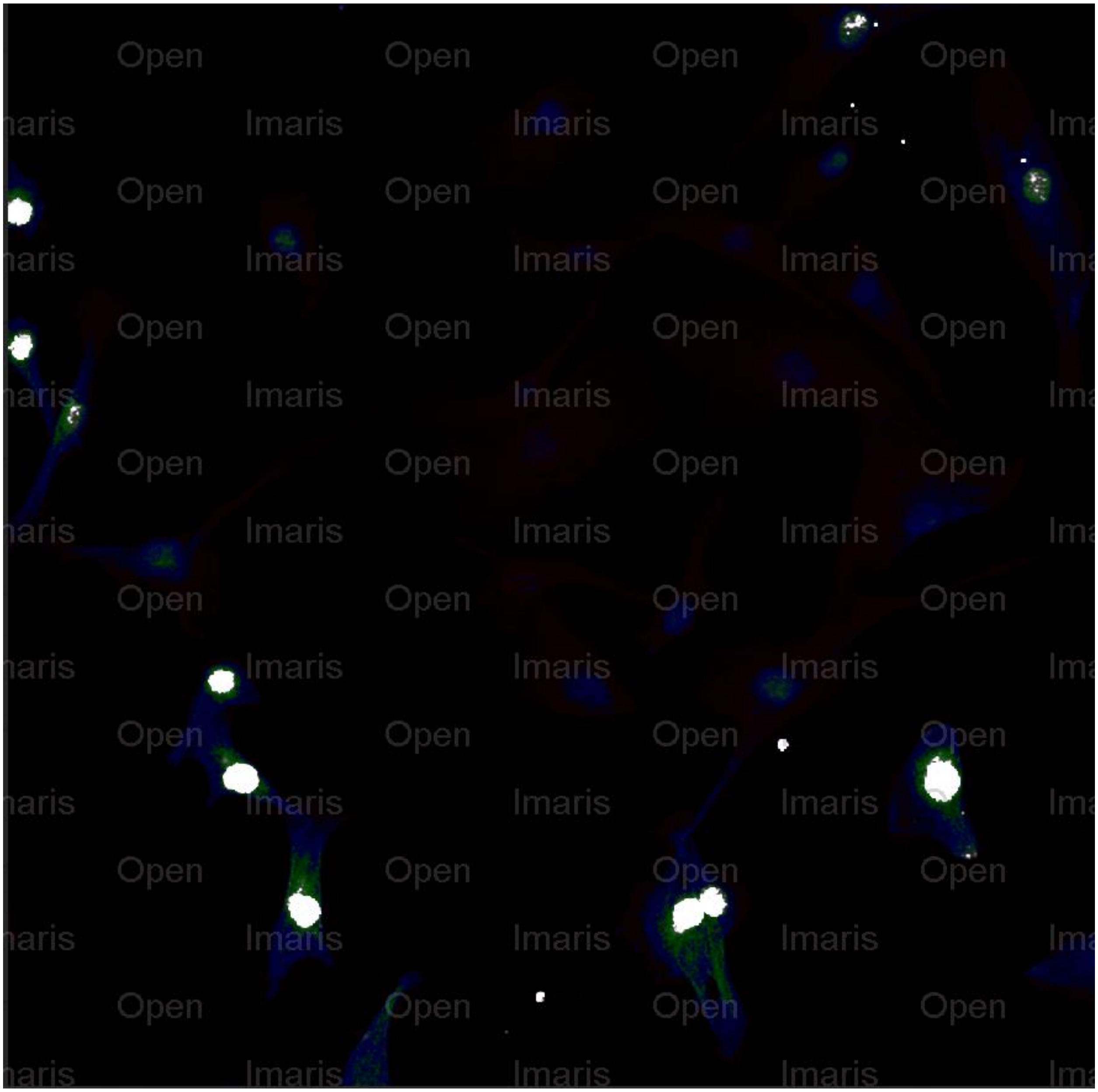
The progress of k-means will be displayed in a window which will pop up automatically after a minute or so when you start the XTension.



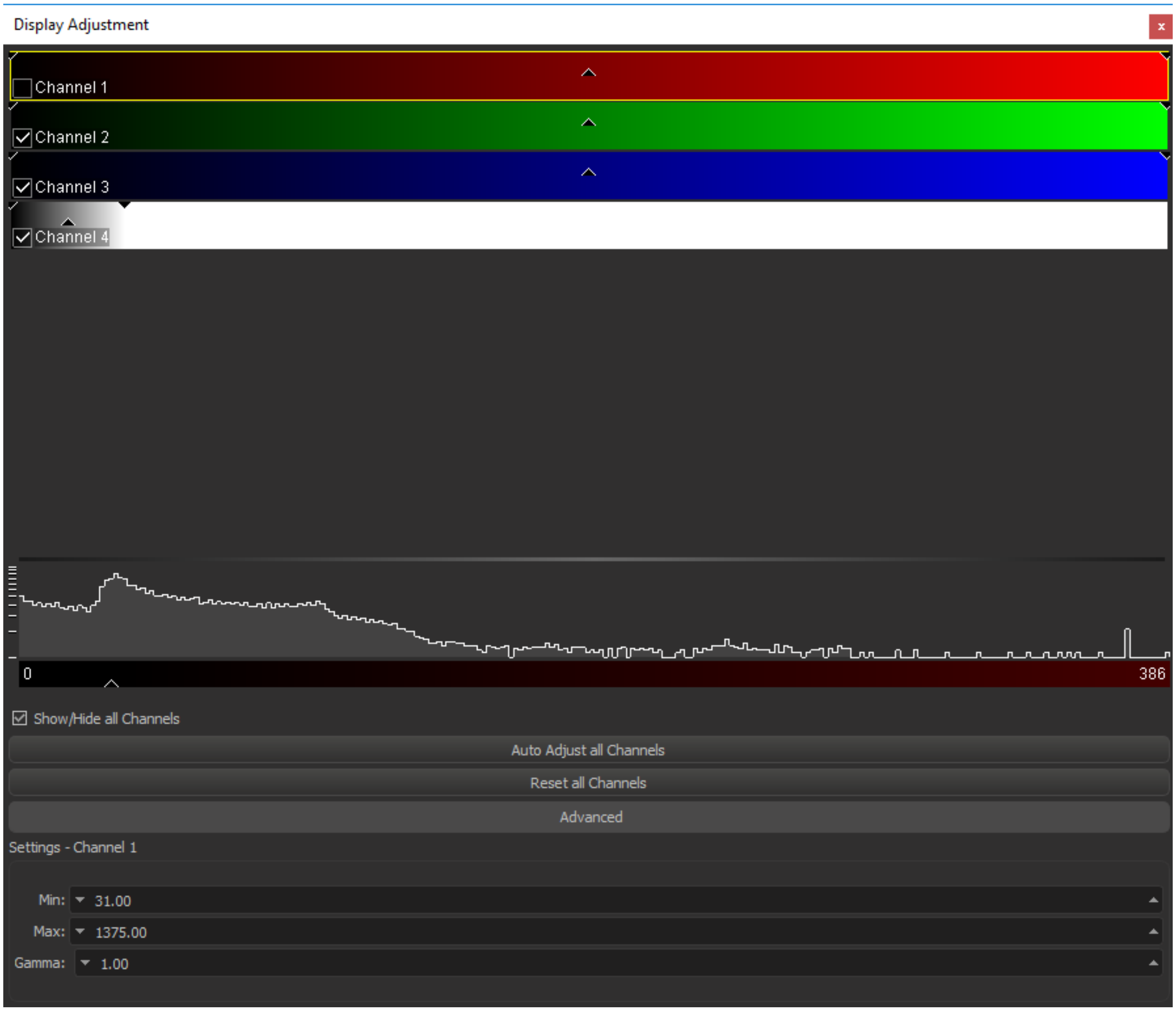
Spectral Unmixing:Loading Data...

3.3 Cleaning up the Results

After running, the Spectral Unmixing XTension will automatically return the results to Imaris. If you don’t use the “infer channel names” option, Imaris won’t automatically know how to color or scale the channels so you may have some weird looking outputs at first. This is easily fixable with some manual adjustment.

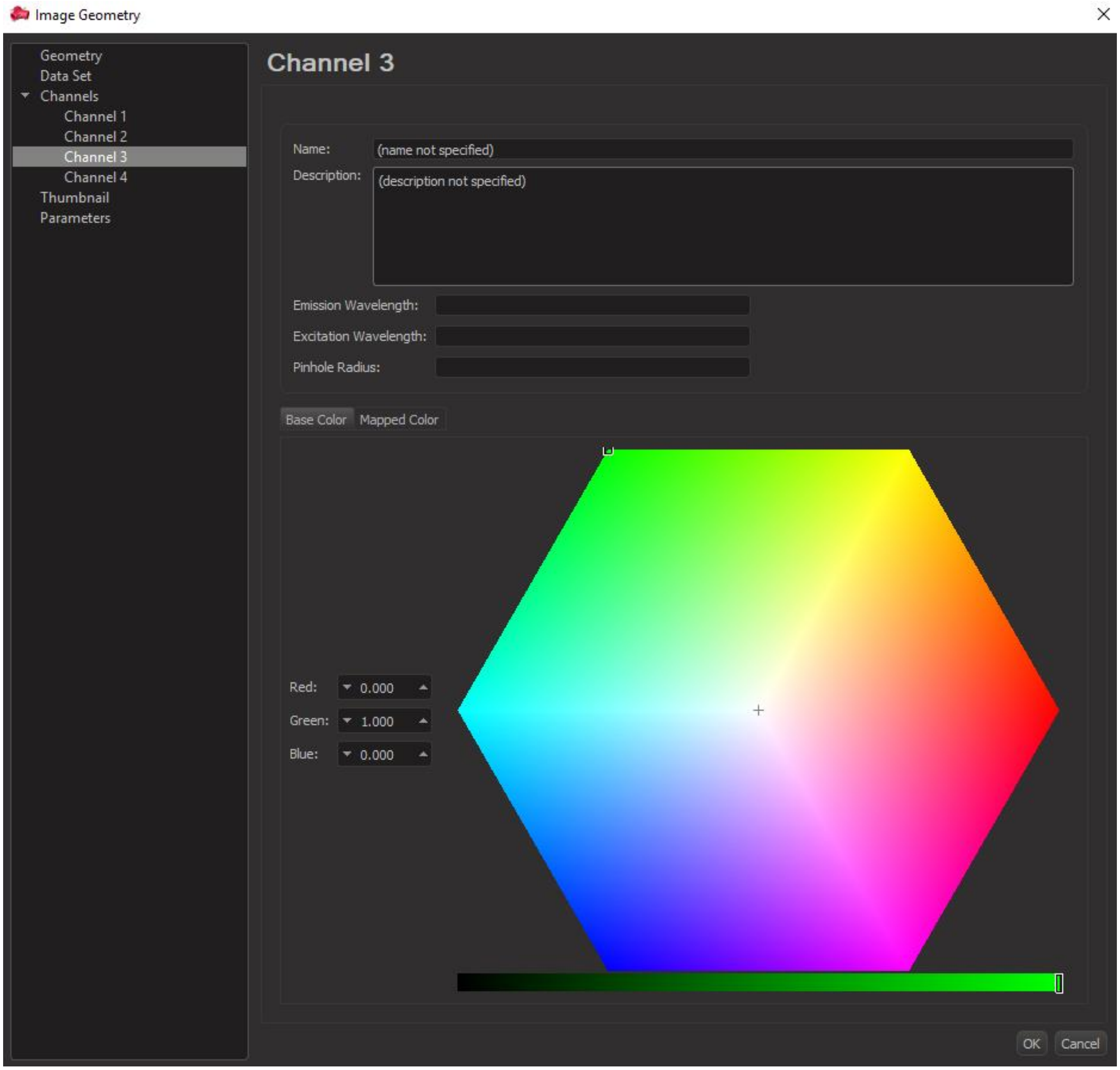


You can manually change the appearance of the channels with Imaris's Display Adjustment tool which can be brought up with ctrl+d on Windows or with *Edit -> Show Display Adjustment*. Here you can adjust channel intensities and turn off the background channel by unchecking it's box.



Min: 31.00
Max: 1375.00
Gamma: 1.00

To make your new channels reflect the colors of the old ones, you can click on "Channel #" and then use the "Base Color" tab of the ensuing pop-up to change the color. Do this for each channel whose display color you want to change. It won't change the data, just its display in Imaris. If you have multiple clusters for the same fluorophore, this is a good time to combine them visually by setting them to be the same color.



Channel 3

Name: (name not specified)

Description: (description not specified)

Emission Wavelength:

Excitation Wavelength:

Pinhole Radius:

Red: 0.000

Green: 1.000

Blue: 0.000

4 Other useful XTensions

4.1 An online community where user-created XTensions are created and discussed can be found here: 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 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.