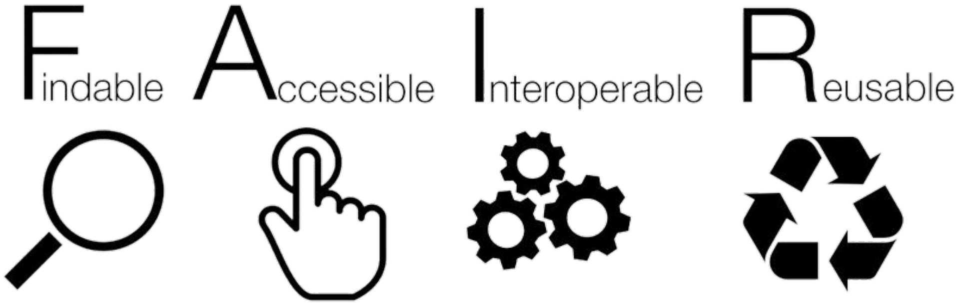
Gammapy: a python package for (not only) gamma-ray astronomy

Claudio Galelli - LUTh, Observatoire de Paris

20/06/2024, 13th CRIS-MAC, Trapani

In the age of open data it is of maximum importance to be



Data and metadata should be **easy to find for humans and computers**

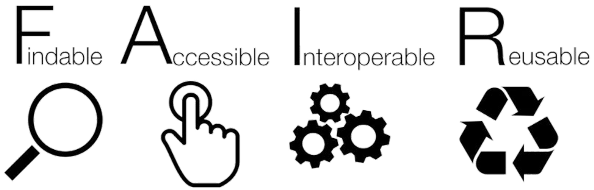
Once the data is found, it should be clear how it is **accessed, including authentication**

The data should be easy to **integrate with other data** and analysis and storage workflows

The **reuse of data** should be **optimized** through data and metadata description

M. Wilkinson et al., *The FAIR Guiding Principles for scientific data management and stewardship*, Scientific Data, 2016
M. Servillat et al., *FAIR high level data for Cherenkov astronomy*, Proceedings of the XXXI ADASS conference, Cape Town, 2021

Open science is not only open data



To achieve true interoperability and reusability of science products, the **analysis software must also follow the same philosophy** as the formats

Article | Open access | Published: 14 October 2022

Introducing the FAIR Principles for research software

Michelle Barker, Neil P. Chue Hong, Daniel S. Katz, Anna-Lena Lamprecht, Carlos Martinez-Ortiz, Fotis Psomopoulos, Jennifer Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez & Tom Honeyman

Scientific Data **9**, Article number: 622 (2022) | Cite this article

17k Accesses | **48** Citations | **231** Altmetric | Metrics

Abstract

Research software is a fundamental and vital part of research, yet significant challenges to discoverability, productivity, quality, reproducibility, and sustainability exist. Improving the practice of scholarship is a common goal of the open science, open source, and FAIR (Findable, Accessible, Interoperable and Reusable) communities and research software is now being understood as a type of digital object to which FAIR should be applied. This emergence reflects a maturation of the research community to better understand the crucial role of FAIR research software in maximising research value. The FAIR for Research Software (FAIR4RS) Working Group has adapted the FAIR Guiding Principles to create the FAIR Principles for Research Software (FAIR4RS Principles). The contents and context of the FAIR4RS Principles are summarised here to provide the basis for discussion of their adoption. Examples of implementation by organisations are provided to share information on how to maximise the value of research outputs, and to encourage others to amplify the importance and impact of this work.

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

The core library for the CTA scientific analysis tools!

But it's used (or tested) by many other experiments!

An astropy associate package!

Core dependencies

matplotlib

astropy

SciPy NumPy

A. Donath et al., *Gammapy: A Python package for gamma-ray astronomy*, 2023, A&A, 678, A157

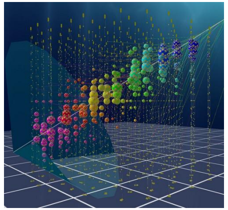
- v1.0 release (Long Term Stable) 2022 Nov 10th
- v1.2 feature release 2024 February 29th
- v1.3 next feature release planned for October 2024

Core data format: **GADF** → A big push towards VO-compliant FAIR data format: **VODF**

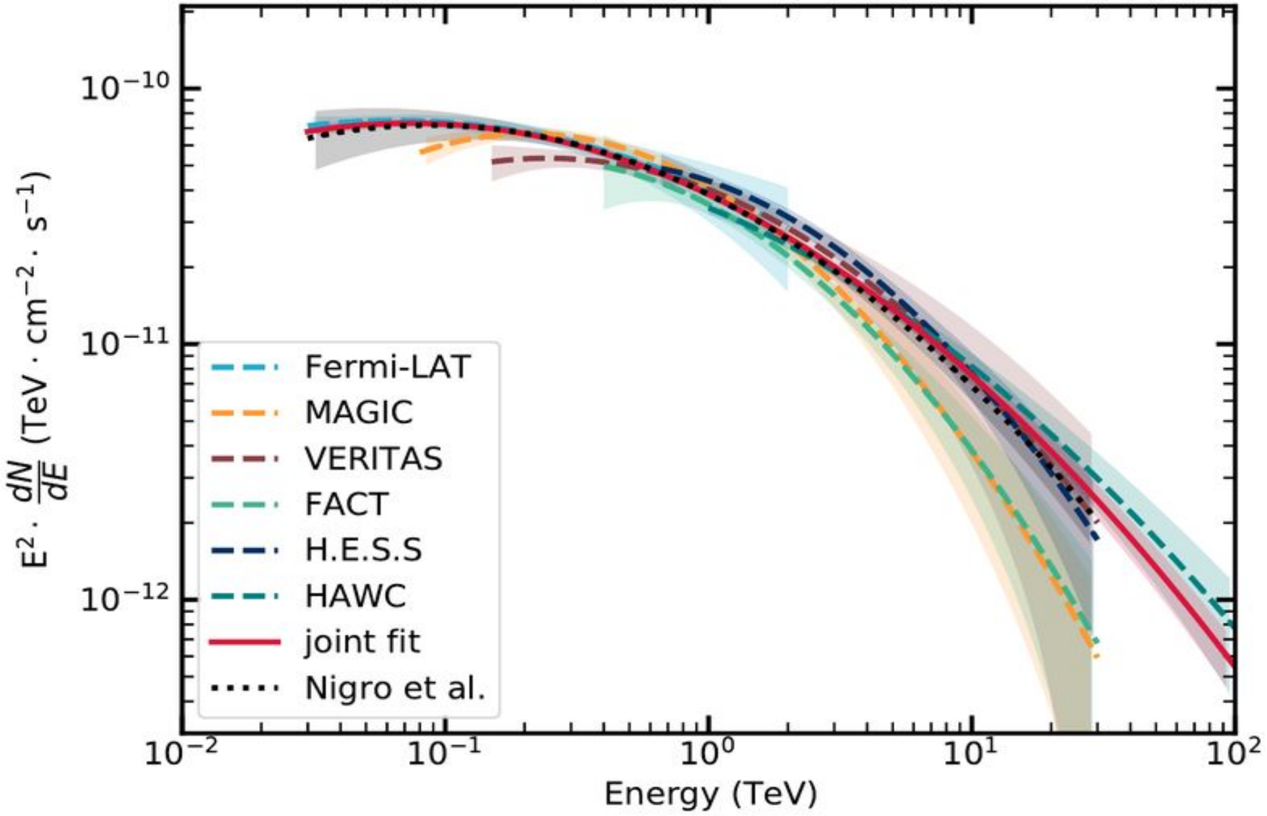
Pointing Gamma-Ray instruments

Slewing Gamma-Ray instruments

Neutrino instruments



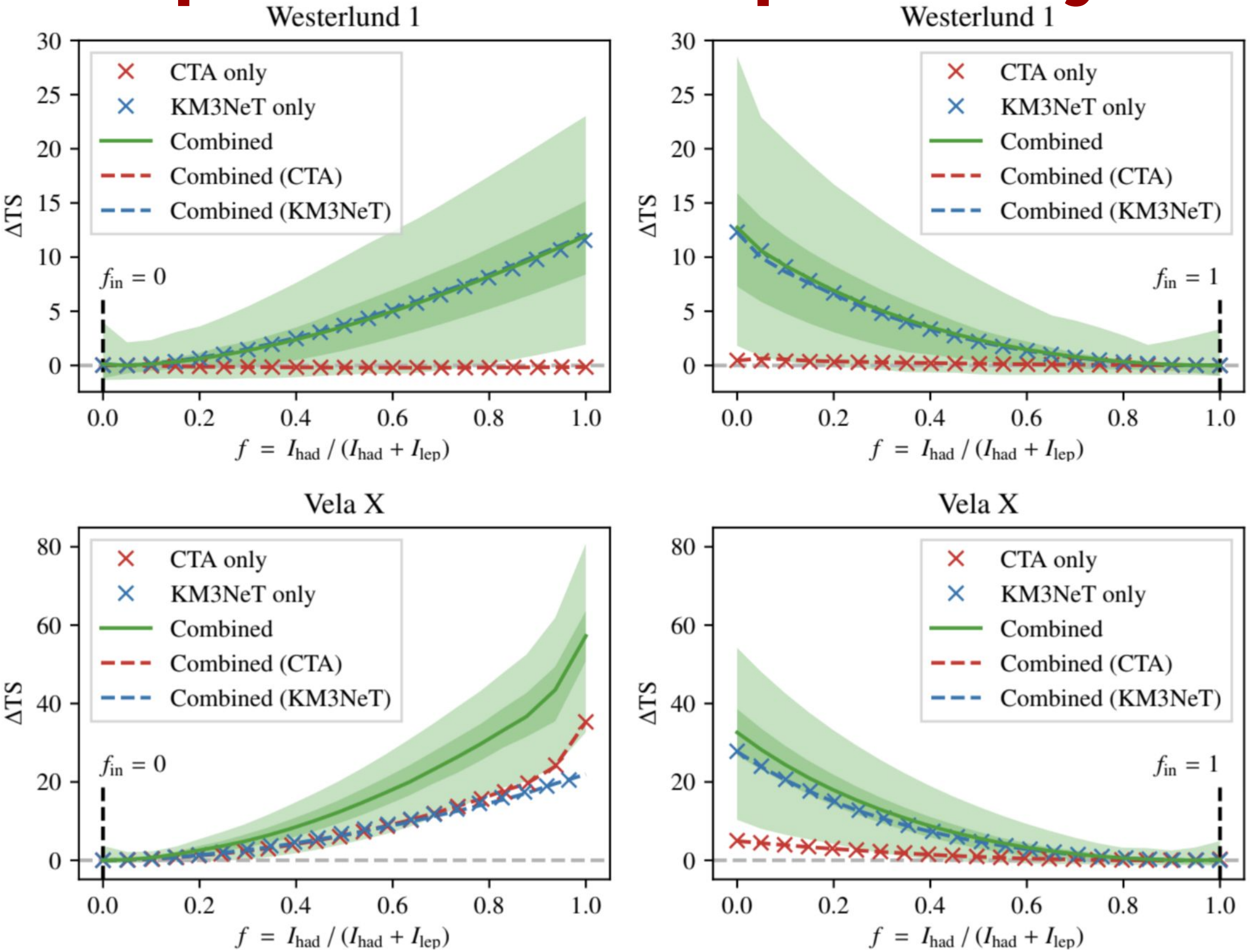
The power of interoperability with γπ A **Python** package for **gamma-ray** astronomy



An example of the power of interoperability: the crab nebula joint fit using Gammapy. From: A. Albert et al., *Validation of standardized data formats and tools for ground-level particle based gamma-ray observatories*, 2022, A&A 667

The power of interoperability with γπ

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

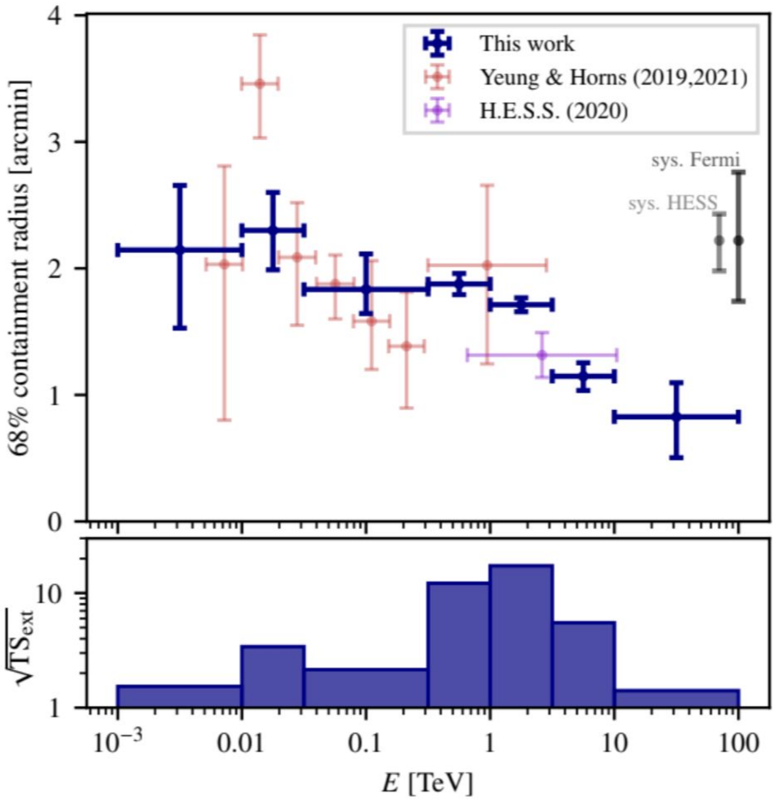


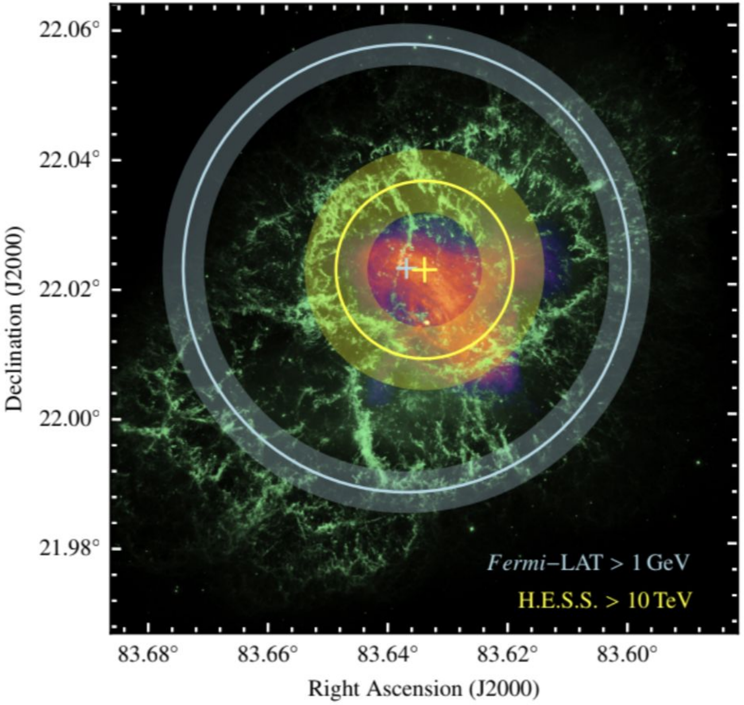
Validation of gammapy in joint analyses with CTA and KM3Net: fit of simulated leptonic and hadronic emission scenarios from different sources (Vela X SNR/Westerlund cluster)

From T. Unbehaun, et al., *Prospects for combined analyses of hadronic emission from γ-ray sources in the Milky Way with CTA and KM3NeT*, Eur. Phys. J. C, 2024

The power of interoperability with γπ

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





Extension of the Crab nebula as seen by Fermi and H.E.S.S. joint fit using Gammapy. From: F. Aharonian et al., *Spectrum and extension of the inverse-Compton emission of the Crab Nebula from a combined Fermi-LAT and H.E.S.S. analysis*, A&A 2024

Have a look!

https://gammapy.org/

Gammapy | About | News | CTA | Contact | Team | Contribute | Documentation | Code of Conduct | Acknowledging

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

Gammapy is an open-source Python package for gamma-ray astronomy built on Numpy, Scipy and Astropy.
It is used as core library for the Science Analysis tools of the Cherenkov Telescope Array (CTA), recommended by the H.E.S.S. collaboration to be used for Science publications, and is already widely used in the analysis of existing gamma-ray instruments, such as MAGIC, VERITAS and HAWC.

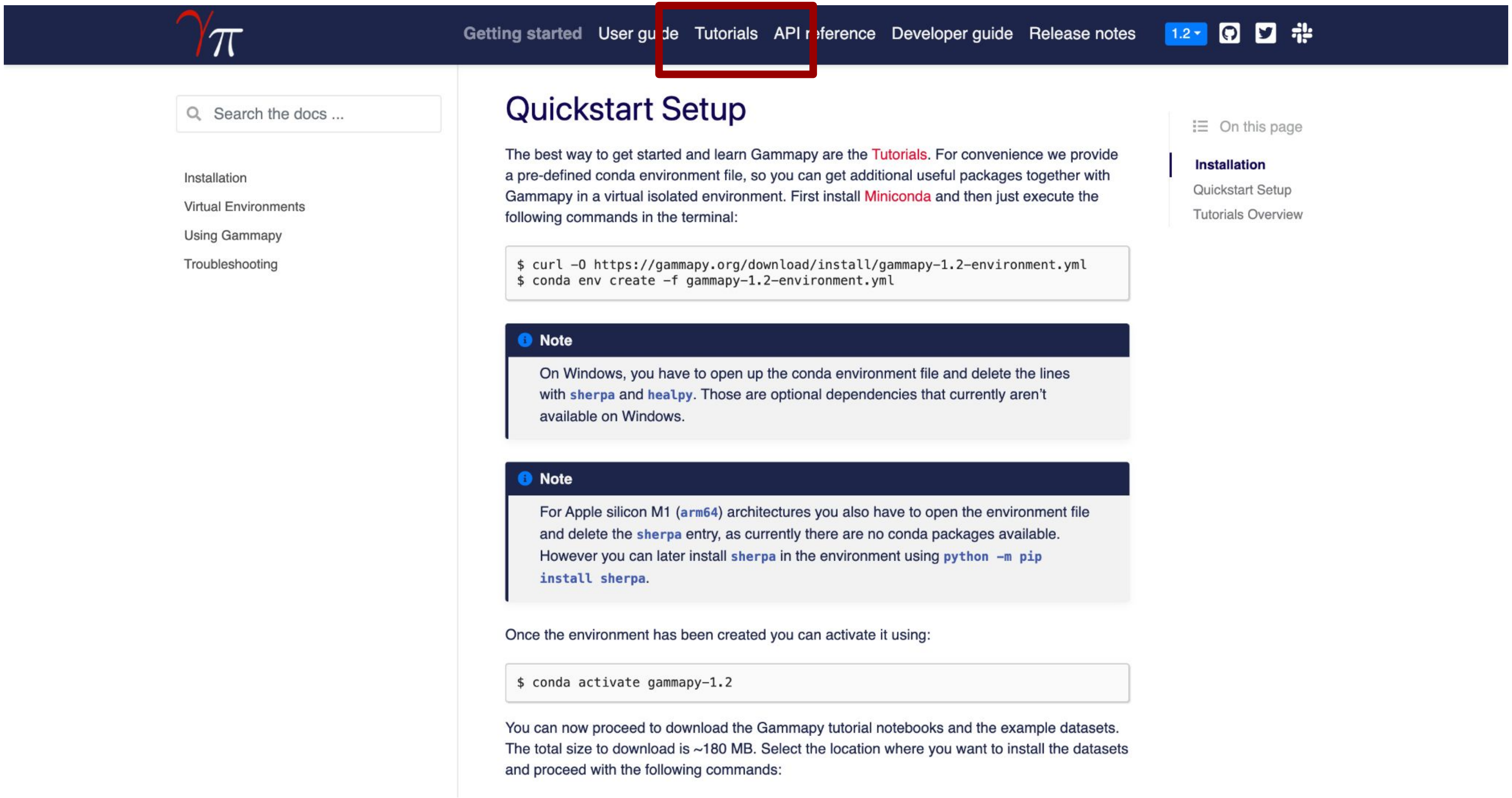
News:

Feb. 29th, 2024:	Minor version release v1.2
Dec. 6th, 2023:	Bug fix release v1.0.2
Oct. 23rd, 2023:	Publication of the first Gammapy paper in A&A (accepted: 7th July 2023) as highlighted paper (link)

Please remember to acknowledge and cite the use of Gammapy!

Get it now!

https://docs.gammapy.org/1.2/ → https://docs.gammapy.org/1.2/getting-started/index.html



Getting started User guide Tutorials API reference Developer guide Release notes 1.2

Search the docs ...

Installation
Virtual Environments
Using Gammapy
Troubleshooting

Quickstart Setup

The best way to get started and learn Gammapy are the Tutorials. For convenience we provide a pre-defined conda environment file, so you can get additional useful packages together with Gammapy in a virtual isolated environment. First install Miniconda and then just execute the following commands in the terminal:

```
$ curl -O https://gammapy.org/download/install/gammapy-1.2-environment.yml
$ conda env create -f gammapy-1.2-environment.yml
```

Note

On Windows, you have to open up the conda environment file and delete the lines with `sherpa` and `healpy`. Those are optional dependencies that currently aren't available on Windows.

Note

For Apple silicon M1 (`arm64`) architectures you also have to open the environment file and delete the `sherpa` entry, as currently there are no conda packages available. However you can later install `sherpa` in the environment using `python -m pip install sherpa`.

Once the environment has been created you can activate it using:

```
$ conda activate gammapy-1.2
```

You can now proceed to download the Gammapy tutorial notebooks and the example datasets. The total size to download is ~180 MB. Select the location where you want to install the datasets and proceed with the following commands:

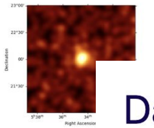
On this page
Installation
Quickstart Setup
Tutorials Overview

Tutorial gallery

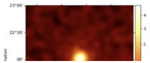
Introduction

The following three tutorials show different ways of how to use Gammapy to perform a complete data analysis, from data selection to data reduction and finally modeling and fitting.

The first tutorial is an overview on how to perform a standard analysis workflow using the high level interface in a configuration-driven approach, whilst the second deals with the same use-case using the low level API and showing what is happening *under-the-hood*. The third tutorial shows a glimpse of how to handle different basic data structures like event lists, source catalogs, sky maps, spectral models and flux points tables.

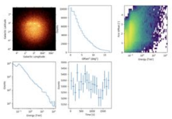


High level ir

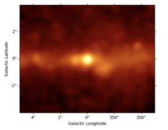


Data exploration

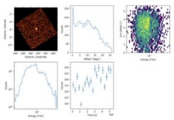
These tutorials show how to perform data exploration with Gammapy, providing an introduction to the CTA, HAWC, H.E.S.S. and Fermi-LAT data and instrument response functions (IRFs). You will be able to explore and filter event lists according to different criteria, as well as to get a quick look of the multidimensional IRFs files.



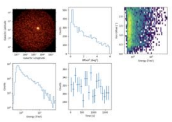
CTA with Gammapy



Fermi-LAT with Gammapy

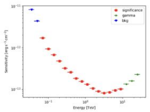


HAWC with Gammapy

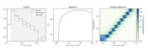


H.E.S.S. with Gammapy

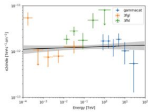
1D Spectral



Point source sensitivity

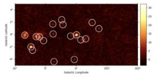


Spectral analysis of extended sources

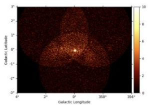


Flux point fitting

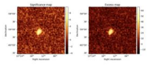
2D Image



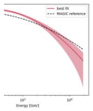
Source detection and significance maps



2D map fitting

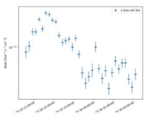


Ring background map

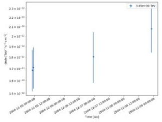


nalysis with
pendent
cuts

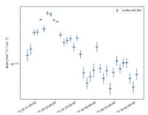
Time



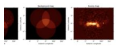
Estimation of time variability in a lightcurve



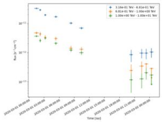
Light curves



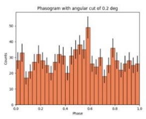
Light curves for flares



nage exploration
ng



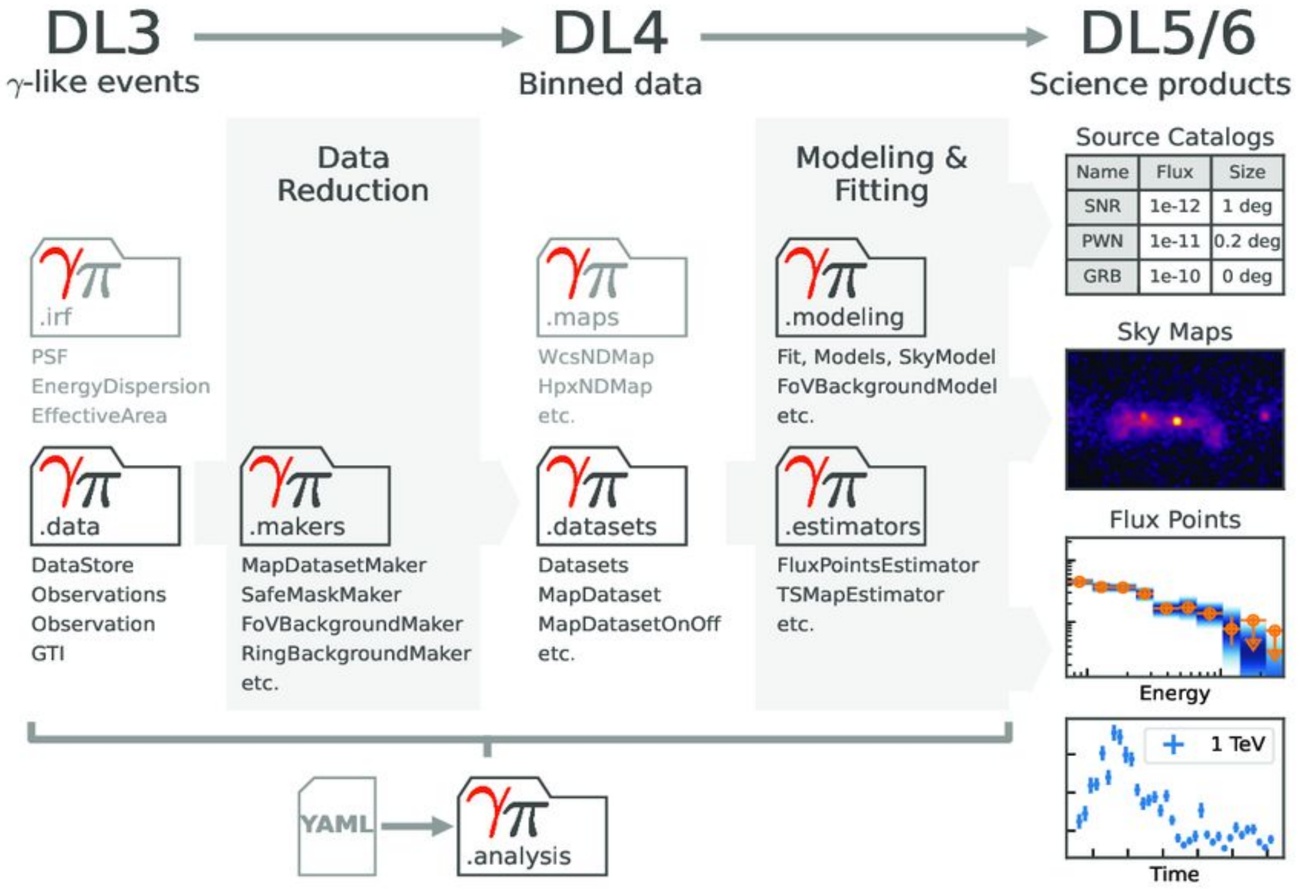
Simulating and fitting a time varying source



Pulsar analysis

https://docs.gammapy.org/1.2/tutorials/index.html

Data levels and code structure



Findability
Accessiblity

Data should be findable and accessible at any level by using of VO compliant formats and Metadata

Interoperability

A common and open analysis software and format, with very general tools in the steps between data levels, leads to more shared analyses

Reusability

Good documentation and metadata associated to each DL and analysis step ensures reusability of data products

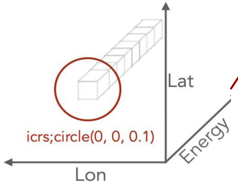
The `gammapy.analysis` workflow: step 1 - data reduction

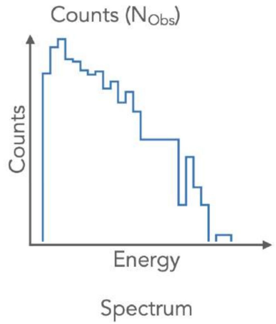
DL3 → Apply event selections (time, offset, etc)
S
patial and energy binning
→ **DL4**

EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
5407363825684	123890826.66805482	84.97964	23.89347	10.352011
5407363825695	123890826.69749284	84.54751	21.004095	4.0246882
5407363825831	123890827.23673964	85.39696	19.41868	2.2048872
5407363825970	123890827.79615426	81.93147	20.79867	0.69548655
5407363826067	123890828.26131463	85.98302	21.053099	0.86911184
5407363826095	123890828.41393518	86.97305	21.837437	4.1240892
5407363826128	123890828.52555823	83.40073	19.771587	1.6680022
5407363826168	123890828.6829524	82.25036	19.22003	4.7649446
5407363826383	123890829.53362775	83.18322	22.008213	0.7920148
...	...	...	...	...

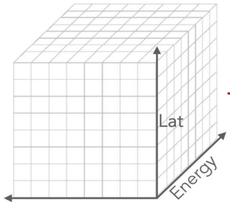
Bin events (and IRFs) into n-dim sky maps

Spectral analysis: Cube with one spatial bin





Generalised case: 3D map



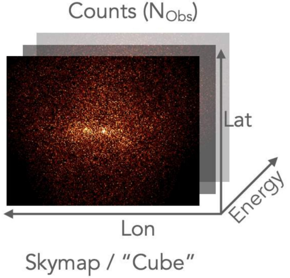
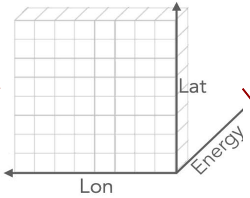
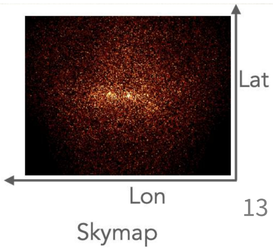


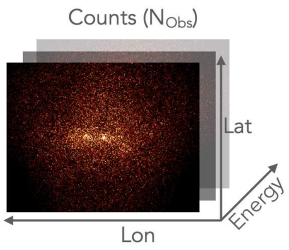
Image analysis: cube with one energy bin





The `gammapy.analysis` workflow: step 2 - data modeling and fitting

DL4 → **Datasets**



Skymap / “Cube”

From gammapy data reduction

or

pre-computed

↓

Fermi, MAGIC, HESS, HAWC, OGIP files…
Also all together!

Models

Spectral
Spatial
Temporal

Fit (can be joint!)

Forward-folding
and
Log-likelihood fitting

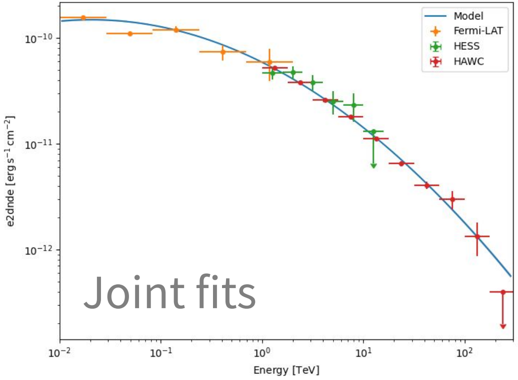
$$N_{Pred}(p, E) = N_{bkg}(p, E) + \Sigma_{src} N_{src}(p, E)$$

Cash statistic:
known background

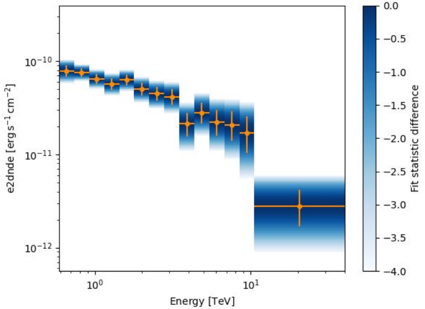
Wstat statistic:
measured background in Off-region

The `gammapy.analysis` workflow: the final products

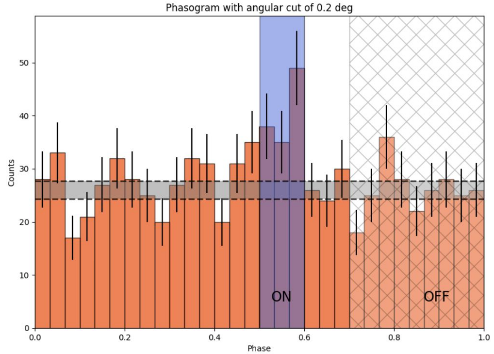
DL5/6



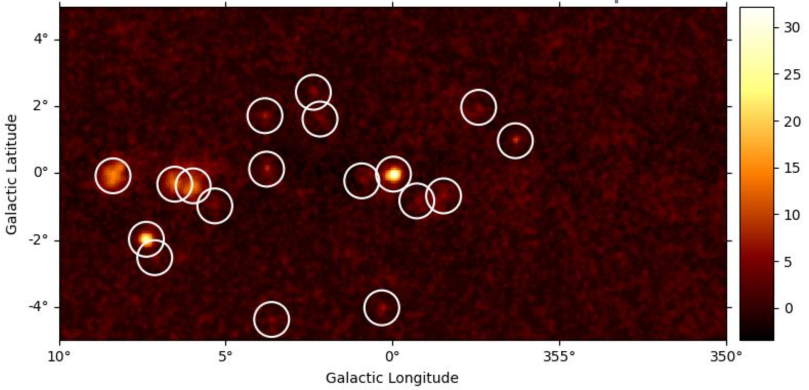
TS profiles

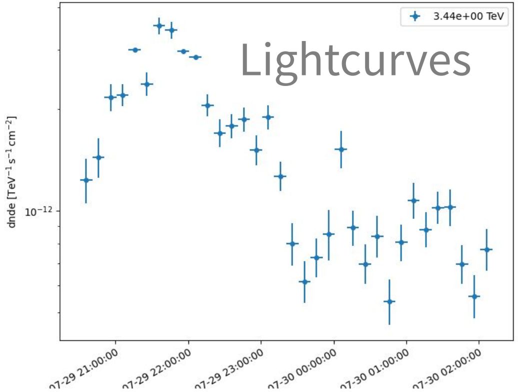


Periodic sources



Source maps





And more:

Custom catalogs
Dark Matter
Fits on DL5

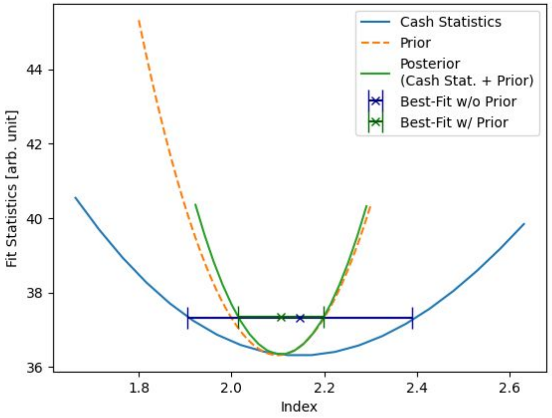
A selection of new features

From gammapy 1.2

Priors

Possibility of adding priors to the probability distribution to add known information about the dataset or parameters of the fit

K. Streil



Metadata

Metadata containers for most of the DL3 to DL5 structures - for provenance information

B. Khelifi, R. Terrier, C. Galelli, K. Feijen

New catalogs

- 1LHAASO catalog
- Update of 4FGL to include DR4

Q. Remy

Time variability

Set of utility functions to evaluate the variability in a lightcurve (especially for AGNs)

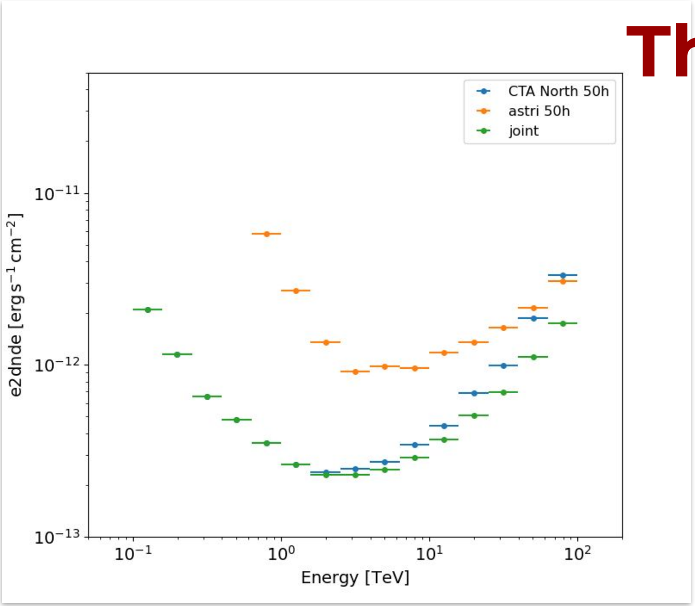
C. Galelli

ObsCore compatibility

Tool to convert information in DataStore to the VO-compliant ObsCore

P. Kornecki

The future



Improve sensitivity computations - especially for joint sensitivity analyses

Integrate with VO-compliant inputs and formats (VODF)

Further develop **distributed computing prototypes** - currently exploring using different tensor libraries such as **Jax**, dask, PyTorch (https://github.com/gammapy/gammapy/pull/5302 https://github.com/gammapy/gammapy/pull/5318)

Aim is to improve efficient computing for large, joint datasets and prepare for machine learning applications

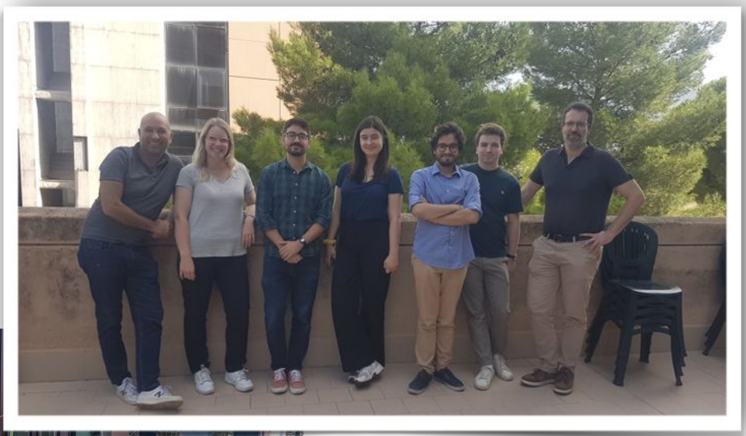
Summary

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

Gammapy has provided interoperable software for many instruments and more are always being added

- Next releases have many key objectives dedicated to **compliance to FAIR4RS and VO standards**
- Distributed computing efforts for future proofing analysis of huge combined archives
- Strong participation in the CTAO SDC

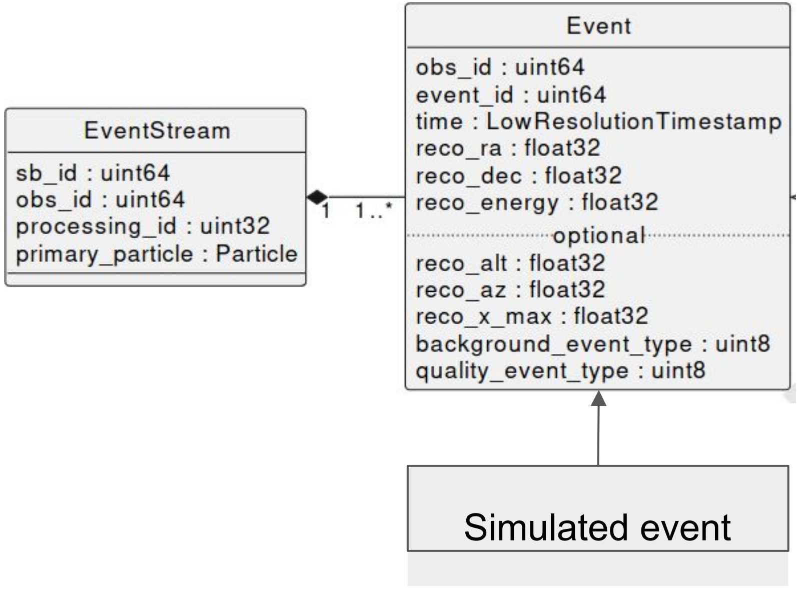
Try it, and if you need us ask for #help on slack! gammapy.slack.com



BACKUP

Basic data format: “Events” and “service data” (DL3)

event = particle detection (gamma, neutrino)



Information derived from simulation: Instrument Response Functions (IRFs)

- Stable Time Interval
- Effective Area
- Energy Dispersion
- Point Spread Function
- Background
- Radius of On region for point-like IRFs

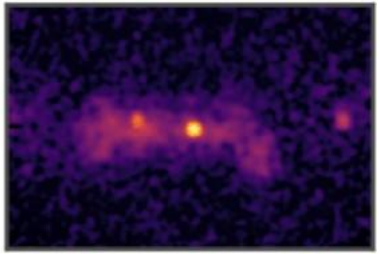
From CTA DL3 data model

Higher levels: Science results

DL4 (Science binned)

- exposure maps
- counts maps
- exclusion maps
- significance maps
- excess maps

Sky Maps



DL5/6 (Science products)

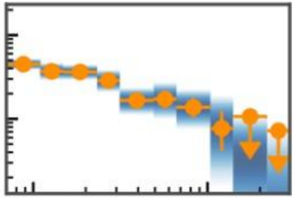
- Flux maps & fit models
  - data cube (3D,4D)
  - 2D sky map
  - light curve
  - spectrum
  - spatio-spectral cube
  - …

Potential future developments with VODF

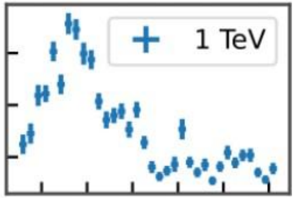
Source Catalogs

Name	Flux	Size
SNR	1e-12	1 deg
PWN	1e-11	0.2 deg
GRB	1e-10	0 deg

Flux Points



Energy



Time

Next steps & open questions

- Starting with a format definition: the one of CTAO, strongly inspired by GADF
    - Allow for multiple IRFs
    - Including different event categories (event types)
    - Choose metadata standards
- Ensuring interoperability (especially with IVOA)
    - Make data discoverable via VO (*ObsCore*)
    - Could contribute to an IVOA interest group if it happens: HEIG (link)
    - Current considerations
        - CTAO Data Model group, DM for High Energy astrophysics, IVOA DM
    - Further workshops in preparation (HEIG in Dec., CTAO data format in jan/feb)

Getting the software

- Quickstart installation with conda/mamba

```
curl -O https://gammapy.org/download/install/gammapy-1.1-environment.yml
conda env create -f gammapy-1.1-environment.yml
conda activate gammapy-1.1
```

See: Getting started

- Installation with pip

```
python -m pip install gammapy[all]
```

← install all dependencies

Note: if you install without using gammapy-1.1-environment.yml make sure to install pydantic<2.0 and matplotlib<3.8 for compatibility

Getting the software

- Download tutorials & associated data

```
gammapy download notebooks
gammapy download datasets
```

datasets are now versioned

If using conda environment, set GAMMAPY_DATA with conda

```
conda env config vars set GAMMAPY_DATA=$PWD/gammapy-datasets/1.1
conda activate gammapy-1.1
```

else set with shell:

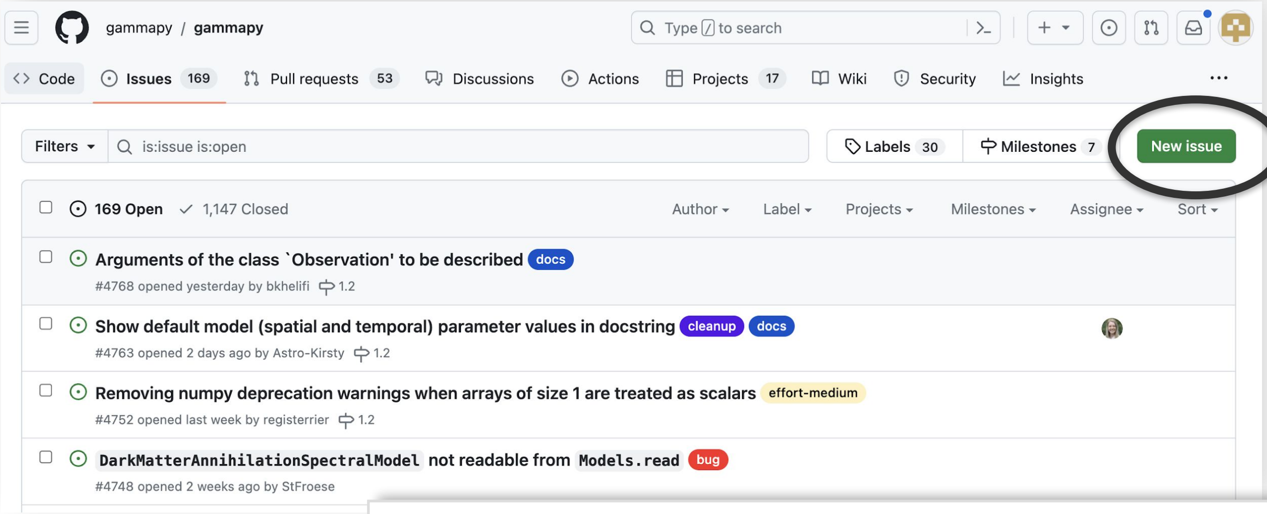
```
export GAMMAPY_DATA=$PWD/gammapy-datasets/1.1
```

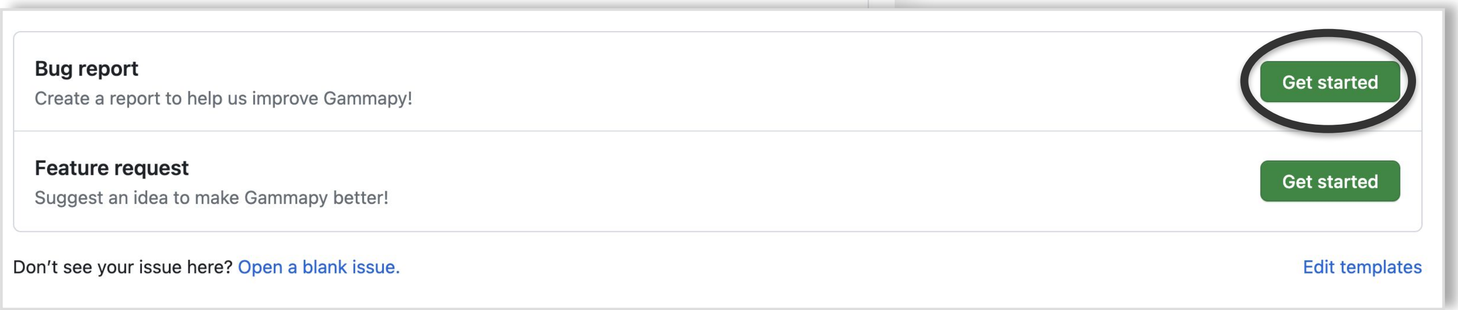
Getting help, reporting issues

- How to provide feedback / get help:
  - #help channel on gammapy.slack
  - #gammapy channel on hesschat.slack.com
  - GitHub discussions , in particular help category

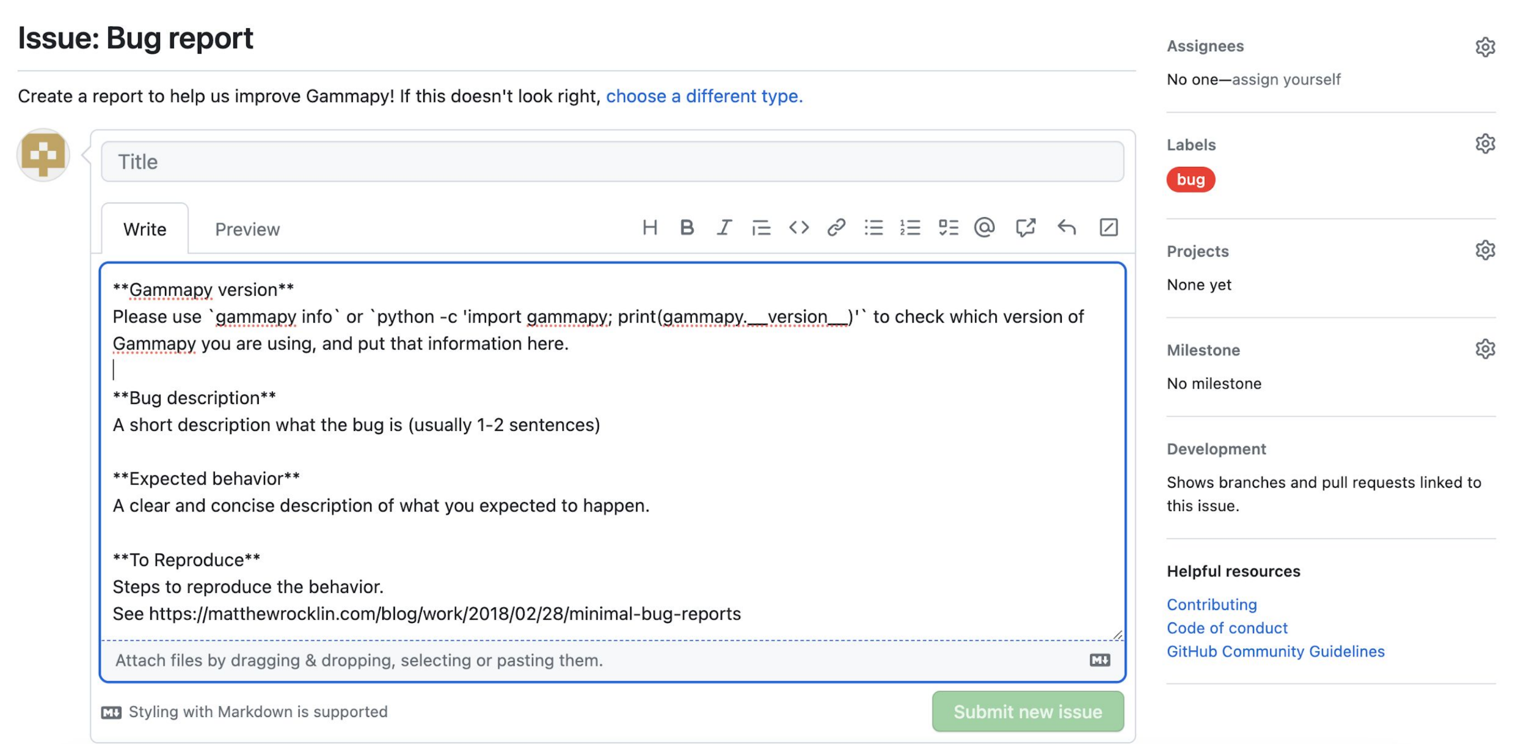
- How to report issues and bugs or request a new feature:
  - GitHub issues (requires creating an account on GitHub)

Reporting issues: on GitHub





Reporting issues: on GitHub



How to contribute?

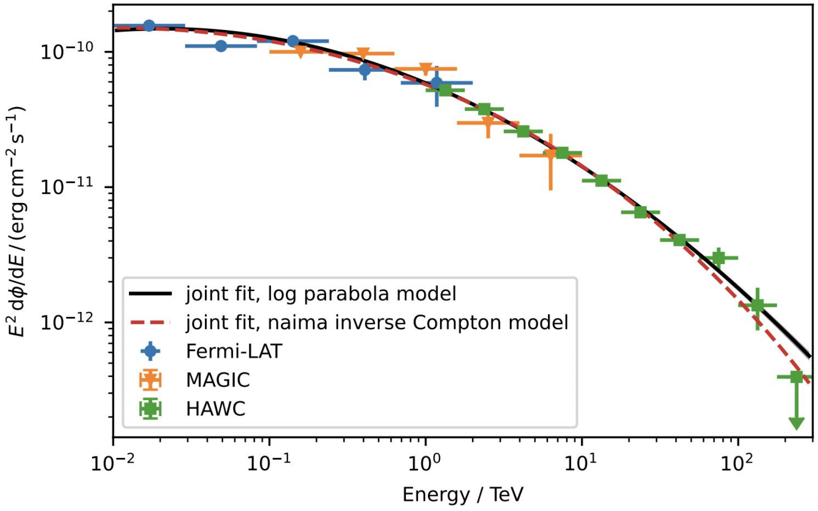
- Follow the developer guide to get set-up

- Get in touch with other developers early
  - #dev channel on gammapy slack
  - discuss contribution during dev calls (on Fridays 2pm)

- Open a Pull Request (PR) on GitHub gammapy repo

Gammapy v1.0 paper is published

- See: A. Donath, R. Terrier, Q. Remy, et al. [2023, A&A, 678, A157]
- Fully reproducible code and figures. See [GH repo] and [pdf file]



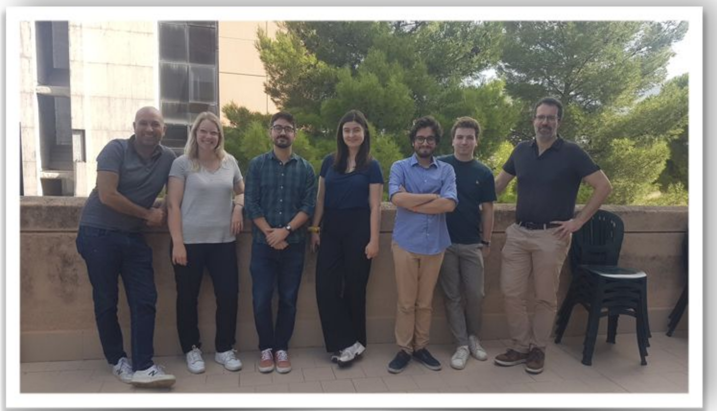
Multi-instrument joint fit of the Crab with a log parabola and inverse Compton models

Fig. 15. Multi-instrument spectral energy distribution (SED) and combined model fit of the Crab Nebula. The colored markers show the flux points computed from the data of the different listed instruments. The horizontal error bar illustrates the width of the chosen energy band (E_{Min}, E_{Max}). The marker is set to the log-center energy of the band, that is defined by $\sqrt{E_{Min} \cdot E_{Max}}$. The vertical errors bars indicate the 1σ error of the measurement. The downward facing arrows indicate the value of 2σ upper flux limits for the given energy range. The black solid line shows the best fit model and the transparent band its 1σ error range. The band is too small be visible.

link to script

Activities in 2023

- 2 coding sprints in 2023
  - 2023 March 20-24th - UCM - Madrid
    - 9 people on-site, 6 online
  - 2023 October 16-20th IASF-INAF Palermo
    - 8 people on-site, 5 online



- ~ 40 Zoom developer meetings : every Friday 2pm CET
- > 450 merged PRs, 2400 commits since Nov 2022
  - 23 contributors

Feature release: priors

- Use cases for priors and possible implementation is presented in PIG 26

- Use cases:
  - prior on physical model parameter (e.g. positivity)
  - background systematics as nuisance parameters
  - unfolding methods for spectra

- v1.1 will provide:
  - priors on single parameters
  - few pre-defined priors, e.g. GaussianPrior
  - serialization in models.yaml
  - associated tutorial

A selection of new features

From gammapy 1.2

- `gammapy.modeling:`
  - priors on single parameters
  - computing pivot energy for any spectral model
- `gammapy.estimators:`
  - light curve variability estimation helper functions and associated tutorial
  - energy dependent morphology estimator
  - flux sensitivity map calculation
- `gammapy.catalogs:`
  - 4FGL DR4 catalog
  - LHAASO catalog
- `gammapy.astro:`
  - additional DM and halo models
- `gammapy.data:`
  - **metadata containers**
  - removal of time intervals from GTI
- `gammapy.maps:`
  - additional support for multiprocessing

Thanks to R. Terrier