



A Python package for
gamma-ray astronomy

Versioned executable user documentation for in-development science tools



Catherine Boisson¹, José Enrique Ruiz², Christoph Deil³, Axel Donath³ and Bruno Khelifi⁴ for the Gammapy team.

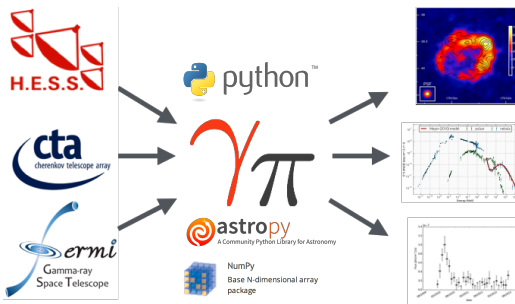
¹ LUTH - Observatoire de Paris, ² Instituto de Astrofísica de Andalucía - CSIC, ³ Max-Planck-Institut für Kernphysik, ⁴ APC - AstroParticule et Cosmologie, Université Paris Diderot.

Abstract

One key aspect of software development is **feedback from users**. This community is not always aware of the developments undertaken in the code base, neither they use the tools and practices followed by the developers to deal with a non-stable software in continuous evolution. **Gammapy provides its beta-tester user community with versioned reproducible environments and executable documentation, in the form of tutorials that are versioned coupled with the code base.** We believe this set-up greatly improves the user experience for a software in prototyping phase, as well as provides a good workflow for developers to deliver versioned and up-to-date documentation.

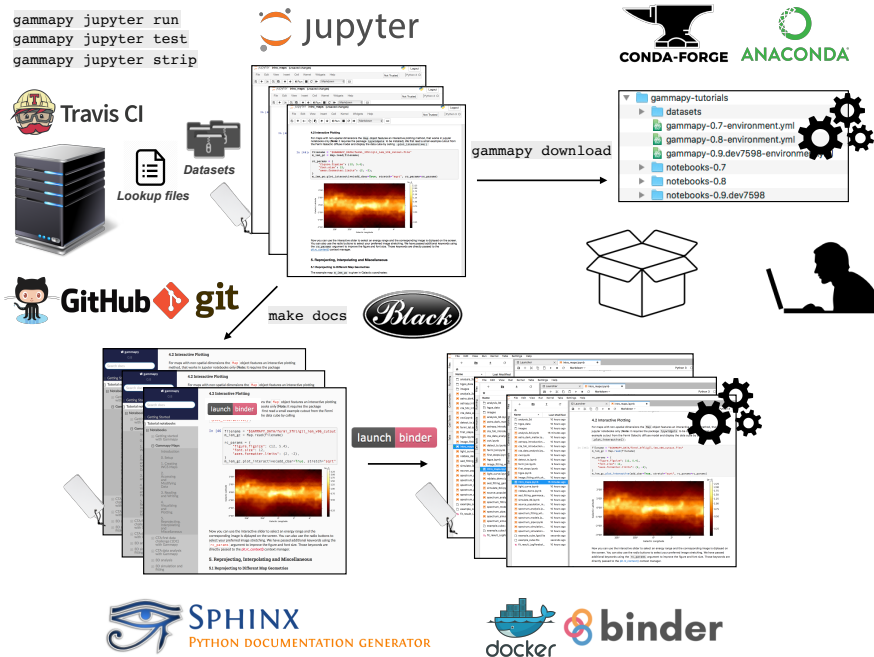
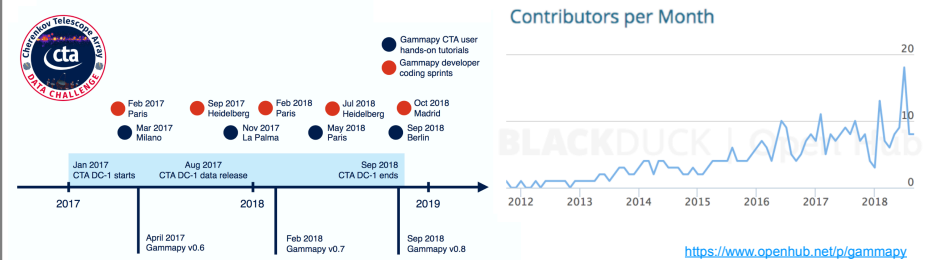
What is Gammapy?

A Python package for gamma-ray astronomy.
A prototype for the Cherenkov Telescope Array
(CTA) science tools.



A software in development

The rise in the contributor and user base together with a high development activity, hinders user feedback from not always up-to-date versions. There's a need for reproducible environments versioned coupled with the code base.



Set-up

- Tutorials integration:** tutorials in the form of *Jupyter* notebooks are integrated into the software documentation with *Sphinx* + *nbsphinx* extension.
- Executable on-line:** the documentation provides links to *myBinder* platform, where tutorials can be executed in the cloud using a versioned kernel provided by a *Dockerfile*.
- Version-coupling:** the base-code, the tutorials and the *Dockerfile* are stored in the same *Github* versioned repository.
- Authoring and review:** seamless code review for the tutorials with `gammapy jupyter` using diff comparisons in pull requests is possible, since the notebooks only store markdown and code cells with no outputs.
- Regression tests:** tutorials execute in *Travis Continuous Integration* system, checking that their output cells do not throw any errors.
- Reproducibility:** deterministic environments are defined for each version of the software in the form of *conda* configuration files, with pinned version numbers for each dependency package.
- Shipping:** `gammapy download` command allows to retrieve versioned tutorials, composed of *Jupyter* notebooks, the datasets needed and the *conda* configuration file to build the environment.
- Maintainability:** for each versioned environment we define its requirements, which tutorials to provide and where to fetch them with centralized index lookup files.

<https://github.com/gammapy/gammapy>

<https://docs.gammapy.org>

<https://gammapy.org>

Command Line Tools

`gammapy download` provides **users** with the means to retrieve any tutorials-related asset for a specific version, whereas `gammapy jupyter` provides **developers** with a tool to work with notebooks in a seamless workflow for the development/review/publish process of the versioned executable tutorials.

```
$ gammapy download tutorials --release 0.8
INFO:gammapy.scripts.downloadclass:Content will be downloaded in gammapy-tutorials/notebooks-0.8
Downloading files [=====] 100%
INFO:gammapy.scripts.downloadclass:Content will be downloaded in gammapy-tutorials/datasets
Downloading files [=====] 100%

**** Enter the following commands below to get started with Gammapy
cd gammapy-tutorials
conda env create -f gammapy-0.8-environment.yml
conda activate gammapy-0.8
export GAMMAPY_DATA=/Users/jer/Desktop/gammapy-tutorials/datasets
jupyter lab
```