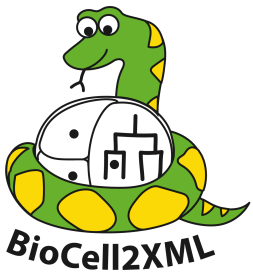


BioCell2XML User Guide

Publication

BioCell2XML was developed by Markus Pennerstorfer, Günther Loose and Carsten Wolff. If you use the program for your work please cite the publication:

Pennerstorfer, M., Loose, G. & Wolff, C. (2019). *BioCell2XML*: A novel tool for converting cell lineage data from SIMI BioCell to MaMuT (Fiji). *Development Genes and Evolution*. https://doi.org/10.1007/s00427-019-00633-9

Installation

BioCell2XML is a program written in *Python* and will run with *Python* version 3.0 or higher. To use *BioCell2XML* you have to have *Python* installed on your machine. MacOS normally comes with *Python 2.7* preinstalled, but it has to be updated to version 3.0 or higher. Instructions for how to install *Python* for any OS are provided at https://www.python.org/downloads/. Installation of additional modules is not required.

The goal of *BioCell2XML* is to transfer cell lineage data created with *SIMI BioCell* (Reality Motion Systems GmbH) to the specialized XML-format that can be viewed and edited with *MaMuT*, a plugin that comes with the open ImageScience software *Fiji* (current Version: 2.0.0). *Fiji* can be downloaded from https://imagej.net/Fiji/Downloads. Details on installation and use of *MaMuT* can be found at https://imagej.net/Getting_started_with_MaMuT.

Requirements

Names of all input files and directories must not contain any space or non-ASCII characters.

If the program is used in interactive mode, the two *SIMI BioCell* files (SBC, SBD), the H5.XML file, and the program file ***BioCell2XML.py*** all have to be located in the same directory.

Image data preparation

File format

By default, many of the 4D microscopy systems provided with the *SIMI BioCell* software save image series in the licenced LWF (lurawave) format (Luratech/Foxit). Unfortunately, most image software will not open these files without installation of a commercial codec.

For use with *Fiji* the images have to be converted to TIFF and then to HDF5/XML.

If your image series are in other format than LWF you can transfer them to TIFF using batch conversion in *Fiji* or other freeware.

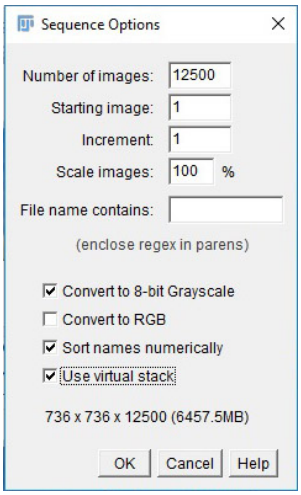
For conversion of LWF files we use the freeware XNView, which allows viewing and batch-conversion of this image format: https://www.xnview.com/en/.

Cropping images

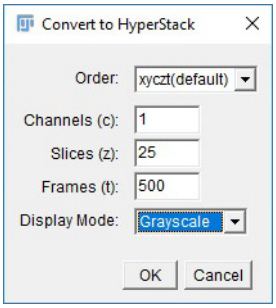
Conversion to TIFF format will increase file size. If your images contain empty regions you can crop these parts from the images to reduce file size and save disc space.

Before cropping you should load your TIFF images into *Fiji* and view them as hyperstack. This will enable you to view the data three-dimensionally over time and check e.g. if the specimen changes position during development. Depending on this observation the crop distances should be chosen.

Import the image series to *Fiji* by selecting '*File/Import/Image Sequence*`, navigating to the directory containing your TIFF images and selecting the first image.

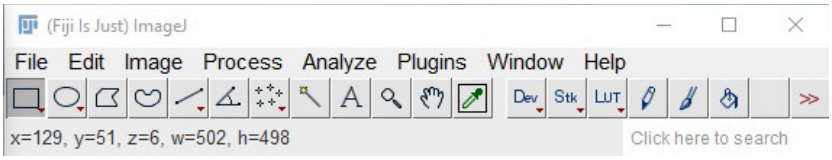


Select '*convert to grayscale*' and '*use virtual stack*'. This will open a viewer where you can browse through your image series, which however are not yet sorted conveniently. From the Image-menu select '*Hyperstacks/Stack to Hyperstack*'.



In the pop-up window enter the dimension information of your image series. Be sure that *channels * slices * frames* equals exactly the number if images in your series.

This will open a window that allows you to scroll through the images in the dimensions channel, slice and time. Evaluate your image data set using these tools and choose the size of your crop window accordingly. A crop window for x/y can be created and modified using the frame select tool. Note the values that are displayed for *x* (distance from the left image border), for *y* (distance from the upper image border), for *w* (width) and for *h* (height) displayed below the toolbar. These values will be needed later when running *BioCell2XML*.

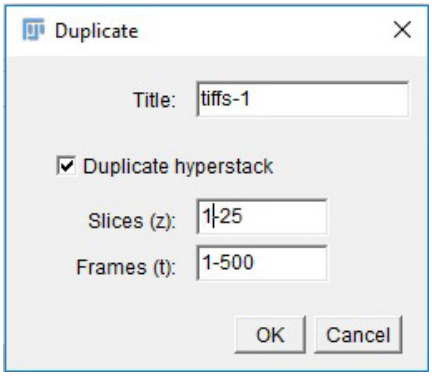


In this example we crop images in *Fiji* by using hyperstack. Depending on the size of your data there can be limitations caused by the RAM of your computer. Images can also be cropped by using batch-processing in *Fiji* or other freeware (e.g. IrfanView). For this the noted values for *x*, *y*, *w* and *h*. must be used.

Select '*Image/crop*'. This will crop the image series in x and y to the size of the selected frame. A new window will open displaying the cropped series.

To crop the data also in z- and/or t-dimension (remove slices or frames) select '*Image/Duplicate*'.

You can specify a new range of slices and frames in the pop-up window. A new hyperstack with these settings will be created, which can take a few minutes to compute. This hyperstack can then be saved to a new folder using '*File/Save As/Image Sequence*'.

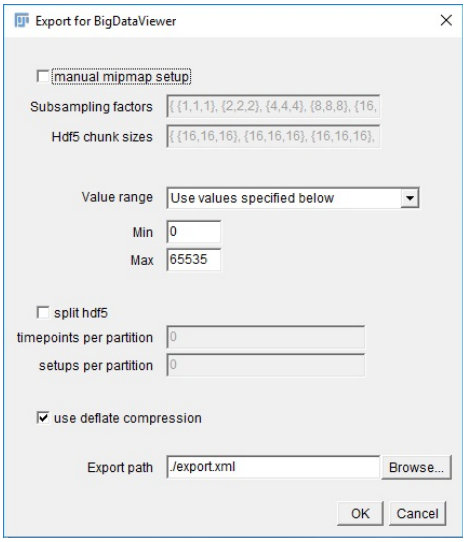


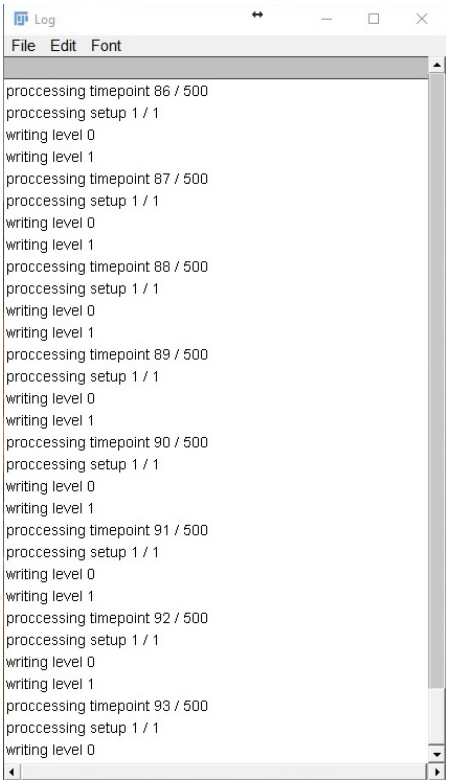
Conversion to HDF5/XML format

MaMuT works with a container format called HDF5 that was developed specifically for real time visualization of large volume data. To convert the TIFF images series to this format we use *Fiji*'s *Big Data Viewer* plugin.

If you have cropped your images using a hyperstack in *Fiji* (as described above) you can simply proceed using the open, cropped, hyperstack. If you have not cropped your data or cropped them using other software, load the TIFF image sequence as a hyperstack, using the correct dimensions, as described at the beginning of the "Cropping images" section.

With the hyperstack window open select '*Plugins/BigDataViewer/export current image as XML/HDF5*'.





Keep the default settings for conversion. Select a directory where the converted image data should be saved and a filename. Keep in mind that these files can be rather large. A log window will open showing the progress of the conversion. Computation can take some time. Two files with the same filename will be created, carrying two different extensions: .h5 and .xml.

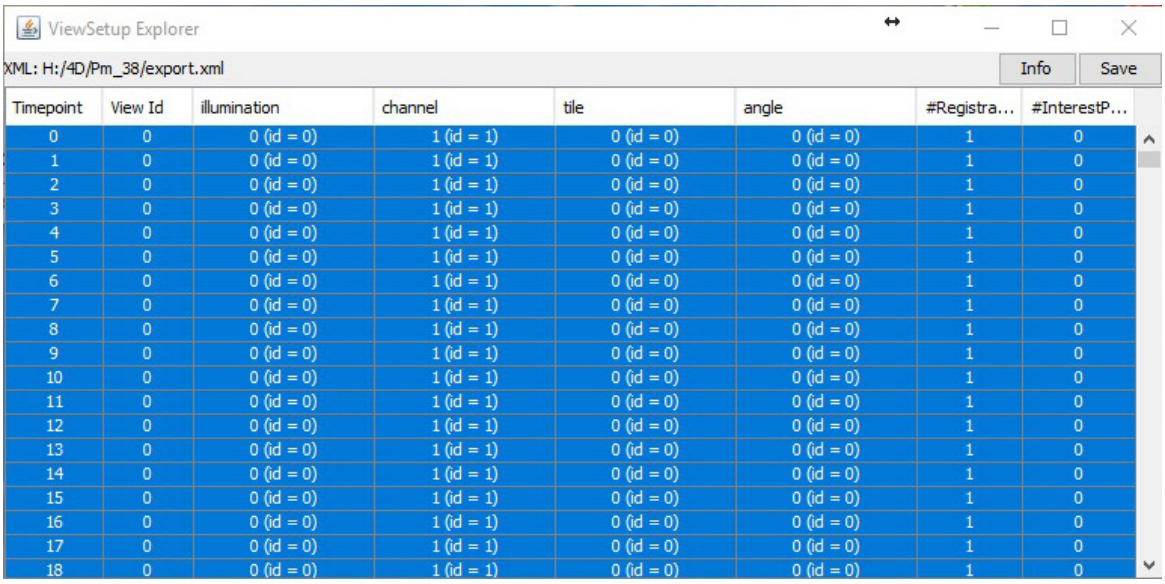
Scaling

In order to obtain the correct proportions in your image data when viewed in *MaMuT* it is necessary to add voxel size information to the XML file. The voxel dimensions can be calculated if the optical specifications of the used microscope are known or by e.g. using an object micrometre. Ideally, the height and width of the image in µm was documented when the images were recorded. Dividing theses value by the x- and y- resolution of the image (e.g. 1024, 1024) will yield the voxel-width and voxel-

height. The distance between slices in µm is usually entered in the recording settings of the microscope software. This value can be used directly as voxel-depth.

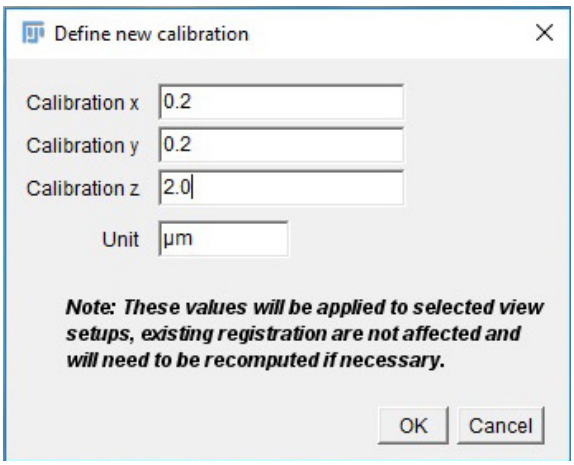
From the '*plugins*' menu select '*Multiview Reconstruction/Multiview Reconstruction Application*'. In the window navigate to the previously created XML file. If the data was read correctly click '*OK*'.

A viewer window showing the image data will open, where you can navigate through your data. Another window called *ViewSetup Explorer* will open showing a list of all time points in the XML file. Select all time points using by right-clicking and '*select all*'. For all selected timepoints right-click again and select '*specify calibration*'.

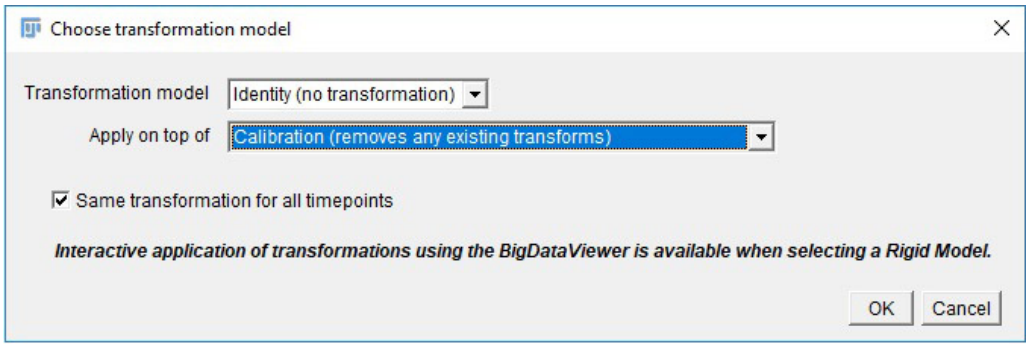


Timepoint	View Id	illumination	channel	tile	angle	#Registra...	#InterestP...
0	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
1	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
2	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
3	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
4	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
5	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
6	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
7	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
8	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
9	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
10	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
11	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
12	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
13	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
14	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
15	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
16	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
17	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0
18	0	0 (id = 0)	1 (id = 1)	0 (id = 0)	0 (id = 0)	1	0

In the window enter the values of your voxel size (voxel-width, voxel -height, voxel-depth for x, y and z, respectively). For unit enter ‘µm’.



Next these settings have to be applied to the data. For this right-click again on all selected timepoints and choose ‘*Apply Transformations*’. In the pop-up window select ‘*Identity (no transformation)*’ and ‘*Calibration (removes any existing transformations)*’.

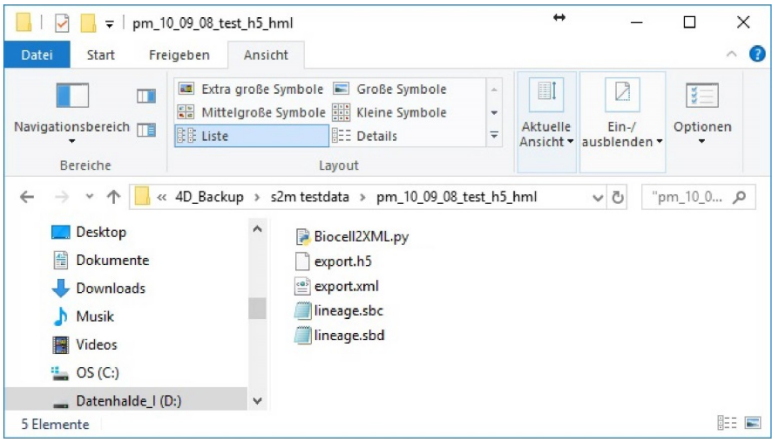


Finally, in the upper right corner of the *ViewSetup Explorer* window click save. A new XML file with the modified calibration settings will be saved to the same directory using the same filename.

To check for successful calibration, you can rotate the view in the viewer window (e.g. using ‘shift + x’ or ‘shift + y’). The image volume should now show realistic proportions.

Convert *SIMI BioCell* lineage data to XML format

Each *SIMI BioCell* lineage consists of an SBC file and an SBD file. Both these files should be saved in the same directory as the corresponding XML and H5 files that have been created previously. If *BioCell2XML* is used in interactive mode, also the program file '***BioCell2XML.py***' has to be copied to the same folder.



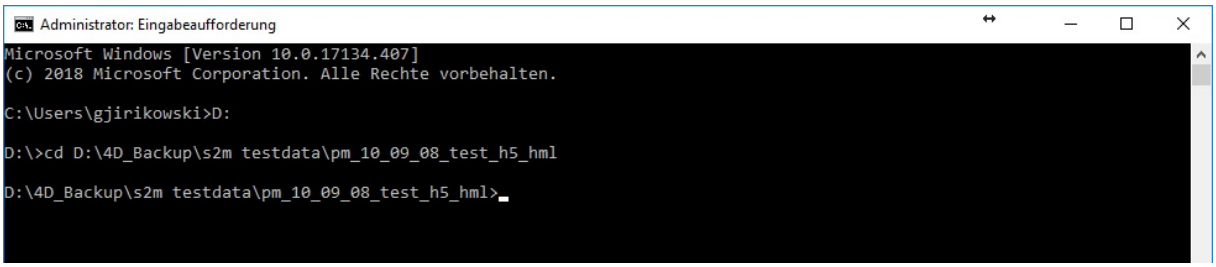
BioCell2XML has to be operated via the command-line interpreter. For users who are less familiar with this, the program can be used in a step-by-step input mode to make its usage as convenient as possible.

Open the command-line interpreter (e.g. under Windows by hitting the windows key and typing 'terminal').

Set the working directory. To change the volume, type the volume letter (e.g. **'D:'**).

Enter **'cd'** followed by the path to the directory containing your files (e.g. copy and paste it from the explorer window). Make sure filenames and directory names do not contain any space characters.

```
cd /path/to/directory
```



Start *BioCell2XML* by entering

```
python3 BioCell2XML.py –p
```

and press the 'enter' key. The argument '*-p*' will run the program in interactive mode.

First you will be prompted to enter the name of the *SIMI BioCell* project (i.e. the name of the SBC and SBD files) and the name of the XML files without file extensions.

```
Enter name of SIMI BioCell project (without file extensions .sbc or .sbd):
```

Enter name of h5.xml file (without file extension .xml):

The next prompt will be:

Did you crop the original image stack? (y/n):

If you cropped your image data, you should reply to this prompt with '***y***'. Then you will be asked to enter the crop distances:

Enter crop distance in pixels for x-coordinates (integer value; will be subtracted from left image border):

Enter crop distance in pixels for y-coordinates (integer value; will be subtracted from upper image border):

Enter crop distance in slices for z-coordinates (integer value; will be subtracted from lower border of image stack):

Use the distances which you noted during the cropping process earlier. If wrong values are entered, the spots of the cell lineage will later not be properly aligned to your images in *MaMuT*.

Mitosis reconstruction and test run

Next you will be asked to choose a mode for mitosis reconstruction:

Choose mode for reconstruction of mitoses:

0: no mitosis reconstruction

1: additional offspring spots at frame of mitosis

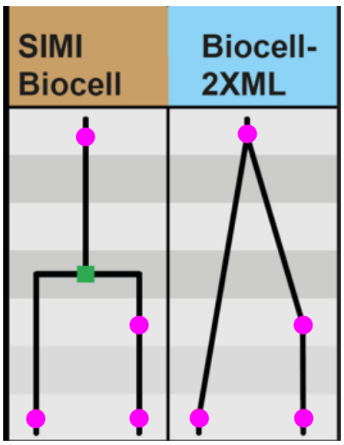
2: additional parent spot one frame before mitosis

3: additional offspring spots at frame of mitosis and additional parent spot one frame before mitosis

(0, 1, 2, or 3):

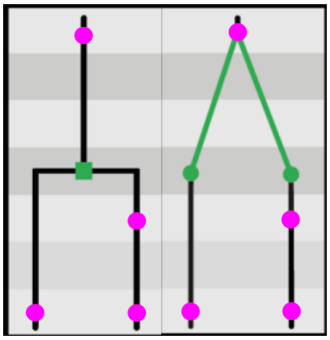
SIMI BioCell saves timepoints for mitoses without saving a corresponding spot. *MaMuT* saves only spots and edges. A direct conversion (Mode 0) will therefore enter splits in the lineage at the position of the last mother spot. This is also the default setting in *BioCell2XML*:

Mode 0 will create edges only between spots that are saved in the SBD file.

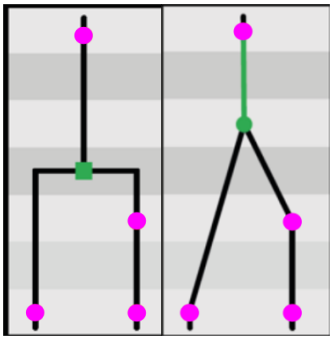


Depending on the specifics of your *SIMI BioCell* lineage and the tasks you want to use the converted lineage for, you may want to include the mitosis times that were entered in *SIMI BioCell* to the splits. *BioCell2XML* provides three different modes of mitosis reconstruction for this:

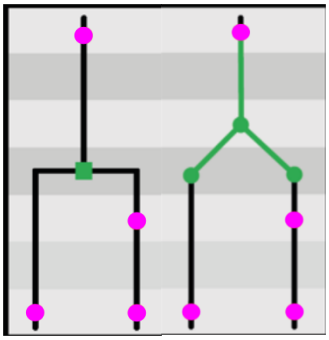
Mode 1 will calculate interpolated spots at the mitosis frame (using the last mother spot and the first daughter spots as references).



Mode 2 will calculate an interpolated mother spot one frame before the mitosis frame.



Mode 3 will give the closest appearance to the visualization of *SIMI BioCell.* Here, *BioCell2XML* will calculate a new mother and new daughter spots separated by only one timeframe.

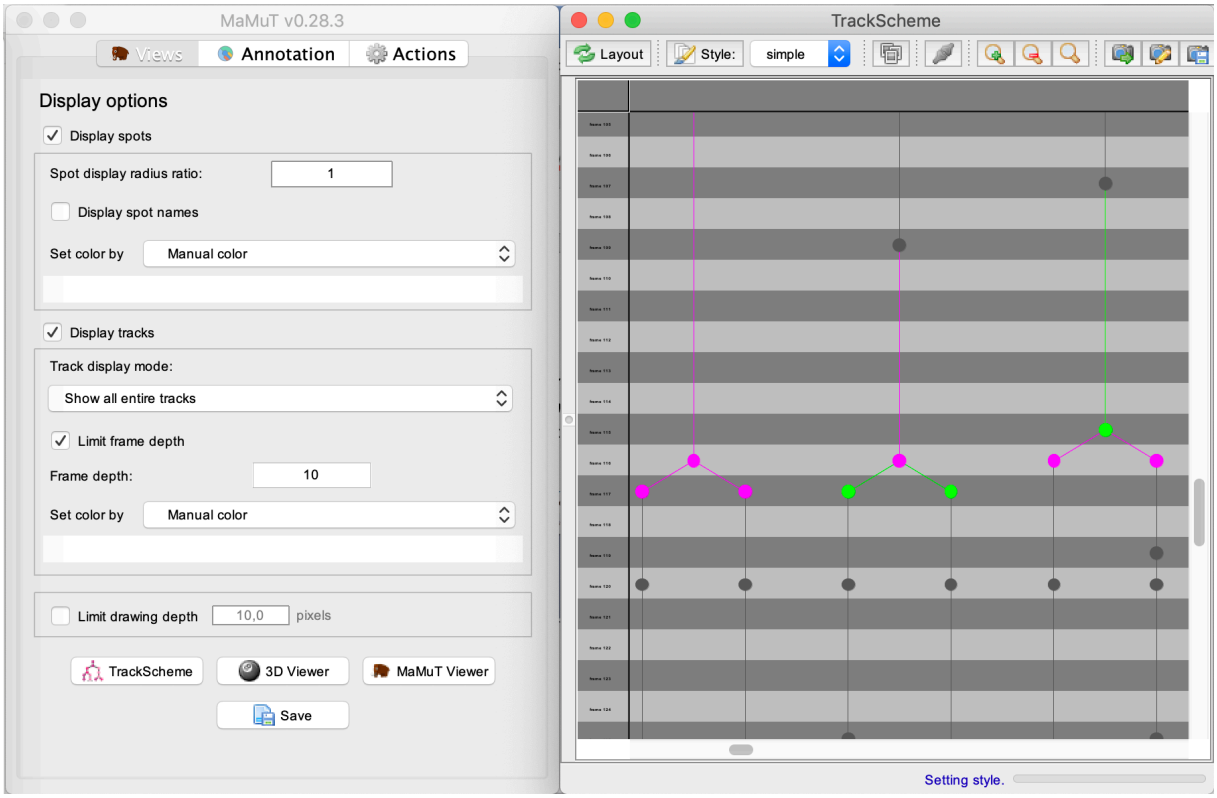


To make the decision for a reconstruction mode easier, you will be asked in the next prompt if you would like to perform a test run:

```
Perform test run (this option will highlight all mitoses spots)? (y/n):
```

This will result in a lineage using the desired mode of mitosis reconstruction but with a helpful color code. Type **'y'** to create the test lineage. The program will run and a file of the type '***lineage_BioCell2XML_test.xml***' will be created in your working directory.

Now open this file in *MaMuT*. From *Fiji*'s plugins menu select '*MaMuT/Open MaMuT Annotation'*. In the *MaMuT* window select the color setting '*Manual color'* in the '*Set color by*' drop-down menus for spots and tracks. Open '*Track Scheme*' and adjust zoom to view the lineage tree.



The color magenta marks mitosis spots (and upstream edges) that were placed in the original *SIMI BioCell* lineage and that coincide with the position of reconstructed mitosis spots. Spots that were newly introduced under the chosen mode of mitosis reconstruction are colored green, together with their upwards connecting edges. The remaining original spots translated from the SBD file are colored grey. The example above shows three color-coded mitoses in *MaMuT*'s track scheme: With the original mother spot one timeframe before the mitosis frame and original daughter spots at the mitosis frame (left), with the original mother spot position one frame before the mitosis frame and daughter spots positions recalculated for the mitosis frame (middle), with original daughter spots at the mitosis frame but the mother spot recalculated (right).

Repeat the test run with the different mitosis reconstruction modes if necessary.

Should your *SIMI BioCell* lineage contain mitoses with only one daughter cell containing spots, this split would normally be displayed as a continuous branch in *MaMuT*. The test run in combination with a mitosis reconstruction allows to identify such single-branch mitoses as, depending on the chosen reconstruction mode, the last mother spot and/or spots at the mitosis frame will be colored. In addition, a legend.txt file containing all the mitoses with only one daughter cell as well as alternative branches (if such are present in the respective *SIMI BioCell* lineage data) will be outputted.

Converting colors, sizes, and other metadata

To continue with regular conversion run the program as described above. Though, when asked if you want to perform a test run type **'n'**. You will be prompted for a few more entries.

Transferring spot metadata:

```
Translate SIMI BioCell spot metadata to XML file? (y/n):
```

Spot metadata is specifically spot size, color, shape, cell fate and user comments. Only size and color can be displayed directly in *MaMuT*. The remaining metadata (if present in the SBD file) will be transferred to the XML file and be displayed indirectly. Also, it can be accessed by other software.

Reply with **'y'** and you will get

```
Translate spot colors? (y/n):
```

This will translate color entries and write them to the *MaMuT* XML file.

The next prompt will be

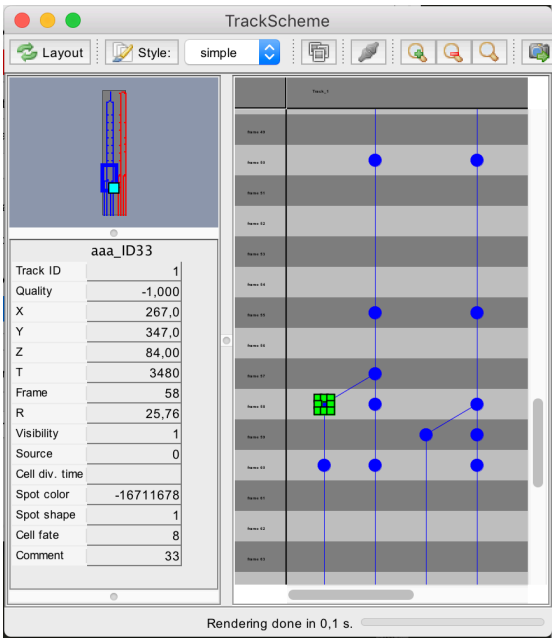
```
Translate spot sizes? (y/n):
```

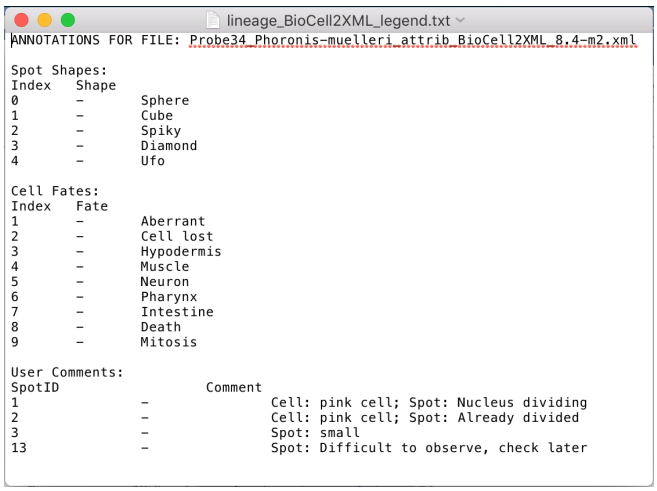
This will calculate spot sizes that represent the proportions seen in *SIMI BioCell*'s 3D View as close as possible when displayed in *MaMuT*'s 3D Viewer. Keep in mind that spot sizes will not be automatically transferred to new spots if you continue to add spots to the translated lineage in *MaMuT*. In this case we recommend to enter **'n'**.

The next prompt considers only the metadata that *MaMuT* cannot display:

```
Translate spot shapes, cell fates, and user comments? (y/n):
```

This will write the entries for shape (the shapes of the artificial spots in *SIMI BioCell*'s 3D View), cell fates and user comments to the *MaMuT* XML file. In addition, a text-file of the type "*filename_legend.txt*" will be created. The text window of *MaMuT*`s *TrackScheme* will show numerical values for these entries, as display of string entries will not work in the current version of *MaMuT*. The *legend-text* file, which can be opened with any text editor, lists the string entries for the numerical entries of spot shapes and cell fates. User specified comments are listed by Spot ID. This way, *SIMI BioCell* metadata can be reviewed using *MaMuT*.





Interpolation

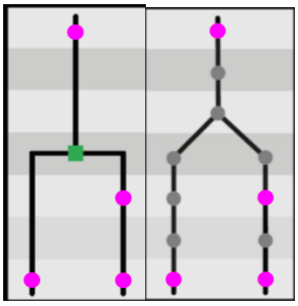
For some applications it may be necessary to calculate interpolated spots for empty frames in the lineage. Furthermore, *MaMuT*'s 3D Viewer does not interpolate spots for empty frames on the fly, as is the case in *SIMI BioCell*. Spots can only be displayed if they are also written into the XML file. Therefore, *BioCell2XML* will perform interpolation as desired.

> Interpolate between spots (i.e. adding mean coordinate spots between existing spots)? (y/n):

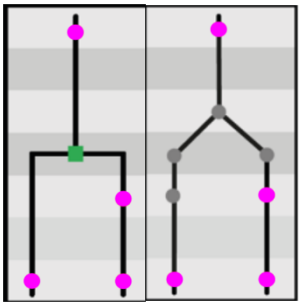
Select **'y'** and you will be prompted for the desired interpolation settings:

> Enter fraction (i.e. distance between spots) for interpolation (integer value):

The fraction is the distance between interpolated spots. Entering **'1'** will result in an interpolated spot for every empty frame (between spots) in the lineage.

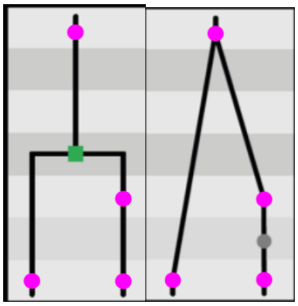


For large lineages and limited RAM it may be convenient to increase the distance between frames for interpolation. If selected, the interpolated spots will be calculated in a way that spots will appear synchronously at least in every frame of the fraction. E.g. enter '**2**' and the lineage will have spots at least at every second frame.



Next you can choose to exclude mitoses from the interpolation process (e.g. if you want to edit the mitoses manually later).

> Omit mitoses in interpolation? (y/n):



Saving the results

Finally, you will be asked to enter a filename for the new *MaMuT* XML file that will be created. Entering a name here is optional, per default the output file bears a name of the type '*lineage_BioCell2XML.xml*'.

```
Enter name of output file (optional; without file extension .xml):
```

The program will run and reply with:

```
Transformation completed:
File 'lineage_BioCell2XML.xml' was written to 'D:/my/working/directory'
```

And, if an additional legend file is written, with:

```
File 'lineage_BioCell2XML_legend.txt' was written to 'D:/my/working/directory'
```

Keep in mind, that if you use the same output filename consistently, the previous files will be overwritten without any warning.

Using the program non-interactively

It is possible to enter the input file names and all conversion specifications as optional arguments in the command line directly. In this mode, it is also possible to import the SBC, SBD, and H5/XML files from a different directory than the one where ***Biocell2XML.py*** is located. This may be helpful, if one wants to convert a high number of *SIMI BioCell* projects (since, in the interpreter window, one can use the up/down keys to get to previously typed command lines, and then simply change file names, file paths, or conversion arguments).

To run the program under this mode (after setting the working directory) enter

```
python3 BioCell2XML.py -simi nameofbiocellproject -h5xml nameofh5xmlfile
```

The argument '*-simi*' is followed by the name of the *SIMI BioCell* project that shall be converted (the name of the SBC or SBD name without file extension). The argument '*-h5xml*' is followed by the name of the H5/XML file (without extension).

This entry will run *BioCell2XML* with default settings. The specific settings can be selected by adding the optional arguments given in the table below in any order.

Argument Name		**Argument Description**
Abbreviation	Long name	
-h	--help	**show this help message and exit**
-p	--prompt	**activate prompt input mode** [interactive mode]
-simi		**name of SIMI BioCell project (without file extensions .sbc or .sbd)** [required argument if prompt input mode is not used]
-h5xml		**name of h5.xml file (without file extension .xml**) [required argument if prompt input mode is not used]
-simipath		**path of SIMI BioCell project files** [if other than current working directory]
-h5xmlpath		**path of h5.xml file** [if other than current working directory]
-x	--crop_x	**crop distance in pixels for x-coordinates** [value will be subtracted from left image border] For example, the argument -x50 (or --crop_x50, respectively) will crop 50 pixels from the left image border
-y	--crop_y	**crop distance in pixels for y-coordinates** [value will be subtracted from upper image border]

-z	--crop_z	**crop distance in slices for z-coordinates** [value will be subtracted from lower border of image stack]
-m	--mitoses	**mode for reconstruction of mitoses (choices: 0, 1, 2, 3)**
-m0	--mitoses0	Mode 0 (default): no mitosis reconstruction
-m1	--mitoses1	Mode 1: offspring spots are added at frame of mitosis
-m2	--mitoses2	Mode 2: parent spot is added one frame before mitosis
-m3	--mitoses3	Mode 3: offspring spots are added at frame of mitosis and parent spot is added one frame before mitosis
-t	--test	**perform test run** [modifications at the mitoses will be highlighted]
-c	--color	**deactivate translation of spot colors**
-s	--size	**deactivate translation of spot sizes**
-d	--metadata	**deactivate translation of spot shapes, cell fates, and user comments**
-i	--interpolate	**interpolate between spots** [place new spots on empty frames by linear interpolation between given spots]
-f	--fraction	**fraction for interpolation** [frame distance between interpolated spots; this reduces file size in very large lineages] For example, the arguments -i -f2 (or --interpolate --fraction2, respectively) will add interpolated spots on every second frame in the lineage
-o	--omitmitoses	**omit mitoses in interpolation** [only branches between splits are interpolated]
-out		**name of output file (without file extension .xml)** [optional]