How Python in Astronomy has Impacted My Professional Life

Karl D. Gordon
STScI, Baltimore, MD USA
Python in Astronomy
Flatiron Institute
30 Apr 2018

kgordon@stsci.edu
@karllark2000
karllark@github

“Have Dust – Will Study”

Impact from PiA16

- Writing packages for myself & others
- Code/data public to reproduce my papers
- More example results in my papers
- Make more material public in general
  - e.g., Talks on speakerdeck
- Working harder to enable others to reproduce my work

Pre 2016

- Mainly coded in IDL and C++
- DIRTY – 3D dust radiative transfer code
  - Versions reaching back to early 1990s
  - Fortan77, then C++; one other main coder
- All my analysis code in IDL (20+ years)
- MIPS Instrument Team Pipeline
  - C++ & IDL, 3 other coders in the same place
- Only released code to collaborators

Why Python in Astronomy 2016

- Strongly encouraged by Megan Sosey to attend
- Expected to learn some python techniques
- Had attended python courses at STScI
  - Learned some terminology
  - Still using IDL for most of my work
- One of my sabbatical goals was to transition to python for some of my work

Highlights from PiA16

- Incredible enthusiasm of community
  - At all career levels (not just the young ones)
- Tutorials on techniques for packages
  - Continuous testing (Travis CI)
  - Documentation (ReadtheDocs)
  - Github issues, milestones, etc.
- Putting material on github is publishing! (Jake Vanderplas)
  - Extreme version: writing your paper on github

After PiA16

- Learning to interact with others in the github system
  - Summoning mental strength to put in pull requests to public repositories that were not my own
  - Interacting on the astropy slack space
- Talk by Erik Tollerud at STScI
  - Astropy works → existence proof and model
  - pull requests must be reviewed by others
    - Like peer review of papers, but in the open!
  - Astropy philosophy → everything in the open
    - Record of how decisions are made

Give back to Astropy

- dust_extinction
  - Potential as an affiliated package
- Become more involved
  - Ask questions on slack
  - Put in (minor) pull requests for astropy and other packages
  - Agreed to be a potential mentor for Google Summer of Code idea on astropy.modeling and MCMC

Why Dust Extinction?

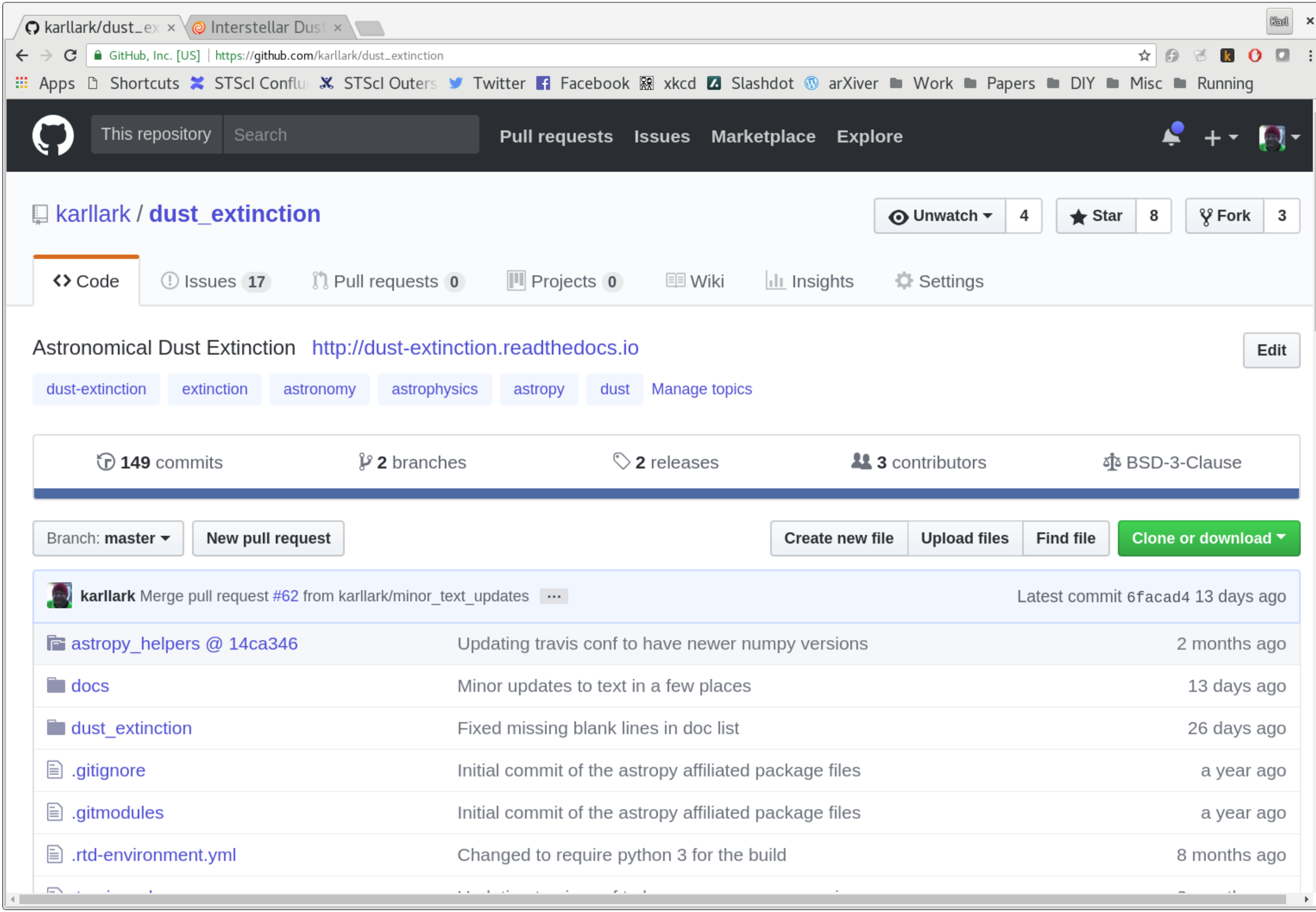
- I measure dust extinction curves
  - One of the main clues to dust grain properties
  - Gordon et al. 1998; 2003; 2009; 2018 (in prep)
- Existing python extinction packages limited
  - E.g., not all models I would use in my research
- Use astropy.modeling framework
  - “Easy” to use, fit data, etc.
- Erik T. encouraged me

dust_extinction goals

- Extinction models for:
  - Those studying dust extinction
  - Those wanting to model/correct extinction
- IDL ccm_unred and fm_unred capabilities
- Included models I often use
  - FM90 and P92 → detailed extinction shape fitting
- Learn the techniques talked about at PiA16

Use Astropy Affiliated Template

- Standing on the shoulders of giants
  - In other words, take advantage of all the hard work others have done
- Astropy affiliate template rocks!
  - Readthedocs
  - Travis CI
  - Packaging instructions for pip installable version
  - Note: cookiecutter is the way to install template!
- Being an official astropy affiliated package not required


karllark/dust_ex
Interstellar Dust
GitHub, Inc. [US] | https://github.com/karllark/dust_extinction
Apps
Shortcuts
STScI Conflu
STScI Outers
Twitter
Facebook
xkcd
Slashdot
arXiver
Work
Papers
DIY
Misc
Running
This repository
Search
Pull requests
Issues
Marketplace
Explore
karllark / dust_extinction
Unwatch 4
Star 8
Fork 3
Code
Issues 17
Pull requests 0
Projects 0
Wiki
Insights
Settings
Astronomical Dust Extinction http://dust-extinction.readthedocs.io
Edit
dust-extinction
extinction
astronomy
astrophysics
astropy
dust
Manage topics
149 commits
2 branches
2 releases
3 contributors
BSD-3-Clause
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
karllark Merge pull request #62 from karllark/minor_text_updates
Latest commit 6facad4 13 days ago
astropy_helpers @ 14ca346
Updating travis conf to have newer numpy versions
2 months ago
docs
Minor updates to text in a few places
13 days ago
dust_extinction
Fixed missing blank lines in doc list
26 days ago
.gitignore
Initial commit of the astropy affiliated package files
a year ago
.gitmodules
Initial commit of the astropy affiliated package files
a year ago
.rtd-environment.yml
Changed to require python 3 for the build
8 months ago

karllark/dust_ex × Interstellar Dust ×

dust-extinction.readthedocs.io/en/latest/

Apps Shortcuts STScI Conflu STScI Outers Twitter Facebook xkcd Slashdot arXiver Work Papers DIY Misc Running

dust_extinction:docs astropy Index Modules Search

Page Contents

Interstellar Dust Extinction

`dust_extinction` is a python package to provide interstellar dust extinction curves.

While there are other python packages that provide some of the extinction curves contained here, the explicit motivation for this package is to provide extinction curves to those using them to model/correct their data and those studying extinction curves directly to better undertand interstellar dust.

This package is developed in the astropy affiliated package template and uses the astropy.modeling framework.

User Documenation

- Flavors of Models
    - Average models
    - R(V) (+ other variables) dependent prediction models
    - Shape fitting models
- Extinguish (or unextinguish) data
    - Example: Extinguish a Blackbody
- Fitting extinction curves

v: latest

This package is developed in the astropy affiliated package template and uses the astropy.modeling framework.

User Documenation

Installation

Quick Start

How to extinguish (redden) and unextinguish (deredden) a spectrum:

Generate a spectrum to use. In this case a blackbody model, but can be an observed spectrum. The **dust_extinction** models are unit aware and the wavelength array should have astropy.units associated with it.


```
import numpy as np
from astropy.modeling.blackbody import blackbody_lambda
import astropy.units as u
```

```
Example: FM90 Fit
=================

In this example, the FM90 model is used to fit the observed average
extinction curve for the LMC outside of the LMC2 supershell region
(G03_LMCAvg ``dust_extinction`` model).

.. plot::
   :include-source:

   import matplotlib.pyplot as plt
   import numpy as np

   from astropy.modeling.fitting import LevMarLSQFitter

   from dust_extinction.dust_extinction import (G03_LMCAvg, FM90)

   # get an observed extinction curve to fit
   g03_model = G03_LMCAvg()

   x = g03_model.obsdata_x
   # convert to E(x-V)/E(B0V)
   y = (g03_model.obsdata_axav - 1.0)*g03_model.Rv
   # only fit the UV portion (FM90 only valid in UV)
   gindxs, = np.where(x > 3.125)

   # initialize the model
   fm90_init = FM90()

   # pick the fitter
   fit = LevMarLSQFitter()

   # fit the data to the FM90 model using the fitter
   #   use the initialized model as the starting point
   g03_fit = fit(fm90_init, x[gindxs], y[gindxs])
```

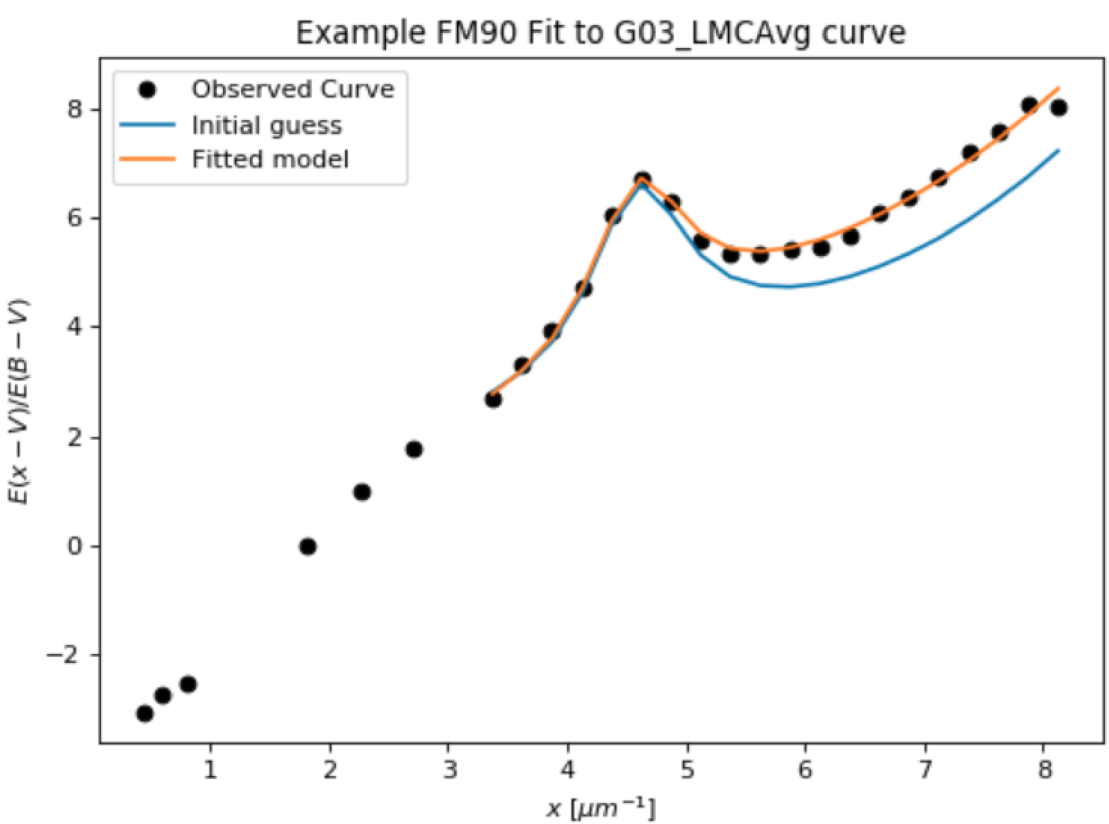
Example: FM90 Fit

In this example, the FM90 model is used to fit the observed average extinction curve for the LMC outside of the LMC2 supershell region (G03_LMCAvg `dust_extinction` model).

```
import matplotlib.pyplot as plt
import numpy as np

from astropy.modeling.fitting import LevMarLSQFitter

from dust_extinction.dust_extinction import (G03_LMCAvg, FM90)

# get an observed extinction curve to fit
g03_model = G03_LMCAvg()

x = g03_model.obsdata_x
# convert to E(x-V)/E(B0V)
y = (g03_model.obsdata_axav - 1.0)*g03_model.Rv
# only fit the UV portion (FM90 only valid in UV)
gindxs, = np.where(x > 3.125)

# initialize the model
fm90_init = FM90()

# pick the fitter
fit = LevMarLSQFitter()

# fit the data to the FM90 model using the fitter
#   use the initialized model as the starting point
g03_fit = fit(fm90_init, x[gindxs], y[gindxs])

# plot the observed data, initial guess, and final fit
fig, ax = plt.subplots()
```

```
ax.set_xlabel('$x$ [$\mu m^{-1}$]')
ax.set_ylabel('$E(x-V)/E(B-V)$')

ax.set_title('Example FM90 Fit to G03_LMCAvg curve')

ax.legend(loc='best')
plt.tight_layout()
plt.show()
```

(Source code, png, hires.png, pdf)



How to Choose a Model

The `dust_extinction` package provides a suite of dust extinction models. Which model to use can depend on the wavelength range of interest, the expected type of extinction, or some other property.

Average Models

Simple Average Curves

These are straightforward averages of observed extinction curves. They are the simplest models and include models for the MW (**GCC09_MWAvg**), the LMC (**G03_LMCAvg**, **G03_LMC2**) and the SMC (**G03_SMCBar**).

One often used alternative to these straight average models is to use one of the parameter dependent models with the average R(V) value. For the Milky Way, the usual average used is R(V) = 3.1.

Model	x range	wavelength range	galaxy
GCC09_MWAvg	0.3 - 10.96	0.0912 - 3.3	MW
G03_LMCAvg	0.3 - 10.0	0.1 - 3.3	LMC
G03_LMC2	0.3 - 10.0	0.1 - 3.3	LMC (30 Dor)
G03_SMCBar	0.3 - 10.0	0.1 - 3.3	SMC

Parameter Dependent Average Curves

The models that are dependent on parameters provide average curves that account for overall changes in the extinction curve shapes. For example, the average behavior of Milky Way extinction curves has been shown to be dependent on R(V) = A(V)/E(B-V). R(V) roughly tracks with the average dust grain size.

The most general model is **G16** as this model encompasses the average measured behavior of extinction curves in the MW, LMC, and SMC. The **G16** model reduces to the **F99** model with f_A= 1.0. If only MW type extinction is expected, then the **F04** model should be considered as it is based on significantly more extinction curves than the **CCM89** or **O94** models.

Model	Parameters	x range [1/micron]	wavelength range [micron]	galaxy

Parameter Dependent Average Curves

The models that are dependent on parameters provide average curves that account for overall changes in the extinction curve shapes. For example, the average behavior of Milky Way extinction curves has been shown to be dependent on R(V) = A(V)/E(B-V). R(V) roughly tracks with the average dust grain size.

The most general model is **G16** as this model encompasses the average measured behavior of extinction curves in the MW, LMC, and SMC. The **G16** model reduces to the **F99** model with f_A= 1.0. If only MW type extinction is expected, then the **F04** model should be considered as it is based on significantly more extinction curves than the **CCM89** or **O94** models.

Model	Parameters	x range [1/micron]	wavelength range [micron]	galaxy
CCM89	R(V)	0.3 - 10.0	0.1 - 3.3	MW
O94	R(V)	0.3 - 10.0	0.1 - 3.3	MW
F99, F04	R(V)	0.3 - 10.0	0.1 - 3.3	MW
G16	R(V)_A, f_A	0.3 - 10.0	0.1 - 3.3	MW, LMC, SMC

Shape Models

The models that focus on describing the full extinction curve shape are usually used to fit measured extinction curves. These models allow features in the extinction curve to be measured (e.g., 2175 A bump or 10 micron silicate feature). The **P92** is the most general as it covers the a very broad wavelength range. The **FM90** model has been extensively used, but only covers the UV wavelength range.

Model	x range	wavelength range	# of parameters
FM90	3.13 - 11.0	0.0912 - 0.32	6
P92	0.001 - 1000	0.001 - 1000	19 (24 possible)

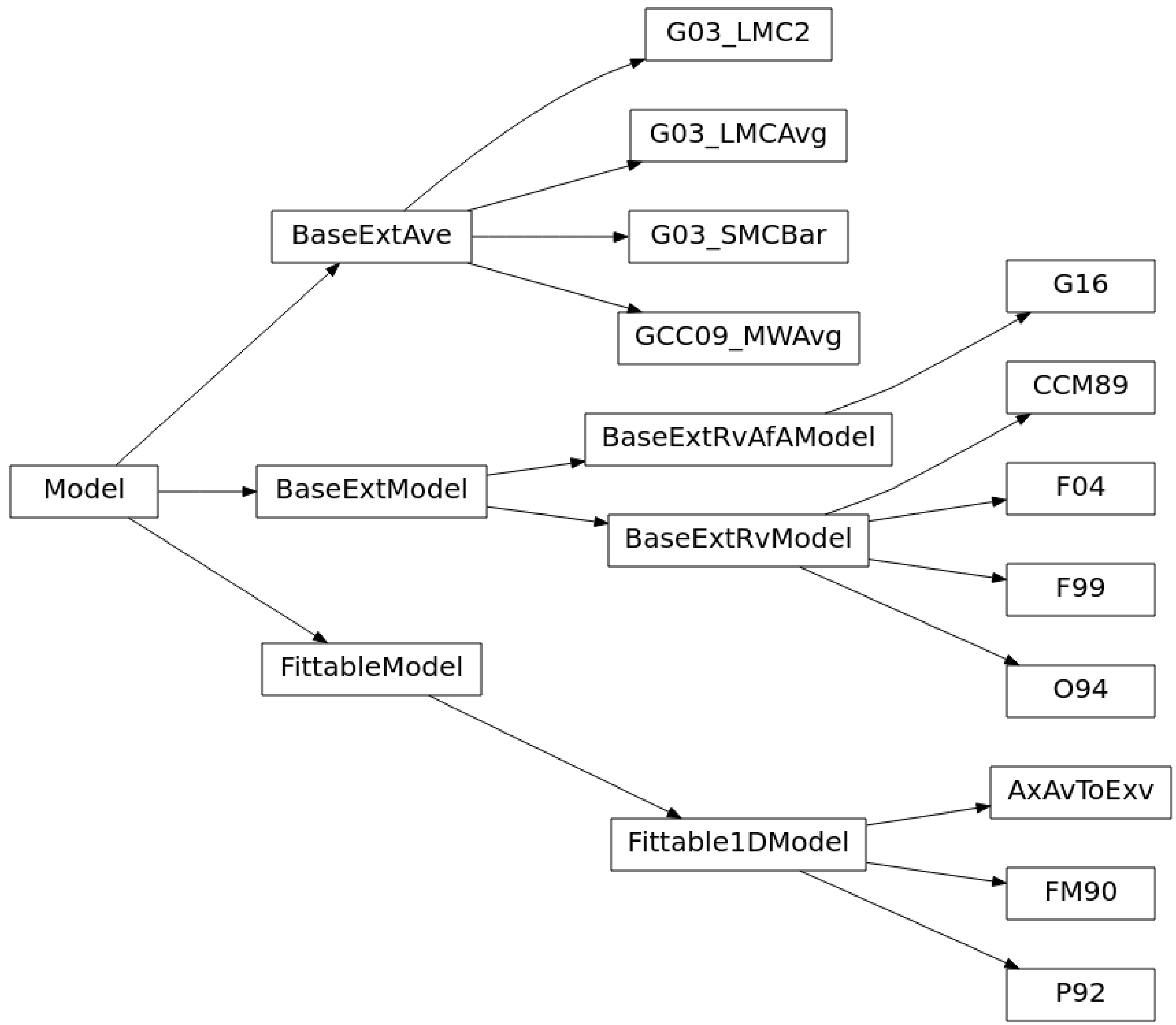
v: latest

Interstellar Dust

dust-extinction.readthedocs.io/en/latest/

Apps Shortcuts STScI Conflu STScI Outers Twitter Facebook xkcd Slashdot arXiver

Class Inheritance Diagram



v: latest

Page Source Back to Top

Created using Sphinx 1.7.2. Last built 11 Apr 2018.

F04

class `dust_extinction.dust_extinction.`**`F04`**(*Rv=3.1*, ***kwargs*) [source]

Bases: **`dust_extinction.dust_extinction.BaseExtRvModel`**

F99 extinction model calculation

Updated with the NIR Rv dependence in
Fitzpatrick (2004, ASP Conf. Ser. 309, Astrophysics of Dust, 33)

See also Fitzpatrick & Massa (2007, ApJ, 663, 320)

Parameters: **Rv: float**

R(V) = A(V)/E(B-V) = total-to-selective extinction

Raises: **InputParameterError**

Input Rv values outside of defined range

Notes

F99 Milky Way R(V) dependent extinction model

From Fitzpatrick (1999, PASP, 111, 63)

Updated with the NIR Rv dependence in
Fitzpatrick (2004, ASP Conf. Ser. 309, Astrophysics of Dust, 33)

See also Fitzpatrick & Massa (2007, ApJ, 663, 320)

Example showing F04 curves for a range of R(V) values

Extinguish/Unextinguish

(Redden/deredden)


```
from dust_extinction.dust_extinction import F99

# define the model
ext = F99(Rv=3.1)
```

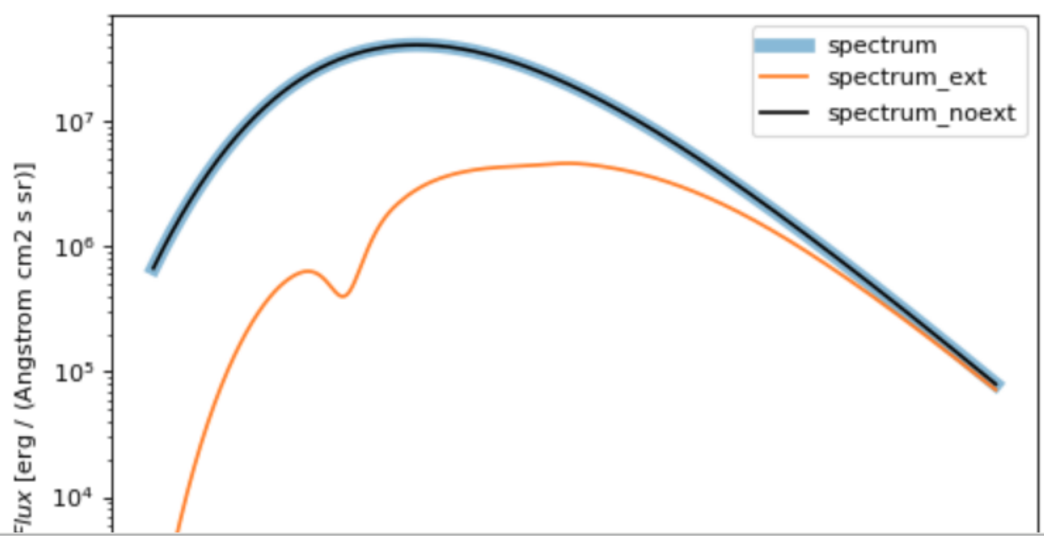
Extinguish (redden) a spectrum with a screen of F99 dust with an E(B-V) of 0.5. Can also specify the dust column with Av (this case equivalent to Av = 0.5*Rv = 1.55).

```
# extinguish (redden) the spectrum
spectrum_ext = spectrum*ext.extinguish(wavelengths, Ebv=0.5)
```

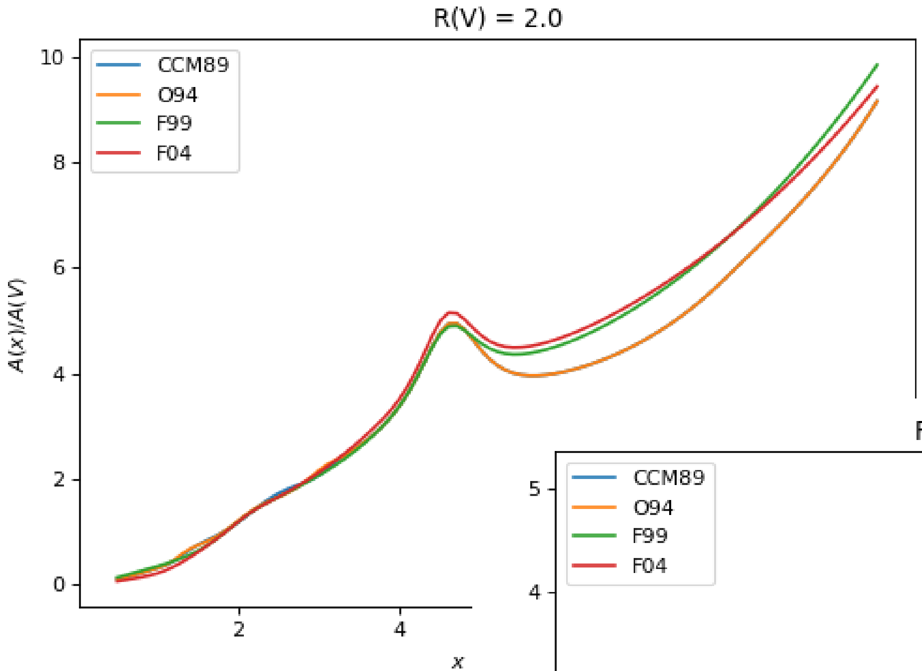
Unextinguish (deredden) a spectrum with a screen of F99 dust with the equivalent A(V) column.

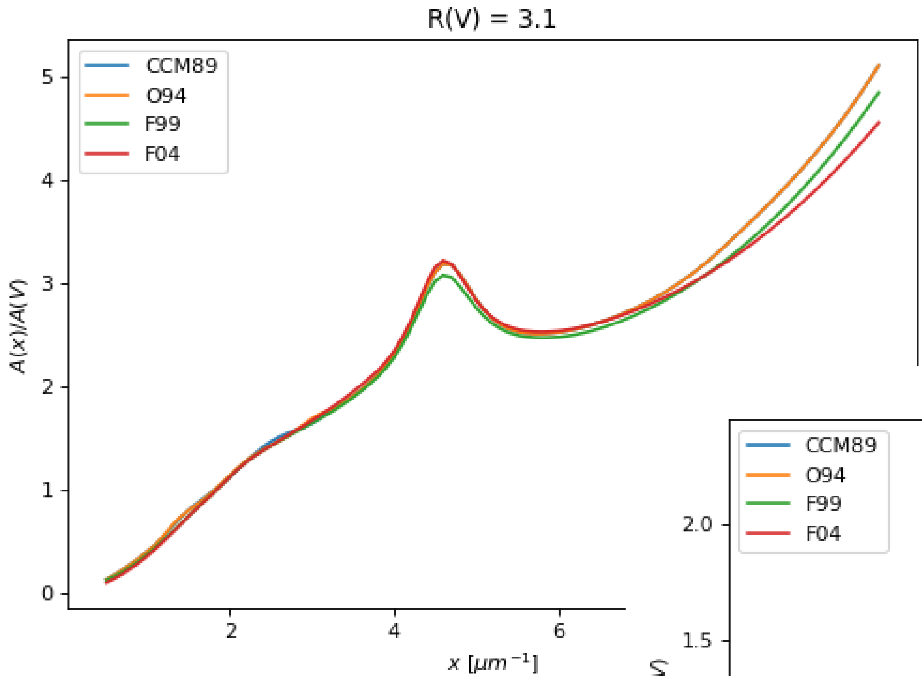
```
# unextinguish (deredden) the spectrum
spectrum_noext = spectrum_ext/ext.extinguish(wavelengths, Av=1.55)
```

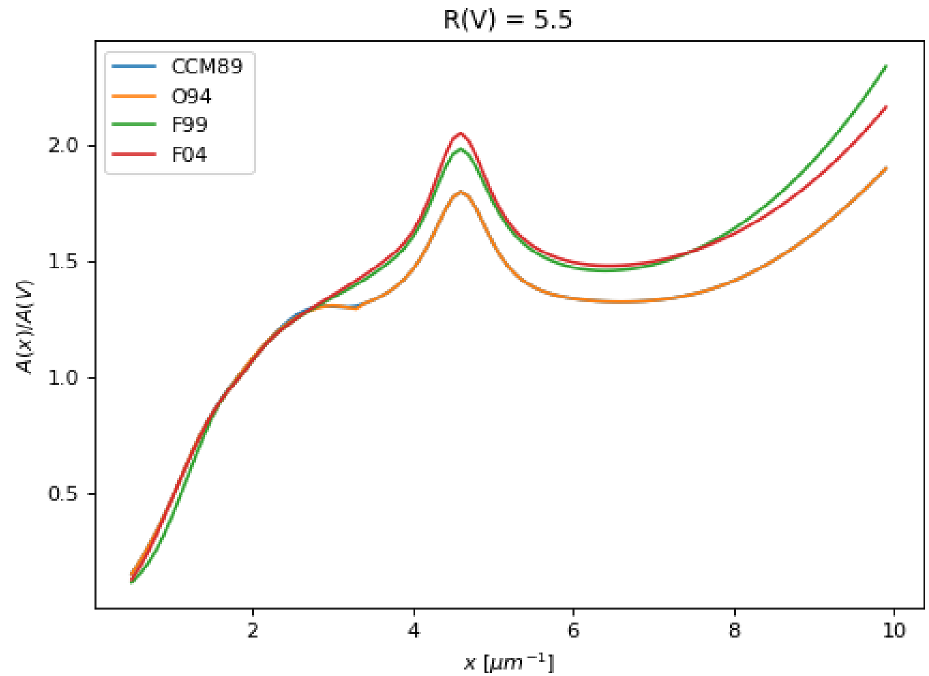
(Source code, png, hires.png, pdf)



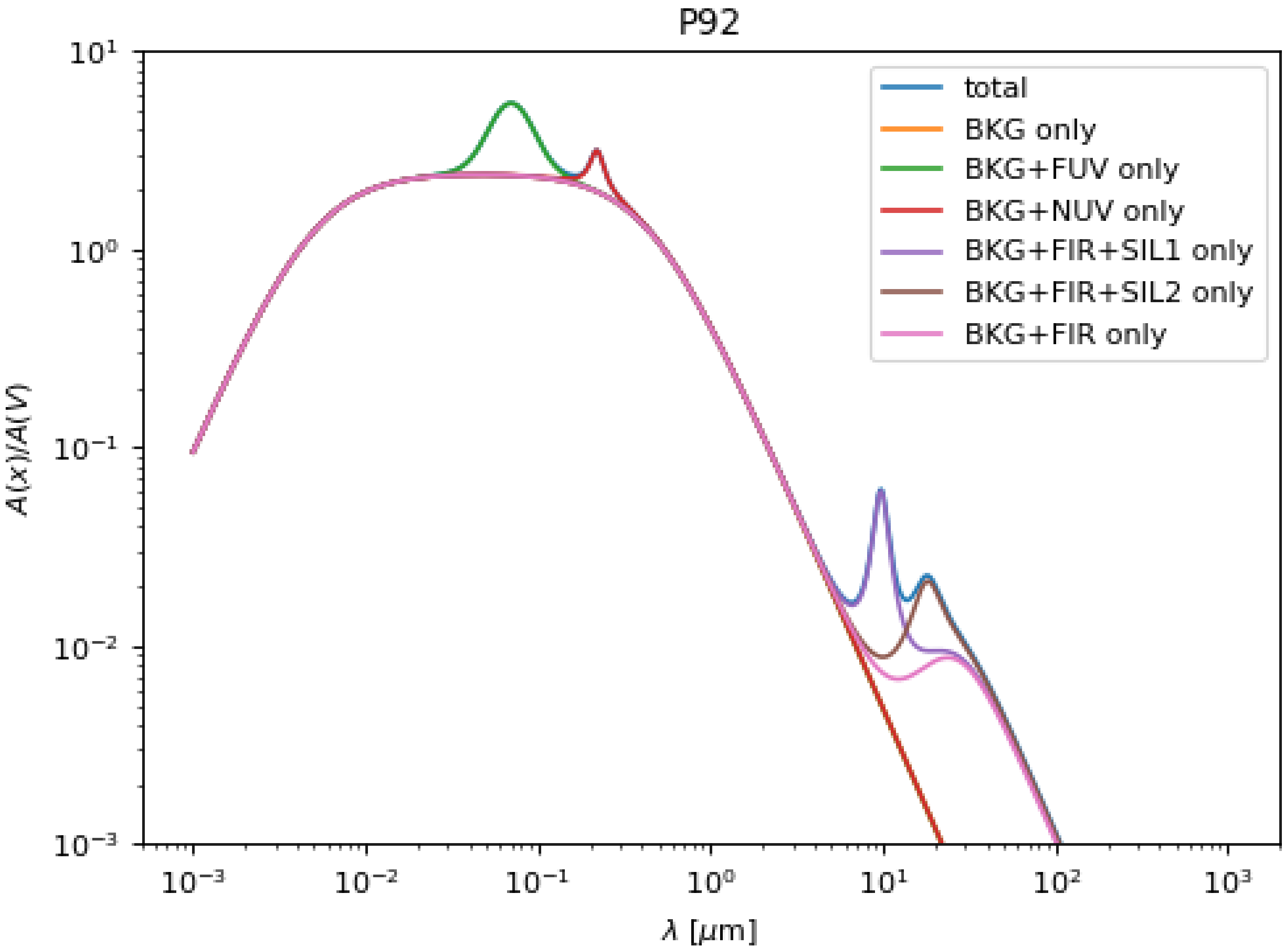
Easily compare different models





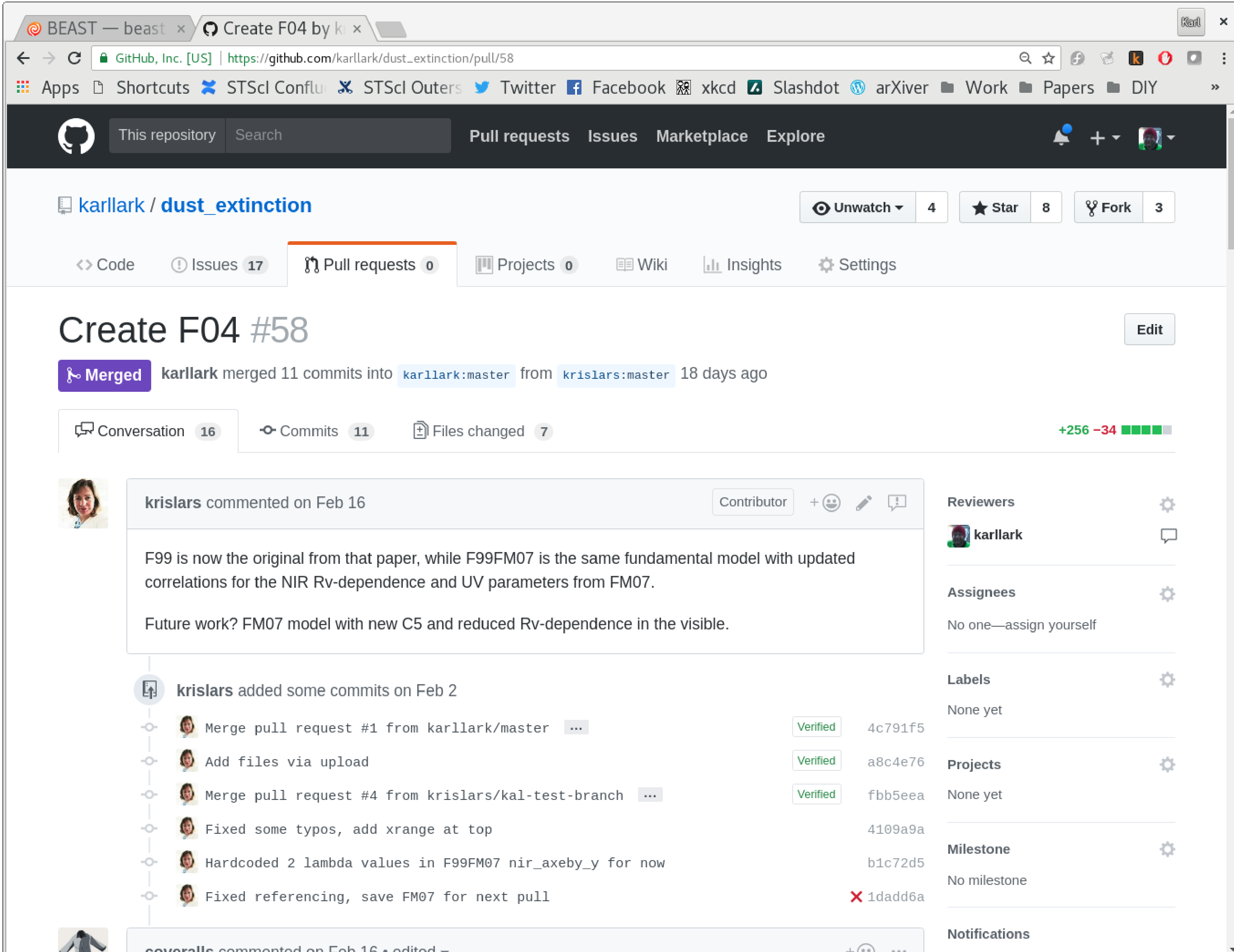


Easy to make illustrations



Tests vs **published** values

```python
@pytest.mark.parametrize("x_invalid_angstrom",
                         u.angstrom*1e4/[-1.0, 0.2, 10.1, 100.])
def test_invalid_micron(x_invalid_angstrom):
    _invalid_x_range(x_invalid_angstrom, CCM89(Rv=3.1), 'CCM89')


def test_axav_ccm89_table3():
    # values from Table 3 of Cardelli et al. (1989)
    #   ignoring the last value at L band as it is outside the
    #   valid range for the relationship
    #  updated for correction for incorrect value in the table for B band
    #    correction from Geoff Clayton via email
    x = np.array([2.78, 2.27, 1.82, 1.43,
                  1.11, 0.80, 0.63, 0.46])
    cor_vals = np.array([1.569, 1.322, 1.000, 0.751,
                         0.479, 0.282, 0.190, 0.114])

    # initialize extinction model
    tmodel = CCM89(Rv=3.1)

    # test (table in paper has limited precision)
    np.testing.assert_allclose(tmodel(x), cor_vals, atol=1e-2)


def get_axav_cor_vals(Rv):
    # testing wavenumbers
```


BEAST — beast
Create F04 by k
GitHub, Inc. [US] https://github.com/karllark/dust_extinction/pull/58
Apps Shortcuts STScI Conflu STScI Outers Twitter Facebook xkcd Slashdot arXiver Work Papers DIY
This repository Search
Pull requests Issues Marketplace Explore
karllark / dust_extinction
Unwatch 4 Star 8 Fork 3
Code Issues 17 Pull requests 0 Projects 0 Wiki Insights Settings
Create F04 #58
Edit
Merged karllark merged 11 commits into karllark:master from krislars:master 18 days ago
Conversation 16 Commits 11 Files changed 7
+256 −34
krislars commented on Feb 16
Contributor
F99 is now the original from that paper, while F99FM07 is the same fundamental model with updated correlations for the NIR Rv-dependence and UV parameters from FM07.
Future work? FM07 model with new C5 and reduced Rv-dependence in the visible.
krislars added some commits on Feb 2
Merge pull request #1 from karllark/master … Verified 4c791f5
Add files via upload Verified a8c4e76
Merge pull request #4 from krislars/kal-test-branch … Verified fbb5eea
Fixed some typos, add xrange at top 4109a9a
Hardcoded 2 lambda values in F99FM07 nir_axeby_y for now b1c72d5
Fixed referencing, save FM07 for next pull 1dadd6a
coveralls commented on Feb 16 • edited
Reviewers
karllark
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications

More Extinction Packages

- measure_extinction
  - Put code **and** documention/howto create extinction curves
  - Export expert user knowledge
  - Writing for postdocs/students in my group – but publicly!
- measured_extcurves
  - Put all the data and resulting extinction curves
  - Occasionally asked for such and often I cannot quickly find the requested info → so never provide

Extinction versus Attenuation

- Education opportunity
- One package vs <u>two packages</u>
  - Discussion in dust_extinction issue (and literature)
- Extinction = absorption and scattering out of line of sight
  - Shape independent on amount of dust
- Attenuation = extinction + scattering into observation
  - Includes radiative transfer effects
  - Generally for regions of galaxies with many stars
  - Shape depends on amount of dust
- Docs a place to put information hard to publish in a “regular” paper

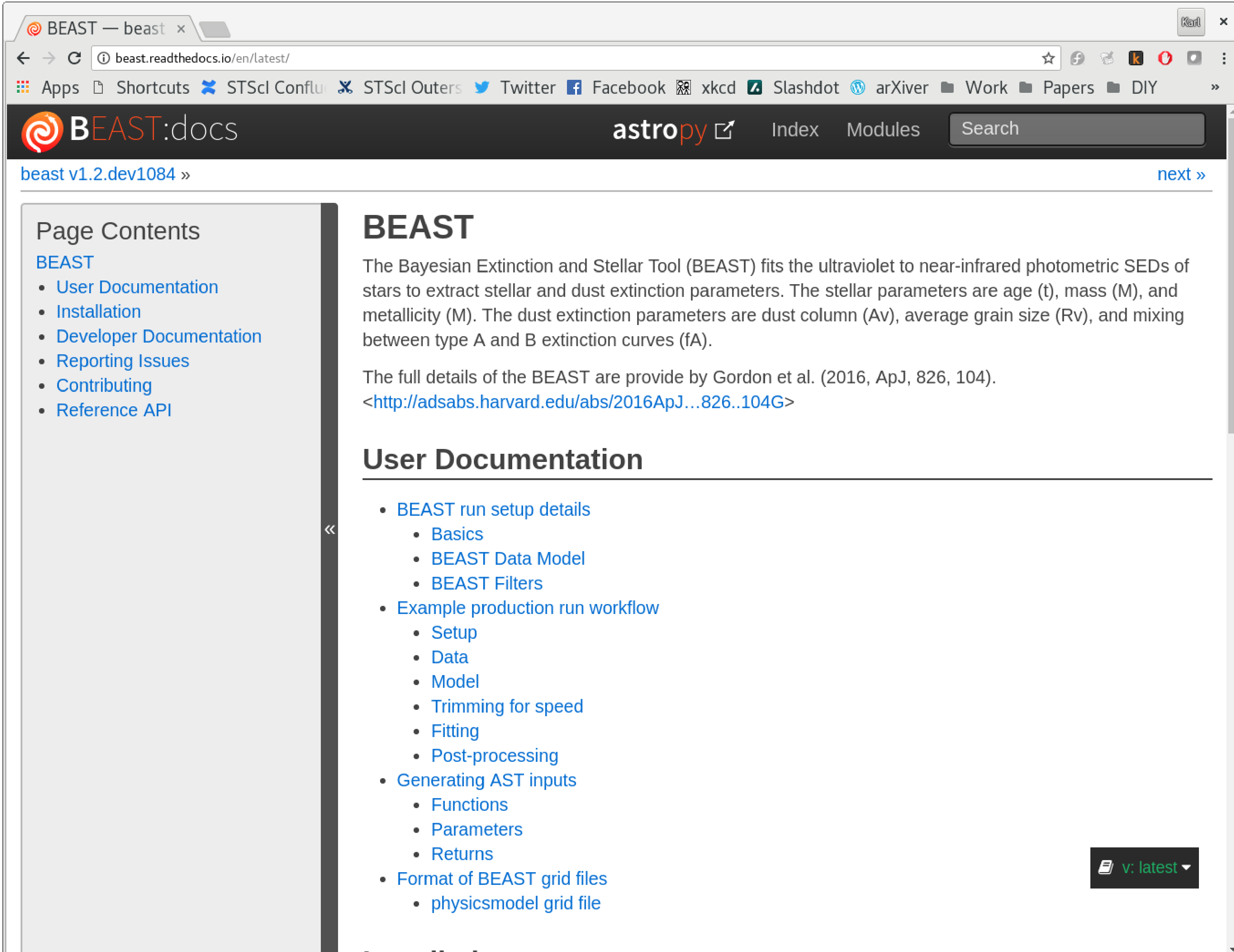
BEAST

- Bayesian Extinction and Stellar Tool
- SED fitting of individual star in galaxies
  - Motivated by PHAT M31 observations – 100+ million(!) stars
- Started in IDL (2010)
- Converted to python (2012) by others
  - Now I had to learn python for real!
  - Finally ready for production runs in early 2016
- After PiA16
  - Push to make public

Motivations to make BEAST public
(from most to least important)

- Future-self (!!)
- Students
- Postdocs
- Collaborators
- Random Strangers

- Documentation, testing, etc.



Page Contents

BEAST

- User Documentation
- Installation
- Developer Documentation
- Reporting Issues
- Contributing
- Reference API

BEAST

The Bayesian Extinction and Stellar Tool (BEAST) fits the ultraviolet to near-infrared photometric SEDs of stars to extract stellar and dust extinction parameters. The stellar parameters are age (t), mass (M), and metallicity (M). The dust extinction parameters are dust column (Av), average grain size (Rv), and mixing between type A and B extinction curves (fA).

The full details of the BEAST are provide by Gordon et al. (2016, ApJ, 826, 104). <http://adsabs.harvard.edu/abs/2016ApJ…826..104G>

User Documentation

- BEAST run setup details
  - Basics
  - BEAST Data Model
  - BEAST Filters
- Example production run workflow
  - Setup
  - Data
  - Model
  - Trimming for speed
  - Fitting
  - Post-processing
- Generating AST inputs
  - Functions
  - Parameters
  - Returns
- Format of BEAST grid files
  - physicsmodel grid file

BEAST Lessons

- Already established code
  - Conversion to python 3 was challenging
- Add in regression tests (confidence building)
- Less code is better (!!)
- Use community code where possible
  - Someone else helps with the support!
- Requests for features → create an issue
  - Encourage others to implement

Some Personal Motivations

- Get others to work with you
  - Recognition is key!
- Be able to find my work later
  - Searching 10 year old directories is hard
- Requires cleaner, clearer code and docs
  - Like writing a refereed paper versus notes
  - Greatly beneficial to self/students/collaborators

Paper writing changes

- Include results of an example calculation
  - Table of values
  - Useful for CI testing
- Include links to github code for each paper
  - Figures, models, etc.
- Put data/results in Zenodo and link
- Dust Extinction
  - Generate average curves, publish in tables
  - Provide example results for R(V)+ relationships
  - Need to generate more averages (inc. a full multiwavelength)

Github Issues

- Write down ideas right away
- Add comments as more thoughts arrive
- Provide a todo list for yourself and others
- Experimenting with using issues to track/motivate my next paper
  - My job includes many interruptions

Use my results, please!

- How to get others to use my results?
  - And cite my papers!
- Doing good work not necessarily enough
  - Can be hard to understand or easily use in own research
  - Witt & Gordon (2000) attenuation curves
- Publishing tables and relationships helps
  - CCM89 → many citations!
- Generating a python package?
  - Easy to use and well documented
  - Python is the emerging standard

Punchline

- Open source your software
  - For yourself, your collaborators, & random people
  - Posting on github **is** publishing
  - Others can reproduce your work
- Use astropy affiliated template if python
  - Documentation → ReadtheDocs
  - Continuous Testing → Travis CI
  - Good coding guidelines → PEP8
  - Packaging instructions → pip installable

Thanks