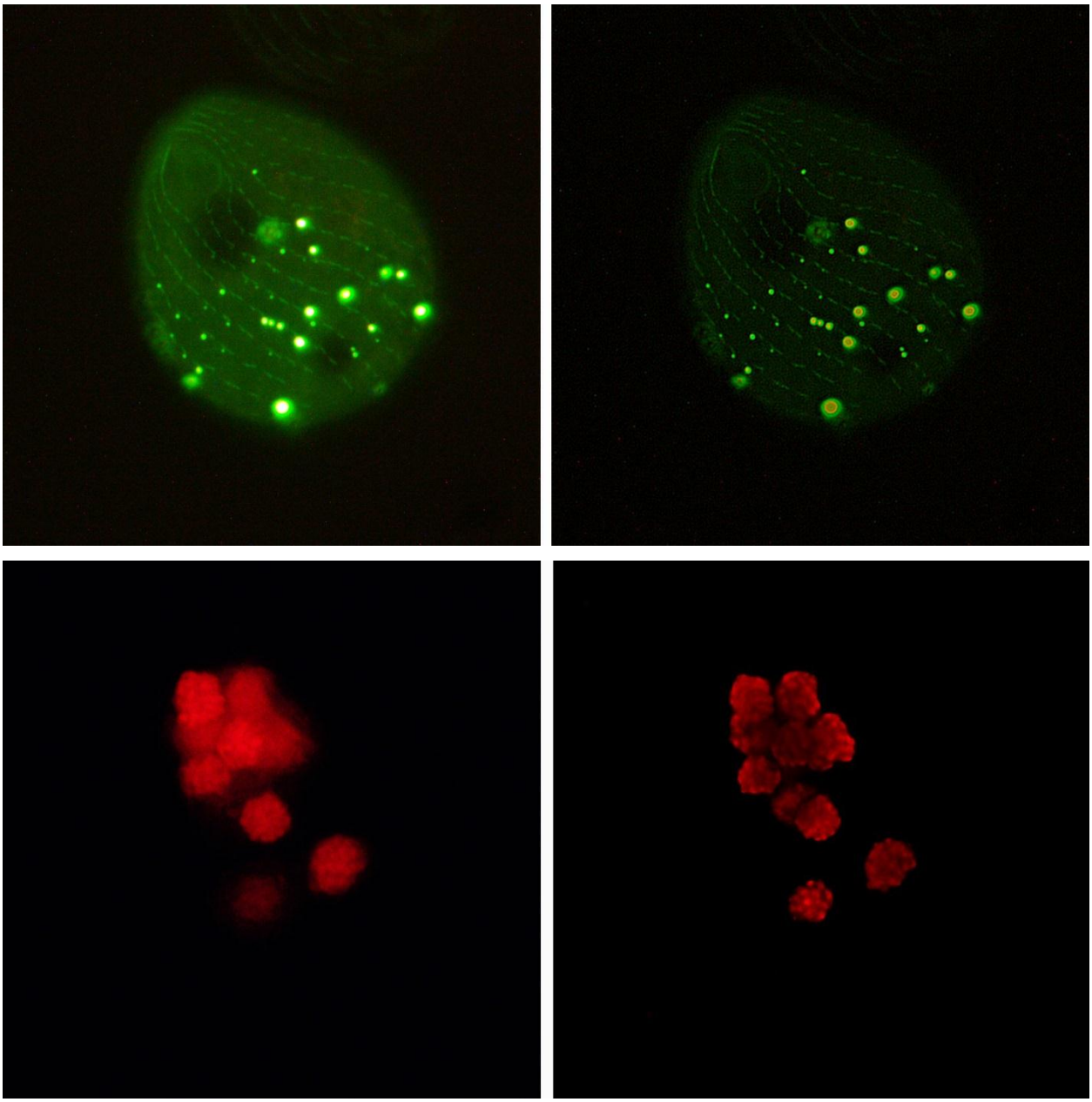
2D and 3D Fluorescence Deconvolution Manual

Asa Giannini and John Giannini*

Biology Department, St. Olaf College, 1520 St. Olaf Avenue, Northfield, MN 55057

* Email: giannini@stolaf.edu

December 15, 2016



Introduction

Image Deconvolution is the general process of using mathematical algorithms to reduce the noise and distortion present in photographs. In the field of Fluorescence Microscopy, much of the distortion and blur comes from the presence of out of focus light from multiple planes. While the microscope is focused on a given layer of the fluorescent slide, the other layers above and below will give off light causing a blurring effect. As such, fluorescent images appear soft and diffused, losing much of the finer detail.

A Point Spread Function (PSF) is the pattern of how a point of light appears spread out and diffuse, given the conditions of a specific microscope, CCD camera, and associated light diffraction. The idea behind deconvolution is that, since images are made up of points, a PSF can be used to calculate what a given image should look like by removing the effects of distortion. In practice, one takes a series of images, called a stack, at different levels of focus effectively slicing the image into different planes. This stack is then processed via an algorithm which processes each slice of the image to remove distortion based on the PSF.

The resulting image stack will appear dimmer, as light has been removed but will also be sharper as the unblurred image is extrapolated. This deconvoluted stack can then be collapsed (in a process called Z-Projection) into a single image which will now appear to have added dimension as well as sharpness. The resultant image will now contain useful three dimensional visual data not present in the original photographs.

If a single image needs to be deconvoluted instead of a 3D stack, a similar but simpler process applies where the PSF is used with a given algorithm to remove light diffraction. This process (called 2D Deconvolution) is advantageous in terms of time and ease since a single image is quicker to acquire and less computationally taxing to deconvolute. However, since much of the light comes from multiple layers of a given image, the results in 3D deconvolution will invariably render cleaner, more-accurate deconvoluted images than those of 2D deconvolution.

The goal of this manual is to provide a simple, free means of performing both 2D and 3D deconvolution. This manual contains all the pertinent information for performing basic deconvolution including creation of a Point Spread Function and a Z-projection. Because some estimation is involved, it is recommended

to try these methods out ahead of time to get the results desired for your situation. None of the methods presented are new, but they have been modified, compiled, and simplified for easy implementation.

While the software presented provides many possible algorithms for use in deconvolution, we focus on two which we have had success with. For 3D deconvolution, we use the Richardson-Lucy system. This method had been empirically shown to provide accurate visual information when the process converges. In other words, with an accurate PSF and good photos, the Richardson-Lucy system tends to give good results. For 2D deconvolution we used the Generalize Tikhonov (reflexive) algorithm which creates a regularization matrix to aid in processing the image. The microscope we used was the Olympus CH40. Our camera was OMAX 14.0MP USB3.0 Digital USB Microscope Camera available here. http://www.amazon.com/OMAX-Microscope-Advanced-Calibration-Compatible/dp/B00FG89B3S

Acquiring and installing the software

For this project you will need ImageJ as well as three plugins for ImageJ. Fiji will also work, but ImageJ is recommended as it has fewer bugs in our experience.

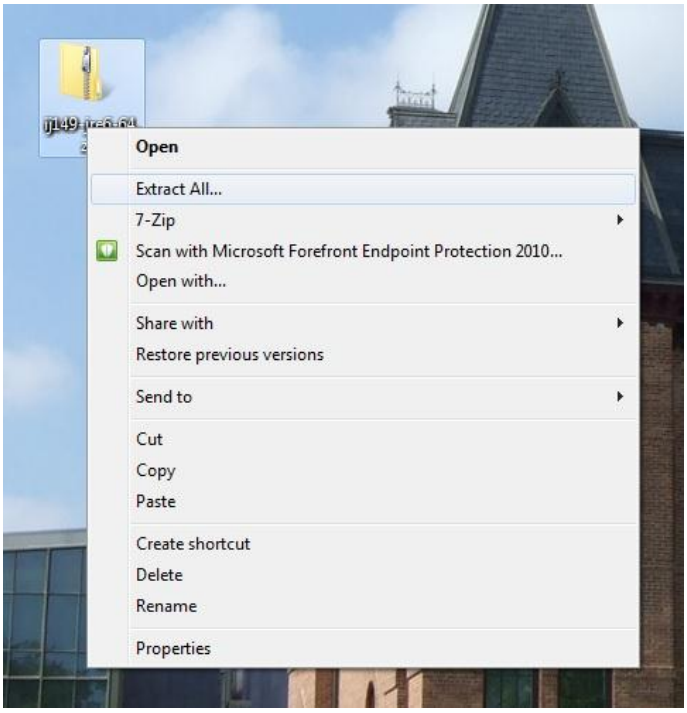


Figure 1 The command for extracting ImageJ

You will first need to download ImageJ from http://ImageJ.nih.gov/ij/download.html. The installation file will be in the form of a .zip file. Move the downloaded file to a folder or your desktop and then right click and select 'extract all (Figure 1). When prompted, uncheck the box that says "Show extracted files when complete" and then select 'extract'. A new folder with the same name as the downloaded file should appear after a few minutes of extraction. Inside that folder is a folder marked 'ImageJ' which contains 'ImageJ.exe' which will launch the ImageJ application.

With ImageJ downloaded and installed, you will need to download and install three plugins. The first is called 'DeconvolutionLab' and is available from http://bigwww.epfl.ch/algorithms/deconvolutionlab/. Follow the instruction on that page, as indicated in Figure 2, to download 'Deconvolutionlab.zip'. Move this file into the folder 'ImageJ' from the step above. Then move it into the folder marked 'plugins' and extract the files by once again right clicking and selecting 'extract all'. You can now delete the original zip file as a new extracted folder has been created.

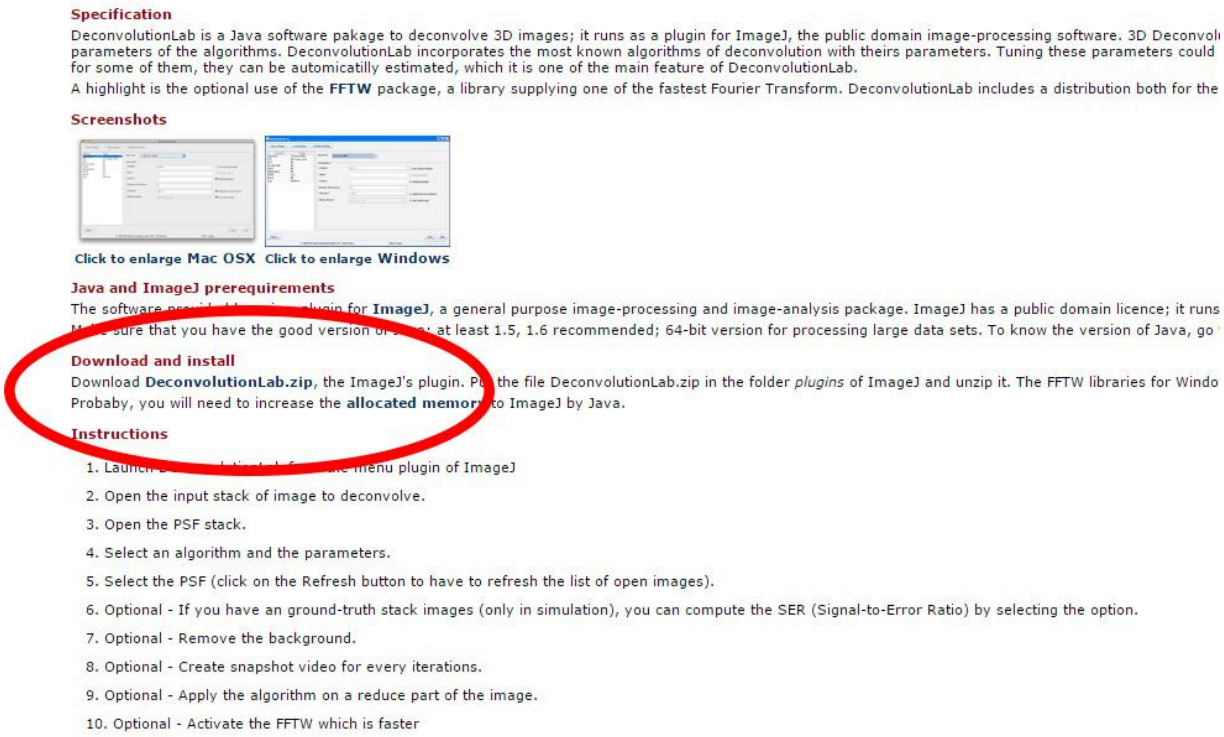


Figure 2 The zip files on the DeconvolutionLab page

Repeat these steps with the zip folder of 'Parallel Spectral Deconvolution" from http://fiji.sc/Parallel_Spectral_Deconvolution. The final plugin is in a slightly simpler form that will not need to be extracted. Download 'Diffraction_PSF_3D.class' from http://fiji.sc/Diffraction_PSF_3D and put it into the plugins folder. With these three plugins downloaded, open your ImageJ.exe application. An interface will appear as shown in Figure 3.

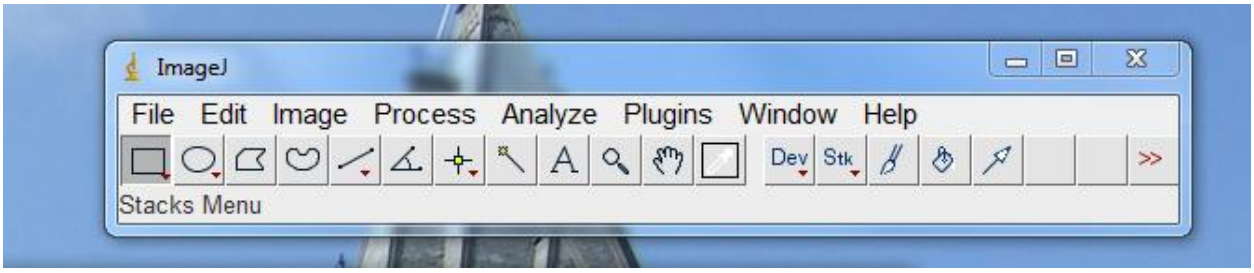


Figure 3 the ImageJ interface

Select the 'Plugins' menu at the top of this bar and then click on 'Install…"(Figure 4). A window will open for you to select files to install. Find the ImageJ folder and then the Plugins folder inside it. Select the folder marked 'Deconvolution Lab' and then select from inside it the second folder marked 'Deconvolution Lab'. Finally double click on 'DeconvolutionLab_.jar'. To repeat the plugin file you want to select is **ImageJ>Plugins>Deconvolution Lab>Deconvolution Lab>DeconvolutionLab_.jar**. After double clicking on this file, you will be prompted to save a copy, click save and the plugin will be installed.

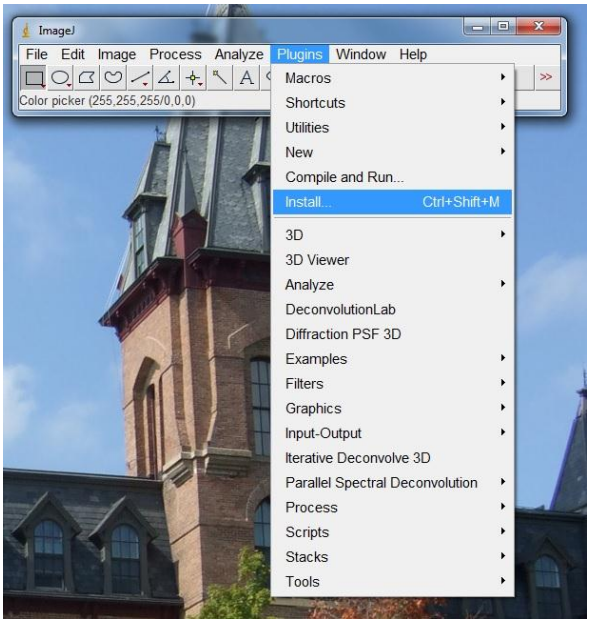


Figure 4 Installing plugins with ImageJ

Perform this same process on the following plugin **ImageJ>Plugins>parallel_spectral_deconvolution-1.9-bin>ParallelSpectralDeconvolution>parallel_spectral_deconvolution-1.9.jar**

And also on

ImageJ>Plugins>parallel_spectral_deconvolution-1.9-bin >ParallelSpectralDeconvolution>parallelcolt-0.7.2.jar

Finally perform this process on **ImageJ>Plugins>Diffraction_PSF_3D.class.** For this file note that when the computer prompts you to save a file and you click save you will be further prompted that this file already exists and do you want to replace it. Click 'yes'. Close ImageJ and open it again to allow installation of plugins to finalize. You should now see three new entries on your ImageJ plugins menu as shown in Figure 5.

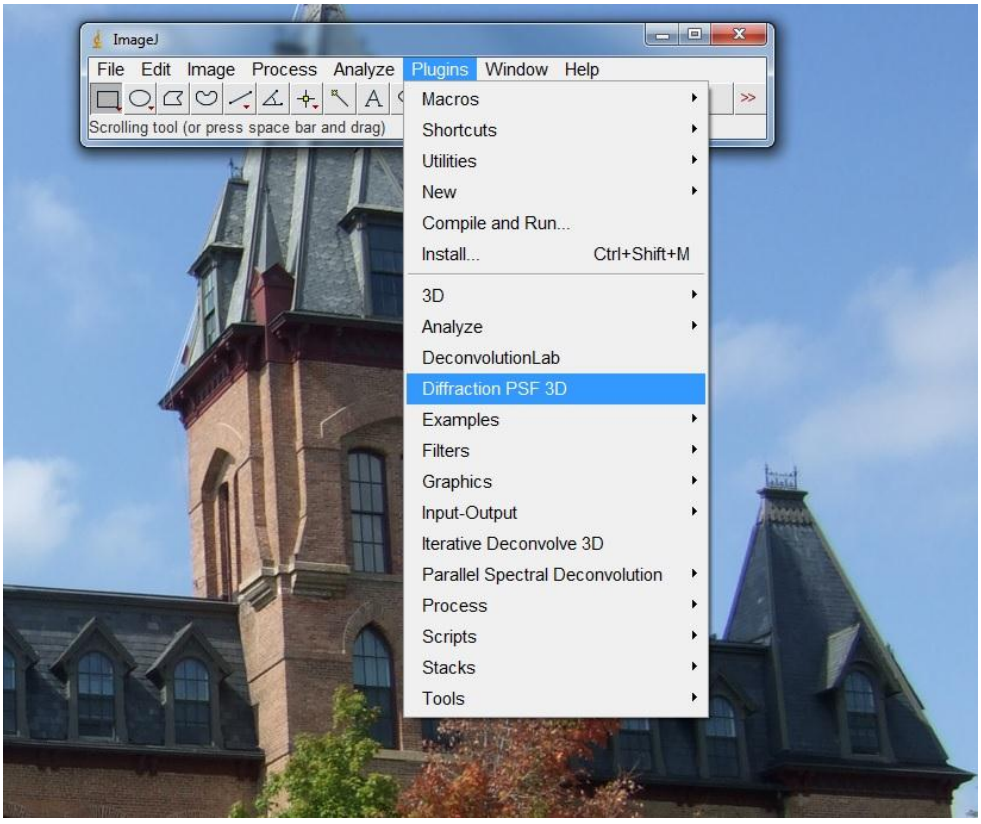


Figure 5 Plugins menu with installed plugins

Try clicking on each to make sure it opens (if it does not, close ImageJ, delete the problematic plugin from the plugins menu and re-download and install).

Creating a PSF Image:

In deconvolution, the key element needed, apart from good photographs, is the Point-spread Function which defines how a given point of light is imaged by your camera and microscope. For the actual process of deconvolution, you need an image (or series of images) that represents the point spread function (which will look like a black image with a white spot or speck in the center). The Point-spread Function is notoriously tricky to calculate empirically, so we opted to create a theoretical PSF using the 'Diffraction PSF 3D' plugin for ImageJ. Open this plugin from your ImageJ menu, you should see the window pictured in Figure 6.

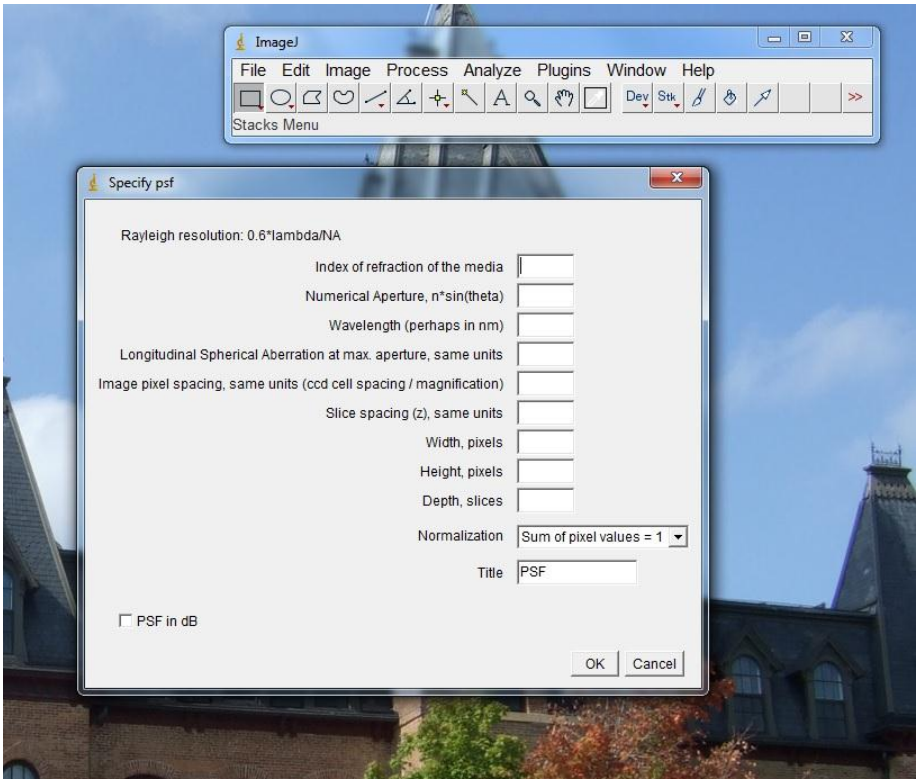


Figure 6 Diffraction PSF 3D Plugin window

There are 11 slots that need to have information put in to create a PSF. We will go through them one at a time:

1. Index of refraction of the media refers to the unitless index of refraction for whatever medium the slide is being viewed under. Often this will be air if no immersion fluid is needed (the index of refraction for air is 1.00029), but if you are viewing through immersion oil or any other

substance the index of refraction will have to be located. Often it is printed on the bottle or available online at the very least.

2. Numerical aperture is a numbered aspect of a given microscope's lens. It should be printed on the lens itself along with what the given magnification is (see photo). Note that as a consequence, images gained via different lenses require different PSFs.
3. Wavelength of the emission fluorescence of the fluorophore in nano-meters. Typically this will be in the 400-600 nm range.
4. Longitudinal spherical aberration is a complex and involved aspect of microscopes that is beyond most lab's resources to estimate. However, an estimate of 0.00 is fairly safe and renders good results in our experience.
5. The CCD cell spacing is perhaps the most involved number required, but also perhaps the most important. A quality ccd camera should list, either online or in its manual, what its pixel size is in micrometers (µm).Finding this number can be difficult for lower quality cameras, but anything of reasonable resolution for performing deconvolution should provide this. You then divide this number by the magnification of the lens. For example If the pixel size of your given camera is 1.4 µm x 1.4 µm, and you were using a 40x lens, you would divide 1.4 µm by 40 to get .035 µm. Now, this is the correct number but the units should be the same as wavelength (nm) so you must multiply by 1000. In our case then the final number we would put into the program for CCD cell spacing is 35.
6. This number becomes important during 3D deconvolution but is irrelevant for 2D (simply put 0). It is the space between the images in your image stack in the same units as CCD cell spacing and wavelength. For example, if I took a photo every .05 µm (50nm), then I would put 50 in this space. For details on how to find this number see "Creating a Z stack". For
7. Width of the image(s) to be deconvolved in pixels. For a 300X500 pixel image, enter 300. Note that if you crop your image, this is the cropped dimensions.
8. Height of the image(s) to be deconvolved in pixels. For a 300x500 pixel image, enter 500. Again if you crop your image, this is the cropped dimensions.
9. This is the number of images in your image stack. For 2D convolution this number is always 1, for 3D simply take the number of images in your image stack. For more information see the section below on 3D deconvolution.
10. This option, a dropdown menu instead of a blank, can simply be left at its default "Sum of pixel values = 1"

11. This space is simply the name of your PSF, which should be noted so that it can be matched up with your given image or image stack.

When you have found and entered all of these numbers hit ‘ok’. After a minute (slightly longer for large image sizes) a mostly black image will appear with white in the center as shown in Figure 7. Click once on this image and then select **file>save as…** from the ImageJ menu for later use. Note that this PSF can be used for any images taken with the same lens, medium, camera, and image size as you specified before. If any of these qualities are different for an image you wish to deconvolute, you should be sure to generate a new PSF image.

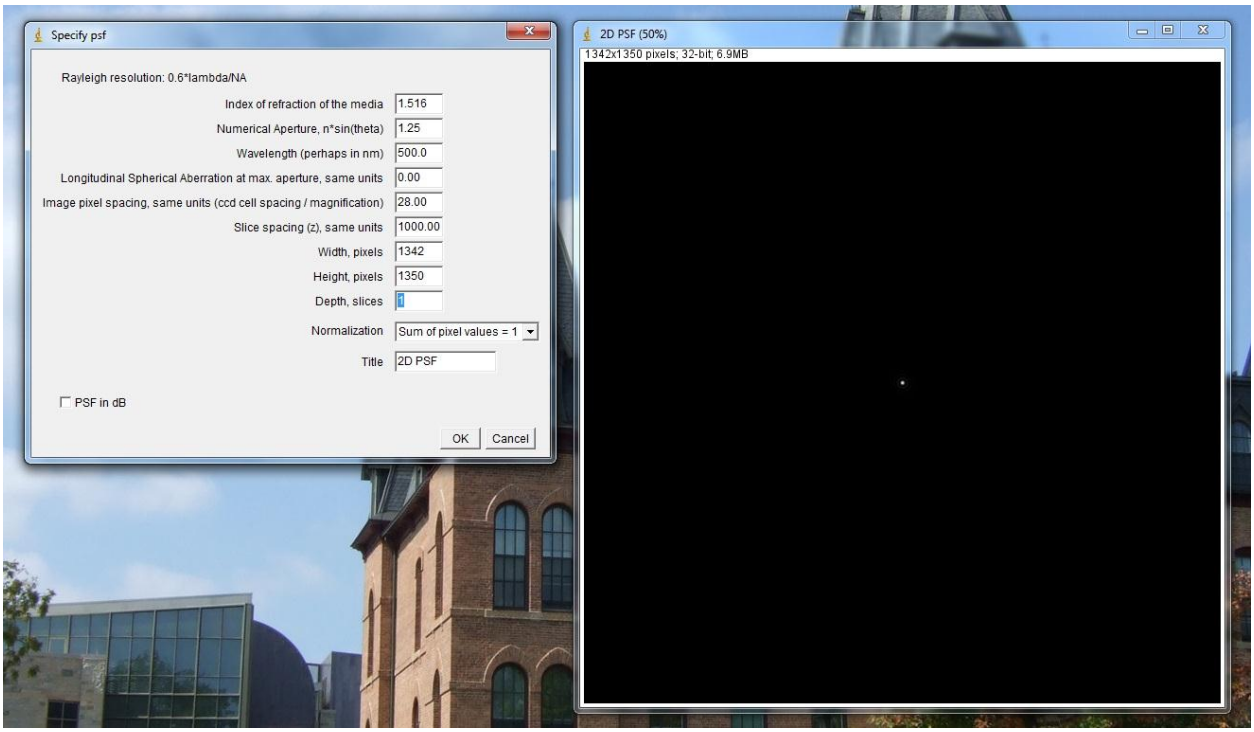


Figure 7 A generated PSF

2D Deconvolution

We will now layout the process for performing 2D convolution, i.e., deconvolution on a single image. The plugin we use only works on images that are in greyscale. There is a way to operate on images in color which we recommend in many cases, especially if more than a single color is present. As such, we will first layout the process for converting an image into grey scale and deconvoluting it and then lay out the extra steps required to deconvolute a color image.

I recommend first making a copy of your image so that you have the original if you need it later. Open the image (or copy) you wish to deconvolute in ImageJ. Select the 'Image' drop down menu and click on 'type'. Then select '8-bit' and your image should become a greyscale image. You can then save it and continue.

Open your PSF image and your greyscale image in ImageJ. From the plugins menu select 'Parallel Spectral Deconvolution' and then click '2D Spectral Deconvolution…". You will see a window with multiple dropdown menus as shown in Figure 8.

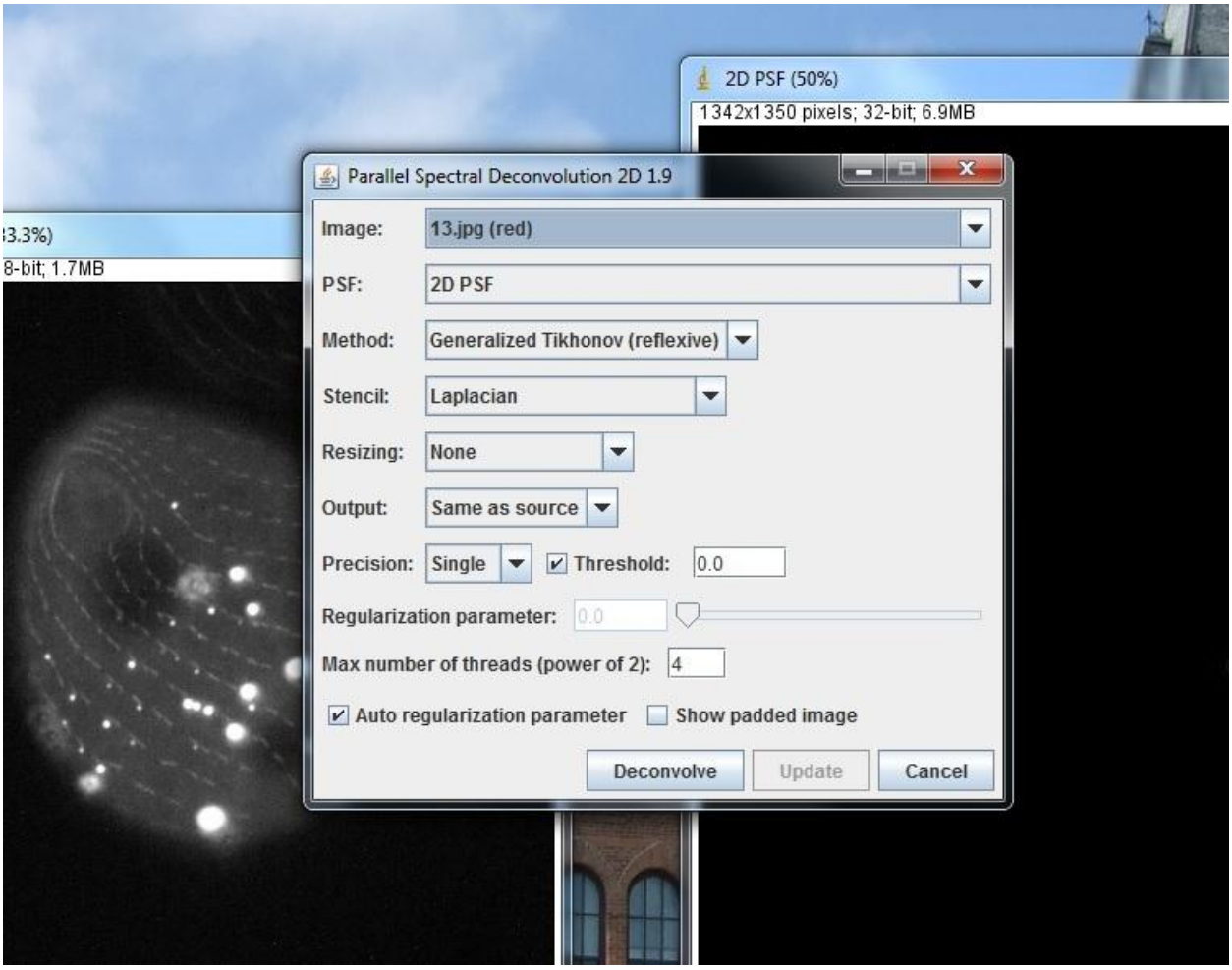


Figure 8 The Parallel Spectral Deconvolution Plugin

In the image drop down menu, select the greyscale image, in the PSF dropdown, select your PSF image. All settings can be left at their defaults (pictured above). Click 'deconvolve' and in 3-7 minutes (longer

for larger images) a deconvoluted image will appear. You can now select the image and save it using ImageJ.

Color 2D Deconvolution

This process relies on the process outlined above so be sure to read and understand that before proceeding. Since the plugin can only handle deconvolution of greyscale images, we will simply split the original color image into three greyscale layers (one for red, one for green, one for blue), deconvolute them separately, and then put them back together into a color image after the process is done.

To begin, open the color image you wish to deconvolute in ImageJ. Select the 'Image' dropdown menu and click on the 'color' tab. Then select 'split channels'. You should now have three greyscale images open labeled with 'red', 'green', and 'blue' to signify which layer they are as shown in Figure 9. Save each of these separately and perform the 2D deconvolution process on each of them. When they have been deconvoluted (and saved with names to remind you of which is 'red,','green', and'blue'). Note that if your image was CMYK type instead of RGB the exact same process applies though you will have four layers instead of three.

Figure 9 An image split into red, green, and blue color channels

From the same Image dropdown menu as before, select 'color' and then 'merge channels'. Then select the appropriate image for the different color categories and leave the others with the *none* option

(Figure 10). Finally, make sure the box for 'Create Composite' is not checked. Click 'ok' and the three images will be merged back into one color image, now deconvoluted. This process seems a little cumbersome at first but becomes straight forward with a little practice and provides some very nice results.

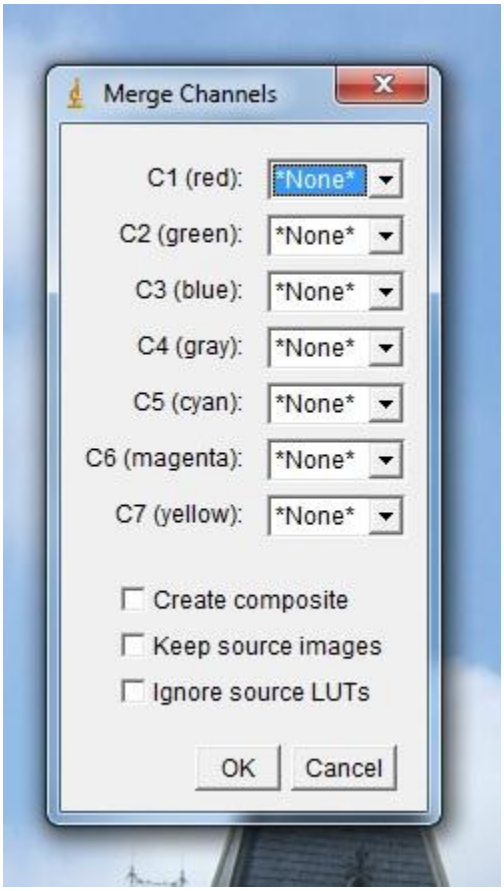


Figure 10 Merging Channels with Image J

As a final consideration, if, after splitting your image into channels, you find that one channel is empty (i.e. the resulting image is pure black) you need not deconvolve this channel. Instead, deconvolve the other channels and, when merging channels simply select the original channels (i.e. black images) for whatever channel was empty.

Limitations of 2D Deconvolution

While 2D deconvolution is a powerful and versatile tool, there are a few distinct limitations. Firstly, in images that are very bright or over-saturated, the brightest points and any glare may appear the wrong

color as the deconvolution process misinterprets these bright parts (Figure 11). For any image where this occurs, we recommend doing a simple greyscale deconvolution which will sharpen without giving any misleading color information. Similarly, if there is too much background noise (i.e. the background appears colored or textured) 2D deconvolution will often render messy results.

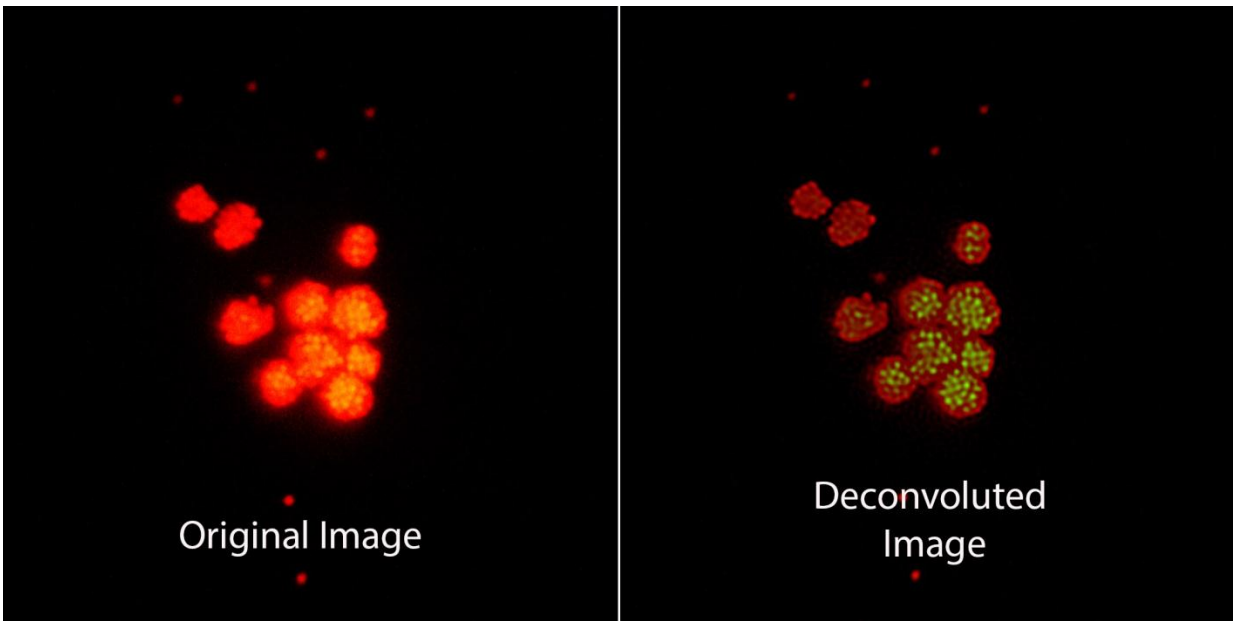


Figure 11

Examples of 2D Deconvolution:

Shown in Figure 12 is an image before and after 2D deconvolution. The left image was taken with our camera at 400x magnification (with a 40x lens). The right image is the same after 2D color deconvolution. Note that in this case we deconvoluted both the red and green channels and did not do so for the blue since it was empty.

In Figure 13, we deconvoluted an image of red fluorescent beads, once again at 400x magnification. In this case only the red color channel had information we wished to deconvolute.

The last example, shown in Figure 14, was taken with the same camera but at 1000x magnification (100x lens) in oil immersion. The image of green fluorescent beads was cropped before deconvolution.

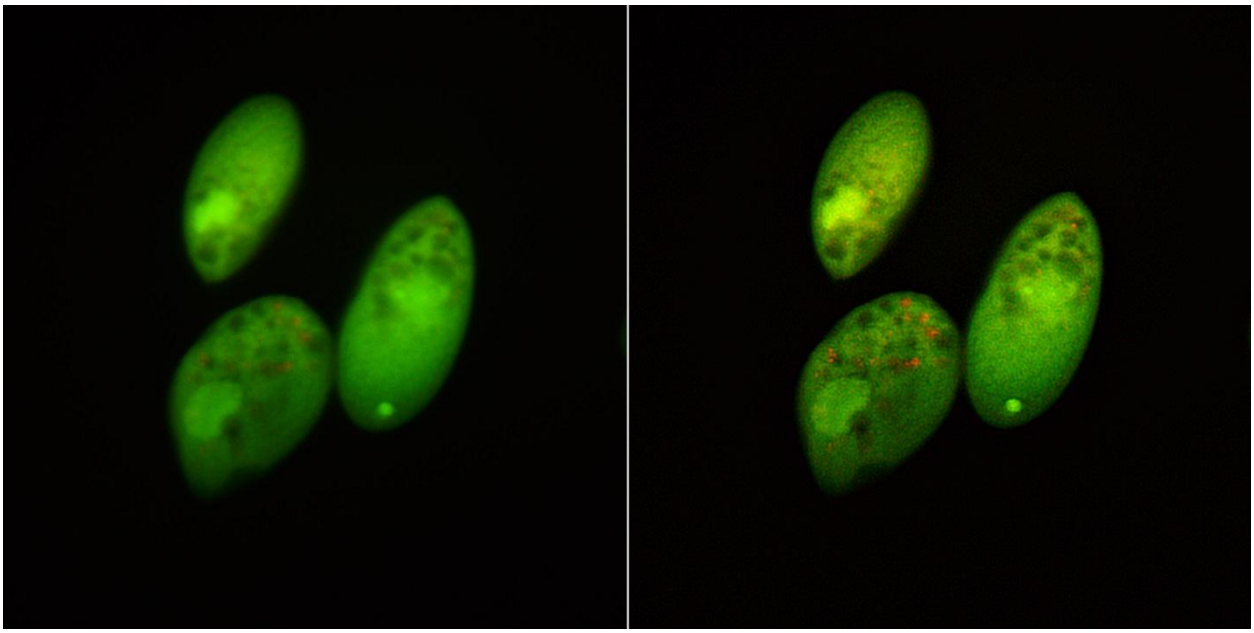


Figure 12 2D Deconvolution at 400x magnification

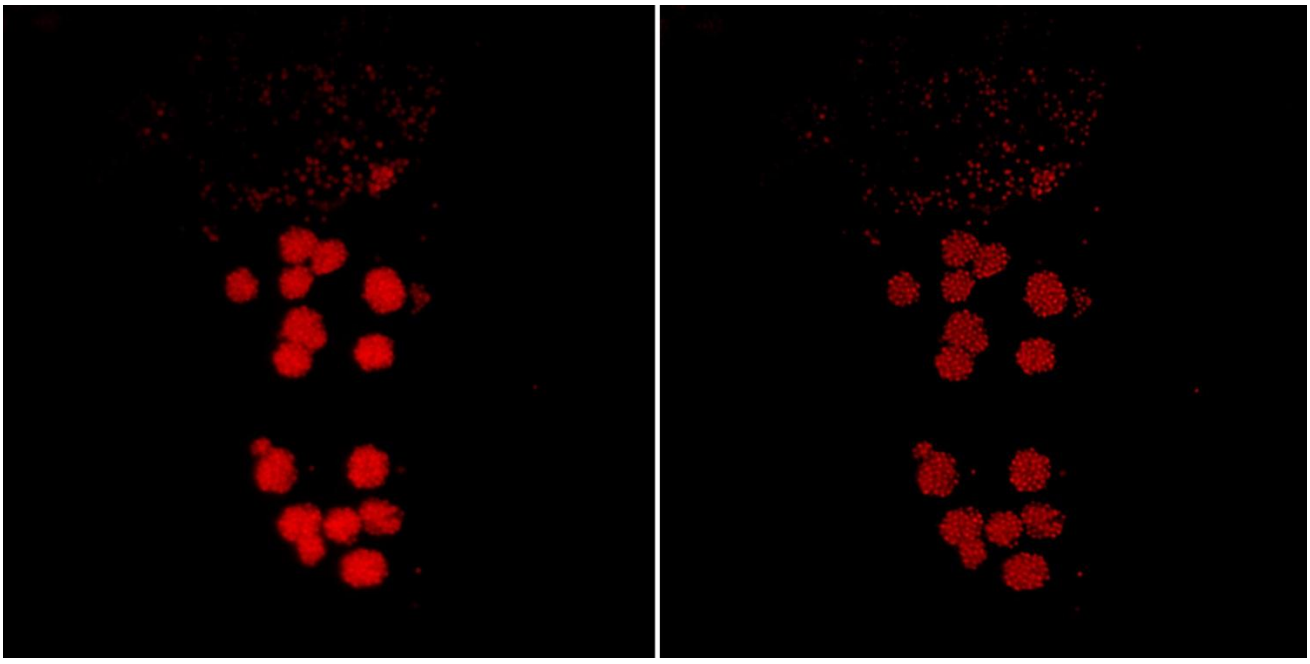


Figure 13 2D Deconvolution of red beads at 400x magnification

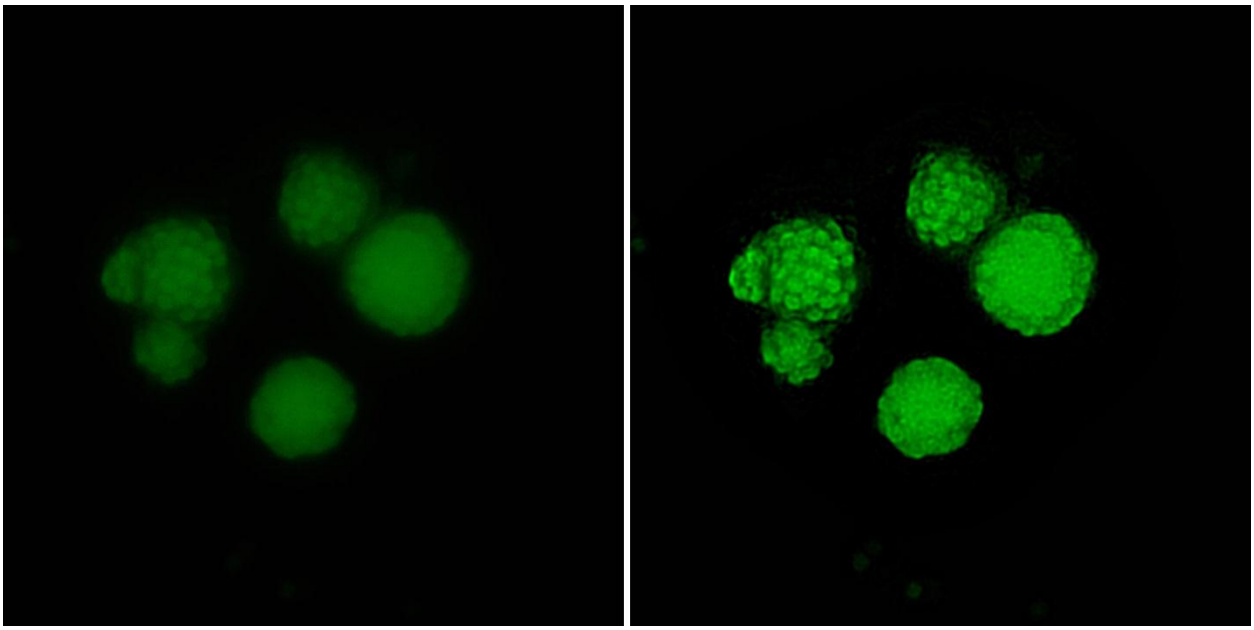


Figure 14 2D Deconvolution of green beads at 1000x magnification

3D Deconvolution

In application, the process of 3D deconvolution is very similar to that of 2D with a few important changes. In 3D deconvolution a stack of images at different depths of focus are effectively cross referenced to eliminate blur from every layer. The algorithm used works in repeated cycles, called iterations, to calculate successively more accurate images. It is possible to over-iterate which will result in artificial patterns of spots or speckles not present in the real subject. This process is more computationally complex than that of 2D deconvolution and can, especially where large images or large numbers of iterations are concerned, take a few hours to complete.

Creating an image stack

For 3D deconvolution you will need a stack of images (that is, a single file containing the photographic slices at different depths). While image acquisition is not specifically a part of this manual, we do suggest that the following formula be used to designate the ideal distance between images (i.e. how far the fine focus is turned between taking each picture). The ideal z-step is defined as follows:

$$\text{z-step} = \frac{\lambda \times RI}{NA^2}$$

Where RI is the refractive index, NA is the numerical aperture of the lens, and λ is the wavelength.

To create a stack first put your photographs in a folder (they will need to be numbered so they start with the image of the least depth and move on to the most depth). Open ImageJ and click the 'File' dropdown menu at top and select 'Import' and then 'Image Sequence'. A navigation window will open where you can find and select the first image (least depth). Once this image is selected, click 'open' and a window will appear labeled 'sequence options'(Figure 15).

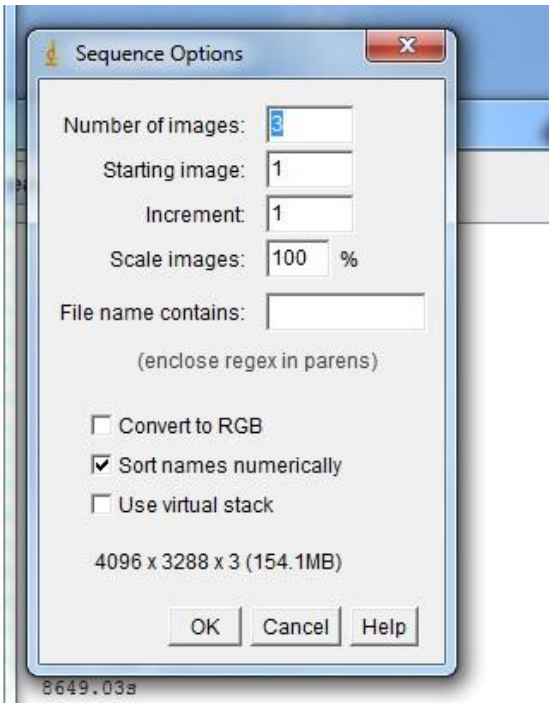


Figure 15 Importing a sequence

From this window you will enter the number of images in your stack and the number of the starting image (this should be 1 unless images that do not belong in the stack are in your folder and appear before the starting image). Leave the other settings at their defaults and click 'ok'. After a minute of processing an image stack should appear containing all of your images. In the lower left hand side of this button is a play button that can be used to cycle through the images in the stack and verify they are in the correct order. If you wish to save this file for later use (so as not to have to reimport it), save it in the format of a '.tiff' or '.tif' file which will preserve the layers. Shown in Figure 16 is a sample image stack in the form of a series.

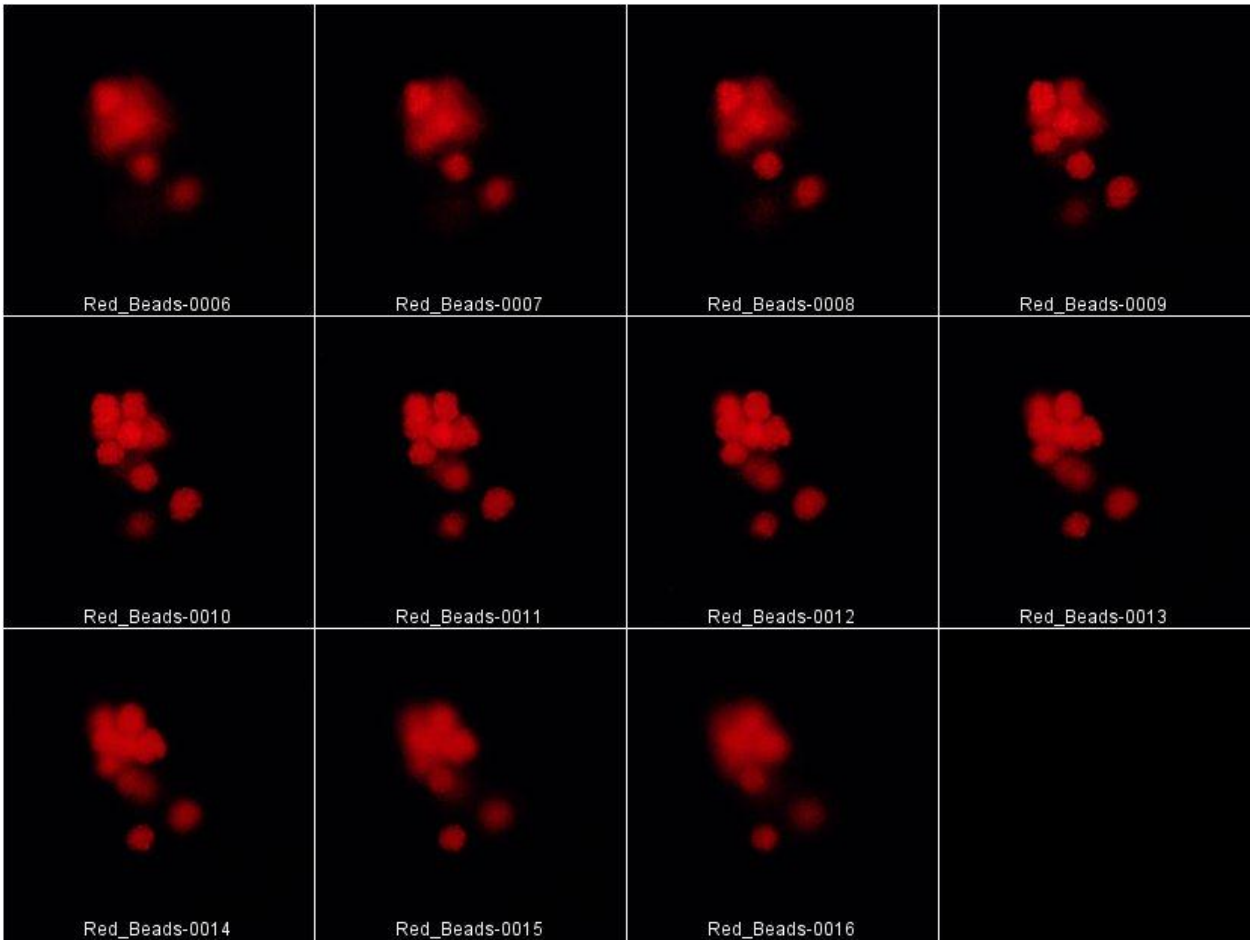


Figure 16 Image Stack shown as series

Optimizing a Z-stack

When your Z-stack is made up of very large images, the process of deconvolution outlined below can use a lot of RAM causing the computer to slow down or crash. Even ideal circumstances will often make for a process that takes more than four hours. Hence we recommend cropping your stack so that only the object you are interested in seeing is viewed. To do this, select the box icon from the ImageJ menu. Then take your cursor over your stack and click and drag to create a box on top of the stack as shown in Figure 17. When you have positioned the box where you want it (i.e. surrounding the object of interest), select 'crop' from the 'Image' dropdown menu (Figure 17). Your entire stack (i.e. every slice within the stack) will now have been cropped as shown in Figure 18. Save your new stack and proceed with deconvolution.

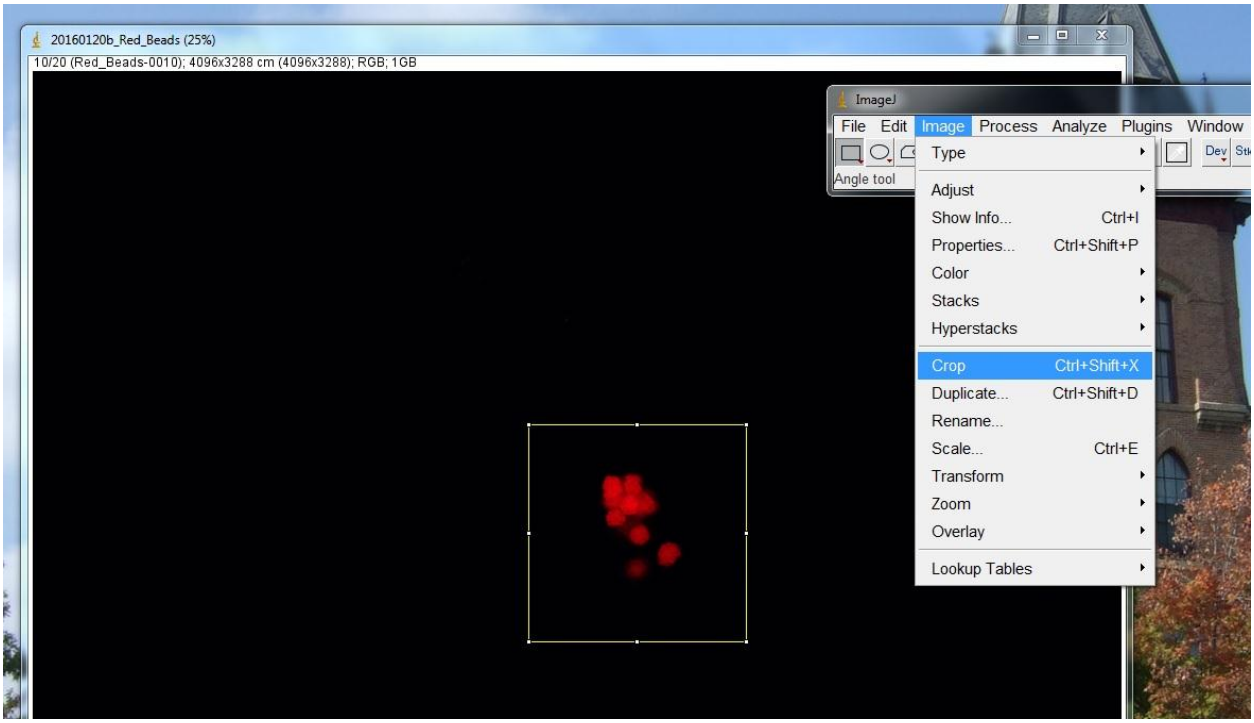


Figure 17 cropping an image stack

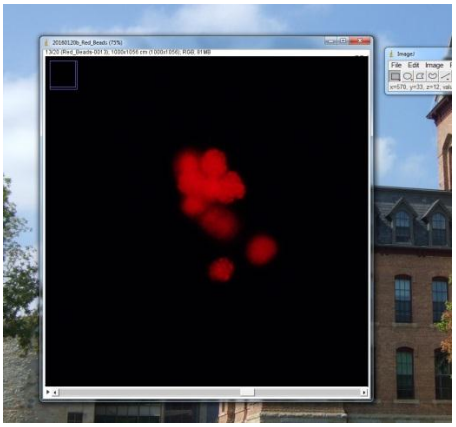


Figure 18 cropped image stack

Generating a PSF

The process is identical to that in 2D deconvolution (see above) with two key differences. For 3D deconvolution the space marked 'Slice Spacing (z), same units' should be have value of the Z-step in your stack, in whatever units you have used for wavelength (likely nm). That is if the space between slices is 1 micron (1000 nm), then you would put '1000' in this box as shown in Figure 19.

Secondly, in the space marked ‘Depth, slices’, you should fill in the number of images in your stack. That is, if you have a stack of eleven images, you would fill in ‘11’ (Figure 19). This is especially important as a PSF of a different dimension (number of images) than a stack will not be accepted in the next step.

Lastly note that if you did crop your stack, the PSF’s dimensions should match those of your cropped stack, not the original stack (i.e., if your stack has been cropped from 4000 x 3900 pixels to 800 x 800 pixels, your PSF should be 800 x 800).

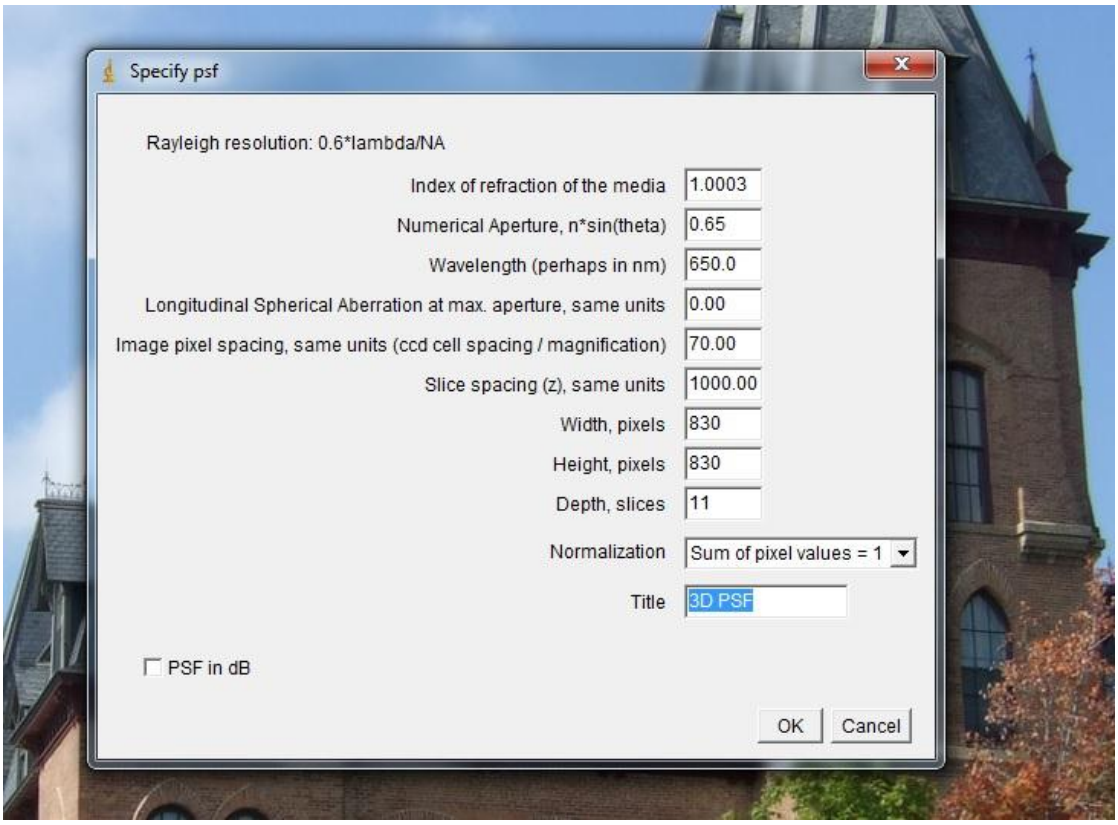


Figure 19 Generating a 3D PSF

The resulting PSF was a stack of slices. Note also that while the resulting image is still black with a white dot, this only appears in the central three slices (an indication of the degree to which the light is diffused). Note in Figure 20 that our resulting PSF has the number in the upper left hand corner and the slider in the lower half indicate that our PSF is a stack, as desired, and not a single image. Be sure and save your PSF (in the form of ‘.tif’ file) for later use.

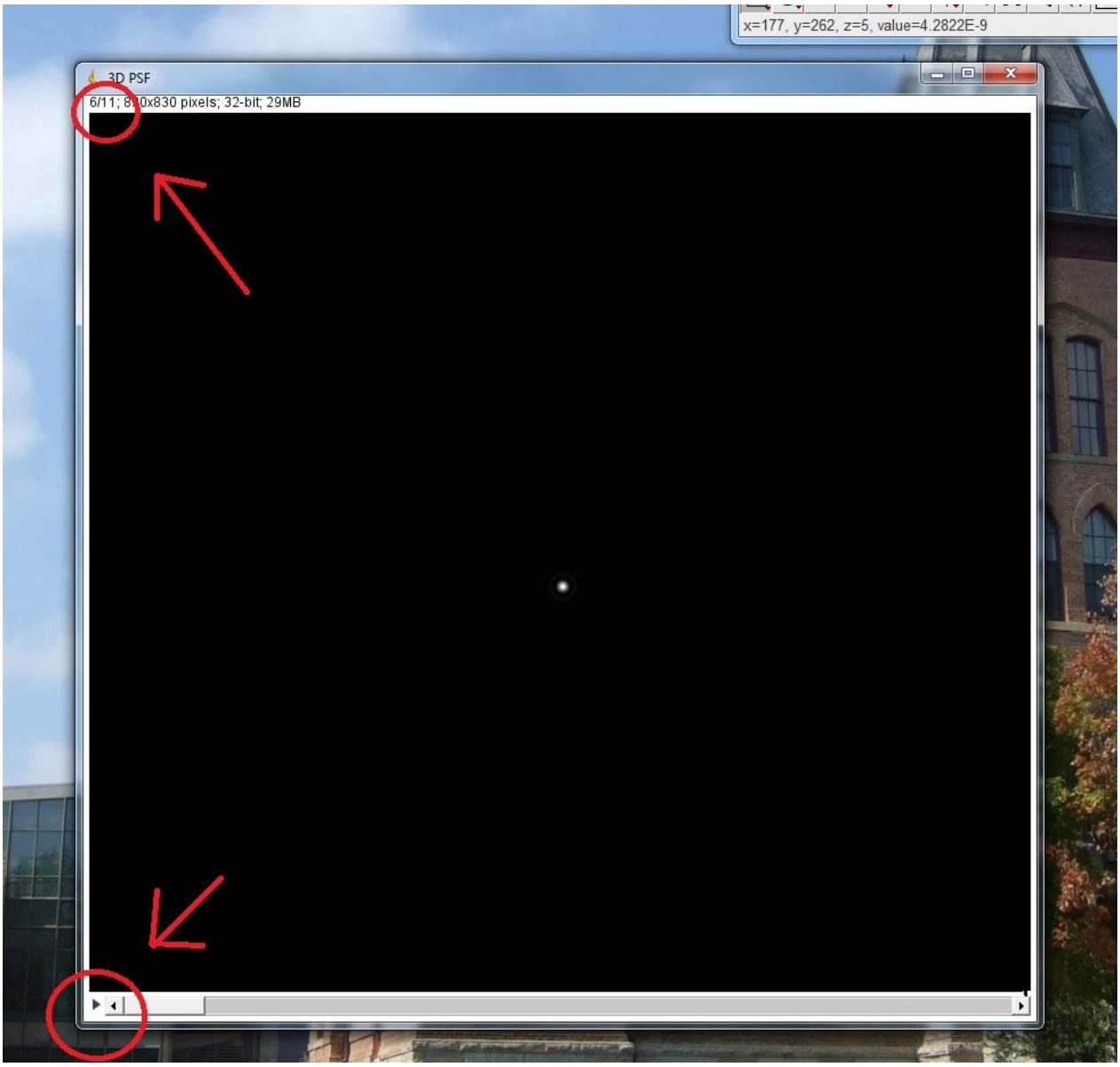


Figure 20 PSF in the form of a stack

Deconvolving the stack in greyscale

If you wish to simply deconvolve in grey scale, use ‘Image>type’ from the dropdown menu and change your image to ‘8-bit’.The process for deconvolving in color will be covered later, so we will assume you have a greyscale image to deconvolute.

Select the plugin that says ‘DeconvolutionLab from your plugins menu. You should see a window that looks like that pictured in Figure 21.

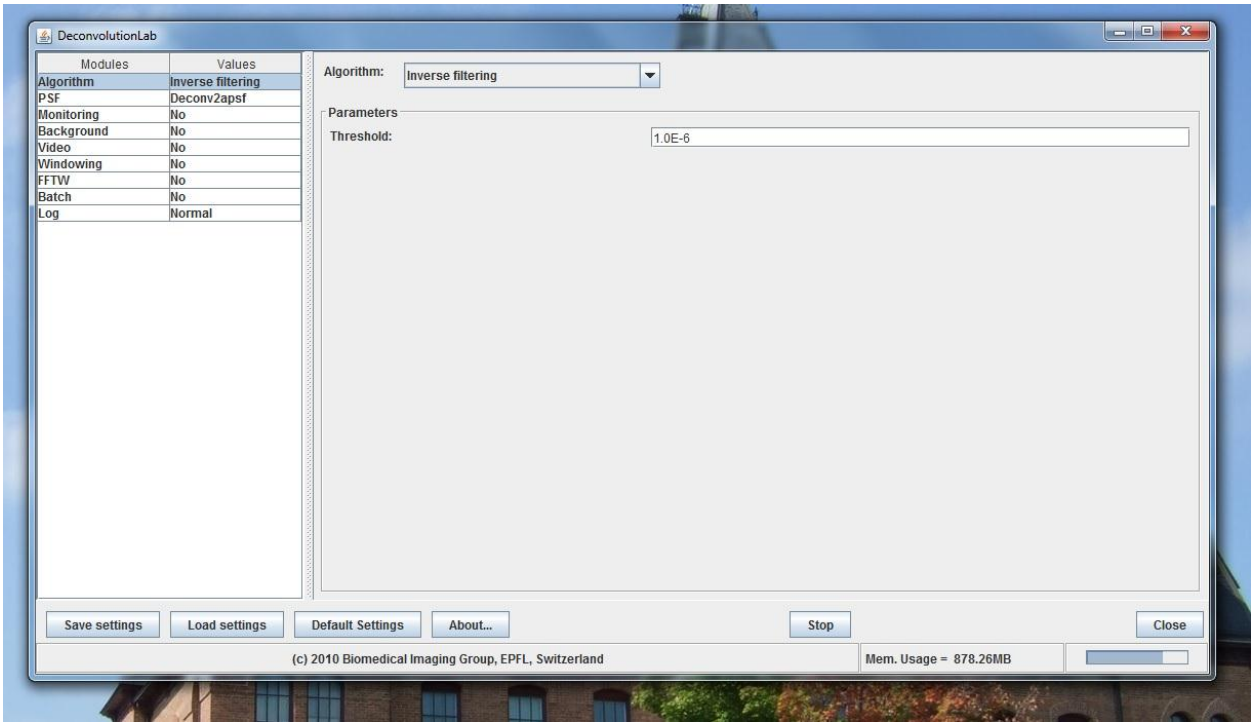


Figure 21 DeconvolutionLab Plugin

Open the stack of images in 8-bit (greyscale) you wish to deconvolute. You may at this point you may crop your stack (see 'Optimizing a Z-stack' above). Now open your PSF of the same dimensions as your stack. In the leftmost column labeled 'Margins' click on 'Algorithm'. Select 'Richardson-Lucy' from the dropdown menu (Figure 21). Make sure the box that says 'Use Input Image' is checked. Specify the number of iterations you wish to use in the box below. For an initial try I suggest 20 as this is not hugely time-consuming allowing you to troubleshoot the process. Later, a value between 40 and 100 should be suitable.

Now, under 'Modules' select 'PSF'. From the dropdown menu, select the PSF image you are going to use. If that image is not an option, but is open in ImageJ, click on 'refresh' and it should appear for the selection. Click on the tab that says 'Log' and from the 'Mode' dropdown menu, select 'normal'. This will allow you to follow the progress of your deconvolution closely.

Before deconvoluting, I recommend clicking the button at the bottom of the window that says 'Save Settings'. This will mean that the defaults will turn Log to 'Normal' and Algorithm to 'Richardson-Lucy' with the given number of iterations, saving you time in the future setting these option.

Now, to deconvolve your image stack in Deconvolutionlab you will first need to click on the Image open in ImageJ on your desktop. By this I mean you should quite literally select it by clicking on its window as if you were going to view it. Then navigate back to Deconvolutionlab and click 'Deconvolve' at the bottom. The log should now pop up with some text in it (if this doesn't happen, simply click on 'Log' under 'Modules').

You should see a line of text reading: 'Starting algorithm "Richardson-Lucy". 000.00s' as shown in Figure 22. This means your deconvolution is initializing. Underneath you should see two more lines (see image below). It is important to verify that the after the words 'Input Image:' you see the name of your stack and after 'PSF:' you see the name of your PSF. If you see the wrong images, it means the wrong images have been selected and you should click 'stop' and start over. Often this problem will occur if one is not careful to make sure that the last image selected before starting was the one you wished to deconvolve.

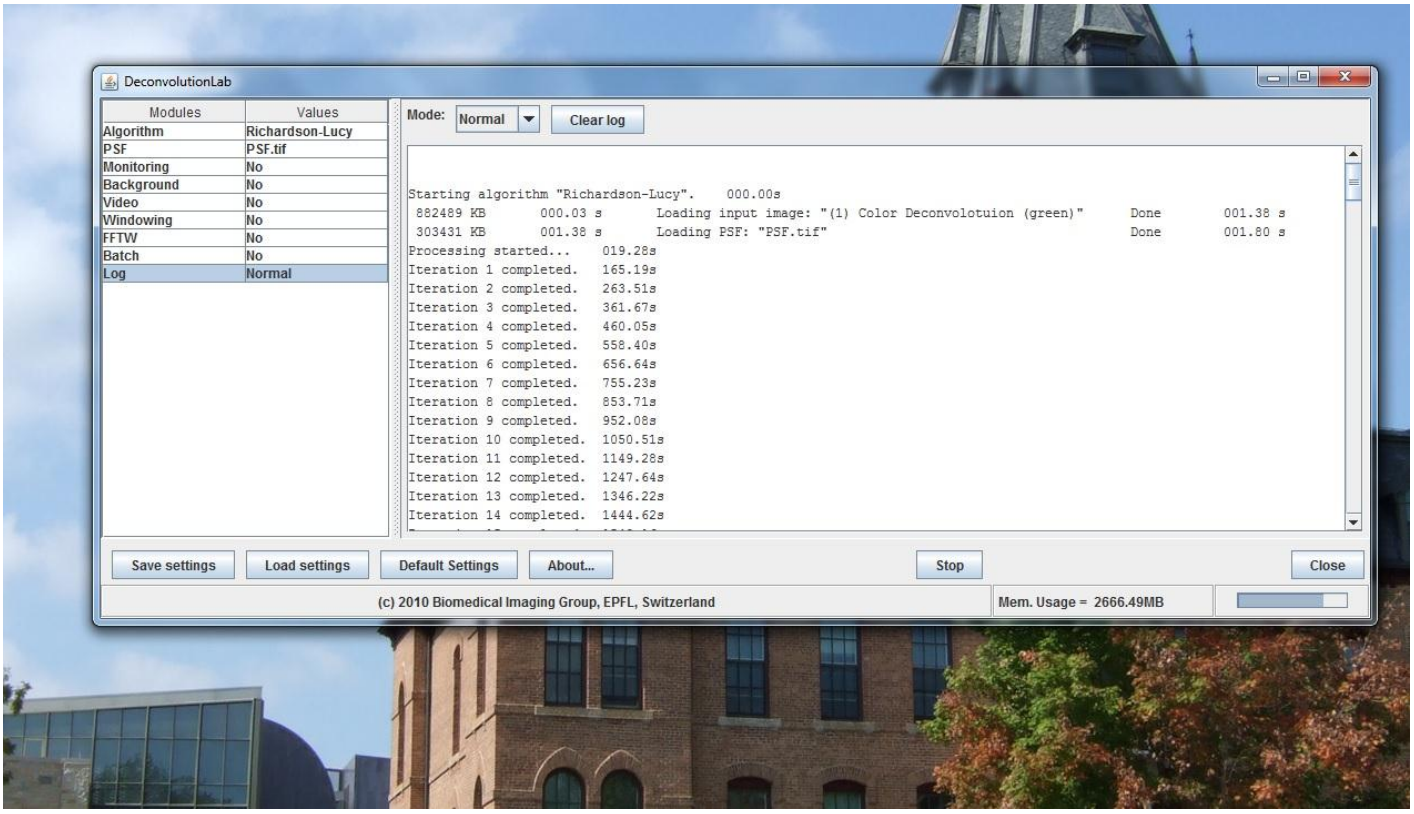


Figure 22 DeconvolutionLab log

Assuming the correct images are selected, after a few minutes text saying 'Iteration 1 completed' along with a tally of how many second have gone by since the deconvolution was started. As more iteration

are finished they will be added to the log until the full number you specified before is complete. When the deconvolution finishes a new deconvoluted stack will open that you can edit or save (we recommend saving right away since after such a long process it is not worth risking losing the final product).

What you should have is a new stack that appears dimmer than what you began with (since the process has removed light) but also sharper. Often the first and last slides which were blurry before will now appear almost black, this is a good sign because it indicates the process was successful in removing light diffraction and blur. Figure 23 shows the deconvolved version of the stack shown in Figure 16 (note that color has been used as outlined in the next section).

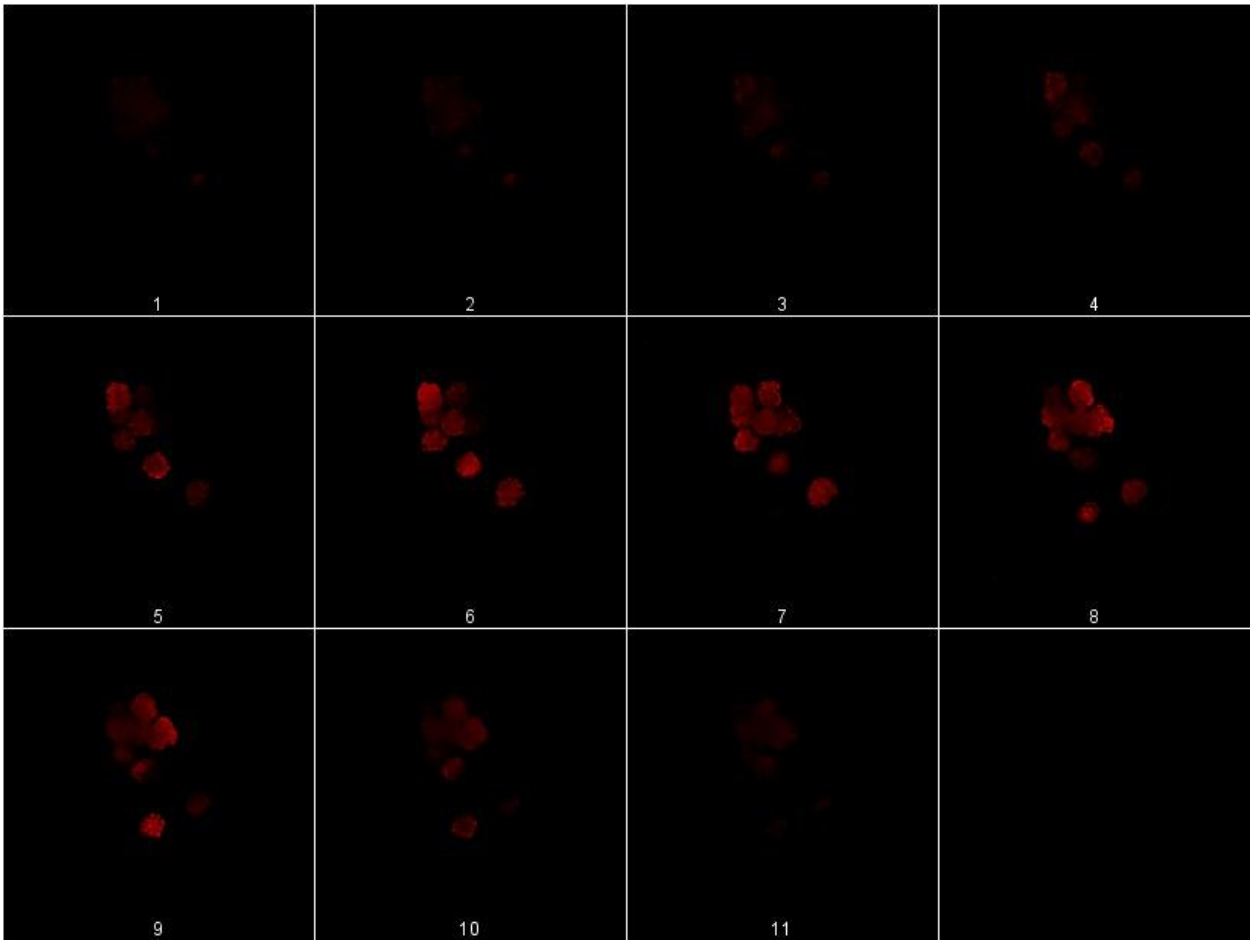


Figure 23 Deconvoluted Image Stack (note that we got color using the process below)

Deconvolving in color

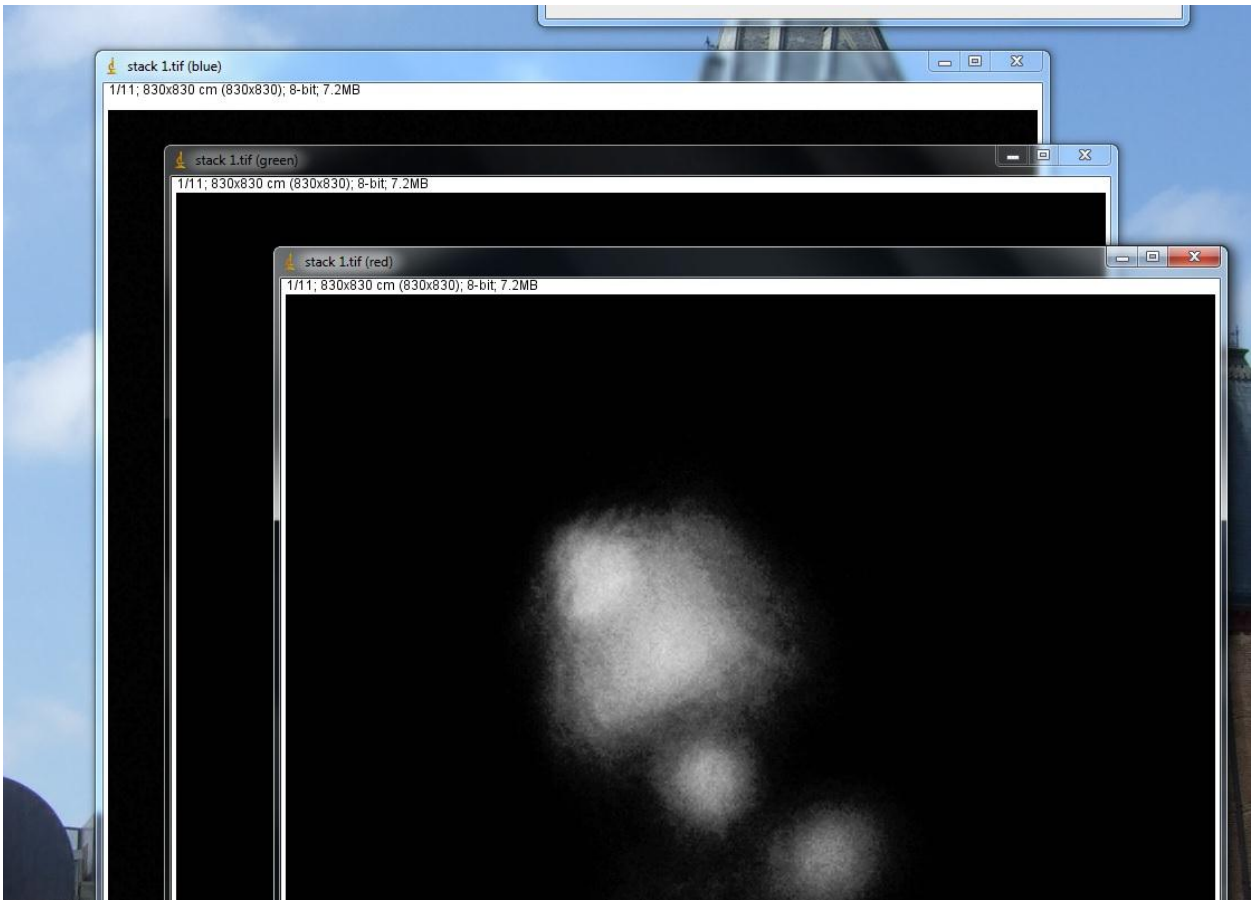


Figure 24 Stack split into three color channels

Much like the 2D process for color deconvolution, a color stack can be split into different channels and deconvoluted separately using the 'image>color>split channels' selection from the dropdown menu (Figure 24). You will now have multiple greyscale stacks corresponding to the different color channels of the image. Once these have been deconvoluted using the process outlined above, click on 'image>color>merge channels. Select the relevant stack for each color channelwhile unchecking the box for 'create composite', to put together a deconvoluted color stack (Figure 25).

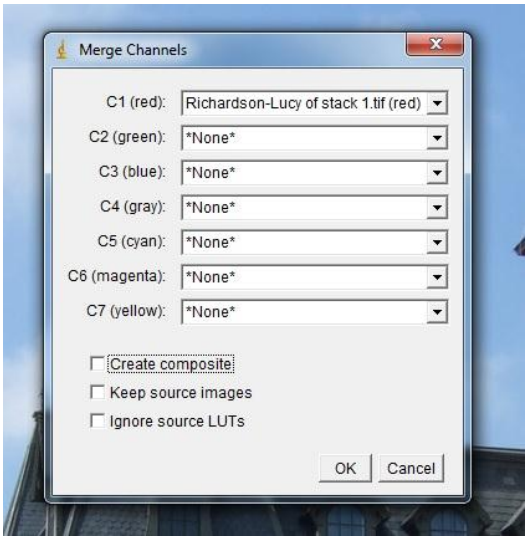


Figure 25 Merging a stack channel

Now, in many cases if the image is primarily one color (for example an image of only green fluorescent beads), you will be better served to deconvolute only the relevant channel as it is faster and less computationally taxing than deconvolving each. To do this, split your stack into channels and deconvolute the relevant channel. Once this single channel stack is deconvoluted, use 'image>color>merge channels' , with your stack selected for the relevant channel and 'none' selected for the others, while being sure to uncheck the box for 'create composite'.

Creating a Z-Projection

When you have a final deconvoluted stack, you will likely want to combine all the images together into a single sharpened image. To do this, open your stack in ImageJ and select 'Stacks' from the 'Image' dropdown menu at top. Then click 'Z-project' which should open the window shown in Figure 26.

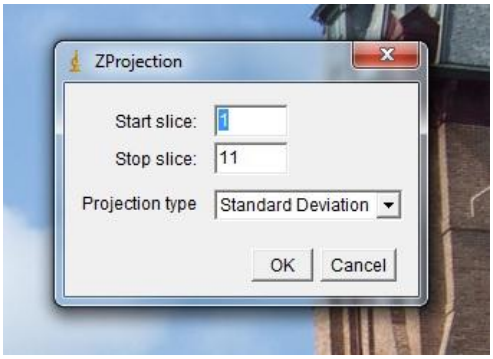


Figure 26 creating a z-projection

The Start slice box should be set to '1', and the 'stop slice' box should be set to the number of images in your stack. From the dropdown menu for 'Projection type', you have a variety of option for how images will be combined. Our best images have come from selecting either 'Standard Deviation' or 'Max Intensity'. When the Projection type has been selected click 'ok' and a new image will appear which can be edited or saved. This image will be your end product, a sharpened image from which the light diffraction has been removed as shown on the right side of Figure 27. For added clarity, adjusting the contrast ('Image'<'Adjust..'<'Brightness/Contrast') can make the image even clearer.

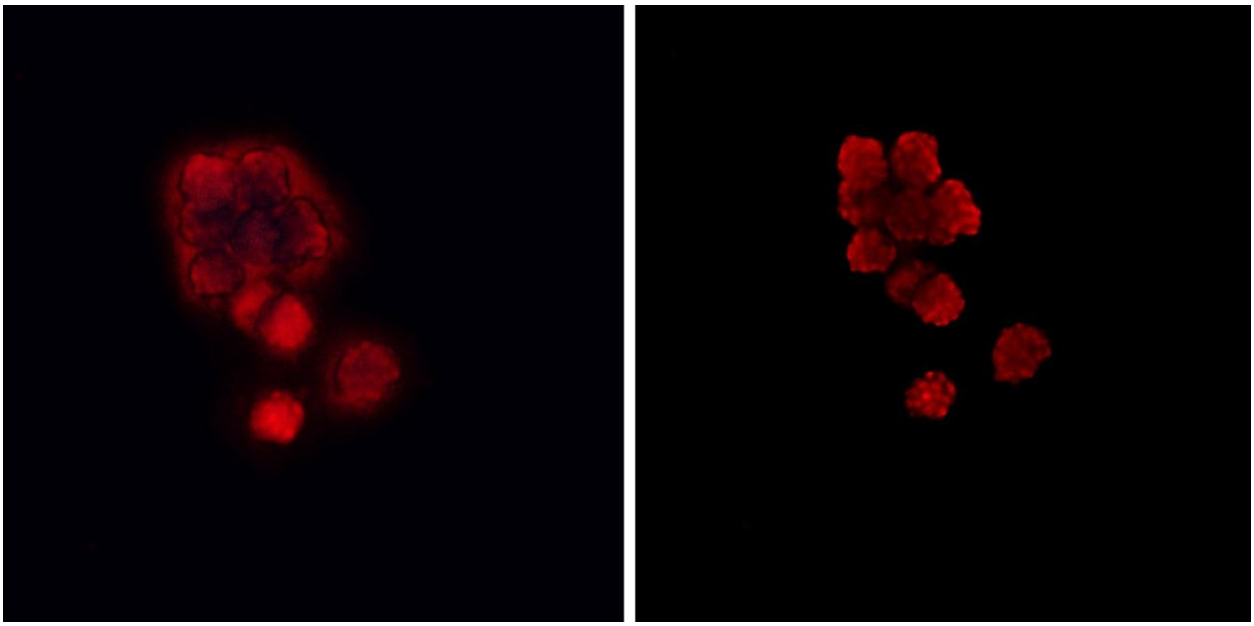


Figure 27 Z-projections of stack before (the left image) and after deconvolution

Limits of 3D Deconvolution

Our goal for this process was to provide an inexpensive means of deconvoluting fluorescent images without needing extremely expensive equipment. The microscope we used was an older model (the Olympus CH40), and our camera (The Omax CS-A35140U3) was relatively inexpensive. However, as a consequence, we discovered certain limitations in the deconvolution process.

Firstly, while we have found the process to work well at up to 400x magnification, we found that 1000x magnification (a 100x lens) did not render sharp results. The reason for this is likely that our microscope is not able to make as small of adjustments between stack slices as desired and our camera's resolution (14.0 megapixels) is not high enough. Theoretically, the processes outline here would work at high

magnification, but in our experience 400x and lower is ideal. Note however that 2D deconvolution works at any magnification without issue.

Secondly, we tried these methods using a 2MP Celestron camera and were unable to get good images. We imagine that there is some minimum resolution required for deconvolution, but without extensive testing it is not clear exactly what that would be. But without a doubt, better resolution will guarantee better results.

Thirdly, getting an accurate theoretical PSF is an extremely important component to deconvolution that will define the quality of your results. In fact, assuming you have a camera of decent resolution and a magnification that is not too great, the PSF is almost always to blame for bad results. Often we found that having the wrong numbers for 'CCD Cell Spacing' or 'Index of Refraction' would prevent us from getting the proper results. There is no foolproof way to check if your PSF is correct, but seeing no white dot or a dot that is very large as shown in Figure 28 are both likely signs that something is wrong.

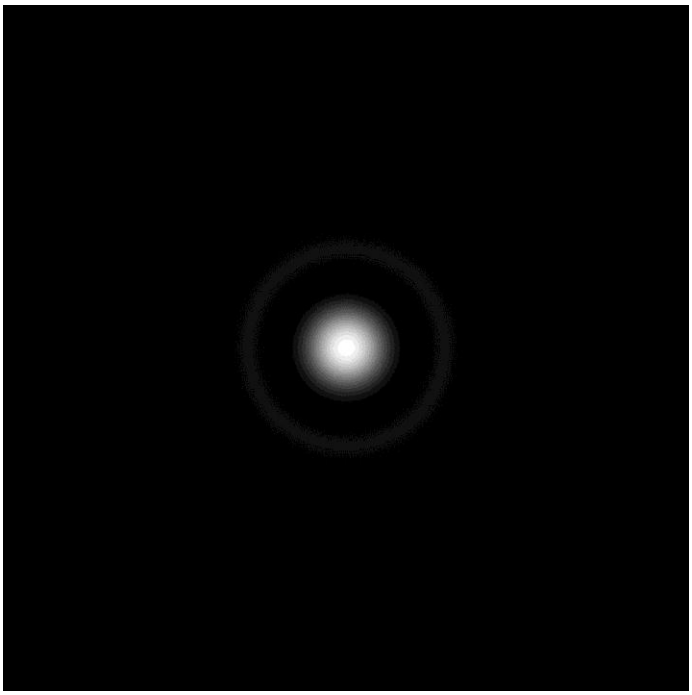


Figure 28 Example of a likely faulty PSF image

Examples of 3D Deconvolution.

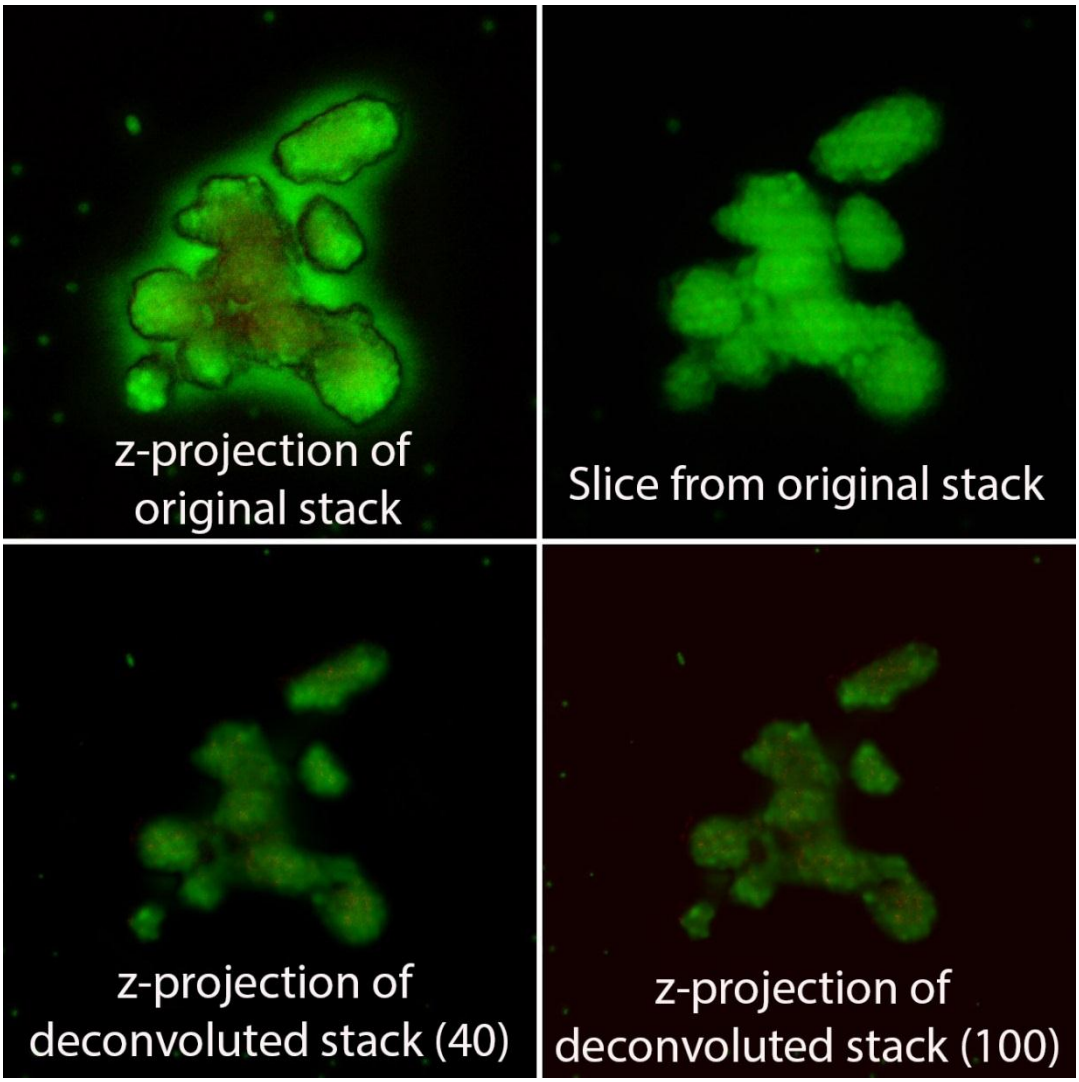


Figure 29 3D deconvolution example

Shown in Figure 29 is an example of color deconvolution with images taken from our Omax camera at 400x magnification (40x lens). The first image is the z-projection of the original stack of image, the second is an example of a slice from our original stack, and the final two images are z-projections of the stack deconvoluted at 40 and 100 iterations

Shown in Figure 30 is an example of another deconvolution, again at 400x magnification with the Omax camera. The first image is a bright field look at the cell and its vacuoles to compare to the fluorescent images. Note also that the final, deconvoluted image was deconvouted with 100 iterations.

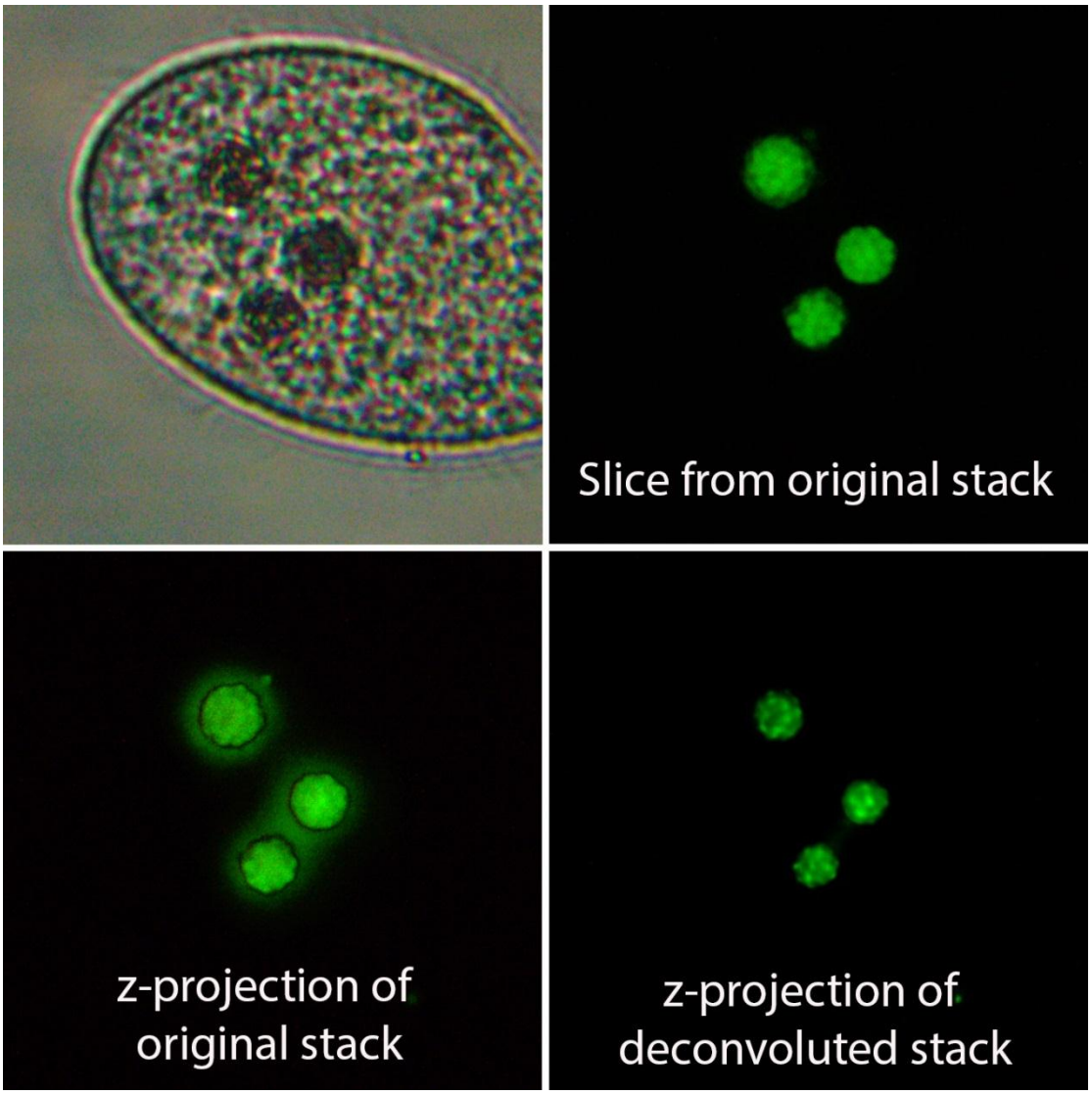


Figure 30 3D Deconvolution example

Appendix A: Example Processes

Included for reference is the exact process we used to get the images in this manual, including all relevant numbers and calculations. While many of these details will of course be different for different labs, this will hopefully provide a foundation for clarification of the process outlined above.

2D Deconvolution Process

We first had a large image with a small cell in its center (4096x 3098 pixels) which we cropped down to center on the cell. The lens used was a 100x lens with a 1.25 numerical aperture (this information was written on the side of the lens itself). We open this image in ImageJ and then open the Diffraction PSF 3D plugin. The picture was taken with oil immersion, which has, according to the bottle itself and index of refraction of 1.5158. The numerical aperture was printed on the side of the lens itself and was 1.25. The wavelength (500 nm) was acquired from the fluorophore. We leave the Spherical Aberration field blank as we could not measure this. The image cell spacing was the size of the pixels for our Omax camera obtained online (1.4 microns) divided by the magnification (100) multiplied by 1000 to convert from microns into nanometers. We doubled our resulting number to make up for a .5x reduction lens attached to our camera to get the result of 28 nanometers. Slice spacing is irrelevant for 2D and was simply left at its default. The width was 1342 pixels and the height was 1350 to match our cropped photograph. The depth of slices was set to 1 since this is a 2D process. We put all these numbers into the plugin and created our PSF.

We then used the 'Image>color>split channels' command to divide our color image into three greyscale images. Observing that the blue channel appeared to be only the points of glare from our original image, we chose to only deconvolute the green and red channels. With these channels open in ImageJ along with the PSF we open the 'Parallel Spectral Deconvolution>2D Spectral Deconvolution' plugin and select an image and the PSF and hit 'Deconvolve'. We perform this process once on each channel to receive two new deconvolved greyscale images which we then merge into a finished deconvolved color image.

3D Deconvolution Process

We began with eleven images (830x830 pixels) taken with our Omax camera at 400x magnification (40x lens). As close as we could determine the z-step between each image is 1.25 microns. We acquired this Z-step value by looking up in the manual for our microscope how far a turn of the fine focus knob moves the microscopes lens.We used the ImageJ 'File>Import>Image Sequence' to create a stack of these eleven images.

We then created a PSF of the proper dimensions. At 400x magnification, no immersion fluid is used so the refraction index was that of air which we acquire online and is 1.0003. The numerical aperture of our 40x lens is .65; once again this was printed on the side of the lens. The wavelength of the emission

fluorophore is 650 nanometers. We continue to leave the Spherical Aberration at 0. The pixel spacing was 1.4 microns, divided by 40 (the magnification), and multiplied by 1000 to convert to nanometers. The resulting number of 35 is doubled once again due to the camera's .5x reductive lens for a final value of 70. The slice spacing was 1.25 microns or 1250 nm, but we estimated just a little smaller for better results, entering 1000. Finally we input the dimensions of the images (830 x830 pixels) and the number of slices to generate our PSF.

The only remaining piece of preparation is splitting our stack into greyscale channels. We select our stack of photographic slices and use the 'Image>Color>Split Channels' command to create three greyscale stacks. Note that for this particular stack we only have red information so we will only deconvolute the red channel, but in many cases this will not be the case. Below is an image of our three new stacks resulting from the color channels being split.

We now open the 'DeconvolutionLab' plugin from the ImageJ plugins menu. Selecting the 'Algorithm' tab on the left we select 'Richardson-lucy' with 100 iterations. On the 'PSF' tab we select our PSF from the dropdown menu. On the 'Log' tab we select 'normal'. .Finally, we selected the stack we wish to deconvolute (the red channel) by clicking it, and clicked 'run' at the bottom of the DeconvolutionLab plugin. The DeconvolutionLab log then counted off the iterations until the deconvolution was complete.

When the program was completed a new stack appeared. This stack was in greyscale so we used the 'Image>Color>Merge Channels' command to convert it back to red. Note that we unchecked the 'Create Composite' box which would have created an image stack in which the channels were not truly merged as desired. The resulting color stack is much sharper and also much dimmer. In fact the first and last few slices which were the most blurred are now entirely black (a sign our process worked correctly).

As a final step we used the 'Image>Stacks>Z-projection…' command to create a Z-projection showing the multiple layers at once. We used the 'Standard Deviation' setting as it gave good results though 'Max Intensity' would also have worked.

References

Arijit Dutta, Aurindam Dhar, Kaustav Nandy, Project report on "Image Deconvolution By Richardson Lucy Algorithm", Indian Statistical Institute, November, 2010.

Hansen, Per Christian, James G. Nagy, and Dianne P. O'leary. *Deblurring images: matrices, spectra, and filtering*. Vol. 3. Siam, 2006.

Lucy, Leon B. "An iterative technique for the rectification of observed distributions." *The astronomical journal* 79 (1974): 745.

Wallace, Wes, Lutz H. Schaefer, and Jason R. Swedlow. "A workingperson's guide to deconvolution in light microscopy." *Biotechniques* 31.5 (2001): 1076-1097.

William Hadley Richardson, "Bayesian-Based Iterative Method of Image Restoration*," J. Opt. Soc. Am.62, 55-59 (1972)