High-Precision Modeling of Large-Scale Structure: An open source approach with Halotools

Andrew P. Hearin[1], Duncan Campbell[2], Erik Tollerud[2,3]
many others

[1] *Yale Center for Astronomy & Astrophysics, Yale University, New Haven, CT*
[2] *Department of Astronomy, Yale University, P.O. Box 208101, New Haven, CT*
[3] *Space Telescope Science Institute, Baltimore, MD 21218, USA*

19 May 2016

ABSTRACT

We present the first official release of Halotools, a community-driven python package designed to build and test models of the galaxy-halo connection. Halotools provides a modular platform for creating mock universes with a rich variety of models of galaxy evolution, such as the HOD, CLF, abundance matching, assembly biased models, cored/cuspy NFW profiles, velocity bias, and many other model styles and features. The package has an extensive, heavily optimized toolkit to make mock observations on a synthetic galaxy population, including galaxy clustering, galaxy-galaxy lensing, galaxy group identification, RSD multipoles, void statistics, pairwise velocities and others. Halotools is written in a object-oriented style that enables complex models to be built from a set of simple, interchangeable components, including those of your own creation. Halotools has a rigorously maintained automated testing suite and is exhaustively documented on halotools.readthedocs.org, which includes quickstart guides, source code notes and a large collection of worked examples. The documentation effectively serves as an online textbook on how to build empirical models of galaxy formation with python. We conclude this paper by describing how Halotools can be used to analyze existing datasets to obtain robust constraints on star-formation and quenching processes at low- and high-redshift, and by outlining the Halotools program to help prepare for the arrival of Stage IV dark energy experiments.

1 INTRODUCTION

Halotools is an affiliated package[1] of Astropy (Astropy Collaboration et al. 2013).

2 PACKAGE OVERVIEW

2.1 Empirical Models

2.2 Mock Observations

2.3 Managing Simulation Data

3 PACKAGE DEVELOPMENT

Halotools has been developed fully in the open since the inception of the project. Version control for the code base is managed using git[2], and the public version of the code is hosted on GitHub[3]. The latest stable version of the code can be installed via `pip install halotools`, but at any given time the `master` branch of the code on `https://github.com/astropy/halotools` may have features and performance enhancements that are being prepared for the next release. A concerted effort is made to ensure that only thoroughly tested and documented code appears in the public `master` branch, though Halotools users should be aware of the distinction between the bleeding edge version in `master` and the official release version available through `pip`.

Documentation of the code base is generated via sphinx[4] and is hosted on ReadTheDocs[5] at `http://halotools.readthedocs.io`. The public repository `https://github.com/astropy/halotools` has a webhook set up so that whenever there is a change to the `master` branch, the documentation is automatically rebuilt to reflect the most up-to-date version of `master`.

Every user-facing class, method and function in

---

[1] `http://www.astropy.org/affiliated`
[2] `http://git-scm.com`
[3] `http://www.github.com`
[4] `http://www.sphinx-doc.org`
[5] `https://readthedocs.io`

Halotools has a docstring describing its general purpose, its inputs and output, and also providing an explicit example usage. The docstring for many functions with complex behavior comes with a hyperlink to a separate section of the documentation in which mathematical derivations and algorithm notes are provided. The documentation also includes a large number of step-by-step tutorials and example analyses. The goal of these tutorials is more than simple code demonstration: the tutorials are intended to be a pedagogical tool illustrating how to analyze simulations and study models of the galaxy-halo connection in an efficient and reproducible manner.

Halotools includes hundreds of unit-tests that are incorporated into the package via the `py.test` framework.[6] These tests are typically small blocks of code that test a specific feature of a specific function. The purpose of the testing framework is both to verify scientific correctness and also to enforce that the API of the package remains stable. We also use *continuous integration*, a term referred to the automated process of running the entire test suite in a variety of different system configurations (e.g., with different releases of `Numpy` and `Astropy` installed, or different versions of the Python language). Each time any Pull Request is submitted to the `master` branch of the code,the proposed new version of the code is copied to a variety of virtual environments, and the entire test suite is run repeatedly in each environment configuration. The Pull Request will not be merged into `master` unless the entire test suite passes in all environment configurations. We use `Travis`[7] for continuous integration in Unix environments such as Linux and Mac OS X and `AppVeyor`[8] for Windows environments.

Pull Requests to the `master` branch are additionally subject to a requirement enforced by `Coveralls`.[9] This service performs a static analysis on the Halotools code base and determines the portions of the code that are covered by the test suite, making it straightforward to identify logical branches whose behavior remains to be tested. `Coveralls` issues a report for the fraction of the code base that is covered by the test suite; if the returned value of this fraction is smaller than the coverage fraction of the current version of `master`, the Pull Request is not accepted. This ensures that test coverage can only improve as the code evolves and new features are added.

4 PLANNED FEATURES

5 ACKNOWLEDGMENTS

REFERENCES

Astropy Collaboration Robitaille T. P., Tollerud E. J., Greenfield P., Droettboom M., Bray E., Aldcroft T., et al., 2013, AAP, 558, A33

---

[6] http://pytest.org
[7] https://travis-ci.org
[8] https://www.appveyor.com
[9] https://coveralls.io