Introduction

The guide covers the use of 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 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 therefor not needed to understand how to use this MVR plugin as its own documentation can be used.

The MVR plugin uses the deconvolution as outlined in the wiki:
https://imagej.net/plugins/multi-view-deconvolution

The deconvolution algorithms within are described in this paper:
https://doi.org/10.1038/nmeth.2929.

Unlike the other guide 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx' where MVR batch processing commands in ImageJ Python scripts are used to generate 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.

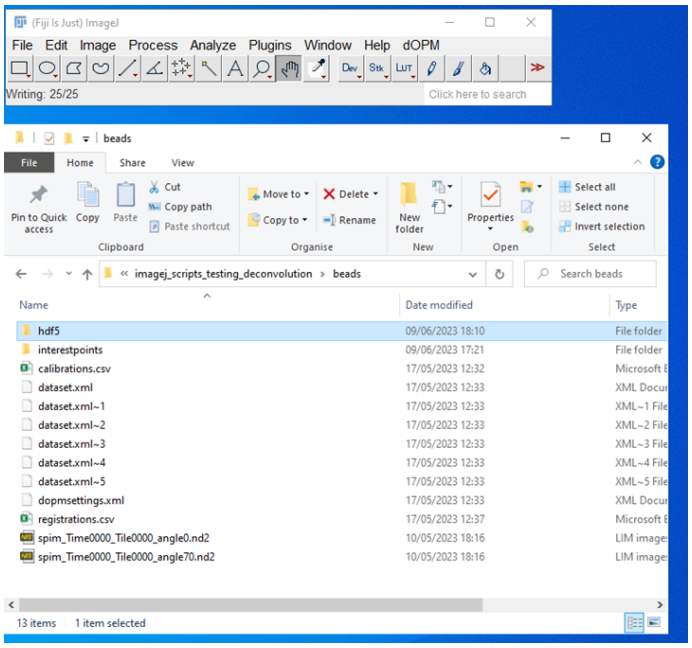
Step 0 – complete steps 1-9 in

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.

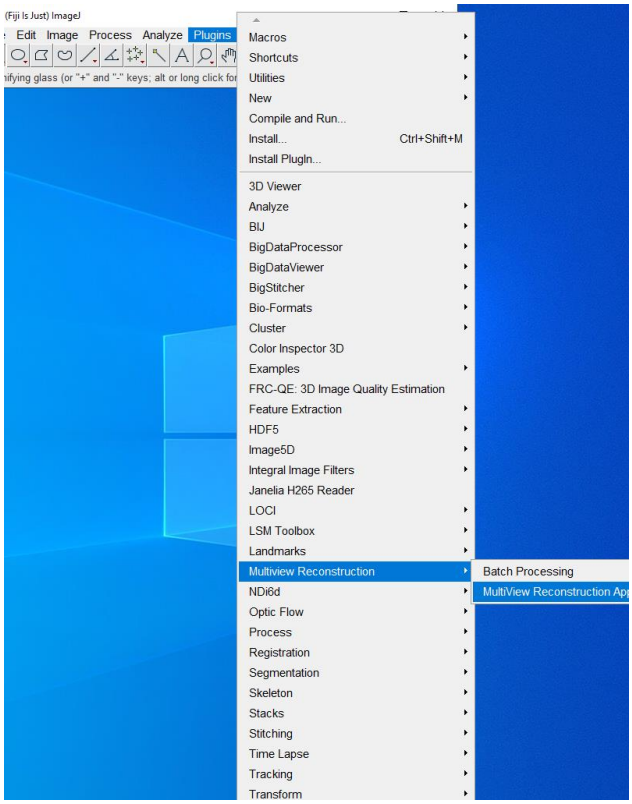
Step 1 – inspect demo sample and bead datasets

- In the other guide you will have practiced making an MVR dataset. In total 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, '.xml' files and some other files '.csv' (the '.csv' are files are not part of MVR plugin instead they are extra files written by additional scripts for the dOPM specific data we work with).
- The screenshot below shows the folder structure for the sample data. For the bead data the folder structure automatically includes a subfolder named 'hdf5' which contains a version

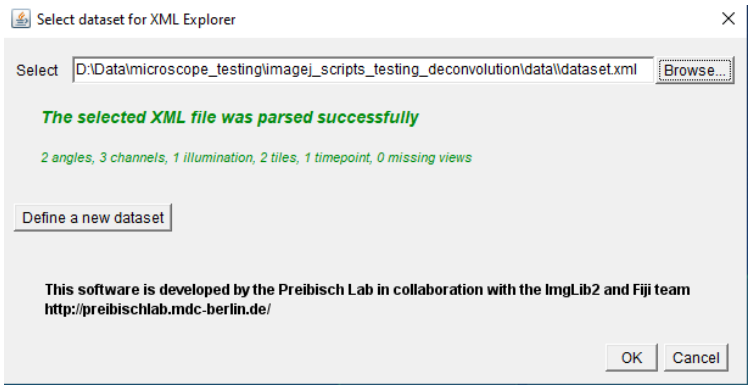
where the raw data has been converted to the hierarchical '.hdf5' data format for fast 3D viewing with the 'Bigdataviewer' tool that is available within the MVR ImageJ plugin – see the other guide this guide refers to and the ImageJ and MVR wiki pages listed above and in this other guide.



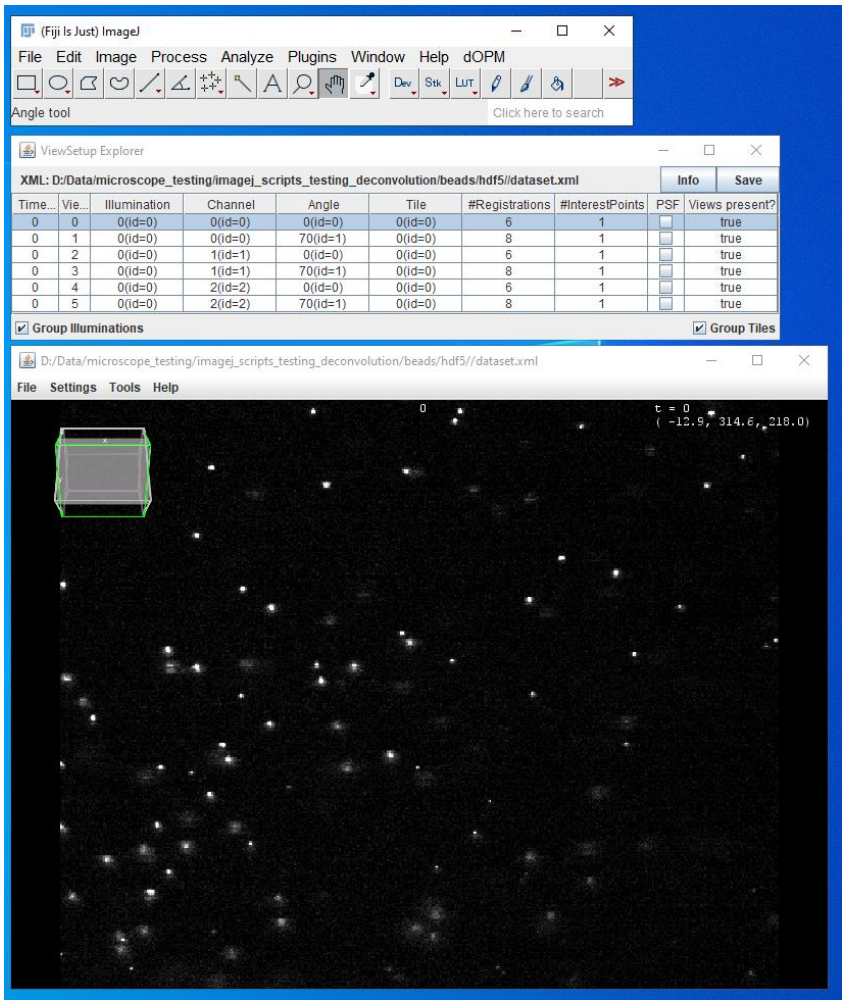
- Open up the MVR bead dataset that you made with the other guide 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx' up to step 4 and check that it is coregistered. Just as with the fusion work, the multi view deconvolution requires successful coregistration of the bead data to work as intended – read this 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx' and references within for understanding.



- Make sure you choose the bead dataset and the ‘hdf5’ subfolder version – i.e., not the sample dataset shown in the screenshot below – this ensures fast 3D viewing



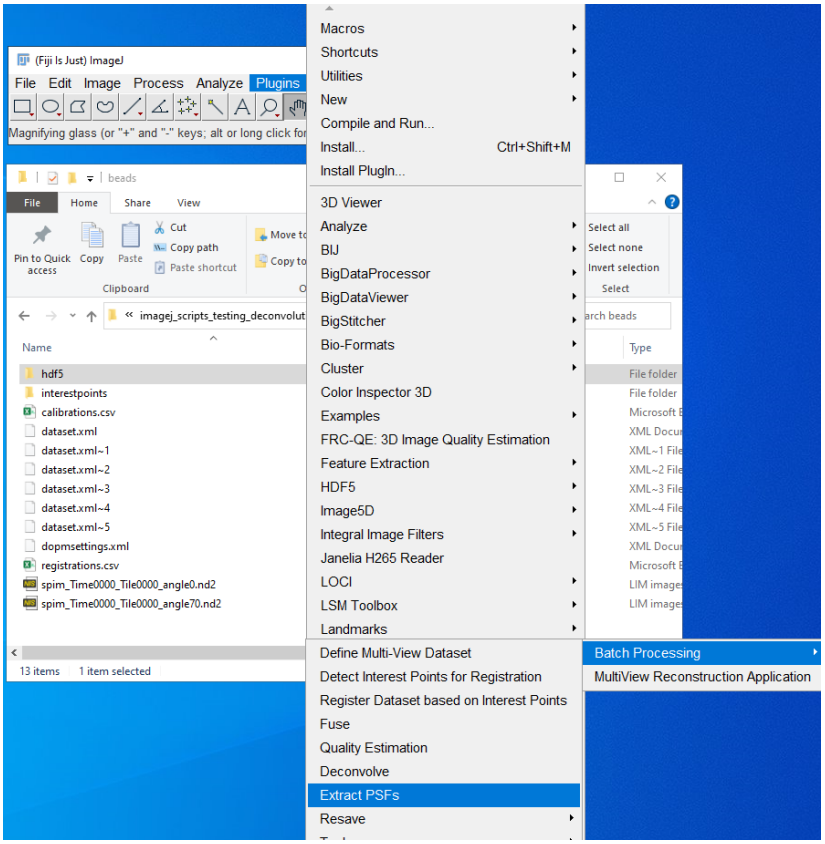
- You should see the following GUIs showing the dataset organisation ‘viewsetup explorer’ and the ‘Bigdataviewer’ tool view of the coregistered bead dataset. Check that the two views are coregistered i.e. the beads from dOPM view 1&2 visibly overlap. See ‘Reslice_dOPM_data_ImageJ_MVR_plugin.docx’ for reminder via that guides steps 1-4.



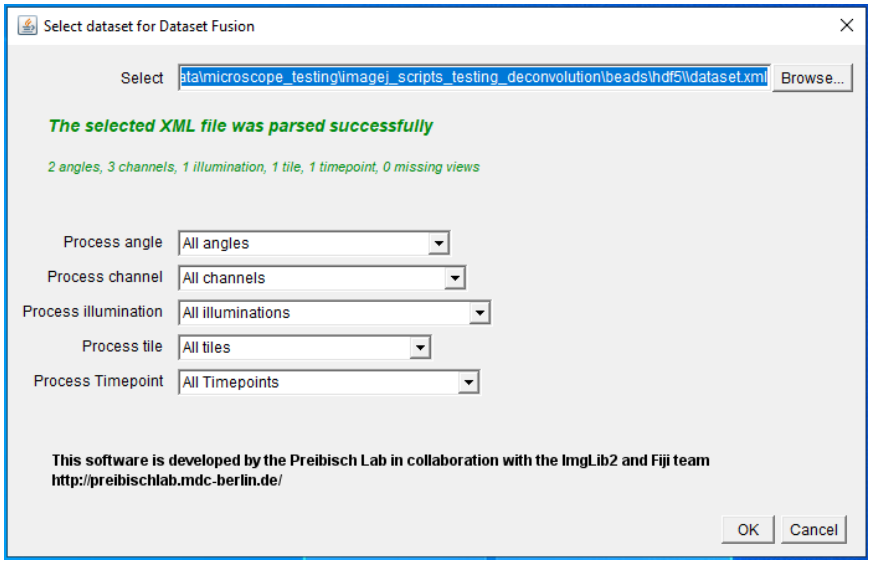
- After confirming that bead dataset is setup and dOPM views 1&2 are coregistered close down the dataset by closing the dataset viewer and the ‘Bigdataviewer’ windows of the MVR plugin in ImageJ.

Step 2 – extract experimentally measured point spread functions

- Only after locating and confirming the coregistration quality of the data proceed to this step. Here we are going to use MVR plugin to extract representative point spread functions from the dataset, for all spectral channels and both of dOPM views 1&2.
- In this case the fluorescent bead dataset is used where the beads 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.
- Now from the ImageJ process tab go to the MVR path > batch processing > Extract PSFs. Batch processing commands are commands you can run with the MVR application tool that we were using before but this time the commands are executed without the MVR application GUI.
- Note: by using ImageJ's Macro recording tool these commands can be recorded and used in scripts like the ImageJ Python scripts used to make the dOPM specific routines used in the other guide 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx'.
  See https://imagej.nih.gov/ij/developer/macro/macros.html for information about ImageJ macros. However, this note is not important for carrying out deconvolution in this guide but shows that the user is free to script solutions for automation of tasks if they are familiar with ImageJ scripting.

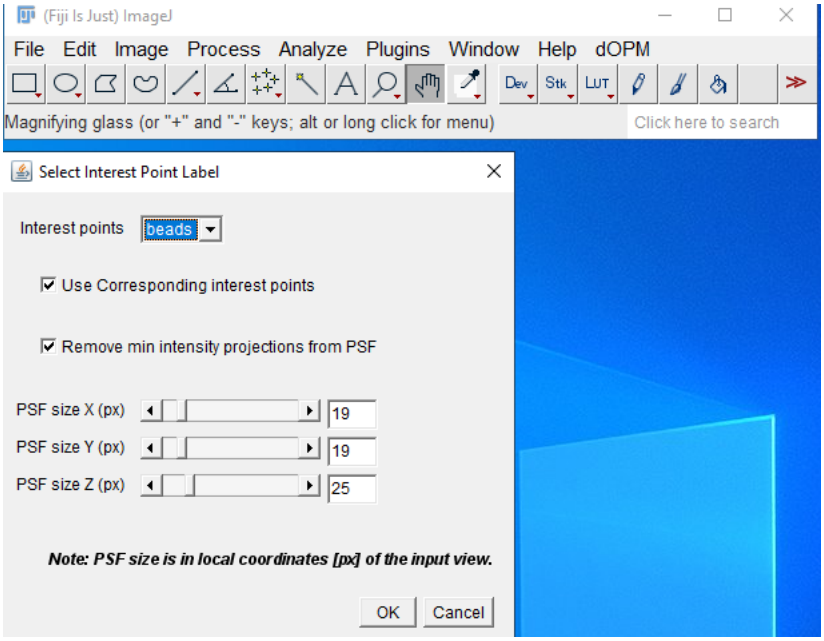


- Once clicked on 'Extract PSFs' the GUI 'Select dataset for Dataset Fusion' pops up.
- Choose the same bead dataset path from Step 1 above.
- Ensure that 'All xxxxxxx' is selected for all the dimensions i.e. angle, channels etc – this is asking which dimensions we want to extract experimental point spread functions – all of them

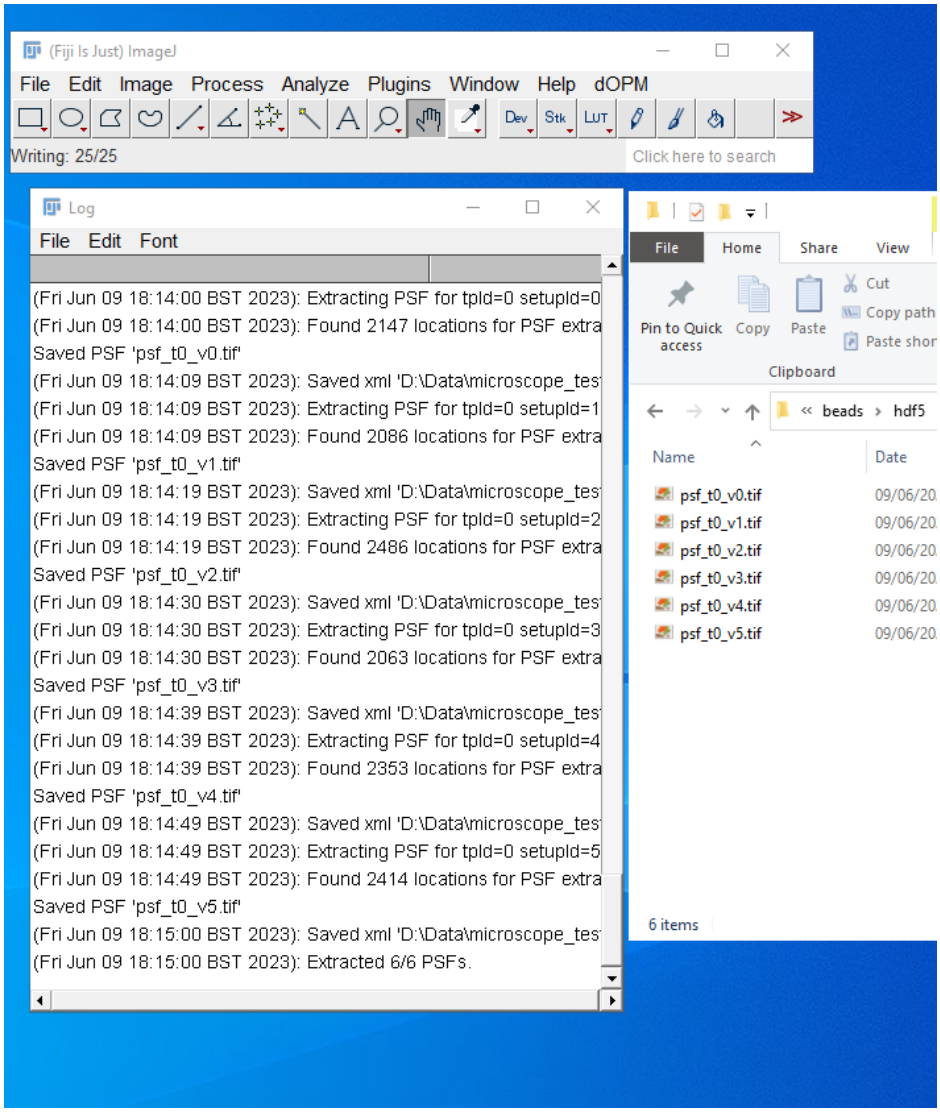


- Press OK and the next dialog box appears allowing control over dimensions of the volume use to store the measured PSF. Leave unchanged the values.
- See the links in the introduction for a deeper understanding of how the authors of the MVR plugin intended this process to be used:

  https://imagej.net/plugins/multiview-reconstruction.
  https://doi.org/10.1038/nmeth.2929.
  https://imagej.net/plugins/multi-view-deconvolution



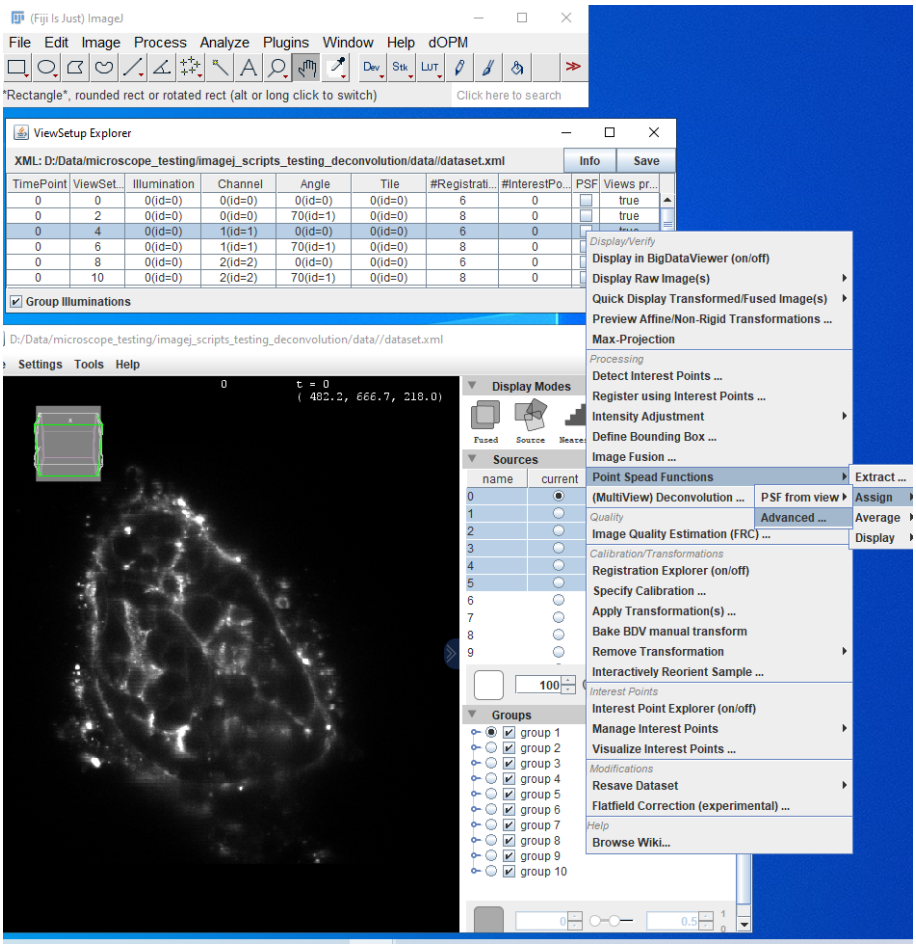
- Run the routine and the log window shows progress. The plugin is extracting PSF estimates from the data, from each channel, and each view. In this demo dataset case, there are two views and 3 channels so 6 PSFs in total.
- The batch command also automatically saves the measured PSF volumes as tiff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



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

- Now open the dataset, using MVR application GUI interface, 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 now has experimentally determined PSFs for each view & channel pair
- Whenever you want to carry out multiview deconvolution of data using MVR for dOPM data, you will need to have data containing 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. In the following steps you manually assign point spread functions to each dOPM view and channel pair. For this two-view, three spectral channel data, 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 a biological sample dataset: open the sample dataset from the demo and that has already been setup from following the previous guide ‘Reslice_dOPM_data_ImageJ_MVR_plugin.docx’
- Looking at the screenshot 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’.

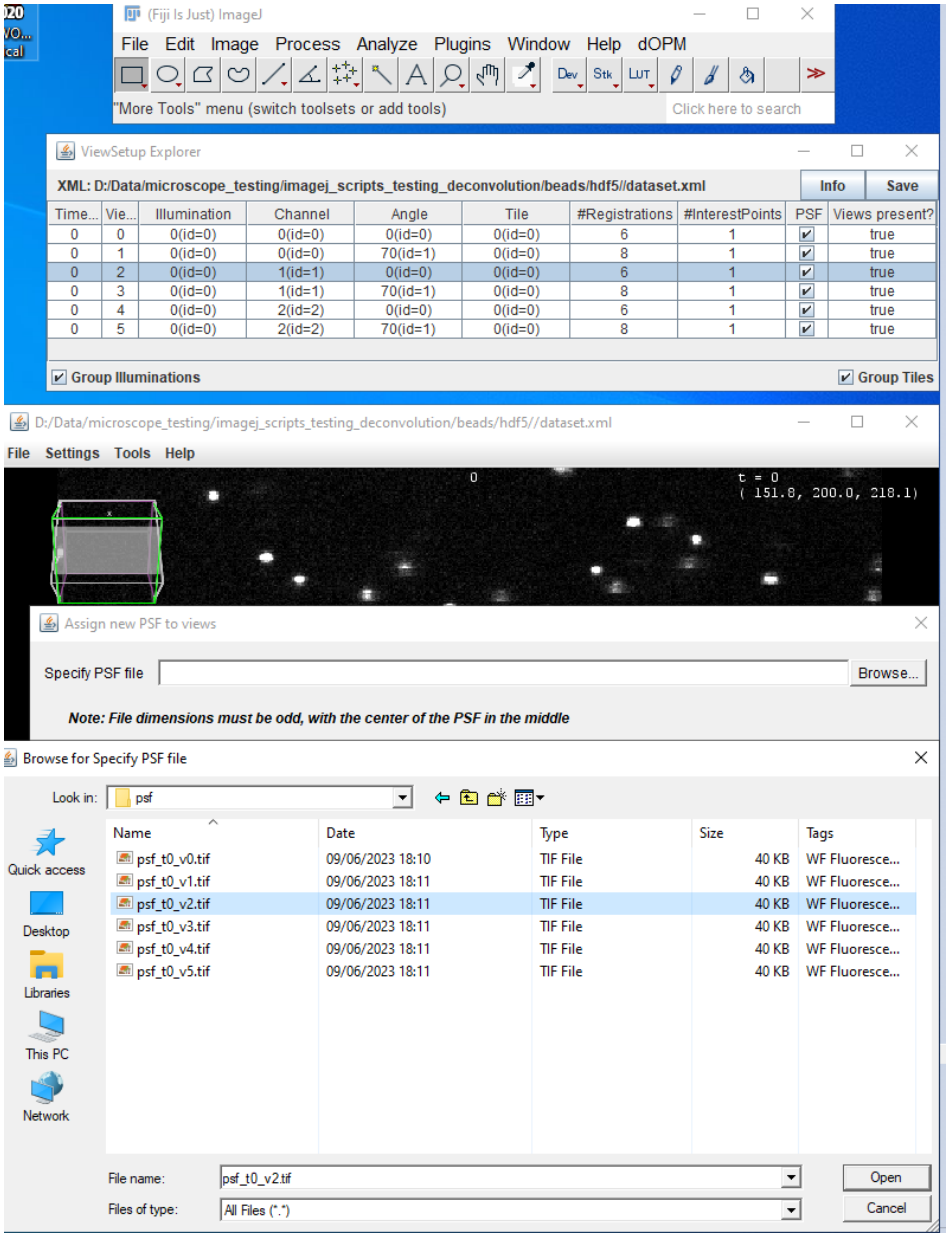


- Another dialog box appears with multiple options in a drop-down list – choose ‘assign PSF to all selected views’. Note because you only chose one dOPM ‘view’ + ‘channel’ you are only assigning one PSF to one particular ‘view’ + ‘channel’ combination i.e. ‘view 1 + channel 0’ and you will need to do 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 the Steps 0-2 in this guide.
- Note the psf stiff stacks as shown in the screenshot below for the bead dataset have automatically bean 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

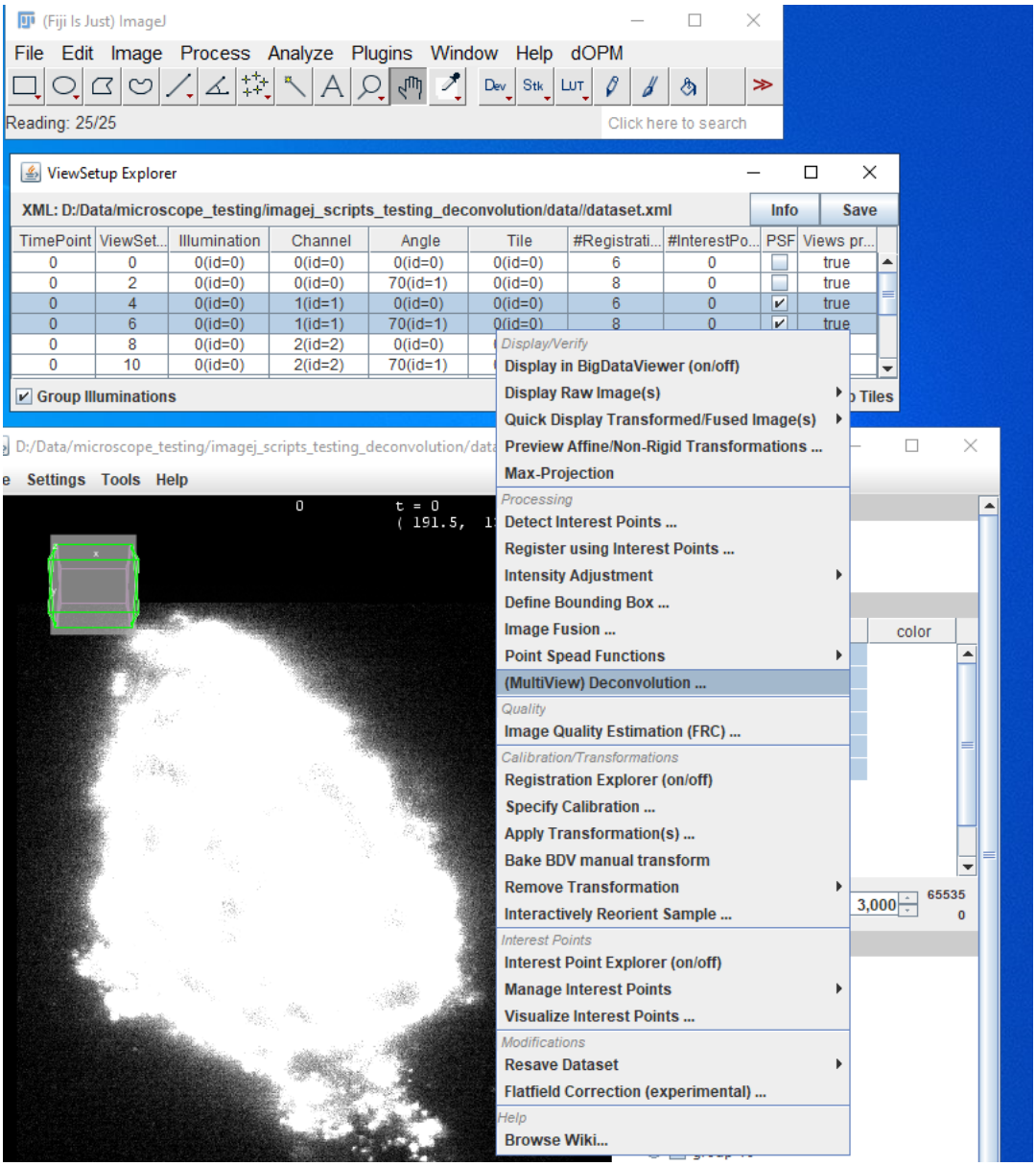
case) assigned to the dimensions from left to right in the ‘viewsetup explorer’ as shown in the screenshot below.

- 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

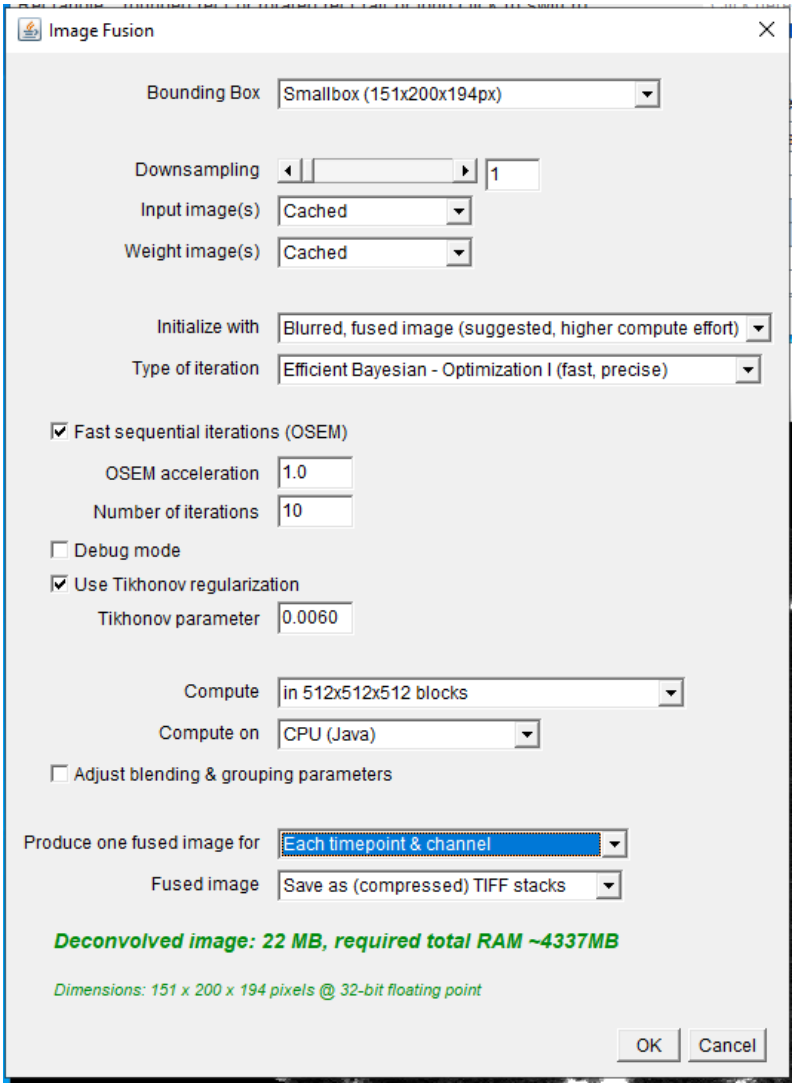


Step – 4 run multi-view deconvolution with default settings

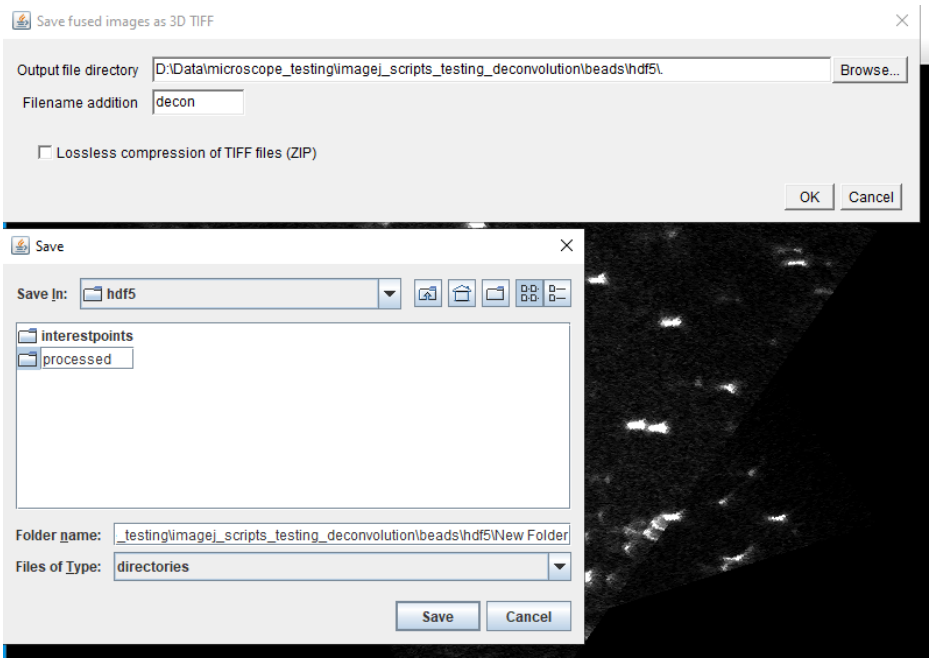
- Now each PSF is assigned to each view + channel in the sample dataset you can do multiview deconvolution. Without assigning PSFs from the bead dataset the MVR plugin would not allow you to do multiview deconvolution.
- Right click on the explorer and click on '(Multiview) Deconvolution..'



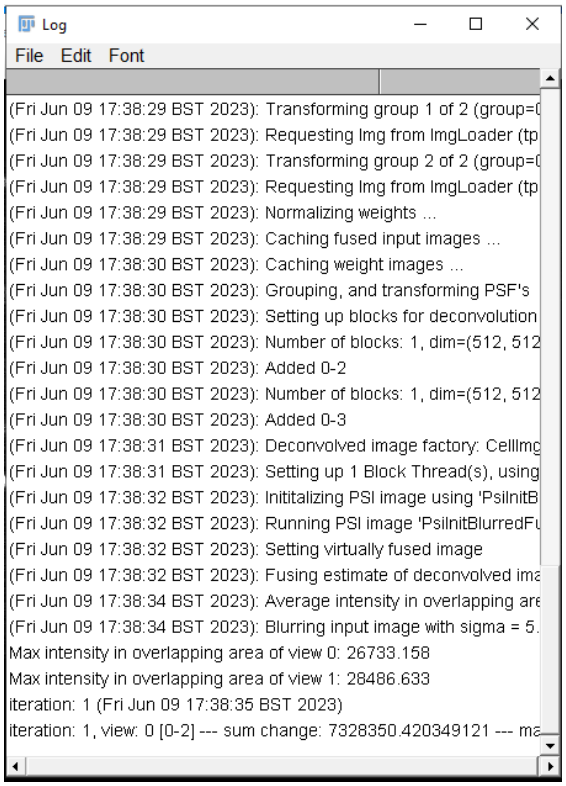
- Leave all default settings as indicated in the screenshot below.
- Note: if you do this on a cropped volume the processing time can be reduced by orders of magnitude which is important as Multiview deconvolution is computationally expensive and typically takes minutes to hours depending on volume size. For this guide we recommend using a bounding box of around 200x200x200 to practice using the routines.
- See 'Reslice_dOPM_data_ImageJ_MVR_plugin.docx' 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 file 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 and explore them in ImageJ or any other tool for 3D processing. In the example below a small bounding box was used to define a cropped volume to speed up the deconvolution for this demonstration.