Preparation

You typically need to install matlab MCR to run starrynite in its compiled form
http://www.mathworks.com/products/compiler/mcr/

You want version 8.0 (32) for 2014 version (8.4 (64) for the current compiled version), if this isn't right the error message when trying to run it will tell you the version you need to download.

The runtime bit version should match the compiler version, not the architecture of your machine, if it still gives you an error after installing the requested number, try the other bit version.

Note that on a PC a reboot is usually necessary for MCR to be loaded.

To run acetree you will usually need the oracle Java rather than standard windows java

http://www.oracle.com/technetwork/java/javaseproducts/downloads/index.html

DEPRECATED: Current Acetree versions do not depend on Java3D, but this section has been left for reference in case older Acetree version is being used:

You'll also need to install Java3D,

For 64 bit pc java, you need the AMD64 version of Java3D. Download the zip binary version. Copy the bin and lib directories in the zip onto the matching directories in your java runtime and/or jdk.

e.g.

C:\Program Files\Java\jdk1.7.0_25

C:\Program Files\Java\jdk1.7.0_25\jre

C:\Program Files\Java\jre1.7.0_25

Java3d is currently unsupported on macs, Acetree should run but the 3d viewer will not

Running Starrynite

Double click Starrynite executable (currently StarryNite2019.exe)

The Order of steps doesn't matter bit if the split check box is going to be set it must be done before choosing the image.

Set start/end time (I would strongly recommend testing on only 3-5 frames the first time you try this to avoid waiting a long time to find something breaks or your parameters are wrong)

Image type selection:

Default image type is rootname[frame#].tif containing where # is the timepoint and the image contains all slices.

The splitimage indicates a format specific to metamorph running two cameras where red is on one side and green on the other, with all the slices in one file as above and _t hard coded as the separator. The color button effects which side of the image is analyzed for nuclei.

.mat files containing image data in a variable stack are also partially supported, and will be automatically detected if selected. Note the resulting acetree xml file will need to be edited before opening.

Look out for spaces in the file or directory names which it handles poorly.

If you want to output 8 bit 'traditional' acetree slices select the 'make 8 bit tiff slices' radio button, the resulting acetree xml file will point to the 8 bit images rather than the originals (default)

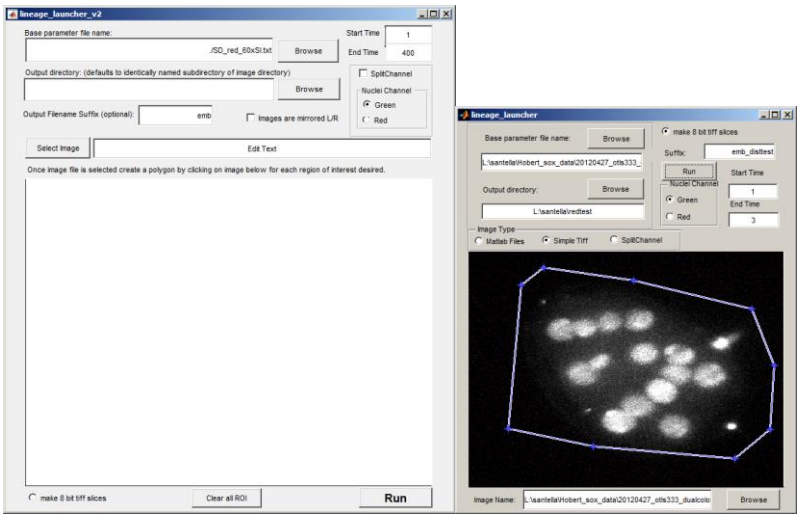
Click on browse by 'base parameter file name:' and select the appropriate param file (examples in parameter directory)

Click browse by 'output directory:' and select directory for acetree file output (if left blank will default to output in image directory)

Click browse near 'image name' select any image from the series to analyze

Draw a loose circle around the embryo, closing it by clicking on first point
Click run

Note the UI should look like that on left, not the deprecated version on right.



On completion all embryos completed will appear in the command window and acetree zip and xml files will be present.

These will include two copies of identical content one suffixed –edited the _edited one will have an auxinfo file accompanying. See Lineage_Editing_Notes.doc for instructions on how to edit this file and the lineage to ensure accurate Sulston naming.

Parameters:

To get a reasonable result its usually just necessary to set the initial diameter, the voxel aspect ratio, slices and multiply the intensity threshold ramp by some constant to match the overall brightness of the imaging. Parameter names are below, they may have different values and be located in different places in different parameter files.

```
xyres=.254;%x,y voxel size
zres=1;  %z voxel size
slices=60;%slices in image (if smaller than real number will be cropped, if
larger than the real number the min of this number and real number will be
used)
firsttimestepdiam=40; %diameter in pixels in first frame

%threshold for noise maxima in images

%in practice this is easiest to tune by running just the first 4-5 frames of
a series making a binary search of multiplier. IE if .75 leaves noise try 1.5
if it removes real nuclei try .375 once the ramp works on the first few
frames it should work for the whole series. (if analyzing c elegans)


parameters.intensitythreshold=[18,20,23,26,36,46,50].*.75;
```

All the parameters are described in:

http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-217

http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-580

Output file format: each time point is a separate text file zipped into a single file, each line in each file represents a cell with the fields below:

```
1(cell ID specific to this file, not always the same as row number), 1(validity
flag: 1 valid, 0 invalid (a cell deleted in acetree)), 24 (predecessor ID in
previous file), 45(successor ID in next file), -1(successor 2 id in next file),
178(x), 57(y), 5.2(7), 11(size), ABalpaaaaa(name), 6547(lineage marker 'weight'),
37259(reporter intensity - Boyle Units), 6036(summed reporter intensity),
162(voxels), (placeholder for forced Acetree name), 37259(SRI), 0(global
correction), 0(local correction), 0(blot correction), 0("crosstalk" correction),
```

```
If auxinfo file is not created its creation can be re-run manually:
java -Xmx500m -jar acebatch2.jar Measure [xmlfilename]
```

Running AceTree

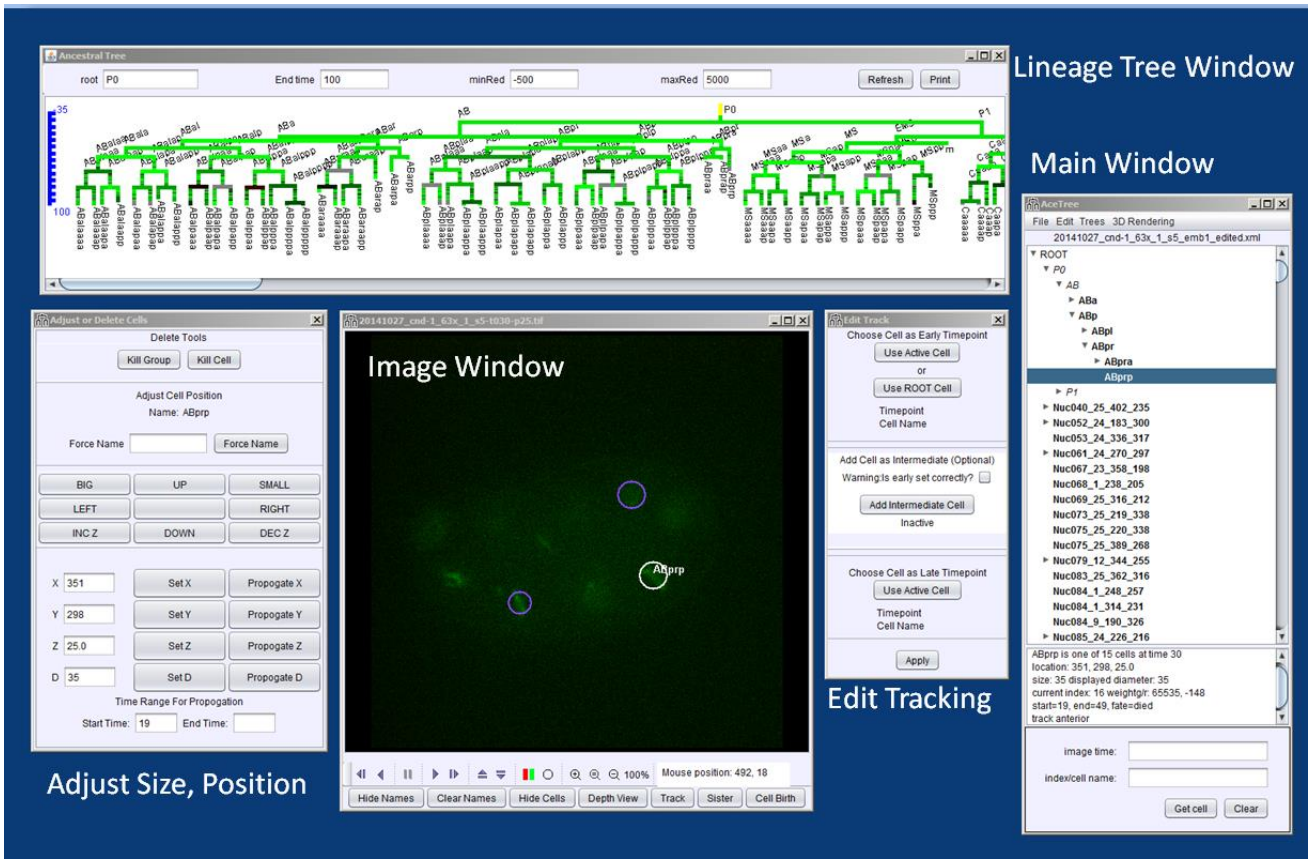
The AceTree.jar can be clicked on to run it, but to test a new installation or debug problems run it from a terminal window. On Mac, linux, cygwin type: 'cd [full path to directory where acetree.jar is]' e.g. 'cd c:/acetree/. On windows terminal type '[driveletter]:' e.g. 'C:' then type cd [full path minus drive letter to acetree.jar] e.g. 'cd acetree\'

On all platforms type 'java –jar AceTree.jar'

DEPRECATED: Launch the 3D window to test java3D, even if you don't need the 3D window you need java3D as computation of cannonical *C. elegans* names depends on Java3D vector math libraries.

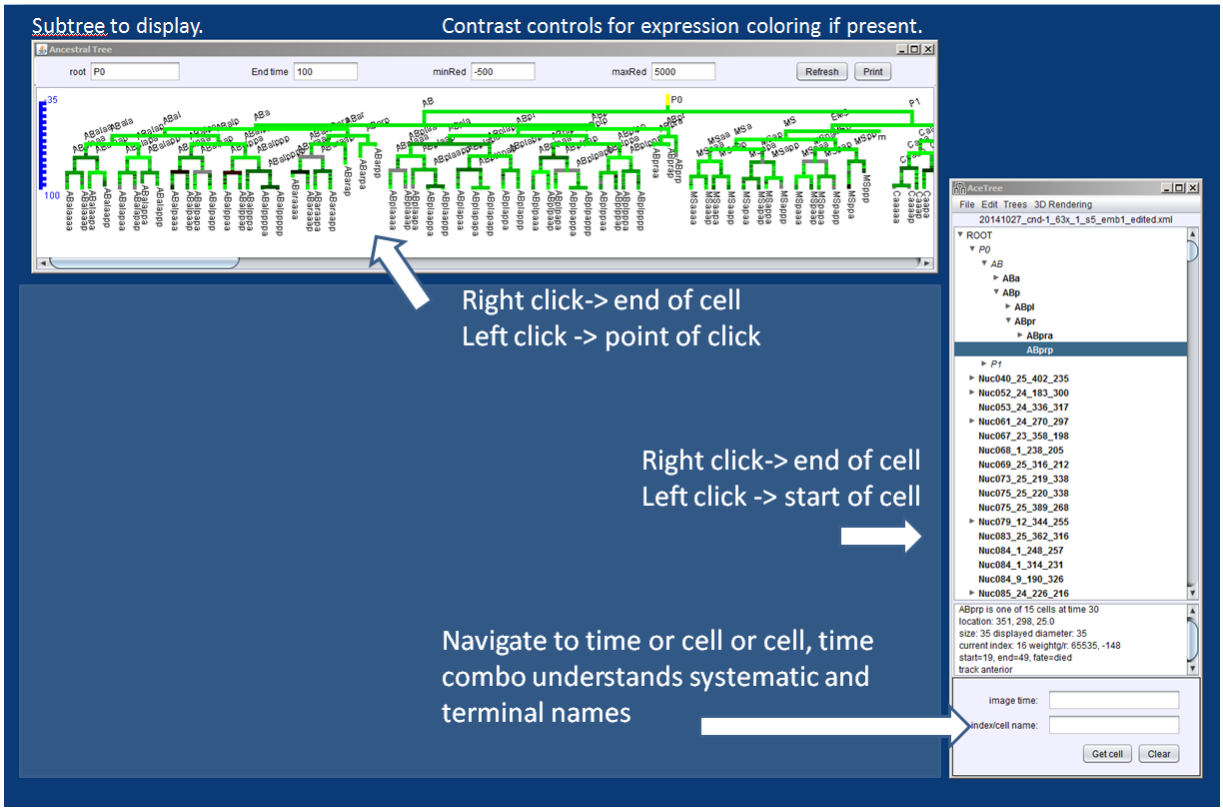
Editing Lineages with Acetree

*note this is a brief general overview of editing tools see Lineage_Editing_Notes.doc for details of establishing correct systematic cell names.



Overview of Acetree UI

Navigation


Subtree to display.
Contrast controls for expression coloring if present.
Right click-> end of cell
Left click -> point of click
Right click-> end of cell
Left click -> start of cell
Navigate to time or cell or cell, time combo understands systematic and terminal names

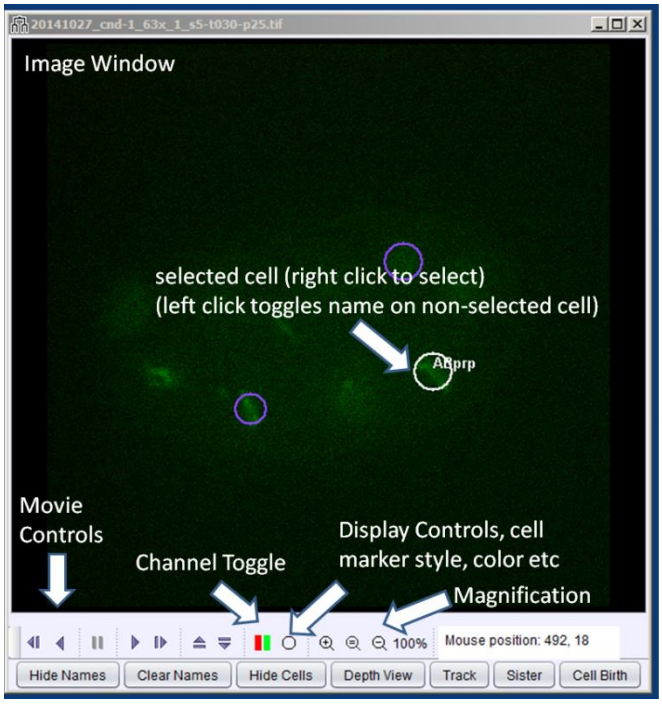
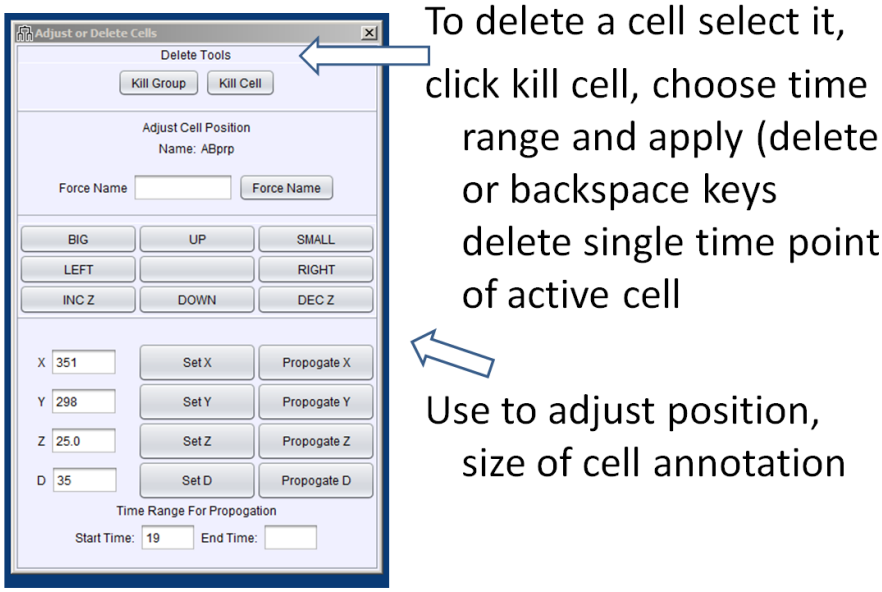


Image Window
selected cell (right click to select)
(left click toggles name on non-selected cell)
ABprp
Movie Controls
Channel Toggle
Display Controls, cell marker style, color etc
Magnification
Hide Names
Clear Names
Hide Cells
Depth View
Track
Sister
Cell Birth

Navigation is done either via the lineage views or by directly moving in space and time in the image window. Note: MIP button not pictured switches image window into maximum projection view (axis aligned).

Editing. Edit tools are found under the file menu Edit->edit tools



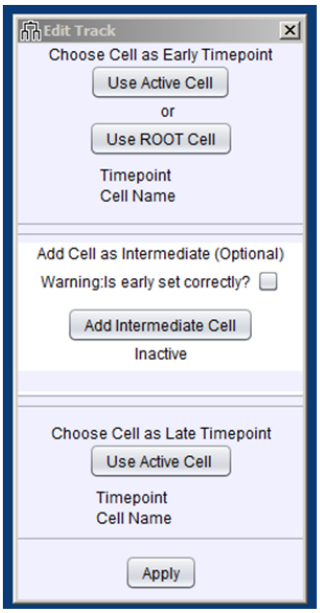
The active cell can be deleted, resize and repositioned using this dialogue

Relinking

1. Select cell at time which is where you want to attach a mis-tracked cell
Click use Active Cell

2. Select cell at time which you want to attach to the early timepoint cell above

Click Apply.



Manual Tracking

1. Select a cell at time which you want to track forward manually
Click Use Active Cell
OR
Click Use Root Cell to start a new cell

2. Tick safety box
click Add Intermediate Cell
click on cell location in image, advance 1 or more frames, click again to continue manual track

Remove a Branch

1. click Use RootCell
2. Select cell at time to detach.
Click apply.

Editing involves searching for errors either via the list of Nuc entries in the main window (each Nuc after the first frame is a cell Starrynite cannot place in the lineage if doing in toto imaging), or looking for topological errors in the Lineage Tree window. Once found these are relinked with the lineage dialogue. Each cell detection (a circle in the image window) can be thought of as a bead that Starrynite has tried to string together with all the others into the complete lineage.

Relinking involves only the top and bottom panels in the Edit Track dialogue, setting the early cell as the last correctly tracked cell, and the late cell as the cell which should be attached to it. If an incorrect cell is attached to the point as well this has to be removed following (before or after) the steps for removing a branch above. If the correct link is made first the result will be a false division in the lineage and the incorrect daughter needs to be removed.

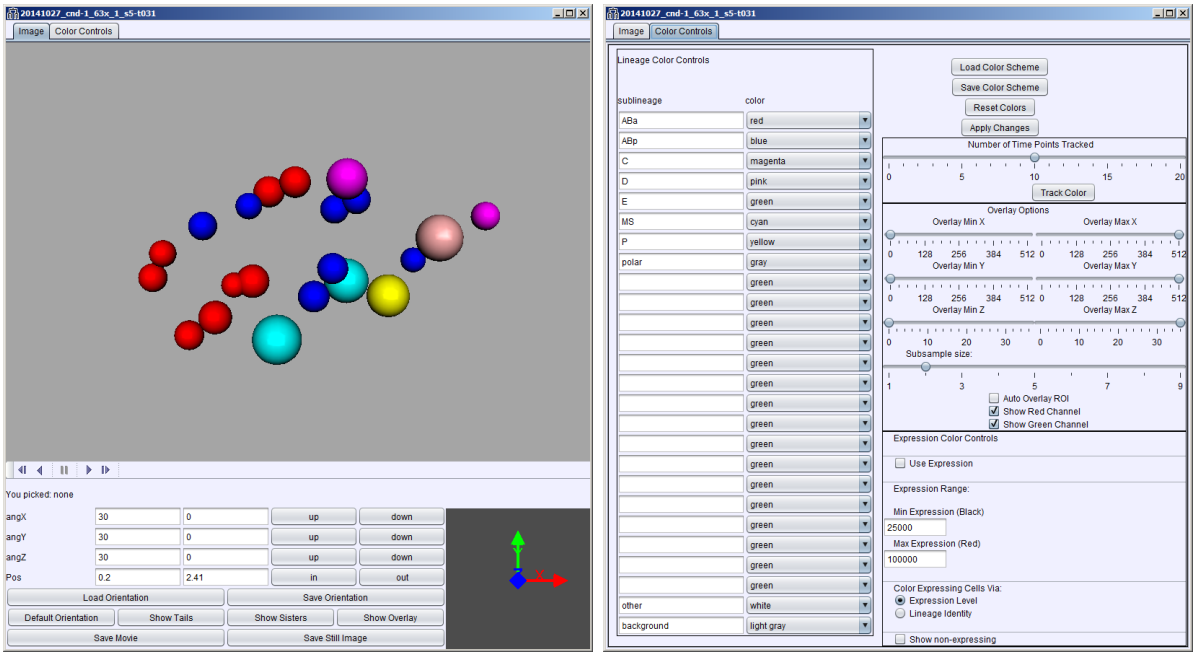
Typically it is simplest to begin in the first frame edit cells found in the Nuc list to a given time point, figuring out where each should go. Then the lineage tree can be checked for implausible divisions or additional lost tracks. This process is then iterated for a later window of time.

Adding a cell manually. Select and choose use active cell in early time box if you want to extend an existing cell manually. Select use root cell in early to start an entirely new cell. Click the safety tick box in the add cell as intermediate (optional) panel, click the add cell button in the same panel. (the panel

will read active in red) Move to the location/time you want to add the cell and click in the image window. Move forward in time and repeat as desired.

Visualization

DEPRECATED: Note this window has been replaced by the WormGUIDES based 3D window which does not use Java 3D



3D window and Color Control tab for 3D window

Controls can load and save color schemes, superimpose image data on the 3D model.

Color control window allows colors to be specified for a sublineage, and to change between lineage and expression data based coloring (when expression information is present in lineage zip file).

Auxinfo creation:

Expression Quantification

Expression quantification is run when starrynite is run to re-run expression analysis on split 16 bit images from the command line type:

cd l:/bin/starryniteII/

java -jar acebatch2.jar Extractor [xmlfilename] [G|R] [end_time]

whether the analysis is run on the 8 bit split images or 16 bit original images depends on the xml file, if only one channel is specified in the xml file (typical for non split images) it will attempt to auto detect the matching location of the requested channel (possible if these are dispim/isim images saved with the naming convention typical in the bao lab)

DEPRECATED syntax for expression quantitation has been revised:

java -cp acebatch2.jar SixteenBitGreenExtractor1 [xml file name] [endtime (e.g. 400)]

to run on 8 bit slice images in the acetree image directory drop the SixteenBit to run on the red channel change Green to Red.

If you run on 8 bit slice images then instead

java -cp acebatch2.jar GreenExtractor1 [xml file name] [endtime (e.g. 400)]