

MorphoGraphX User Manual

www.MorphoGraphX.org

This manual is written in a tutorial style and gives step by step instructions for image processing and analysis with MorphoGraphX. Example confocal stacks used in this manual can be downloaded from the data page on the MorphoGraphX website www.MorphoGraphX.org.

Even if you are only interested in doing full 3D segmentation, we recommend to go through the entire tutorial since the procedures are very similar for curved surface (2.5D) and 3D. Important differences when analyzing volumetric cells are pointed out in the section for 3D segmentation and 3D mesh analysis.

Table of Contents

1. Introduction to MorphoGraphX.....	5
1.1. Definitions.....	5
1.2. Data analysis guidelines.....	6
1.3. Sample image data sets.....	6
2. Confocal image guidelines.....	7
3. The user interface.....	9
4. Loading samples (stacks or meshes).....	12
4.1. Getting started.....	12
4.2. Troubleshooting.....	14
5. Global shape extraction.....	15
5.1. Creating a mesh of the global shape.....	15
5.2. Troubleshooting.....	17
6. Experiment with processes.....	18
6.1. Processes in “Stack/Filters”.....	18
6.2. Processes in “Stack/Morphology”.....	19
6.3. Processes in “Mesh/Signal”.....	19
7. 2.5D images.....	20
8. Segmenting cells.....	22

8.1. Manual seeding and segmentation.....	22
8.2. Automatic seeding and segmentation.....	23
Troubleshooting.....	25
9. Mesh subdivision.....	26
10. Cell Geometry quantification.....	28
11. Quantifying Fluorescence.....	32
12. Measure processes, attribute maps & data export.....	34
12.1. Measure processes.....	34
12.2. Attribute maps.....	34
12.3. Data export.....	35
13. Lineage tracking.....	36
14. Comparing data from two time points.....	38
15. Principal directions of growth (PDGs).....	40
16. Custom Axis Directions.....	47
16.1. Custom directions with Bezier.....	48
16.2. Custom directions using a distance heat map.....	48
17. Mesh Deformation and Growth Animation.....	50
17.1. Creation of Deformation Functions.....	51
17.2. Semi-Automatic Parent Labeling.....	53
17.3. Mesh Mapping or Deformation.....	54
17.4. Morphing Animations.....	55
17.5. Advanced Features for Morphing Animations.....	56
17.6. Growth Analysis using Deformation Functions.....	57
18. Quantifying signal orientation.....	60
19. Improving the stack signal using deep learning.....	62
20. 3D segmentation.....	63
21. Mesh 3D analysis and quantification.....	67
21.1. Heat maps & Measures 3D.....	67
21.2. Cell Lineage Tracking 3D.....	69
21.3. Change Maps 3D.....	70
21.4. PDGs 3D.....	70
21.5. 3D Visualization Options.....	71

21.6. Custom Directions for 3D meshes.....	71
21.7. Signal orientation for 3D meshes.....	72
22. Cell shape analysis using Principal Component Analysis (PCA).....	73
22.2. For 3D meshes.....	73
23. Organ-centric coordinate systems.....	74
23.1. The Cell Distance Measure.....	74
23.2. Bezier Line and Grid.....	74
23.3. Further types of organ coordinates.....	75
24. Cell atlas and cell type classification.....	75
24.1. Cell Atlas Root.....	75
24.2. Cell Atlas Meristem.....	76
24.3. CoordX.....	76
24.4. Cell Type Classification using a single heat map.....	76
24.5. 3D Cell Layer Classification using a surface mesh.....	76
24.6. Cell Type Classification using two measures.....	77
24.7. Cell Type Classification using SVMs.....	78
25. Cell Division Analysis.....	81
25.1 Division Analysis 2.5D.....	81
25.2. Division Analysis 3D.....	84
26. Quantifications of the raw stack signal without mesh.....	85
27. Animation.....	86
28. Data Export.....	87
29. Creating workflows (tasks).....	88
30. Installation.....	91
30.1. Computer Requirements.....	91
30.2. Installation.....	91
30.3. After installation – enabling the ALT key.....	91
31. Writing custom processes (plugins).....	93
32. Compiling MorphoGraphX from source.....	95
32.1. Linux.....	95
32.2. Windows.....	95
32.3. Mac.....	95

33. Command line.....	96
34. Measure tables.....	97
References.....	100
Sample Data.....	102

1. Introduction to MorphoGraphX

MorphoGraphX is an open source platform for the visualization and processing of 3D image data. MorphoGraphX can manipulate 3D image stacks and use them to extract the global shape of organs as a triangular mesh which can be interpreted as curved surface images (2.5D). Such curved images capture enough of the 3D geometry to give an accurate representation of cell size and geometry while retaining much of the simplicity of 2D images. Additionally, MorphoGraphX is also able to create and analyze volumetric segmentations in full 3D.

1.1. Definitions


The following are some definitions of the main objects used in MorphoGraphX:

Stack. A stack corresponds to a 3D image of a biological sample (e.g., a confocal laser scanning microscopy (CLSM) image). There are 2 stacks in MorphoGraphX, Stack 1 and Stack 2, to allow the comparison of different samples from different time points. The size of a stack is defined in voxels, which are the equivalent of pixels for 3D images. The stack whose sub-tab is selected under the “Main” tab is the active stack (Figure 1). The non-active stack is called the other stack. Most processes operate only on the active stack. A stack contains 2 copies of the image data called stores and 1 associated mesh which are described below. All objects in the stack share the same reference system and cannot be translated or rotated independently. Different stacks, however, can be moved independently.

Store. There are 2 stores in each stack, corresponding for example to the original 3D image and the same data after image processing. Usually, the original sample is loaded into the Main Store. Once the original stack has been modified by a process, the result (e.g., the 3D image after blurring) is stored separately in the Work Store, without over-writing the Main Store. This allows to go back to the original data to try different parameters or processes, as there is no “undo” function in MorphoGraphX. Further processing will then by default (when the “work” store checkbox is checked) use the Work Store as the source and overwrite it with the results. Each store has a separate color, brightness, transparency, etc.

Mesh. A mesh is a triangulated curved surface. Typically, the mesh is extracted from a stack after some image processing operations are performed on it. Its vertices contain a label which can be used to segment the mesh into cells. The mesh can store additional data in so-called attribute maps, such as: parent labels, heat maps, principal directions of growth (PDGs) or other arbitrary data structures, which are created by processes (see the section “Attribute Maps” for more details).

Process. Most operations in MorphoGraphX are performed by processes including many of the menu options such as saving a stack or mesh. They can be found in the “Process” tab where they are arranged in a hierarchical folder view: first grouped into sub-tabs for stack, mesh, and miscellaneous processes, and then in folders below. When selecting a process, its available parameters appear in the box below. Changed parameter values can be saved in the project (.mgxv) file. To run a process, double click it, or highlight it and press the run

arrow  at the top right. If a process fails, an error dialog box appears, specifying the problem, such as invalid parameters. Processes can be organized into custom pipelines called tasks (see section “Tasks” for more details).

1.2. Data analysis guidelines

Before starting to work with MorphoGraphX it is helpful to have a general idea about the goal of the image/data analysis. Most quantification pipelines require the generation of a segmented mesh. MorphoGraphX has two mesh types: surface (2.5D) meshes or volumetric (3D) meshes. The creation and error-correction of such meshes, which often contain hundreds of cells, can be time consuming and greatly depends on the quality of the underlying stack data. Generally, 3D segmentations require higher quality data and can be more cumbersome to correct than 2.5D segmentations.

In some (usually rarer) cases it might be sufficient to only generate a surface mesh (e.g. for the quantification of the general curvature, see also Figure 6) or to directly work on the stack (e.g. for some types of the quantification of the stack signal).

1.3. Sample image data sets

Visit the website www.MorphoGraphX.org to download the sample data sets:

Kierzkowski_2012: 2 time points of confocal images of a tomato meristem with cell wall (PI) and auxin signaling (DR5) markers. This data set is best used for organ surface extraction and surface mesh segmentation. It already contains meshes and cell lineages. In this user guide we use this data set in most of the sections related to 2.5D images.

Bassel_2014: Root and hypocotyl of a mature embryo of *Arabidopsis thaliana*. This data is from a fixed and cleared sample and is suitable for segmentation in 3D and will be used for sections related to the segmentation and processing of volumetric cells.

Hervieux_2016: Time lapse confocal images including 8 time points of a sepal of *Arabidopsis thaliana*. This data is used for some more advanced features in the later sections such as: creating a morphing animation or bulk data export.

Strauss_2021: Multiple data sets of different plant organs (mainly *Arabidopsis thaliana*) that were used as sample data sets in the MorphoGraphX 2.0 paper (Strauss et al. 2021).

2. Confocal image guidelines

MorphoGraphX has mostly been used to process 3D confocal image stacks. Data collection for use in MorphoGraphX can be a little different than for other uses. In most cases, people optimize data collection so that individual slices look good to the human eye. This is not always the best for 3D data analysis and visualization. The following are some tips to help get the most out of the images:

16 bit images. If possible acquire images with 16 bits per voxel. This is the native image format for MorphoGraphX and any image loaded into the software will be converted into that format. Compared to 8bit, 16bit images have higher dynamic range, and it will be easier to extract features in darker areas of the image. If 16-bit collection is not available on the microscope, use 12 bits, or the highest available setting.

Cubic voxels. The confocal typically has a higher resolution in the X and Y dimensions than it does in Z. To partially compensate for this, it is important to use a small Z step. Try to make the voxels close to cubic for best results. For example, if the XY resolution is 0.5 μm , then use 0.5 or 1 μm for the z step. An even smaller Z step like 0.25 μm can further improve the segmentation but comes at the cost of imaging time, light exposure and photobleaching of the sample.

Averaging. It is better to add real planes (by reducing the Z step) than using line or frame averaging.

Image saturation. It is best to have the images just below the point where they are saturated. The controlling software of confocal microscopes usually has a mode to show saturated pixels. Turn the gain up until you just start to see some saturated pixels. For wall stains where you do not need to quantify the fluorescence, it can help to over-saturate in order to get a good contrast between the inside and the outline of the cell. To compare fluorescence between samples, it is necessary to not saturate the channels you want to quantify.

Wall Marker. To get a nice segmentation, you need a method to stain the cell walls or the plasma membrane. This can be done with propidium iodide (PI) or a marker line. With the correct detection range, PI can be reliably separated from both GFP and YFP using the same excitation wavelength. Since most confocal microscopes can capture different light frequencies to different channels simultaneously, you can collect PI wall stain signal, and GFP fluorescence marker signal in the same amount of time/exposure. MorphoGraphX will use the PI wall stain for surface extraction and segmentation, and then the GFP channel for gene expression analysis. You will have to optimize your imaging a bit to get the best results for your samples.

Start in the right place. Be sure to start data collection a few microns above the sample. The collection too close to the cells surface will result in a flat spot at the top of the sample. Be careful not to miss the bottoms of cells if you are doing 3D segmentation, as you can only quantify cells which are complete.

Time-lapse imaging. If you want to quantify growth or changes in signal intensity over time, the same sample may have to be imaged multiple times, up to several days. Repeated exposure to a light source (e.g., confocal laser) results in considerable stress for biological samples and might interfere with the results. The time intervals between image acquisitions should be long enough to allow the sample to recover. On the other hand, the sample should not grow too much in between images, otherwise it can become too difficult to identify the cells between successive time points. As a rule of thumbs, the cells should not divide more than twice during the time intervals.

Use Fiji to get TIF files. LOCI's Bio-formats tools (<http://openmicroscopy.org/info/bio-formats>) can read the output files of most popular microscope and have made plugins for ImageJ. We recommend Fiji (<https://imagej.net/Fiji>), which is a distribution of ImageJ that contains these, along with many other useful plugins. After opening a stack in Fiji, split it into separate windows for each channel, as MorphoGraphX currently requires there be only one channel per file. From Fiji, save each channel as a multi-slice TIF file.

If voxel sizes are wrong. The TIF format has no standard way to specify the voxel size in Z. ImageJ writes this in the TIF tags, and MorphoGraphX is can read this. If this doesn't work for some reason then MorphoGraphX allows you to change the voxel size after you load the image (see section 4 "Loading Samples"). If you need to do this, you can get the voxel (pixel) sizes using the microscope software or from the metadata (text file) in ImageJ.

3. The user interface

Start a MorphoGraphX session with the example data loaded by clicking on the “example_session.mg xv” in the Kierzkowski_2012 dataset.

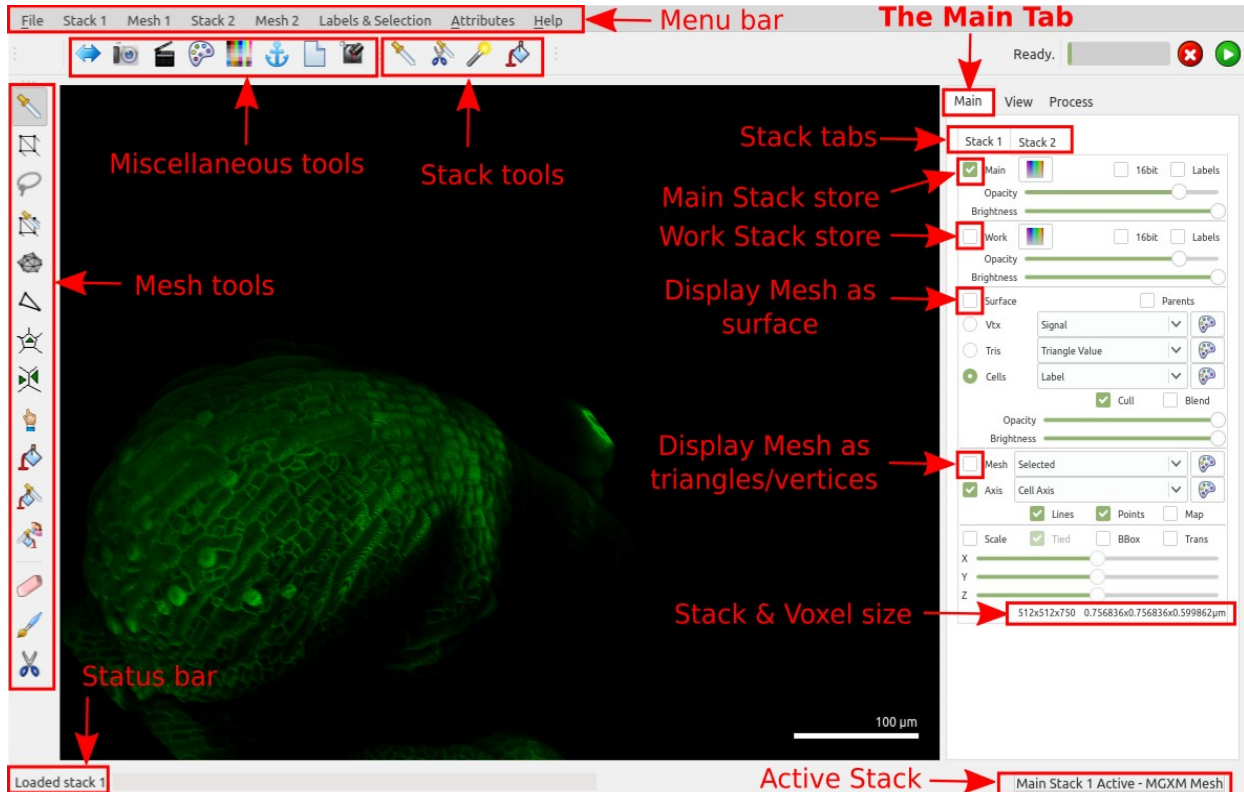


Figure 1. Main menu, toolbars and the “Main” tab.

Tools from the toolbars will work only if the “Alt”-key is pressed on. Some of the tools are enabled under specific conditions, which will be described in the information bar if you try to use them. You can drag and drop the toolbars around to change their position in the MorphoGraphX window.

Since two different stacks (Stack 1 and Stack 2) and associated meshes (Mesh 1 and Mesh 2) can be displayed simultaneously, you have to specify which one is being processed (Active stack). Select the active stack and mesh in “Stack tabs” by switching between the Stack 1 and Stack 2.

The input for stack operations (3D filters and the like) is chosen by checking either the Main or Work Store (if both are checked, the Work Store is being used). The output (modified image) is always stored in the Work Store.

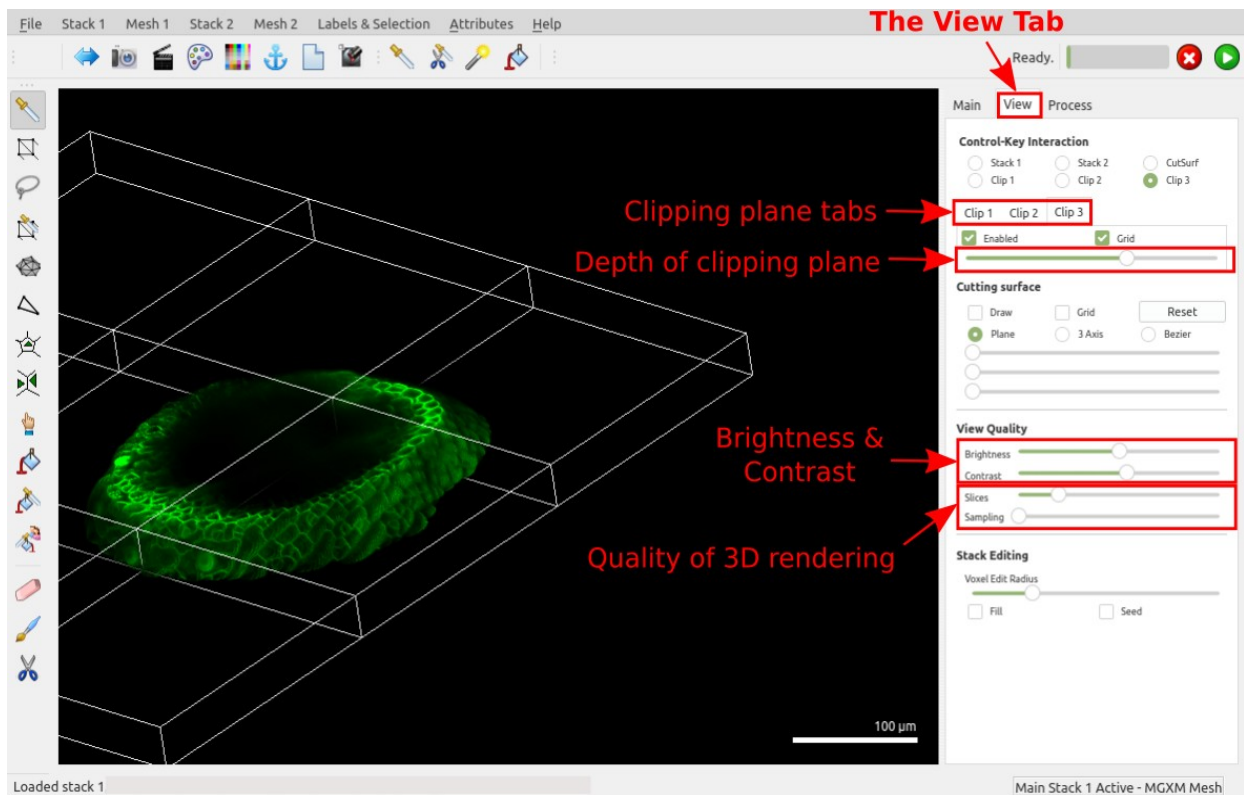


Figure 2. The “View” tab with an enabled clipping plane.

Global brightness and contrast can be tuned in the View tab, as opposed to the brightness and opacity of individual stacks and meshes that are controlled from the “Main” tab. Depending on the speed of your machine, 3D rendering can run fast or slow. You can alter the quality of 3D images display using the Slices and sampling to make 3D rendering faster when the stacks are rotated.

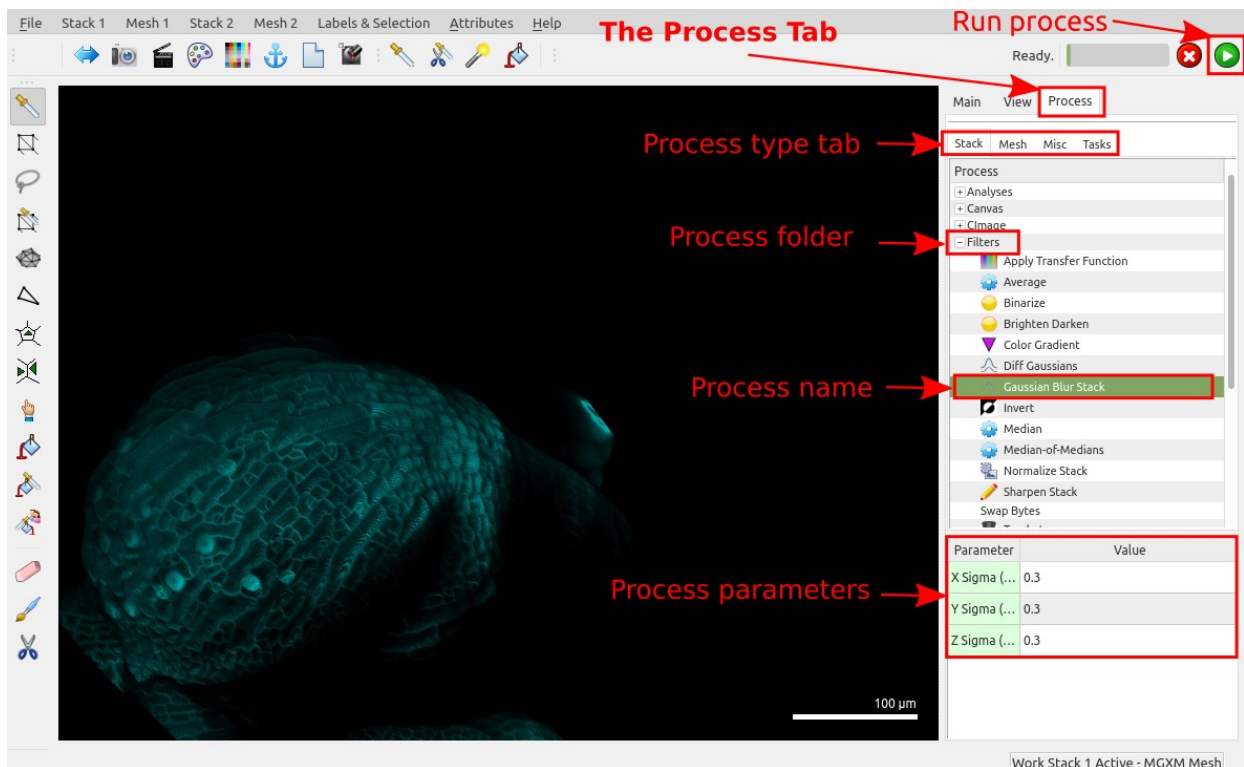


Figure 3. The “Process” tab with “Stack”, “Mesh”, “Misc”, “Tasks” process types. The processes are organized in folders. The box below the processes shows the different parameters available and their set value.

Processes are bundled in groups, that you can unfold by clicking on the “+” icon next to the group name. To run a process, press the “Run” button or double click on the process name. While running a process, make sure that the right stack number (1 or 2) and store (“Main” or “Work”) are active.

4. Loading samples (stacks or meshes)

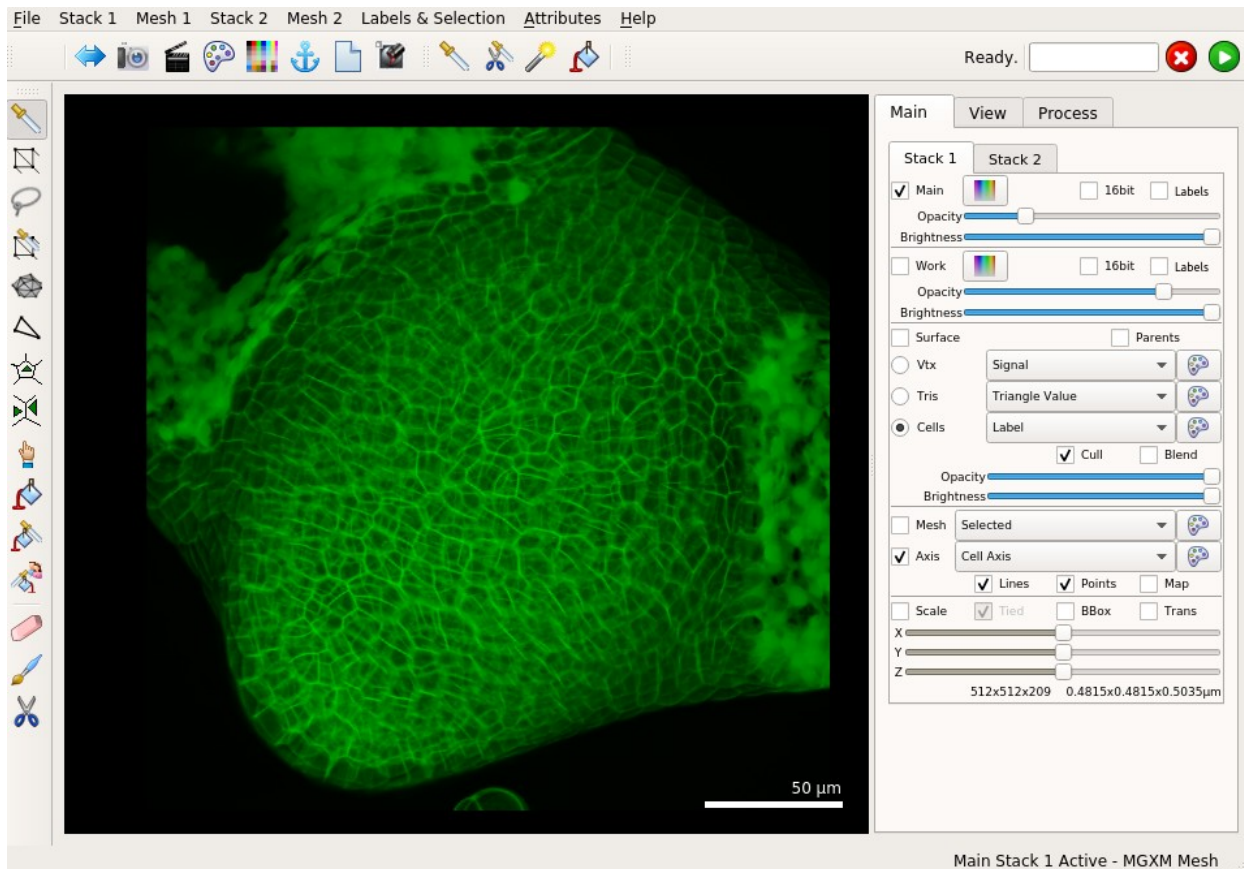



Figure 4. A loaded confocal stack in the main window of a tomato shoot apex from the Kierzkowski_2012 data set.

4.1. Getting started

This section covers how to start MorphoGraphX, load data, and experiment with the visualization tools. Proceed as follows:

Start MorphoGraphX. Double-click on the green MGX icon  on your desktop or open an existing session by double-clicking on a “.mgxv” file. You can also create an empty “.mgxv” file and open it. After MorphoGraphX starts, you will see the main window and a text window (called terminal) underneath. Do not close the terminal, if you do, MorphoGraphX will shut down. Informational messages from processing operations or errors will be displayed in the terminal. Please report crashes and any associated messages to us and we will try to fix them.


Load a stack. Drag and drop a TIF image stack onto the main window. This will load your sample into the Main Store of Stack 1. Drag and drop the file with the Alt-key pressed to


load the sample into Stack 2. You can also load samples with the Stack menu (for example, under “Stack 1/Main/Open”).


Importing file series. It is also possible to import image series with the “Import series” option in Stack menu. In this case, you will need to specify the Z-step yourself.

Loading a pre-existing mesh. Meshes are usually extracted from stacks, but they can also be loaded separately from the stacks. Load meshes the same way as stacks, either by dragging and dropping the files into the main MorphoGraphX window, or by using the Mesh menu (e.g., “Mesh 1/Load”).

Experiment with the visualization tools. MorphoGraphX uses QGLViewer to handle the display interaction. The left mouse button allows you to rotate the sample, the right button (or left button + Shift-key) translates, and the wheel is used to zoom. Mouse and keyboard codes are listed under “About QGLViewer” in the Help menu. There are many visualization tools in MorphoGraphX. It is recommended to load two different samples in Stack 1 and Stack 2 and take some time to experiment with:


Rotation-translation. *Left click* on the display area and moving the mouse rotates both stacks (that is, changes the camera view angle), while the *right click* translates the point of view. You can check the camera movements relative to the coordinate system (X, Y, Z) by pressing the A-key. Double left-click to rotate the camera to the nearest view aligned with one of the axes (X, Y or Z). To move each stack independently, specify which stack should be displayed using the check boxes in “Control-Key interaction” within the View tab and press the Control key while rotating or translating the stack. Reset all the rotations and translations by clicking on the “Reset view” icon  from the Miscellaneous toolbar. Press the A-key again to hide the coordinate system.

Transfer functions. The transfer function determines how the 16-bit voxel values (from 0 to 65535) get mapped to colors. In the Main Tab, click on the “Edit labels” icon  to display the color scale, or transfer function, of each stack. A pop-up window will appear. Use the “Predefined color maps” menu to change the type of transfer function. Double-click on the histogram to add a marker (black triangle) and displace the markers to modify the color map. Right-click on the histogram to reverse the transfer function.

Clipping planes. In the “View” tab, select one of the clipping planes in the Clip 1 (X plane), Clip 2 (Y plane) and Clip 3 (Z plane) tabs. To display the boundaries of the clipping plane, check the “Grid” checkbox. Modify the depth of the clipping plane with the slider. To display only the part of the stack inside the clipping plane, check “Enabled” on. You can move the clipping planes around the stack by holding the Ctrl-key. Note that the correct radio button for the Control-Key-Interaction must be set in the View tab. Reset the clipping planes rotation with . To get a cross section of the sample, align the camera angle with one of the axes (e.g., Z) and enable the clipping plane perpendicular to that axis (e.g., Clip 3) to obtain a cross section. Select the Control-key interaction for this plane. Slice throughout the sample using the middle mouse button while pressing

the Ctrl-key. You can also rotate and move the clipping plane by using the Ctrl-key with the left or right mouse button. You can visualize more than one clipping plane at the same time.

Slices. Under “View Quality” in the “View” tab, use the “Slices” slider to modify the number of stack slices displayed. Try to rotate the stack while the slice number is either minimal or maximal and notice if it makes a difference in the rendering speed.


Save the MorphoGraphX session. To save the session with all the parameters used (such as stack rotations, clipping planes, custom colors and process parameters) and the path of the stack and mesh data loaded, in the file menu, select “save” or “save as”. This will create a MorphoGraphX session (.mgxv) file that you can use to start MorphoGraphX and continue where you left off. The data (stacks and meshes) are not saved in the .mgxv file, but will then be automatically loaded from the folder saved in the project file. Parameters saved within the project file can also be accessed via “Edit parameters” tool .

4.2. Troubleshooting

Symptom: “The stack is loaded, but not visible”. Try the following solutions:

The store into which you loaded your data might not be active. Check the 'main' and 'work' checkbox in Main tab.

The stack or mesh can be out of the field of view. Use  to reset the view.

The stack color depth might be 8 or 12 bits instead of 16 bits. By default, the stacks are supposed to be 16 bits and the color-mapping is optimized for this color depth. Check the transfer function by clicking on  in the Main Store of the stack you just loaded. In the “Transfer function editor” window, click on “Auto Adjust”. If the data is 8 or 12 bits, only part of the histogram will be filled (1/16 or the full range in case of 12 bits data, 1/256 in case of 8 bits). Although it is possible to run all the stack processes on non-16 bits images, we recommend you to re-scale the data to 16 bits for processing. Use the process “Stack/Filters/Brighten Darken” with an amount of 16 to convert 12 bits images, or 256 for 8 bits images. Next run the process “Stack/Multi-stack/Copy Work to Main Stack”. Reset the transfer function of the Main Stack in the “Transfer function editor”. Now the stack should look fine and you can use it for further processing. Save it using the Stack menu (you can over-write the original stack or save separately the 16 bits version).

The stack voxel size might be wrong. In the Main tab, check the voxel size (see section 2). If it does not correspond to the pixel size and to the z-steps used during acquisition, resize the voxels using the process “Stack/Canvas/Change Voxel Size”. If the size is already correct in some of the directions (X, Y or Z), enter a value of 0 for these and their size will not be changed.

5. Global shape extraction

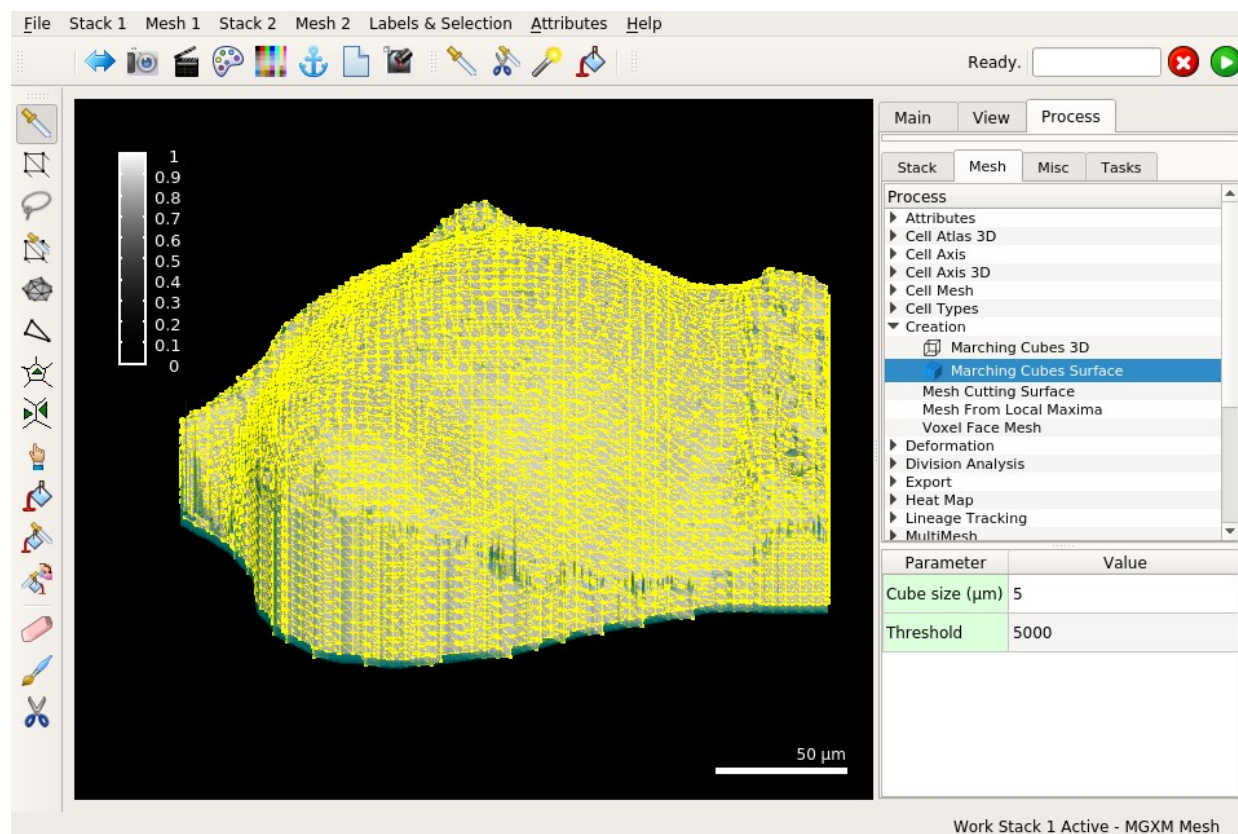


Figure 5. Surface mesh created from the confocal stack.


5.1. Creating a mesh of the global shape

The first step in the segmentation process is to extract the global shape of your sample. We will demonstrate this process with the Kierzkowski_2012 data set of a tomato shoot meristem. This organ has a high curvature, and a plain 2D segmentation would result in significant distortion of the cells. To process this stack we will extract a 2.5D curved mesh as follows:



Load the stack. Make sure that the stack with PI staining is loaded into Stack 1 Main and Main Stack 1 is active (see also section 4).

Blur the stack. Run the process “Stack/Filters/Gaussian Blur Stack” with values of 0.3, or “Stack/Filters/Average”. Note that the color changes from green to cyan as the resulting blurred image is stored in the Work Stack (with it’s default color cyan) which is checked automatically.

Edge detect. Use the process “Stack/Morphology/Edge detect”. This will create a solid structure representing the global shape of your object. If the threshold is too high holes might appear which can be closed by lowering the threshold or by using the process “Stack/Morphology/Fill Holes”. Tip: for optimal visualization of this solid shape, turn the

opacity of the Work Stack to the maximum. At this point it is also possible to use the “Voxel Edit” tool  with the Alt-key to erase parts of the stack that you do not need. Note that this operation only works on the Work Stack. Clipping planes can be used to restrict the “Voxel Edit” to a small slice of the image and to assess how well the filled stack matches the surface of the original image.

Extract the surface. Run the process “Mesh/Creation/Marching Cubes Surface”, with a cube size of 5 μm . As a general rule, the cube size should be roughly as large as the cells if you want to extract only the general organ shape, and several times smaller if you want to capture the curvature of individual cells.


Trim off the bottom. Later, the mesh will be subdivided and the signal projected to make a 2.5D image. To reduce the number of vertices in the final mesh, it is advisable to remove areas that do not represent the surface or are outside the region of interest, such as the bottom. In the Main tab, ensure that the “Mesh” checkbox, the “Lines” checkbox and the “Points” checkbox are selected, and “View” option is set to “All”. This will enable the visualization of all vertices of the mesh. Click the “Select points in mesh” tool  on the left and hold the Alt-key to select the bottom vertices of the apex. They should turn red. Hit the delete key to remove them. To make this easier, it is nice to have the apex in a horizontal position. You can do this by left-double clicking on it. Try to delete the bottom cleanly. Tip: A common problem at this stage are small fragments that are not connected to the main part of the mesh or holes in the mesh. Fragments can be found using the “Select Connected Area” mesh tool  on the main mesh and then invert the selection using “Mesh/Selection/Invert Selection”. Holes can be eliminated by selecting all vertices at the edge of the hole and using the process “Mesh/Structure/Merge Vertices”.

Smooth the mesh. Run the process “Mesh/Structure/Smooth Mesh” several times. You should now have a mesh following the global shape of the sample. Note that too many smoothing steps will cause a shrinkage of the mesh. This can be reversed by using the process “Mesh/Structure/Shrink Mesh” using the negative parameter value. Tip: Smoothing can also be applied on selected vertices only.

Verify the result. Check how accurate the mesh is with the clipping planes. Be sure to turn off the Work Store, and turn the Main Store back on when you do this. As the clipping planes are made thinner, the image will get darker, you can increase the opacity to counteract this. You want your mesh to resemble the shape of the original data. If the mesh is far away from the surface (does not align tightly with the shape of your stack), increase the threshold for the edge detect. For the sample data try 25,000.

Save your work. There is no Undo operation in MorphoGraphX, so it is important to save your work frequently. In the Mesh menu, choose Mesh 1 and Save. Give your mesh a meaningful name (e.g. “Coarse Mesh T0”).

5.2. Troubleshooting

Symptom: “The Edge Detect results in a strange shape”. If the stack is loaded upside-down, the Edge Detect will extract the shape of the sample’s bottom, which will often be quite rough since the signal on the bottom is not so sharp. Check if the stack is loaded in the correct orientation by pressing the “A” key and . This will show you the orientation of the X, Y and Z axis. If the arrow of the Z axis is pointing downward, run the process “Stack/Canvas/Reverse Axes”, with parameters: (X “No”, Y “No”, Z “Yes”). You can then over-write your stack for further use. Press the “A” key again to make the axis arrows disappear.

Symptom: “The mesh is full of holes”. Try first to run the whole procedure again with a lower threshold for the Edge Detect. If this does not help, use the process “Stack/Morphology/Fill Holes” just after running the Edge Detect. Also a higher amount of blurring the stack signal can help to counteract this problem.

Symptom: “There are spikes on the sample shape after Edge Detect”. The threshold for edge detection is probably too low. Alternatively you can try to increase the blur radius.

6. Experiment with processes

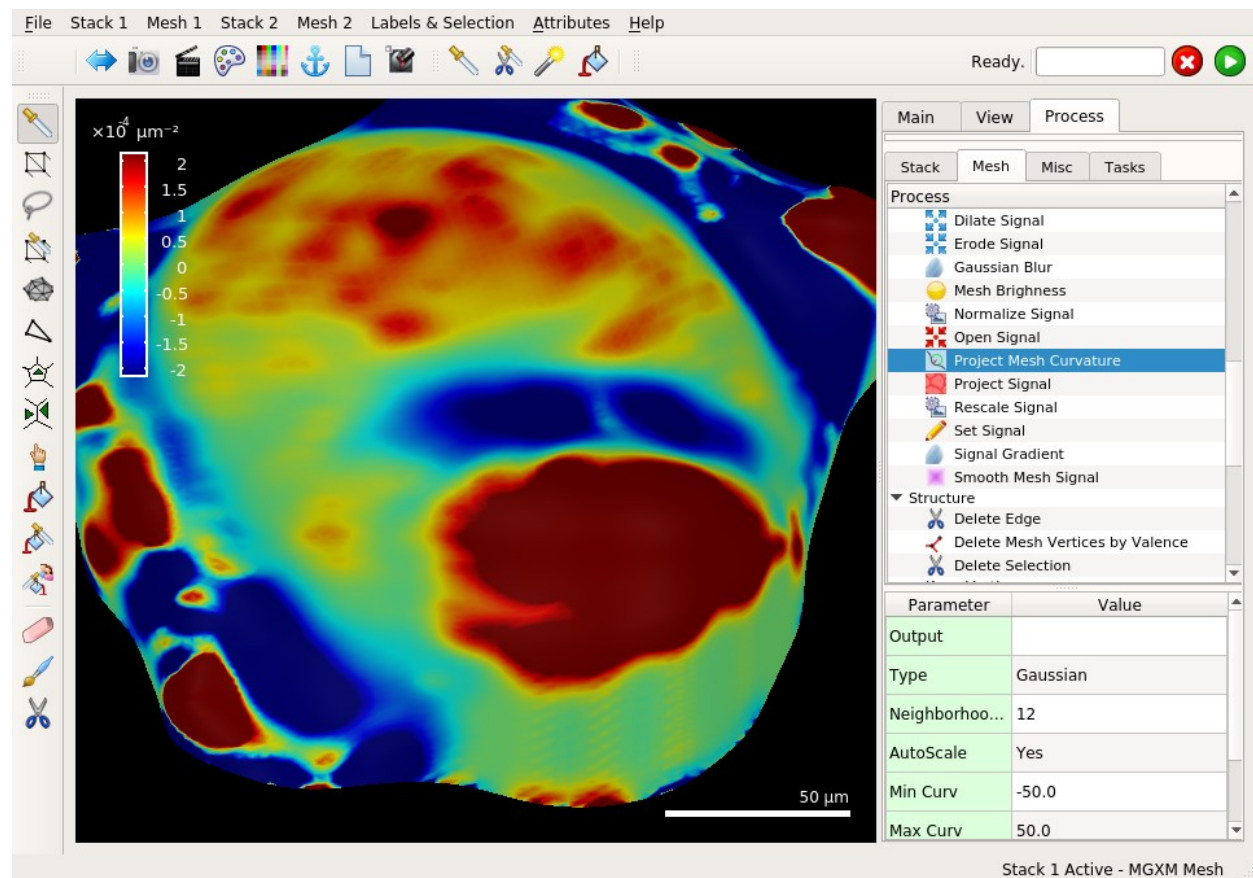


Figure 6. Surface mesh colored by amount of local curvature.

Each data set is different, and require different parameters and methods for processing. We recommend to experiment with processes, in order to adapt them to your data. The parameters used here work well with the sample data, but most likely will need to be optimized for your data.

Note: check which stack is currently active, so that you know if the process runs on the original data (Main Store) or on an image already modified by the last process (Work Store).

6.1. Processes in “Stack/Filters”

There is a wide variety of different filter processes in “Stack/Filters”. Here we introduce only a few, for details check out the tool tips of the processes and their parameter. They can also be accessed in the “Help → Process Docs” menu.

Gaussian Blur Stack. Try different values for the Sigma parameter (*i.e.*, the radius of Gaussian blur). Compare the result (in Work Store) and the original data (in Main Store)

using the clipping planes to get only a cross section. Larger radius should be more efficient at reducing the image noise, but also results in a more blurry image.

Normalize Stack. This will increase the contrast in the image, enhancing both the speckle noise and low signal areas. As above, use the clipping planes to observe the effect of normalization on the stack. You can perform a Gaussian blur before normalization to get the best results.

Invert. Transforms low signal into high signal and vice-versa. This can be useful if you want to segment features which are dark in the original image.

6.2. Processes in “Stack/Morphology”

It is good to perform all the following operations with a clipping plane enabled to observe the effect on a cross section.

Dilate. Morphological operator to extend the size of the stack.

Erode. Morphological operator to reduce the size of the stack.

Closing. Is a Dilation followed by an Erosion and can be used to close small holes. The opposite operation is **Opening** which is able to open up holes and get rid of noise.

6.3. Processes in “Mesh/Signal”

Project Mesh Curvature. Project the Gaussian curvature. For best results, use change the transfer function to Jet. Experiment with different neighborhood and Autoscale values.

7. 2.5D images

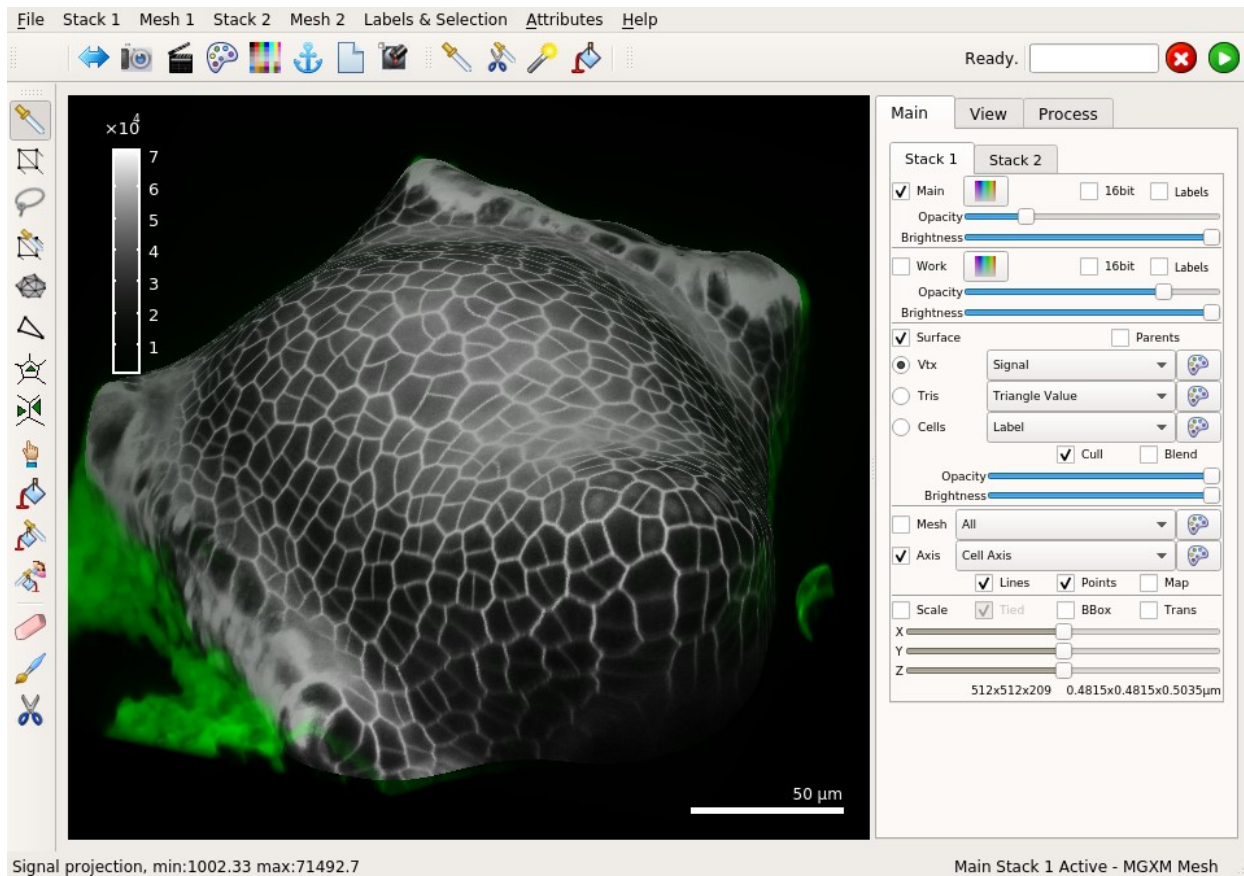


Figure 7. The projected cell wall signal on the surface mesh.

Many samples have too much curvature to simply segment them in 2D. MorphoGraphX was designed to work on curved (2.5D) images. This allows it to be used on samples that have too much curvature to process in 2D. In practice most biological samples contain some curvature, and MorphoGraphX's use of 2.5D images can simplify their processing. To create a 2.5D image perform the following:

Subdivide the mesh and project the signal onto the mesh. Run the process “Mesh/Structure/Subdivide”, then run “Mesh/Structure/Smooth Mesh”. Ensure that the “Surface” checkbox is selected, and deselect the “Mesh” checkbox. Then run the process “Mesh/Signal/Project Signal”. The stack Main Store must be selected and the Work Store deselected, so that the original data is projected, not the processed stack. At this point you should see the outlines of the cells. If the image is not fine enough, repeat the Subdivision and the Smoothing. Only subdivide when it visibly improves the image quality as large meshes are slower to process. Define the Minimum and Maximum distance in “Project Signal” process. These distances are in the negative direction of the surface normal and tell MorphoGraphX which part of the signal to project onto the surface. In this example, try 3 µm

for the minimal and 6 μm for the maximal distance. For better visualization of the mesh, you can turn down the opacity of the stack with the slider in the Main tab.

Visualize projected data. If you would like to see exactly what data was projected onto the mesh, run the process “Stack/Mesh Interaction/Annihilate”. Make sure the distances are the same as those you used for the projection. If you want to keep this stack you can use the stack menu option Stack 1/Save. You have now created a 2.5D image of the surface layer of cells in the meristem. Have a look to see how it compares with the original sample data. You can adjust the mesh and stack opacity to see both the labels and the signal at the same time. If you turn on the mesh visualization to “Cells”, you will see the wireframe outlines of the cells (once they are segmented).

8. Segmenting cells

After projecting the fluorescence signal on the mesh, the surface can be segmented into cells. The segmentation is done by placing labeled seeds on the mesh surface and then extending them based on the signal using the watershed algorithm. Label seeds can be placed either manually or automatically. While the automatic seeding saves a lot of time, we recommend to get familiar with the manual seeding first, as it helps understand how to use the automatic procedure. Even meshes with the best signal will require some manual correction in areas where the signal is less clear or missing.

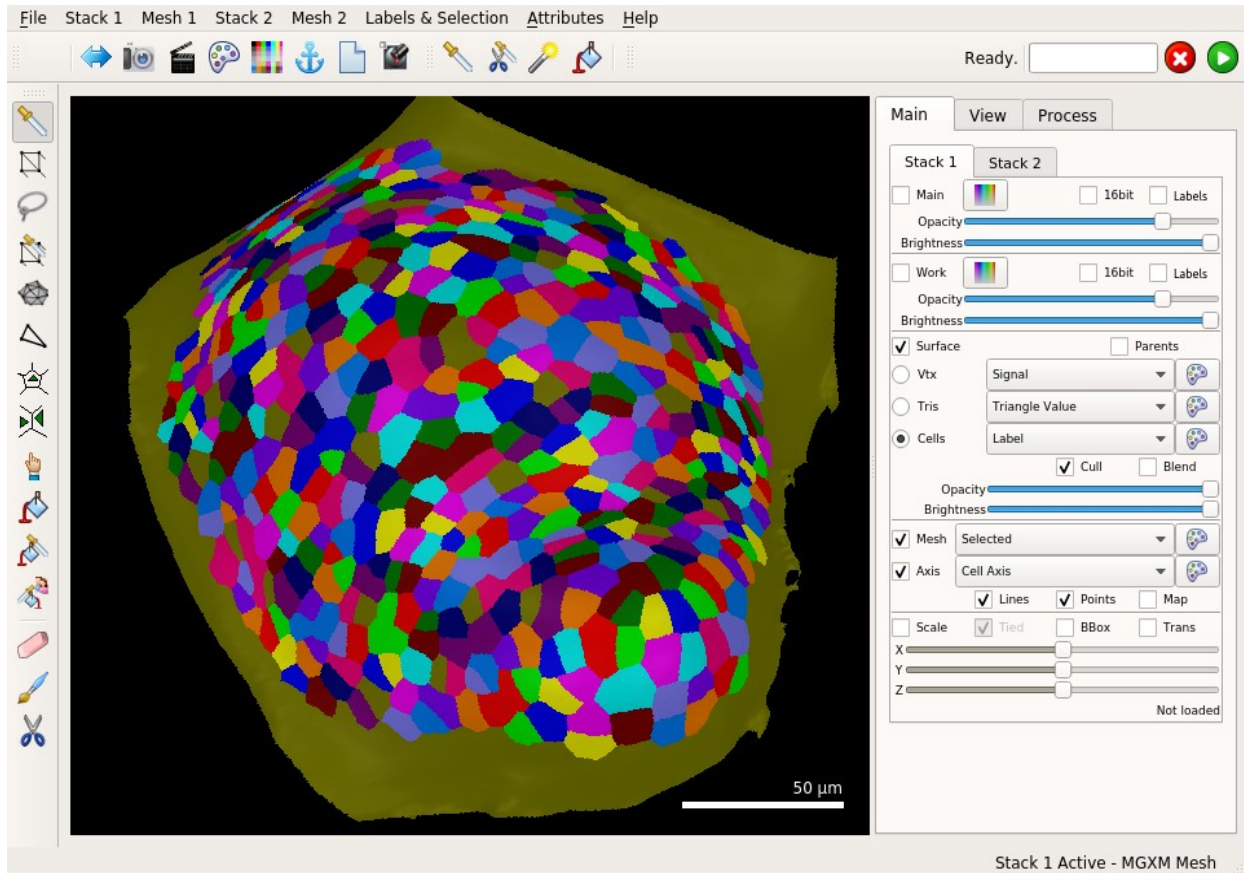




Figure 8. Completely segmented surface mesh.




8.1. Manual seeding and segmentation




Blur the cell outlines. Run the process “Mesh/Signal/Gaussian Blur”, or “Mesh/Signal/Smooth Mesh Signal”. This will close potential gaps in the signal at the cell borders and remove noise. For the Gaussian blur, use a radius close to the width of the cell border (here, 1 μm or below). Note that too much blurring as well as too little can result in faulty segmentation.

Seed the cells. Select the “Add new seed” tool  from the mesh tools panel. Ensure that the “Surface” and “Cells” (with option “Labels”) checkboxes are selected. With the Alt-key pressed, click on each cell to add a seed. If you click and hold you can draw on the surface with the seed. This can help to direct the segmentation, or to label two cells with the same color. When finished seeding the area of interest, draw a line around the entire area with a single seed to create an outside cell which acts as a border (Figure 9). You can hold shift to continue with the current seed, or select the “Add current seed” tool .

Propagate the labels with watershed. Once all the cells of interest are seeded, run the process “Mesh/Segmentation/Watershed Segmentation”. Probably you will have segmentation errors, which you need to fix.

Fix segmentation errors. Errors can be found easily when visualizing the signal together with the cell borders. For visualizing the cell borders turn the Mesh visualization on and select “Cells” in the drop-down menu. Common segmentation errors are over-segmentations (one cell has more than one label), under-segmentations (multiple cells share one label) and areas where walls are not recognized properly.

Over-segmentation is fixed by merging cells together. This can be done by using the “Pick Label” tool  from the Mesh toolbar (not the Volume toolbar) to select the desired label and later use the “Fill label” tool  and Alt+left-click on cells to overwrite the other label with the picked one. The color will change after each click. The “Pick and Fill” tool  combines these two operations.

Under-segmentation is fixed by relabeling cells. First, clear the current label by clicking on the “Label color” icon  from the Miscellaneous toolbar in the top left corner (note: the color of the icon corresponds to the one of the selected label). The color will be replaced with the transparent fill. Then use the “Fill label” tool  from the Mesh toolbar to clear the incorrect cells. You can then re-seed the empty cells using  and re-run the watershed. At this point it is very important to save your data regularly.

Wall recognition problems are handled the same way as under-segmentations. First, clear the affected area, re-seed the cells and run the watershed. Here, it can be helpful to guide the watershed by drawing lines with the seeds that are close to the actual walls.

8.2. Automatic seeding and segmentation.

The automatic seeding and segmentation consists of several sub-processes. You can perform it as one process or as a series of separate processes. We suggest to try the latter option at the beginning because it will allow you to optimize the parameters for your sample and therefore make the procedure more effective.

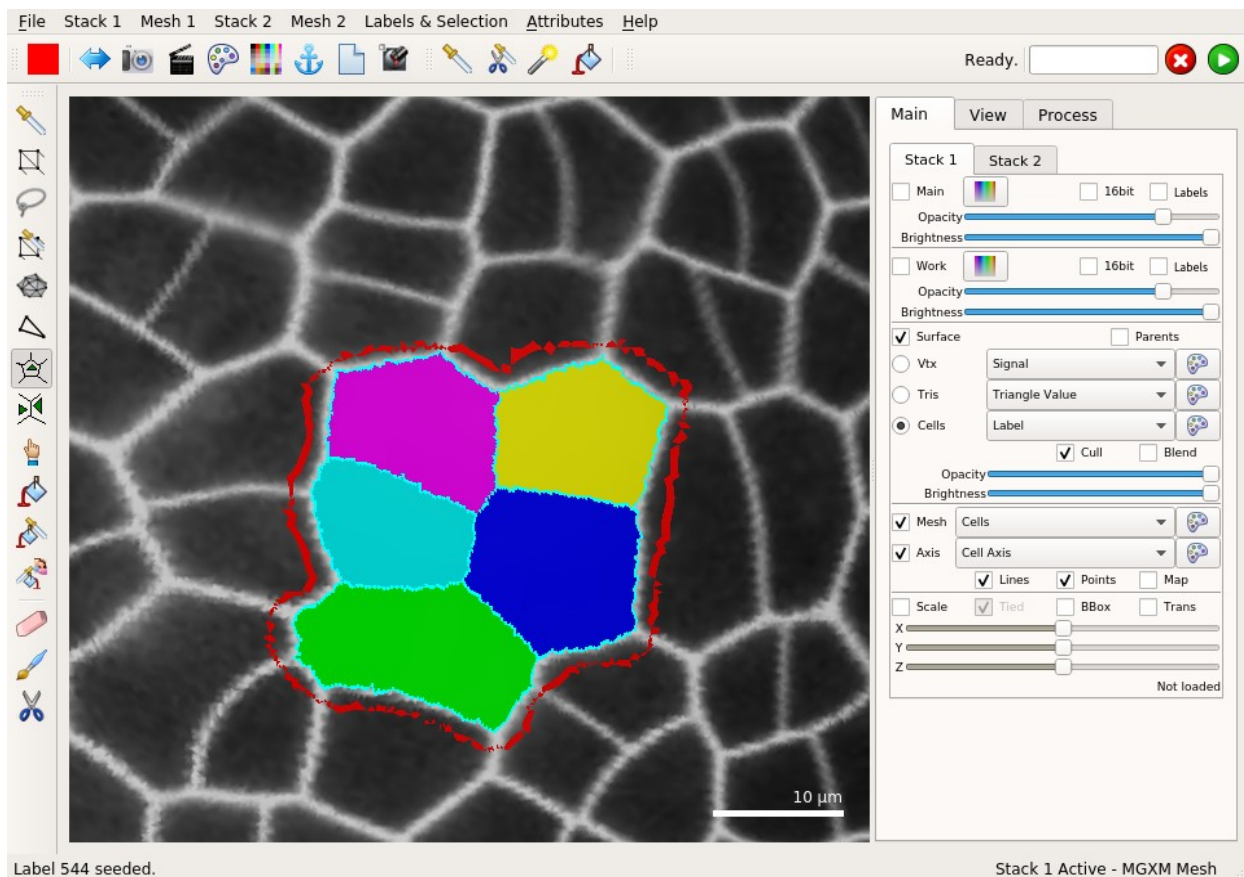


Figure 9. Drawing of the “border cell” after manual seeding of an area of interest.

To perform the automatic segmentation, do the following:

Find seeds for the watershed automatically. This is done by first blurring the signal (process “Mesh/Signal/Gaussian Blur”, with the radius parameter of approximately the radius of the smallest cells in micrometers). Try several values at the first attempt, as over- or under-estimation can result in incorrect segmentation. For the sample data, try 2 μm . Next, run the process “Mesh/Segmentation/Auto Seeding”. A seed will be put at local minima of signal (dark regions) within a given radius. Set the same cell radius as in the previous process, typically near the radius of the smallest cells (here, 2 μm). If you use a larger radius the small cells might be under-segmented (fused). Clear the seeds with the process “Mesh/Segmentation/Segmentation Clear”.

Re-project the original signal onto the mesh. Run the “Mesh/Signal/Project Signal” with the same Min and Max distance parameters as originally used (see section 7).

Blur the cell outlines again. Run the process “Signal/Gaussian Blur”, with a radius roughly equal to half the cell outline width (try 1 μm or below).

Segment the cells. Run “Mesh/Segmentation/Watershed Segmentation” process as in the manual segmentation. Probably you will observe that in some cases one cell has been

seeded several times (*i.e.*, over-segmented). Merge the over-segmented cells based on signal using the process “Mesh/Segmentation/Combine Labels” (described below).

Normalize mesh signal (optional). This process can enhance the color contrast between the inside of the cell and its borders, which is important for the next step (merging cells). However, if the signal is strong and clear, it is not necessary to run this process. To try it, run the process “Mesh/Signal/Normalize Signal”. This time indicate a radius around the size of the larger cells (in the meristem sample data, 5 μm). This process can be useful when the signal is generally low or with high background noise. If you get a lot of internal signal inside the larger cells after running this process, the radius was too small.

Merge the over-segmented cells. Run the process “Mesh/Segmentation/Combine Labels”. Estimate the width of the cell border for your sample in micrometers (*e.g.*, try 1 μm) and put it as the value for the “Border distance” parameter. The “Threshold” parameter indicates the ratio of cell wall brightness to the brightness of cell interior. Therefore, it should be higher than 1. The exact value depends on the quality of signal in your sample. For the sample data, try a value between 1.5 and 1.8. Two cells will be merged if their ratio of common border signal vs. average signal is lower than the threshold value.

If you want to perform all the above processes as one, run the process “Mesh/Segmentation/Auto Segmentation”. You will not be able to set all the necessary parameters and decide whether or not you want to perform signal normalization.

Correct errors. Examine the surface carefully and search for segmentation errors, especially in the regions where the signal is relatively low. Correct them in the same way as for manual segmentation.

Troubleshooting

Symptom: “Cells are under-segmented (not all cell walls are recognized)”. Try lowering the cell radius during signal blurring and seeding.

Symptom: “The mesh combine regions process over or under merges cells”. Try increasing the border distance if the signal extends a bit into the cells, or changing the threshold for merging. If the intensity of the signal varies significantly in different parts of the mesh, try normalizing the mesh signal.

9. Mesh subdivision

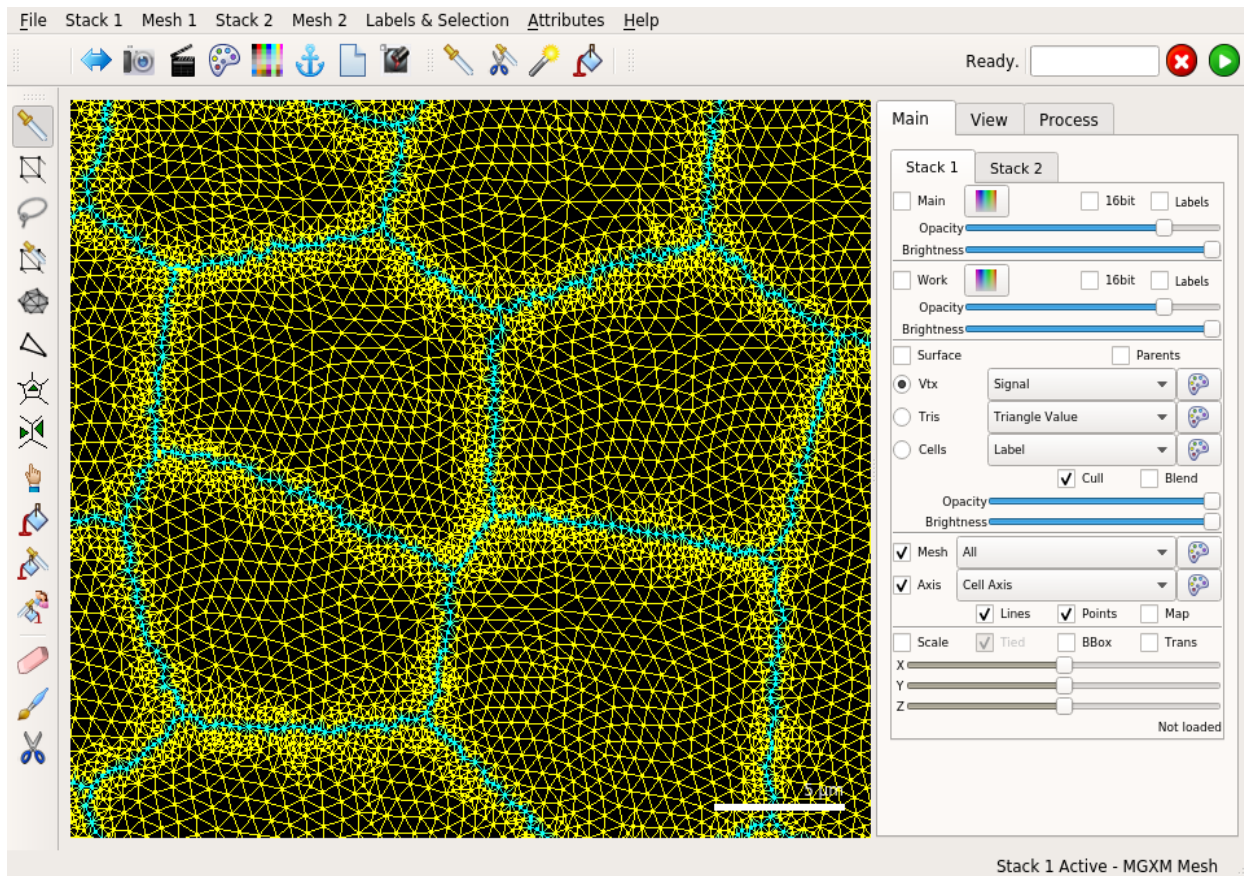


Figure 10. More triangles at the cell border (cyan lines) allow a higher accuracy for the segmentation.

After creating the initial segmentation, the walls may be jagged due to the coarseness of the mesh. To mitigate this without making the mesh too large, increase the resolution near the cell boundaries. If the computer has sufficient memory and/or the stack is small, you can also subdivide the entire mesh (“Mesh/Structure/Subdivide”) before the segmentation. To subdivide near the borders proceed as follows:

Subdivide near the walls. Run the process “Mesh/Structure/Subdivide Adaptive Near Borders”. This will clear the labels within a given distance from the border (specified with the “Border Dist” parameter), and subdivide triangles larger than the “Max Area” parameter, which are near the borders. The “Border Dist” parameter should be smaller than half of the size of the smallest cells (otherwise whole cells might be cleared) and larger than the size of the mesh triangles (otherwise it won’t subdivide anything).

Re-project and re-segment. To re-project the signal, run the process “Mesh/Signal/Project Signal”; to smooth it run the process “Mesh/Signal/Smooth Mesh Signal”, and then re-run the watershed segmentation.

Repeat the above two steps until the image is sharp. With each repetition, you can reduce the border distance since the triangles near the border will be smaller. When the triangles get smaller than the voxels, you will see squarish blocks of signal on the surface when you zoom in. These are the voxels of the image itself, and further subdivision will not increase the quality of the segmentation.

Finalize the mesh. Use the process “Mesh/Cell Mesh/Fix Corner Triangles”. This will remove the black or gray triangles that sometimes appear at the junctions between cells that were not labeled by the segmentation process.

Save the mesh file. We again remind the reader that there is no “undo” function in MorphoGraphX, so it is important to save versions of the mesh frequently in order to avoid losing work.

Now the projected image should look nice and sharp, and the cell borders should be much less jagged.

10. Cell Geometry quantification

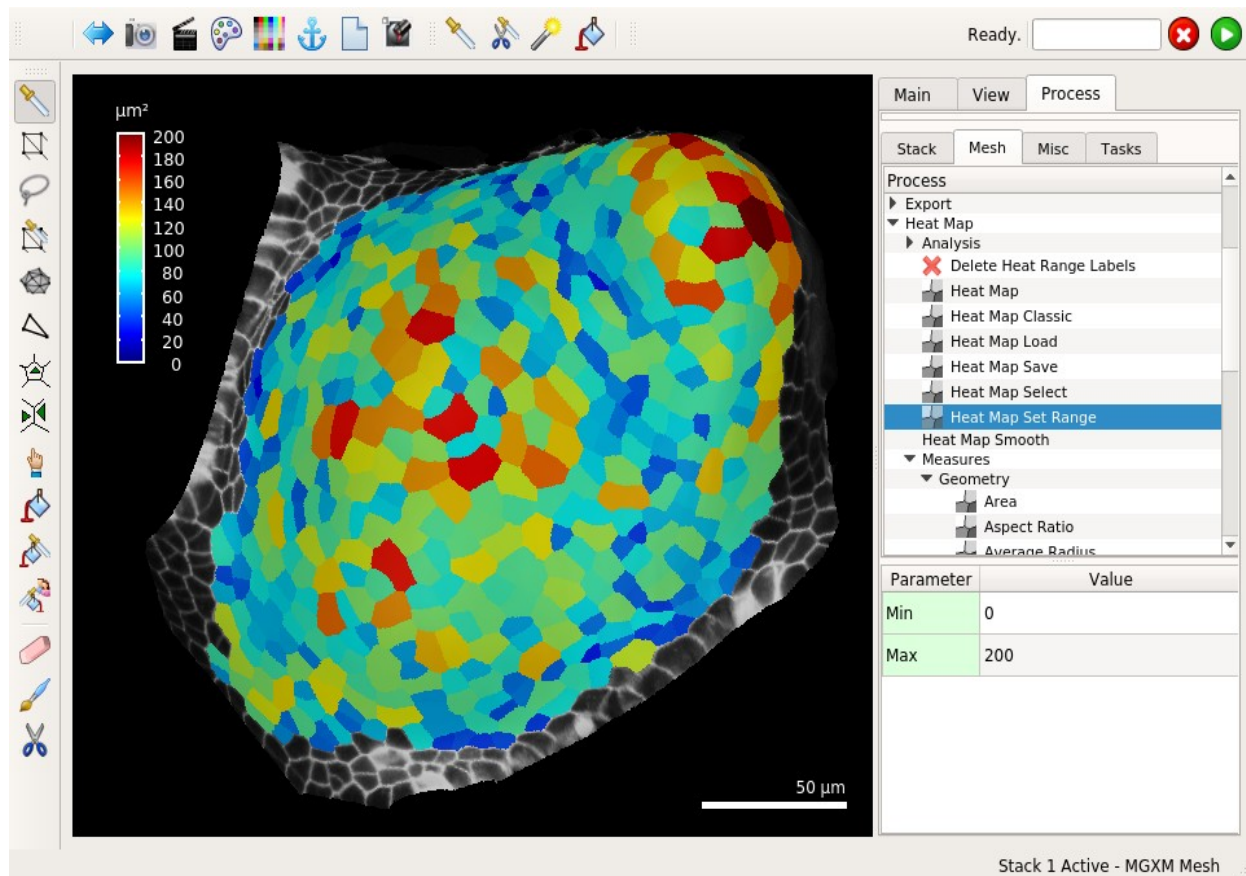


Figure 11. Heat map of cell area.

Once the mesh is segmented into cells, some basic analysis can be performed on it.

Create a heat map. Load the segmented mesh and choose a process in the folder “Mesh/Heat Map/Measures” for the quantification of the cell geometry and other features, e.g., “Mesh/Heat Map/Measures/Geometry/Area” to create a heat map that colors the cells by their area. Alternatively, the process “Mesh/Heat Map/Heat Map” can be used, which opens a separate GUI with extended options (figure 12). Upon running a measure process, it creates a heat map and an attribute map of the measure (see also section 12).

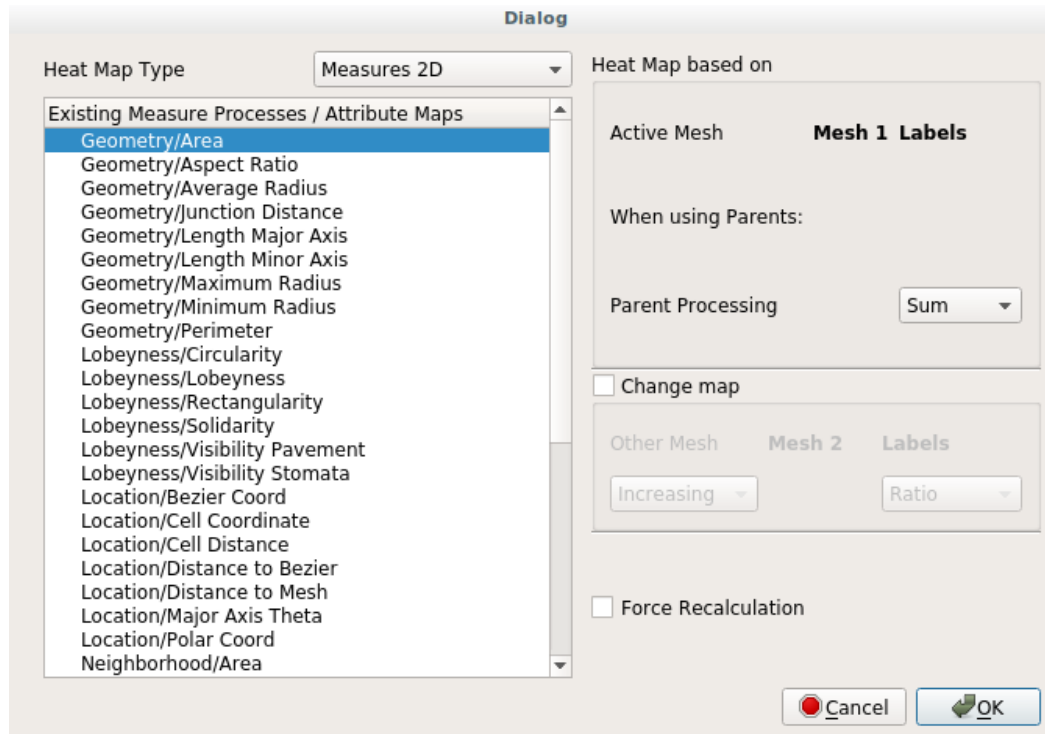



Figure 12. The dialog of the “Mesh/Heat Map/Heat Map” process. Left: “Measure” processes (starting with a “/”) and attribute maps (at the end). Right: the options allowing user to create heat maps based on the parent labels or to create change maps of any measure

Change the range of the scale bar by running the process “Mesh/Heat Map/Heat Map Set Range”.

Change the colors of the heat map by using the color dialog in the Main tab next to “Cells” and the “Label Heat” dropdown menu. By clicking on the “Cell heat colormap” icon  beside the dropdown menu, a “Transfer function editor” GUI dialog window will open. The color model can be changed from RGB to HSV or cyclic HSV to discern the luminance of the image from color information. The background color and the interface can also be edited. The color map can be chosen among a list of predefined color maps and can be modified by moving the black arrows on top of the color scale. Once done, the map can be saved and exported. The exported color map will be saved as .fct function file and can later be imported back in MGX.

Smooth the heat map across the cell neighborhood with “Mesh/Heat Map/Heat Map Smooth”. Heat values can be weighted by cell area and the result can be directly exported to an attribute map.

Clean the heat map. Select “Mesh/Heat Map/Operators/Clean Heat Map” to remove unwanted entries, *i.e.* values for non-existent cells, from the heat map. The “min valid” and “max valid” parameters are used to set the range of valid cells from the heat map; all cells

out of this range will be deleted. The “selection” parameters allows you to work only on manually selected cells and the cells not selected will be removed.

Bin cells or values of the heat map. Use the active heat map to create a binning (discretization) based on cells or values and set the parent label according to the bin by running “Mesh/Heat Map/Operators/Heat Map Binning”. Cells can be binned by selecting “Bins Cells” mode, heat map values are binned by “Bins Heat”, while “Threshold” mode creates only two bins based on the parameters in the process “Mesh/Heat Map/Heat Map Select”.

Normalize the heat map values by running “Mesh/Heat Map/Operators/Heat Map Normalize”. The values will be normalize to a 0 to 1 range.


Scale the heat map values with “Mesh/Heat Map/Operators/Heat Map Scale Values” according to the chosen type (inverse, square, sqrt, log, pow, ordinal, add, mult). Log uses the base of 10. The parameter (P) is used for pow (P), add (+P), mult (*P). The inverse of 0 is 0.

Select parent cells according to heat map values with “Mesh/Heat Map/Operators/Heat Map Scale Values” and “direction” parameter set to Heat -> Parent. The opposite can also be performed by changing the “direction” value.

Transform attribute maps into heat map with “Mesh/Heat Map/Operators/Heat Map Transform”. By running the process, a GUI dialow window will open. Two different attribute maps can be combined together by ticking “Combine Maps” and selecting two base attribute maps. The attribute map can also be transformed in a new manipulated heat map by ticking “Transform Map”, selecting the type of map, type of transformation and lower and upper threshold. A name can be assigned to the generated map in “Name of Manipulated Map”. Note that only “Combine Maps” or “Transform Map” can be ticked at one time, not both.

Export the heat map to a CSV file using the process “Mesh/Heat Map/Heat Map Save”. CSV files can be opened by almost any other external data analysis software or as a spreadsheet. There are more options to export larger amount of data more efficiently which are explained in section 12. The heat map can also be exported to an attribute map by running “Mesh/Heat Map/Operators/Export Heat to Attr Map” or transformed . The process generates an attribute map with a chosen name from the active heat map values.

	A	C	D	E	F	G
1	Label	Value	Area (μm²)	Center_X	Center_Y	Center_Z
2		11 48.8824	48.8824	50.4784	-75.0226	-15.8784
3		12 63.5403	63.5403	56.0198	-71.792	-16.3212
4		13 55.2761	55.2761	60.9629	-69.9763	-17.7705
5		14 77.1842	77.1842	66.1393	-64.5359	-14.7436
6		15 54.2675	54.2675	72.2261	-59.1070	-12.6070
7	Cell label	Value associated to color map = cell area	Cell center coordinates (x,y,z).			
8		17		78.292	-40.6395	-0.896271
9		18 35.7367	35.7367	82.4887	-34.1991	0.561473
10		19 48.8837	48.8837	82.2042	-16.8996	7.12617
11		22 44.182	44.182	83.6806	7.03876	8.70603
12		23 25.3023	25.3023	82.2381	2.04306	9.36707
13		24 37.9115	37.9115	85.3274	-4.73887	7.77237

Figure 13. Example of spreadsheet output after running “Heat Map Classic” of type “Area, Geometry”. This file is contained in the sample data folder under “meristem_T0_cell_area.csv”. The cell labels are listed along with their areas. If you click the “Map” checkbox inside MorphoGraphX, it will show the cell label numbers. You can also use the “Pick label” tool  in the Mesh toolbar to find out which label belongs to which cell.

11. Quantifying Fluorescence

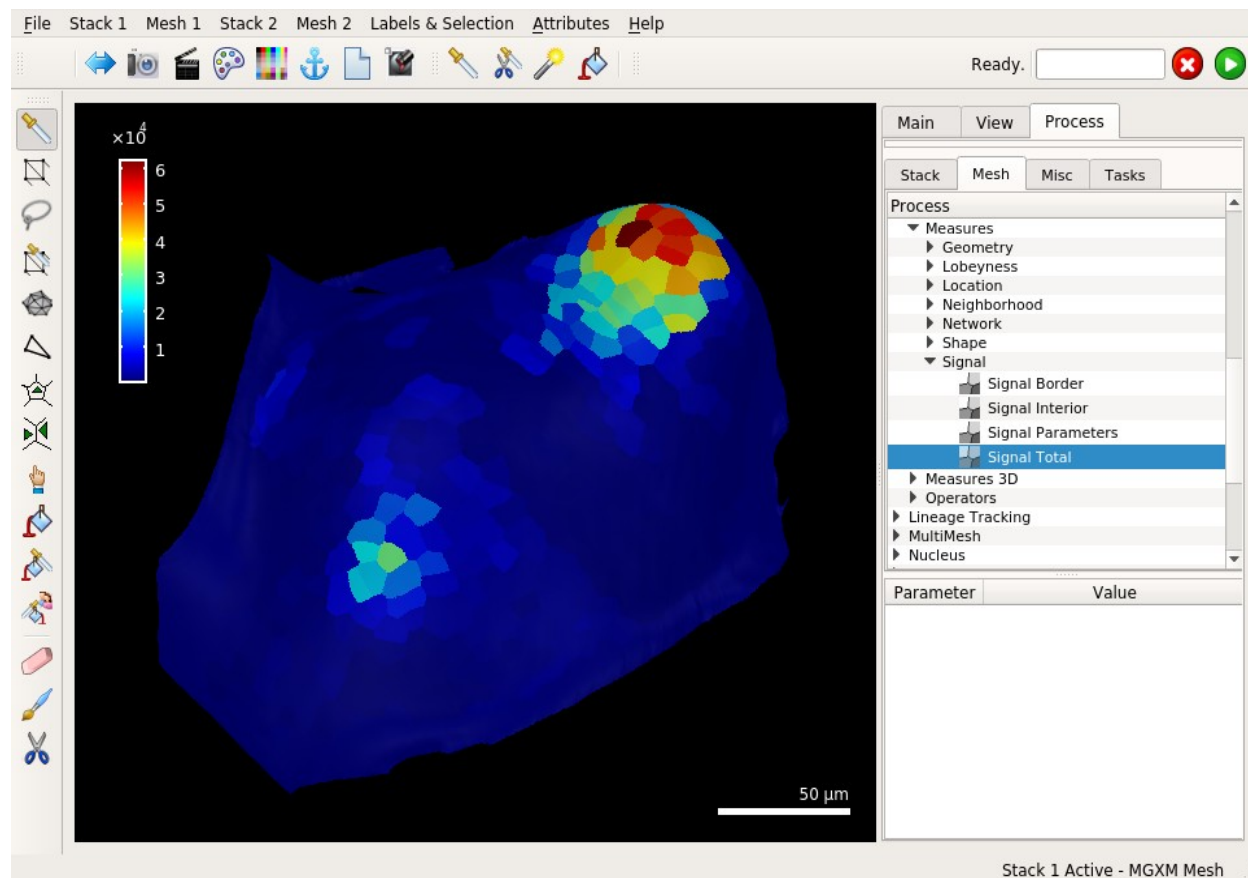


Figure 14. Heat map of the intensity of a projected fluorescence signal.

Gene expression can be quantified by calculating the intensity of a nuclear or cytoplasmic marker for each of the segmented cells. The data must have been acquired at the same time as the cell wall/plasma membrane marker in a different channel. The Kierzkowski_2012 dataset of a tomato meristem can be used for this purpose as the wall is stained with PI and a DR5 fluorescence was collected in a separate channel. We can now analyze this data based on your segmentation:

Load the Data. Use the Stack menu to load the fluorescence channel you want to quantify into the Main or Work Store of the stack corresponding to your mesh. The mesh now contains the cell geometry information, so that the stack can be replaced (e.g., load “meristem_T0_DR5.mgxs” into Stack 1, Work).

Project the data onto the mesh. Run the process “Mesh/Signal/Project Signal”. At this point, be sure to have the surface visualization set to “Vtx” (with “Signal”), the Mesh visualization set to “Cells” and the correct stack active (e.g., Work Stack 1). You should now see which epidermal cells are expressing DR5 at a given depth. The default parameters Min Dist and Max Dist are set to 1 and 5, which means that the signal is taken only between 1

and 5 μm away from the curved surface. This depth corresponds to the epidermal layer in the example tomato meristem data. Try to change the depth parameter (e.g., 6 and 12, corresponding to the sub-epidermal layer) to see the effect on the projection, and compare this with the whole stack.

Make a heat map. Run “Mesh/Heat Map/Measures/Signal” to quantify the amount of marker signal, for the total signal use Signal Total. A heat map will be displayed giving the average signal intensity per cell.

12. Measure processes, attribute maps & data export

12.1. Measure processes

Measure processes are a standardized way to generate heat maps in MorphoGraphX 2.0. They are located in the “Mesh/Heat Map/Measures” folder and categorized into sub-folders. They can also be accessed using the GUI of the “Mesh/Heat Map/Heat Map” process. A complete list of the implemented measures can be found at the end of the user guide.

12.2. Attribute maps

Attribute maps are new to MorphoGraphX version 2.0. They are data containers that are stored within the mesh and can contain different types of data generated by processes. The process “Mesh/Attributes/Manage Attributes” (also accessible via the Menu “Attributes/Mesh 1” or “Mesh 2”) is dedicated to managing attribute maps. It opens a window showing all attribute maps currently stored with the mesh and allows you to delete existing attribute maps from the mesh.

The most important applications of attribute maps are as follows:

Parent labels, heat maps and cell axis data. These are mesh related data structures already known from the first version of MorphoGraphX. They are now also stored as attribute maps (see sections 9, 10 and 13).

Measures. All measure processes create a heat map (see sections 10 and 12.1). They also automatically generate an attribute map in a standardized format with a name “Measure Label Double” followed by the name of the measure. Whereas only one heat map can be visualized at a time, there can be as many measure attribute maps as needed. There are also different measure attribute maps:

Measure Label Double stores for every cell (parent) label a single floating point number (“Double”). This is typically the information of a heat map. Such attribute maps can be loaded as heat map using the “Mesh/Heat Map/Heat Map” process (the existing attributes are listed in the GUI). The current heat map can be exported as an attribute map using the process “Mesh/Heat Map/Operators/Export Heat to Attr Map”.

Measure Label Int stores an integer for every label. An example for this are the parent labels or cell type labels. Such attribute maps can be loaded as parent labels the using the “Mesh/Lineage Tracking/Parent Import from Attr Map” process. The current parent or cell type labels can be exported as attribute map using the process “Mesh/Lineage Tracking/Parent Export to Attr Map”.

Measure Label Vector stores a vector for every label. This type is relatively rare, but for instance used for storing cell centers.

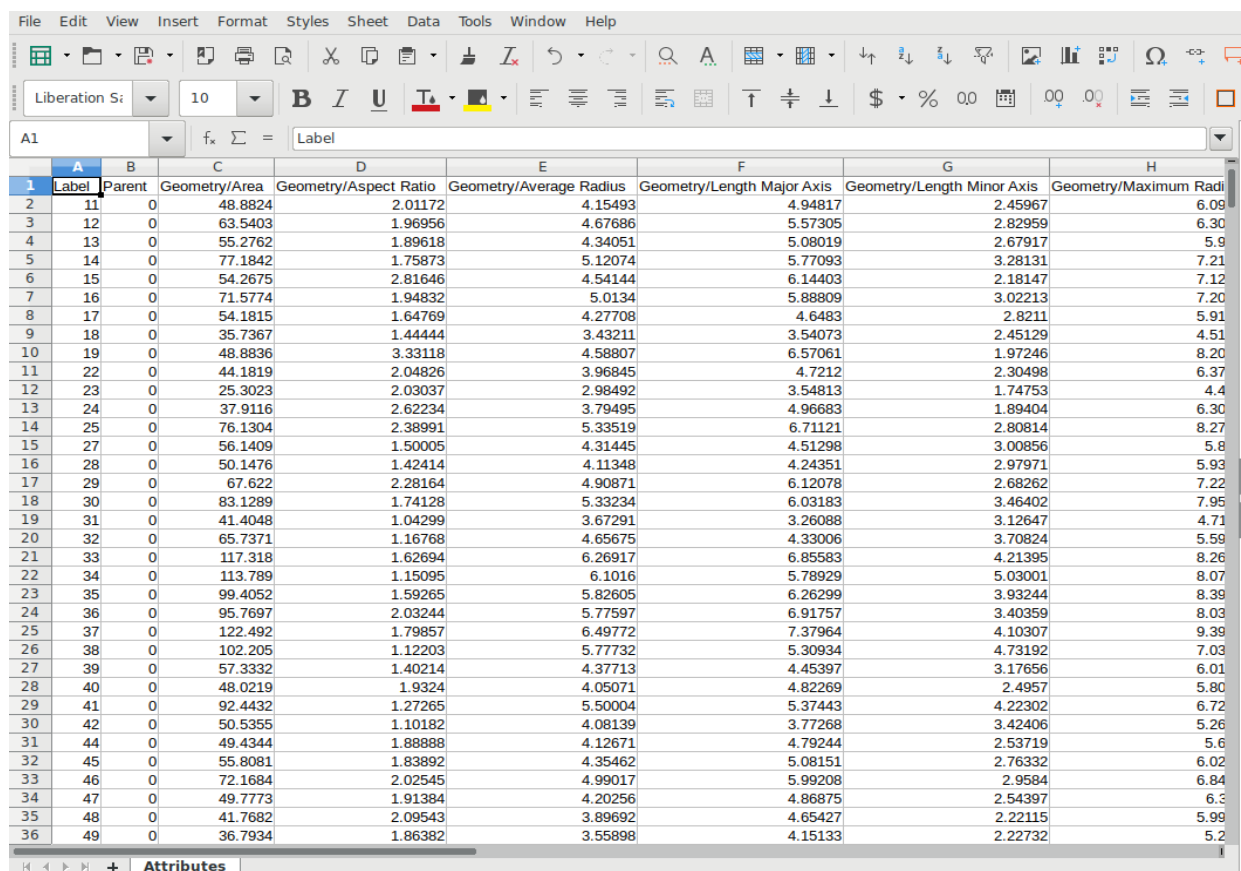
Measure Label Tensor stores a tensor (a 3 by 3 matrix) for every label (example: PDGs or any cell axis type). There are the processes “Mesh/Cell Axis/Cell Axis Export To Attr Map” and “Cell Axis Import From Attr Map” to interact with this type of attribute maps.

For more details about **Cell axis and custom directions** see section 16.

12.3. Data export

Measure attribute maps of all kinds can be exported into CSV files using the process “Mesh/Attributes/Save to CSV”. This process opens a user interface where the currently present measure attribute maps are shown and the user can chose which ones are to be saved into a single CSV file.

Note: Some data analysis software benefits from adding additional columns to the exported data, such as: the name of the sample or the time point. For this an extended version of the export process can be used: “Mesh/Attributes/Save to CSV Extended”. Here additional columns are added based on the parameters “Genotype”, “Sample”, “Timepoint” and “Stage”.



	A	B	C	D	E	F	G	H
	Label	Parent	Geometry/Area	Geometry/Aspect Ratio	Geometry/Average Radius	Geometry/Length Major Axis	Geometry/Length Minor Axis	Geometry/Maximum Radi
2	11	0	48.8824	2.01172	4.15493	4.94817	2.45967	6.08
3	12	0	63.5403	1.96956	4.67686	5.57305	2.82959	6.30
4	13	0	55.2762	1.89618	4.34051	5.08019	2.67917	5.9
5	14	0	77.1842	1.75873	5.12074	5.77093	3.28131	7.21
6	15	0	54.2675	2.81646	4.54144	6.14403	2.18147	7.12
7	16	0	71.5774	1.94832	5.0134	5.88809	3.02213	7.20
8	17	0	54.1815	1.64769	4.27708	4.6483	2.8211	5.91
9	18	0	35.7367	1.44444	3.43211	3.54073	2.45129	4.51
10	19	0	48.8836	3.33118	4.58807	6.57061	1.97246	8.20
11	22	0	44.1819	2.04826	3.96845	4.7212	2.30498	6.37
12	23	0	25.3023	2.03037	2.98492	3.54813	1.74753	4.4
13	24	0	37.9116	2.62234	3.79495	4.96683	1.89404	6.30
14	25	0	76.1304	2.38991	5.33519	6.71121	2.80814	8.27
15	27	0	56.1409	1.50005	4.31445	4.51298	3.00856	5.8
16	28	0	50.1476	1.42414	4.11348	4.24351	2.97971	5.93
17	29	0	67.622	2.28164	4.90871	6.12078	2.68262	7.22
18	30	0	83.1289	1.74128	5.33234	6.03183	3.46402	7.95
19	31	0	41.4048	1.04299	3.67291	3.26088	3.12647	4.71
20	32	0	65.7371	1.16768	4.65675	4.33006	3.70824	5.59
21	33	0	117.318	1.62694	6.26917	6.85583	4.21395	8.26
22	34	0	113.789	1.15095	6.1016	5.78929	5.03001	8.07
23	35	0	99.4052	1.59265	5.82605	6.26299	3.93244	8.39
24	36	0	95.7697	2.03244	5.77597	6.91757	3.40359	8.03
25	37	0	122.492	1.79857	6.49772	7.37964	4.10307	9.39
26	38	0	102.205	1.12203	5.77732	5.30934	4.73192	7.03
27	39	0	57.3332	1.40214	4.37713	4.45397	3.17656	6.01
28	40	0	48.0219	1.9324	4.05071	4.82269	2.4957	5.80
29	41	0	92.4432	1.27265	5.50004	5.37443	4.22302	6.72
30	42	0	50.5355	1.10182	4.08139	3.77268	3.42406	5.26
31	44	0	49.4344	1.88888	4.12671	4.79244	2.53719	5.6
32	45	0	55.8081	1.83892	4.35462	5.08151	2.76332	6.02
33	46	0	72.1684	2.02545	4.99017	5.99208	2.9584	6.84
34	47	0	49.7773	1.91384	4.20256	4.86875	2.54397	6.3
35	48	0	41.7682	2.09543	3.89692	4.65427	2.22115	5.99
36	49	0	36.7934	1.86382	3.55898	4.15133	2.22732	5.2

Figure 15. Example output file generated by the process “Mesh/Attributes/Save to CSV Extended”.

13. Lineage tracking

In time lapse data the lineage of cells can be tracked by assigning cells parent labels from a previous time point.

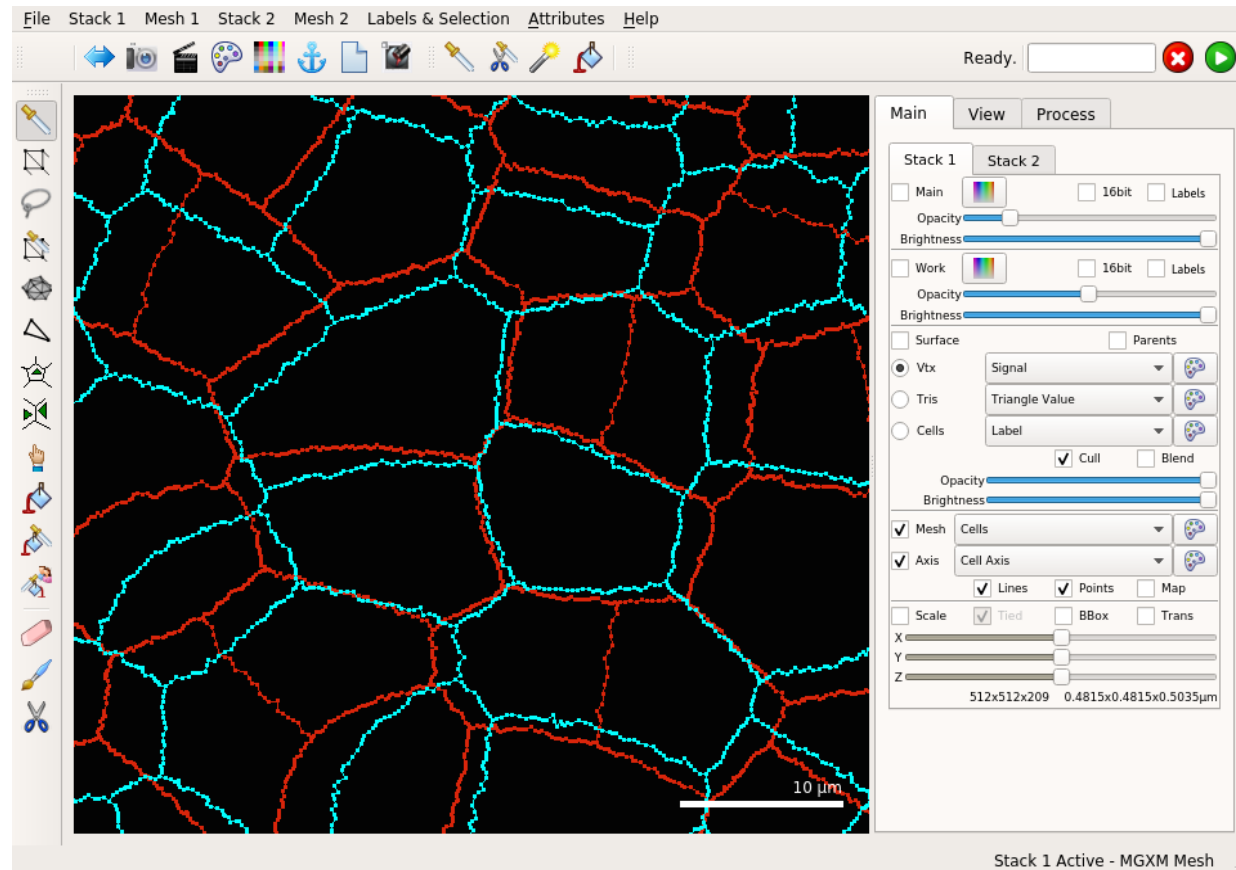



Figure 16. Roughly aligned cell border outlines. Mesh 1 (cyan) and Mesh 2 (red) of the tomato meristem.

Before analyzing the development of cells (e.g., make a growth map) between 2 time points, we need to first segment a second time point of the time-lapse in the same way at the first one.


In this example, we assume that data for the first time point is stored in Stack 1 and for the second one in Stack 2. Generally, labels on the first time point will not correspond to the second time point. In order to compute a growth map, we first need to associate the labels of the “daughter cells” (in Stack 2) with the labels of their “parents” (in Stack 1). While we added a pipeline to accomplish this task in a semi-automatic way (see section 17.2), we still suggest to go through the manual procedure at least once. The automatic method still requires few manually labeled cells and might also leave cells unlabeled that need to be filled in manually.

The manual way to lineage track cells is as follows:

Go to Stack 1. In the Main tab, turn “Stack” off. Turn “Surface” off with “Labels” active. Turn “Mesh” on with the “Cells” option selected. You should now see the cell outlines of Mesh 1.

Go to Stack 2, Turn “Stack” off. Turn “Surface” on with “Parents” active. The labels will disappear as no parents are defined yet. Turn “Mesh” on with the “Cells” option selected. To easily distinguish the cell mesh of the second time point from the first one, change its color using the “Colors Editor” tool .

Now set the Control-Key-Interaction to Stack 1 on the “View” tab. This will allow to move Stack 1 independently from Stack 2. Looking from the side, hold the Control key and move the wireframe cell outline of Stack 1 some distance above Stack 2. Now, rotate the view to look from the top (see figure 16). Looking from the top, try to align a few center cells as precisely as possible. If the growth was relatively large, you can also scale Mesh 1 to increase its size (in the “Main” tab, the “Scale” checkbox). In general, it won’t be possible to superimpose all cells at the same time. This is not a problem as Mesh 1 can be moved later after having labeled a few cells.

Transfer labels from Mesh 1 to parents of Mesh 2. Make sure the “Surface” visualization and “Parents” is checked for Stack 2, and that Stack 2 is active (*i.e.*, Stack 2 tab is selected). All editing operations work only on the active stack. Select the “Grab label from other surface” tool  from the mesh toolbar on the left. Now, when you hold the Alt key, the tool will transfer the label of the cell you are looking through from Mesh 1 onto parents of Mesh 2. Try a few and verify that you are getting the correct labels. Transfer all of the labels in this way. In the second time point, it may happen that some of the cells have divided. Both daughter cells will get the same label but will remain separated.

Verifying the parent labels. Mistakes are common when doing the parent labeling for large meshes. MorphoGraphX allows to find them using two different methods: The first is the process “Mesh/Cell Axis/PDG/Check Correspondence” that checks whether cell junctions are identical in Mesh 1 and Mesh 2. This process is explained in section 15. Another method is to compute the cellular growth heat map and look for outliers (see section 14).

Save the parent labels. Run the process “Mesh/Lineage Tracking/Save Parents”. Note: Make sure that the stack of the second time point (in this case, Stack 2) is active when you save the parents! Otherwise, you will save an empty file. If you save your mesh now, the parents will be saved with it in an attribute map. However, we still recommend saving parents as separate CSV files for several reasons. First, you have better control by putting sample description, date, etc. in the file name. Second, when working with time-lapse data across multiple time points, you will be able to combine multiple files later. Finally, there are other processes that use the parent labels as a secondary cell label to assign cell types or zones within an organ and might overwrite the parent labels.

Troubleshooting:

Symptom: “It is not possible to superimpose all cells at the same time”. This is normal. Start parent labeling with a few cells in the center of your sample and then move Stack 1 along Stack 2 while gradually adding new parent labels.

Symptom: “Samples in corresponding time points have very different sizes”. In either Stack 1 or Stack 2, check the “Scale” box on the bottom of the stack tab. Now, you can scale your stack along all three axes, which can make parent labeling easier.

14. Comparing data from two time points

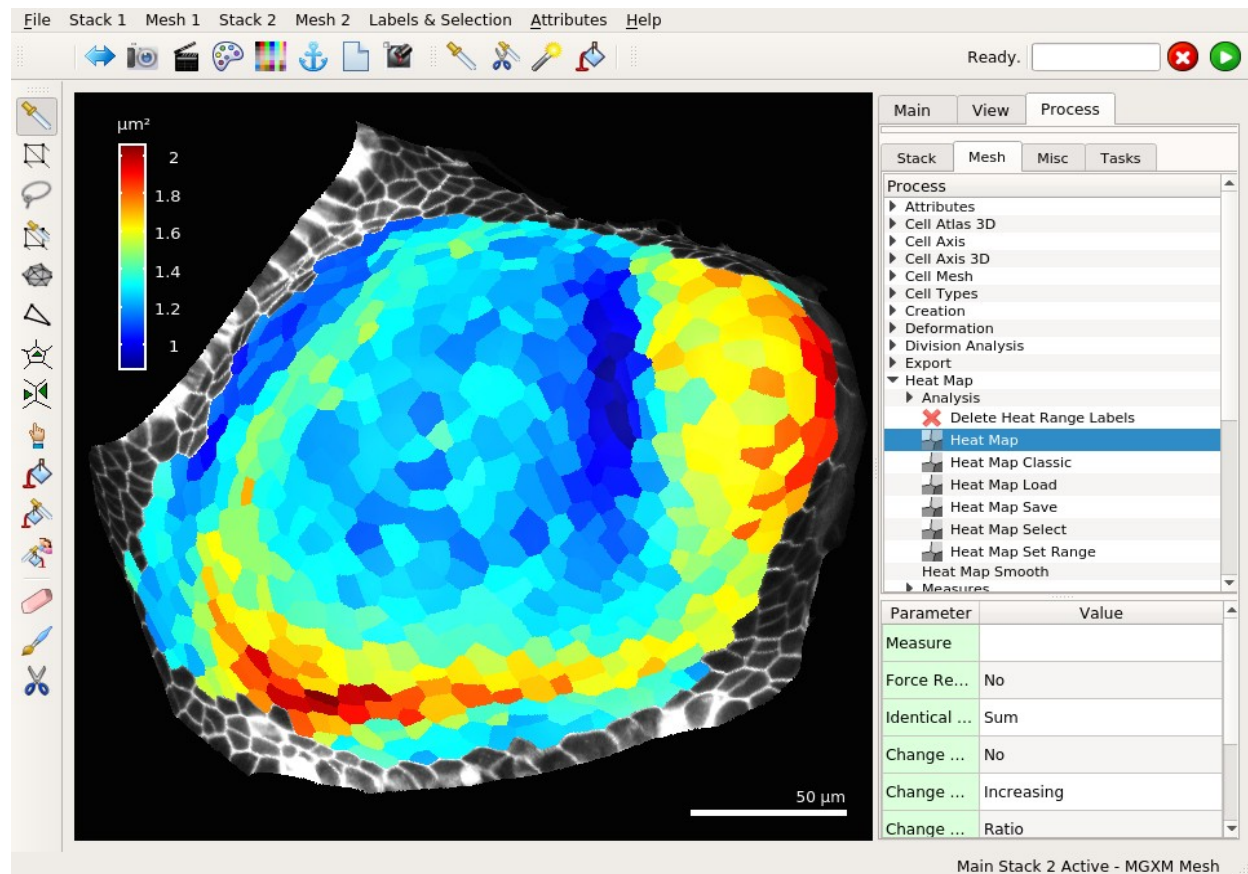


Figure 17. Heat map of area change (growth) between two time points.

With a complete cell lineage, changes between the two time points of the sample can be quantified. The most basic way is to create heat maps of cell growth and proliferation. This will correspond to the change in area of cells and number of cells over the time point. In order for this to work, it is important that the parent labeling (lineage tracking) is perfect. Mistakes will appear as outliers in the growth map and are very easy to spot. It is very unusual for lineage tracking to be perfect the first time.

To create the growth map:

Run the heat map process. Load the meshes you want to compare. In Mesh 1 (consisting of an earlier time point) select “Labels” and in Mesh 2 select “Parents”. You should see corresponding cells on both meshes painted with the same color. Now run the process “Mesh/Heat Map/Heat Map”. Select the “Geometry/Area” measure, the “Change Map” check box and chose a direction. If you are visualizing the result on the first time point, then set the change map to “Increasing” to quantify cell area increase over time. MorphoGraphX will autoscale the color range to the data and will color the cells with the most expansion in red, and those with the smallest in blue.

- Tip: You can create change maps for any kind of data. Simply choose a different measure to create a change map for instance for “Signal Average” or “Circularity”. Note that cell divisions greatly affect the results of change heat maps. By default, the heat values of daughter cells are summed and the change with the value of the parent cell is quantified, which is the appropriate method for cell area. In some case, it might be appropriate to average the heat values of the daughters instead. This can be done by selecting “Parent based heat map” and change the dropdown menu to “Average”. Generally it is important to check which mesh is active and whether labels or parent labels are active when computing change maps. The GUI of the process “Mesh/Heat Map/Heat Map” indicates these two things.

Fix problems. If everything is perfect, you will see a heat map like in the picture at the beginning of this section (see figure 17). More likely, you will see obvious outliers, individual cells which are much bluer than their neighbors, or much redder. Go to these cells and fix any segmentation errors. Note that the error might be in either Stack 1 or Stack 2, or it might have been a problem with the parent labeling transfer. Do not forget to save the parents after correction.

Create a cell proliferation map. Run the process “Mesh/Lineage Tracking/Heat Map Proliferation”. This process will work only for Stack 2 where you have “Parents” active. This will display the heat map showing how many cells originate from one parent label.

15. Principal directions of growth (PDGs)

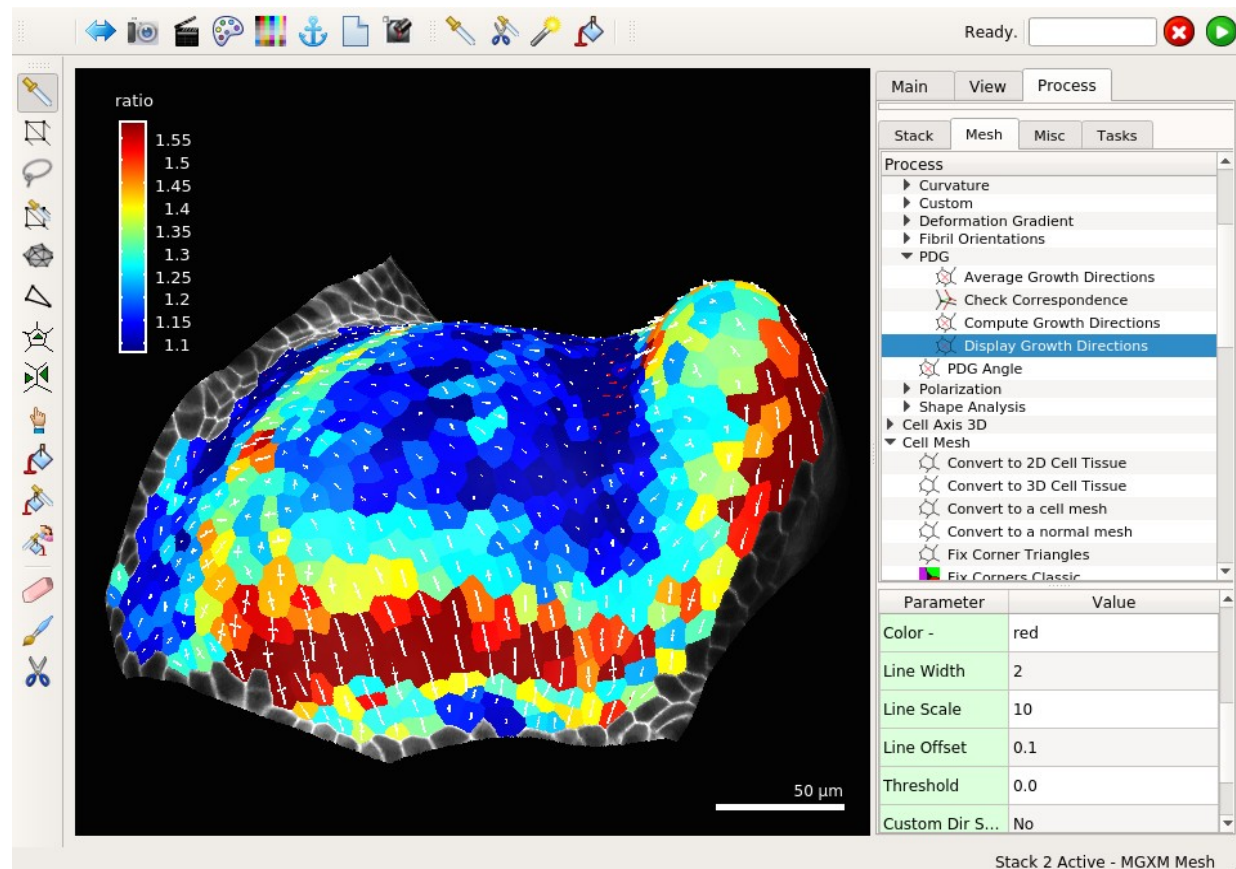


Figure 18. Segmented mesh with lines indicating the amount and direction of the maximum and minimum growth direction.

Principal directions of growth (PDGs) describe the growth anisotropy at the level of individual cells. The output of this process are vectors that, for each cell, give the direction of maximal and minimal growth and can be visualized with white lines on top of the cells. PDGs are computed based on cell junctions, which are the vertices that are at the intersection between 3 different labels. Before we can compute the PDGs, we have to identify which junctions in the second time point correspond to junctions in the first one.

Fix corners. If you haven't done so after the segmentation, run the process "Mesh/Cell Mesh/Fix Corner Triangles" for both meshes. This will make sure that all the junctions between cells are correctly segmented and that there is no triangle in the mesh that is left without a label. Fix Corners should report 0 vertices after running. Otherwise, run the process again. Save each mesh once Fix Corners is complete.

Find the correspondence between junctions. Since the labels are different between the two meshes, load the parents in the second time point. Run "Mesh/Cell Axis/PDG/Check Correspondence". The output should be a color map, with cells which are correctly identified

in blue and cells causing problems in red. The vertices from the first time point (Stack 1) that could not be correctly identified in the second time point (Stack 2) are selected and should appear in red. Make sure Stack 1 is active and has “Mesh,” “Lines,” and “Points” selected and that you have ticked “Parents” on Stack 2.

Troubleshooting Check Correspondence. The Check Correspondence process can abort and throw an error. This typically happens if there is some irregularity in one of the two meshes such as:

- Unlabeled vertices
- Vertices, edges or triangles disconnected from the main mesh which are sometimes difficult to see
- Several cells with the same label, or a cell split into several fragments
- Holes or tunnels in the mesh.

Figure 19 shows an example. To remove such errors manually, use the “Pick Label” and “Add Current Seed” tools. It is also possible to run the processes “Mesh/Cell Mesh/Tools/Relabel Fragments” followed by “Mesh/Cell Mesh/Fix Corner Triangles” or “Mesh/Cell Mesh/Tools/Extend Cells”.

If the Check Correspondance process repeatedly fails, make sure the process “Mesh/Cell Mesh/Convert to a cell mesh” runs on both meshes. If it fails on one of the meshes then you need to find and fix its problems first.

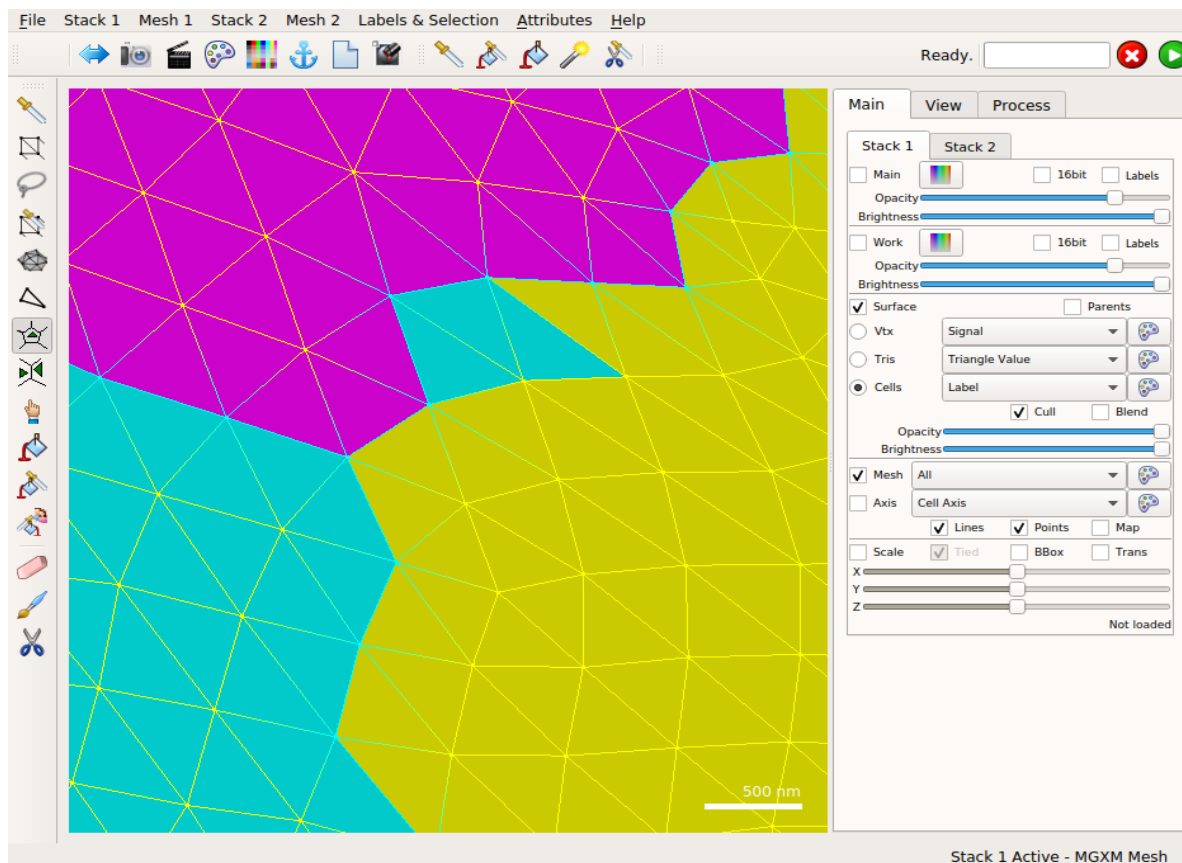




Figure 19. Typical problem, which will prevent the “Check Correspondence” process to run properly. Some triangles of the cyan cell are disconnected from its main part.

Correct the errors found by check correspondence. If the check correspondence process ran without error, it will return with heat maps on both meshes. Red cells highlight sources of errors in the correspondence between junctions. Most of the time they will be caused either by wrong parent labeling or missing parents, which can be fixed easily by modifying the parent labels, or by an “exchange” in neighborhood (see Figure 20). Use the “Pick label”  and “Add current seed” tools  to modify the meshes. Save the meshes after correction and re-run “Check Correspondence” until all the cells that are to be analyzed are blue. This process will assure that your lineage is correct and it is advisable to run it (with consecutive time point pairs) also if you have multiple time points.

Compute the PDGs. Once the correspondence is complete, run the process “Mesh/Cell Axis/PDG/Compute Growth Directions” with Stack 1 active. PDGs are computed for each cell individually using all corresponding cell junctions of that cell.

Display the PDGs or alter the visualization parameters using the process “Mesh/Cell Axis/PDG/Display Growth Directions”. You can choose what type of heat map and which vectors to display. StretchMax (resp. StretchMin) is the value of deformation (stretch ratio) in the maximal (resp. minimal) direction. A stretch ratio of 1 means no deformation, 2 means

an elongation by 100%, 0.8 a shrinkage of 20%. The product of StretchMax and StretchMin is equivalent to the areal growth of the cell polygon. The color and size of the PDG vectors can also be modified. By default, vectors corresponding to expansion (stretch ratio > 1) are displayed in white, while red is used to draw the direction of shrinkage (stretch ratio < 1). The “Threshold” parameter is used to display PDGs axis only in cells for which the anisotropy is above a given value. Note that all “Custom” options for heat map or vectors require custom directions which are described in section 16.

Save the PDGs with the process “Mesh/Cell Axis/Cell Axis Save”. The maximal and minimal directions of growth (3D vectors) will be saved together with the values of deformation (stretch ratio) associated to them as a CSV file (see figure 22). PDGs are also stored as attribute maps but we recommend to keep a backup CSV file.

Load the PDGs with the process “Mesh/Cell Axis/Cell Axis Load”, “Type: PDG”. Make sure that parent labels are also loaded on the second time point before loading the PDGs.

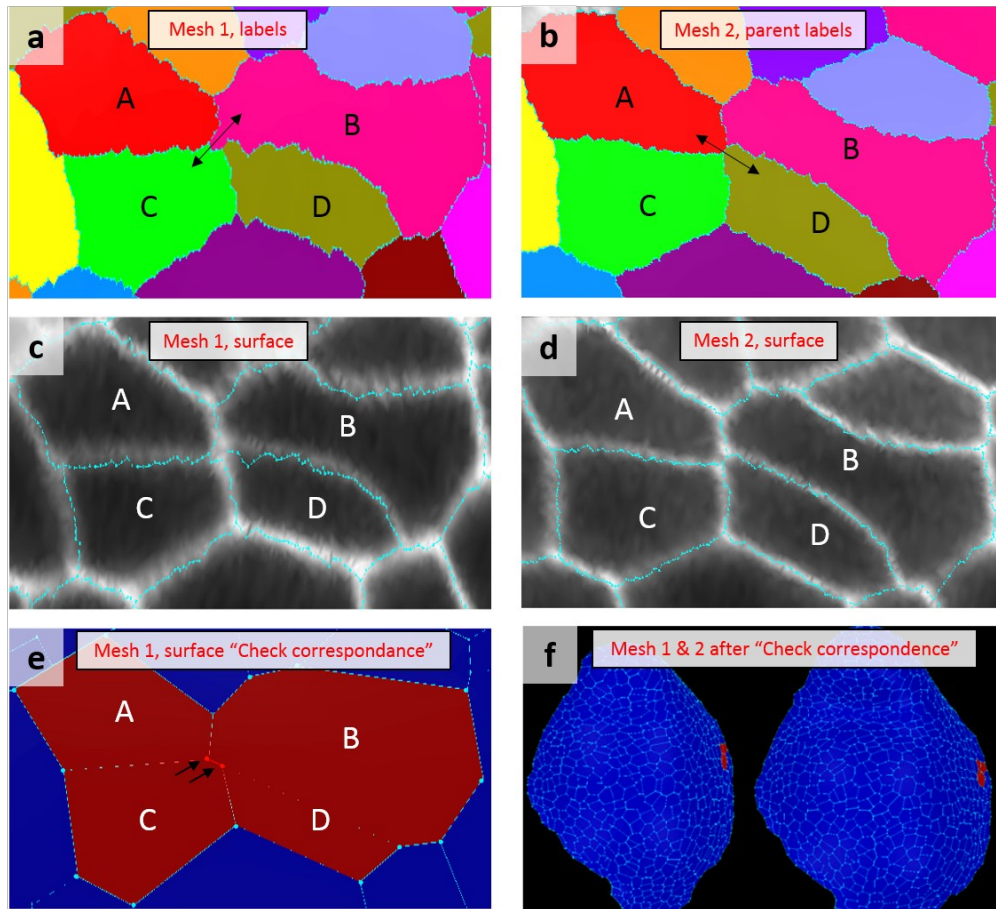


Figure 20. Typical segmentation error found by “Check Correspondence”. Junctions between cells are recognized in both meshes based on the identity of cells in contact. If two junctions are very close to each other, it can happen that a small segmentation error lead to an “exchange” in neighbors. For example in Mesh 1 (a) the cells B and C are in contact, forming 2 junctions (C,B,A) and (C,B,D). In Mesh 2 (b), the cells A and D touch each other, forming the junctions (A,D,C) and (A,D,B), which do not exist in Mesh 1. An inspection of the signal projection in Mesh 1 (c) and Mesh 2 (d) reveals that the segmentation is wrong on Mesh 1, due to fuzzy signal in this region. The “neighbor exchange” is clearly made visible after running “Check correspondence”: cells involved are colored in red and the junctions of Mesh 1 that could not be identified in Mesh 2 are selected (arrows) (e). The faulty cells (in red) are easy to spot on the whole meshes (f).

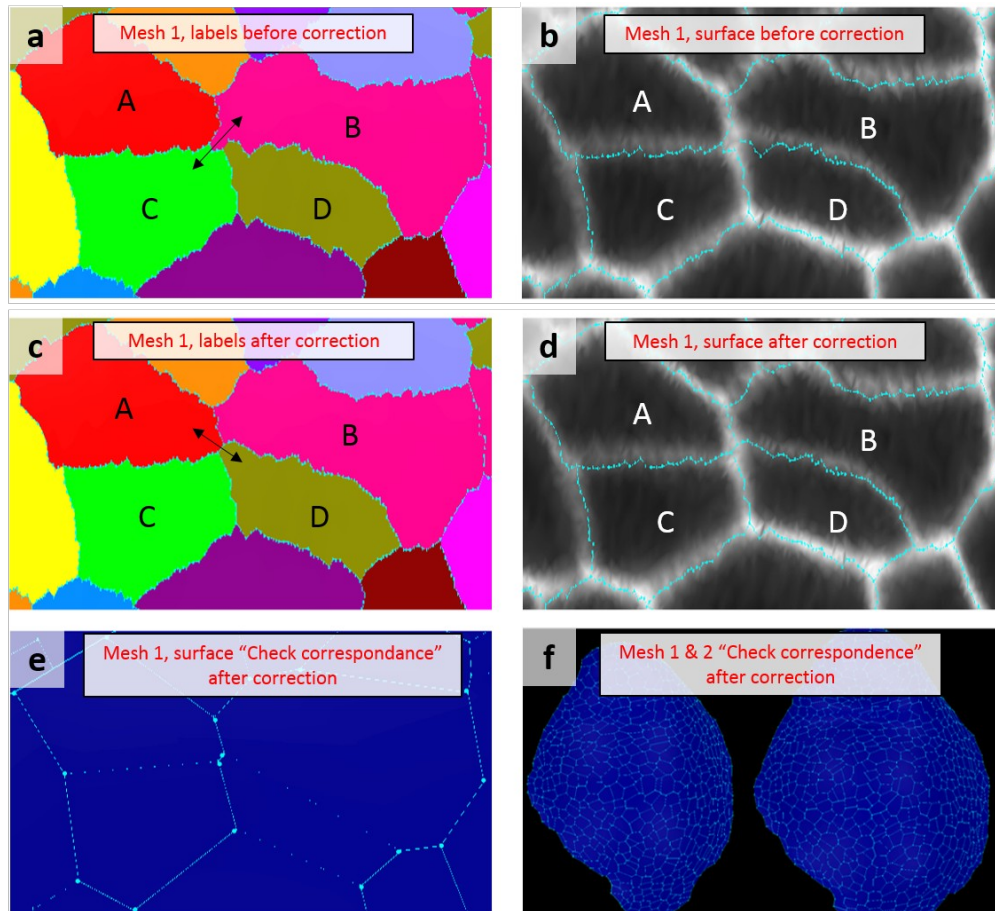


Figure 21. Correction of segmentation error. The “neighbor exchange” in Mesh 1 (a and b) can be fixed (c and d) using the “Pick label” to select label “A” and “D” and “Add current seed” to re-label some of the mesh triangles. Remember to always save the mesh after correction. Re-run “Check correspondence” after correction. The cells should now appear in blue (e). If the entire meshes are free of mistakes (f), you can now proceed to the PDG computation.

Label	xMax	yMax	zMax	xMin	yMin	zMin	stretchRatioMax	stretchRatioMin	stretchRatioNormal
11	0.808674	0.152521	-0.568141	0.417878	0.530822	0.737296	1.36584	1.12149	0
12	0.0459534	-0.429039	-0.902116	0.91003	0.390428	-0.139328	1.18375	1.09646	0
13	0.701228	-0.0755756	-0.70892	0.609485	0.579429	0.541101	1.21568	1.06645	0
14	0.157844	-0.498626	-0.852325	0.92973	0.365856	-0.0418541	1.11782	1.01572	0
15	0.669186	0.549817	0.499891	-0.726385	0.34214	0.596075	1.08283	0.934686	0
16	0.175587	0.718707	0.672778	-0.945426	-0.067433	0.318781	1.17112	1.02809	0
17	0.80365	0.57271	0.161708	-0.521343	0.54652	0.655376	1.13657	1.0241	0
18	0.517941	0.795346	0.314901	-0.82012	0.357042	0.447129	1.14812	0.947165	0
19	0.732752	-0.280129	-0.620163	0.474081	0.863932	0.169908	1.19901	1.05209	0
20	0.222034	-0.95443	-0.199412	0.916084	0.274236	-0.292547	1.05828	0.948829	0
21	0.314793	0.94877	-0.0272171	-0.839277	0.291628	0.458877	1.11244	0.996109	0
22	0.0356987	-0.960362	-0.27646	0.91145	0.144742	-0.385109	1.10891	0.971614	0
23	0.151915	-0.974467	-0.165334	0.879981	0.209515	-0.426305	1.07984	0.990876	0
24	0.357347	-0.922172	-0.147993	0.823839	0.385875	-0.415198	1.10429	0.949434	0
25	0.344349	-0.93732	-0.0534352	0.885577	0.343185	-0.313015	1.101	0.991371	0
26	0.363329	-0.919665	0.149023	0.929467	0.368778	0.00973245	1.14697	0.930028	0
27	0.0769949	-0.942366	0.325604	0.993182	0.0438233	-0.108022	1.23288	1.03614	0
28	0.068294	0.86515	-0.496841	-0.989862	0.120925	0.0745049	1.25662	1.01234	0
29	0.555252	0.567086	-0.608366	-0.793557	0.580179	-0.183461	1.15226	1.06154	0
30	0.725661	0.662623	0.133219	0.579347	0.130916	-0.691658	1.15979	1.11586	0
31	0.248723	0.607983	-0.753985	-0.848309	0.512431	0.133365	1.13804	1.06428	0
32	0.484808	-0.782897	0.389915	0.717225	0.100729	-0.689523	1.13436	1.10419	0
33	0.13745	0.639773	-0.761212	0.968977	0.429743	0.19144	1.05380	1.03540	0
34	0.432946	-0.667783	-0.761981	-0.815889	0.13365	-0.40408	1.13657	1.055	0
35	0.954361	-0.191982	0.0247028	0.0247028	0.70324	1.29113	1.0600	1.0600	0

Figure 22. Example of PDG output file from the “Cell Axis Save” process. The first column of the CSV file gives the label of the cell on the first time point (which is the same as the parent label on the second time point). The columns “xMax”, “yMax” and “zMax” give the (x, y, z) coordinates of the direction (unitary vector) of maximal growth, while the column “stretchRatioMax” contains the magnitude of stretch in this direction. Similarly, “xMin”, “yMin” and “zMin” define the direction of minimal growth, “stretchRatioMin” the stretch ratio. “stretchRatioNormal” gives the value corresponding to the axis perpendicular to both minimal and maximal growth directions. Since we compute only the 2D deformation of the cell in its average plane, this third value is zero.

16. Custom Axis Directions

PDGs (chapter 15) and other cell axis quantifications (such as Shape PCA data from chapter 22 or microtubule directions from chapter 18) are calculated for each cell individually and can point in different directions for different cells. It can be useful to align them into a specific direction, e.g., to measure the cell growth along the proximal-distal axis of an organ. Reoriented axis should be treated with care as this can introduce artifacts into the data. For instance, a rotation of the initial cell axis by exactly 45 degree eliminates any difference between the max and the min axis and therefore the anisotropy. A custom axis alignment can be computed using the following steps:

Create or load any cell axis onto your mesh (see the previous section on how to do that for the PDGs). The original cell axis will be deformed into the desired directions and need to be present for this calculation.

Create the custom directions. There are different ways to create custom directions: either by using a Bezier line/grid or by using a distance heat map (see Figure 23). Both methods are described in the following points, but only one method is required to create the custom directions. The Bezier can be preferred when the tissue has a simple topology or when the organ axes follow a fix direction (e.g., a central line through the organ). For tissues with a higher curvature, the creation of custom directions using a cell distance heat map can be an easier solution. Nevertheless, the next two points describe both methods. Once custom directions are created, they are stored in the attribute map “Mesh Label Tensor Custom Directions”.

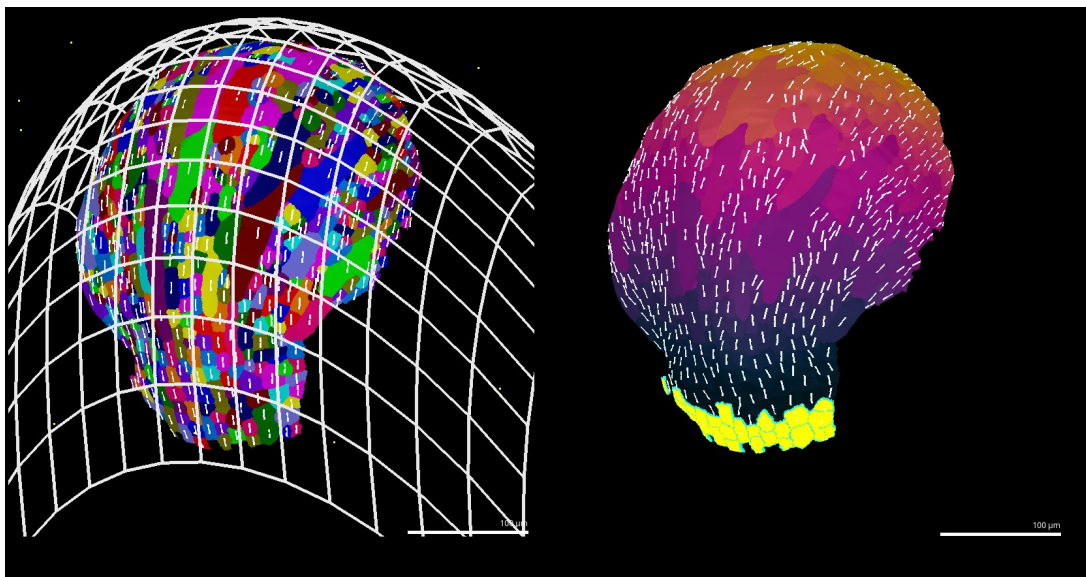



Figure 23. Creation of custom directions using a Bezier grid (left side) or using the gradient of a cell distance heat map (right side, selected cells in yellow).

16.1. Custom directions with Bezier

Create custom directions using the Bezier by going to the “View” tab under “Cutting Surface”, select “Draw”, “Grid” and “Bezier” and click the “Reset” button and align it with the mesh by moving the control points to suitable locations. Once the Bezier is created, it should be visible as a white grid and yellow supporting points on the screen (make sure “Draw” and “Grid” in the “View” tab are selected). For creating a Bezier grid also the process “Misc/Bezier/New Bezier” can be used.

To change the shape of the Bezier surface: (a) select the supporting points with the “Select points in Mesh” tool  by drawing a rectangle around the points while pressing the Alt-key and the mouse left-click button; (b) drag the selected points with the mouse to the desired location by pressing Alt-key and the mouse right-click button. As the Bezier grid is located in the Cutting Surface, it is also possible to rotate and translate it independently from the meshes using the Control-Key and the mouse while having selected the “Cutting Surface” in the “Control-Key Interaction” menu in the “View” tab. The alignment of the Bezier grid should be as parallel to the mesh as possible. Once it is aligned, run the process “Mesh/Cell Axis/Custom/Create Bezier Grid Directions”. Confirm the presence and orientation of custom directions by using “Mesh/Cell Axis/Custom/Display Custom Directions”, select “X”, “Y”, “Z” individually to confirm their directions.

Note: It is also possible to transform the Bezier grid into a line (using the process “Misc/Bezier/Collapse Bezier Points”). Custom directions can then be generated using “Mesh/Cell Axis/Custom/Create Bezier Line Directions”

Tip: Only the the “Grid” checkbox is checked the Bezier is visible and it’s supporting points can be selected and moved. However, then the mesh vertices can’t be selected. To select mesh vertices again, turn of the “Grid” checkbox.

16.2. Custom directions using a distance heat map

Create custom directions using a distance heat map. The directions that are generated will follow the “heat flow” through the tissue. A way to create a continuous heat map throughout the tissue is the Cell Distance Measure (see also chapter 23 “Organ-centric coordinates systems”). It calculates the shortest path from a cell to a selected cell. To create it, select some cells at the base of the organ and run the process “Mesh/Heat Map/Measures/Location/Cell Distance” with the “Wall Weight” parameter set to “Euclidean”. Now run the process “Mesh/Cell Axis/Custom/Create Heatmap Directions”.

Re-align the previously created cell axes using the custom directions. Go to the PDG “Display” process “Mesh/Cell Axis/PDG/Display Growth Directions” and select any “Custom” option for “Heatmap” (e.g., “StretchCustomX”) and/or “Show Axis” (e.g., “StrainCustomX”) to visualize the cell axis properties in the custom direction. It is also possible to calculate the angle between the original and custom cell axis using the process “Mesh/Cell Axis/Custom/Custom Direction Angle”.

Note: When creating the Bezier from a distance heat map, locally directions can appear more noisy, depending on cell sizes. It is possible to smooth local noise using the process “Mesh/Cell Axis/Custom/Smooth Custom Directions”.

17. Mesh Deformation and Growth Animation

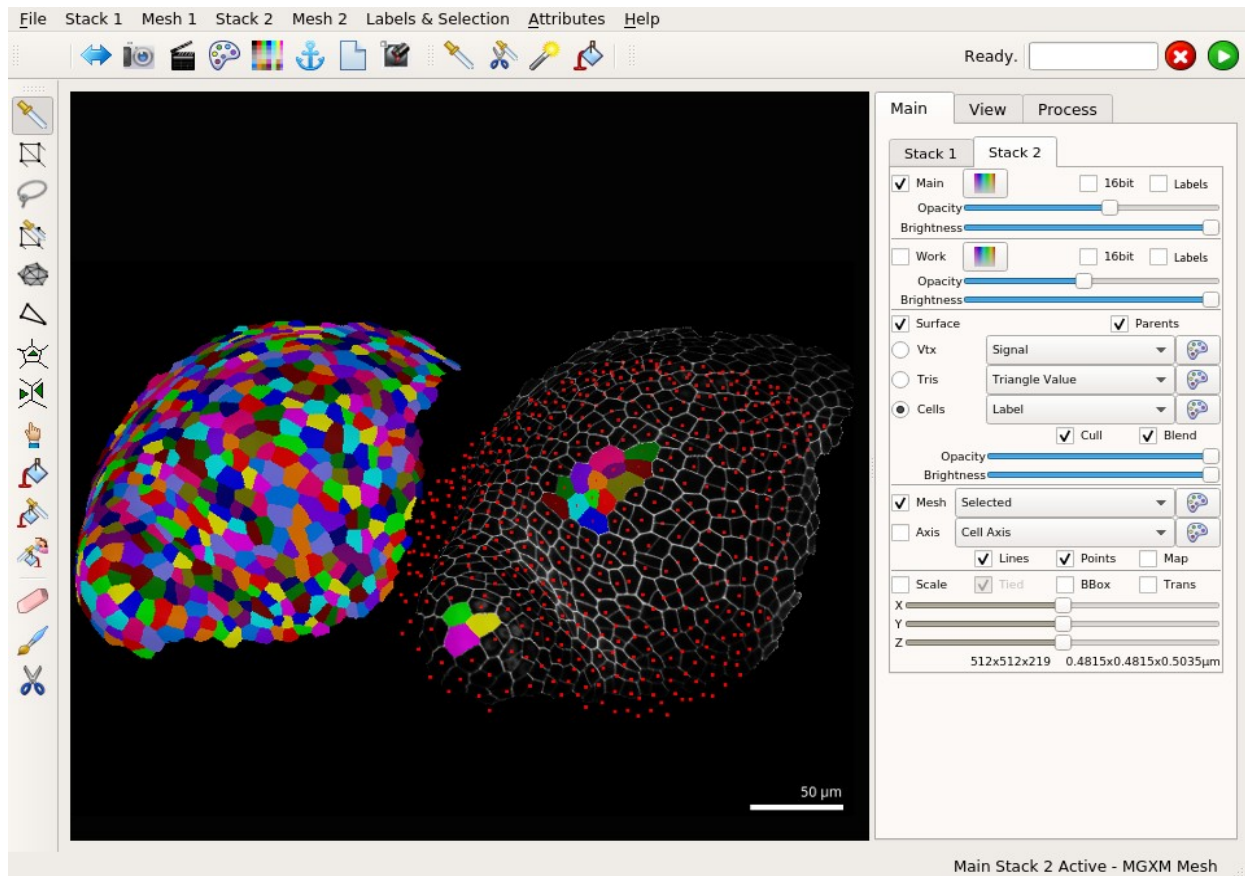


Figure 24. Mesh 1 (left) projected onto Mesh 2 (right) using the parent labeled cell centers in Mesh 2 as landmarks. The red points on Mesh 2 are the projected cell centers. They match exactly for the landmarks, but can deviate when moving away from them.

MorphoGraphX 2.0 introduces deformation functions. Given a set of landmark points in two different meshes, a deformation function is able to map any points of Mesh 1 onto Mesh 2. Landmarks are defined by cell and parent labels and can be cell centers and/or junctions of the cells.

Deformation functions typically have a good accuracy for points close to the provided landmark points and a lower quality further away from landmarks. When creating a deformation function the best results are obtained by providing as many input points, as close to the regions of interest, as possible. However, having too many points can make the creation of the deformation function slow.

Deformation functions have several applications in MorphoGraphX:

- Semi-automatic lineage tracking (see chapter 17.2).
- Mesh mapping and morphing (see chapter 17.3 and 17.4).

- Quantifying subcellular growth (see chapter 17.5).
- 3D PDGs (see chapter 21.4).

17.1. Creation of Deformation Functions

To create a deformation function on a 2.5D segmented surface mesh, perform the following:

Load the two segmented meshes that need to be mapped into Mesh 1 and Mesh 2. In the following, Mesh 1 is assumed to be the active mesh. Mesh 1 has to be the earlier/"parent" mesh while Mesh 2 is the later mesh with the parent labels.

Load or create a parent labeling in Mesh 2. Associations between the points (vertices) of different meshes are created using landmarks. In MorphoGraphX, landmarks are identified based on the cell and parent labels (e.g., if a cell x in Mesh 1 is equivalent to a cell y in Mesh 2, then the parent label of cell y has to be set to x).

Once an initial parent labeling is present, run "Mesh/Deformation/Mesh 2D/Create Deformation" process:

You can specify the direction of the deformation ("active \rightarrow other" or the other way around) and which landmark points based on the parent labeling will be used: cell centers, cell junctions or both. Cell junctions are automatically found based on the cell parents and are only used if they are uniquely defined, that is junctions between cells that are all parent labeled.

The parameter "Junction Merging Threshold" specifies the minimum distance between two junctions. If two neighboring junctions are too close they will be merged into a single point to prevent the creation of local artifacts or instabilities due to noise.

Note the CG step (Conjugate Gradient step) parameter, which specifies how the system of equations that define the deformation will be solved. A value of "-1" creates an exact solution (the system will be solved directly). A value ≥ 1 will create a conjugate gradient approximation of the solution with the number of steps specified. Generally, using "-1" is recommended, unless the solver fails or takes too long to compute a solution. In such cases a CG step between 5-50 are usually sufficient to produce a good approximation of the deformation.

Note: Solving time depends on how many landmark points (centers and/or junctions) exist, and may take from a few seconds for small meshes (<500 landmarks) to several hours for (>3000 landmarks).

Once computed, deformations can be exported to a file and an attribute map using the process "Mesh/ Deformation/Deformation Save or Export" or imported using the process "Mesh/Deformation/Deformation Load From File". Now the deformation is created and saved into the attribute map "Implicit Deformation" of Mesh 1.

Note: Deformation Save or Export can save the deformation function to a file or export it to an attribute map with a user defined name. Deformation Load From File always loads (and overwrites) a deformation into the attribute map “Implicit Deformation”. If you have created a deformation in the direction “other→active” you will have to save it to a CSV file first and then load it into the other mesh before it can be used in the other direction.

17.2. Semi-Automatic Parent Labeling

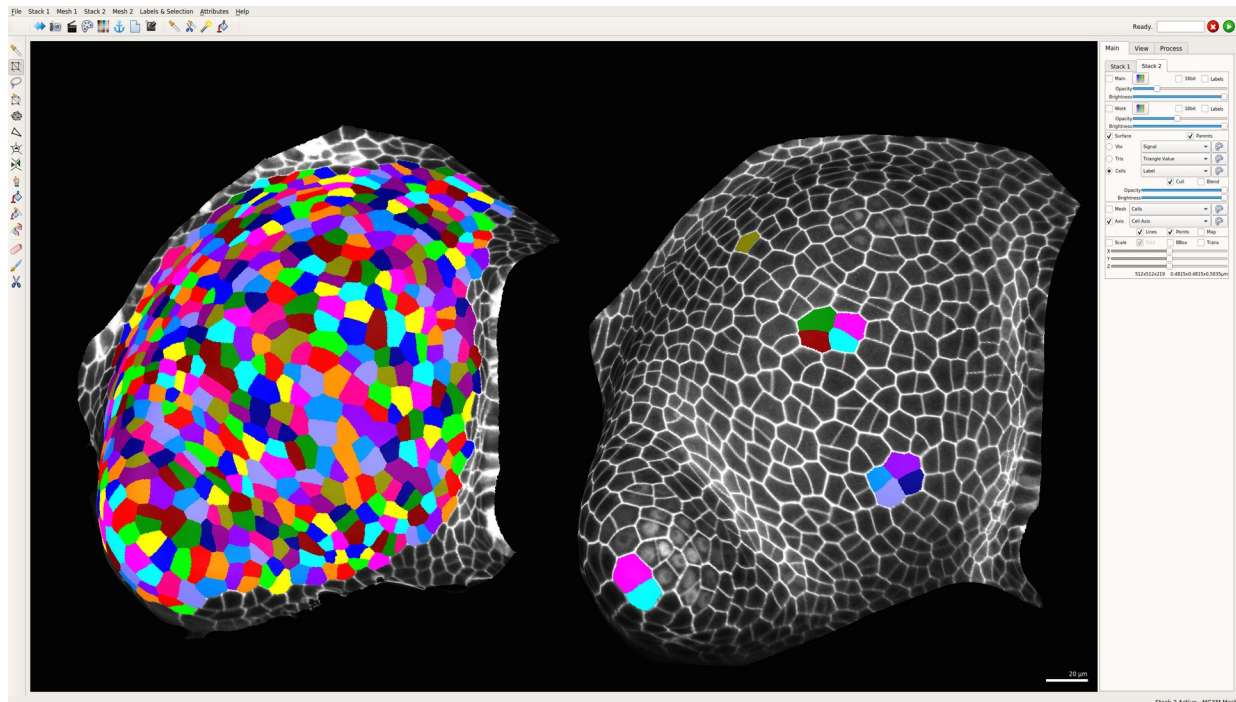


Figure 25. Tomato meristem. Time point 1 segmented (left) and time point 2 (right) with a few (manually) parent labeled cells and a projection.

Using deformation functions, it is possible to automatically parent label a mesh using just a few manually specified correct parent labeled cells:

Load the two time points of your meshes into Mesh 1 (parent, earlier time point) and Mesh 2 (daughters, later time point).

Manually parent label a few cells, preferably located at the extremes of the mesh (local maxima, saddle points, cells along the border of the mesh). Then run the process “Mesh/Deformation/Set Correct Parents” with Mesh 1 active, which creates an attribute map with the manually assigned parents and marks them as “correct”.

Run the process “Mesh/Deformation/Mesh 2D/Auto Parent Labeling 2D” which repeatedly calls the process “Mesh/Deformation/Mesh 2D/Auto Parent Labeling Step 2D” to do one step of the labeling. In each step the following happens:

A deformation function using the current parent labels is created. The deformation is used to map Mesh 1 onto Mesh 2.

New parents are searched in the direct neighborhood of already parent labeled cells. Therefore, locations of either cell centers (in mode Find Cells = “Simple”; finds only cells that haven’t divided) or junctions (Find Cells = “All”; finds cells that have or have not divided) of Mesh 1 are mapped onto Mesh 2. The best fitting cells are parent labeled.

The new parent labeling is tested for errors by running the process “Check Correspondence”. Mis-annotated parent labels are reset (with the exception of manually fixed parents and parents that are at least “Auto Fix Parents” away from the current border) and another iteration starts.

Note: The processes “Mesh/Deformation/Mesh 2D/Auto Parent Labeling 2D” and “Mesh/Deformation/Mesh 2D/Auto Parent Labeling Step 2D” call the process “Mesh/Deformation/Create Deformation Cell Surface” with the specified parameters there.

Note: If the quality of the found parent labels is poor, consider providing more initial parent labels. You can also use the processes “Mesh/Deformation/Mesh 2D/Deform Mesh” and “Mesh/Deformation/ Mesh 2D/Mesh Projection” to verify the quality of the deformation function.

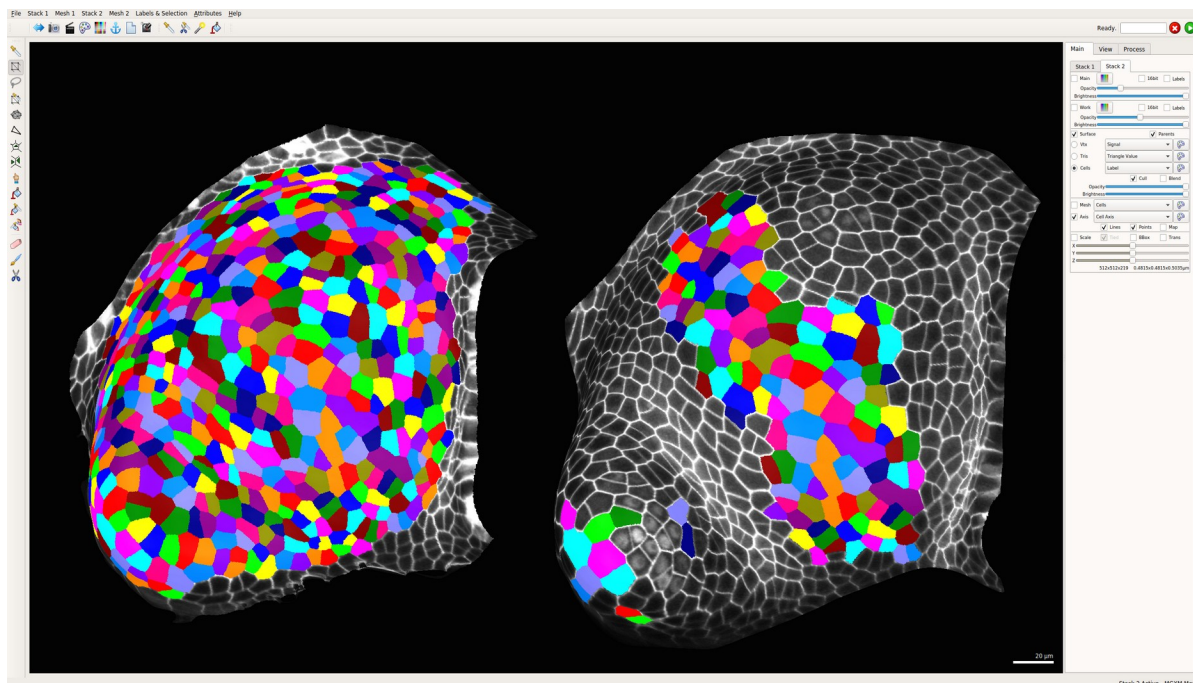


Figure 26. Tomato meristem after a few steps of automatic parent labeling.

17.3. Mesh Mapping or Deformation

A mesh can be mapped onto a target mesh using a deformation function:

Create or load a deformation function (see section 17.1).

Project the mesh onto a target mesh. Run the process “Mesh/Deformation/Mesh 2D/Mesh Projection”. This process creates new vertices in the other mesh based on the chosen parameters (cell centers or junctions) which allows a direct comparison of mapping and target mesh. All new vertices are selected and can therefore be deleted easily (delete key when Mesh 2 is active) to restore the initial state of Mesh 2.

A mesh can be deformed into a target mesh using a deformation function:

Create or load a deformation function (see section 17.1).

Deform the mesh into a target mesh. Run the process “Mesh/Deformation/Mesh 2D/Deform Mesh” to apply the currently present deformation in the “Implicit Deformation” attribute map of the active mesh.

To reset the deformation applied using above process, run the process “Mesh/Deformation/Reset Mesh”.

17.4. Morphing Animations

Morphing animations from Mesh 1 to Mesh 2 can be created by first computing interpolated steps between Mesh 1 and its mapping. All necessary processes are located in the folder “Mesh/Deformation”. To create a mapping or a morphing follow these steps:

Import one or more implicit deformations created as described in section 17.1. Typically the deformation is loaded from a file using the process “Deformation/Deformation Load From File” and then renamed by using the process “Deformation/Deformation Save or Export” by exporting it to an attribute map with a name of your choice. Make sure to give the attribute maps different and appropriate names (e.g., “DefT0T1”, “DefT1T2”).

Create the morphing. Once imported into attribute maps, run the process “Deformation/Morphing/Create Morphing” after specifying the names of all attribute maps of the deformations that are to be used for the morphing. This process will create additional attribute maps of the positions of all vertices after the deformations have been applied.

Remove undesired rotations or translations. Depending on the relative orientation of Mesh 1 and Mesh 2, a deformation function might introduce undesired rotations or translations. Such effects can be minimized by the following:

Run “Deformation/Morphing/Set Origin” to select cell/vertex as origin.

For a constant axis, select a cell/vertex and run the process “Deformation/Morphing/Set Axis 1” and “Deformation/Morphing/ Set Axis 2”. These processes will find the normal of the selected cell/vertex and keep them fixed in respect to the viewing position.

Manually orient and position each mesh and save them. When saving, remember to check “Transform” in the saving dialog window (before creating the implicit deformation).

This method is preferred when creating animations involving more than 2 meshes.

Load signal or heat values onto the mesh. It is possible to load signal or heat values for each individual time points. The values will be blended in the morphing animation (see chapter 17.5)

Save the mesh before starting the morphing animation. Note that the mesh file size can increase substantially (depending on the number of vertices and deformations applied). To start the animation run the process “Deformation/Morphing/Run Morphing”. Parameters control:

The number of intermediate interpolated steps.

The pause duration after every step.

Whether or not a heat map is visualized (See also 17.5)

Whether a spline function is used to interpolate positions (produces smoother animations when running more than one deformation).

Whether mesh signal is blended (requires a “Signal Projection” for each deformation, see 17.5).

Note: For large meshes, it is not necessary to add a pause between steps as the computations can take a long time. It is, however, advisable to record a video to get screenshots of every animation step.

Note: For creating animations across multiple time points with meshes of different sizes it is advisable to run the morphing backwards (from the largest to the smallest) and reverse it for the video, as the meshes of the later time points consist of more vertices which increases the overall quality.

Warning: Deleting vertices from the mesh will invalidate the signal projection attribute map. If vertices are eliminated from the mesh, “Mesh/Deformation/ Morphing/Create Morphing” and all advanced features (see below) must be re-run.

17.5. Advanced Features for Morphing Animations

Signal Projection

The signal of a target mesh can be projected on the active mesh using deformation functions. In the following, instructions for a single deformation function are provided:

Create or load a deformation function (see section 17.1):

Create morphing to gradually change a mesh into the other mesh. Run the process “Mesh/Deformation/Morphing/Create Morphing”. (Specify the name of the deformation function attributes here)

Load the target mesh.

Load parent labels in the active mesh (active → target).

Create a signal projection. Run “Mesh/Deformation/Morphing/Create Signal Projection”, set the value of “Apply Deformation” to the index of the deformation that maps the active mesh to the target mesh (*i.e.*, the corresponding parameter in “Mesh/Deformation/Morphing/Create Morphing”). The signal projection will be stored as an attribute map in the active mesh.

For multiple deformation functions, the preceding actions must be repeated for each deformation. Note: Deleting vertices from the mesh will invalidate the signal projection attribute map. If vertices are eliminated from the mesh, “Mesh/Deformation/Morphing/Create Signal Projection” must be re-run.

Heat Maps

It is possible to visualize heat maps on a morphing animation via the “Display Heat Map” parameter in the “Run Morphing” process. Some heat maps can be computed “live” during the animation on the individual interpolation steps. It is also possible to pre-load heat maps and blend between them (similar to the signal projection) using the process “Create Heat Projection” and the steps above for the Signal projection (“Create Heat Projection” replaces the “Create Signal projection” process in the last step.

17.6. Growth Analysis using Deformation Functions

The derivatives of deformation functions contain information on how strong the deformation is at any given point in any direction. This information can be used to deduct information about the stretch and the strain of the mesh, similar to the computation of the PDGs (see chapter 15). Due to the continuity of the deformation function, growth information can be computed for any point of the mesh, which enables the computation of sub-cellular growth as well as growth of cells.

First, create a deformation function using “Mesh/Deformation/Mesh 2D/Create Deformation Cell Surface” or “Mesh/Deformation/Mesh 3D/ Create Deformation Cell Volumes”, depending on your mesh type.

For 2D meshes, we strongly advise to use cell junctions as material points. A deformation function of the cell centers will cause a smoothing, which decreases the level of detail. (see figure 27).

Depending on noise and the quality of the segmentation, junctions that are too close should be merged. Use the “Junction Merging Threshold” parameter appropriately (see also above).

Once the deformation function is computed, run the process “Mesh/Cell Axis/Deformation Gradient/Compute” (for 3D meshes, go to the “Mesh/Cell Axis 3D” folder). This process has parameters to choose between growth on the vertex level (sub-cellular) or growth on the cell

level. The growth tensor is always calculated in the 3D space. However, for 2.5D meshes the 3D tensor should be projected onto the mesh surface to capture the amount of surface growth appropriately.

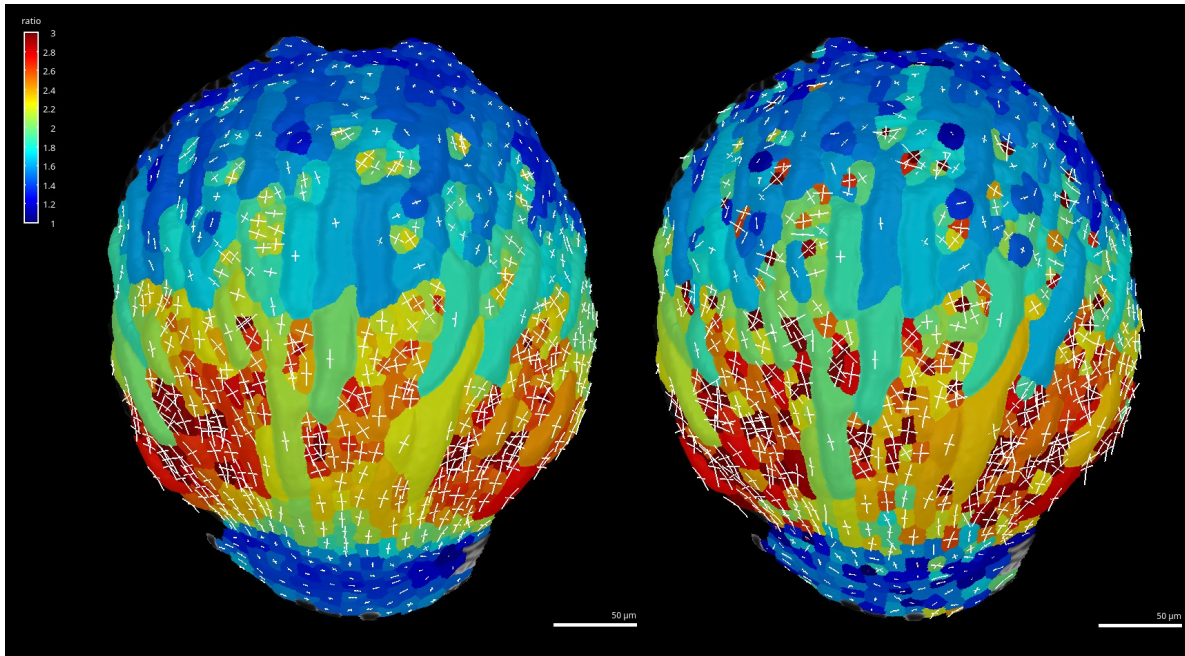


Figure 27. PDGs created from deformation functions (the Hervieux_2016 data set). Left side: Using cell centers as landmarks. Right side: Using cell junctions as landmarks. The heat map on the right shows more detail on a single cells level (e.g., stomatal growth).

With the calculated growth information, the visualization can be changed using the processes “Mesh/Cell Axis/Deformation Gradient/Display Cell Gradient” and “Mesh/Cell Axis/Deformation Gradient/Display Vertex Gradient”.

Note: In the case of 2.5D meshes, we advise to remove potential bulges in the cells prior to the growth analysis using deformation functions. For this, use the process “Mesh/Cell Mesh/Tools/Flatten Mesh” (maybe followed by a few steps of “Mesh/Structure/Smooth Mesh” with only vertices in the inside of cells selected using “Mesh/Selection/Select Special”, only cell vertices = Yes). This process keeps the cell borders in place but projects and smooths interior vertices.

As the growth information is projected onto the surface, local surface topology can influence the growth significantly. As the growth data typically is extracted only using cell junctions, the interior cell surface should be on the same “height level” as the cell junctions (see Figure 28).

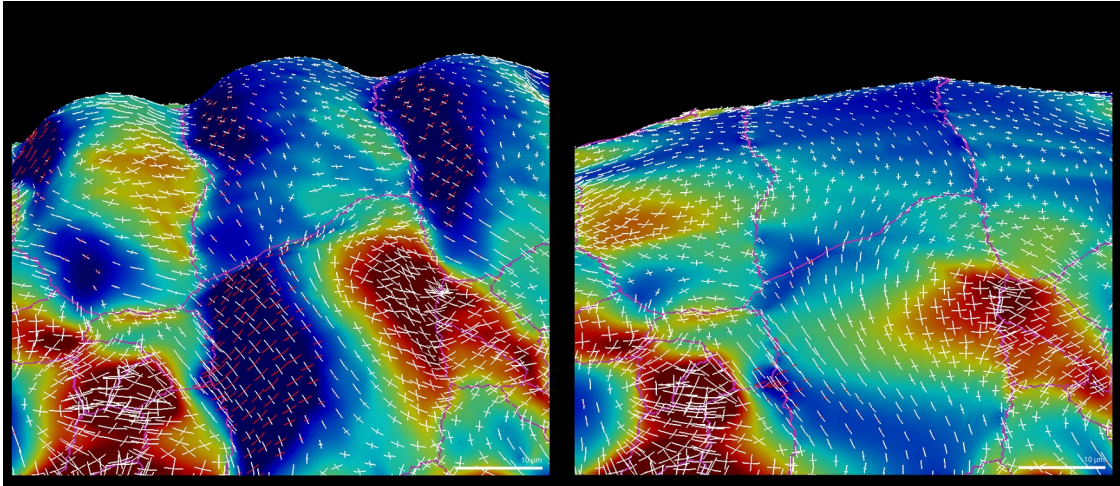


Figure 28. Vertex PDGs on the sepal of the Hervieux_2016 data created from the same deformation function. Left side: Some cells of the initial cells with strong bulging. Right side: The same cells after using the process “Mesh/Cell Mesh/Tools/Flatten Mesh”. Bulges can strongly alter the growth rates on the vertex level and should always be removed when quantifying growth using deformation function on surface meshes.

18. Quantifying signal orientation

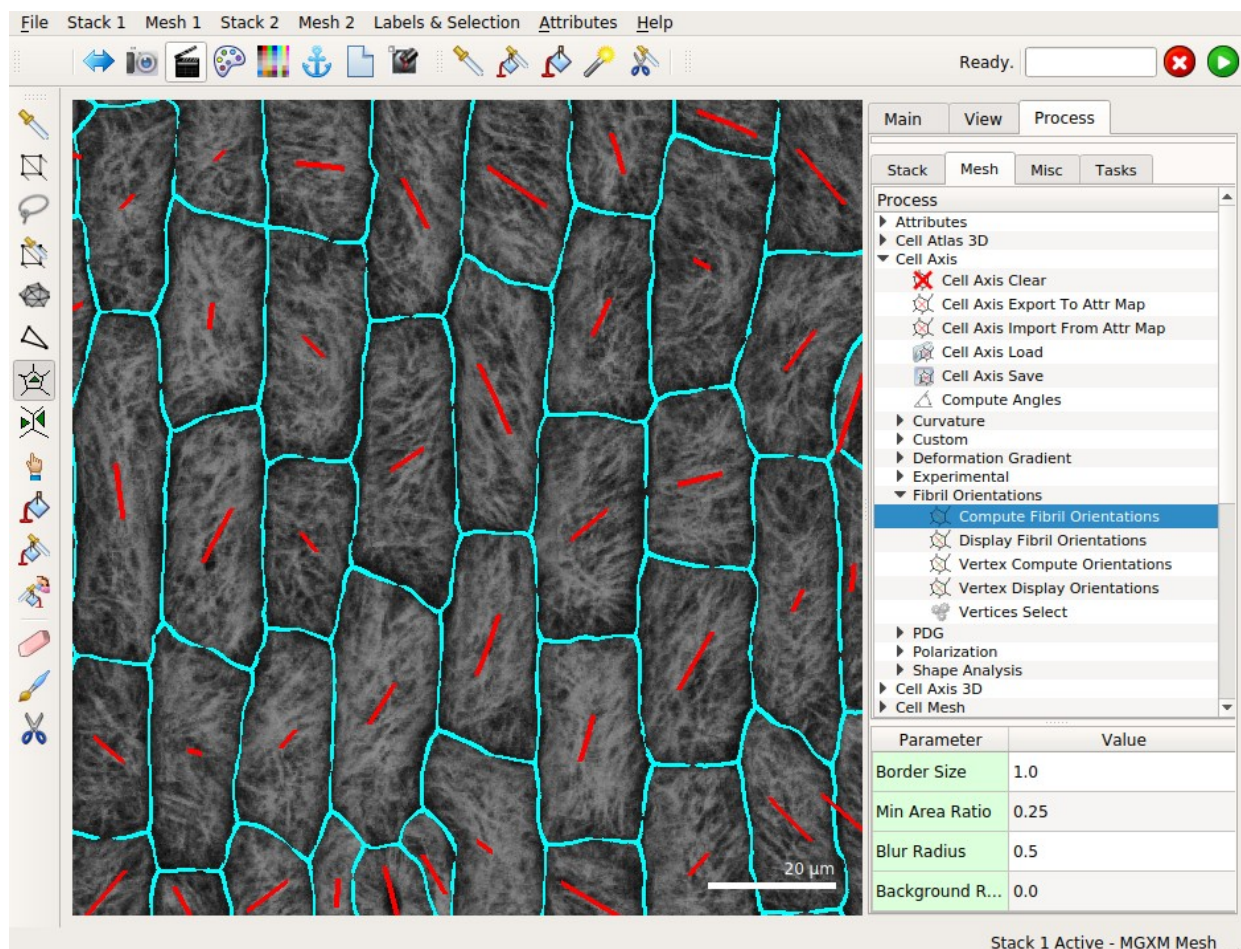


Figure 29. Quantification of microtubule orientation.

18.1. Fibril tool

The Cell Axis can also be used to quantify the orientation of cortical microtubules (MT). The vectors of average orientation are stored in the same way as the PDGs. The angle detection is based on ImageJ plugin FibrilTool (Boudaoud *et al.*, 2014). If you have a good fluorescent marker of MTs (tubulin), it will be sufficient for both segmenting the cells and quantifying MT orientation.

Create a mesh as described in the previous sections.

Segment the mesh. Project signal from deeper regions of the stack (up to 10 μm) and perform the segmentation as described above.

Project MT signal on the segmented mesh. This time, use signal that is closer to the surface (try between 1 and 5 μm until you get the best possible resolution of MTs).

Quantify orientation per cell. Run the process “Mesh/Cell Axis/Fibril Orientations/Compute Fibril Orientations”. The 'Border size' parameter defines the area around the cell wall that should not be considered for the quantification. “Min Area Ratio” indicates the threshold below which cells will not be quantified (excludes very small cells). The orientation vectors will be stored in the Cell Axis which means you can save, load and visualize them the same way as the PDGs. This quantification is performed for the whole cell. Visualization parameters can be changed using the process “Mesh/Cell Axis/Fibril Orientations/Display Fibril Orientations”.

Quantify orientation on a vertex level. It is possible to quantify the MT orientation in an area around selected vertices (and not needing or ignoring the cell segmentation). For this, you first need to select the vertices of the region of interest. You can also use the process “Mesh/Cell Axis/Fibril Orientations/Vertices Select” to automatically select random vertices within a certain distance from each other and the cell borders. Once you have all vertices of interest select, run the process “Mesh/Cell Axis/Fibril Orientations/Vertex Compute Orientations”. Finally you can change visualization parameters using the process “Mesh/Cell Axis/Fibril Orientations/Vertex Display Orientations”.

18.2. Signal polarity

Similar to the quantification of microtubule orientation (see previous section) it is also possible to quantify the overall polarity of a signal projected onto a segmented surface mesh. This pipeline is meant to be used for signals that are located at the cell border, such as PIN-FORMED (PIN). Like for the other cell axis tools there are distinct processes for “compute” and “display” in the “Mesh/Cell Axis/Polarization” folder.

The “Compute” process creates equally size angular bins originating from the cell center and computes the average signal in those bins. The bins are then used as input for a Principal Component Analysis (PCA) to determine amount and direction of maximum and minimum signal intensity at the cell border.

There are two parameters for the “Compute” process:

- Border Size: indicates the size (thickness) of the cell border, only signal at the cell border will be used for the quantification
- Minimum inner are: indicates the minimum size of the inner area (cell area – border area), relative to the total cell area. Small cells (that would only exist of border area) are then ignored.

The “Display” process contains the usual options for cell axis (see also previous section about PDGs or Fibril orientations).

19. Improving the stack signal using deep learning

Recently, deep learning pipelines that are able to improve and de-noise confocal images have become popular. For instance PlantSeg can be used to create cell wall predictions from a raw confocal input and further segment it into cells (Wolny et al., 2020). PlantSeg's output file can readily be loaded into MorphoGraphX. MorphoGraphX also implements an interface to the pipeline using Convolutional Neural Networks (CNNs) from (Eschweiler et al., 2019) with the process "Stack/CNN/CNN Prediction".

Upon running the process a network for the prediction can be chosen. 3 different networks are part of the process and can be readily used. Depending on the size of the input stack and the GPU of the workstation this process can run from seconds to several minutes.

20. 3D segmentation

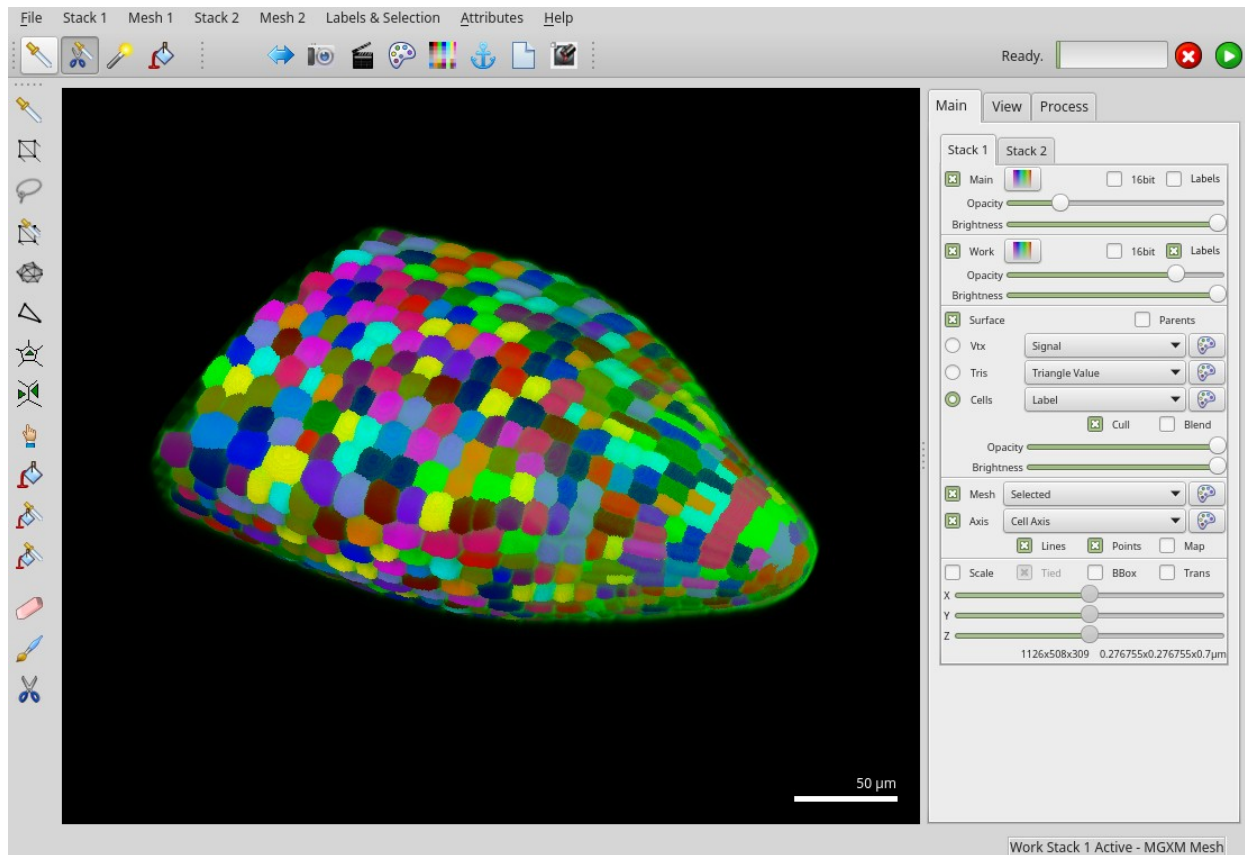


Figure 30. 3D segmented stack of the Bassel_2014 sample data.

With confocal stacks of good quality, it is possible to fully segment cells in 3D. Best results can be achieved with very dense stacks (fine Z step) and a small pinhole diameter. Consider also using a deep learning pipeline to further improved the cell wall signal in the stack (see also chapter 19). Since it is particularly difficult to seed in 3D, MorphoGraphX uses an auto-seeded watershed from the C++ library called the “Insight Toolkit (ITK)”. It is also possible to start with a segmented stack that was created using other segmentation software platforms.


To do the 3D segmentation perform the following steps:



Load the stack. You can use the example data (radicle of a mature *Arabidopsis thaliana* embryo) from the “Bassel_2014” folder provided on the www.MorphoGraphX.org website.

Blur to reduce noise. Run the process “Stack/Filters/Gaussian Blur Stack”. Use a radius slightly larger than the width of the cell walls in the sample. In this example, try 1 μm .


Segment. Run “Stack/ITK/Segmentation/ITK Watershed Auto Seeded”. This can take some time, but you do not have to do any seeding. Adjust the threshold according to the results of the segmentation (here, try 3000). If a sample is highly under-segmented (cells fused), then

decreasing the threshold is needed. If the sample is over-segmented (single cells segmented into multiple pieces), then the threshold needs to be increased.

Delete the outside. The segmentation fills the entire volume with labels and the sample will be buried inside somewhere. To remove the outside label, select the “Delete Label in Volume” tool  from the Volume tool bar. First save the segmented volume (in the Work Store) under the Stack menu as there is no undo function.

Correct the over-segmentation errors. The auto-seeded segmentation almost always requires correction for over-segmented cells. These multiple segments can be fused together into a single segment by using the color picker  and bucket  from the Volume tool bar to merge cells. Use the clipping planes under “View” tab (see section 4) to correct for errors inside the sample.

Correct the under-segmentation errors. Fixing under-segmentation errors is also possible for 3D segmentations. To do this, delete first the label of the under-segmented cells. Next, activate a clipping plane and place new cell labels inside of the now empty cells. Finally run the ITK Watershed process again to fill the seeded cells.

Extract the volumes. When you are happy with the segmentation, extract the mesh using the process “Mesh/Creation/Marching Cubes 3D”. The “Cube size” parameter defines how far apart vertices are in the mesh. For a fine and detailed mesh, cube spacing should be smaller (for the sample data try 3 μm and above). The trade-off to this is a larger file size for the mesh. Cells are automatically labeled with the same values as the stack labels. Once the mesh is extracted you can edit each cell individually using the “Select Connected Area” tool . Different volumetric cells are not connected as they do not share vertices (they have duplicated vertices in shared walls instead), so this tool will select only a single cell.

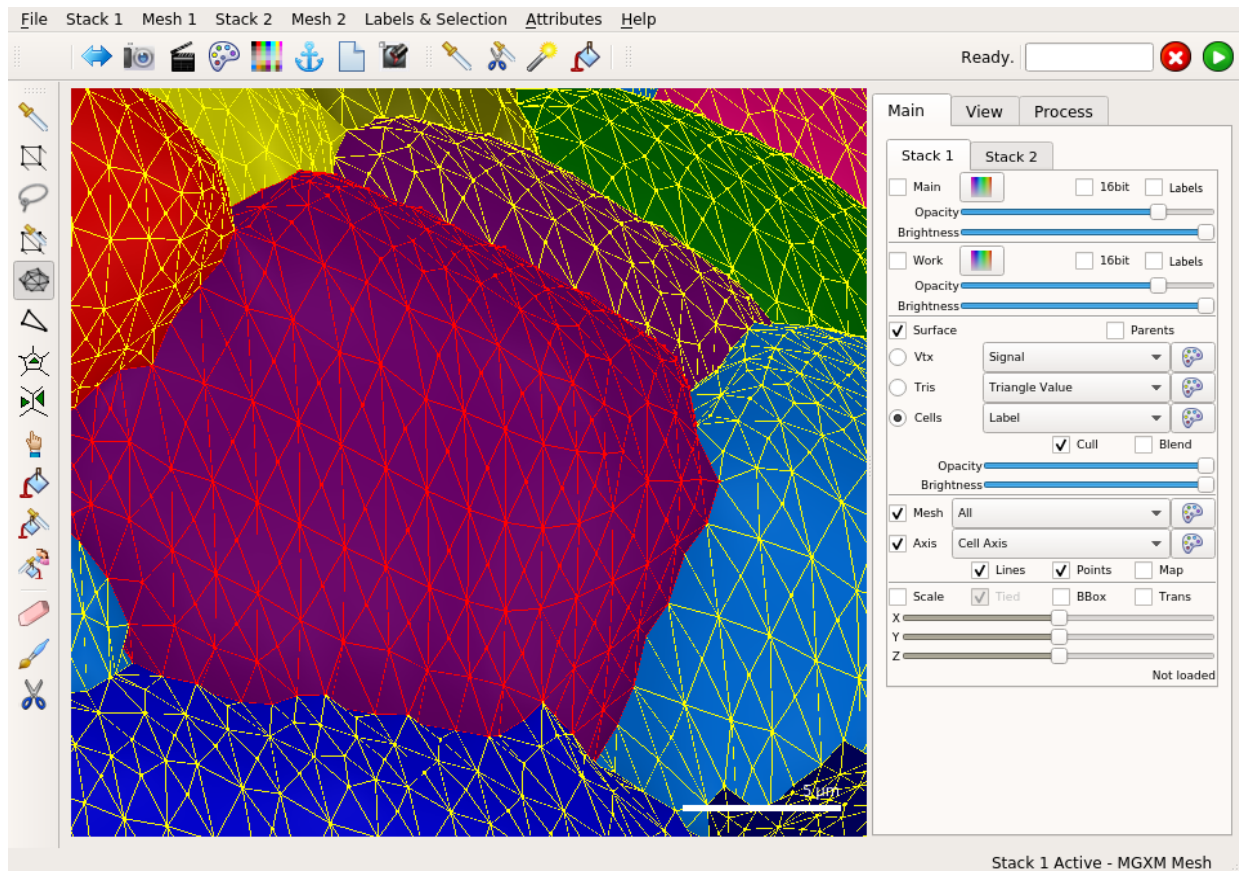


Figure 31. Volumetric cell in a mesh selected by the “Select Connected Area” tool.

To get an optimal 3D segmentation, experiment with different values for the blurring and the threshold parameter for the ITK segmentation to see if you can nicely segment all the cells with as little manual correction as possible. You cannot split cells, so if the stack is under-segmented, then you must re-run the segmentation with a lower threshold. If a stack is highly over-segmented, then extensive manual correction is needed. Optimizing the segmentation in the first instance is the quickest way forward.

Troubleshooting

Symptom: “Error message after running ITK Watershed (*‘Number of objects greater than maximum of output pixel type.’*)”. Increase the threshold to get fewer cells.

Symptom: “A lot of cells in the segmented stack have the same color”. Make sure that the cells are not fused (have different labels) using the “Pick a label from the volume” tool . If you have a lot of cells in your sample (more than 30), check on the “16bit” option on the Work Stack. This will change the label color display. Note that labeled stacks use a different color mapping from normal ones. The numbers representing the voxels are mapped to a color table which is typically 16 colors, and repeats. If the number of labels exceed 16, cells with different labels can be displayed with the same colors. Also the process

“Stack/Segmentation/Correct Labeling Colors” can be used to relabel the stack with distinct colors for neighboring cells.

Symptom: “The software is slow to run after mesh extraction”. Adjust the cube size for the 3D marching cubes to see the effect on the shape extraction of the cells. Do not go too small or it will create too many vertices and be very slow to run. If the cube size is too large then the shapes of the cells will not be accurately captured.

21. Mesh 3D analysis and quantification

Generally, the analysis of 3D meshes is similar to 2.5D meshes. However, many analysis pipelines have separate processes for 3D meshes as the overall mesh structure is different in 3D (volumetric cells) compared to 2.5D (flat cells).

21.1. Heat maps & Measures 3D

Just like for 2.5D meshes, the geometrical (and others) properties of volumetric cells can be quantified and visualized as a heat map. 3D meshes are typically much larger (number of vertices per cell) than 2.5D meshes and computations on them can be more time consuming. For this reason, most measures are created by a single process which creates and saves the appropriate attribute maps: “Mesh/Heat Map/Analysis/Cell Analysis 3D”. This process is very central for 3D meshes as it is also required for most other processes handling cellular information of 3D meshes. For instance, it also computes (and saves) cell coordinates and the cell neighborhood graph.

After running this process you can create heat maps using the processes in “Mesh/Heat Map/Measures 3D/Geometry”. Measures in the other sub folders in “Measures 3D” may require additional input such as the Cell Atlas measures (see section 24) or the Cell Length Custom measures, which requires a previously created custom direction (see section 16).

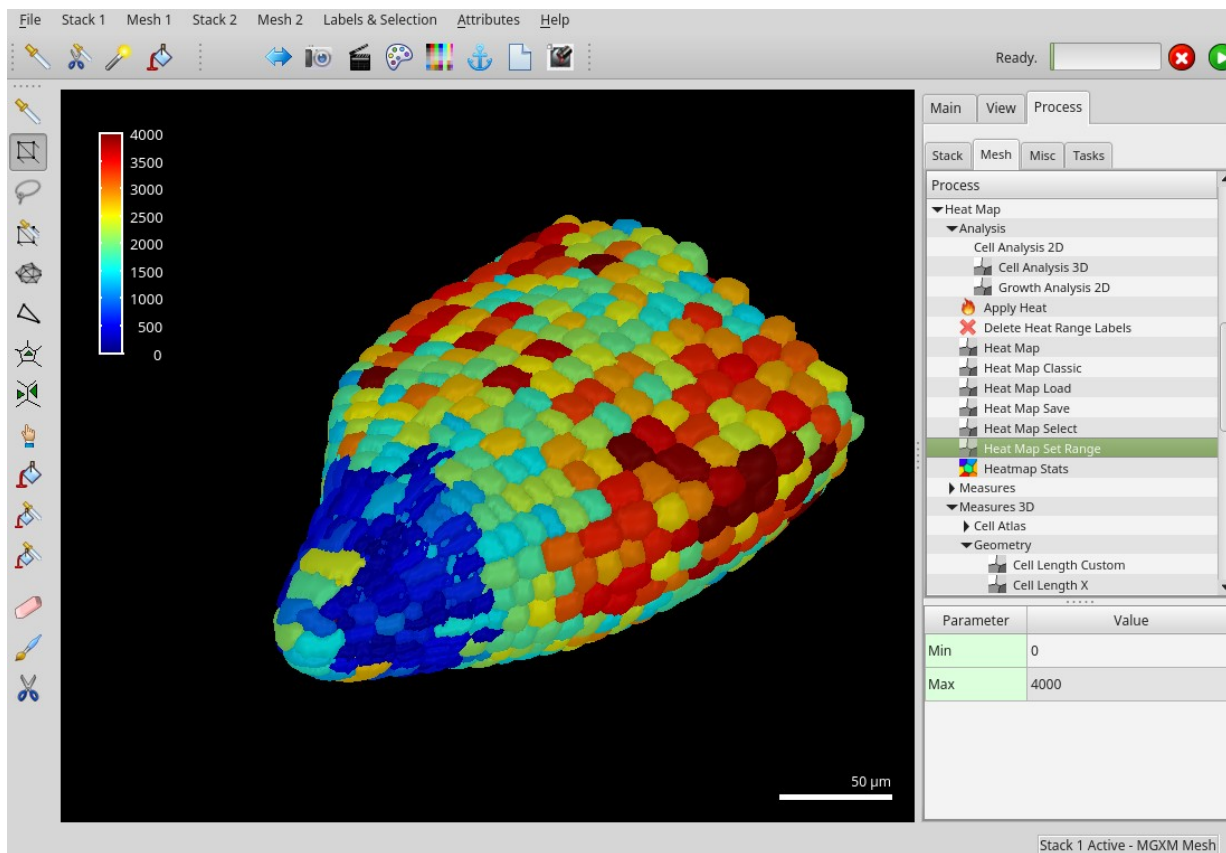


Figure 32. Heat map of cell volumes of a 3D mesh created using the Bassel_2014 data set.

21.2. Cell Lineage Tracking 3D

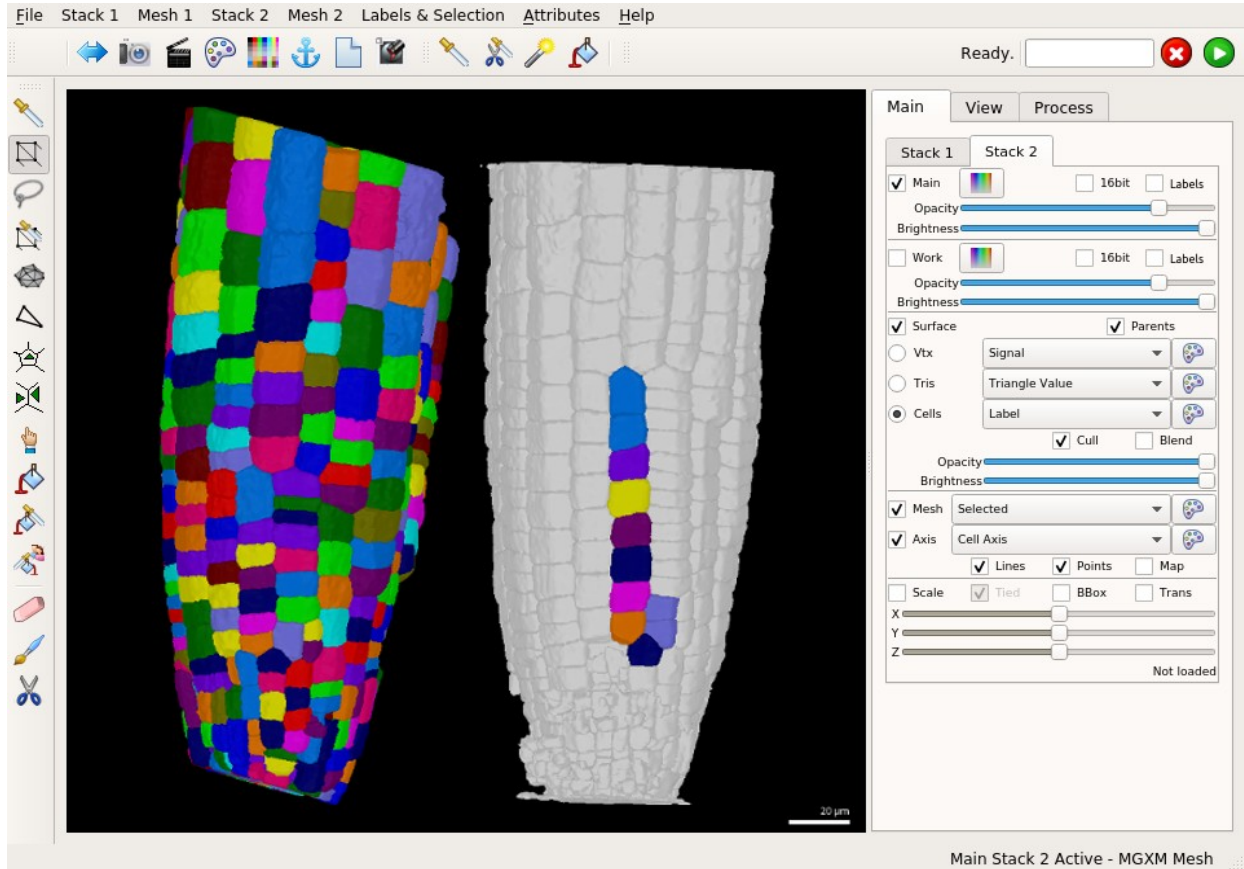




Figure 33. Manual 3D parent labeling using the “Pick Label in Mesh 1 and Fill Parent in Mesh 2” tool.

In section 13, we described the procedure to parent label (or lineage track) cells for surface meshes. This method does not work for 3D meshes. Instead, the “Pick Label in Mesh 1 and Fill Parent in Mesh 2” mesh tool  can be used to rapidly parent label time lapse data.

Open the meshes. We assume that Mesh 1 holds the earlier time point and Mesh 2 the later one. Mesh 2 then has to be parent labeled and the “Parents” checkbox has to be ticked. Also activate .

Fill the parent labels. The tool will toggle the “Stack” tabs and therefore alternate between Mesh 1 and Mesh 2 with each Alt+left-click. If you are already on Mesh 1, it will also pick the label of the cells you are clicking on. Once the label is picked and Stack 2 is active, go to Mesh 2 and with each Alt+right click it will fill the parent cell of the picked label. Use the tool to parent label the outermost layer.

- Tip: Sometimes the tool might get “stuck” and refuses to pick labels. This is a bug which currently has not been fixed. To reset it, it is usually enough to toggle the “Parent” checkbox On/Off and continue as before.

Go to the next layer. Once the outermost layer is parent labeled, those cells obstruct the view. You can now either use clipping planes, delete the outside layer or activate an exploded view before continuing with the deeper layers. Make sure you save your progress by saving the parent label file before altering the mesh.

Semi-automatic method: Similar to 2.5D meshes (check section 17.2), it is possible to semi-automatically parent label 3D meshes. Hereby, the error correction step is less reliable, so that it is advised to run the process step by step and manually correct errors in between. To run a single step, use the process “Mesh/Deformation/Mesh 3D/Auto Parent Labeling Step 3D”.

21.3. Change Maps 3D

Change maps for 3D meshes can be created the same way as for 2.5D meshes: Using the process “Mesh/Heat Map/Heat Map”, choose the measures/attributes to compare between two meshes, tick the “Change” checkbox. For 3D meshes it is more important to keep track that the measure or heat map has already been created as attribute map, otherwise the computation of the change map might fail. See also section 14 and 21.1.

21.4. PDGs 3D

Similar to surface (2.5D) meshes, growth directions can be computed using a lineage tracked sample and deformation functions also for 3D meshes. The overall procedure is a bit more complicated as it utilizes deformation functions (see chapter 17) and allows for more options compared to the 2.5D process. Proceed as follows:

Load the meshes. We assume that Stack 1 holds the earlier time point and Stack 2 the later time point (with parent labels).

Create a deformation function (see also section 17.1). As we now have two volumetric meshes, use the process “Mesh/Deformation/Mesh 3D/Create Deformation Cell Volumes”.

Create the growth directions (PDGs) of your choice. The process “Mesh/Cell Axis 3D/Deformation Gradient/Compute” allows the computation of the 3D PDGs. There are several options available for the visualization. PDGs can be visualized:

- I. **For the whole 3D cell (Mode = Cell or Stack).** Hereby a 3D growth tensor is assigned to each cell. The tensor is determined from the deformation function either at the cell centroid (Mode = Cell) or averaged across the inside voxels of a cell (Mode = Stack, requires a loaded segmented stack with the same labels as the cells). It generally is better to use the “Stack” mode if possible.
- II. **On the vertex level (Mode = Vertex).** This mode is similar to the subcellular PDGs for surface meshes. For each vertex the growth tensor is shown projected onto the surface of the mesh.

- III. **For individual cell walls (Mode = Cell Wall).** Same as the vertex mode, but tensors of the same cell wall are averaged. This mode requires the computation of the attribute map of individual cell walls (process “Mesh/Cell Mesh/Tools/Label Cell Walls” with “Only Attr Map” = Yes).

Once generated there are 3 different “Display” processes available, one for each of above options.

21.5. 3D Visualization Options

It can be a challenge to visualize 3D meshes with multiple cell layers, let alone additional information such as heat maps or growth directions. In the process folder “Mesh/Visualization 3D”, MorphoGraphX contains processes to facilitate the visualization which are described here. All of those processes require the attribute map of cell centroids being present. It can be created using the process “Mesh/Heat Map/Analysis/Cell Analysis 3D”.

Explode Mesh. Moves cells away from the origin of the coordinate system (can be visualized when pressing “A”) by an amount and direction depending on the cell centroids. Cells are separated and deeper cell layers become visible as the tissue appears to be “exploded”. There are options to restrict the explosion to selected cells.

Translate by Parent. Moves cells by a specified amount based on their parent label (which is thought to be a cell type label here).

Restore Original Mesh. Restores the initial mesh without visualization options.

Rotate Camera. Rotates the camera by a specified amount around the Cartesian axes. This process is useful when recording videos of a sample.

21.6. Custom Directions for 3D meshes

The functionality of custom directions for 3D meshes is similar to 2.5D meshes (see section 16.1 “Custom Directions with Bezier”). However, to generate the directions different processes need to be used, which can be found in the folder “Mesh/Cell Axis 3D/Custom”.

Custom directions for 3D meshes can be generated from:

- a **Bezier grid.** Create a Bezier grid and adjust it first (see also section 16.1). Directions correspond to the directions along x, y and the normal of the grid
- a **Bezier line.** Create a Bezier grid, collapse its points (“Misc/Bezier/Collapse Bezier Points” and adjust them first. (see also section 16.1). Direction correspond to Longitudinal (along the Bezier line), Radial (towards the line) and Circumferential (around the line).

- For 3D meshes a Bezier line can also be created automatically by selecting a cell file (it is important that each selected cell only has 2 neighbors, except the start and stop cell) and running the process “Misc/Bezier/Bezier From Cell File” (make sure to run the Mesh/Heat Map/Analysis/Cell Analysis 3D” process first to compute the cell centroid coordinates).
- a **surface mesh**. This option requires a (unlabeled) surface mesh in mesh2 (besides the labeled mesh in mesh1) and will compute for each cell a direction towards the nearest surface point.
- as **orthogonal directions** to any two directions created with above methods

For visual verification of the directions the process “Mesh/Cell Axis 3D/Custom/Display Custom Orientations” can be used. As directions are drawn at the cell centroid position (inside of each cell) it helps to lower the opacity of the mesh surface to make them visible.

21.7. Signal orientation for 3D meshes

Similar to the pipeline described in section 18.2 “Signal polarity” it is possible to perform this operation for 3D meshes. The general principle is the same: Signal is quantified in bins created from the cell centroid in 3D space. A Principal Component Analysis (PCA) determines max and min directions and amount of the signal.

The processes can be found in the “Mesh/Cell Axis 3D/Polarization” folder. The “Compute” process is required to perform the computation, after the “Display” process is used to visualize different heatmaps or change the general appearance of the data visualization.

22. Cell shape analysis using Principal Component Analysis (PCA)

For the advanced analysis of cellular shape geometry MGX offers processes that allow to run a PCA on the mesh triangles or cell voxels.

22.1 For 2.5D meshes

For the analysis of segmented surface meshes use the process “Mesh/Cell Axis/Shape Analysis/Compute Shape Analysis 2D”. For every cell the input for the PCA is the center points of the mesh triangles of the cell weighted by their area. Here it helps to have a relatively fine mesh. It is not recommended using the process on meshes with relatively few triangles per cell or cell meshes (that were generated using the process “Mesh/Cell Mesh/Convert to a cell mesh”). Depending on the overall cell shapes the span correction parameter can be set to “cuboid” or “ellipse”. This acts as a factor on the length of the displayed axis.

After running the “Compute” process the process “Mesh/Cell Axis/Shape Analysis/Display Shape Axis” allows the visualization of different heat maps and adjusting the appearance of the shape axes. The “Max” axis shows the direction and extension of the cell’s longest axis, whereas the “Min” axis is orthogonal to it and shows the direction and extension of the shortest cell axis. When custom directions have been computed previously, then also the cell’s extension along those custom axes can be computed and visualized (see section 16 “Custom Directions”).

Note that some processes in the “Mesh/Heat Map/Measure” folder already utilize the shape analysis processes. For instance, the “Geometry/Aspect Ratio” measure is identical with the shape analysis “Ratio Max/Min”. Details can be found in the description of the measure processes.

22.2. For 3D meshes

The shape analysis of 3D meshes works in a similar way as for 2.5D meshes (with processes for “Compute” and “Display”). There are dedicated processes for 3D meshes in the folder “Mesh/Cell Axis 3D/Shape Analysis”. For 3D meshes there are 2 options for the input for the PCA: 1) Mesh triangles and 2) Voxels from a segmented stack. Both methods give very similar results, however, the “voxel” method is thought to be more accurate. For the “voxel” method a segmented stack with the same labels (so ideally the one that was used to generate the mesh) needs to be present in the same stack.

Instead of 2 PCA axes (for “Max” and “Min”) in the case of 2.5D meshes, in 3D there are 3 axes for “Max”, “Min” and orthogonal to those two axes a “Mid” axis, which length is in between the values of “Max” and “Min”.

Note that axis in 3D will be inside of the cell, so it is useful to lower the opacity of cells in the main tab.

23. Organ-centric coordinate systems

To analyze heat maps of cell geometry or development further, it can be helpful to create organ-centric coordinate systems as well as to distinguish different cell types. There are many ways to accomplish those tasks in MorphoGraphX. This section will introduce the most important ones.

23.1. The Cell Distance Measure

The easiest way to create a continuous coordinate through an organ is the cell distance measure. It takes selected cells as input and calculates from every other cell in the organ the shortest path through the tissue to a selected cell.

Load a segmented mesh (2.5D or 3D).

Select the cells that should be the “origin” of the cell distance measure.

Measure the cell distance. Run the process “Mesh/Heat Map/Measures/Location/Cell Distance” for 2.5D meshes or “Mesh/Heat Map/Measures 3D/Location/Cell Distance” for 3D meshes. Note that there are different options on how connections between cells can be weighted for the calculation of the cell distance (parameter “Wall Weights”):

- Euclidean: measures the euclidean distance between cell centroids (in μm). Most commonly used for organ-centric coordinate systems.
- 1: measures how many cells need to be crossed.
- 1/Wall: Larger shared walls are now easier to cross (somewhat similar to the diffusion of a chemical compound)

Export the cell distance heat maps. Once one or more cell distance heat maps are created, they can be exported to CSV files together with other cell geometry or growth measures and analyzed.

23.2. Bezier Line and Grid

Another way to create a custom coordinate system in MorphoGraphX is using the Bezier grid or line (see section 16.1 “Custom Directions with Bezier” for general information on the Bezier handling). Compared to the cell distance measure above, the Bezier is a better choice for Radially symmetric organs (see also the section Cell Atlas Root 20.5) or organs where a Bezier line of grid can easily be aligned with the organ axes. (A Bezier line or grid can be more accurate in creating organ-centric coordinates, but it can also be much more time consuming to properly aligning it.)

Processes that create, save or load a Bezier are located in the “Misc/Bezier” folder.

23.3. Further types of organ coordinates

There is a number of further distance measures in the Mesh/Heat Map/Measures/Location folder to compute distances between meshes or the mesh and the Bezier (e.g. Distance to Bezier, Distance to Mesh) or to compute cell coordinates within different coordinate systems (e.g. Cell Coord, Polar Coord).

- Bezier processes require a Bezier line or grid (see also section 16.1 “Custom directions with Bezier” for more information on how to handle the Bezier)
- “Cell Coordinate” computes heat maps based on the world coordinate of cells along X, Y or Z directions. The origin of the world coordinates can be visualized by pressing “A”.
- “Distance to Mesh” computes the distance of cells to another mesh (needs to be loaded in mesh2)
- “Polar Coord” computes longitudinal, radial and circumferential coordinates based on an axis of the world coordinate system (press “A” to visualize it).
- “Major Axis Theta” computes the angle between the major axis of a cell and a reference direction (see also section 21 “Cell Shape Analysis using PCA”).

2.5D and 3D processes are in separate folders, but have similar names and functionalities.

24. Cell atlas and cell type classification

MorphoGraphX uses the “parent label” (which is primarily used for lineage tracking) as secondary cell label for assigning cell types or regions within an organ. Parent labels can be manually set or reset and saved or loaded using the processes in the folder “Mesh/Lineage Tracking”. Only one set of parent labels can be present at the same time. Due to this limitation, make sure to backup existing parent labels (“Mesh/Lineage Tracking/Save Parents”) before using any process for cell type classification. All following guides require a segmented mesh (2.5D or 3D).

24.1. Cell Atlas Root

This Add-on is designed for creating a 3D cell atlas and to classify cell types in radially symmetric organs such as roots.

Hereby an intrinsic coordinate system is created using a central line defined by a Bezier and a surface mesh. The processes for this pipeline are located in the folder “Mesh/Cell Atlas 3D/Root”. Check out the research article Montenegro-Johnson *et al.* (2015) for more details about the method and the method paper Stamm *et al.* (2017) for a detailed user guide.

24.2. Cell Atlas Meristem

Besides roots, we also developed a set of processes dedicated for analysis and cell type labeling of meristems.

The processes can be found in the folder “Mesh/Cell Atlas 3D/Meristem”. A detailed user guide for this analysis pipeline can be found in the supplementary information of Montenegro-Johnson *et al.* (2019).

24.3. CoordX

CoordX is an extension of Cell Atlas (section 24.1) and the cell distance measure (section 23.1) which allows to cell type and create organ coordinates of more complicated organs, such as the ovule of Arabidopsis. A separate user guide can be found in the supplements of Vijayan *et al.* (2022).

24.4. Cell Type Classification using a single heat map

Load a heat map of your choice. In order to capture geometrical differences, use a measure from “Mesh/Heat Map/Measures/Geometry”; for creating regions in the organ, go to the “Mesh/Heat Map/Measures/Location” folder (see also the section above). But in principle, any heat map that is loaded onto the mesh can be used. MorphoGraphX uses the “parent label” as secondary cell label. Here it is used as cell type label.

Using a single heat map to create a cell type labeling can be useful when:

- Sorting cells according to a threshold (e.g., “find all cells larger than 200 μm^2 ” or “find all cells within 200 μm^2 of a given center point”). For this use the process “Mesh/Heat Map/Heat Map Select” to select cells within a certain interval of the heat values. Now set the parent label of the selected cell using “Mesh/Lineage Tracking/Set Parent”.
- Sort cells into equally sized heat bins or equally sized cell bins according to the values of a heat map.

For this use the process “Mesh/Heat Map/Operators/Heat Map Binning”. There, the “Mode” parameter lets you choose whether the bins should be based on cell number (creating bins with the same number of cells) or on heat value (creating bins of equally sized heat intervals).

24.5. 3D Cell Layer Classification using a surface mesh

It is possible to create a cell type labeling on a 3D mesh for different cell layers (moving away from the surface of the organ). This method is part of the tools of Cell Atlas Meristem introduced above, however, it is possible to use it as a standalone process outside of the Cell Atlas Meristem pipeline in “Mesh/Heat Map/Measures 3D/Location/Mesh Distance”.

24.6. Cell Type Classification using two measures

Using two independent measures or heat maps allows the separation of multiple different cell types as it was also demonstrated in the 3D Cell Atlas add-on for radially symmetric organs such as roots (Montenegro-Johnson *et al.*, 2015). Note that there is an extensive method paper on how to use this add-on (Stamm *et al.*, 2017). For MorphoGraphX 2.0, we generalized the tools to use them for any kind of data and heat maps. To classify cell type using two measures follow these steps:

Generate the attribute maps of all measures and heat maps you want to use.

Run the process “Mesh/Cell Types/Classification/Tools/Cell Property Map 2D”. This will open a user interface with a plotting area in the center. The plot is generated by the attribute maps in the drop down menus “X-dim” and “Y-dim” (see Fig 34).

There are two main modes which can be changed with the “Mode” radio buttons: Cells and Heat. In Cell mode the data is plotted as points individually for all cells. In Heat mode, a Gaussian function is applied on the points. This will cause close points to overlap and form joined heat regions which can facilitate the clustering. In Heat mode, small white crosses appear to visualize local maxima in the 2D heat map.

When in Heat mode, a weighting can be applied to the amount of heat for each cell based on a third attribute map. This map can be chosen in the “Heat Type” dropdown menu.

To classify cells: Create a cell cluster by left-clicking into the heat map. A larger white cross will appear with the label “1”. Mouse-over this cell cluster and press any number between 1 and 9 to change the label. Left-mouse drag and drop the cell cluster to change its location. You can create as many clusters as necessary to cluster all the cells (see Fig 34).

Now press “Update Parents” to assign the cluster labels to the cells nearest to the associate labels.

More options are available in the GUI for changing the sigma parameter (when in “Heat” mode), to re-scale the viewing area and to run a k-means clustering step.

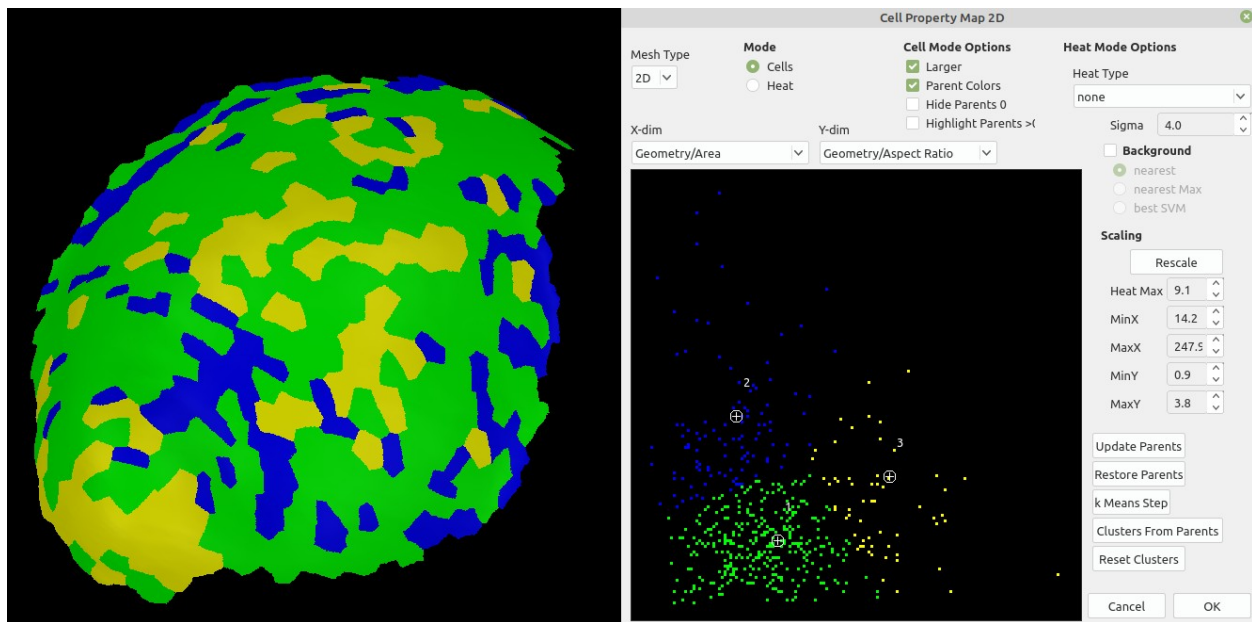


Figure 34. Cell Property Map 2D with clusters for the cell type classification.

24.7. Cell Type Classification using SVMs

SVMs (Support Vector Machines) are an established machine learning technique for the classification of data points. For MorphoGraphX 2.0, we implemented processes which interface with the libSVM C++ library ([link](#)) and enable the use of SVMs for cell type classification:

Run the process “Mesh/Cell Types/Classification/A Select Measures”, which opens a user interface with two lists: The left one shows all the measure processes (all processes in the “Mesh/Heat Map/Measures” folder). The right one shows the existing attribute maps in the mesh. Clicking on a measure in the left table will compute its values for the mesh and create the associated attribute map, thus creating an entry in the right table (unless it already existed). When working on a 3D mesh, it is required to run the process “Mesh/Heat Map/Analysis/Cell Analysis 3D” beforehand as the measures can not be created on the fly.

After creating all the required measures, select the measures you want to use to separate the data in the right table of the GUI and hit OK.

To avoid having to wait frequently, compute the attribute maps of all measures beforehand using the process “Mesh/Heat Map/Analysis/Cell Analysis 2D”. Running this process can be time consuming for larger meshes, but reduces waiting times in the GUI of the Select Measures

process particularly for measures that are computational demanding to calculate (e.g., “Lobyness/Visibility Stomata”). It creates attribute maps for each measure that can be saved with the mesh.

Custom heat map data can be loaded into MGX (process “Mesh/Heat Map/Heat Map Load”) and needs to be exported into an attribute map (“Heat Map/Operators/Export Heat to Attr Map”) before it appears in the right hand side table.

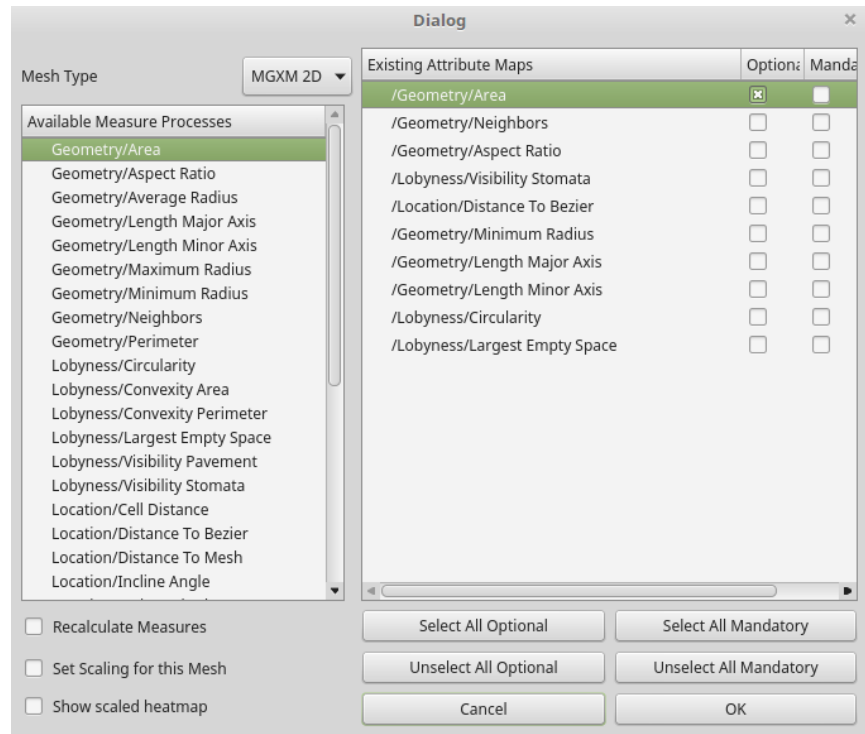


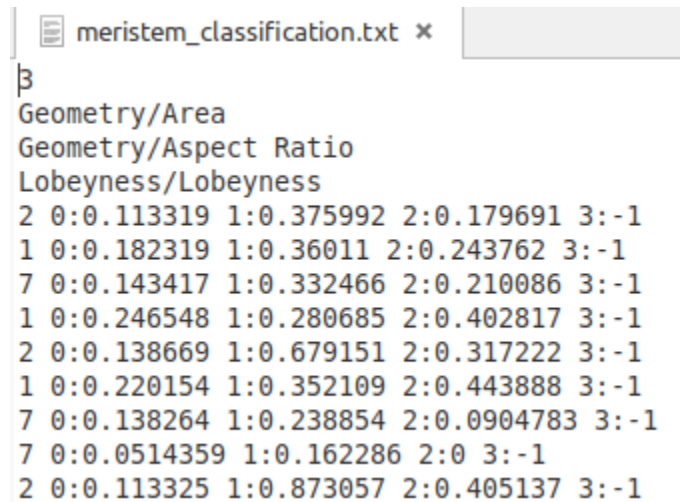
Figure 35. Select Measures GUI dialog. The left list shows the measure processes and the already existing attribute maps are in the right table.

Provide the training data. Assign parent labels to some representative cells as training data using “Mesh/Cell Types/Classification/B Specify Cell Types”. This process is identical with the process “Mesh/Lineage Tracking/Set Parent” and sets the parent label of selected cells to the specified value. All cells with a parent label >0 will be considered as training/testing data when creating the training data file.

Create the text file used for the SVM training by the libSVM library using the process “Mesh/Cell Types/Classification/C Write Training Data”. This process is used to create a training data file which can be read by the libSVM library to generate and train a SVM model. It finds the cells with parent label >0 and their values of the selected measures and writes them to a file.

There is also an option to add (append) the training data of the loaded mesh to an existing training data file. This is useful when the data from multiple meshes is written to a single training file.

When there is not much training data available (only a few cells) or when the data is not very balanced (some classes with only few cells, others with many), the process “Mesh/Cell Types/Classification/D Training File Augmentation” can extend the existing data with artificial entries, created from the real data with some added noise.



```

meristem_classification.txt
β
Geometry/Area
Geometry/Aspect Ratio
Lobeyness/Lobeyness
2 0:0.113319 1:0.375992 2:0.179691 3:-1
1 0:0.182319 1:0.36011 2:0.243762 3:-1
7 0:0.143417 1:0.332466 2:0.210086 3:-1
1 0:0.246548 1:0.280685 2:0.402817 3:-1
2 0:0.138669 1:0.679151 2:0.317222 3:-1
1 0:0.220154 1:0.352109 2:0.443888 3:-1
7 0:0.138264 1:0.238854 2:0.0904783 3:-1
7 0:0.0514359 1:0.162286 2:0 3:-1
2 0:0.113325 1:0.873057 2:0.405137 3:-1

```

Figure 36. Training File format for the libSVM library.

Continue the process as follows:

Train the SVM model using the process “Mesh/Cell Types/Classification /SVM/E Train SVM Model” and selecting the training file from the file dialog. It is also possible to load previously created models using the process “Mesh/Cell Types/Classification/SVM/F Load Model”. Note that all attribute maps that were used to create the model need to be present.

Classify the cells using the process “Mesh/Cell Types/Classification/SVM/G Classification”. Here, a classification threshold can be set. Only cells with a confidence value > threshold will be classified. To force the classification of all cells, set the threshold to 0.

25. Cell Division Analysis

MorphoGraphX offers a number of processes to analyze actual cell divisions or to simulate potential ones. All processes are located in the folder “Mesh/Division Analysis”.

25.1 Division Analysis 2.5D

This section walks you through the most important processes to analyze the division of cell in 2.5D meshes. Processes are in the group "Mesh/Division Analysis".

Open two segmented surface meshes. The analysis of actual cell divisions requires two segmented meshes of sequential time points, *e.g.*, T0 and T1). The division analysis works only when “Parents” are activated”. Currently, parent-labeled single cells and cell pairs (with the same parent label) can be analyzed.

Find the flat approximation of an actual division plane. Select a single parent label of recently divided cells. For finding the flat approximation of the actual division plane run the process “Analysis 2D/Division Plane Approximation”. Note that this process is only possible for a cell pair.

- **Simulate potential divisions for single cells.** Select exactly one label, either a cell or (when parents ticked) a single parent label of recently divided cells. Run the process “Analysis 2D/Division Analysis”. This process simulates division planes of certain criteria and computes properties such as division plane area and daughter cell sizes. When ran on a cell pair, the flat approximation plane is also computed.

By default the division goes through the center (of gravity) of the cell. A custom division point can be set, (parameter Custom Vertex: Yes will choose a selected vertex as new division point).

Further parameters allow to specify the number of planes to be tested (higher will test a finer resolution of planes). To test division planes in the vicinity of the division point set a discrete step size and number.

This process creates the attribute map “Cell Division Simulated Planes”, which is used by visualization and data export processes explained further below.

Simulate potential divisions for whole mesh. Activate the parent labels and run the process “Analysis 2D/Division Analysis Multi. The process repeatedly runs the single cell process on all single cells or cell pairs depending on the parameters. Test Single tests only single cells while Test Double tests cell pairs. When Test Single is set on Yes, the parameters of the single cell process will be used for the number of planes and the division point. Note that the Custom Point does not work for the process.

Analyze the actual division plane. To visualize a flat approximation of the actual division plane between 2 daughter cells run the process “Analysis 2D/Division Plane Approximation”.

This process creates the attribute map “Cell Division Actual Plane” which is used by visualization and data export processes explained further below.

Visualize the data. To visualize simulated division planes use the process “Mesh/Division Analysis/Display and Filter Planes”. This process will overwrite the other mesh with division planes of certain optimality defined by the parameters set. The parameters are here described:

Search on: the Simulated planes Attr Map or the Filtered Attr Map can be used. After running the process the Filtered Attr Map is replaced by the result.

Filter: criterion to filter planes.

Number of Planes: maximum number of planes in the filtered data (could be fewer depending on other restrictions).

Min Angle: the planes are sorted by the “Filter” criterion and then the best one is selected. When moving to the less optimal planes, it is checked whether their angle is at least these many degrees different than the already chosen ones.

Draw Planes: it creates the filter planes in the other mesh.

Reset Other Mesh: resets the other mesh before creating the planes.

Plane Size: diameter of the drawn planes.

Heat Map: map of the drawn planes.

Display Actual: if possible (cell pair) also draws the actual plane. The actual plane can be found among the other planes as it has no heat value (gray color) and is labeled with parent label of the cell pair (figure 36).



Figure 37. Filtered planes with heat map coloring. The actual plane is shown in gray.

Export the data. To export the data of the attribute maps, run the process “Common/Export Plane Data”. Select which data you want to save and specify the file name in the GUI dialog window.

Reset the attribute maps by running “Common/Reset Division Data”.

Quantify angles of planes in custom directions. Create custom directions with a default Bezier grid (see section 16.1), make sure the custom directions are created on the parent mesh. To compute the angle between Actual or Filtered Planes and Custom Directions or the Actual Plane, run “Common/Compute Division Plane Angles”. Make sure to have the custom directions present. After running the process, the heat map of the data is displayed on the screen. An attribute map with the name “Measure Label Double Angle” is created. Note that the attribute map is based on the parent labels.

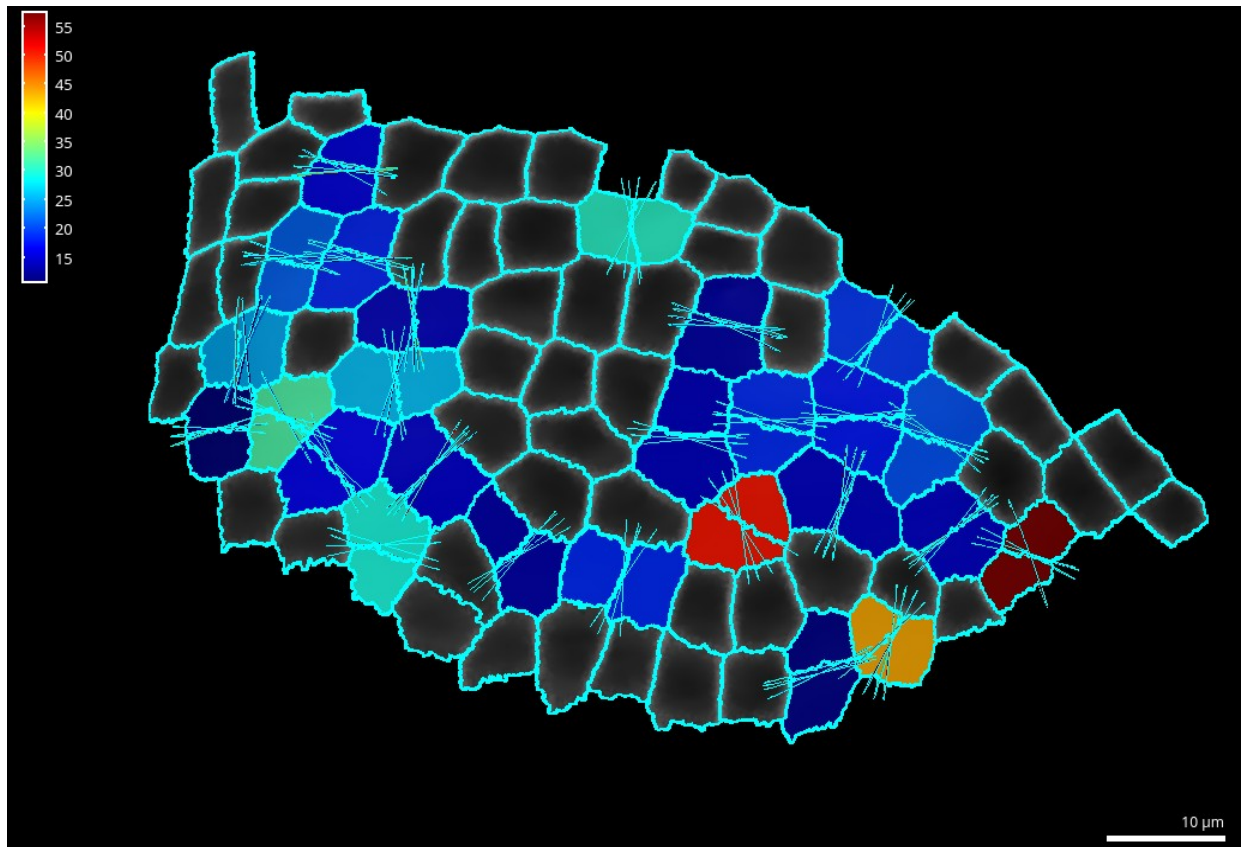


Figure 38. Surface mesh with division analysis on all divided cells (cell pairs). The “Common/Display and Filter Planes” was run to visualize the best planes and, afterwards, the “Common/Compute Division Plane Angles” process was run to see the heat map of the maximum angle between filtered and actual planes on the surface mesh.

25.2. Division Analysis 3D

The mesh structure of 2.5D and 3D meshes is different, which results in different processes for those two mesh types. Apart from that, the visualization and data export options are identical. Use the processes in the “Analysis 3D” folder (instead of the ones in the “Analysis 2D” folder) to generate the attribute map data.

26. Quantifying the stack signal without a mesh

MorphoGraphX allows certain types of quantification of the raw stack signal without the need of a segmented mesh. This sections introduces some of the available processes.

- **Stack/Analyses/Compute Volumes:** computes cell volumes based on a segmented stack
- **Stack/Analyses/Export Histogram Bezier:** exports a histogram of stack signal along a user-defined Bezier line (see also section 16.1 “Custom Directions with Bezier” for general information on the Bezier handling). The process has numerous parameters and can also visualize the bins inside the stack. It was implemented for Liu et al. 2022
- **Stack/Analyses/Export Histogram Circular:** Similar to above Bezier process, this process quantifies the stack signal around the origin (press “A” to visualize the world coordinate axes). This process was implemented for Zhu et al. 2020

27. Animation

MorphoGraphX offers the possibility to record translations, rotations and clippings of stacks and meshes in the form of animations. All processes are located in the folder “Misc/Animate”.

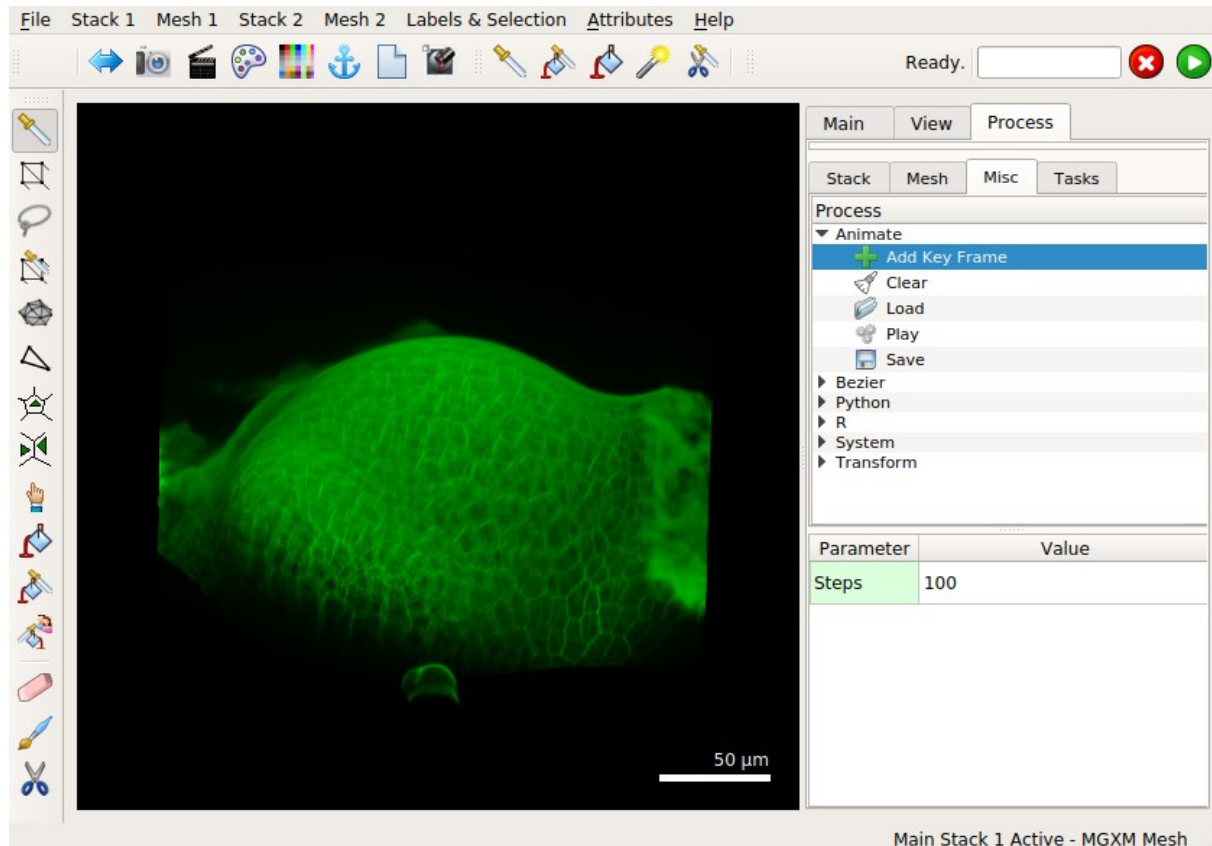


Figure 39. Overview of the “Misc/Animate” processes for creating animations.

To create an animation use the process “Add Key Frame”. The “Steps” parameter is used to specify the number of frames to be used from the previous saved ones. The starting frame steps can be set to 1. After moving the sample, save the additional frames with a step number of 100 for a smooth transition between the saved key frames. MorphoGraphX will interpolate the movement of the sample from the previous frames to the following ones. The saved frames can be removed with the “Clear” process.

To play an animation run “Play”. The animation can be played in loop by setting Yes in the *Loop* parameter.

To export/import an animation, you can save the animation as a .txt file with “Save”. The file can later be imported back into MorphoGraphX using the “Load” process.

28. Data Export

There are numerous ways to export data from MorphoGraphX to CSV files. CSV files can be read by other software such as Microsoft Excel, R or MatLab to perform an advanced statistical analysis or to create graphs.

To export a heat map use the process “Mesh/Heat Map/Heat Map Save”.

To export multiple heat maps, their data needs to be present as attribute maps (see section 12). When running a “Mesh/Heat Map/Measures” process (see section 10), an attribute map is created automatically. All other kind of heat map data can be exported to an attribute map using the process “Mesh/Heat Map/Operators/Export Heat to Attr Map”. This process also allows the copying of heat maps between meshes and cell label/parent labeling.

To export the cell neighborhood (size of shared walls between cells) use the processes “Mesh/Export/Save Cell Neighborhood 2D” or “Save Cell Neighborhood 3D”.

MorphoGraphX allows the following simple statistical analyses:

A histogram of a cell-based measure can be created from an attribute map by running the process “Mesh/Export/Histogram Exporter”.

The correlation of two measures can be calculated from their attribute maps using the process “Mesh/Export/Correlation Exporter”.

29. Creating workflows (tasks)

While processing data you will often use the same processes in a particular order. It is then convenient to organize the processes into tasks.

Open the Edit User Tasks window. Right click below the “Process/Tasks” tab. Click on “Edit tasks” and an Edit User Tasks window will open (figure 39). Here, the list of processes available is shown in the left panel and the list of tasks created is shown in the right one.

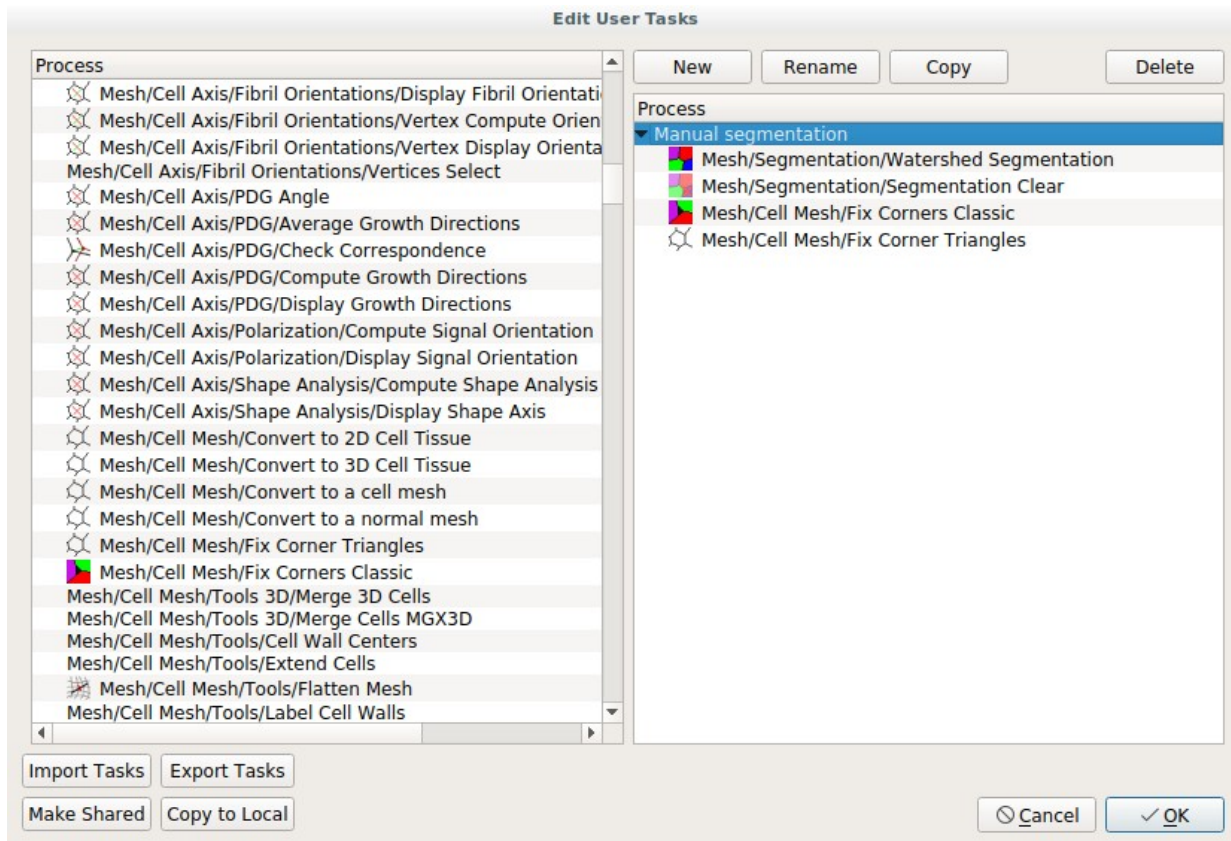


Figure 40. Creating and editing tasks in the Edit User Tasks window.

To create a task, click on “New” and assign a name to the task. Select the desired processes and drag them in the right panel on the task created. Click on the arrow beside the task, the list of the processes added to the task will be visible.

To rename a task, highlight the task of interest and click on “Rename”. Assign the new name in the pop-up window and press OK.

To edit a task, drag the processes above or below other ones so their order will change. A process cannot be deleted from a task, but it can be moved to another task.

To duplicate a task, highlight the task name and click on “Copy”. Assign a name to the task’s copy in the pop-up window and press OK.

To delete a task, highlight the task name and click on “Delete”. Press “Yes” in the pop-up window to confirm.

To save a task, click on “Export Tasks” and assign the name to the task. A .task file will be created (figure 39). Saving the MorphoGraphX session file (.mgxv format) will also save the parameters used in the tasks.

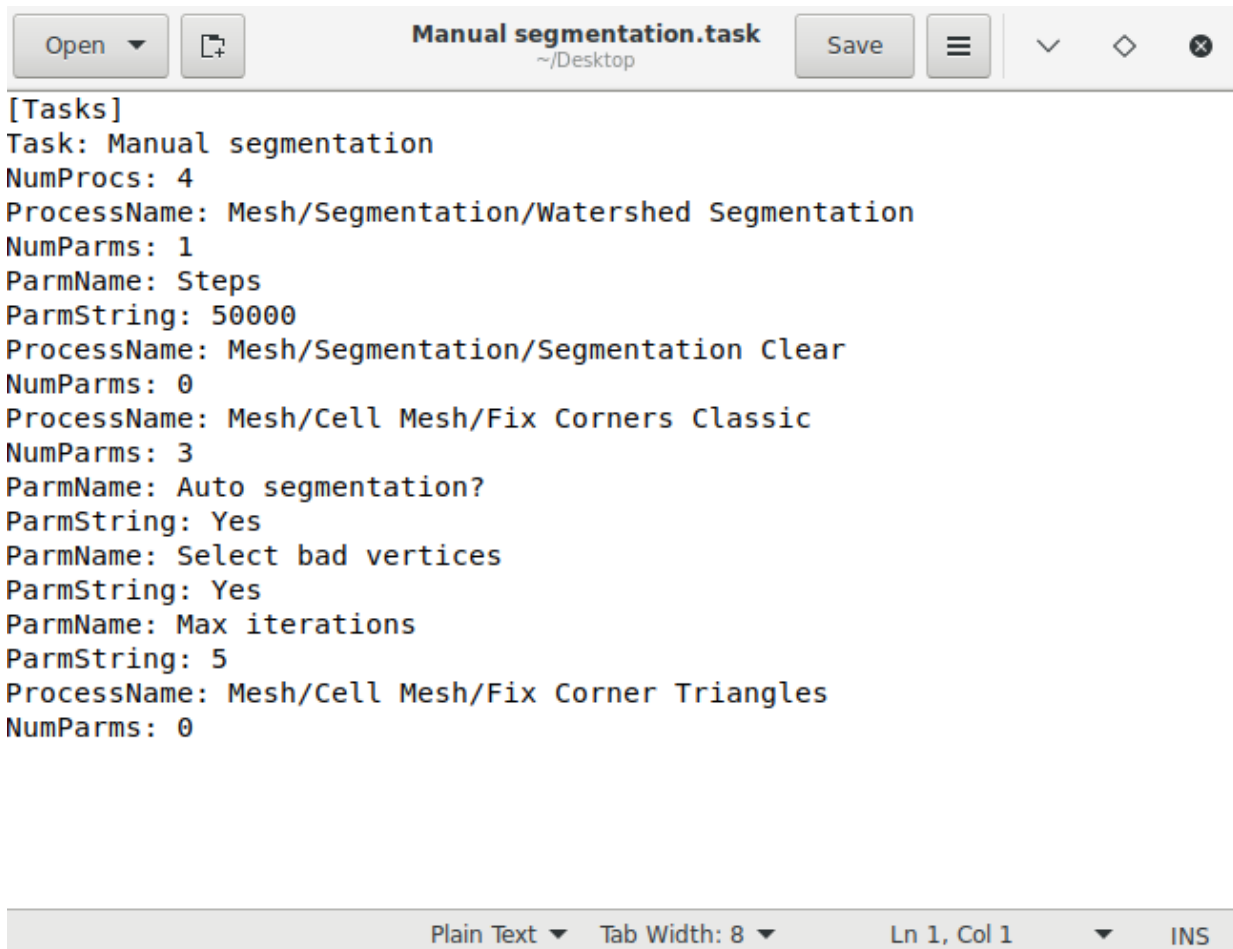


Figure 41. A .task file is created when exporting a task. The name assigned to the task and the number of processes included are displayed in “Task:” and “NumProcs:”, respectively. The processes included and their parameters are listed afterwards.

To import a task, click on “Import Tasks”. Select the desired .task file and press “Open”. The task with its processes will appear in the right panel of the Edit User Tasks window.

To share a task, click on “Make Shared”. An asterisk will appear at the end of the name of the task to identify tasks that are shared.

To copy a shared task to local, click on “Copy to Local”. A copy of the shared task will appear in the right panel of the Edit User Tasks window.

To confirm the changes, press OK. The tasks created or modified will be visible in the “Process/Tasks” tab.

30. Installation

30.1. Computer Requirements

To run MorphoGraphX you need a computer with enough main memory (at least 8 GB of RAM, the more the better) and a graphics card supporting OpenGL 3 running Windows or Linux. Mac OS is currently not supported; however, you should be able to run it on a Mac with a multiboot configuration, or possibly in a virtual machine. If you need to process large stacks, we recommend a recent nVidia graphics card with 4 GB or more dedicated video memory that supports Cuda. A multicore CPU is also beneficial for many processes.

30.2. Installation

MorphoGraphX is straightforward to install, however, it can be tricky to get the nVidia graphics drivers and Cuda installed correctly. Install the version of Cuda that matches your installation package, which should also install the correct graphics driver. If the version you need is not in the repositories, it may be necessary to install a newer version from the nVidia web site. It is always recommended to install this from packages.

MorphoGraphX is available as a Debian package which can be downloaded from www.MorphoGraphX.org. Binary versions are available for Linux Mint and Ubuntu. It is important to get the correct release, as different versions of Linux ship with different versions of the libraries that MorphoGraphX depends on.

Use your favorite software manager to install the MorphoGraphX package (or just double click on it). In order to use the 3D segmentation and other tools from the Insight Toolkit, you will need to install the ITK package as well. A compiled version of the ITK package is available on the MorphoGraphX web site for convenience. This will be the same as the version available from www.itk.org, although it will have some optional components compiled in. If you would prefer to compile ITK yourself, be sure to enable the `Module_ITKReview` in CMake to enable the use of the MorphologicalWatershed filter.

30.3. After installation – enabling the ALT key

After MorphoGraphX has been installed, it should show up in the menu system, or you can start it by typing “mgx” at the command line or double-clicking on a MorphoGraphX project (.mgxv), mesh (.mgxm) or stack (.mgxs) file. When MorphoGraphX first starts, double check that there are no errors in the terminal window. If the graphics card is identified properly, and the memory is allocated to Cuda, it is most likely everything installed correctly.

The next task is to enable the ALT key. In Linux Mint and Ubuntu, the ALT key is used to move windows around. MorphoGraphX uses this key for user interaction, so you will need to change the key used by Linux to the “Super” (or Windows) key. In Mint cinnamon, this is done by right clicking on tasks bar, and going into “All-Settings” - “Windows”. There you change the mouse

modifier key from Alt to the Super key. Different window managers use different methods to make this setting.

Troubleshooting

Symptom: The viewer window is not black, and you get shader errors in the terminal window, or when you load a stack it is a single solid color. This means that the nVidia graphics driver is not installed properly.

Symptom: Libraries (.so files) not found. Sometimes libraries are not found immediately after installation. At the command line type:

```
$ sudo ldconfig
```

Symptom: You see the following error message in the terminal window when MorphoGraphX starts: “Cuda holdMem, cannot allocate 8 Meg, giving up”. This means that the Cuda driver is not installed properly. If you can get any application that uses Cuda working, then MorphoGraphX should work as well

31. Writing custom processes (plugins).

It is possible to extend MorphoGraphX by writing your own processes. Processes can be compiled into shared object (.so) files and are loaded when MorphoGraphX starts. They can be installed in a system area for all users, or in the user's home directory. Run the command:

```
$ MorphoGraphX --all-process
```

to print the plug-in directories. An overview for the process documentation can be found in the “process” namespace in the Doxygen programmer documentation, which is available from the help menu in MorphoGraphX. After installing MorphoGraphX, this documentation can also be found here:

```
“/usr/local/share/doc/MorphoGraphX/html/index.html”.
```

The best way to start developing processes is to start from a sample available from the MorphoGraphX website: www.MorphoGraphX.org. In the “Samples” plug-in pack there are the following simple processes:

StackGammaFilter - This process implements a simple gamma filter on the stack. It demonstrates how to read and write information from the stack, and how to collect parameters from the user interface. Note that this process inherits from the StackProcess class.

MeshGammaFilter - This process implements a simple gamma filter on the mesh signal. In order to use this process, you will need to have extracted a mesh, and projected the signal on in (see section 6). Note that this process inherits from the MeshProcess class.

ExportMeshToFile - This process demonstrates how to write mesh data to a text file. It saves the mesh in simple OBJ (Wavefront) and PLY (Stanford Polygon) formats. The process also demonstrates how to use drop-down pick boxes in the GUI and how to call a dialog for file selection.

ITKMedianImageFilter - This plugin demonstrates how to call filters from the Insight Toolkit (ITK) image processing library. MorphoGraphX provides an image “source” for input, and a “sink” that writes the data back to the Work Store.

To compile a plugin, extract the archive into a directory. You can then type:

```
$ qmake
```

```
$ make
```

```
$ sudo make install
```

By default, the processes are set to install into the system area. In cases where you do not have sudo rights, or you wish to install only for yourself, you can copy the “.so” file to your local process directory. This will be in your home directory under:

```
“.mgx/processes”
```

See above to list the process directories.

To create a new process, it is easiest to copy an existing one. For the StackGammaFilter process, there are 5 files:

StackGammaFilter.hpp – C++ process headerfile

StackGammaFilter.cpp – C++ process file

StackGammaFilter.png – Icon file

StackGammaFilter.qrc – Qt resource file (points to icon file)

StackGammaFilter.pro – Qmake project definition file.

In order to compile processes you will need to have g++ and the Qt5 development tools installed.

32. Compiling MorphoGraphX from source

The source code for MorphoGraphX is released alongside the binary distributions. CMake is used for the build system.

32.1. Linux.

In order to compile MorphoGraphX, you will need to install the following packages:

`libtbb-dev libtiff-dev libboost-all-dev cimg-dev libgsl-dev libqt5opengl5-dev ocl-icd-opencl-dev libtriangle-dev libtet1.5-dev libcgall-dev libqcustomplot-dev libeigen3-dev`

Some of the addons will require further packages to be installed.

32.2. Windows.

An installer for a (non-Cuda) windows version is provided, but is less tested than on Linux.

32.3. Mac.

OpenGL does not support a recent enough version of OpenGL to support MorphoGraphX rendering in addition to immediate mode. This means that it is unlikely to be possible to have a native Mac version without substantial changes to the viewer (libQGLViewer.com). Apple appears to be dropping support for OpenGL in the future. Anyone that would like to contribute to MorphoGraphX by helping to develop a Mac version is very welcome to get in touch.

MorphoGraphX does however work (non-Cuda version) in a virtual machine. Another option is a dual boot that includes Linux or Windows.

33. Command line.

You can start MorphoGraphX or access useful information using a terminal.

To launch the software:

```
$ mgx
```

To start a session from an existing project file:

```
$ mgx YourProjectFile.mgxv
```

To view all the available options:

```
$ mgx --help
```

You should see the following message:

```
Usage: MorphoGraphX [<Command>] | [--debug] [--addlibrary LibraryPath] [--model ModelPath [--run]]  
[FILE.mgxv]
```

Flags:

--debug	- Launch MorphoGraphX in debug mode
--model	- set the model/process
--run	- runs the model/process automatically
--addlibrary	- adds a process library to MorphoGraphX

Commands:

--dir	- Print the application directory and exit
--process	- Print the process directory and exit
--user-process	- Print the user process directory and exit
--all-process	- Print all the directories searched for processes and exit
--resource	- Print the resource directory and exit
--include	- Print the include directory and exit
--lib	- Print the library directory and exit
--version	- Display the version and revision and exit
--help -h	- Print this help

34. Measure tables

Table 1: Measures for cells on surface projections (2.5D).

Measure	Unit	Description
Geometry		
Area	um ²	Area of the Cell (sum of its triangle area)
Aspect Ratio	-	Ratio of Length Major Axis and Length Minor Axis (see below)
Average Radius	um	Average distance from the center of gravity of a cell to its border
Junction Distance	um	Max or min distance between neighboring junctions of a cell
Length Major Axis	um	Length of the major axis of the 2D Shape Analysis: (Computes a PCA on the triangle positions weighted by their area)
Length Minor Axis	um	Length of the minor axis of the 2D Shape Analysis: (Computes a PCA on the triangle positions weighted by their area)
Maximum Radius	um	Maximum distance from the cell center to its border
Minimum Radius	um	Minimum distance from the cell center to its border
Perimeter	um	Sum of the length of the border segments of a cell
Lobeyness		
Circularity	-	$\text{Perimeter}^2 / (4 * \pi * \text{Area})$
Lobeyness	-	Ratio of cell perimeter and convex hull perimeter. 1 for convex shapes
Rectangularity	-	Ratio of cell area and the area of the minimum rectangle that can contain the cell. 1 for rectangular shapes, lower values for irregular shapes.
Soldarity	-	Ratio of the convex hull area and the cell area. 1 for convex shapes, higher values for complicated shapes.
Visibility Pavement	-	Simply 1-Visibility(Stomata)
Visibility Stomata	-	Estimate of visibility in the cell. Returns 1 for convex shapes, and decreases with the complexity of the contour.
Location		
Bezier Coord	um	Associated Bezier coordinate of a cell. Requires a Bezier grid.
Bezier Line Coord	um	Associated Bezier coordinate of a cell. Requires a Bezier curve.
Cell Coordinate	um	Cartesian coordinate of a cell
Cell Distance	um/ cells	Distance to the nearest selected cell (finds the shortest path to a selected cell, different distance measures: euclidean, cell number or 1/wall area)
Distance to Bezier	um	Euclidean distance to the Bezier curve or grid
Distance to Mesh	um	Euclidean distance to the nearest vertex in the other mesh
Major Axis Theta	°	Angle between the long axis of the cell and a reference direction

Polar Coord	°/um	Polar coordinate around a specified Cartesian axis
Network		
Neighbors	count	Number of neighbors of a cell
Betweenness Centrality	-	Computes the betweenness centrality of the cell connectivity graph. Edges can be weighted by the length of the shared wall between neighboring cells
Betweenness Current Flow	-	Computes the betweenness current flow of the cell connectivity graph. Edges can be weighted by the length of the shared wall between neighboring cells
Signal		
Signal Border		Average amount of border signal in a cell
Signal Interior		Average amount of interior signal in a cell
Signal Parameters		Advanced and general process which allows the setting of parameters to compute different kinds of signal quantifications
Signal Total		Average amount of total (=border + interior) signal in a cell
Other Measure processes		
Mesh/Lineage Tracking/Heat Map Proliferation	cells	Proliferation
Mesh/Cell Axis/Custom/Custom Direction Angle	°	Angle between a Cell Axis and a Custom Axis
Mesh/Division Analysis/Compute Division Plane Angles	°	Angle between division planes and/or cell axes

Table 2: Measures for meshes with volumetric (3D) cells

Measure	Unit	Description
Cell Atlas		
Cell Length (Circumferential, Radial, Longitudinal)	um	Cell length as determined by 3D Cell Atlas Root (Shoot rays from the cell center to the side walls to measure cell size along organ-centric directions)
Coord (Circumferential, Radial, Longitudinal)		3D organ coordinates as determined by 3D Cell Atlas Root
Geometry		
Cell Length (Custom, X, Y, Z)	um	Cell length along the specified direction (cell size is measured as in 3D Cell Atlas Root (see above))
Cell Wall Area	um ²	Total area of the cell wall
Cell Volume	um ³	Volume of the cell
Outside Wall Area	um ²	Cell wall area that is not shared with another neighboring cell

Outside Wall Area Ratio	%	Proportion of Cell Wall Area that is not shared with a neighbor cell
Location		
Bezier Coord	um	Associated Bezier coordinate of a cell. Requires a Bezier curve or grid.
Cell Coordinate	um	Cartesian coordinate of a cell
Cell Distance	um/cells	Distance to the nearest selected cell (finds the shortest path to a selected cell, different distance measures: euclidean, cell number or 1/wall area)
Distance to Bezier	um	Euclidean distance to the Bezier curve or grid
Mesh Distance	um	Euclidean distance to the nearest vertex in the other mesh
Network		
Neighbors	count	Number of neighbors of a cell
Betweenness Centrality	-	Computes the betweenness centrality of the cell connectivity graph. Edges can be weighted by the length of the shared wall between neighboring cells
Betweenness Current Flow	-	Computes the betweenness current flow of the cell connectivity graph. Edges can be weighted by the length of the shared wall between neighboring cells

References

- Bassel, G.W., Stamm, P., Mosca, G., de Reuille, P.B., Gibbs, D.J., Winter, R., Janka, A., Holdsworth, M.J. and Smith, R.S., 2014. Mechanical constraints imposed by 3D cellular geometry and arrangement modulate growth patterns in the Arabidopsis embryo. *Proceedings of the National Academy of Sciences*, 111(23), pp.8685-8690.
- de Reuille, P.B., Routier-Kierzkowska, A.L., Kierzkowski, D., Bassel, G.W., Schüpbach, T., Tauriello, G., Bajpai, N., Strauss, S., Weber, A., Kiss, A. and Burian, A., 2015. MorphoGraphX: A platform for quantifying morphogenesis in 4D. *Elife*, 4, p.e05864.
- de Reuille, P.B., Robinson, S. and Smith, R.S., 2014. Quantifying cell shape and gene expression in the shoot apical meristem using MorphoGraphX. In *Plant Cell Morphogenesis* (pp. 121-134). Humana Press, Totowa, NJ.
- Eschweiler, D., Spina, T. V., Choudhury, R. C., Meyerowitz, E., Cunha, A., & Stegmaier, J. (2019, April). CNN-based preprocessing to optimize watershed-based cell segmentation in 3D confocal microscopy images. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (pp. 223-227). IEEE.
- Hervieux, N., Dumond, M., Sapala, A., Routier-Kierzkowska, A.L., Kierzkowski, D., Roeder, A.H., Smith, R.S., Boudaoud, A. and Hamant, O., 2016. A mechanical feedback restricts sepal growth and shape in Arabidopsis. *Current Biology*, 26(8), pp.1019-1028.
- Jackson, M.D., Duran-Nebreda, S., Kierzkowski, D., Strauss, S., Xu, H., Landrein, B., Hamant, O., Smith, R.S., Johnston, I.G. and Bassel, G.W., 2019. Global topological order emerges through local mechanical control of cell divisions in the Arabidopsis shoot apical meristem. *Cell systems*, 8(1), pp.53-65.
- Kierzkowski, D., Runions, A., Vuolo, F., Strauss, S., Lymbouridou, R., Routier-Kierzkowska, A.L., Wilson-Sánchez, D., Jenke, H., Galinha, C., Mosca, G. and Zhang, Z., 2019. A Growth-Based Framework for Leaf Shape Development and Diversity. *Cell*, 177(6), pp.1405-1418.
- Kierzkowski, D., Nakayama, N., Routier-Kierzkowska, A.L., Weber, A., Bayer, E., Schorderet, M., Reinhardt, D., Kuhlemeier, C. and Smith, R.S., 2012. Elastic domains regulate growth and organogenesis in the plant shoot apical meristem. *Science*, 335(6072), pp.1096-1099.
- Liu, S., Strauss, S., Adibi, M., Mosca, G., Yoshida, S., Dello Ioio, R., Runions, A., Andersen, T. G., Grossmann, G., Huijser, P., Smith, R. S., & Tsiantis, M. (2022). Cytokinin promotes growth cessation in the Arabidopsis root. *Current Biology*.
<https://doi.org/https://doi.org/10.1016/j.cub.2022.03.019>
- Montenegro-Johnson, T.D., Stamm, P., Strauss, S., Topham, A.T., Tsagris, M., Wood, A.T., Smith, R.S. and Bassel, G.W., 2015. Digital single-cell analysis of plant organ development using 3DCellAtlas. *The Plant Cell*, 27(4), pp.1018-1033.

Montenegro-Johnson, T., Strauss, S., Jackson, M.D., Walker, L., Smith, R.S. and Bassel, G.W., 2019. 3DCellAtlas Meristem: a tool for the global cellular annotation of shoot apical meristems. *Plant methods*, 15(1), p.33.

Sapala, A., Runions, A., Routier-Kierzkowska, A.L., Gupta, M.D., Hong, L., Hofhuis, H., Verger, S., Mosca, G., Li, C.B., Hay, A. and Hamant, O., 2018. Why plants make puzzle cells, and how their shape emerges. *Elife*, 7, p.e32794.

Stamm, P., Strauss, S., Montenegro-Johnson, T.D., Smith, R. and Bassel, G.W., 2017. In Silico Methods for Cell Annotation, Quantification of Gene Expression, and Cell Geometry at Single-Cell Resolution Using 3DCellAtlas. In *Plant Hormones* (pp. 99-123). Humana Press, New York, NY.

Strauss, S., Sapala, A., Kierzkowski, D. and Smith, R.S., 2019. Quantifying Plant Growth and Cell Proliferation with MorphoGraphX. In *Plant Cell Morphogenesis* (pp. 269-290). Humana, New York, NY.

Strauss, S., Runions, A., Lane, B., Eschweiler, D., Bajpai, N., Trozzi, N., ... & Smith, R. S. (2021). MorphoGraphX 2.0: Providing context for biological image analysis with positional information. *BioRxiv*.

Vijayan, A., Tofanelli, R., Strauss, S., Cerrone, L., Wolny, A., Strohmeier, J., Kreshuk, A., Hamprecht, F. A., Smith, R. S., & Schneitz, K. (2021). A digital 3D reference atlas reveals cellular growth patterns shaping the arabidopsis ovule. *ELife*, 10, 1–38. <https://doi.org/10.7554/ELIFE.63262>

Vijayan, A., Strauss, S., Tofanelli, R., Mody, T. A., Lee, K., Tsiantis, M., Smith, R. S., & Schneitz, K. (2022). The annotation and analysis of complex 3D plant organs using 3DCoordX. *Plant Physiology*, kiac145. <https://doi.org/10.1093/plphys/kiac145>

Wolny, A., Cerrone, L., Vijayan, A., Tofanelli, R., Barro, A. V., Louveaux, M., ... & Kreshuk, A. (2020). Accurate and versatile 3D segmentation of plant tissues at cellular resolution. *Elife*, 9, e57613.

Zhu, M., Chen, W., Mirabet, V., Hong, L., Bovio, S., Strauss, S., ... & Roeder, A. H. (2020). Robust organ size requires robust timing of initiation orchestrated by focused auxin and cytokinin signalling. *Nature plants*, 6(6), 686-698.

Sample Data

Sample data sets available for download at www.MorphoGraphX.org:

Kierzkowski 2012 - tomato meristem time-lapse

Bassel 2014 - mature *Arabidopsis* embryo cleared for 3D segmentation

Hervieux 2016 - *Arabidopsis* sepal time lapse