reproducibility after the fact

Jes Ford

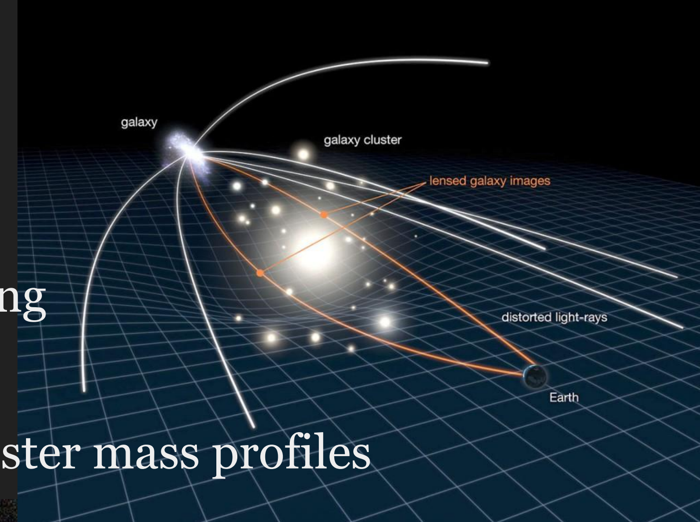
Moore/Sloan Data Science Postdoc

eScience Institute

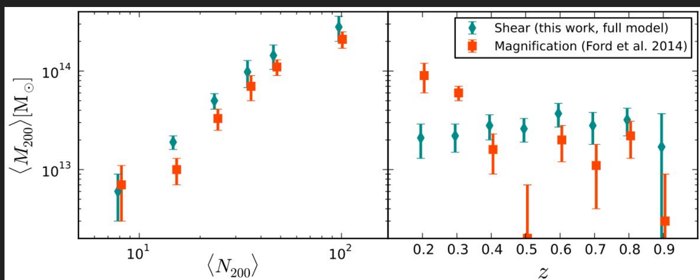
University of Washington

~~reproducibility~~: my PhD

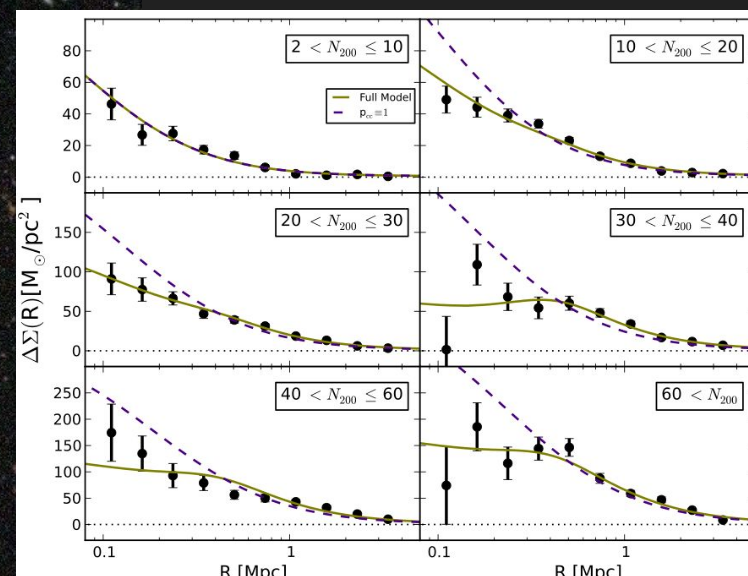
galaxy clusters & gravitational lensing



Measuring galaxy cluster mass profiles

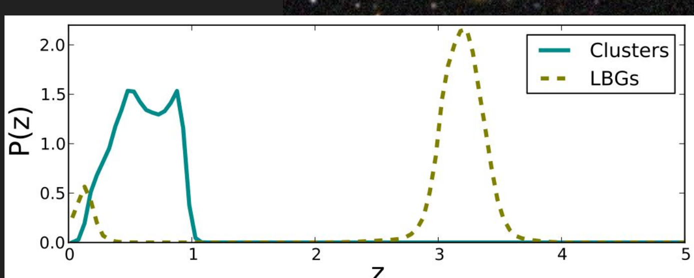


Quantifying centroid issues



Developing new approaches for measuring gravitational lensing (magnification)

Determining galaxy cluster scaling relations



Not easily reproducible by others.

Not even easily reproducible by me!

reproducibility: motivation

- Spending significant time responding to emails...
  - Students, Postdocs trying to do similar work
- Frustration that others would be reinventing same wheels
- External Examiner for my thesis encouraged code release
- Disorganized code → reuseable software, for others and myself

reproducibility: process

- Questions:
  - Which parts of the analysis should I focus on making reproducible?
  - What would be most useful to others? What is unique?
- Decision to focus on modelling
  - Most of my research time was spent in this step - lots of choices to make
  - Centroiding computationally intensive
  - No other public code exists for this (that I can find)
- Create a package
  - mix of Python, C, and shell scripts → pure Python (ease of use)
  - Documentation
  - Testing
  - Visibility / findability
  - *Learning Process!*

This repository Search Pull requests Issues Gist

jesford / cluster-lensing

Unwatch 2 | Star 1 | Fork 1

Code | Issues 1 | Pull requests 0 | Wiki | Pulse | Graphs | Settings

Galaxy Cluster and Weak Lensing Tools http://jesford.github.io/cluster-lensing — Edit

241 commits | 3 branches | 2 releases | 1 contributor

Branch: mas... | New file | Upload files | Find file | HTTPS | https://githu | Download ZIP

New pull request

jesford Merge pull request #4 from jesford/flexible-cosmology ... Latest commit e20ed2d a day ago

clusterlensing	update docstring with new cosmology type	a day ago
docs	add command to upgrade package	a month ago
.gitignore	add .gitignore	5 months ago
.travis.yml	add Python 3.4, 3.5 to Travis CI checks	a month ago
CHANGES.md	update change log with cosmology	a day ago
HOWTO_RELEASE.md	update to development version v0.1.2-git	a month ago
LICENSE.md	update year in license	a month ago
MANIFEST.in	add MANIFEST.in	2 months ago
README.md	link to paper, include pip upgrade command	22 days ago
demo.ipynb	fix typo in docstring EQ	2 months ago
requirements.txt	re-add Binder requirements	23 days ago
setup.cfg	add cfg file	2 months ago
setup.py	update to development version v0.1.2-git	a month ago

README.md

Galaxy Cluster and Weak Lensing Tools

build passing | license MIT License | powered by AstroPy

Documentation

The full **cluster-lensing** documentation is online here.

I am starting to put together a brief software paper describing this package, which I plan to submit to a journal. You can see this paper in the making, and send me feedback if you like, by going here.

Try out the **cluster-lensing** package, no commitment (no downloads) necessary. You can play with the demo notebook online from here: launch binder

Installation

cluster-lensing

Star 1

Navigation

Quick search

Go

Enter search terms or a module, class or function name.

Fork me on

Tutorial

A complete walkthrough of all the functionality and tools available in this project is provided in the Demo.

A simple example use case is as follows. Suppose you want the differential surface mass density $\Delta\Sigma(r)$ profiles for a handful of galaxy clusters. Lets say they are at redshifts $z = 0.1, 0.2$, and 0.5, and have masses of 1×10^{15}, 5×10^{14}, and $2 \times 10^{14} M_\odot$, respectively.

After installing **cluster-lensing**, all we have to do is:

```
import numpy as np
from clusterlensing import ClusterEnsemble
z = [0.1, 0.2, 0.3]
c = ClusterEnsemble(z)
c.m200 = [1e15, 5e14, 2e14]
r = np.arange(0.5, 5, 0.5)  #radial bins
c.calc_nfw(r)
```

Then the attribute **c.deltasigma_nfw** will contain an array of $\Delta\Sigma(r)$ profiles, one for each of the three clusters:

```
>>> print c.deltasigma_nfw
[[ 216.99031097  131.96892957   89.95900137   65.95785776   50.81725
    40.57785901   33.28891018   27.89244619   23.77114581]
 [ 159.82908955   88.92279328   57.75958551   41.06296211   30.95764
    24.32100583   19.69970451   16.33693743   13.80449899]
 [  99.4563379    49.5200608    30.40868664   20.87864071   15.36566
    11.85760144    9.47172553    7.76726675    6.50260522]] solMass /
```

Let's say you are concerned about the accuracy of your clusters' centroid estimates. We can easily calculate the miscentered $\Delta\Sigma(r)$ profiles by passing the optional offsets parameter to the **calc_nfw()** function. The offsets are given in units of Mpc, just like the radius, and represent the width of a 2D Gaussian offset distribution.

```
>>> c.calc_nfw(r, offsets=[0.1, 0.1, 0.1])
>>> print c.deltasigma_nfw
[[ 198.81572771  129.96652087   89.16550619   65.7334123    50.65140
    40.49991719   33.22670747   27.85454194   23.73961746]
```

reproducibility: reflections

- Reproducible?
  - roughly 50% of the work presented in my papers can now be easily reproduced
- Useful??
  - Time will tell... I know of several researchers using it now
  - (for myself, definitely, I learned a ton)
- Time investment???
  - most of my time for ~ 2 months

- Future Work: finding the balance
  - Make the other 50% of my past work reproducible?
  - OR... Focus on getting new (reproducible) results?

reproducibility: training

- To encourage reproducible research, we first need to teach it
  - Simple awareness as well as the tools
- How do we teach software skills? ... *Mostly we don't*.
  - Minimal coursework, often less applicable to our own research
  - Workshops like Software Carpentry, focusing on practical hands-on skills

- New Software Carpentry capstone (short lesson) on reproducible research
  - Primary Goal: promoting awareness and best practices around reproducibility
  - Secondary Goals: tying together previous lessons on Python, Git & GitHub, practice with documentation and licenses, exposure to pair programming and collaboration

https://github.com/jesford/python-reproducible-research software carpentry

Reproducible Research

Being able to reproduce results (whether they are from someone else or your own past self) is important for the progress of science. Making the steps and materials of a research project public, and easy to understand, allows scientists to build off of previous work to create new scientific breakthroughs faster. This lesson is a short exercise in creating a reproducible work flow, which should generate discussion on best practices and some difficulties that come up in practice.

Prerequisites

Learners should have already covered basic programming in Python, and version control with Git and GitHub, at the level of python-novice-inflammation and git-novice.

Topics

Other Resources