Track 'n' R

User guide

***Florent Pantin**, 2013-2018*

Last revision: 28/08/2018

GitHub repository

To cite Track 'n' R in publications please use:

Fox S., Southam P., Pantin F., Kennaway R., Robinson S, Castorina G., Sánchez-Corrales Y. E., Sablowski R., Chan J., Grieneisen V., Marée A. F. M., Bangham J. A., Coen E. **2018**. Spatiotemporal coordination of cell division and growth during organ morphogenesis. *PLoS Biology*, in press.

Contents

Description

Track 'n' R is an ImageJ toolset allowing to run R scripts that help in the analysis of PointTracker data.

Once a leaf growth tracking project has been completed using PointTracker, this toolset allows the user to:

- draw the leaf outlines on the processed images
- replot growth from PointTracker processed data
- plot growth (actual data or contour estimation) and division from PointTracker raw data between several selected images or at regular time intervals
- analyze cell lineages

Installation

This toolset will work under **Windows only**. It requires Java, ImageJ and R. If you have installed a **64-bit version of Java**, make sure **ImageJ** uses this version. R will come with both 32-bit and 64-bit.

- Install Java for Windows if not already on your computer.
- Install ImageJ for Windows if not already on your computer.
- Install R for Windows (base) if not already on your computer.
- Download the *Track-n-R-master.zip* archive from GitHub and unzip it.
- Put the file ***Track 'n' R.txt*** in the subfolder ***\macros\toolsets\*** of the ImageJ directory (generally *C:\Program Files\ImageJ\macros\toolsets\*).
- Put the folder ***Track 'n' R*** in the subfolder ***\macros*** of the ImageJ directory (generally *C:\Program Files\ImageJ\macros\*).
- The installation is finished.

NB: This toolset was last tested under Windows 7 Enterprise SP1 64-bit using:

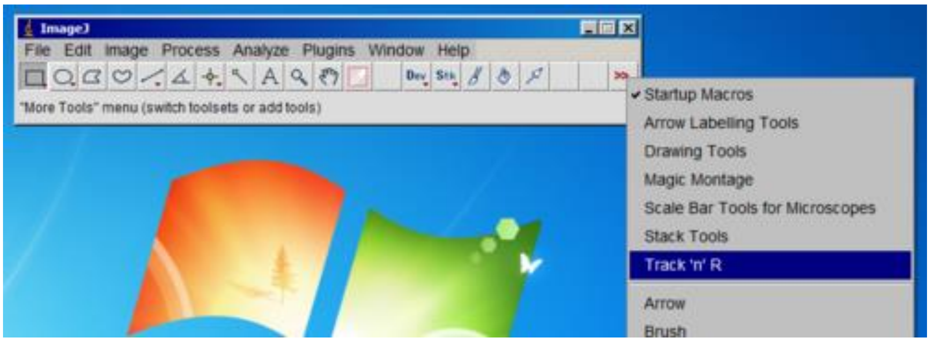
- Java jre-8u181-windows-x64
- ImageJ 1.52e
- R x64 3.5.1
- data files from PointTracker 0.5

The macros will create some temporary files in the root of the D:\ drive so **the user needs to have a D:\ drive with permissions of writing.** If there is no D:\ drive on your system, a USB flash drive can be used for this purpose (only a few Ko will be temporarily required).

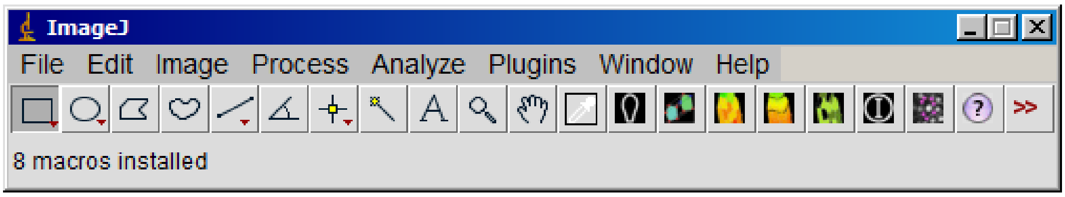
Use

The general principle is to call some R scripts from ImageJ. An exception is 'Draw leaf outlines', which is strictly an ImageJ macro. The core R functions are written in the file *Track_n_R.r* that can be found in the *...\ImageJ\macros\Track 'n' R* subfolder. The user only needs to pass the relevant parameters using ImageJ's dialog boxes.

- Open ImageJ.
- Click on the **>>** button and select the Track 'n' R toolset.


ImageJ
File Edit Image Process Analyze Plugins Window Help
"More Tools" menu (switch toolsets or add tools)
✓ Startup Macros
Arrow Labelling Tools
Drawing Tools
Magic Montage
Scale Bar Tools for Microscopes
Stack Tools
Track 'n' R
Arrow
Brush

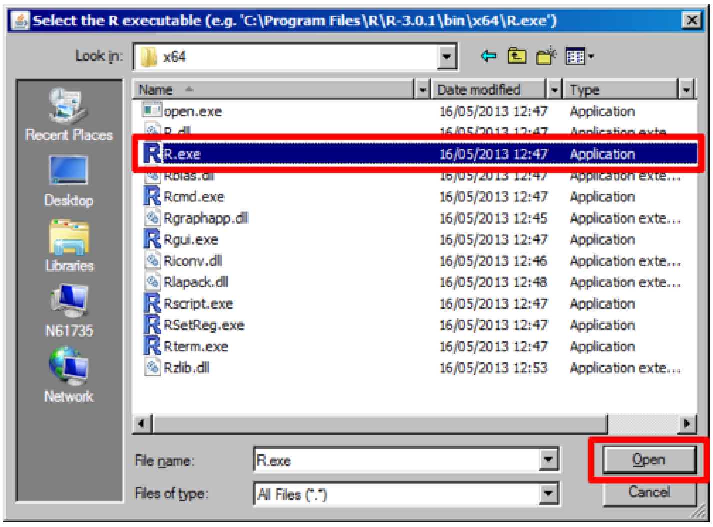
- A series of icons have now appeared on the right of your toolbar.



8 macros installed

- Just press one of the icons to run the associated macro and follow the instructions.

- On the first time that one of the macros (except 'Draw leaf outlines') is run, the user will be asked to locate the R executable ***R.exe***. Do not forget to adapt the path of the R version to the version of Java:
    - if you are running Java **64-bit**, select *R.exe* from the subfolder ***\bin\x64\***
    - if you are running Java **32-bit**, select *R.exe* from the subfolder ***\bin\i386\***


Select the R executable (e.g. 'C:\Program Files\R\R-3.0.1\bin\x64\R.exe')
Look in:
x64
Recent Places
Desktop
Libraries
N61735
Network
Open
Cancel

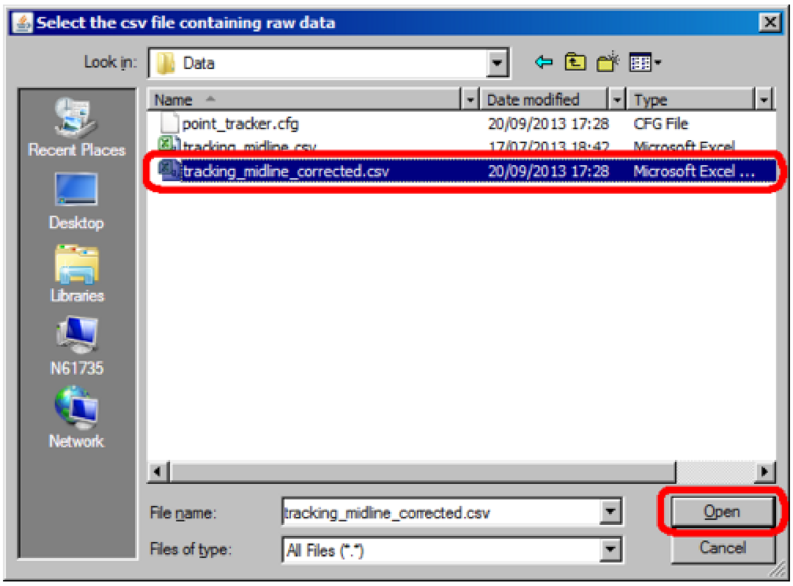
Name	Date modified	Type
open.exe	16/05/2013 12:47	Application
R.dll	16/05/2013 12:47	Application exte...
R.exe	16/05/2013 12:47	Application
Rblas.dll	16/05/2013 12:47	Application exte...
Rcmd.exe	16/05/2013 12:47	Application
Rgraphapp.dll	16/05/2013 12:45	Application exte...
Rgui.exe	16/05/2013 12:47	Application
Riconv.dll	16/05/2013 12:46	Application exte...
Rlapack.dll	16/05/2013 12:48	Application exte...
Rscript.exe	16/05/2013 12:47	Application
RSetReg.exe	16/05/2013 12:47	Application
Rterm.exe	16/05/2013 12:47	Application
Rzlib.dll	16/05/2013 12:53	Application exte...

File name: R.exe

Files of type: All Files (*.*)

- A file named ***Rpath.txt*** will then be saved in the subfolder *...\ImageJ\macros\Track 'n' R*, which requires **writing permissions to be enabled for the ImageJ folders**. Then the user won't need to enter this information again, unless *R.exe* or *Rpath.txt* has been moved (see Debug section).

- Then the user is prompted for the ***.csv*** file of the raw data (or the ***.xls*** file of the processed data for the macro 'Replot growth from PointTracker'):



Select the csv file containing raw data

Look in: Data

Name	Date modified	Type
point_tracker.cfg	20/09/2013 17:28	CFG File
tracking_midline.csv	17/07/2013 18:42	Microsoft Excel
tracking_midline_corrected.csv	20/09/2013 17:28	Microsoft Excel ...

File name: tracking_midline_corrected.csv

Files of type: All Files (*.*)

Note that the ***.csv*** file should be located in the ***\Data*** subfolder of the project directory, while the ***.xls*** file should be located in the ***\Growth*** subfolder of the project directory. If not, an error message will be displayed and the user will be prompted to select a correct csv/xls file. **Avoid spaces in the name of the *.csv or *.xls files**, as Track 'n' R cannot manage those files in some systems.

An **example dataset** is provided with the software. It can be found in the ***\Help\Example dataset*** subfolder. It contains several subfolders:

- a subfolder ***\Processed*** with 7 images (oriented upwards, yet this is not mandatory)
- a subfolder ***\Leaf Shape*** in which the 7 corresponding leaf outlines have already been drawn (see below, Draw leaf outlines)
- a subfolder ***\Data*** with 6 PointTracker csv files. Since some cells are not visible from the beginning (e.g. at the leaf margin) while others disappear from the focal plane (e.g. at the leaf tip), several datasets had to be produced in PointTracker to take the most of the information available between each time step. Accordingly, csv file #1 should be used with images 1 and 2, csv file 2 should be used with images 2 and 3, etc.
- a subfolder ***\Divisions*** in which the 6 corresponding division files have been already generated (see below, Plot cell division, Plot intervals or Analyze lineages)

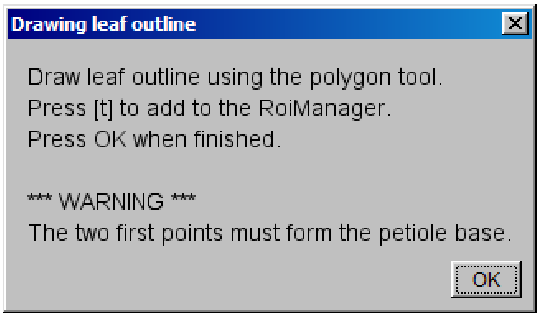
To use this example dataset, it is recommended to copy and paste it in the subfolder ***...\ImageJ\macros\Examples***, because writing permissions are generally denied for R in the other ImageJ subfolders (see Debug). You can also copy and paste it in ***My Documents***, for example.

Draw leaf outlines

> ImageJ macro: *Draw_Leaf_Outlines.txt*

When starting the analysis of a new project, **it is strongly advised to run the macro 'Draw leaf outlines' before any other**, as leaf outlines will be used to calculate the orientation of the leaf (unless the user provides it manually). Please note leaf outlines have already been drawn in the provided example dataset.

The user is prompted to draw leaf outline of each picture tracked in PointTracker. Each **_*.png_** image should be stored in the subfolder **\Processed** of the project directory. Each outline will be saved in its own **_*.txt_** file containing XY coordinates in a subfolder **\Leaf Shape** which will be automatically created in the project directory. If some leaf outlines had already been digitized, the user will not be prompted to digitize them again (except if the associated _*.txt_ file have been deleted).

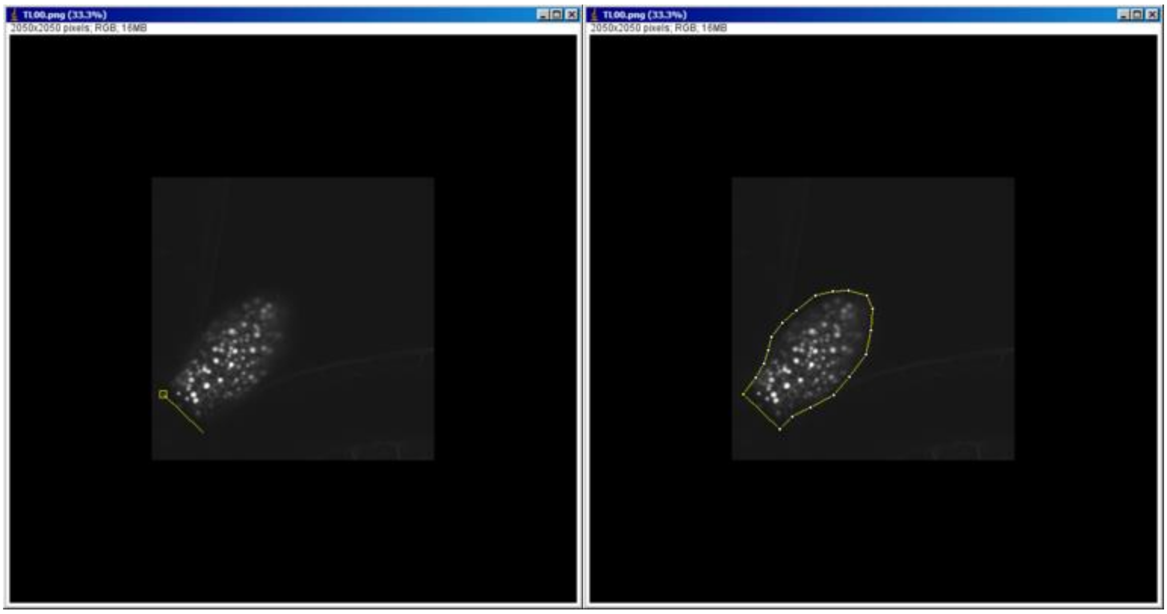


Drawing leaf outline

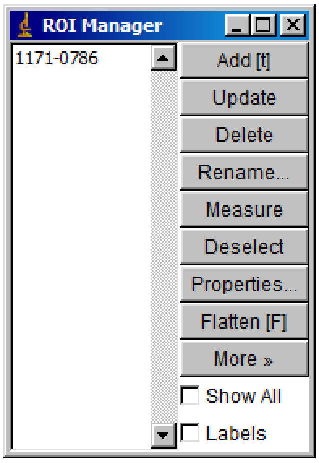
Draw leaf outline using the polygon tool.
Press [t] to add to the RoiManager.
Press OK when finished.

*** WARNING ***
The two first points must form the petiole base.

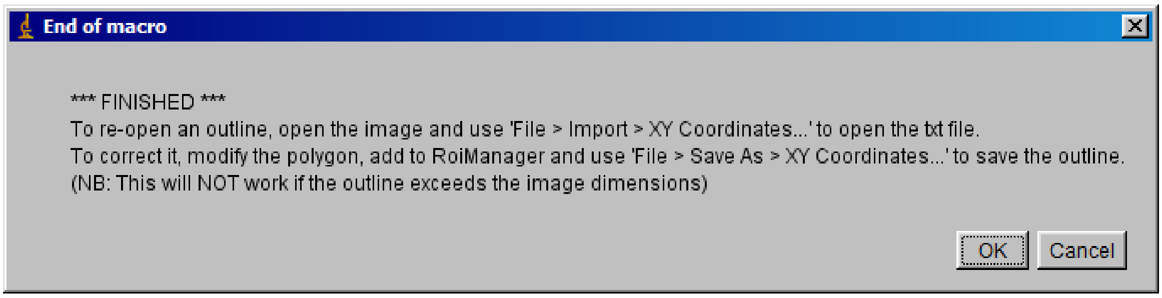
The original image is automatically enlarged to allow drawing outside the margins of the original image (useful if parts of the leaf are missing in some frames, e.g. due to growth). As mentioned in the dialog box, **the first two points of each leaf outline shall be each side of the petiole base**.



When finished, press the key 'T' and the leaf outline will be added in the ROI Manager:



Press 'OK', and continue to digitize the leaf outline of each image. When all images are processed, this final dialog box will appear:



*** FINISHED ***
To re-open an outline, open the image and use 'File > Import > XY Coordinates...' to open the txt file.
To correct it, modify the polygon, add to RoiManager and use 'File > Save As > XY Coordinates...' to save the outline.
(NB: This will NOT work if the outline exceeds the image dimensions)

As indicated in the text box, it is possible to modify any leaf outline by re-open the image, and then open the associated **.txt* file using 'File > Import > XY Coordinates...'. Once the polygon is modified, press 'T' to add to ROI Manager and save as XY Coordinates. **It will not work for those outlines which have been drawn outside the margins of the image.**

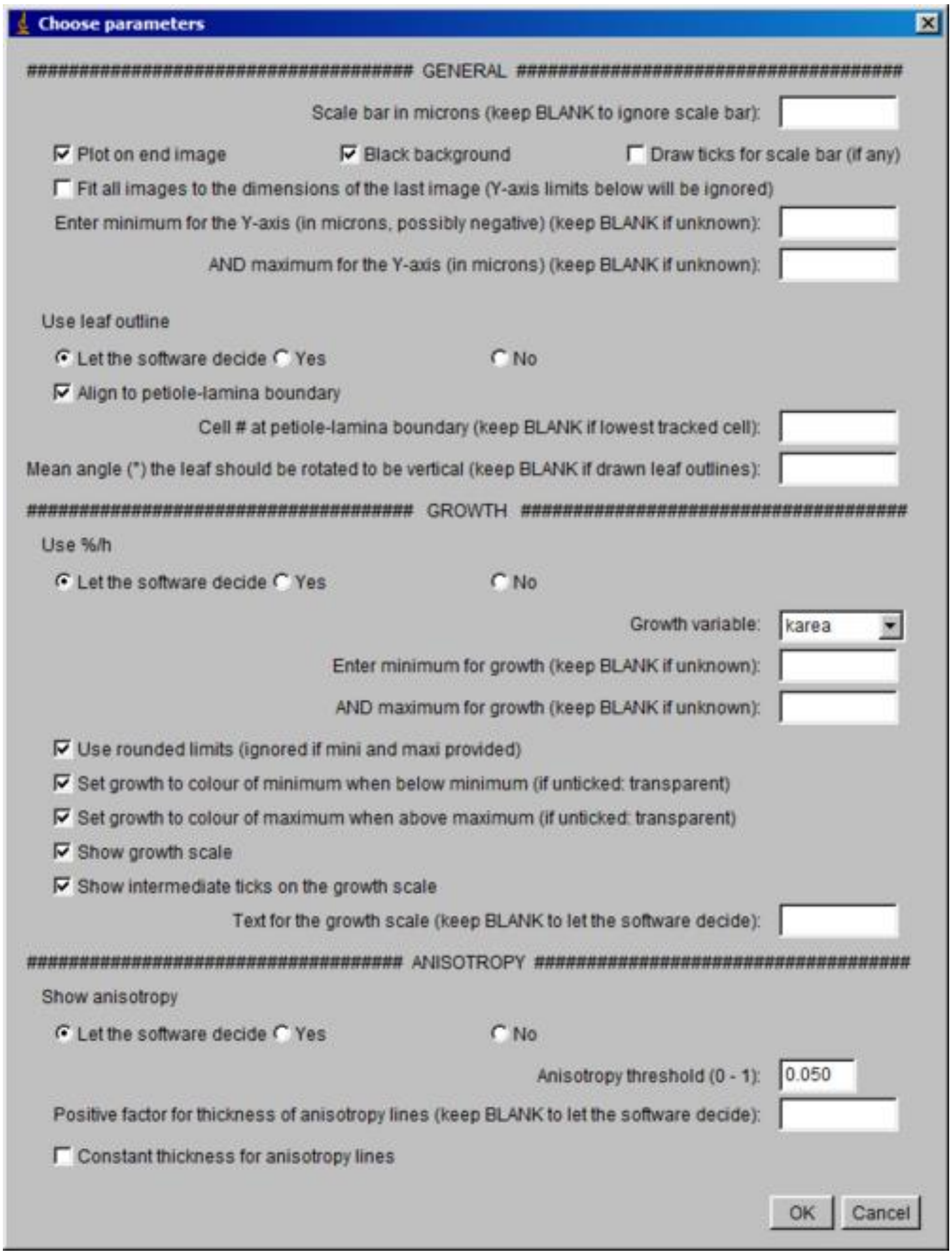
NB: the other macros will use these leaf outlines to re-orient the leaf as follows. A first reference point will be set as the middle of the segment formed by the two points at the petiole base. A second reference point will be computed as the centroid of the leaf outline. Then a rotation will be performed so that i) the first and the second reference points are aligned vertically and ii) the first point is basal and the second point is distal. k_{ml} will be computed as the growth rate parallel to this axis, while k_{per} will be computed as the growth rate perpendicular to this axis.

Replot growth from PointTracker

- ImageJ macro: *Replot_Growth_From_PointTracker.txt*
- Javascript: *Replot_Growth_From_PointTracker.js*
- R script: *Replot_Growth_From_PointTracker.r*

This macro **replots growth** between each image used to generate the **.xls* file (processed data). This is the only macro of the toolset which does not use the **.csv* file (raw data). **Use this macro only if you have already plotted growth from PointTracker.** Note this is not available in the example dataset. If you want to plot growth from scratch, use the macro Plot growth instead.

Once the **.xls* file has been selected, a dialog box of parameters is opened:



Choose parameters

GENERAL

Scale bar in microns (keep BLANK to ignore scale bar):

☑ Plot on end image ☑ Black background □ Draw ticks for scale bar (if any)

□ Fit all images to the dimensions of the last image (Y-axis limits below will be ignored)

Enter minimum for the Y-axis (in microns, possibly negative) (keep BLANK if unknown):

AND maximum for the Y-axis (in microns) (keep BLANK if unknown):

Use leaf outline

● Let the software decide ○ Yes ○ No

☑ Align to petiole-lamina boundary

Cell # at petiole-lamina boundary (keep BLANK if lowest tracked cell):

Mean angle (°) the leaf should be rotated to be vertical (keep BLANK if drawn leaf outlines):

GROWTH

Use %/h

● Let the software decide ○ Yes ○ No

Growth variable: karea

Enter minimum for growth (keep BLANK if unknown):

AND maximum for growth (keep BLANK if unknown):

☑ Use rounded limits (ignored if mini and maxi provided)

☑ Set growth to colour of minimum when below minimum (if unticked: transparent)

☑ Set growth to colour of maximum when above maximum (if unticked: transparent)

☑ Show growth scale

☑ Show intermediate ticks on the growth scale

Text for the growth scale (keep BLANK to let the software decide):

ANISOTROPY

Show anisotropy

● Let the software decide ○ Yes ○ No

Anisotropy threshold (0 - 1): 0.050

Positive factor for thickness of anisotropy lines (keep BLANK to let the software decide):

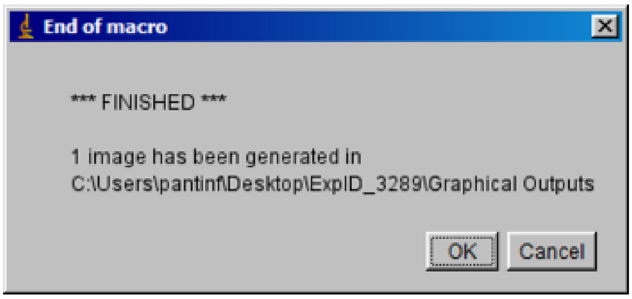
□ Constant thickness for anisotropy lines

Clicking 'OK' without changing any parameter will make the R script executes with the default values. The main parameters the user might wish to change are:

- The scale bar as a number in micrometres.
- The checkbox 'Fit all images to the dimensions of the last image'. If ticked, all images will be plotted at the same scale, namely the one of the last image plotted.
- The identifier (number) of the cell at the petiole-lamina boundary. If not entered, alignment will be done based on the lowest tracked vertex.
- The growth variable: k_{area}, k_{maj}, k_{min}, θ (angle of main direction of growth to horizontal), ϕ (rotation rate), anisotropy, $k_{perpendicular\ to\ midline}$, $k_{parallel\ to\ midline}$.

Where an entered value is irrelevant, the user will be prompted to enter the faulty parameter again.

When the macro finishes, a dialog box is displayed with the count of images saved in the subfolder ***\Graphical Outputs\*** which will be automatically created in the project directory.



End of macro

*** FINISHED ***

1 image has been generated in
C:\Users\pantinf\Desktop\ExpID_3289\Graphical Outputs

The images will be saved with the prefix **'Growth from PT__'**. Please note that **if any graphical output with the same name is already present, it will be overwritten**.

Note that a few differences with the images produced in PointTracker:

- The leaf is re-oriented upwards.
- The original image of the leaf is not displayed.
- Leaf outline may or may not be displayed. **If you choose not to display the leaf outline, make sure you enter the right angle to re-orient the leaf upwards.**
- The thickness of the lines representing the main direction of growth in each cell is directly proportional to the anisotropy, except if the checkbox 'Constant thickness for anisotropy lines' is ticked.

Below are some examples of graphical outputs obtained using an **.xls* file containing growth computed between four images. In the first line, all parameters were kept to default, except that the scale bar was set at 100 µm. In the second line, the scale bar was set at 500 µm with ticks, the checkbox 'Black background' was unticked, the check box 'Fit all images to the dimensions of the last image' was ticked, and the growth variable was set to k_{ml} instead of k_{area}. Note that anisotropy is plotted by default only when k_{area} is selected.

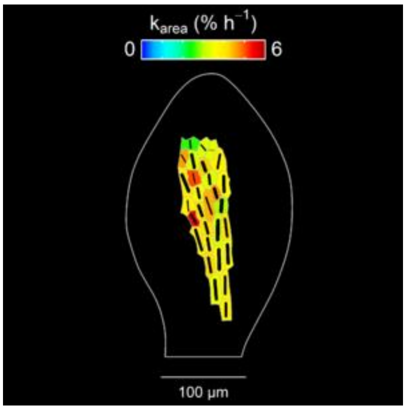


Image 1 to Image 2

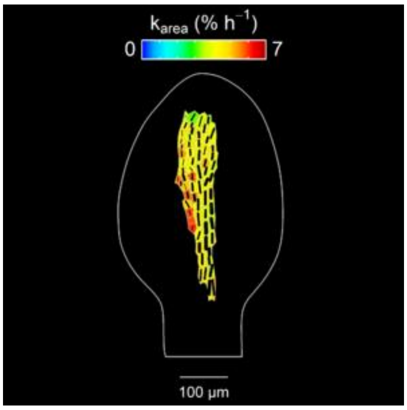


Image 2 to Image 3

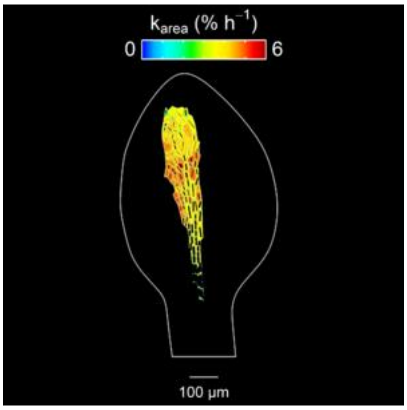


Image 3 to Image 4

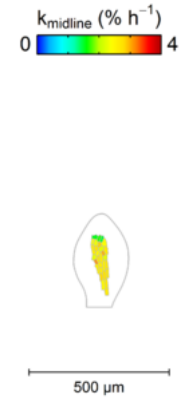


Image 1 to Image 2

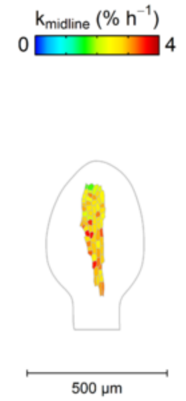


Image 2 to Image 3

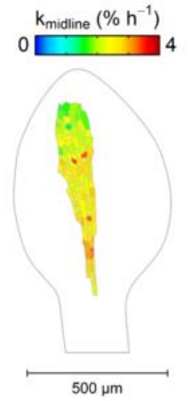
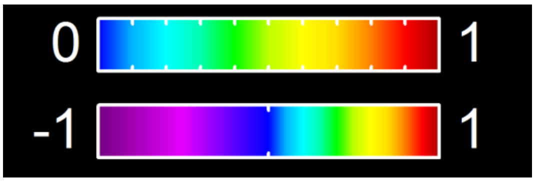


Image 3 to Image 4

Note that the colour scale used in Track 'n' R is now the R translation of Richard's *rainbowMap.m*. When there are only positive values, it is therefore exactly the same as in PointTracker, as well as in GFtbox when using the Matlab code:

```
        m = leaf_plotoptions( m, 'zerowhite', false );
```

When there are negative values, it will extend to purple as in GFtbox, while PointTracker does not.

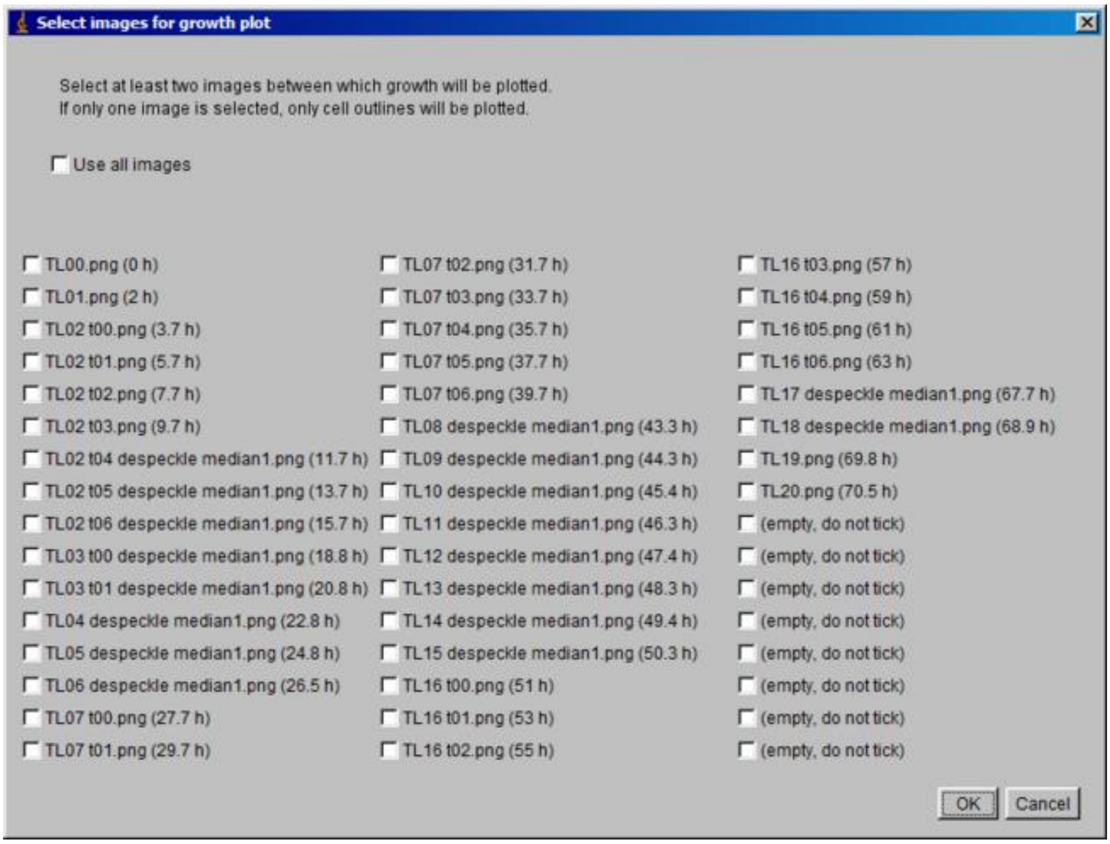


Plot growth

> ImageJ macro: *Plot_Growth.txt*
> Javascript: *Plot_Growth.js*
> R script: *Plot_Growth.r*

This macro recomputes and plots **growth** between each image of a selected set. It uses the raw data stored in the **.csv* file. Note that the 'BackwardDense' method of PointTracker is used to calculate growth, with 100 points and exponential correction. Thus, the graphical outputs are exactly the same as those produced using Replot growth from PointTracker (if the **.xls* file was generated using the 'BackwardDense' method with 100 points and exponential correction).

Once the **.csv* file has been selected, a dialog box is opened containing the list of available images:



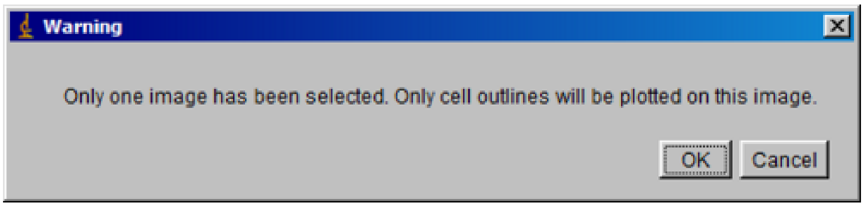
Select images for growth plot

Select at least two images between which growth will be plotted.
If only one image is selected, only cell outlines will be plotted.

□ Use all images

□ TL00.png (0 h)
□ TL01.png (2 h)
□ TL02 t00.png (3.7 h)
□ TL02 t01.png (5.7 h)
□ TL02 t02.png (7.7 h)
□ TL02 t03.png (9.7 h)
□ TL02 t04 despeckle median1.png (11.7 h)
□ TL02 t05 despeckle median1.png (13.7 h)
□ TL02 t06 despeckle median1.png (15.7 h)
□ TL03 t00 despeckle median1.png (18.8 h)
□ TL03 t01 despeckle median1.png (20.8 h)
□ TL04 despeckle median1.png (22.8 h)
□ TL05 despeckle median1.png (24.8 h)
□ TL06 despeckle median1.png (26.5 h)
□ TL07 t00.png (27.7 h)
□ TL07 t01.png (29.7 h)
□ TL07 t02.png (31.7 h)
□ TL07 t03.png (33.7 h)
□ TL07 t04.png (35.7 h)
□ TL07 t05.png (37.7 h)
□ TL07 t06.png (39.7 h)
□ TL08 despeckle median1.png (43.3 h)
□ TL09 despeckle median1.png (44.3 h)
□ TL10 despeckle median1.png (45.4 h)
□ TL11 despeckle median1.png (46.3 h)
□ TL12 despeckle median1.png (47.4 h)
□ TL13 despeckle median1.png (48.3 h)
□ TL14 despeckle median1.png (49.4 h)
□ TL15 despeckle median1.png (50.3 h)
□ TL16 t00.png (51 h)
□ TL16 t01.png (53 h)
□ TL16 t02.png (55 h)
□ TL16 t03.png (57 h)
□ TL16 t04.png (59 h)
□ TL16 t05.png (61 h)
□ TL16 t06.png (63 h)
□ TL17 despeckle median1.png (67.7 h)
□ TL18 despeckle median1.png (68.9 h)
□ TL19.png (69.8 h)
□ TL20.png (70.5 h)
□ (empty, do not tick)
□ (empty, do not tick)
□ (empty, do not tick)
□ (empty, do not tick)
□ (empty, do not tick)
□ (empty, do not tick)
□ (empty, do not tick)
□ (empty, do not tick)

Selecting 'Use all images' will plot growth between each image of the list. If only one image is selected, a message will be displayed, warning that only cell outlines will be plotted on this image:



Warning

Only one image has been selected. Only cell outlines will be plotted on this image.

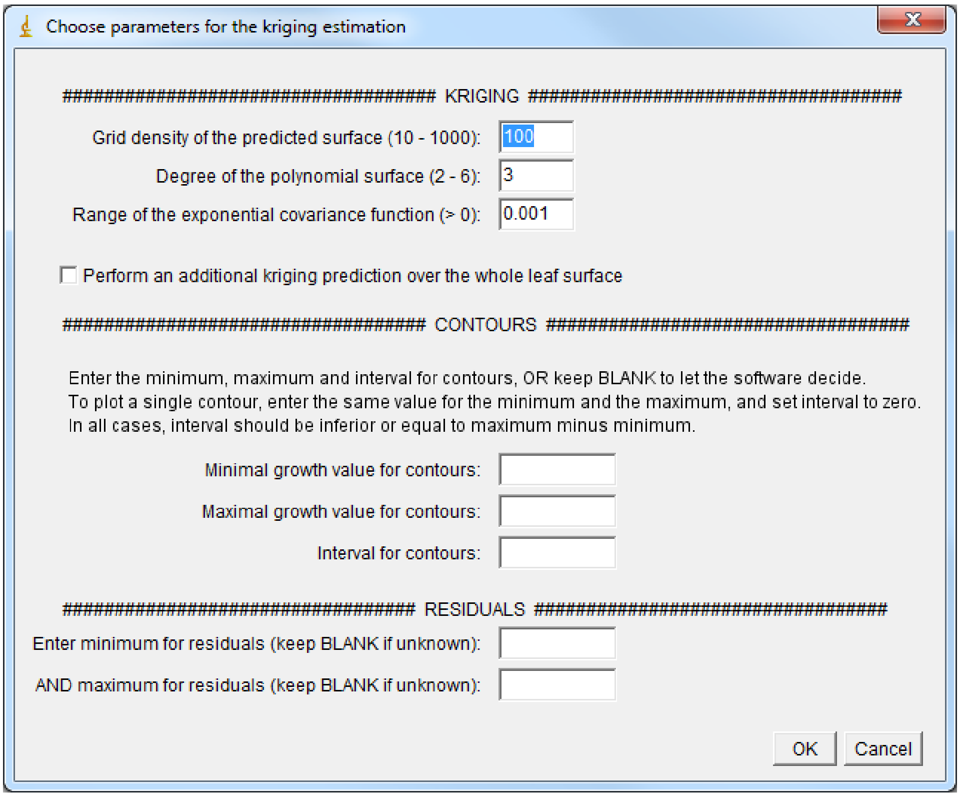
The next steps of the macro are the same as in Replot growth from PointTracker. Graphical outputs are also the same. The images will be saved with the prefix **'Growth__'**.

Plot contours

> ImageJ macro: *Plot_Kriging.txt*
> Javascript: *Plot_Kriging.js*
> R script: *Plot_ Kriging.r*

This macro performs and plots a **surface fitting of growth** between each image of a selected set, and draws the **isocontours** of the surface at selected values. It uses the raw data stored in the **.csv* file. The trend kriging surface (with a specified polynomial degree) is fitted by generalized least-squares using the exponential covariance function (with a specified range parameter). **Individual values are also estimated for each cell, as well as the residuals with the measured value.**

Once launched, this macro opens the same dialog boxes as in the Plot growth macro. Nevertheless, at least two images need to be selected. An additional dialog box will be displayed with parameters specific to the kriging procedure.



Choose parameters for the kriging estimation

###################################### KRIGING ######################################

Grid density of the predicted surface (10 - 1000): 100

Degree of the polynomial surface (2 - 6): 3

Range of the exponential covariance function (> 0): 0.001

☐ Perform an additional kriging prediction over the whole leaf surface

##################################### CONTOURS #####################################

Enter the minimum, maximum and interval for contours, OR keep BLANK to let the software decide.
To plot a single contour, enter the same value for the minimum and the maximum, and set interval to zero.
In all cases, interval should be inferior or equal to maximum minus minimum.

Minimal growth value for contours:

Maximal growth value for contours:

Interval for contours:

#################################### RESIDUALS ####################################

Enter minimum for residuals (keep BLANK if unknown):

AND maximum for residuals (keep BLANK if unknown):

Clicking 'OK' without changing any parameter will make the R script executes with the default values. Where an entered value is irrelevant, the user will be prompted to enter the faulty parameter again.

The user might wish to change the grid density for kriging prediction. Default is 100, as a good compromise between quality of the prediction and time required to perform and plot the estimated surface. 1000 will give slightly better visual outputs, but will take much longer. The polynomial degree and the range parameter can also be changed. Increasing their value will make the surface more convoluted. It is also possible to extrapolate the kriging to the parts of the leaves which were not tracked, but the result is generally not relevant (except if the largest part of the leaf was tracked). If the corresponding box is ticked, an additional map will be generated.

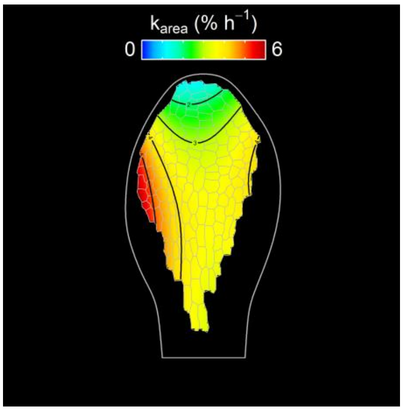
The user may also wish to change the levels of the plotted contours. If missing (the default), the levels plotted will be all multiples of 1 %·h^{-1} (or 0.01 h^{-1}) within the range of the maximum and minimum of the variable. To ignore contours, just enter values that are out of range.

Finally, it is also possible to set the limits for the scale of the residuals. It is advised to run the macro first without entering values to see the data, and then adjust the limits if relevant (generally we want a symmetric scale such as between -1 and 1).

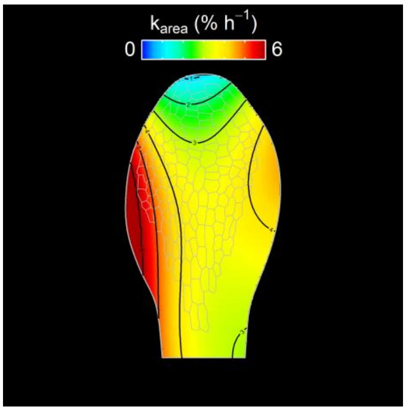
The images will be saved with the prefix **'Contours__'**. 3 to 4 maps per image pair are generated:

- the fitted surface over the region tracked:
  Contours__(name of growth variable)__TrackedRegion__....png
- the fitted value averaged for each tracked cell:
  Contours__(name of growth variable)__Cell__....png
- the difference between the observed and fitted value for each tracked cell:
  Contours__(name of growth variable)__CellResidual__....png
- if ticked, the fitted surface extrapolated over the whole leaf:
  Contours__(name of growth variable)__WholeLeaf__....png

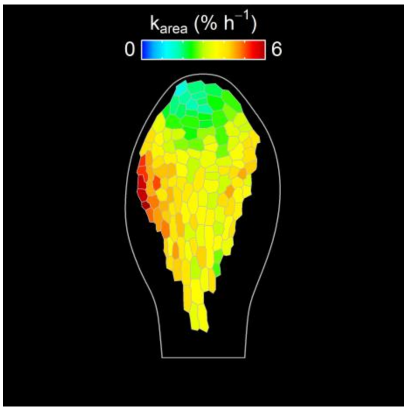
Below are graphical outputs using k_{area} computed between the 3rd and 4th image of the example dataset. All parameters were maintained to their default value, except that growth was capped between 0 and 6 %·h^{-1}, and that additional whole-leaf extrapolation was selected. Note that the map of measured k_{area} (bottom-left) was obtained using the Plot growth macro applied on the same dataset. The map of the residuals is the difference between measured and cell-averaged fitted k_{area}.



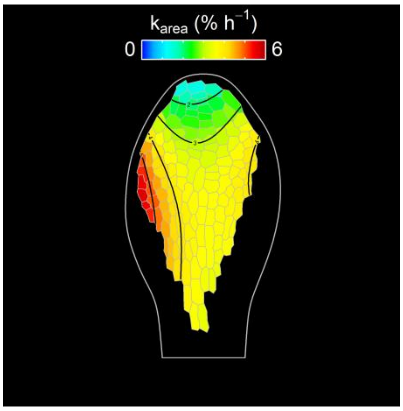
Tracked region



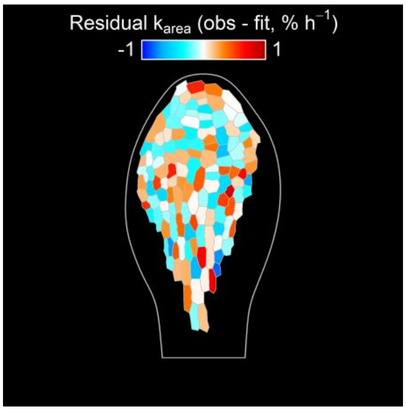
Whole leaf extrapolation



Measured



Cell-averaged



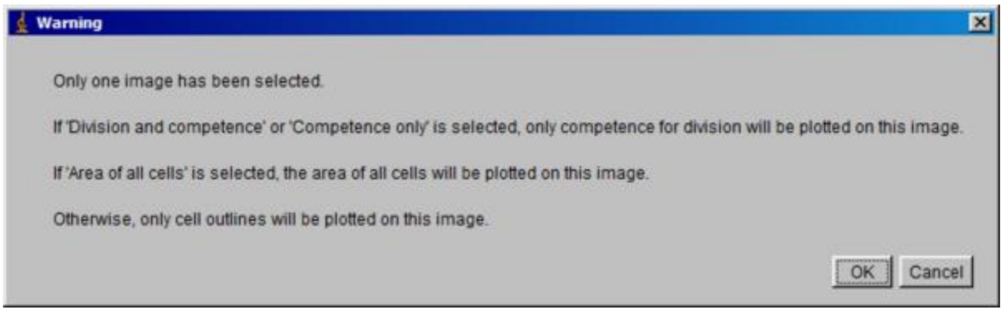
Residuals

Plot cell division

> ImageJ macro: *Plot_Division.txt*
> Javascript: *Plot_Division.js*
> R script: *Plot_Division.r*

This macro can plot not only **cell division** and/or **competence for division** between each image of a selected set, but also **cell progeny**, **area of all cells**, **cell area at division**, **symmetry of division** or **duration of cell cycle**. It uses the raw data stored in the **.csv* file.

Once launched, this macro opens the same dialog boxes as in the 'Plot growth' macro. Note that if only one image is selected, a warning message will be displayed:



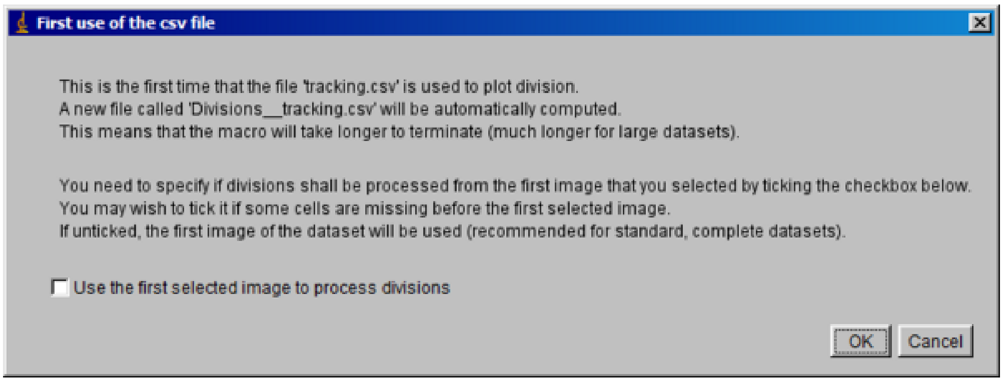
Only one image has been selected.

If 'Division and competence' or 'Competence only' is selected, only competence for division will be plotted on this image.

If 'Area of all cells' is selected, the area of all cells will be plotted on this image.

Otherwise, only cell outlines will be plotted on this image.

If this is the first time the user plots division from the selected PointTracker **.csv* file using this macro (or 'Plot intervals' or 'Analyze lineages'), this dialog box will open:



This is the first time that the file 'tracking.csv' is used to plot division.
A new file called 'Divisions__tracking.csv' will be automatically computed.
This means that the macro will take longer to terminate (much longer for large datasets).

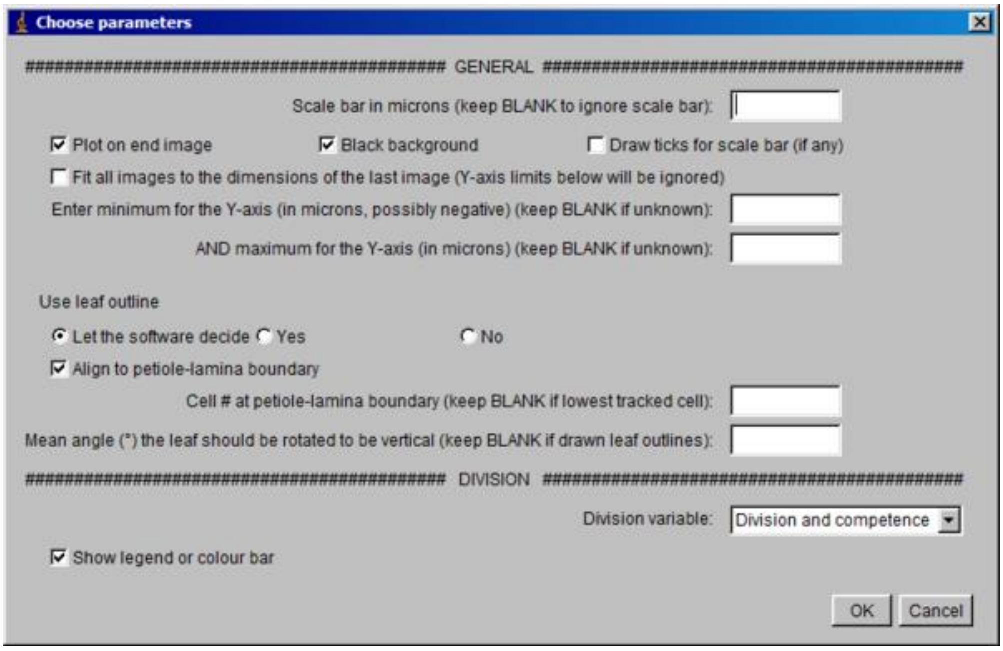
You need to specify if divisions shall be processed from the first image that you selected by ticking the checkbox below.
You may wish to tick it if some cells are missing before the first selected image.
If unticked, the first image of the dataset will be used (recommended for standard, complete datasets).

□ Use the first selected image to process divisions

It warns the user that the software needs to generate a file named *Divisions__<name of PointTracker csv file>.csv*, which will be saved in the subfolder *Divisions*\. This file will be used every time the macro is called; this means that the macro will be quite long to terminate only on the first time it is run for each selected PointTracker **.csv* file. **Note that if your dataset contains cell lineages which do not exist from the first image of the dataset, you should tick the checkbox 'Use the first selected image to process divisions' or the macro will fail.**

WARNING: if for any reason you modified the PointTracker **.csv* file, e.g. by adding new cells or modifying existing cells, Track 'n' R won't be aware of this and won't prompt you to generate the file *Divisions__<name of PointTracker csv file>.csv* again. This will generate an error when you call the macro. The solution is to manually delete the file *Divisions__<name of PointTracker csv file>.csv* before running the macro so that the Track 'n' R will prompt you again to generate it.

Then a dialog box opens with a reduced set of parameters:



Choose parameters

GENERAL

Scale bar in microns (keep BLANK to ignore scale bar):

☑ Plot on end image ☑ Black background □ Draw ticks for scale bar (if any)

□ Fit all images to the dimensions of the last image (Y-axis limits below will be ignored)

Enter minimum for the Y-axis (in microns, possibly negative) (keep BLANK if unknown):

AND maximum for the Y-axis (in microns) (keep BLANK if unknown):

Use leaf outline

● Let the software decide ○ Yes ○ No

☑ Align to petiole-lamina boundary

Cell # at petiole-lamina boundary (keep BLANK if lowest tracked cell):

Mean angle (°) the leaf should be rotated to be vertical (keep BLANK if drawn leaf outlines):

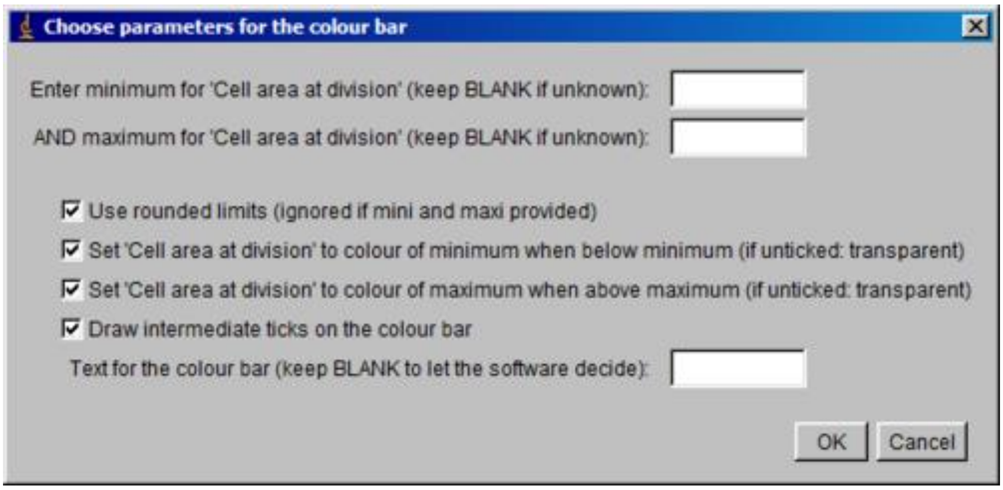
DIVISION

Division variable: Division and competence

☑ Show legend or colour bar

Clicking 'OK' without changing any parameter will make the R script executes with the default values.

If the user selects another variable than division and/or competence, another dialog box will open with parameters to control the colour palette and its range:



Choose parameters for the colour bar

Enter minimum for 'Cell area at division' (keep BLANK if unknown):

AND maximum for 'Cell area at division' (keep BLANK if unknown):

☑ Use rounded limits (ignored if mini and maxi provided)

☑ Set 'Cell area at division' to colour of minimum when below minimum (if unticked: transparent)

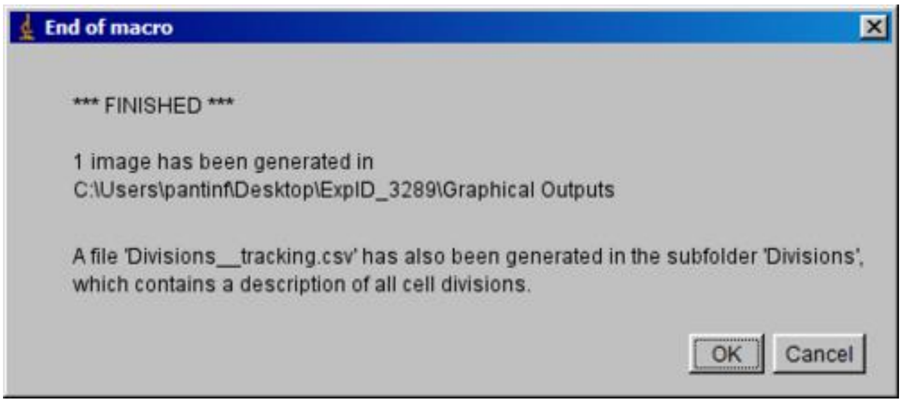
☑ Set 'Cell area at division' to colour of maximum when above maximum (if unticked: transparent)

☑ Draw intermediate ticks on the colour bar

Text for the colour bar (keep BLANK to let the software decide):

Again, clicking 'OK' without changing any parameter will make the R script executes with the default values.

The images will be saved with the prefix **'Division__'**. A dialog box is displayed with the count of images saved in the subfolder ***\Graphical Outputs***, and a statement that the file ***Divisions__<name of PointTracker csv file>.csv*** has been correctly saved in the subfolder ***\Divisions*** (first use only):



End of macro

*** FINISHED ***

1 image has been generated in
C:\Users\pantinf\Desktop\ExpID_3289\Graphical Outputs

A file 'Divisions__tracking.csv' has also been generated in the subfolder 'Divisions',
which contains a description of all cell divisions.

Below are some graphical outputs obtained with the provided example datasets, in which not all cells (especially the ones at the leaf margin) are visible from the beginning. The file *tracking_TL01-TL02_divide.csv* was used to generate the plot 'Image 1 (Initial competence)' for which only the first image was ticked, as well as the plot 'Image 1 to Image 2' in which only images 1 and 2 were ticked. The file *tracking_TL02-TL03_divide.csv* was used to generate the plot 'Image 2 to Image 3' in which only images 2 and 3 were ticked, etc. By default, both division and competence are plotted:

- Cells coloured in **light green** (**= cell division**) are the ones that have undergone cell division between the first and the second image.
- Cells coloured in **dark green** (**= competence for cell division**) are the ones that will divide again between the second image and the last image of the dataset. Note that if a cell has divided and is competent, it will be displayed with the colour of division.
- Cells with the colour of the text (white if the background is black, and vice-versa) are the cells that did not divide between the first and the second image AND will not divide again (at least until the last image of the dataset).

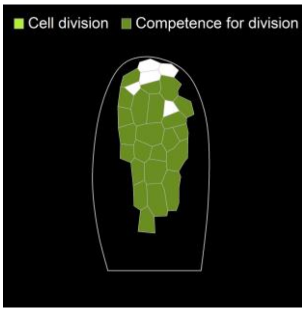


Image 1 (Initial competence)

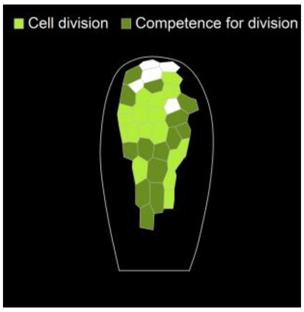


Image 1 to Image 2

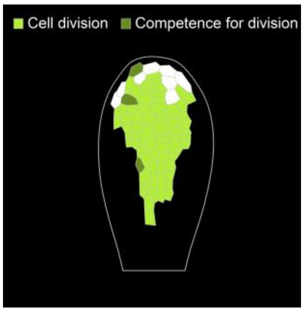


Image 2 to Image 3

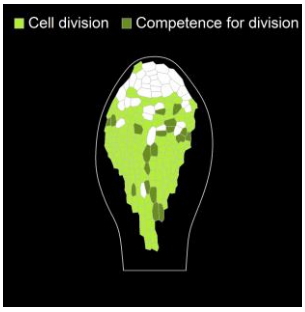


Image 3 to Image 4

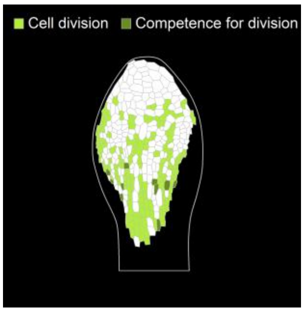


Image 4 to Image 5

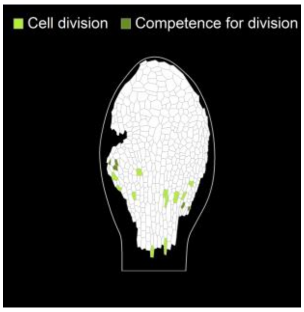


Image 5 to Image 6

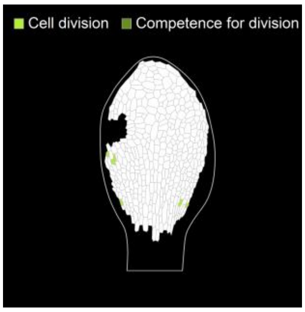


Image 6 to Image 7

Examples of other variables from another dataset are shown below:

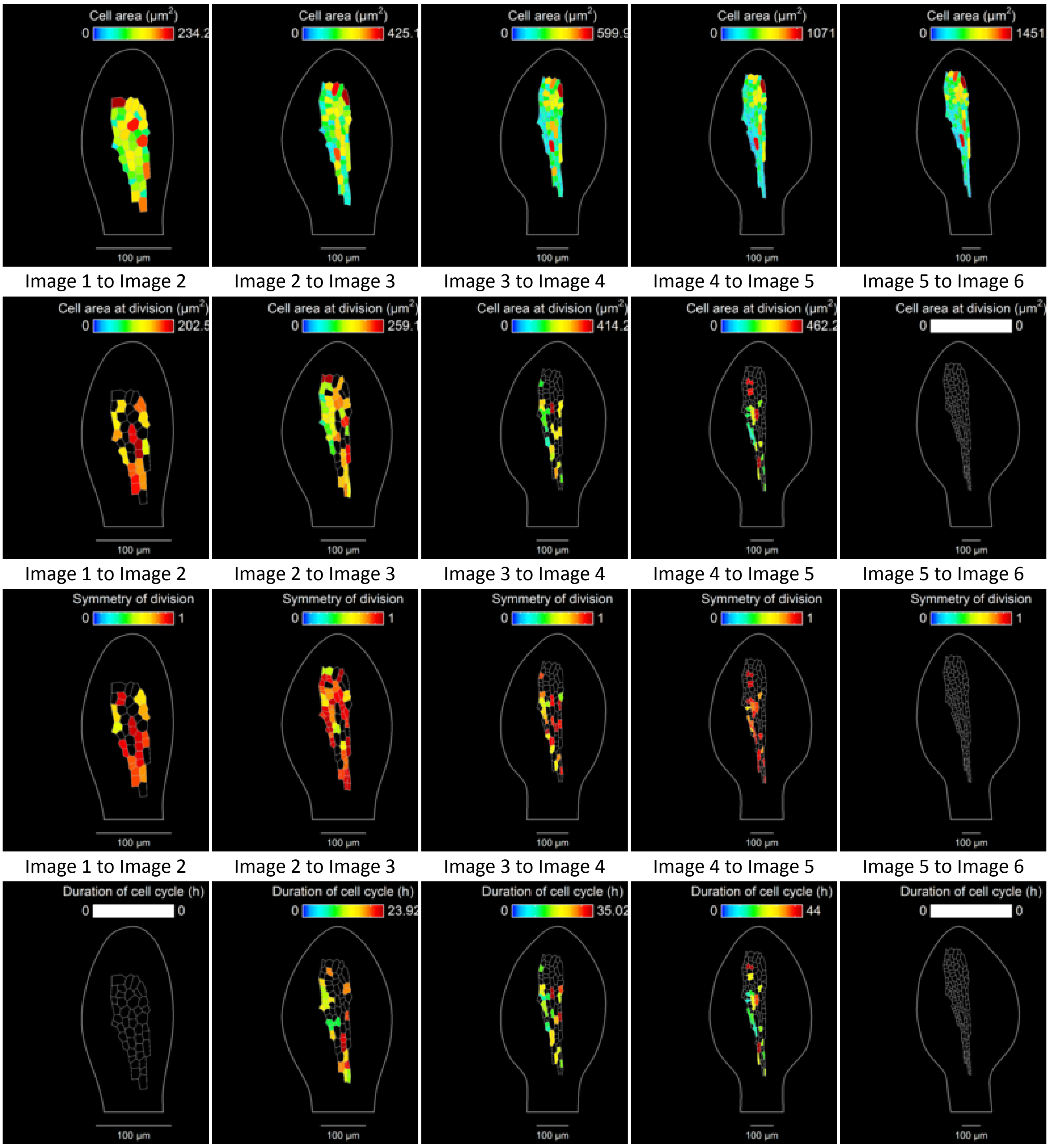


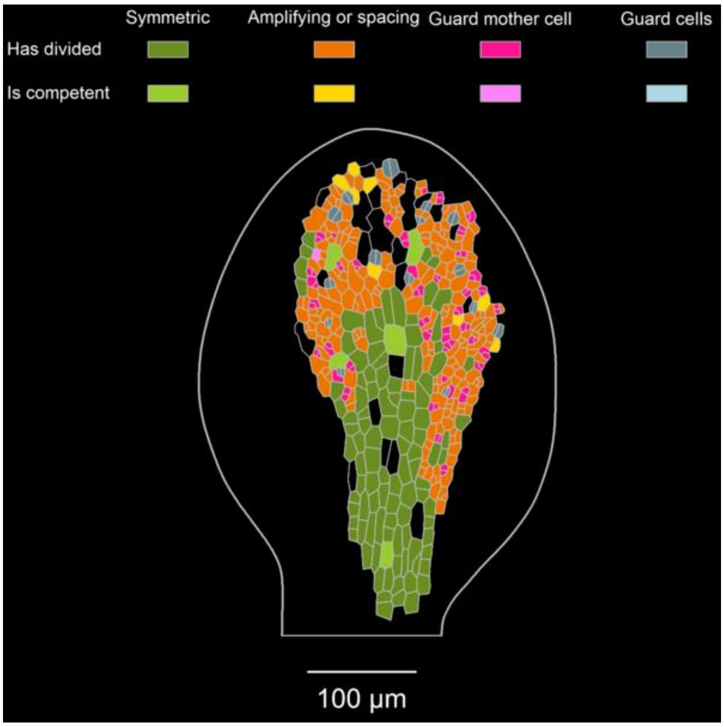
Image 1 to Image 2 Image 2 to Image 3 Image 3 to Image 4 Image 4 to Image 5 Image 5 to Image 6

Image 1 to Image 2 Image 2 to Image 3 Image 3 to Image 4 Image 4 to Image 5 Image 5 to Image 6

Image 1 to Image 2 Image 2 to Image 3 Image 3 to Image 4 Image 4 to Image 5 Image 5 to Image 6

Image 1 to Image 2 Image 2 to Image 3 Image 3 to Image 4 Image 4 to Image 5 Image 5 to Image 6

There is also the possibility to plot division and/or competence **by cell type or class**. Below is an example where classes were defined as 'Symmetric', 'Amplifying or spacing', 'Guard mother cell', and 'Guard cells', but **it can be done with any class that the user defines**.



To do that, you need to have created a file called **'DivisionClass.xlsx'** in the ***\Divisions\*** subfolder. A template can be found in the subfolder *...\ImageJ\macros\Track 'n' R\Help\Example of Excel sheet for division class\*. Two sheets are required:

- the sheet 'DivClass' in which each division, represented by its mother in the column 'Cell', shall be manually associated with a class number (an integer starting from 1) in the column 'Class'. Only these two columns are useful (in purple in the template). Do not change their names. This sheet is basically a copy-and-paste of the file *Divisions__<name of PointTracker csv file>.csv*, sorted by lineage and in which columns have been rearranged (for convenience, but this is not required).
- the sheet 'ClassName' in which each class number (column 'Class') should have a class name (column 'ClassName'), a colour for division (column 'DivisionColor') and a colour for competence (column 'CompetenceColor'). Names of R colours can be found e.g. here.

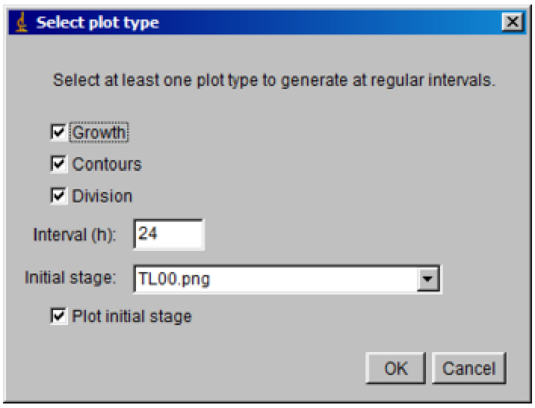
Note that the name of the xlsx file, sheets and required columns shall not be modified.

Plot growth, contours or division at regular intervals

> ImageJ macro: *Plot_Intervals.txt*
> Javascript: *Plot_Intervals.js*
> R script: *Plot_Intervals.r*

This macro plots **growth, contours and/or division at regular intervals** from a selected initial image. The images used for the plots will be **automatically selected as a function of the set interval**. It uses the raw data stored in the **.csv* file. This is useful if your **.csv* file extends over a long period of time without lineages added or deleted (e.g. this not usable in the provided example dataset).

The first dialog box allows the user to define the parameters specific to this macro:



Select plot type

Select at least one plot type to generate at regular intervals.

☑ Growth
☑ Contours
☑ Division

Interval (h): 24

Initial stage: TL00.png

☑ Plot initial stage

The user must:

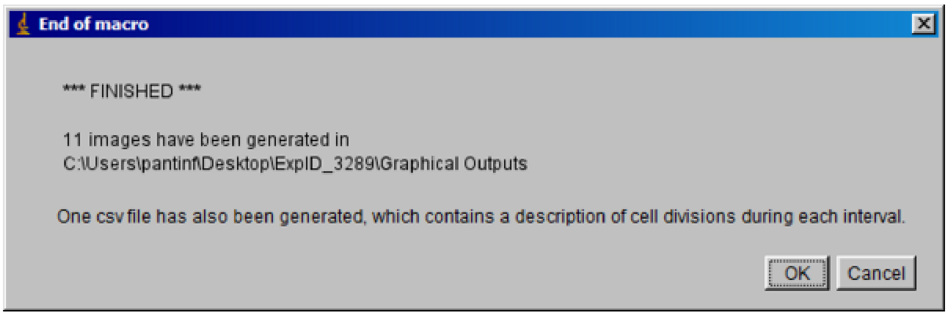
- select at least one plot type among growth, contours and division. The three of them are selected by default.
- select the image from which to start the plots. Default is the first image of the dataset.
- set an interval (in hours) at which the plot type will be produced. Default is 24 h.

The user can plot the initial stage, i.e. the cell outlines at the initial stage (if growth is selected) and/or the competence of all cells at the initial stage (if division is selected). Default is True.

If not already done using one of the macro 'Plot cell division' (or 'Analyze lineages'), you will be warned that a file named *Divisions__<name of PointTracker csv file>.csv* will be created in the subfolder *Divisions*\ (see above, Plot cell division section).

A series of one or two dialog boxes allows then to select the parameters depending on which plot type the user selected. Where an entered value is irrelevant, the user will be prompted to enter the faulty parameter again. When the macro finishes, a dialog box is displayed with:

- the count of images saved in the subfolder ***Graphical Outputs*** which will be automatically created in the project directory.
- if 'division' was selected as a plot type, the name of the **.csv* file containing information about cell division within the selected intervals, and saved in the subfolder ***Divisions*** which will be automatically created in the project directory.



*** FINISHED ***

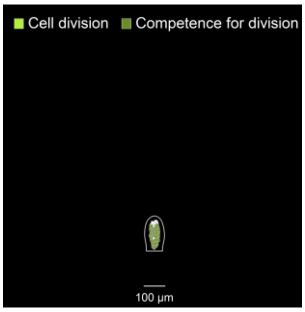
11 images have been generated in
C:\Users\pantinf\Desktop\ExpID_3289\Graphical Outputs

One csv file has also been generated, which contains a description of cell divisions during each interval.

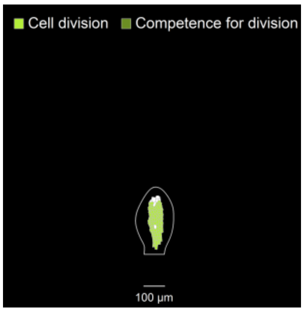
The images will be saved with the prefix **'Intervals__'**, and the **.csv* file with the prefix **'Divisions__'**.

Below are examples of outputs obtained in a single run by using the default parameters, except:

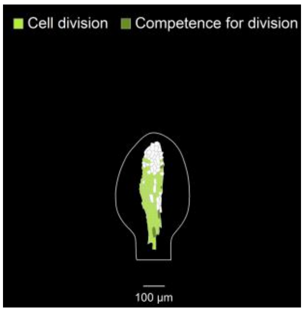
- setting the scale bar to 100 µm
- ticking the check box 'Fit all images to the dimensions of the last image'
- using k_{ml} as the growth variable
- setting the minimum and maximum for growth to 0 %·h^{-1} and 4 %·h^{-1}, respectively



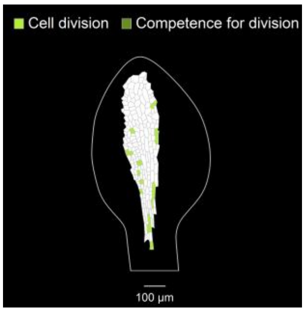
Division - Initial competence



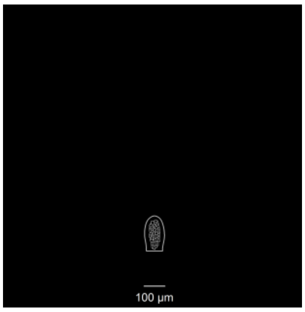
Division - 00 h to 24 h



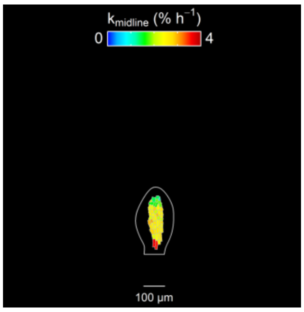
Division - 24 h to 48 h



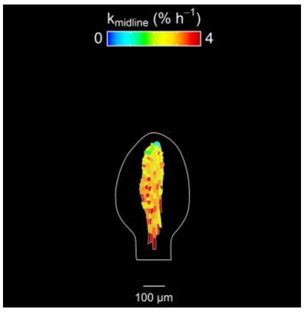
Division - 48 h to 72 h



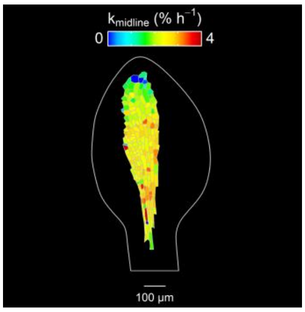
Growth - Initial cell outlines



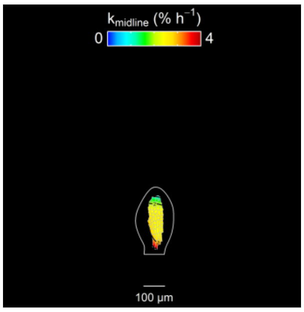
Growth - 00 h to 24 h



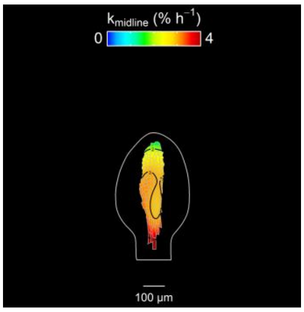
Growth - 24 h to 48 h



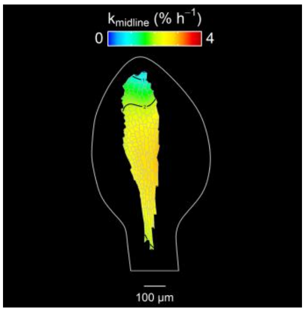
Growth - 48 h to 72 h



Contour - 00 h to 24 h



Contour - 24 h to 48 h



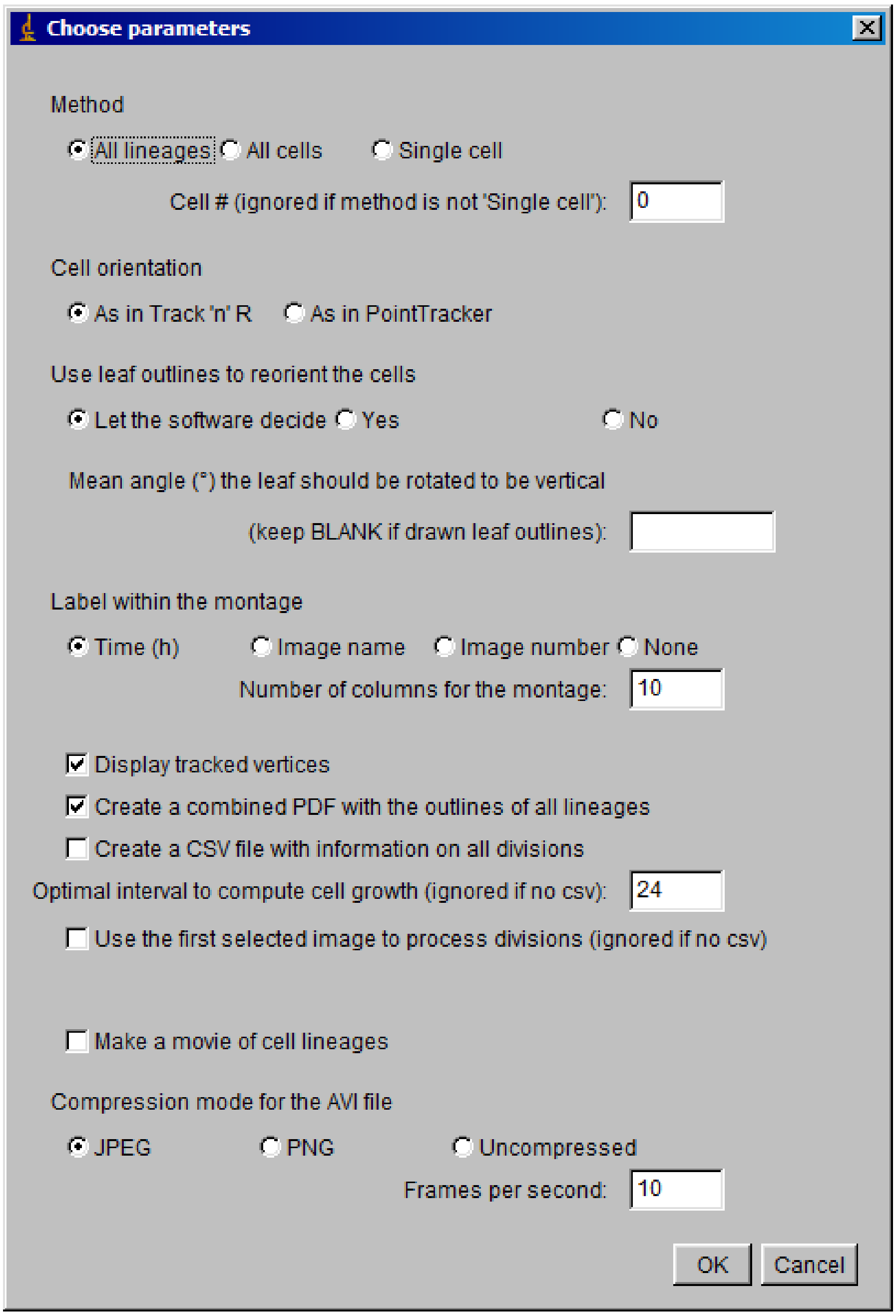
Contour - 48 h to 72 h

Note that 6 other maps will be produced (cell-averaged kriging, residuals), but not shown here.

Analyze lineages

- > ImageJ macro: *Analyze_Lineages.txt*
- > Javascript: *Analyze_Lineages.js*
- > R script: *Analyze_Lineages.r*
- > ImageJ macro called by the R script: *Analyze_Lineages_called.txt*

This macro creates **montages of cell lineages** (ancestors and descendants) with a **.pdf* of the corresponding cell numbers and outlines. After having selected the list of images to be used, the user is prompted to select the parameters for the analysis:



Choose parameters

Method

● All lineages ○ All cells ○ Single cell

Cell # (ignored if method is not 'Single cell'): 0

Cell orientation

● As in Track 'n' R ○ As in PointTracker

Use leaf outlines to reorient the cells

● Let the software decide ○ Yes ○ No

Mean angle (°) the leaf should be rotated to be vertical

(keep BLANK if drawn leaf outlines):

Label within the montage

● Time (h) ○ Image name ○ Image number ○ None

Number of columns for the montage: 10

☑ Display tracked vertices

☑ Create a combined PDF with the outlines of all lineages

□ Create a CSV file with information on all divisions

Optimal interval to compute cell growth (ignored if no csv): 24

□ Use the first selected image to process divisions (ignored if no csv)

□ Make a movie of cell lineages

Compression mode for the AVI file

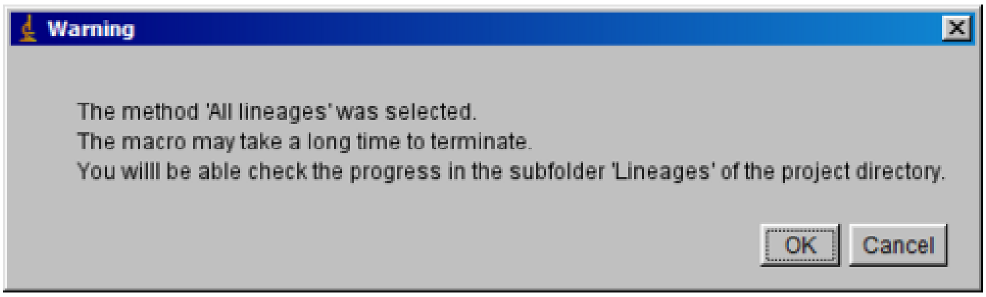
● JPEG ○ PNG ○ Uncompressed

Frames per second: 10

The user can choose:

- the method for selecting lineages. If **'All lineages'** is selected (default), the lineage of all cells present at the first image will be mounted. If **'All cells'** is selected, the lineage of all cells of the dataset will be mounted. If **'Single cell'** is selected, only the lineage of the cell # provided by the user on the line below will be mounted. There will always be one **.png* file per lineage or cell.
- the orientation of the cells: as in Track 'n' R (the leaf is reoriented upwards according to the leaf outline and the mean angle) or as in PointTracker
- the label of the montage: time, image name, image number (in the list of the processed images), or none
- the number of columns in the montage
- to display the tracked vertices (default) or not
- to generate one combined **.pdf* file for the whole analysis (default), or one **.pdf* file per lineage or cell
- to create a file called ***Divisions__<name of PointTracker csv file>.csv*** (in a subfolder ***\Divisions*** which will be automatically created in the project directory) with detailed information about all divisions, including cell growth computed over an interval before division as close as possible to the optimal interval set in the next line. Note that contrary to the *Divisions__*.csv* files generated by the macro 'Plot intervals', cell growth will be computed between (1) the image anterior to the image of division which is at closest to the optimal interval and (2) the image of division. If the time of division is so early that its difference with the optimal interval is anterior to the first tracked image, then cell growth will be computed between (1) the first image of the dataset and (2) the image posterior to the image of division which is at closest to the optimal interval. NB: tick the checkbox 'Use the first selected image to process divisions' if your dataset contains some cells which are absent at the beginning (see above, Plot cell division).
- to generate an **.avi* movie for each selected lineage

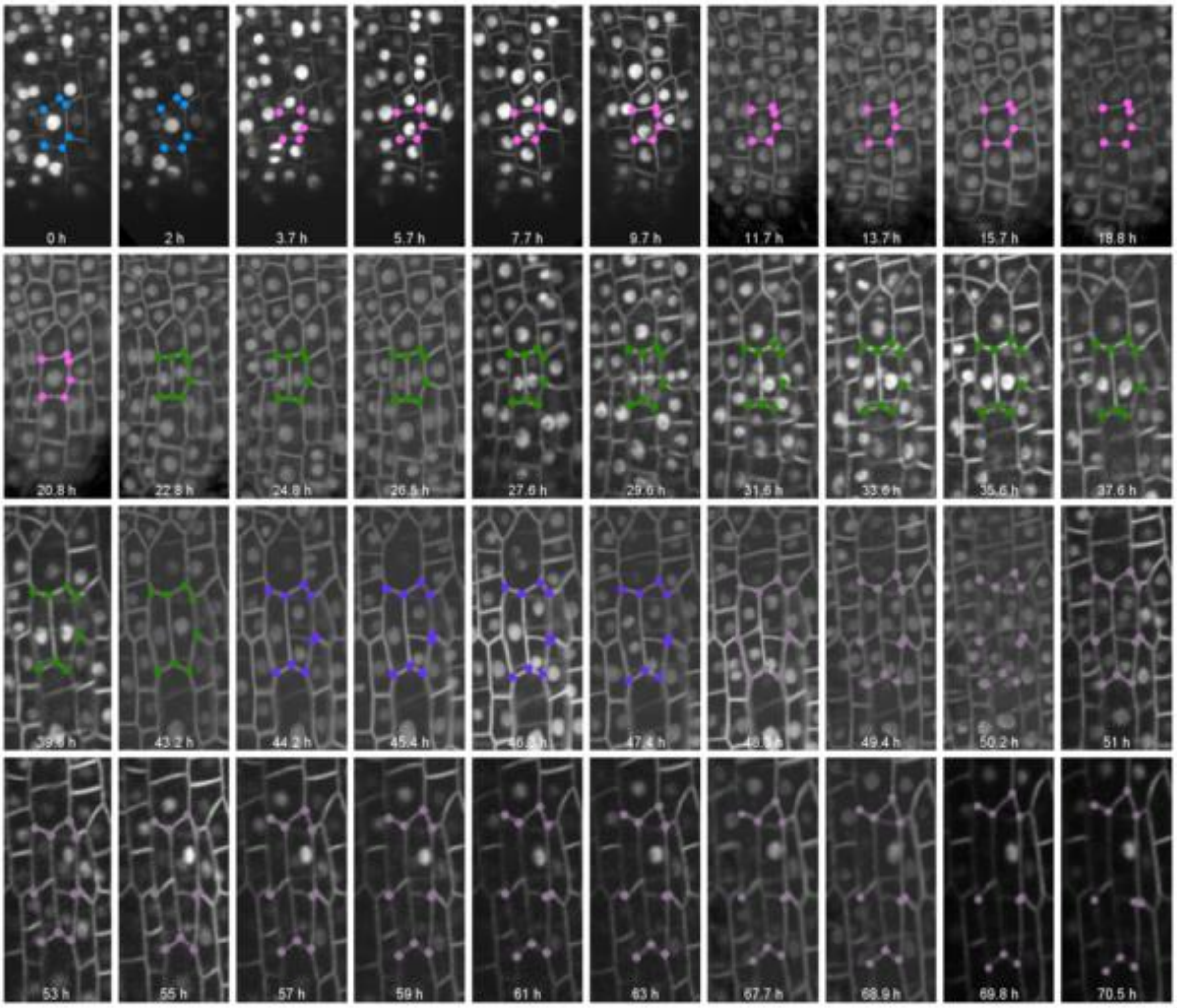
If one of the method 'All lineages' or 'All cells' has been selected, the macro can run for a long time, depending on i) the number of cells the user tracked, ii) the number of images selected and iii) whether the user asked to generate the CSV file. **For large datasets, it may take several hours.** A warning message will be displayed. Click 'OK' to continue or 'Cancel' to abort the macro:



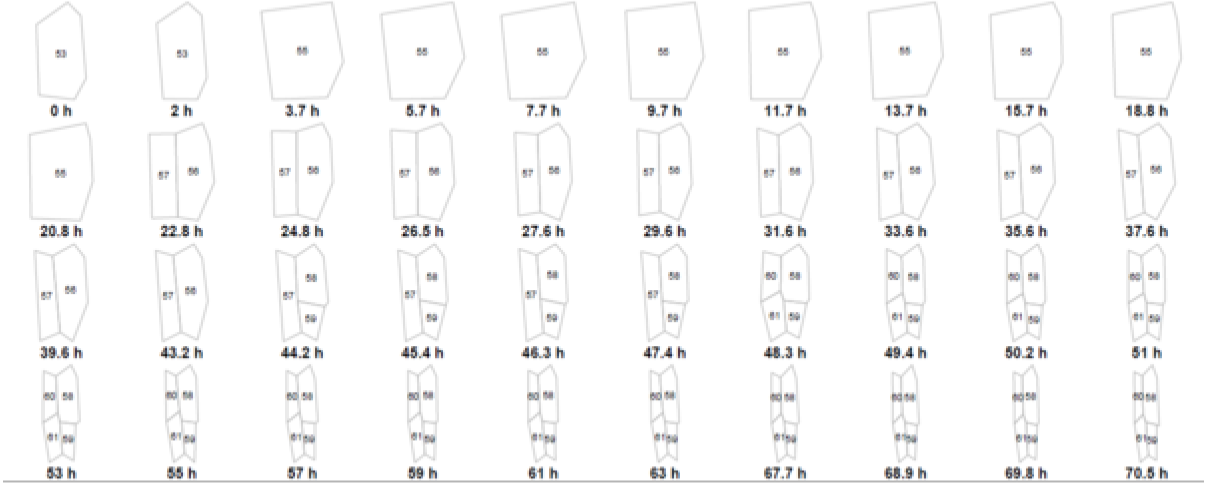
Warning

The method 'All lineages' was selected.
The macro may take a long time to terminate.
You willl be able check the progress in the subfolder 'Lineages' of the project directory.

The output is one or several **.png* and **.pdf* files saved with the prefix of the method, in a subfolder ***\Lineages\*** which will be automatically created in the project directory. Below is one example of lineage obtained with the target cell # 55, which is born on the third frame:



Montage



Cell numbers

On the montage, the colour of the vertices changes each time there is a division event in the lineage. The colour is randomly chosen, except that:

- the target cell (as in the name of the file) is always pink
- its direct mother cell (if any) is always blue
- its direct daughter cells (if any) are always green

Now the montage with cell numbers and outlines also displays the position of the cell lineage on the first and last leaf outlines.

Movies of developing cells (with or without the mark of tracked vertices) can be obtained by ticking the checkbox 'Make a movie of cell lineages'.

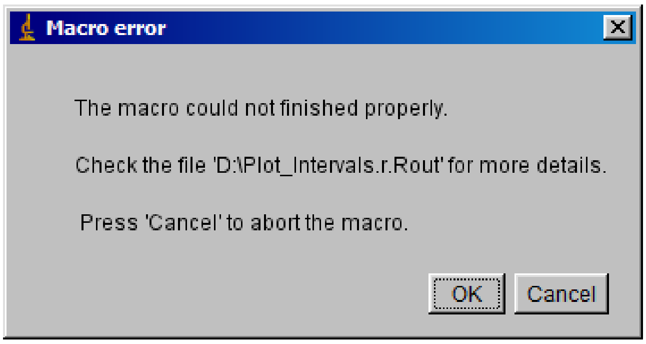
NB: contrary to the other macros, the R script called by ImageJ will itself call an ImageJ macro within another (hidden) ImageJ instance.

Help

Clicking on the Help button will open **this user guide** with your default PDF reader.

Debug

- If one macro cannot terminate and/or that this dialog box appears, then something wrong happened in R:



Macro error

The macro could not finished properly.

Check the file 'D:\Plot_Intervals.r.Rout' for more details.

Press 'Cancel' to abort the macro.

The name of an **.Rout* file (with * being the name of the R script called by the ImageJ macro) stored in the root of the D:\ drive will be displayed in this dialog box (note the macro may fail). If this happens:

1. press 'Cancel' to abort the macro
2. close ImageJ and reopen it
3. try to run the macro again to see if the bug is reproducible
4. if the bug happens again, open the **.Rout* file using Notepad to have a clue of what is the problem
5. if you cannot manage to debug this, send this file to Florent Pantin with a description of what you tried to do

Note that all macros (except 'Draw leaf outlines') may fail while ImageJ does not report the abovementioned message under some circumstances. If you have the feeling that the macro is not working anymore, if you do not see any progress in the Graphical outputs, and if the final completion message does not pop up, the R script may have fail without ImageJ could detect it. In this case, open your D:\ root and remove all the files that have been put here by Track 'n' R, e.g. if you used the Plot Interval macro:

- *D:\Parameters.txt*
- *D:\Plot_Intervals.r*
- *D:\Track_n_R.r*
- *D:\Plot_Intervals.r.Rout*

Then ImageJ should warn you that the macro has aborted. Press OK. At this point, repeat steps 2 to 5 as described above.

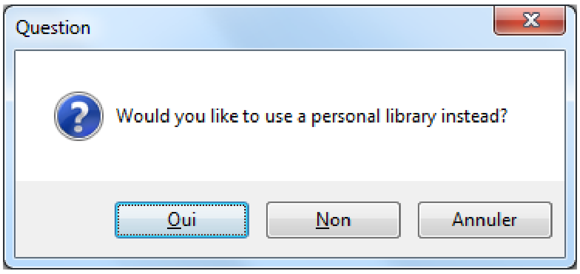
- If this is the first time you run R on your computer and that permissions to write in the C:\ drive are not enabled, the macro may fail and generate an **.Rout* file with something like this:

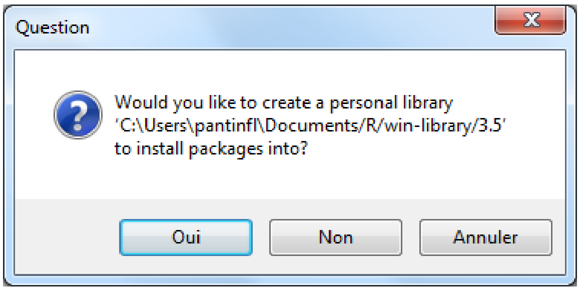
```
Warning in install.packages(pkg) :
  'lib = "C:/Program Files/R/R-3.5.1/library"' is not writable
Error in install.packages(pkg) : unable to install packages
Calls: source -> withVisible -> eval -> eval -> install.packages
Execution halted
```

then you will have you to set a path for the additional libraries that Track 'n' R needs to install. For this purpose, open R GUI (e.g. *C:\Program Files\R\R-3.5.1\bin\x64\Rgui.exe*). In the console, copy and paste the following code and press return:

```
options(repos = "http://cran.ma.imperial.ac.uk/")
for (pkg in c("XLConnect", "spatial", "sp", "splancs", "rgdal", "RSEIS",
"rgeos", "grDevices", "rJava"))
  {
  if (!pkg %in% installed.packages()[, "Package"])
    {
    install.packages(pkg)
    }
  }
```

Two windows will pop up consecutively, click 'Yes':


Would you like to use a personal library instead?


Would you like to create a personal library
'C:\Users\pantinfl\Documents/R/win-library/3.5'
to install packages into?

Wait a few minutes during package installation, and close R when finished. You should now be able to run Track 'n' R from ImageJ.

- If the **.Rout* file ends with:

```
Error in setwd(wd): cannot change working directory permission
Execution halted
```

this means that permissions are not enabled for R in the current directory. In this case it is advised to move your project directory in a writable folder such as ***My Documents***.

- The version of R installed on your computer must match what is written in the *Rpath.txt* file of the *…\ImageJ\macros\Track 'n' R* folder. If you selected the wrong file or if your version has changed since the first time you used Track 'n' R, just delete *Rpath.txt* and you will be prompted again to locate R.exe the next time you use Track 'n' R. If you cannot make the software to produce the file automatically for some reasons, you can do it manually. Create a file named *Rpath.txt* with Notepad and just type your R path in the first line, e.g.:

```
C:\Program Files\R\R-3.0.1\bin\x64\R.exe
```

- If the R library 'rJava' cannot install properly, and you have a message like this:

```
Loading required package: rJava
library(rJava)
Error : .onLoad failed in loadNamespace() for ‘rJava’, details:
call: fun(libname, pkgname)
error: JAVA_HOME cannot be determined from the Registry
```

then you have probably installed the wrong version of Java. See this webpage for more details.

- If you are using i) the macro 'Plot cell division', ii) the macro 'Plot intervals' and asking for a division variable or iii) the macro 'Analyze lineages' and asking for the CSV file, and that this error appears:

```
Error in Divisions$Lineage[i] <- Lineage : replacement has length zero
Calls: do.call -> <Anonymous> -> process.AllDiv
Execution halted
```

or even something like this one:

```
Error: Unable to parse: POLYGON((-18.8409884266078 48.1651655066508,-
16.2678255172883 39.0719279840682,-18.8409884266078 48.1651655066508))
GEOS reported: "IllegalArgumentException: Invalid number of points in
LinearRing found 3 - must be 0 or >= 4"
Execution halted
```

then it is most likely that you are working on a PointTracker file containing cells that have not been tracked from the first image of the dataset (ie cells have been added at later images), and i) you have forgotten to tick the checkbox 'Use the first selected image to process divisions' at the first time you called the PointTracker csv file that you are using or ii) you have asked to plot images on images anterior to the image in which all lineages are present. If i) delete the file named *Divisions__<name of PointTracker csv file>.csv* in the subfolder *\Divisions*, and do not forget to tick the checkbox next time you run the macro. If ii) you should use later images on which all cells are present.