The Journal of Open Source Software

ytree: A Python package for analyzing merger trees

Britton D. Smith[1] **and Meagan Lang**[2]

1 University of Edinburgh **2** University of Illinois at Urbana-Champaign

DOI: 10.21105/joss.01881

Software

- Review
- Repository
- Archive

Editor: Juanjo Bazán

Reviewers:

- @mgckind
- @aureliocarnero

Submitted: 30 October 2019
Published: 18 December 2019

Summary

The formation of cosmological structure is dominated, especially on large scales, by the force of gravity. In the early Universe, matter is distributed homogeneously, with only small fluctuations about the average density. Overdense regions undergo gravitational collapse to form bound structures, called halos, which will host galaxies within them. Halos grow via accretion of the surrounding material and by merging with other halos. This process of merging to form increasingly massive halos is naturally conceptualized as an inverted tree, where small branches connect up to continually larger ones, leading eventually to a trunk.

One of the main products of cosmological simulations is a series of catalogs of halos within the simulated volume at different epochs. Halos within successive epochs can be linked together to create merger trees that describe a halo's growth history. An example of such a merger tree is shown in Figure 1. A variety of algorithms and software packages exist for both halo identification and merger tree calculation, resulting in a plethora of different data formats that are non-trivial to load back into memory. A range of negative consequences arise from this situation, including the difficulty of comparing methods or scientific results and users being locked into less than ideal workflows.

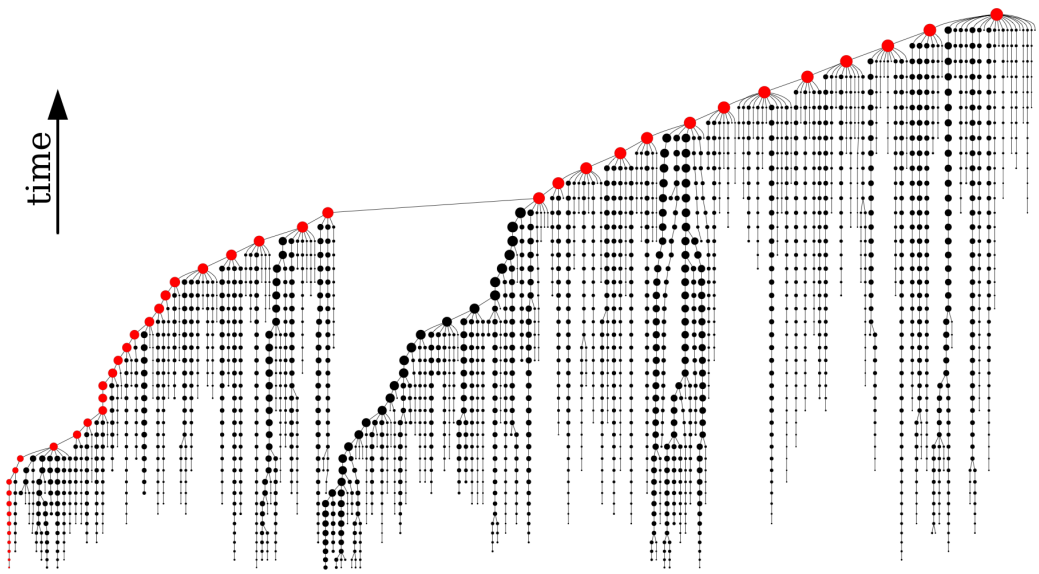

Figure 1: A visualization of a merger tree. Each circle represents a halo with lines connecting it to its descendent upward and its ancestors downward, with the size of the circle proportional to the halo's mass. Red circles denote the line of the most massive ancestors of the primary halo at a given epoch. The merger tree was created with the `consistent-trees` (Behroozi et al., 2013) merger tree code, loaded by `ytree`, and visualized with `pydot` (Pydot, 2008) and `graphviz` (Gansner & North, 2000).

The ytree package (Smith & Lang, 2018) is an extension of the yt analysis toolkit (Turk et al., 2011) for ingesting and analyzing merger tree data from multiple sources. The ytree package provides a means to load diverse merger tree data sets into common Python data structures, analogous to what yt does for spatial data. A merger tree data set loaded by ytree is returned to the user as a NumPy (van der Walt, Colbert, & Varoquaux, 2011) array of objects representing the final halo, or node, in each merger tree. Each node object contains pointers to node objects representing its immediate ancestors and descendent. This allows the user to intuitively navigate the tree structure.

Data fields, such as position, velocity, and mass, can be queried for any node object, for the entire tree stemming from a given node, or for just the line of most significant progenitors (typically the most massive). Field data are returned as unyt_quantity or unyt_array objects (Goldbaum, ZuHone, Turk, Kowalik, & Rosen, 2018), subclasses of the NumPy array with support for symbolic units. All data structure creation and field data loading is done on-demand to limit unnecessary computation. Analogous to yt, derived fields can be created as linear combinations of any existing fields by supplying a function that accepts a dictionary-like object that can be expected to contain arrays of the dependent field data. Any portion of a merger tree data set can be saved to a ytree format (based on HDF5 and using h5py (Collette, 2013)) that has somewhat faster field loading than most of the supported data formats. This also allows a subset of data to be extracted for greater portability and for saving newly created fields resulting from expensive analysis.

The ytree package has been used for semi-analytic galaxy formation models (Côté, Silvia, O'Shea, Smith, & Wise, 2018); following halo trajectories in zoom-in simulations (Hummels et al., 2019); and for studying simulated galaxy properties (Garrison-Kimmel et al., 2019; Smith et al., 2018).

Acknowledgements

Britton acknowledges the amazing yt community for being amazing as well as financial support from NSF grant AST-1615848. M. Lang would like to acknowledge the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative for supporting her contributions to this work through Grant GBMF4561 to Matthew Turk.

References

Behroozi, P. S., Wechsler, R. H., Wu, H.-Y., Busha, M. T., Klypin, A. A., & Primack, J. R. (2013). Gravitationally Consistent Halo Catalogs and Merger Trees for Precision Cosmology, *763*, 18. doi:10.1088/0004-637X/763/1/18

Collette, A. (2013). *Python and hdf5*. O'Reilly.

Côté, B., Silvia, D. W., O'Shea, B. W., Smith, B., & Wise, J. H. (2018). Validating Semi-analytic Models of High-redshift Galaxy Formation Using Radiation Hydrodynamical Simulations, *859*(1), 67. doi:10.3847/1538-4357/aabe8f

Gansner, E. R., & North, S. C. (2000). An open graph visualization system and its applications to software engineering. *SOFTWARE - PRACTICE AND EXPERIENCE*, *30*(11), 1203–1233. doi:10.1002/1097-024X(200009)30:11<1203::AID-SPE338>3.0.CO;2-N

Garrison-Kimmel, S., Wetzel, A., Hopkins, P. F., Sanderson, R., El-Badry, K., Graus, A., Chan, T. K., et al. (2019). Star formation histories of dwarf galaxies in the FIRE simulations: dependence on mass and Local Group environment, *489*(4), 4574–4588. doi:10.1093/mnras/stz2507

Goldbaum, N. J., ZuHone, J. A., Turk, M. J., Kowalik, K., & Rosen, A. L. (2018). Unyt: Handle, manipulate, and convert data with units in python. *Journal of Open Source Software*, *3*(28), 809. doi:10.21105/joss.00809

Hummels, C. B., Smith, B. D., Hopkins, P. F., O'Shea, B. W., Silvia, D. W., Werk, J. K., Lehner, N., et al. (2019). The Impact of Enhanced Halo Resolution on the Simulated Circumgalactic Medium, *882*(2), 156. doi:10.3847/1538-4357/ab378f

Pydot. (2008). Pydot. *GitHub repository*. https://github.com/pydot/pydot; GitHub.

Smith, B. D., Regan, J. A., Downes, T. P., Norman, M. L., O'Shea, B. W., & Wise, J. H. (2018). The growth of black holes from Population III remnants in the Renaissance simulations, *480*(3), 3762–3773. doi:10.1093/mnras/sty2103

Smith, B., & Lang, M. (2018, February). Ytree: Merger-tree toolkit. doi:10.5281/zenodo.1174374

Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., & Norman, M. L. (2011). yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data, *192*(1), 9. doi:10.1088/0067-0049/192/1/9

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science and Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37