



## Visualizing three-dimensional data produced by the 3D Cell Explorer

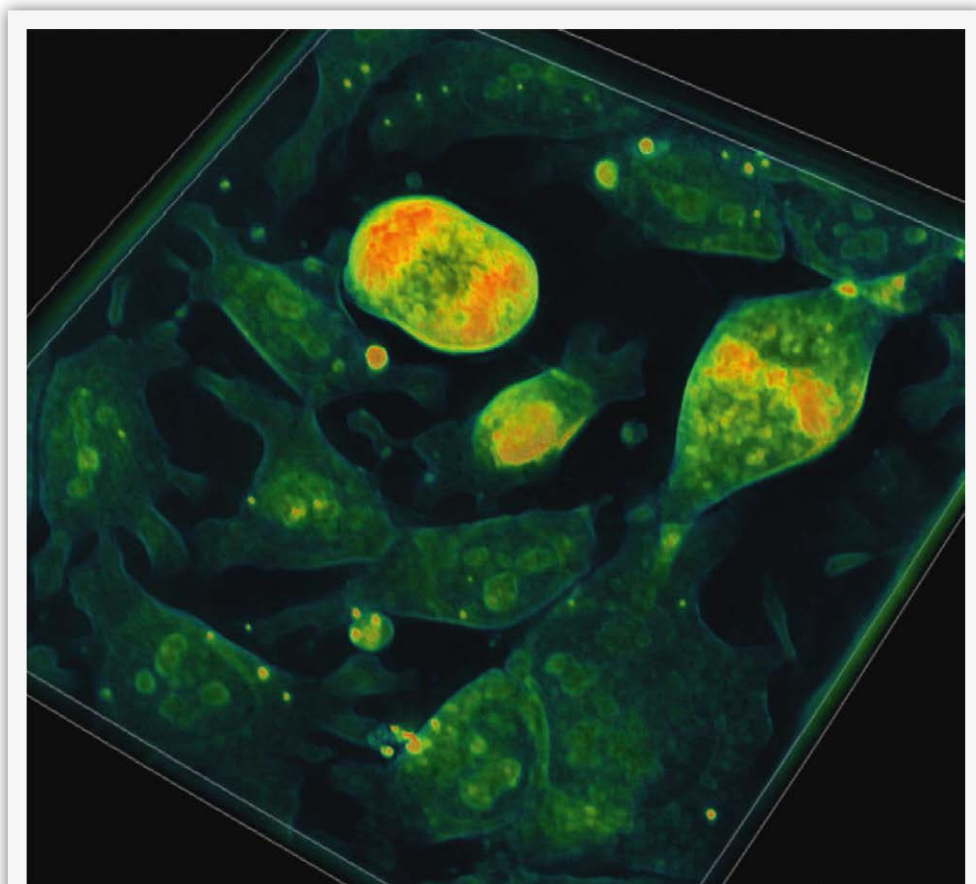
**Nanolive SA**  
Chemin de la Dent d'Oche 1a | 1024 Ecublens | Switzerland

## Abstract

Visualizing tomographic data in an efficient way is essential for designing further 3D image analysis. This is a rather complex task that requires advanced visualization modes, easy camera control, precise transfer function interface and tight control of the transparency. In this application note we will describe how to export properly the data produced with the **3D Cell Explorer**, how to prepare the data and how to visualize it in optimal ways. To do so, we will guide you through the usage of two open source software programs, the 3D ImageJ suite and **Tomviz**. Later application notes will explain how to further analyze data in such software.

## Introduction

A major problem in the field of three-dimensional imaging is to visualize three-dimensional data in simple but powerful ways that allow the experimenter to grasp the complexity of a dataset at a glance (**Figure 1**). Numerous dedicated solutions are developed in the academic or private world. We choose here to present you two open source software programs that are, to our view, the best in the field; the 3D ImageJ suite of FIJI (Schmid, Schindelin, Cardona, Longair, & Heisenberg, 2010) and Tomviz, a tomographic visualization tool (tomviz.org).

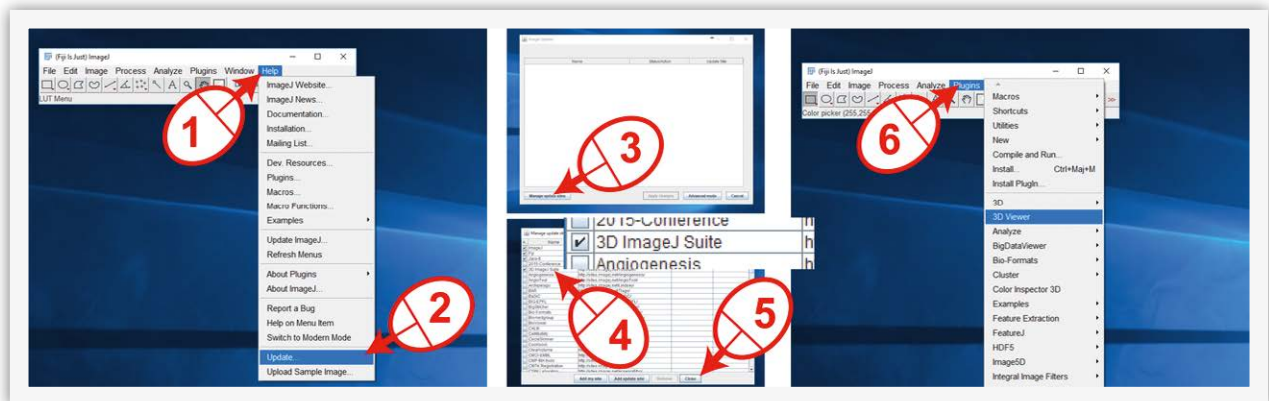


**Figure 1:** Visualization in Tomviz (tomviz.org) of a monolayer growing mouse embryonic stem cells acquired with the 3D Cell Explorer

## Prerequisite

Before being able to generate a nice 3D visualization of your data, three software programs need to be installed. First, you will need the latest version of **STEVE**, the software that allows to control your 3D Cell Explorer, second, the **latest version of FIJI** (Schindelin et al., 2012; Schneider, Rasband, & Eliceiri, 2012) ([fiji.sc](http://fiji.sc)) with the **3D ImageJ suite** (**Figure 2**) and third, the latest version of **Tomviz** ([tomviz.org](http://tomviz.org)).

Update FIJI to get the 3D ImageJ suite, click **Help > Update...** (**Figure 2, steps 1-2**) press **manage update** (**Figure 2, step 3**). It could be that FIJI asks you for a couple of update rounds. Just go through, restart as many times as necessary (usually just once) until you can go through the process by ticking on the **3D ImageJ suite** option (**Figure 2, step 4**). Now you can close (**Figure 2, step 5**), restart FIJI and check that **3D** and **3D viewer** are present in the **Plugin** menu.



**Figure 2:** Installation of the 3D ImageJ suite

Finally, you will need a nice acquisition file generated with the 3D Cell Explorer. We must insist on the obvious; everything starts with having excellent acquisitions. If you are not sure to be in optimal imaging conditions, please follow our guidelines (<http://nanolive.ch/supporting-material/>) or contact us (<http://nanolive.ch/contact/>), we will answer to you with great pleasure.

## Exporting data

The first step after your dataset acquisition is to export the data under the proper format. Both visualization software programs work with stacks of images under .tiff format. Use the exporting tool of STEVE to export your dataset as a **anyname.tiff** stack. After your acquisition, when your dataset is loaded into STEVE (if not load it with File > Load), do **File > Export** (**Figure 3, steps 1-2**), in the panel that opens change format to Tiff (**Figure 3, step 3**), provide a location and a name to your file and export it (**Figure 3, steps 4-5**). The product of this export contains 96 slices, covering the acquired Z-dimension. It might well be that not all slices contain useful information on your sample. Such slices will only generate noise in your 3D visualization and need to be removed from the **anyname.tiff** file before opening it in a 3D visualization tool.

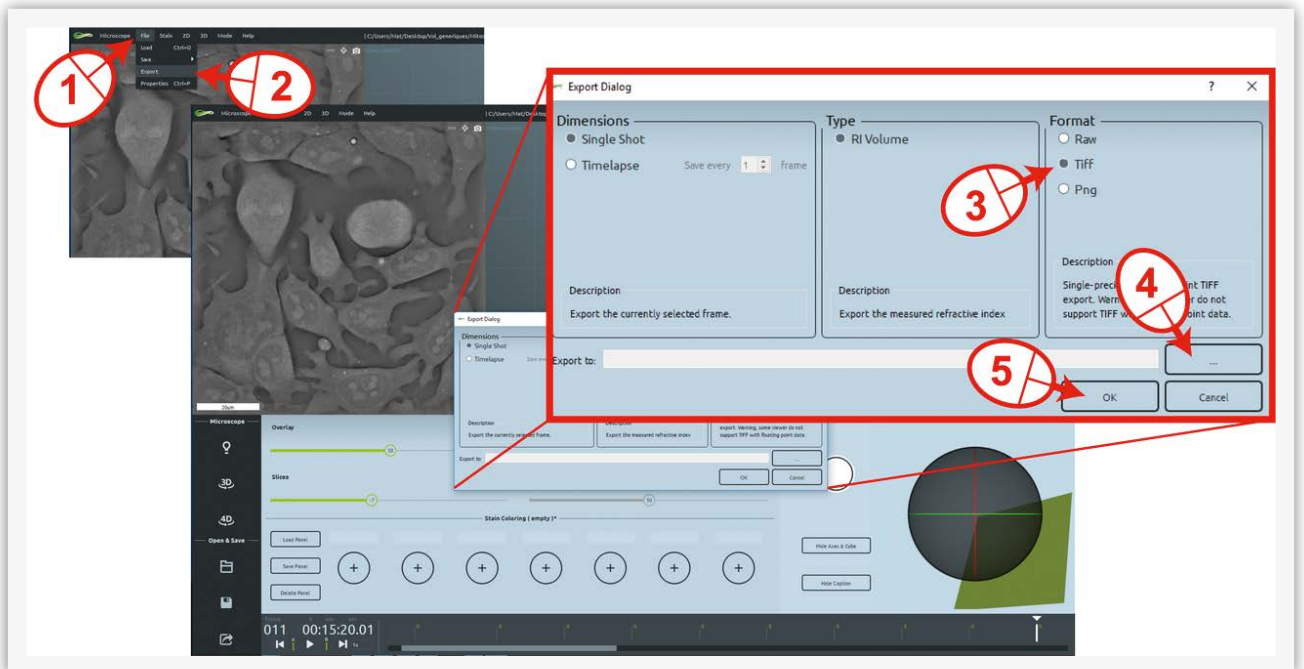


Figure 3: Exporting .tiff from STEVE

## Removing unwanted Z-slices

Start FIJI, drag and drop the **anyname.tiff** file in the ImageJ panel or open it with the open function, you will be able to navigate through the slices that compose your **anyname.tiff** file. Go to a slice that contains proper signal (best focus), you can now adapt the display thresholds: **Image> Adjust>Window/Level** then **press Auto** such that the signal is shown with the best dynamic (**Figure 4, steps 1-4**).

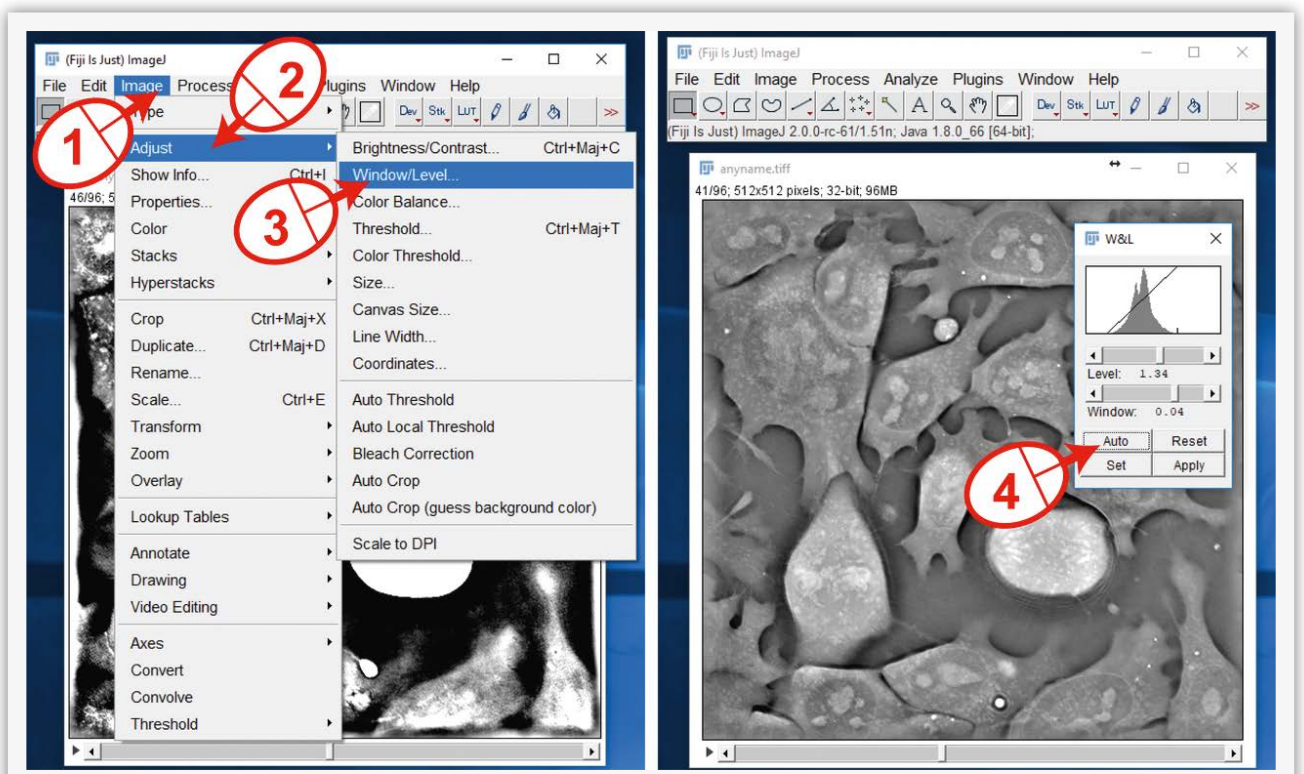


Figure 4: Thresholding with FIJI

The next step is to note the number of the first and last slices that contain relevant signal. Then click on **Image>Stack>Tools>SliceKeeper**. You can now enter the boundaries of a new stack of images, enter the first and last slice numbers you just defined as well as an increment of 1. The result is a new stack that you need to save under the name **anyname\_trim.tiff** for example (**Figure 5, steps 1-3**). This file is now ready to be opened in the 3D viewer of FIJI or in Tomviz.

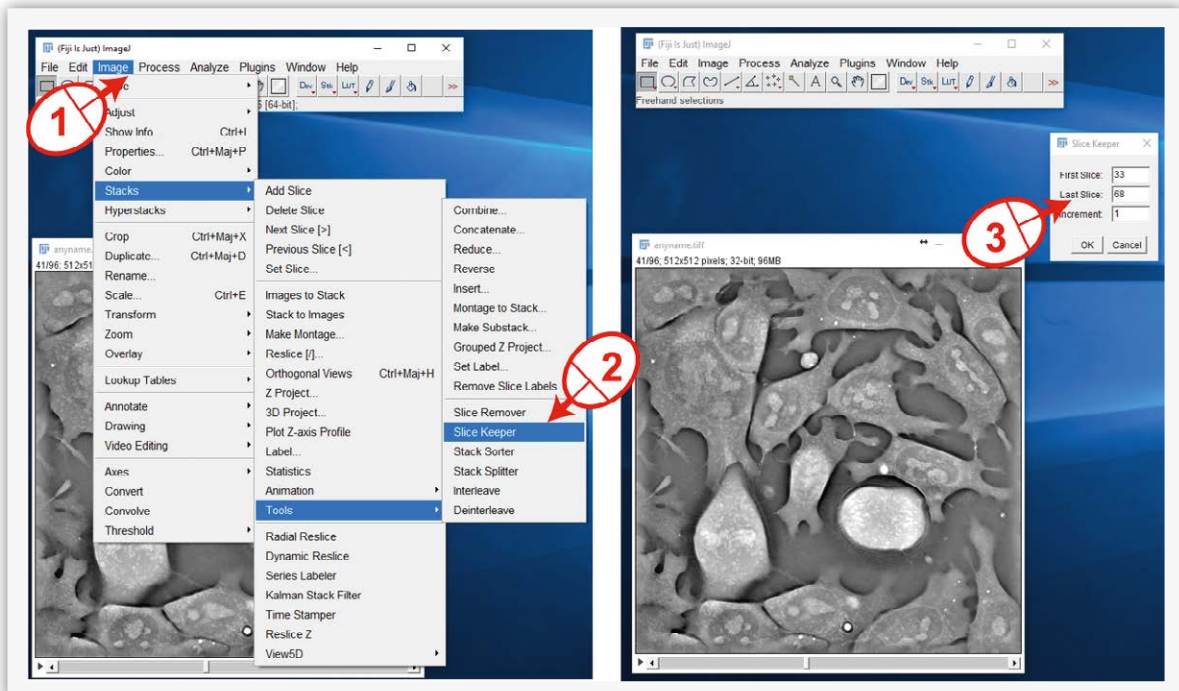


Figure 5: Creation of an optimal .tiff stack for visualization

## 3D visualization in FIJI

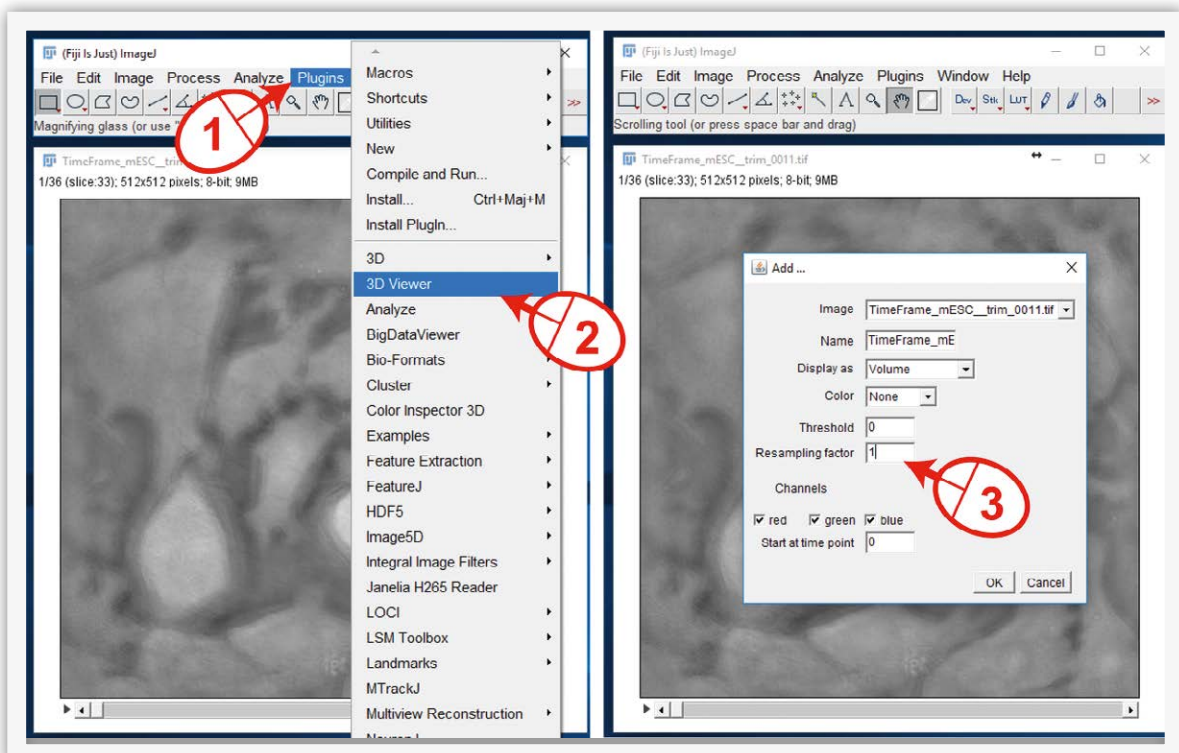


Figure 6: loading a .tiff file in FIJI's 3D viewer

Open the file **anyname\_trim.tiff**, then go to **Plugin>3D viewer** (Figure 6, steps 1-2), answer yes if asked to transform into 8-bit data, ignore the Java prompt even if some red text appears! Finally, in resampling factor, enter 1 if you have a powerful computer, leave 2 otherwise, the image will be less fine but it will be lighter to handle for your computer (Figure 6, step 3). This option for visualization is the less powerful, however the analysis capabilities are powerful due to the installation of the 3D ImageJ suite as we will see in a following application note.

## 3D visualization in Tomviz

### Loading and adding a volume visualization

Open the file **anyname\_trim.tiff** within Tomviz: **File > Open Data**. Your dataset appears in the column on the left (Figure 7). Tomviz works with a workflow logic, you will apply “treatments” to your dataset that will allow you to observe it the way you like. First, add a volume visualization: **Visualization > Volume** (Figure 7, steps 1-3), and activate the function **enable gradient opacity** (Figure 7, step 4).

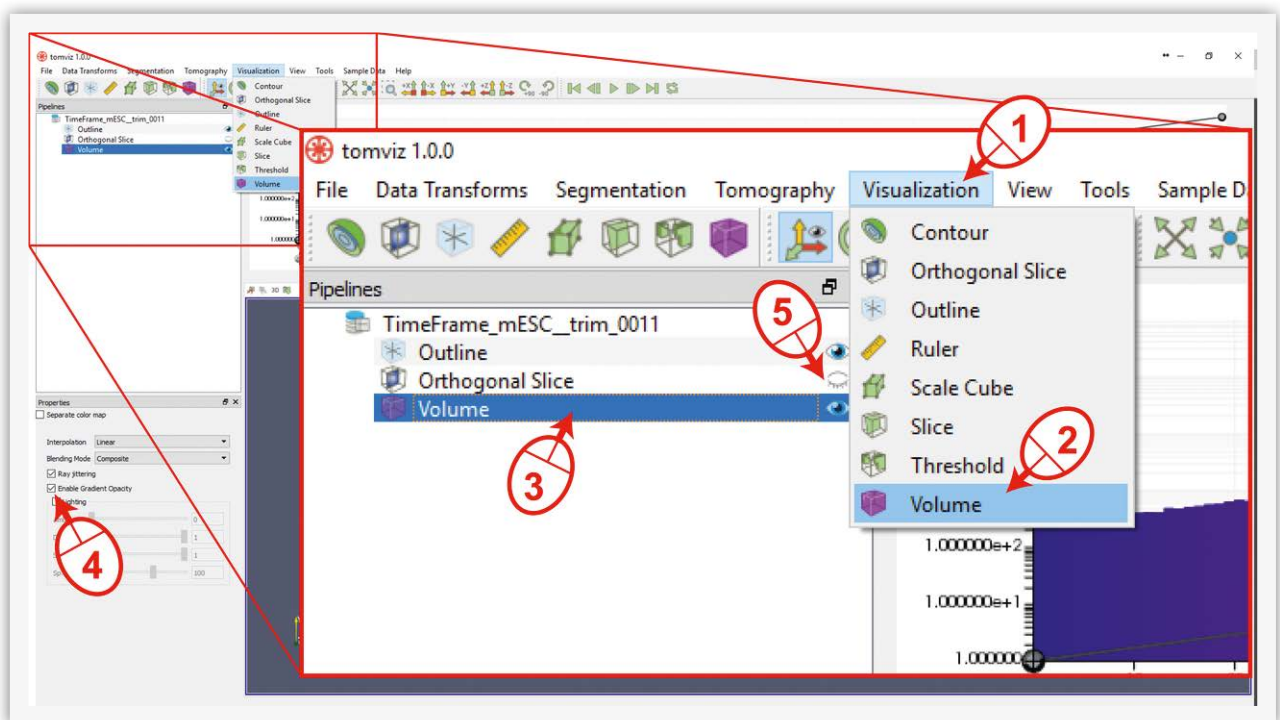


Figure 7: Visualizing a .tiff stack as volume in Tomviz

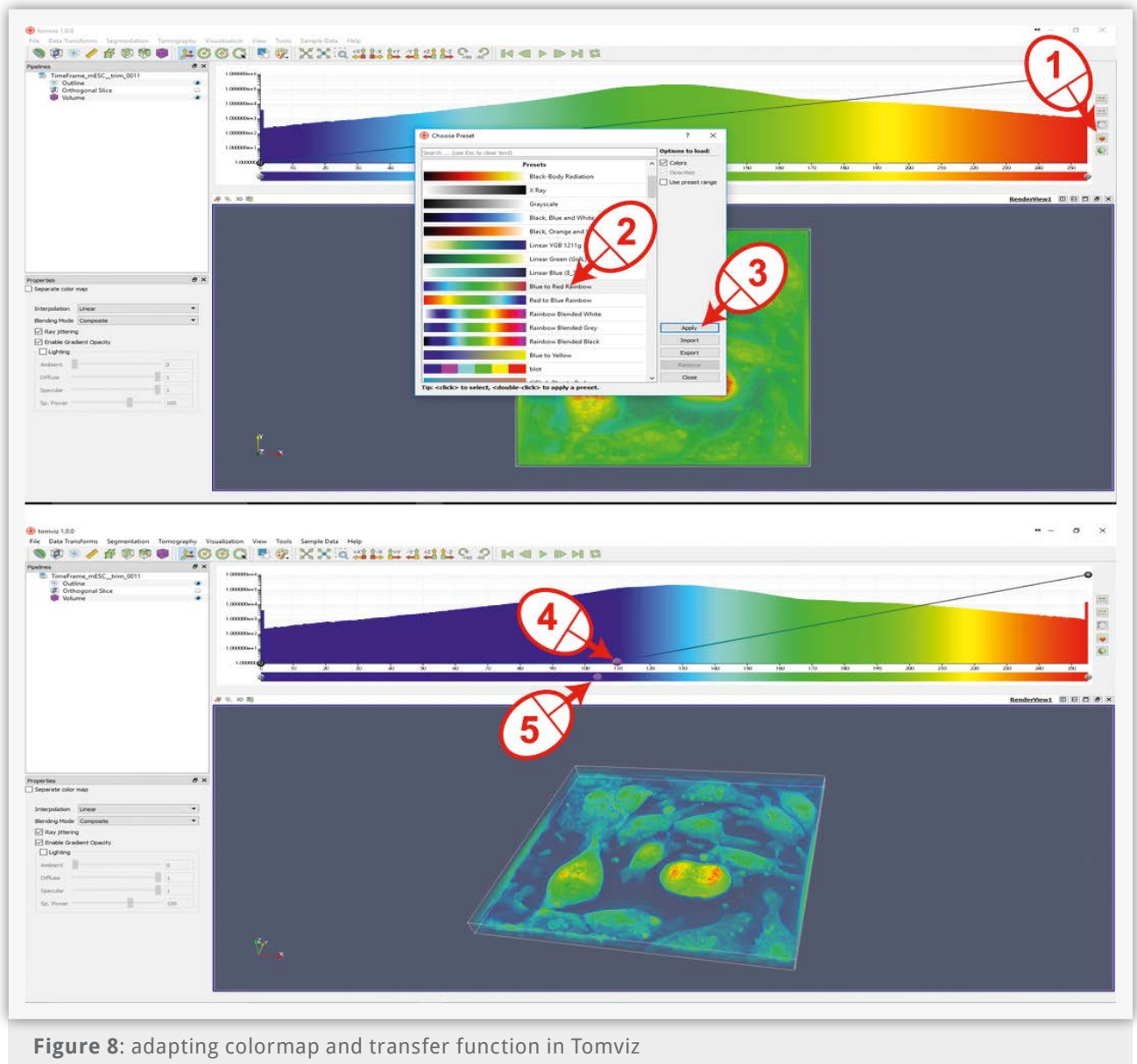
Note that in versions of Tomviz later than 1.0, we observe that this option is activated by default and is not accessible anymore. Second, inactivate the orthogonal slice view that would otherwise superpose on your volume view (Figure 7, step 5).

You will find in the **Visualization** menu a set of view modes that you might enjoy, do not hesitate to test them. Mistakes are of no cost since you can simply remove any unwanted steps from the pipeline column and go back to a previous better state.

### Changing colormap and adapting the transfer function

You can now change the way your volume is displayed, go on the right and click on the colormap folder and choose one you like. We like the Blue to Red Rainbow (Figure 8, steps 1-3).

You will observe that the volume changes color as well as the histogram. Let's focus on the histogram (Figure 8).



The upper central panel displays the distribution of your signal and how it scales with color. You can also observe a straight line over it, this is the way the signal is transferred for display, i.e. the transfer function (Figure 8, step 4). We recommend first to add one point on the transfer function by clicking once on the line and to reduce the transfer function to 0 up to when you see meaningful signal disappearing. In this way you remove the background signal even more.

We like then to scale the color map accordingly by adding a cursor point on the left and drag it below the transfer function point or close (Figure 8, step 5). During the whole process of color adjustment, we recommend observing the volume under different angles in the lower central panel. By now, you might have a volume that looks nice. At this stage, we recommend playing with colormaps and the transfer function to obtain the rendering you like. Finally, Tomviz proposes a clear user manual that will get you far in your visualization capabilities. Therefore, go to **Help > User Guide**.

## References

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., ... Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods*, 9(7), 676–682. <https://doi.org/10.1038/nmeth.2019>

Schmid, B., Schindelin, J., Cardona, A., Longair, M., & Heisenberg, M. (2010). A high-level 3D visualization API for Java and ImageJ. *BMC Bioinformatics*, 11(1), 274. <https://doi.org/10.1186/1471-2105-11-274>

Schneider, C. A., Rasband, W. S., & Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7), 671–675. <https://doi.org/10.1038/nmeth.2089>