BMC Bioinformatics

AceTree: a major update and case study in the long term maintenance of open-source scientific software

--Manuscript Draft--

Manuscript Number:		
Full Title:	AceTree: a major update and case study in the long term maintenance of open-source scientific software	
Article Type:	Software	
Section/Category:	Imaging, image analysis and data visualization	
Funding Information:	National Institutes of Health (U01 HD075602)	Not applicable
	National Institutes of Health (R24OD016474)	Not applicable
	National Institutes of Health (GM097576)	Not applicable
Abstract:	Background AceTree, a software application first released in 2006, facilitates exploration, curation and editing of tracked C. elegans nuclei in 4D fluorescence microscopy datasets. Since its initial release, AceTree has been continuously used to interact with, edit and interpret C. elegans lineage data. In its eleven year lifetime, AceTree has been periodically updated to meet the technical and research demands of its community of users. This paper presents the newest iteration of AceTree which contains extensive updates, demonstrates the new applicability of AceTree in other developmental contexts, and presents its evolutionary software development paradigm as a viable model for maintaining scientific software. Results Large scale updates have been made to the user interface for an improved user experience. Tools have been grouped according to functionality and obsolete methods have been removed. Internal requirements have been changed that enable greater flexibility of use both in C. elegans contexts and in other model organisms. Additionally, the original 3-dimensional viewing window has been completely reimplemented. The new window provides a new suite of tools for data exploration. Conclusion By responding to technical advancements and research demands, AceTree has remained a useful tool for scientific research for over a decade. The updates made to the codebase have extended AceTree's applicability beyond its initial use in C. elegans and enabled its usage with other model organisms. The evolution of AceTree demonstrates a viable model for maintaining scientific software over long periods of time.	
Corresponding Author:	Zhirong Bao UNITED STATES	
Corresponding Author Secondary Information:		
Corresponding Author's Institution:		
Corresponding Author's Secondary Institution:		
First Author:	Braden Katzman	
First Author Secondary Information:		
Order of Authors:	Braden Katzman	
	Doris Tang	

	Anthony Santella
	Zhirong Bao
Order of Authors Secondary Information:	
Author Comments:	

Title

AceTree: a Major Update and Case Study in the Long Term Maintenance of Open-Source Scientific Software

Name, institutional addresses, and email addresses of all authors

Braden Katzman, Doris Tang, Anthony Santella and Zhirong Bao

Developmental Biology Program, Sloan-Kettering Institute, New York, New York

bradenkatzman@gmail.com, huiyao.tang@gmail.com, santella@mskcc.org, baoz@mskcc.org

Corresponding author

Zhirong Bao

Abstract

Background

AceTree, a software application first released in 2006, facilitates exploration, curation and editing of tracked *C. elegans* nuclei in 4D fluorescence microscopy datasets. Since its initial release, AceTree has been continuously used to interact with, edit and interpret *C. elegans* lineage data. In its eleven year lifetime, AceTree has been periodically updated to meet the technical and research demands of its community of users. This paper presents the newest iteration of AceTree which contains extensive updates, demonstrates the new applicability of AceTree in other developmental contexts, and presents its evolutionary software development paradigm as a viable model for maintaining scientific software.

Results

Large scale updates have been made to the user interface for an improved user experience. Tools have been grouped according to functionality and obsolete methods have been removed. Internal requirements have been changed that enable greater flexibility of use both in *C. elegans* contexts and in other model organisms. Additionally, the original 3-dimensional viewing window has been completely reimplemented. The new window provides a new suite of tools for data exploration.

Conclusion

By responding to technical advancements and research demands, AceTree has remained a useful tool for scientific research for over a decade. The updates made to the codebase have extended AceTree’s applicability beyond its initial use in *C. elegans* and enabled its usage with other model organisms. The evolution of AceTree demonstrates a viable model for maintaining scientific software over long periods of time.

Keywords

C. elegans, 4D, 3D, fluorescence microscopy, automated lineaging, embryogenesis, affine transformation, interface

Background

The invariant lineage of the nematode *C. elegans*[1] makes the organism a powerful model for studying developmental processes. StarryNite, a software package released in 2006, performs automated lineage extraction by segmenting and tracking fluorescently labeled nuclei in 4D microscopy datasets[2]. AceTree, a companion program built to view and edit the nuclear tracking data generated by StarryNite, facilitates interpretation validation and quality control of StarryNite results[3].

AceTree, developed beginning in 2005, has since its initial release provided a comprehensive set of tools for interacting with lineage data, both in a 2-dimensional viewing window where tracks are superimposed on nuclear images and as an abstracted lineage tree[4]. Users can explore their data both in time and space, by moving up and down within and between annotated image stacks. Additionally, a 3-dimensional viewing window provides an abstract view of nuclear positions as a cloud of 3D spheres.

This representation of the data provides a more intuitive sense of the positions of cell bodies in space than can easily be achieved by moving between 2-dimensional image slices.

Continuously in use for the eleven years since its initial release, AceTree has been periodically updated to meet the technical and research demands of its community of users. The software has proved to be a useful tool in research, necessitating evolutionary changes as software libraries and microscopy technology have evolved.

AceTree's latest release provides a multitude of changes aimed at meeting the demands of its community and incorporates new features for visualization and analysis. A large-scale user-interface update adds new tools, removes obsolete ones and facilitates improved accessibility of key functionality. A revised image loading pipeline supports greater flexibility in input images. Revisions to canonical name assignment allow for the free orientation of embryos in 3-dimensional space and an entirely new 3-dimensional viewing window provides a new suite of methods for exploring cell positions.

Implementation

AceTree is written in Java, and has been updated to Java 1.8 to allow the use of new language and library features and remove dependencies on deprecated libraries. AceTree's new 3-dimensional visualization window, derived from the WormGUIDES atlas[5], is written in Java using the JavaFX 8 platform. Development of the software is carried out in the open-source IntelliJ integrated development environment (IDE). The program is packaged as a cross platform JAR (Java Archive) file and has been tested on Linux (Ubuntu 14.04, 16.04), Windows (7 Professional, 10) and macOS (10.13 High Sierra).

Github provides source code and instructions for development setup: https://github.com/zhirongbaolab/AceTree

Results

Interface

The user-interface has been rearranged to better organize tools, grouping features with shared purposes together when possible, see Figure 1. Viewing controls such as time and plane, color channel selection and controls for cell selection and labeling have been moved to the image window in order to concentrate display controls in a toolbar within the main 2D image window. Editing tools have been reorganized, placing manual tracking and track editing tools together. The file menu has been updated by grouping functionality more systematically and removing obsolete tools.

The rearranged user-interface also integrates new image controls. The image window now includes zoom and brightness levels controls.

Flexibility

A collection of changes have been made to increase the flexibility and usefulness of AceTree in a variety of developmental contexts. Later stages of *C. elegans* embryonic development are increasingly accessible due to advances in imaging and techniques for computationally untwisting embryos after muscular twitching begins[6]. In toto imaging of other organisms is also increasingly possible[8, 9] while navigating and interpreting large datasets remains challenging. New Acetree features address previous limitations and benefit the *C. elegans* research community while in many cases also increasing AceTree's usability with other model organisms.

Functional name data from the *C. elegans* Parts List[1] has been fully integrated into AceTree. Search functionality throughout uses functional and systematic names interchangeably. This extension is useful later in embryonic development as terminal cells can be more easily recognized by their functional names.

Systematic name assignment code has always been built into AceTree. Originally, name assignment was manually rerun when users needed to update naming during tree edits. Now, name assignment is automatically updated with every user edit to the lineage.

AceTree first supported naming only on canonically oriented embryos. Later functionality was added to allow the naming of randomly positioned embryos, removing the need to orient embryos canonically on the slide or in post-processing. However, the assumption remained that embryos were mounted compressed[3]. With this mounting method the Left-Right (LR) axis of the 4-cell stage embryo aligns with the axial direction. Though this mounting is convenient in many circumstances it is often desirable to image the embryo from different orientations in order to better observe specific structures. Additionally, new imaging approaches, such as the dual-view inverted selective plane illumination microscope (diSPIM)[7], require an uncompressed mount, meaning embryos can be rotated randomly around their Anterior-Posterior (AP) axis. To support naming in these contexts, a new, optional naming mode has been introduced in which the AP and LR vectors of the 4-cell embryo are directly specified. These values are used to translate between image and canonical embryo space, allowing embryos to be named even when arbitrarily oriented in 3D. Two caveats remain, expected division orientation vectors are still based on data from compressed embryos, and in some cases division axes can be significantly different relative to the body axes under the two mounting conditions, resulting in an increased rate of naming errors. In addition, expected division axes are missing for many tenth round divisions. Naming in these cases continues to revert to default body axis based naming. Collecting empirical division axis expectations for the tenth round and in uncompressed embryos remains future work.

Fluorescence microscopy has evolved enormously in the past decade. New techniques have enabled complete imaging in larger organisms like drosophila and zebrafish[8, 9] with larger image volumes, longer developmental times and tens of thousands, instead of hundreds, of cells. In light of these advancements, AceTree has been extended to support longer movies and higher cell counts. Restrictions on maximum slices and frames have been removed and loading and updating internal data structures has been optimized to allow much larger files to be effectively loaded and edited. Names can now be manually assigned to any cell, even when no *C. elegans* embryo is detected, allowing completely manual naming to be used when desired. This collection of functionality simplifies the use of AceTree for other model organisms, see Figure 3.

AceTree was originally developed to work with 8bit images, but greater bit depth is currently available from most sensors. Acetree has been extended to read 16bit images and dynamically map them to display depth using interactive black and white point controls for each channel.

3D Window

Many users find it challenging to build up a mental image of the 3D relative positions of objects by moving through an image stack. Often, it is easier to understand the relative position of nuclei in an abstract 3D model. This has made the 3D window an important AceTree feature from its first release. Initially, this window was implemented in Java3D, a high-level scene graph API (Application Programming Interface) for JAVA. Since then, Java3D has become a community source project, no longer directly supported by Oracle[10]. JavaFX is now the regularly maintained, integrated high level 3D graphics library of the Java Runtime Environment and Java Development Kit (JRE, JDK)[11].

Lack of support means that Java3D is difficult to install, and has not functioned on macOS platforms for some time. To address these deprecations, an entirely new 3D window for browsing the embryo was built in the context of the WormGUIDES neurodevelopmental atlas[5]. Built in JavaFX, this 3D window has been integrated into Acetree to serve as a replacement for the original 3D window, see Figure 2.

In addition to a 3D display with controls, this viewer provides a new search interface for data exploration. Users can search for cells and color the nuclear position model by lineage name, functional name, parts list description, connectome, gene expression and ancestry.

Discussion/Conclusion

AceTree has undergone serious revisions in its eleven year lifetime. Its main windows have been largely reorganized and its internal representations extended and generalized. At this point, much of its core functionality has been either greatly extended or entirely rewritten from its initial state.

The continuous evolution of the AceTree software package is an intriguing case study in maintaining actively used scientific software. For over a decade, AceTree has been an important tool for scientific research in developmental biology labs, and has continually evolved to meet technology and research demands.

Typically, software is maintained in two ways, either by a team of dedicated developers in a commercial or infrastructure grant context, or by large scale open-source community efforts. Given its relatively modest but dedicated user base, AceTree has been maintained differently, with a small group of primary developers intermittently working on AceTree at different times during its lifetime. The changes that AceTree has undergone are a product of feedback from its community of users and changes in the software packages that it utilizes.

AceTree is not a heavily funded effort with full time staffers. Rather, AceTree has been maintained over a long period of time by a small circle of core labs that it serves. Maintenance is fueled by researchers who use it, incentivizing its continued availability and application in the community.

Often, scientific software is released with the intention of ongoing use and adaption by the open-source community. In reality, many of these projects are released and never used. AceTree's continued usage and its responsiveness to the community demonstrate a model for how scientific software can work in the ever changing dynamics of the open-source user community. It is a niche tool driven by low level, ongoing development over a relatively long time frame.

We believe that the success and continued utility of AceTree establishes its evolutionary software development paradigm as a viable path for niche open-source scientific software. We felt it compelling to present this release of AceTree both because the updates significantly widen the possible community of users, and as an example of the practical concerns encountered when maintaining a fairly complicated code base over a decade timescale with limited developer resources.

Methods

Some of the new features available in the software required building interfaces between old and new code. Two main interfaces are worthy of detailed description. First, in order to maintain the original lineage naming paradigm yet allow users to lineage uncompressed embryos, we created a new method for transforming an uncompressed embryo's orientation to the expected canonical orientation. Second, to utilize AceTree's internal data representation in the context of the 3D window built for WormGUIDES, we created an abstract interface for representing the underlying lineage data that adheres to the StarryNite model specification.

To support uncompressed reorientation, we created the CanonicalTransform class to transform any orientation supplied by the user to the canonical orientation of *C. elegans* (anterior to the left and dorsal up)[1], an internal requirement of AceTree for lineage naming as division expectations are stored in a canonical coordinate system. The user defines the 3-dimensional orientation of the embryo by supplying two vectors, AP and LR, in the metadata AuxInfo_v2.xml file. The CanonicalTransform class finds the transform from these vectors to their canonical orientations by computing the axis-angle representation of the transform[12]. The transform calculation includes the special degenerate cases of the axis-angle representation when the supplied axis is already canonical or flipped-canonical i.e. collinear. The two resulting transformation matrices, AP and LR, are then concatenated to create a single, affine transformation. This transform is then applied to all division axes before they are propagated to existing naming code which assigns lineage names based on the direction of these divisions in a canonical orientation.

To interact with the AceTree data representation in a WormGUIDES context, we created the NucleiMgrAdapter class to package AceTree's data orderly and efficiently. The NucleiMgrAdapter class in AceTree's source code implements the LineageData interface defined in the WormGUIDES package. This adapter bundles AceTree's internal representation of the nuclei files, defined in the NucleiMgr class, into a form interpretable by WormGUIDES via the LineageData interface. This adapter is used to instantiate a WormGUIDES application instance in the WormGUIDESWindow class on a dedicated thread.

Availability and Requirements

Project Name: AceTree

Project Home Page: https://github.com/zhirongbaolab/AceTree

Operating Systems: Linux, Windows, macOS

Programming Language: Java

Other requirements: JRE 1.8 or higher

License: GNU GPL

Any restriction to use by non-academics: None

List of Abbreviations

4D, 3D, 2D, LR, AP, IDE, API, JAR, JRE, JDK

Declarations

Ethics approval and consent to participate: Not applicable

Consent for publication: Not applicable

Availability of data and material: Not applicable

Competing interests: The authors declare that they have no competing interests.

Funding: This work was supported by NIH grants U01 HD075602 and R24OD016474 and GM097576.

Authors' contributions

ZB and AS designed the software features. BK, DT, and AS engineered and programmed the software. BK wrote the manuscript with significant input from AS and ZB.

Acknowledgements

Thanks to current and former Bao Lab members Pavak Shah, Li Fan and Zhuo Du for their advice and feedback on AceTree features, also to all the WormGUIDES team, especially PIs Daniel Colon-Ramos, Hari Shroff, and Bill Mohler. Special thanks to all those in the Waterston Lab at University of Washington who originated the project, particularly to the late Thomas Boyle, AceTree's original developer, for his initial assistance in navigating the source code.

References

[1] Sulston, J. E., E. Schierenberg, et al. The Embryonic-Cell Lineage of the Nematode Caenorhabditis-Elegans. Developmental Biology. 1983;100(1):64-119.

[2] Murray, J. I., Z. Bao, et al. The lineaging of fluorescently-labeled Caenorhabditis elegans embryos with StarryNite and AceTree. Nat Protoc. 1983;1(3):1468-1476.

[3] Bao, Z., J. I. Murray, et al. Automated cell lineage tracing in Caenorhabditis elegans. Proc Natl Acad Sci USA. 2006;103(8):2707-2712.

[4] Boyle, T. J., Z. Bao, et al.. AceTree: a tool for visual analysis of Caenorhabditis elegans embryogenesis. BMC Bioinformatics 2006;7:275.

[5] Santella A, Catena R, Kovacevic I, et al. WormGUIDES: an interactive single cell developmental atlas and tool for collaborative multidimensional data exploration. BMC Bioinformatics. 2015;16(1):189. doi:10.1186/s12859-015-0627-8.

[6] Christensen RP, Bokinsky A, Santella A, Wu Y, Marquina-Solis J, Guo M, Kovacevic I, Kumar A, Winter PW, Tashakkori N, et al. Untwisting the *Caenorhabditis elegans* embryo. Elife. 2015;4:e10070.

[7] Kumar, A., Y. Wu, et al. Dual-view plane illumination microscopy for rapid and spatially isotropic imaging. Nat. Protocols. 2014;9(11):2555-2573.

[8] Lemon, W. C., S. R. Pulver, et al. Whole-central nervous system functional imaging in larval Drosophila. 2015;6:7924.

[9] Ahrens, M. B., M. B. Orger, et al. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. Nat Meth. 2013;10(5):413-420.

[10] Java 3D API. In: Java SE Desktop Technologies. Oracle. http://www.oracle.com/technetwork/articles/javase/index-jsp-138252.html. Accessed 10 Nov 2017.

[11] Getting Started with JavaFX Release 8. In: JavaFX. Oracle. 2014. https://docs.oracle.com/javase/8/javafx/JFXST.pdf. Accessed 10 Nov 2017.

[12] Jacobson N. Basic Algebra I: Second Edition. 2nd ed. Dover Publications; 1985

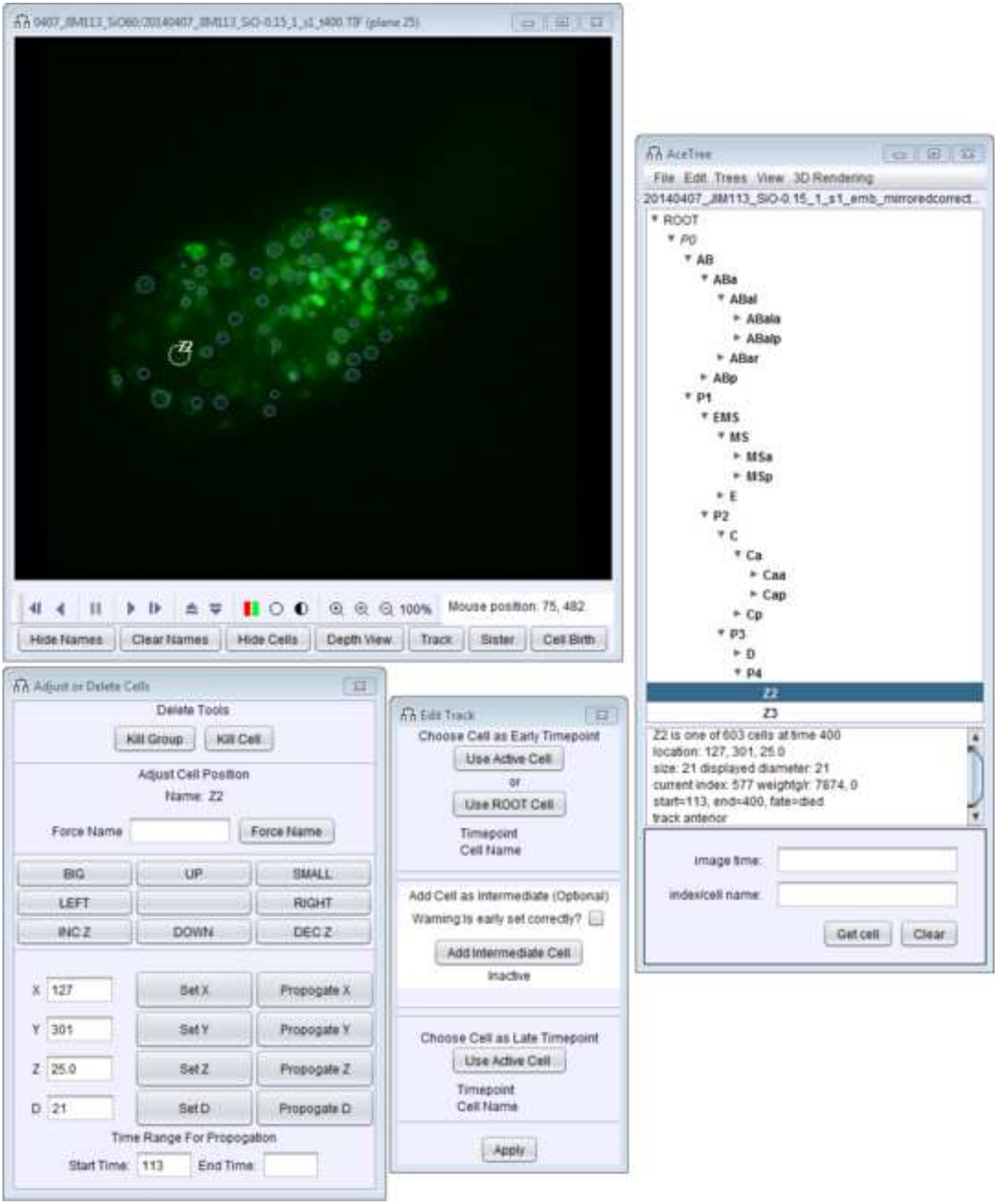
Figures, tables additional files

Figure 1 - An overview of the revised AceTree user interface. All display related functionality and commonly used toggles that control the appearance of the image window (time, plane, color channels, zoom, labels, tracking) are now in the image window (top left). Cell editing tools (bottom left) and track editing tools (bottom right) have been grouped in separate windows to better organize tools while enabling individual users to create their own preferred layout.

Figure 2 - An overview of the new 3-dimensional viewing window. Rules can color cells based on a broad array of search criteria including adult neuronal connectivity. The ‘Coloring Layers’ show the presynaptic and electrical connections of the amphid neuron ASGL and the head neuron URYVL. Color striping indicates that multiple rules apply to the striped entity. Here, the stripes on ASGL and URYVL indicate the wiring relationships between them in the adult. The ‘Display Options’ tab provides a key for the model annotations (right). Other searched criteria that can be used include lineage name, functional name, ancestry, and gene expression.

Figure 3 - A drosophila embryo in AceTree[8]. AceTree can support interpreting and lineaging for large datasets using optimized loading and editing methods and a generalization of the force naming tool.

Figure 1



0407_JIM113_SiO60/20140407_JIM113_SiO-0.15_1_s1_t400.TIF (plane 25)

Mouse position: 75, 482

AceTree

20140407_JIM113_SiO-0.15_1_s1_emb_mirroredcorrect

```
▾ ROOT
  ▾ P0
    ▾ AB
      ▾ ABa
        ▾ ABal
          ▸ ABala
          ▸ ABalp
        ▸ ABar
      ▸ ABp
    ▾ P1
      ▾ EMS
        ▾ MS
          ▸ MSa
          ▸ MSp
        ▸ E
      ▾ P2
        ▾ C
          ▾ Ca
            ▸ Caa
            ▸ Cap
          ▸ Cp
        ▾ P3
          ▸ D
          ▾ P4
              Z2
              Z3
```

```
Z2 is one of 603 cells at time 400
location: 127, 301, 25.0
size: 21 displayed diameter: 21
current index: 577 weightgr: 7874, 0
start=113, end=400, fate=died
track anterior
```

Image time:

index/cell name:

Adjust or Delete Cells

Delete Tools

Adjust Cell Position

Name: Z2

Force Name

X 127

Y 301

Z 25.0

D 21

Time Range For Propogation

Start Time: 113 End Time:

Edit Track

Choose Cell as Early Timepoint

or

Timepoint
Cell Name

Add Cell as Intermediate (Optional)

Warning:Is early set correctly? □

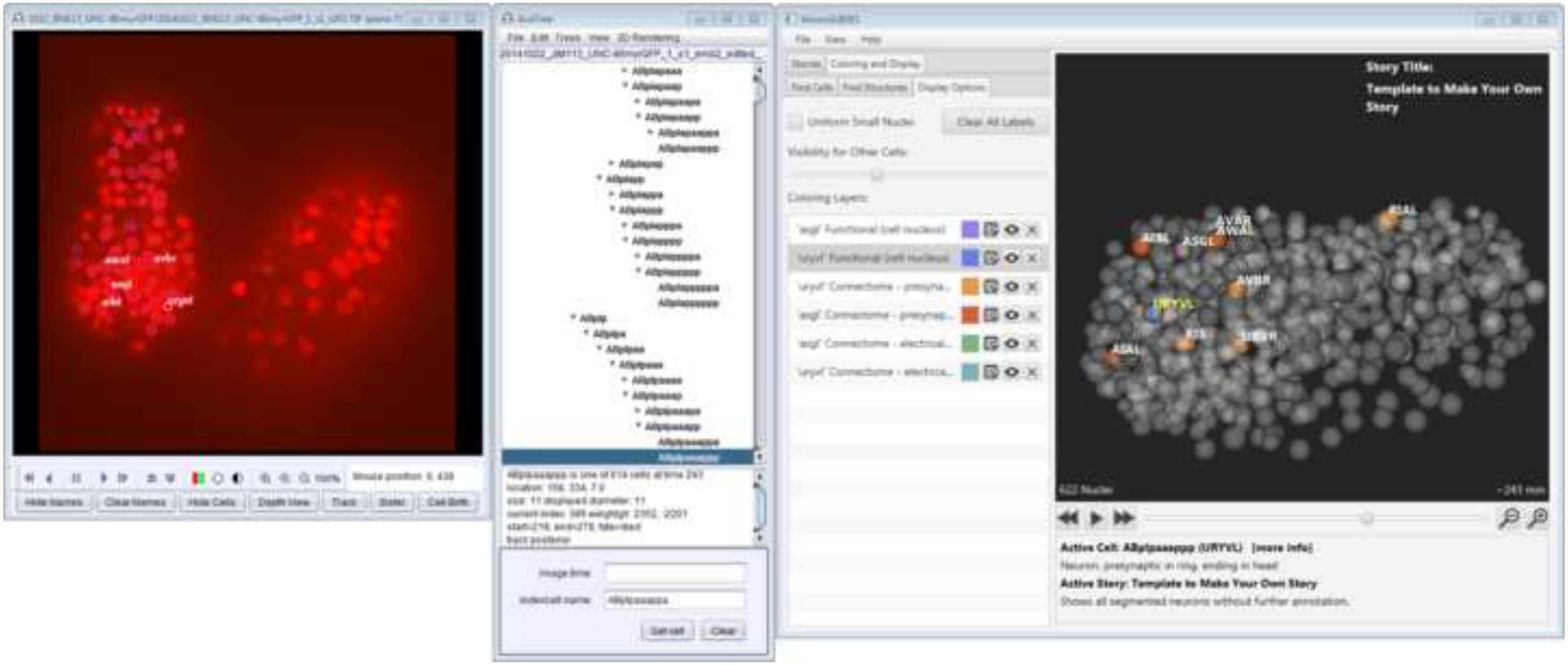
Inactive

Choose Cell as Late Timepoint

Timepoint
Cell Name

Figure 2

Click here to download Figure figure2.png



ABplpaaaapa is one of 614 cells at time 243
location: 156, 334, 7.0
size: 11 displayed diameter: 11
current index: 385 weightgfp: 2352, 2201
start=216, end=278, fate=died
track posterior

image time:
index/cell name: ABplpaaaapa
Set cell
Clear

□ Uniform Small Nuclei
Clear All Labels
Visibility for Other Cells:
Coloring Layers:

'asgl' Functional (cell nucleus)
'uryvl' Functional (cell nucleus)
'uryvl' Connectome - presyna...
'asgl' Connectome - presynap...
'asgl' Connectome - electrical...
'uryvl' Connectome - electrica...

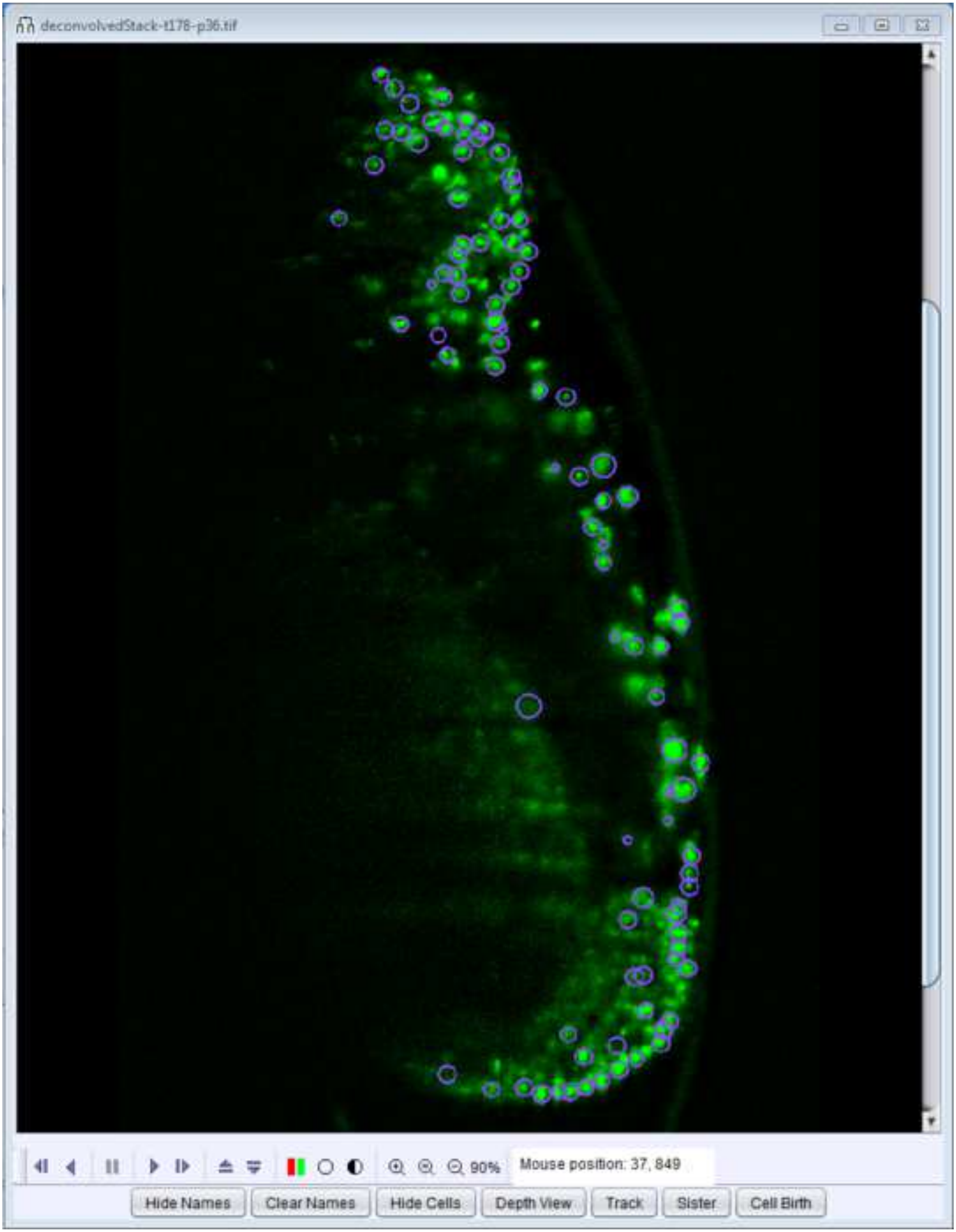
Story Title:
Template to Make Your Own Story

622 Nuclei
~243 min

Active Cell: ABplpaaappp (URYVL) [more info]
Neuron, presynaptic in ring, ending in head
Active Story: Template to Make Your Own Story
Shows all segmented neurons without further annotation.

Figure 3

Click here to download Figure figure3.png



deconvolvedStack-t178-p36.tif
90%
Mouse position: 37, 849
Hide Names
Clear Names
Hide Cells
Depth View
Track
Sister
Cell Birth