

User guide – Deconvolving dOPM data using the ImageJ Multi-View Reconstruction (MVR) plugin

Hugh Sparks and Chris Dunsby

v1.00

13/06/2023

Contents

Introduction	2
Step 0 – Complete steps 1-9 in the Reslice_dOPM_data_ImageJ_MVR_plugin	2
Step 1 – Inspect demo sample and bead datasets	2
Step 2 – Extract experimentally measured point spread functions (PSFs)	5
Step 3 – Assign experimentally measured point spread functions to another dataset	7
Step – 4 Run multi-view deconvolution with default settings.....	10

Introduction

This guide covers iterative multi-view deconvolution of dual-view oblique plane microscope (dOPM) data as of 2023. Before following this guide, custom scripts need to be run to prepare the dOPM data into an MVR dataset, which includes determining the affine transformations on the raw data in order to deskew and coregister the dual-view raw dOPM data. To prepare the MVR dOPM dataset follow the 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx' guide up to step 9.

The paper describing the dOPM setup can be found here: <https://doi.org/10.1364/BOE.409781>.

The deconvolution steps in this guide are entirely based on the Multi-View Reconstruction (MVR) plugin <https://imagej.net/plugins/multiview-reconstruction>. This guide is therefore not intended to describe how to use the MVR plugin, as this is described in its own documentation.

The MVR plugin uses the deconvolution as outlined in the MVR wiki:

<https://imagej.net/plugins/multi-view-deconvolution>

The deconvolution algorithms used by the MVR plugin are described in this paper:

<https://doi.org/10.1038/nmeth.2929>.

Unlike the guide on reslicing dOPM data, 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx', where MVR batch processing commands are called from ImageJ Python scripts that use GUI dialog boxes to prompt user interaction (<https://imagej.net/plugins/multiview-reconstruction>), this guide involves only using MVR plugins tools directly.

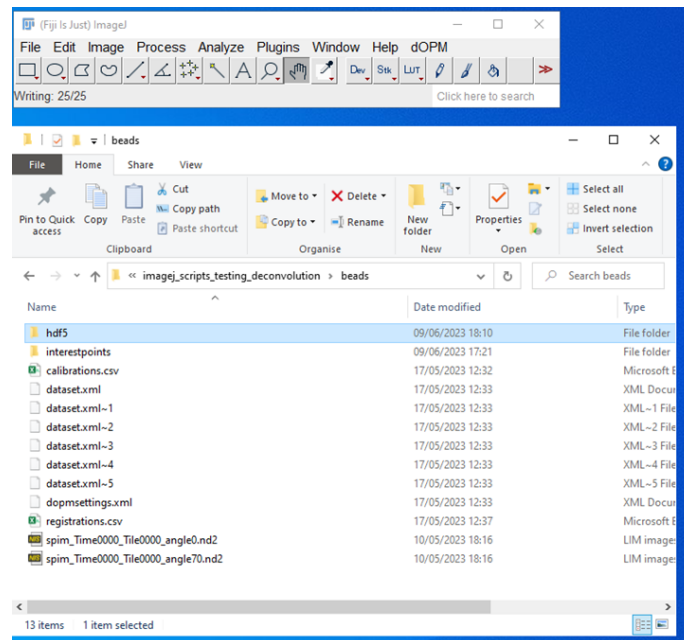
Note: all of the steps below can be carried out purely using MVR batch processing commands, which allows you to do everything with scripts and avoid using the MVR applications GUIs and dialog boxes – for example, to carrying out deconvolution on large datasets. This guide is for becoming familiar with MVT deconvolution routines.

Step 0 – Complete steps 1-9 in the Reslice_dOPM_data_ImageJ_MVR_plugin

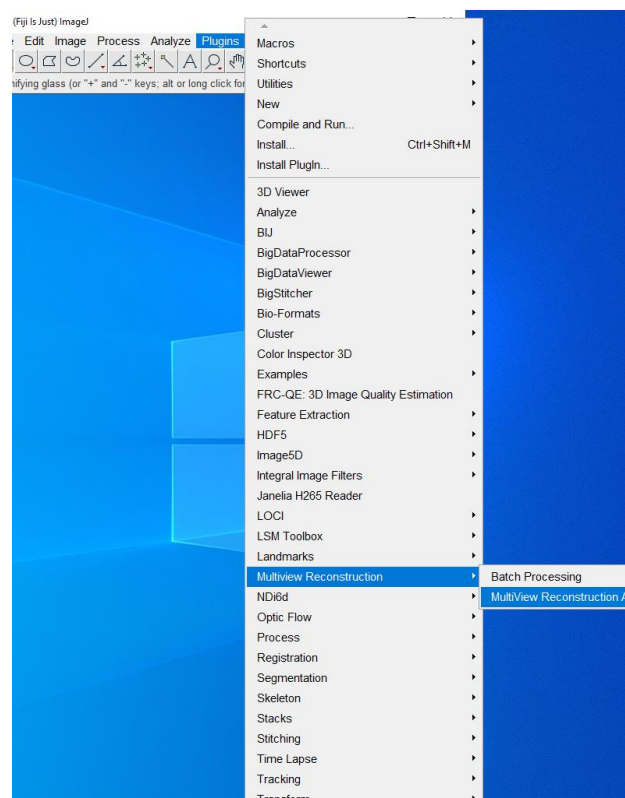
Complete steps 0-9 in the 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx'. This guide relies on the same demo data and ImageJ version used in this other guide and the tasks here assume you have downloaded these files and prepared the data up to step 9. You are going to make use of the same MVR datasets from this previous guide.

Step 1 – Inspect demo sample and bead datasets

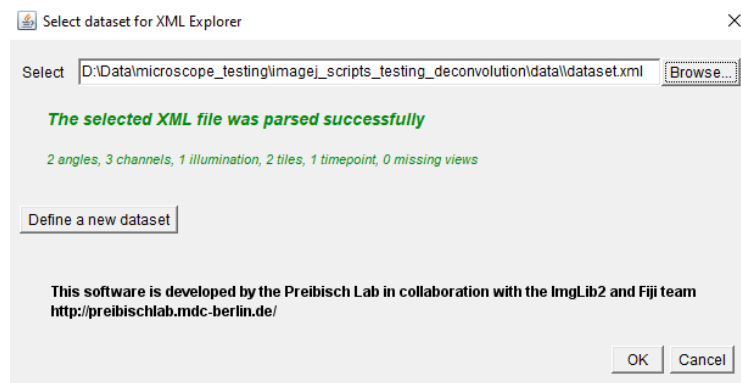
- In the Reslice_dOPM_data_ImageJ_MVR_plugin guide, you will have practiced making an MVR dataset. Overall, you would have made an MVR data set of fluorescent beads in 3D suspended in agarose and another dataset of a biological sample - an organoid with three fluorescent labels. Both bead and organoid data were acquired with the same scan settings and spectral channels – see 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx' for more discussion on this.
- Each MVR dataset generated consists of the raw data and .xml files. Additional .csv files are produced by the dOPM ImageJ Python scripts and are specific to the dOPM data.
- The screenshot below shows the folder structure for the bead data. The folder structure automatically includes a subfolder named 'hdf5' which contains a version where the raw data has been converted to the hierarchical '.hdf5' data format for fast 3D viewing with the 'Bigdataviewer' tool that is available within the MVR ImageJ plugin.



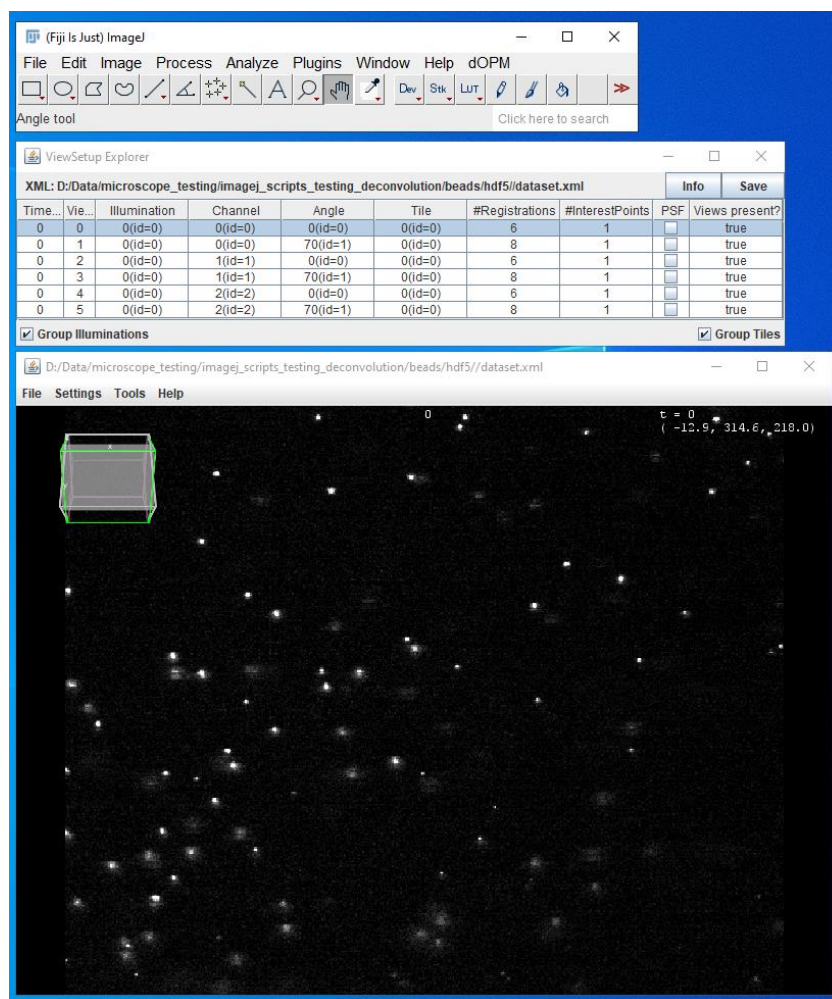
- Open up the MVR bead dataset that you made by following the Reslice_dOPM_data_ImageJ_MVR_plugin guide up to step 4 and check that it is co-registered. Just as with the fusion work, the multi view deconvolution requires successful co-registration of the bead data to work as intended.



- Make sure you choose the 'hdf5' version of the bead data – i.e., not the raw bead dataset shown in the screenshot below – as this ensures fast 3D viewing.



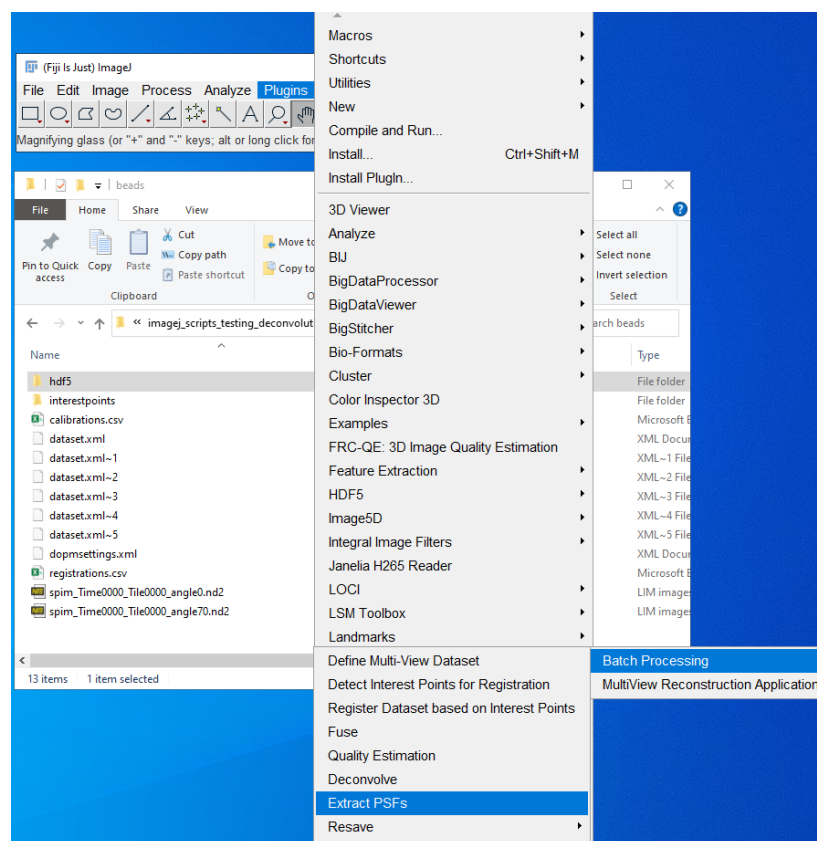
- You should see the following GUIs showing the dataset organisation 'viewsetup explorer' and the 'Bigdataviewer' tool view of the co-registered bead dataset. Check that the two views are co-registered, i.e. that the beads from dOPM view 1&2 visibly overlap.



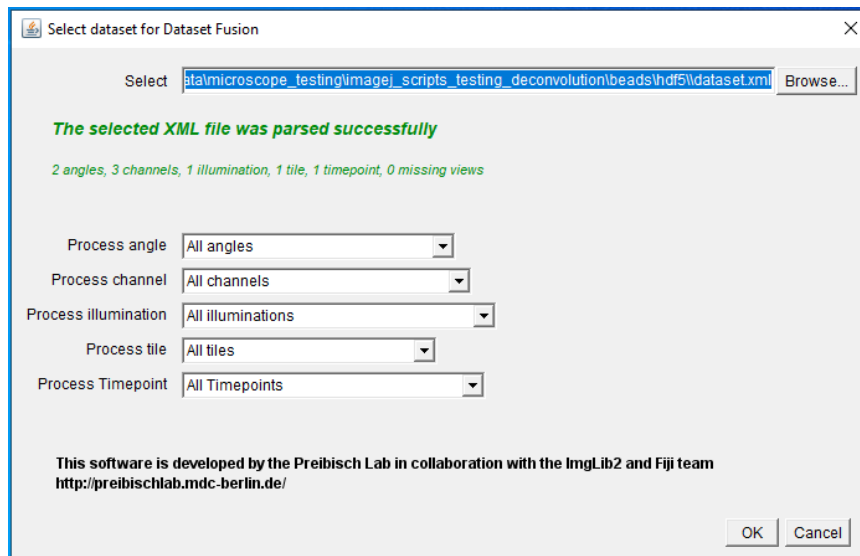
- After confirming that the bead dataset is setup and that dOPM views 1&2 are co-registered, close down the bead dataset by closing the dataset viewer and the 'Bigdataviewer' windows of the MVR plugin in ImageJ.

Step 2 – Extract experimentally measured point spread functions (PSFs)

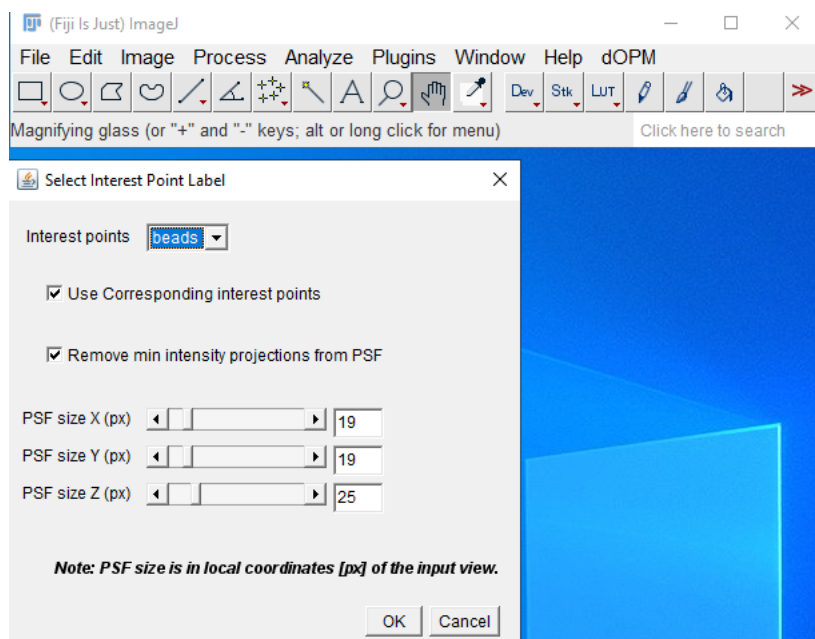
- Step 1 must be completed before proceeding to this step. In this step, the MVR plugin is used to extract representative point spread functions from the bead dataset for all spectral channels and both dOPM views.
- The fluorescent bead dataset is used for this, as the bead size is below the diffraction-limited performance of the microscope. The MVR plugin has routines that automatically extract experimental PSFs based on the assumption that small bright dots in the datasets are formed by sub-diffraction scale sources of fluorescence.
- From the ImageJ process tab select the option MVR path > batch processing > Extract PSFs. N.B. in the Reslice_dOPM_data_ImageJ_MVR_plugin these batch processing commands were initiated by the dOPM Python scripts; here they are executed directly.
- Note for users once they are familiar with deconvolving dOPM data: By using ImageJ's Macro recording tool, these commands can be recorded and used to develop your own scripts similar to the ImageJ Python Reslice_dOPM_data_ImageJ_MVR_plugin script. See <https://imagej.nih.gov/ij/developer/macro/macros.html> for information about ImageJ macros.



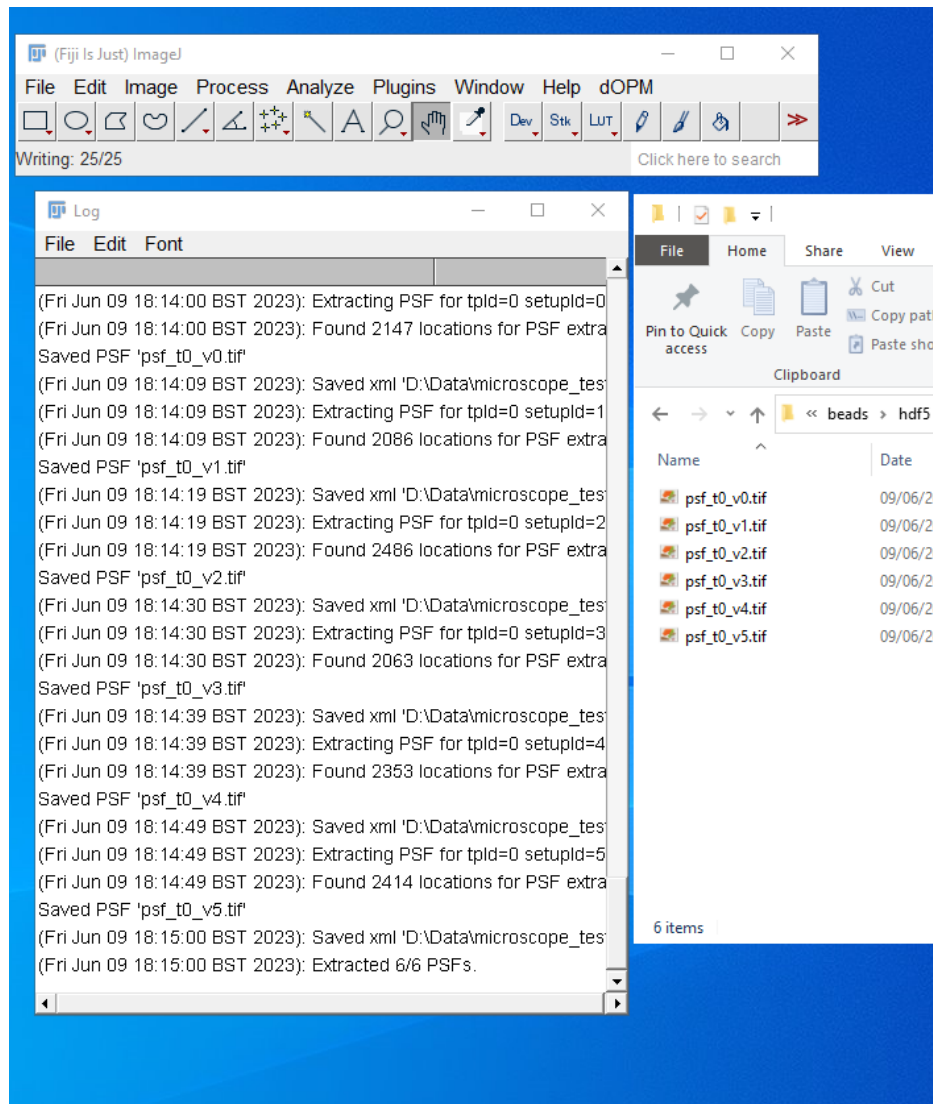
- Once 'Extract PSFs' has been selected, then the GUI 'Select dataset for Dataset Fusion' pops up.
- Choose the same bead dataset path from Step 1 above.
- Ensure that 'All xxxxxx' is selected for all the available dimensions, i.e. angle, channels etc – this GUI is asking which dimensions you want to extract experimental point spread functions from, and it is necessary to do this for all of them.



- Press OK and the next dialog box appears allowing control over the dimensions of the volume use to store the measured PSF. Leave the values unchanged.
- See the links in the introduction for a deeper understanding of how the authors of the MVR plugin intended this process to be used:
<https://imagej.net/plugins/multiview-reconstruction>
<https://doi.org/10.1038/nmeth.2929>.
<https://imagej.net/plugins/multi-view-deconvolution>



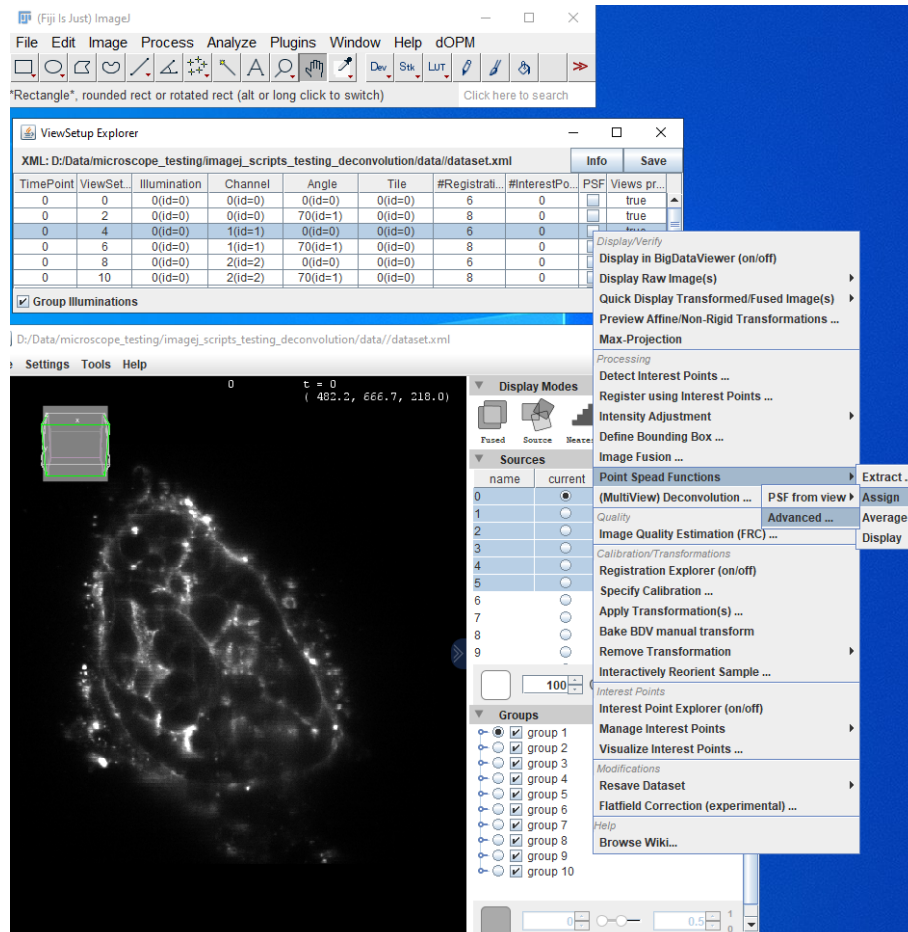
- Run the routine and the log window shows progress. The plugin is extracting PSF estimates from the data for each channel and each view. In this demo dataset, there are two views and 3 channels, so 6 PSFs are extracted in total.
- The batch command also automatically saves the measured PSF volumes as tiff stacks with logical names into a subdirectory of the bead dataset folder – see the partial view of the file explorer window on the righthand side of the screenshot below.



Step 3 – Assign experimentally measured point spread functions to another dataset

- Now use the MVR application GUI interface to open the dataset that you want to carry out multiview deconvolution on.
- This could be the bead dataset, which is a good way to check deconvolution performance. In the case that the dataset to be deconvolved is the same as the dataset from which you extracted experimentally measured PSFs, you can skip the following parts in this section and go straight to Step 5, because MVR already has experimentally determined PSFs for each view & channel pair for this data.
- Whenever you want to carry out multiview deconvolution of data using MVR for dOPM data, you will need to have data acquired using sub-diffraction limited sources of fluorescence. Typically, this will not be the same data as the biological sample so you will need to import experimentally measured PSFs. The following steps describe how to manually assign point spread functions to each dOPM view and channel pair. For the two-view, three spectral channel data provided, six manually assigned PSF are needed, and this comes from the bead data acquired in the same way. This manual assignment does not scale well for hundreds of organoids and timepoints so this process will be automated in scripts in the future.

- This guide assumes you want to carry out multiview deconvolution of the example biological sample data provided: open the sample dataset from the demo that was already set up in the Reslice_dOPM_data_ImageJ_MVR_plugin guide.
- As shown in the screen shot below, right click on the dOPM 'view' + 'channel' that you want to assign a point spread function to and then 'point spread functions > assign > advanced'.



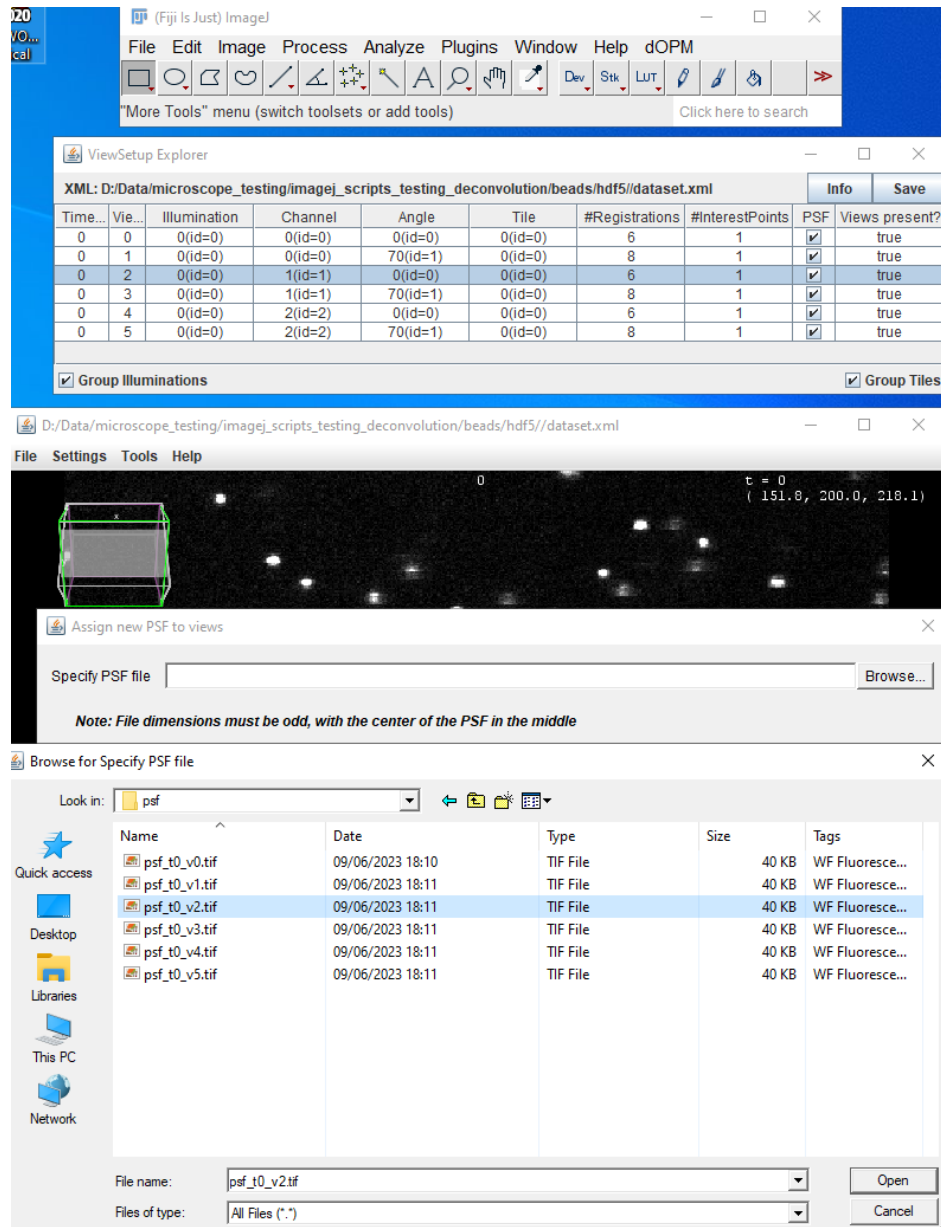
- Another dialog box appears with multiple options in a drop-down list – choose 'assign PSF to all selected views'. Note: because you only chose one dOPM 'view' + 'channel' you are only assigning one PSF to one particular 'view' + 'channel' combination i.e. 'view 1 + channel 0' and you will need to do this six times in total in this case because the sample data has six 'view' + 'channel' combinations.



- A new dialog box prompts you to select the PSF file. Choose the matching PSF from the bead dataset folder you worked on in Steps 0-2 of this guide.
- Note the psf tiff stacks shown in the screenshot below for the bead dataset have automatically been saved with a logical naming scheme that starts with $t\{i\}_v\{ii\}$ and the numbers increment

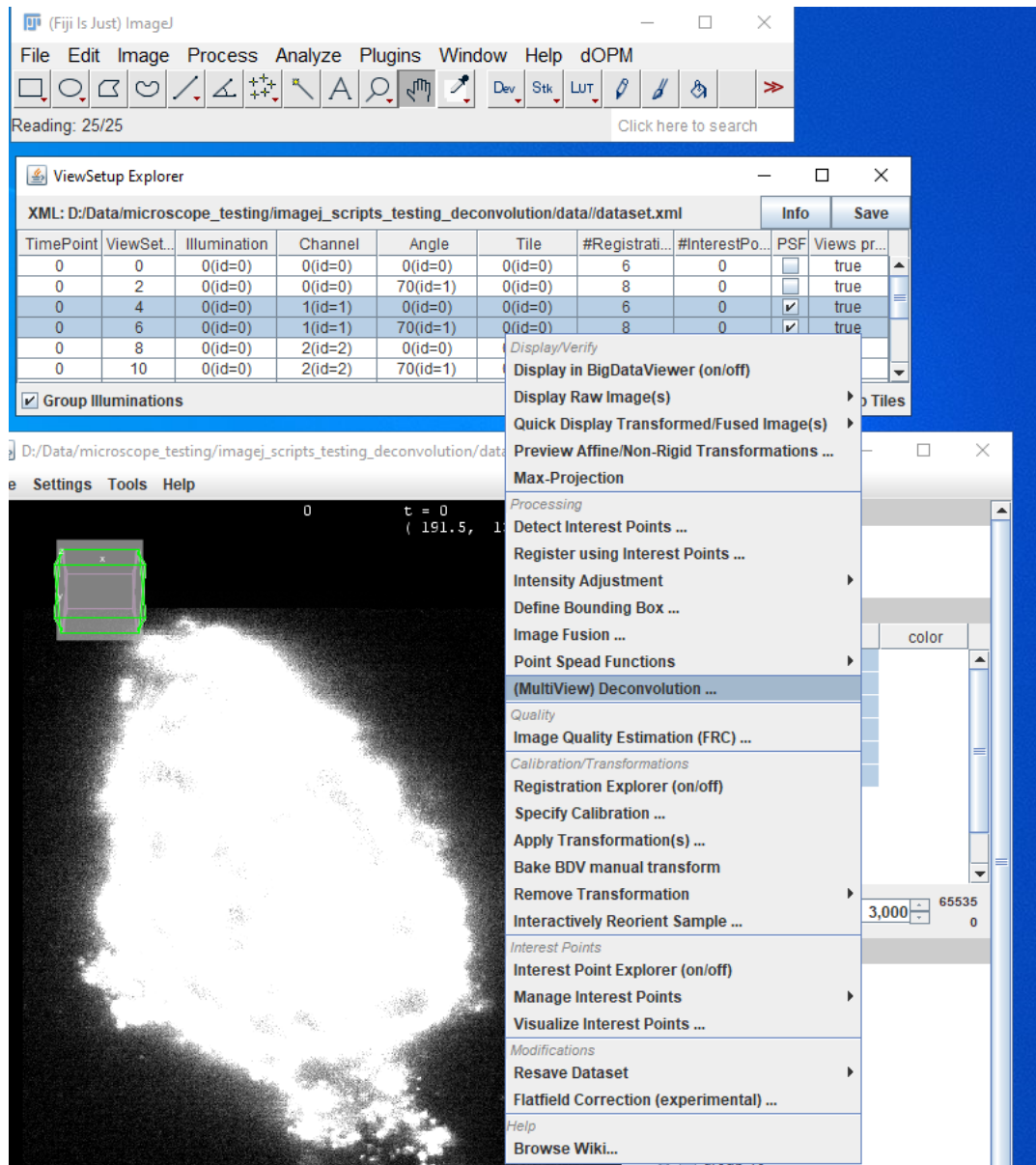
according to the IDs per raw volume (a single view and channel in dOPM case) assigned to the dimensions from left to right in the 'viewsetup explorer' as shown in the screenshot below.

- For instance, there is one time point, 3 channels, 2 angles and 1 tile for the bead data. This means the psf file names:
 - psf_t0_v0 maps to time 0, view 0, illumination 0, channel 0, angle 0 and tile 0
 - psf_t0_v1 maps to time 0, view 0, illumination 0, channel 0, angle 1 and tile 0
 - psf_t0_v2 maps to time 0, view 0, illumination 0, channel 1, angle 0 and tile 0
 - psf_t0_v3 maps to time 0, view 0, illumination 0, channel 1, angle 1 and tile 0
 - psf_t0_v4 maps to time 0, view 0, illumination 0, channel 2, angle 0 and tile 0
 - psf_t0_v5 maps to time 0, view 0, illumination 0, channel 2, angle 1 and tile 0

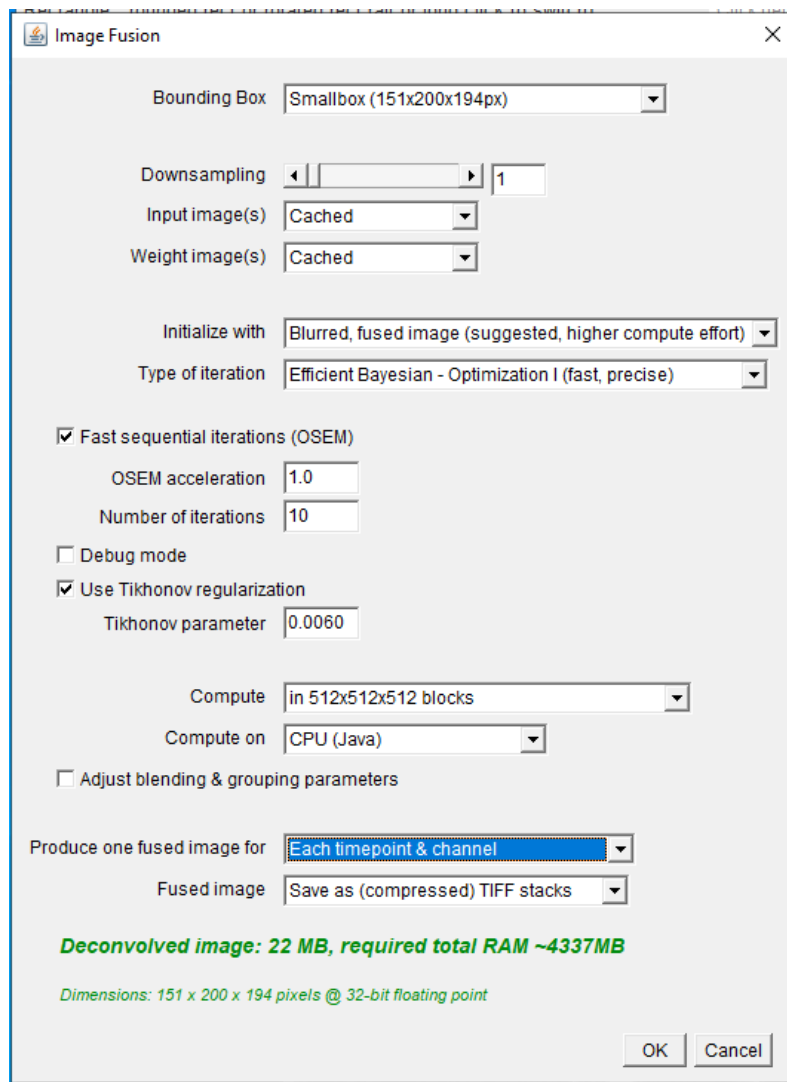


Step – 4 Run multi-view deconvolution with default settings

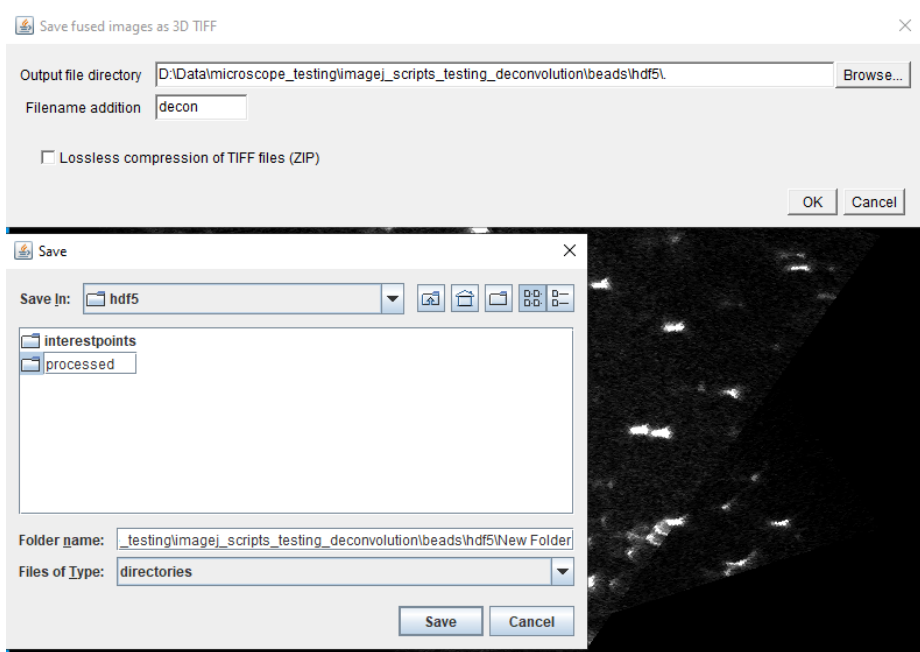
- Now that each PSF is assigned to each view + channel in the sample dataset you can do multiview deconvolution.
- Right click on the explorer and click on '(MultiView) Deconvolution...'



- Leave all default settings as indicated in the screenshot below.
- Note: if you do this on a cropped volume the processing time can be reduced by orders of magnitude, which is important as multiview deconvolution is computationally expensive and typically takes minutes to hours depending on volume size. For this guide we recommend using a bounding box of around 200x200x200 voxels for familiarisation with the process.
- See the Reslice_dOPM_data_ImageJ_MVR_plugin guide for instructions on using bounding boxes to crop the extract processed data to smaller volume to save computing time and memory requirements (RAM and disk storage space).
- Finally ensure 'save as compressed tiff stacks is selected' and then click OK.



- Make a new subdirectory within the dataset folder for deconvolved data and consider having a filename addition (prefix) as shown in screenshot below.



- Click save and monitor progress of deconvolution from the Log window.



- Once deconvolution is complete go to the defined save path in the previous step and you should find deconvolved z-stacks of the data. These can be explored in ImageJ or any other tool for 3D visualisation. In the example below, a small bounding box was used to define a cropped volume to speed up the deconvolution for this demonstration.

