Primed Track :: Documentation

Requirements

Primed Track requires several tools:

- Imaris:
  - ImarisXT (IPSS)
  - Imaris MeasurementPro (Reference Frame)
  - Imaris Track
  - Imaris Lineage
- MATLAB:
  - IceImarisConnector: https://github.com/aarpon/IceImarisConnector

For the (optional) tree validation, the following is required:

- Python 2.7:
  - pIceImarirConnector: https://github.com/aarpon/pIceImarisConnector/releases/latest
  - Zhang-Shasha Tree edit distance (https://zhang-shasha.readthedocs.io/en/latest):
    `python -m pip install zss`
  - treelib: `python -m pip install treelib`
  - numpy: `python -m pip install numpy`
  - scipy: `python -m pip install scipy`

Usage

Cell segmetation

- Detect green and red cells using the **Spot detector** in Imaris. Use a low threshold to segment all cells even at the cost of including spurious spots. Allow spot radius to be adapted to more accurately fit the volume of the segmented cell. Bright but very small spots can easily be filtered out during segmentation.

First validation

Using the "BSSE > Primed Track > Validate cells in embryo" ImarisXTension (MATLAB)

- Use the green spot positions to estimate the embryo diameter and discard green spots that are likely to be outside of the embryo. The radius of the embryo is roughly estimated as the median of all maximal inter-spot distances.
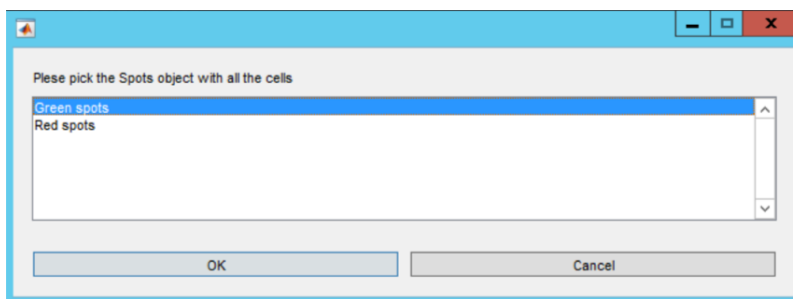
Figure 1: Select the green cells

- Search for spots that occur within a small defined distance from a spot in the same channel, discard all wrongly segmented double spots on one nucleus and replace them by one new spot.
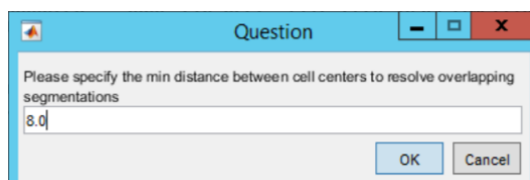

Figure 2: Resolve double segmentations

- A user-defined multiplicative factor can optionally compensate for estimation errors and prevent cells at the boundary of the embryo to be discarded should this constraint be too stringent.
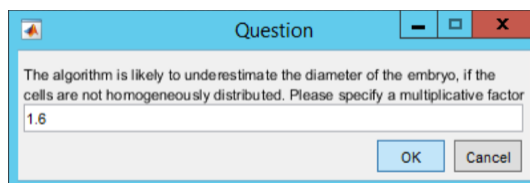

Figure 3: Correct for understimation of embryo diameter

- Discard all spurious red spots that do not colocalize with a green spot. Note that due to the equilibrium between protonated and de-protonated chromophore, green to red photoconversion of pcFPs is never exhaustive and will always retain a green population, rendering this quality control step possible.
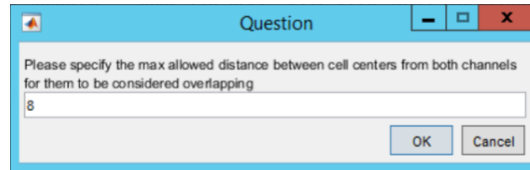
Figure 4: Remove red spots that do not overlap with green spots

- A red spot discarded during the first validation can optionally be recovered if there is a valid red spot in previous time point within a user-defined search radius. This adjustment compensates for remaining miss-segmentations in the green channel.
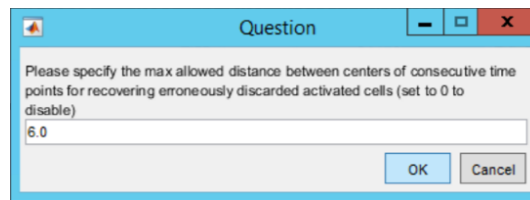

Figure 5: Recover discarded red spots

Embryo alignment (drift and rotational correction) / Cropping

Using the "BSSE > Primed Track > Set reference frames from cells" ImarisXTension (MATLAB)

- Imaris Reference Frame Objects are created in MATLAB for each time point: their origin is set at the position of the center of mass (COM) of the green spots and their orientation is given by the vector.
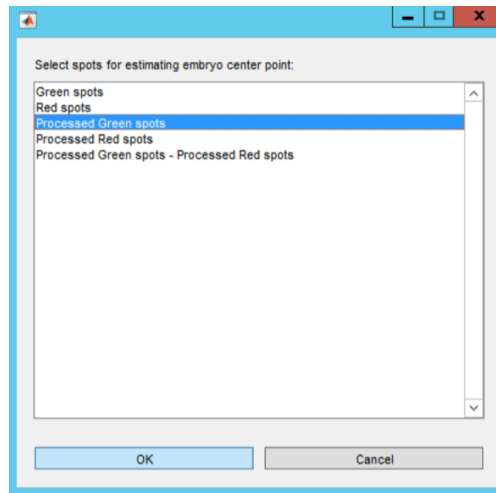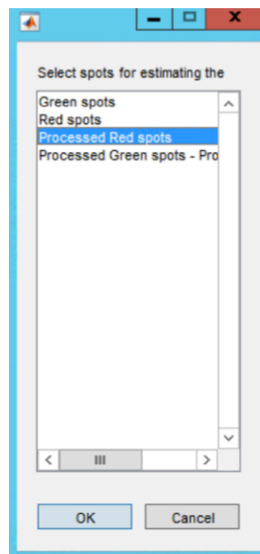
Figure 6: Select processed green spots

Figure 7: Select processed red spots

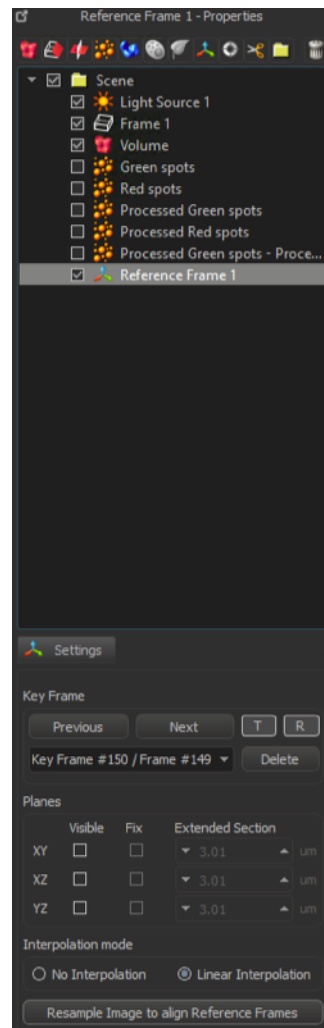- Perform resampling using the reference frame object in Imaris.

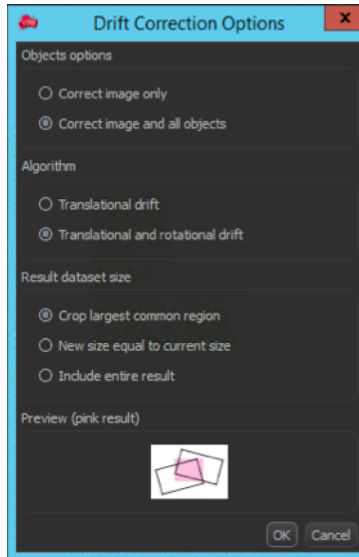Figure 8: Select created reference frame in Imaris

Figure 9: Perform registration

Using the "BSSE > Primed Track > Correct rotation around reference frame axis" ImarisXTension (MATLAB)

- The previous correction still has one degree of freedom. The rotation angle around the reference frame axis is obtained by comparing the positions of the green spots at timepoints t and t - 1 over 360 1-degree rotations and by choosing the angle that minimizes the cell drift between time points.
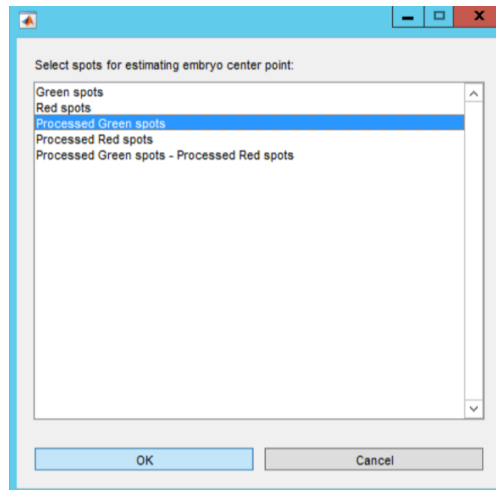

Figure 10: Select the processed green spot

- Perform resampling using the reference frame object in Imaris and crop the result to the smallest bounding volume.
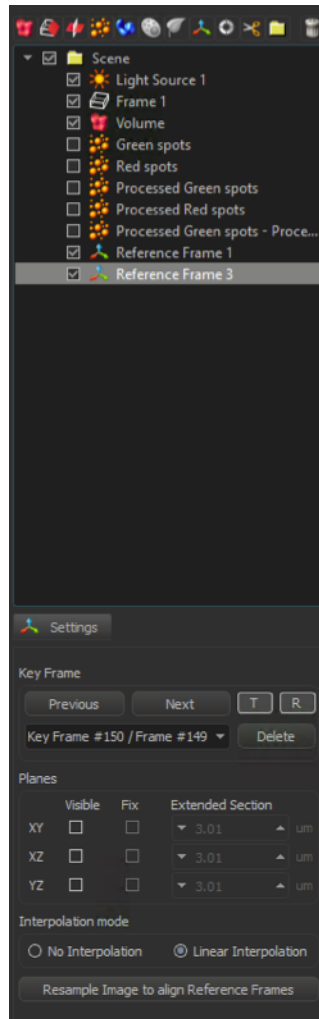

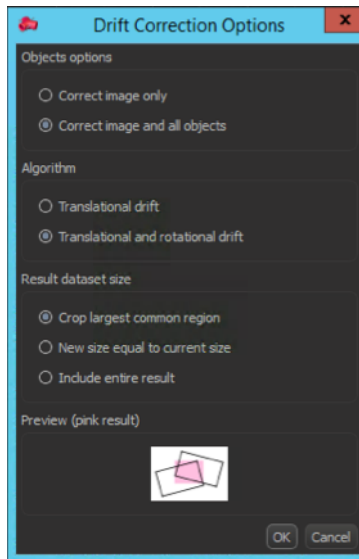Figure 11: Select created reference frame in Imaris

Figure 12: Perform registration

Second validation and subsetting

Using the "BSSE > Primed Track > Validate cells in embryo" ImarisXTension (MATLAB)

- To pick up red cells that were not recovered previously, re-run the first validation on the re-aligned embryo (follow the steps from above).
- Create a new spot object that contains the subset of green cells that do not colocalize with red cells.
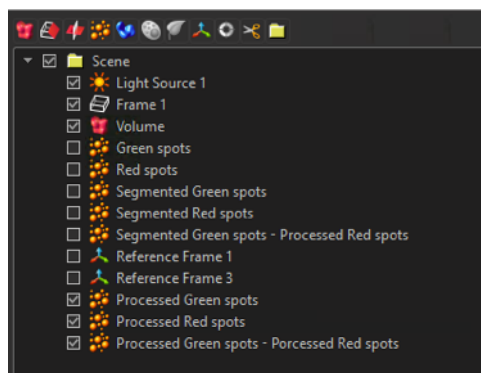

Figure 13: Subsetting

Lineage tree reconstruction

- Imaris' Lineage module is used to track the cells over time and reconstruct their lineage tree. The subsetting in the previous step allows us to reduce the complexity of the lineage tracing problem by breaking it down into two simpler, computationally less expansive, disjoint problems.
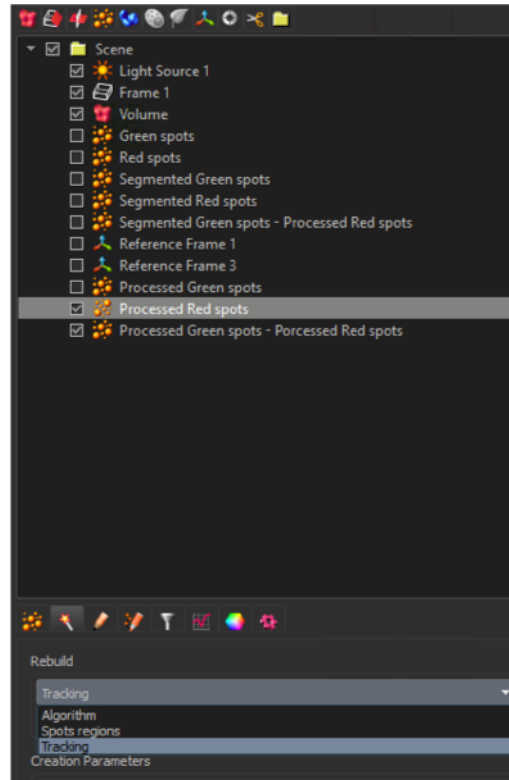

Figure 14: Lineage tree reconstruction

Comparative Analysis of lineage trees (optional)

Using the "BSSE > Primed Track > Calculate total distance between lineage trees" ImarisXTension (python)

- Run the python XTension to calculate the total distance between lineage trees before and after correction.
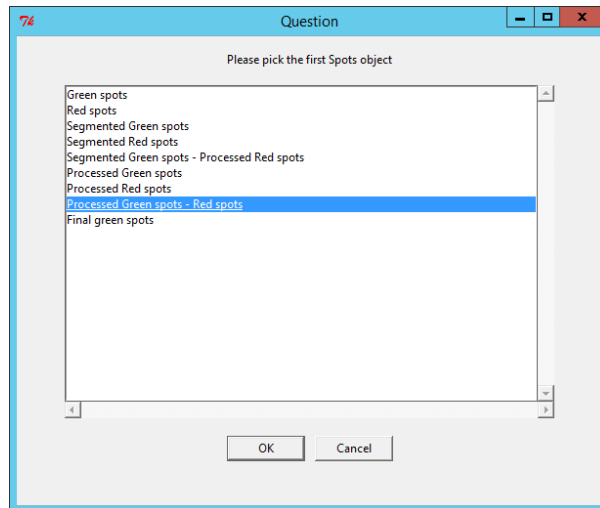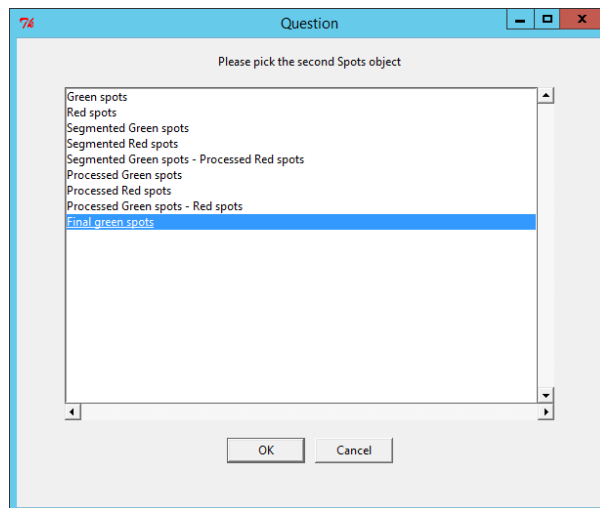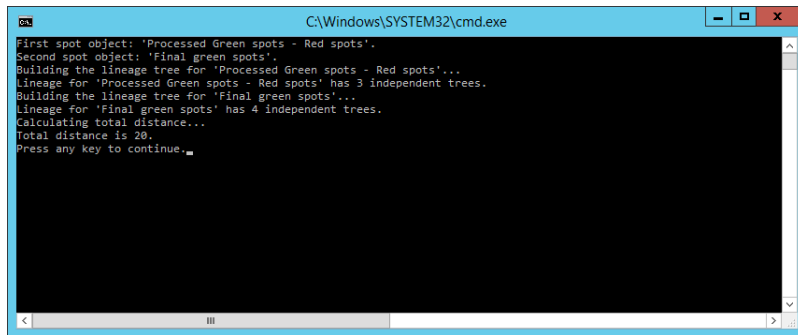
Figure 15: First lineage tree

Figure 16: Second lineage tree

Figure 17: Output of the lineage tree distance calculation