hierArc Documentation

Release 1.1.1

Simon Birrer

Aug 19, 2022

CONTENTS:

Hierarchical analysis of strong lensing systems to infer lens properties and cosmological parameters simultaneously.

The software is originated from Birrer et al. 2020 and is in active development.

- Free software: BSD license
- Documentation: https://hierarc.readthedocs.io.

CHAPTER

ONE

FEATURES

The software allows to fit lenses with measured time delays, imaging information, kinematics constraints and standardizable magnifications with parameters described on the ensemble level.

CHAPTER

TWO

INSTALLATION

```
$ pip install hierarc --user
```

CHAPTER

THREE

USAGE

The full analysis of Birrer et al. 2020 is publicly available here . A forecast based on hierArc is presented by Birrer & Treu 2020 and the notebooks are available at this repository. The extension to using hierArc with standardizable magnifications is presented by Birrer et al. 2021 and the forecast analysis is publicly available here. For example use cases we refer to the notebooks of these analyses.

CHAPTER

FOUR

CREDITS

Simon Birrer & the TDCOSMO team.

Please cite Birrer et al. 2020 if you make use of this software for your research.

4.1 Installation

4.1.1 Stable release

To install hierArc, run this command in your terminal:

```
$ pip install hierarc
```

This is the preferred method to install hierArc, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

4.1.2 From sources

The sources for hierArc can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/sibirrer/hierarc
```

Or download the tarball:

```
$ curl -OJL https://github.com/sibirrer/hierarc/tarball/main
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

4.2 Usage

To use hierArc in a project:

```
import hierarc
```

4.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.3.1 Types of Contributions

Report Bugs

Report bugs at https://github.com/sibirrer/hierarc/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

Write Documentation

hierArc could always use more documentation, whether as part of the official hierArc docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/sibirrer/hierarc/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.3.2 Get Started!

Ready to contribute? Here's how to set up *hierarc* for local development.

1. Fork the *hierarc* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/hierarc.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv hierarc
$ cd hierarc/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 hierarc tests
$ python setup.py test or pytest
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/sibirrer/hierarc/pull_requests and make sure that the tests pass for all supported Python versions.

4.3.4 Tips

To run a subset of tests:

```
$ pytest tests.test_hierarc
```

4.3.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

4.4 Credits

4.4.1 Development Lead

- Simon Birrer <sibirrer@gmail.com>

4.4.2 Contributors

- Ji Won Park jiwoncpark
- Aymeric Galan aymgal

4.5 History

4.5.1 0.1.0 (2020-02-05)

- First release on PyPI.

4.5.2 1.0.0 (2020-06-29)

- First stable release.

4.5.3 1.1.0 (2021-07-26)

- Standardizable magnifications added

4.5.4 1.1.1 (2021-12-29)

- using CosmoInterp module from lenstronomy

4.6 Indices and tables

- genindex
- modindex
- search