

A **Python** package for **gamma-ray** astronomy

GETTING STARTED & MISC INFOS & STATUS & PLANS & DISCUSSION

Christoph Deil, MPIK Heidelberg
January 13, 2017



Gammapy



- A prototype for the CTA science tools (proposed in Kashiwa)
- A Python package for gamma-ray astronomy, built on Python, Numpy, Scipy, Astropy
- 27k lines of code, 21 contributors
- Used for H.E.S.S. (e.g. Galactic plane survey), Fermi-LAT and CTA
- Will participate in CTA data challenge
- See [presentation by Christoph Deil](#) and [docs.gammapy.org](#)



This is the summary slide from Oct 2017 CTA consortium meeting.
I will not go over that content again today. See link to slides above.

OUTLINE FOR TODAY'S CALL

- Some misc info on Python/Astropy/Gammapy ...
 - How to get started as a user or contributor
 - Status and plans of some packages / projects
- Open discussion
 - Your thoughts & wishes & plans!
 - Attempt to schedule meeting(s).

SOFTWARE USE IN ASTRONOMY — AN INFORMAL SURVEY

2015arXiv150703989M

Do you use software in your research?

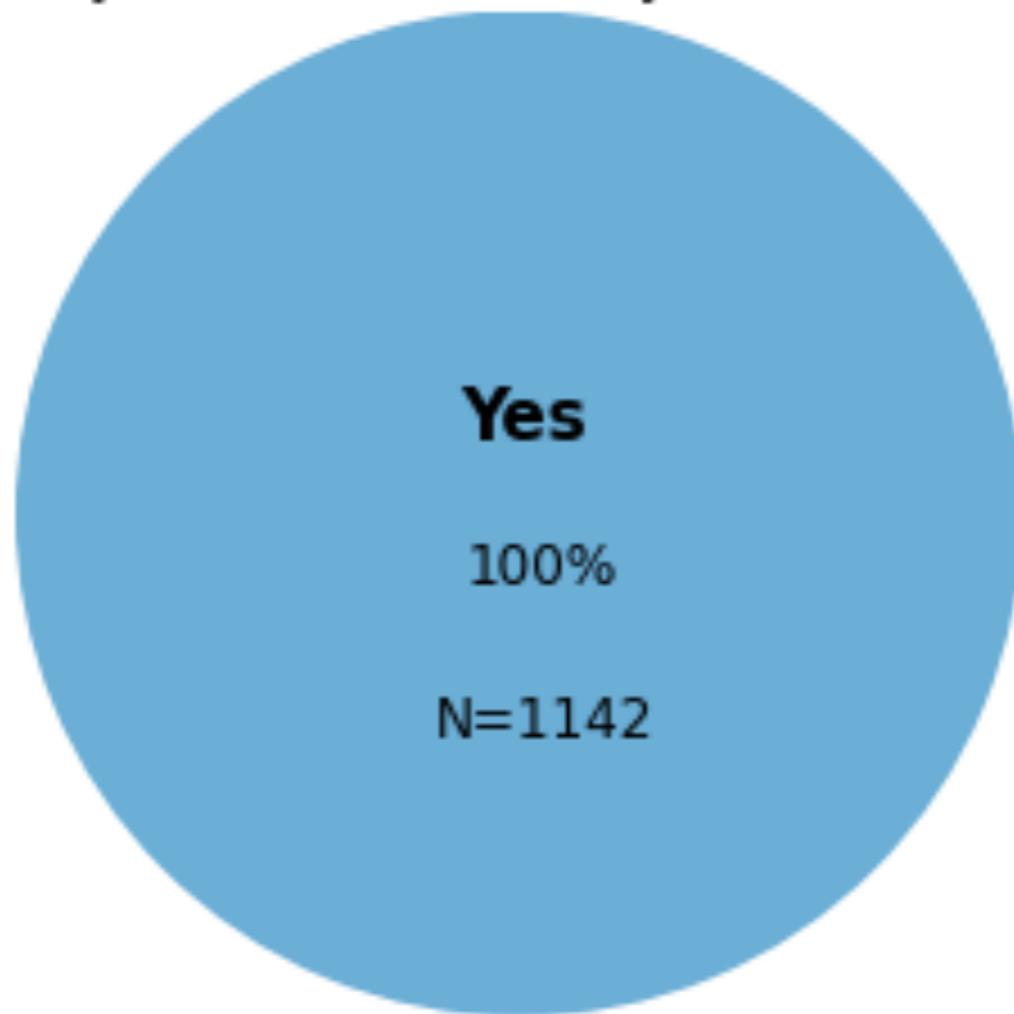


FIG. 2.— Responses to the question “Do you use software in your research?”. 100% of survey participants answered in the affirmative.

GETTING STARTED IS HARD ...

- So many new things to learn before you're a productive gamma-ray astronomer these days ...
 - Python — def, class, import, self, list, dict, .py
 - IPython — terminal, notebook, commands
 - Numpy — array-oriented computing
 - Scipy, scikit-image, scikit-learn
 - what is available? which algorithms are useful to me?
 - matplotlib — pyplot, pylab, figure, artist, backend, ...
 - Astropy — SkyCoord, Time, Table, BinTable, ...
 - Sherpa — fitting, modeling, low- and high-level API
 - Tools — pytest, Sphinx, pip, conda, Jupyter, git, Github
 - Formats — FITS, ECSV, YAML, JSON, ...
(no stable standard formats for gamma-ray astronomy yet)
 - Gammapy, ctapipe, Fermipy, Naima, ...

... BUT REALLY IT'S NOT SO BAD.

- Some things are better, some worse compared to the C++/
ROOT world or IDL world or whatever you used up to now.
- Overall Python:
 - nice high-level language
 - good for interactive data analysis and scripting
 - good for large libraries and apps
 - widely used in science and industry



PYTHON

.....



- <https://www.python.org/>
- Created 25 years ago by Guido van Rossum.
- Named after Monty Python, a British sketch comedy troupe.
- If you don't know them, go watch some clips on YouTube!
- Python snake only (reluctantly) accepted as logo much later.
- Action item: from now on, start every Gammapy meeting with a Monty Python sketch!



*Python 2 doesn't work
if you're French!*

```
$ python2
>>> len('Léa')
4
```

```
$ python3
>>> len('Léa')
3
```

*Supporting Python
2 & 3 is easy as pie.
Use common subset.*

```
# Only Python 2
>>> print 'Hi'

# Both Python 2 & 3
>>> print('Hi')
```

PYTHON 2 OR 3?

- Python 3 better in many small ways,
hard to explain in 1 min.
- Python 3.6 came out Dec 2016.
~ 20% faster and smaller dictionaries.
- ctapipe is Python 3 only.
- Many packages will drop Python 2 support soon:
python3statement.org
- Sherpa supports Python 3 since 2016.
- Gammapy will keep Python 2.7 support for now,
mainly because the Fermi ST / Fermipy don't
support Python 3 yet.
- **Recommendation:** use Python 3 (3.4 or later)
(unless you're using Fermi ST, for now)

WHICH PYTHON?

- System **/bin/python** or **/usr/bin/python** is what your operating system uses. Often a version that's 10 or more years old.
Usually a terrible choice to install your scientific software stack.
- Python packaging is complex and weird in detail.
(This is not a Python-specific issue. It's the same for most languages running on millions of computers since decades (e.g. Fortran, C, C++). New languages sometimes are better.)
- But Python packaging has hugely improved in recent years.
Works well for most users, most of the time.
 - `conda install numpy scipy astropy`
 - `pip install antigravity`
- But installing software is still confusing and sometimes fails.
Need to train users and developers how pip, conda and import work.

PIP INSTALL ANTIGRAVITY

```
$ pip install antigravity --user  
Collecting antigravity  
  Downloading antigravity-0.1.zip  
Installing collected packages: antigravity  
  Running setup.py install for antigravity ... done  
Successfully installed antigravity-0.1
```

```
$ python -c 'import antigravity'
```

What did this do?

- Fetch package from PyPI
- Install it in site-packages

What did this do?

- Find package via sys.path
- Import it
- Try it out!

CONDA INSTALL NUMPY SCIPY ASTROPY

- Conda is a binary package manager: <http://conda.pydata.org/docs/>
- Install any software (not just Python)
 - on Linux, Mac, Win, terminal or graphical installer, in \$HOME
- Anaconda is a free distribution (bundle of packages) by a company:
www.continuum.io/downloads
- conda-forge.github.io is a new community-run project
- There's also an Astropy and Sherpa channel on anaconda.org.
- Anaconda & Conda-Forge will maybe merge.
pip and conda will most likely never merge.
- If you want to learn more, there's [conda-myths-and-misconceptions](#) by Jake van der Plas and a recent podcast on conda ([link](#))

TUTORIAL RESOURCES (GENERAL)

- There's a ton of other good free resources available:
 - 100 numpy exercises
 - <http://www.scipy-lectures.org/>
 - A pytest tutorial and Sphinx tutorial
 - <http://www.astropy.org/astropy-tutorials/>
<http://astropy.readthedocs.io/> (the “getting started” sections)
 - <http://python4astronomers.github.io/>
 - Really 1000s of more good free tutorials and 100s of books.
The problem is how to pick and not waste time.

TUTORIAL RESOURCES (GENERAL)

- My suggestion would be this:
- If you're new to Python,
Numpy, Scipy, Pandas, Matplotlib
and don't know where to start ...
- Start here:
 - <https://github.com/jakevdp/WhirlwindTourOfPython>
 - <https://github.com/jakevdp/PythonDataScienceHandbook>
- Should we all start using this for teaching / tutorials?
(don't waste time developing materials, common reference)



PYVIDEO

The screenshot shows a web browser displaying a PyVideo.org page. The title of the video is "Modules and Packages: Live and Let Die!". It was recorded on Friday, April 10, 2015, by David Beazley. The video player shows a man speaking at a podium, with a terminal window displaying Python code and its output. The PyVideo.org navigation bar includes links for Start, Events, Tags, Speakers, About, and Contribute Media.

Description

All Python programmers use the import statement, but do you really know how it works and what it allows? This tutorial aims to take a deep dive into every diabolical issue related to modules, packages, and imports. When we're done, you'll finally be ready to unleash your million line micro framework on the world!

Details

- <http://pyvideo.org/>
- Links to videos (all freely available on YouTube) from 6000+ presentations from 140 events.
- Many PyCon, PyData, ... conferences do high-quality recordings.
- Example: David Beazley explaining how packages & modules & import work in Python ([link](#))
- Overall not the best way to learn stuff, more if you want an overview of what's going on ...

SAFARIBOOKSONLINE

- For MPG members, 40,000 books from Safari freely available <https://aleph.mpg.de> (web viewer isn't nice, but it's there)
- I've been using <https://www.safaribooksonline.com/> for a few months to learn via their video courses, tutorials and books.
- It's expensive (400 Euro/year). There's a 10 day free trial, so you could just use that and check it out, and do a course or two.
- I liked Beazley's Python course, and started with the data science one:

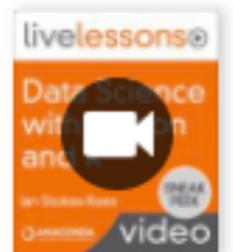


Python Programming Language

by **David Beazley**

Publisher: Addison-Wesley Professional

Published: August 2016



Data Science with Python and R (Anaconda Video Series)

by **Ian Stokes-Rees**

Publisher: Addison-Wesley Professional

Published: January 2017

TUTORIAL RESOURCES (GAMMA-RAY ASTRONOMY)

- Resources produced for tutorials / workshops:
 - <http://python4mpik.readthedocs.io/>
(half-day workshop at MPIK from 2013, possibly a bit outdated)
 - <http://gammipy.github.io/PyGamma15>
(full-week workshop at MPIK from 2015, with many talks and tutorials, all materials available)
- For Gammipy:
 - Some tutorial-style documentation is here: <http://nbviewer.jupyter.org/github/gammipy/gammipy-extra/blob/master/index.ipynb>
 - The documentation isn't in good shape. Needs a lot of work and your help!
 - Other examples are scattered, mostly in slides and HESS Confluence pages.
- For ctapipe I don't know about tutorials.
Docs are here: <https://cta-observatory.github.io/ctapipe/>

GAMMAPY DOCUMENTATION

Instrument response function (IRF) functionality (`gammapy.irf`)

Introduction

`gammapy.irf` handles instrument response functions (IRFs):

- Effective area (AEFF)
- Energy dispersion (EDISP)
- Point spread function (PSF)

Most of the formats defined at [IACT IRFs](#) are supported. Otherwise, at the moment, there is very little support for Fermi-LAT or other instruments.

Most users will not use `gammapy.irf` directly, but will instead use IRFs as part of their spectrum, image or cube analysis to compute exposure and effective EDISP and PSF for a given dataset.

Getting Started [¶](#)

TODO: document

Using `gammapy.irf`

If you'd like to learn more about using `gammapy.irf`, read the following sub-pages:

- [IRF Theory](#)
- [Effective area](#)
- [Energy dispersion](#)
- [Point spread function](#)

Reference/API

- Every sub-package has a Sphinx docs page with useful content.
- Should be structured in a uniform way (like in Astropy)
- This is **not** the case at the moment.
- Need to distribute the work or find someone that takes a week or two full-time to write docs.
- A small pull request to improve the docs is a great way to start contributing to Gammapy and learn git / Github / review workflow.

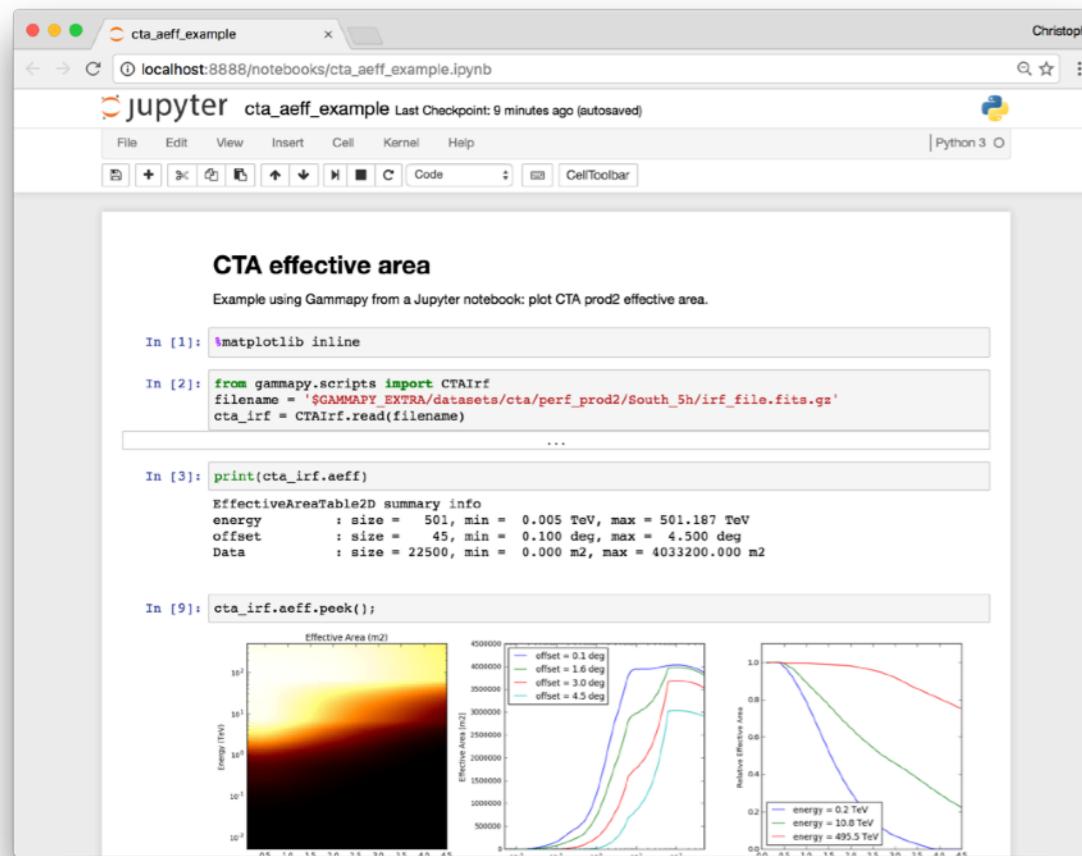
GAMMAPY DOCUMENTATION

Notebooks

Getting started guides

To get started with gamma-ray data analysis:

- [IACT DL3 data with Gammapy](#) (H.E.S.S. data example)
- [Image analysis with Gammapy](#) (H.E.S.S. data example)
- [Spectral analysis with Gammapy](#) (H.E.S.S. data example)
- [Source detection with Gammapy](#) (Fermi-LAT data example)
- [Spectrum simulation and fitting](#) (CTA example)

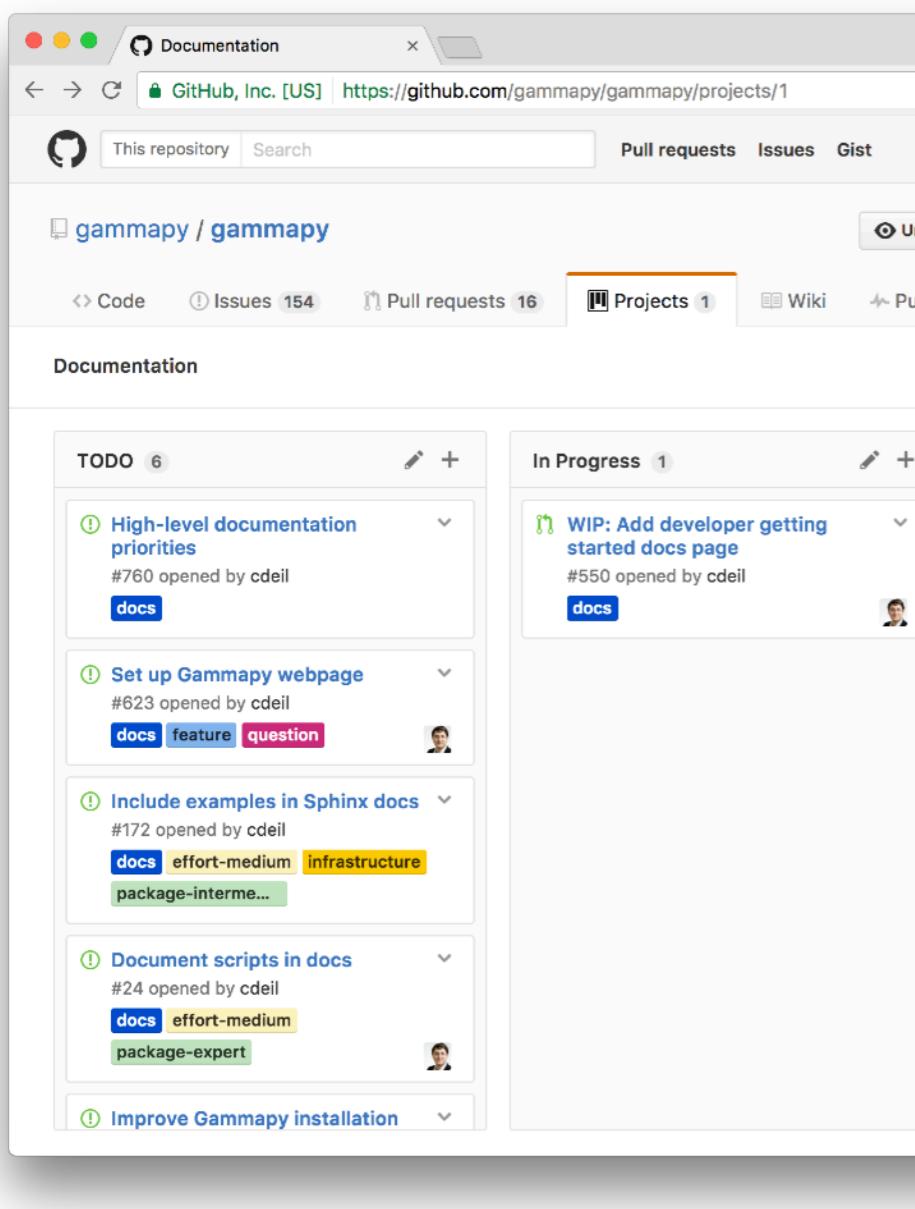


- In addition to the Sphinx docs, we have IPython notebooks.
- Again, notebooks should have a uniform structure, with intro at the top and exercises at the end.
- Existing notebooks need work, and we need more. For now, start using them or author one.
- Very soon we need to make a concerted effort to improve the Gammapy docs.

GAMMAPY COMMUNICATION

- Mostly asynchronous: Git & Github & Mailing list & Slack
- Please sign up to:
 - <https://groups.google.com/forum/#!forum/gammapy>
 - <https://lists.nasa.gov/mailman/listinfo/open-gamma-ray-astro>
- Use private <https://gammapy.slack.com> for help chat (git, Github, errors, questions). Email me if you want access.
- Should do face-to-face meetings and regular telcons.
- Sub-tasks / roles / responsibilities / project management to be discussed at a follow-up call or the next f2f meeting soon.

GAMMAPY GITHUB

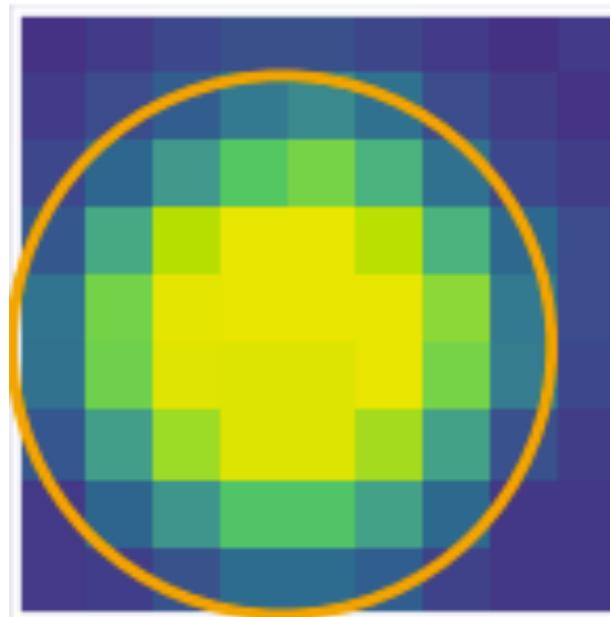
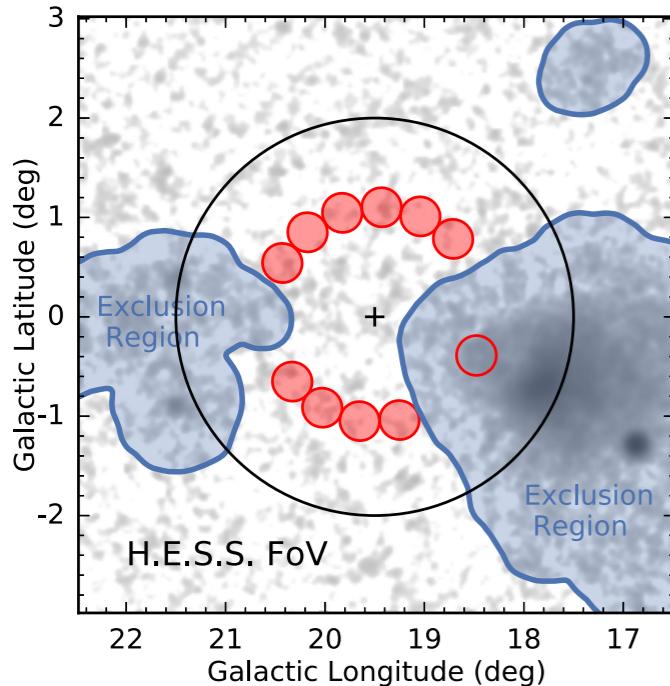


- Git and Github is great
 - Good contribution model (pull request & tests & code review)
 - Scales to many simultaneous activities
 - Gammapy has ~ 100 open issues and usually a few in-progress pull requests
- Git and Github has so far not been good at picture project management.
- Now there's the “projects” feature, maybe we should try it?
- To check it out, I've started one for “documentation”.
- Need help with this “planning” and “management” aspect.

OTHER PROJECTS

- CTA data challenge
 - Started, will be a major effort in the coming months
- H.E.S.S. public test data release
- <https://github.com/gammapy/gamma-cat>
- Next slides have some info on other related projects you might not be aware of ...

ASTROPY.REGION



-
- astropy-regions.readthedocs.io
- Sky & pixel regions, bounding box, mask, ds9 region files, matplotlib plotting, ...
- Developed by Johannes and me, with other Astropy developers
- Will make v0.2 release very soon. Plan to “finish” it and merge in Astropy core in ~ April.
- For now available as a separate package. Please try it out and give feedback!
- Already used in Gammapy, for now an extra dependency. Need to clean up old region-based code in Gammapy (circle & box & mask).

PINT

- <https://github.com/nanograv/PINT>
- “PINT is not Tempo3”
 - a new code for high-precision pulsar timing
- Built on astropy.coordinates (ERFA)
and astropy.time.Time (two float64)
- Developed by Scott Ransom, Paul Ray, Paul Demorest,
Anne Archibald, Luo Jing, and other pulsar timing experts.
- Python 3 almost working in master.
Some setup / packaging work to do.
Simple pip or conda install pint coming soon.
- Paper with extensive validation against TEMPO and TEMPO2 this summer.
- Arache, Marion, Roberta — PINT / Gammapy for H.E.S.S. pulsar analysis.



MODELING / FITTING



- Modeling / fitting is what many (most?) users want: a likelihood fit.
- Deciding on a framework has been and still is hard for Gammapy
- `astropy.modeling`, Sherpa
<https://saba.readthedocs.io>
- Also <http://astromodels.readthedocs.io/> by Giacomo Vianello (used by 3ML) and Fermi ST / Fermipy an Naima has something.
- Recently started `gammapy.utils.modeling` to support XML serialisation (needed for CTA data challenge and liked by some).
- I see modeling and DL3 data model / IRFs as the largest areas where a lot of development in Gammapy is needed in 2017.

PLANS

- Gammapy 0.6 and 1.0 release when?
- Gammapy paper based on 0.6 in ~ April?
<https://github.com/gammapy/gammapy-paper>
- Meeting at MPIK March 20 - 24, 2017
<https://github.com/open-gamma-ray-astro/2017-03-ogra-meeting>
- Need other meetings (user / dev on Gammapy) and presence at HESS / CTA meetings.

DISCUSSION

- Any comments or question?
Please speak up!
- Meeting(s) in February or March?
How long? MPIK or Paris? Length / focus?
- Monthly Gammipy telcons?
- Need to start some structured project management
(milestones, roles, commitments).
Discuss in next call?