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

Gammapy

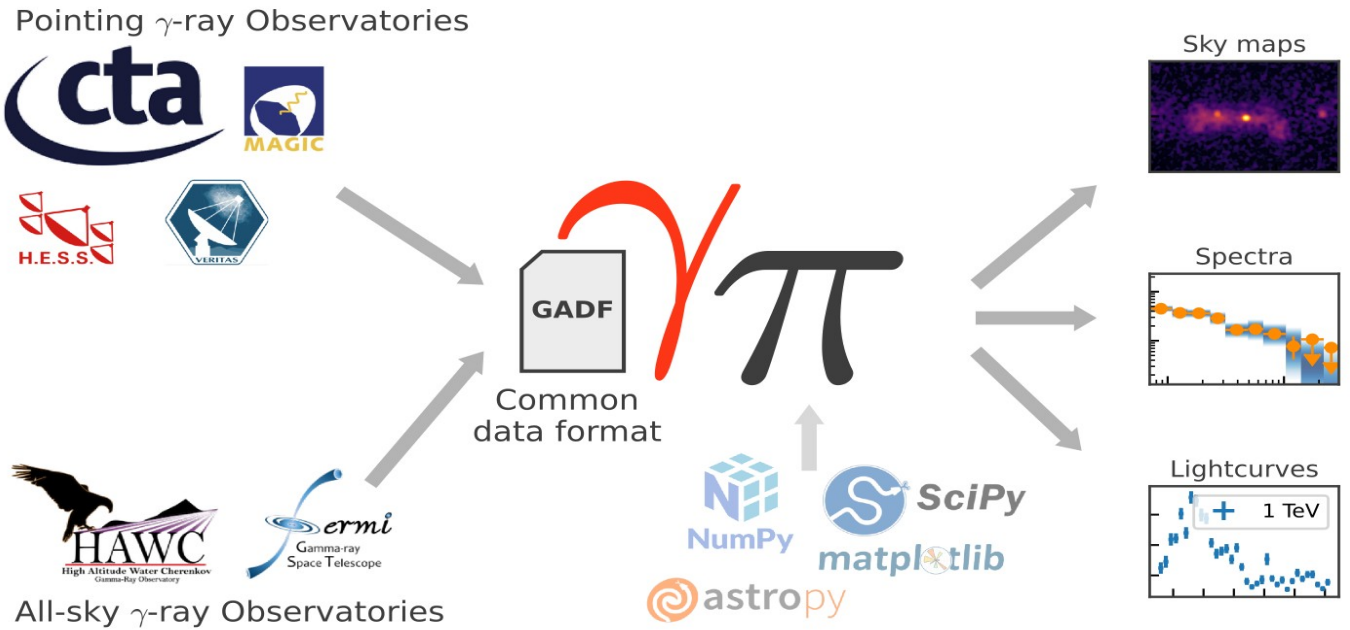
Python package for gamma-ray astronomy into the Open Science

Bruno Khélifi
Project manager, for the Gammapy team

RICAP-24, Villa Tuscolana, Frascati
September 26th, 2024

Open Science Tool: its concept

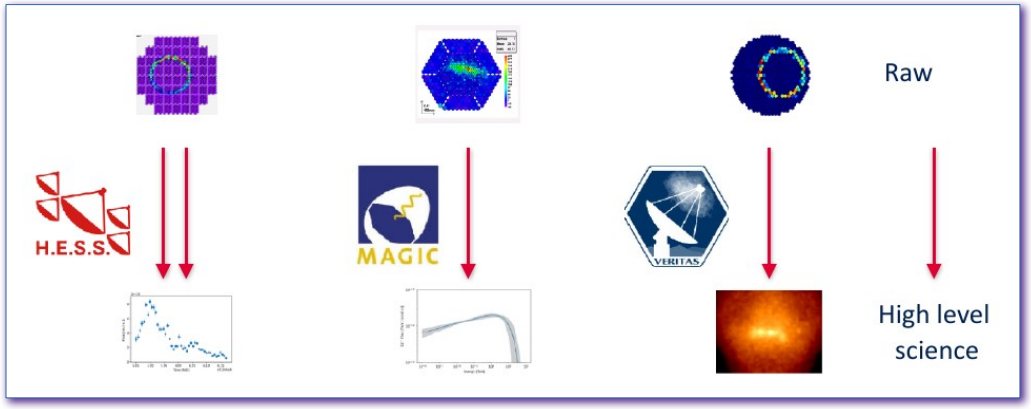
Python library to analyse high-level γ-ray data



Designed to analyse several data sets

The landscape 10 years ago

Proprietary data and formats, closed software tools



VHE community worked in a totally competitive and closed mode

- Except few MoUs around scientific projects

Some ‘dreamers’ worked towards the opening of the VHE astrophysics

- Data format standardization: open initiative ‘Open Gamma-Ray Astro’ → GADF format
- Open Science Tools: Gammapy

Deil, C., *et al.*, ASTERICS 2016 (link)

Better results
Interoperability between instruments
Respect of the FAIR principles

Early steps of Gammapy: 2014 – 2017

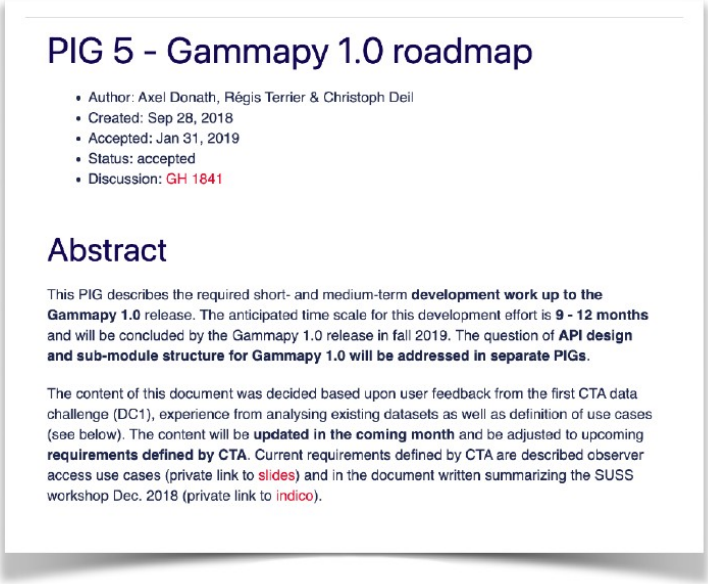
- Github repository creation in August 19th, 2013 (`TevPy`)
- First Gammapy release (v0.1) on August 25th, 2014
- Project evolved into a generic library for TeV astronomy and in prevision of the CTAO science analysis tool

« We would like to introduce Gammapy to the community and present our vision of Gammapy as a future community-developed, general purpose analysis toolbox for γ-ray astronomers. […] Its scope will continuously grow and we hope that many users and developers show interest in open and reproducible γ-ray astronomy with Python. As long-term goal we would like Gammapy to turn into a fully community-developed package. »

Donath, A., Deil, C. et al. ICRC 2015

Towards the LTS v1.0 : 2018 – 2022

- Rapid development cycle with frequent releases (~ 2 month)
  - From v0.7 to 0.20
- Structuration of the library & abstraction of analysis steps
- 19,000 commits from more than 80 contributors
- June 2021: Gammapy selected as official CTAO Science Analysis Tool
  - Used for the CTAO Real Time Analysis
- Version 1.0 released Nov. 10th, 2022



PIG 5 - Gammapy 1.0 roadmap

- Author: Axel Donath, Régis Terrier & Christoph Deil
- Created: Sep 28, 2018
- Accepted: Jan 31, 2019
- Status: accepted
- Discussion: GH 1841

Abstract

This PIG describes the required short- and medium-term **development work up to the Gammapy 1.0** release. The anticipated time scale for this development effort is **9 - 12 months** and will be concluded by the Gammapy 1.0 release in fall 2019. The question of **API design and sub-module structure for Gammapy 1.0 will be addressed in separate PIGs**.

The content of this document was decided based upon user feedback from the first CTA data challenge (DC1), experience from analysing existing datasets as well as definition of use cases (see below). The content will be **updated in the coming month** and be adjusted to upcoming **requirements defined by CTA**. Current requirements defined by CTA are described observer access use cases (private link to slides) and in the document written summarizing the SUSS workshop Dec. 2018 (private link to indico).

See v1.0 Gammapy paper: Donath et al. (2023)

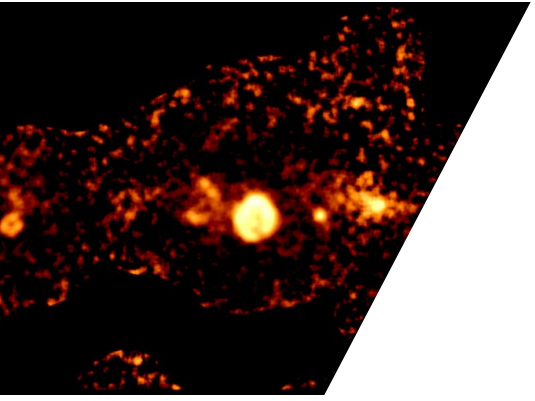
From 2014 to 2024

Talk content :

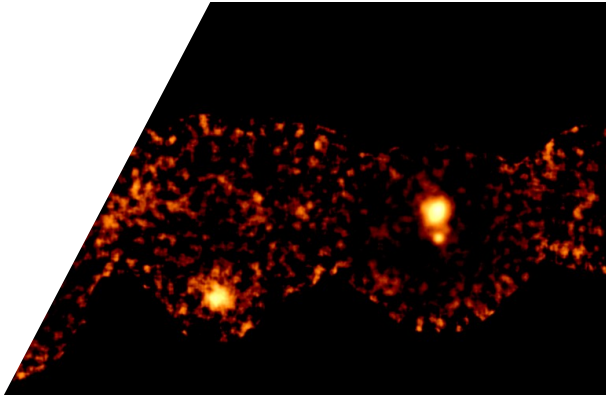
1- Design and features

2- Joint multi-instrument analyses

3- An open science project



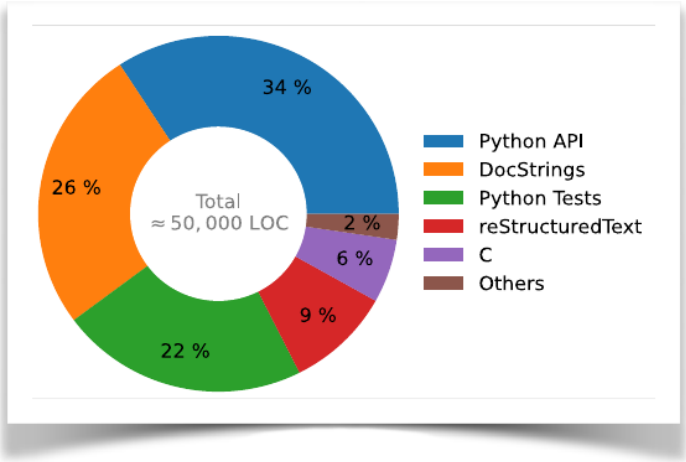
Design and features



The Gammapy library

Lightweight Python software

- Astropy-affiliated package
- Fermipy depends on Gammapy



Openly developed on Github

- 8-10 core contributors
- More than 80 contributors from the whole γ-ray astronomy community and beyond

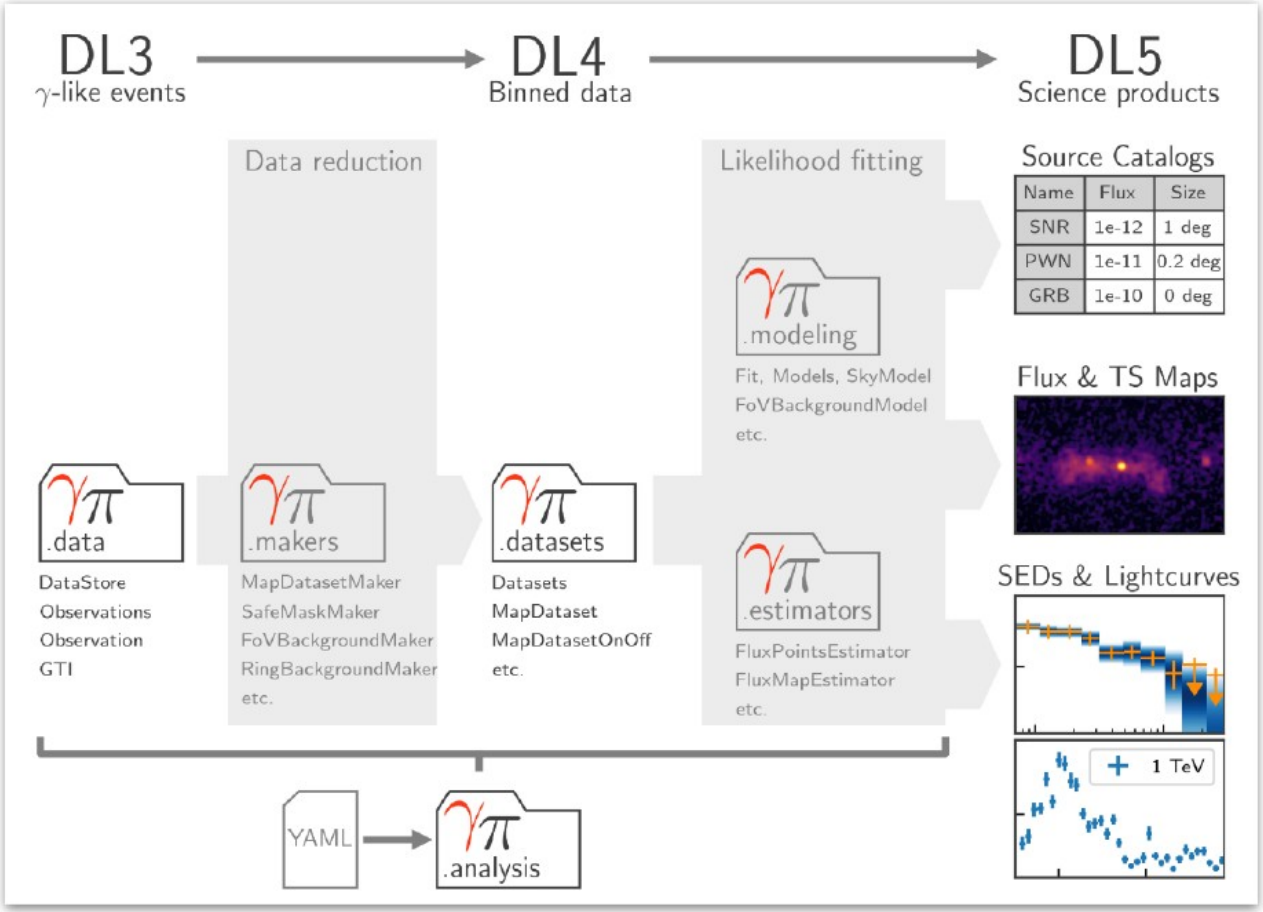
Distributed via PyPi and conda-forge

Data analysis workflow & package structure

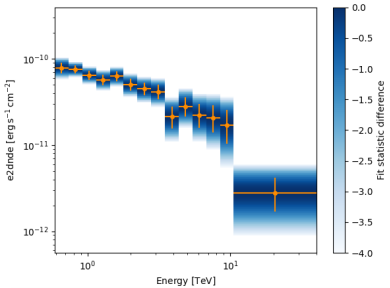
2-step analysis procedure:
- data aggregation and reduction (DL3 to 4)
- modeling / fitting (DL4 to 5)

Allow for joint data modeling at DL4 level

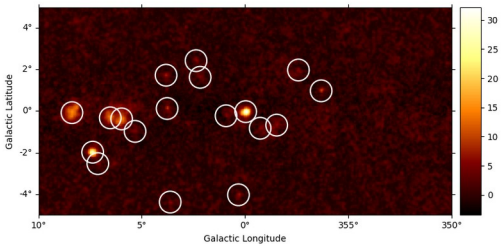
Flexible modeling library:
- physical models (e.g. naima)
- user designed models



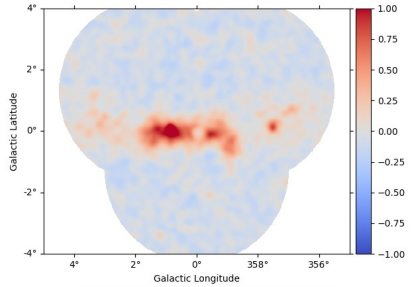
Typical analysis use cases



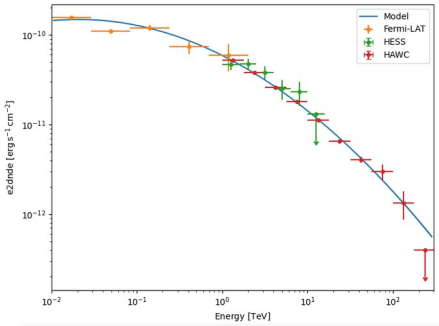
1D spectral analysis



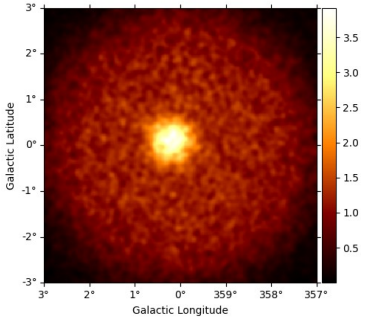
Source detection



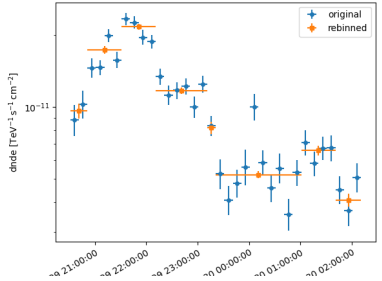
3D analysis



Multi-instrument fitting

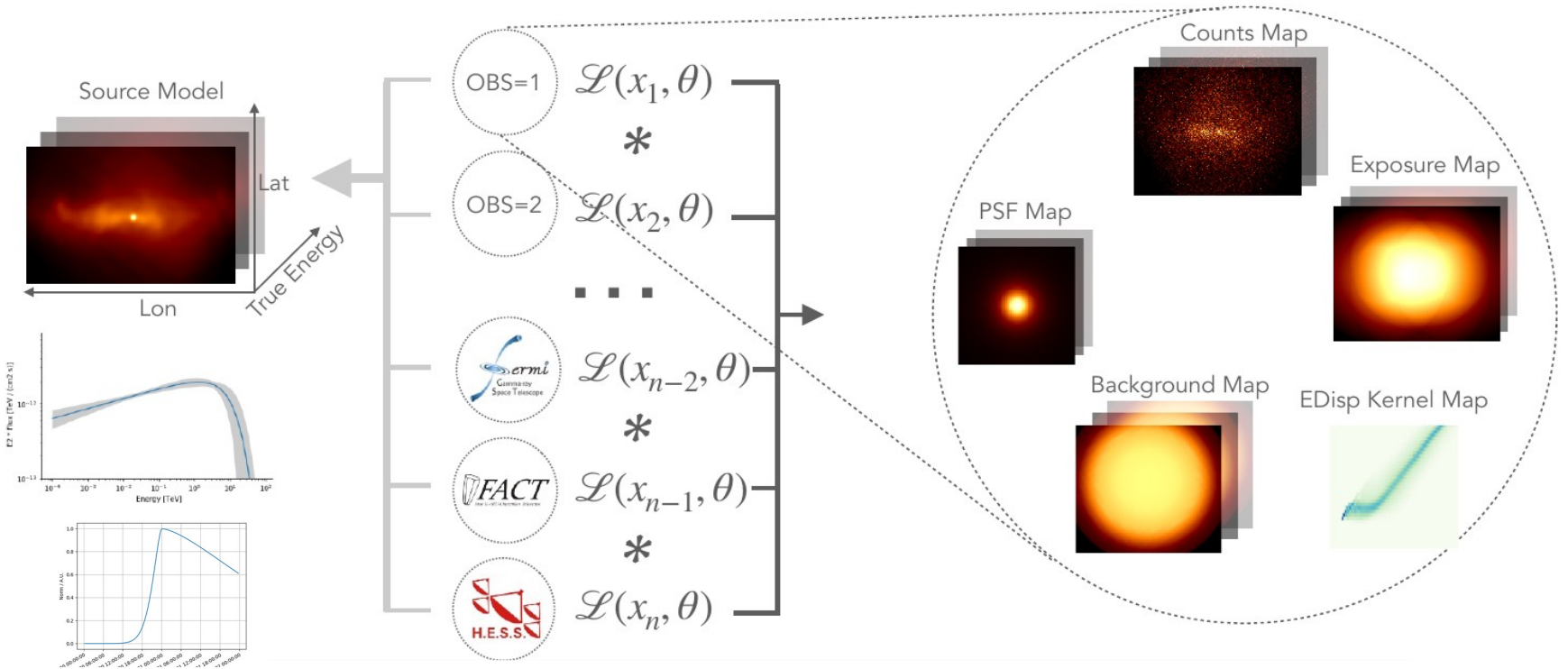


Observation simulation

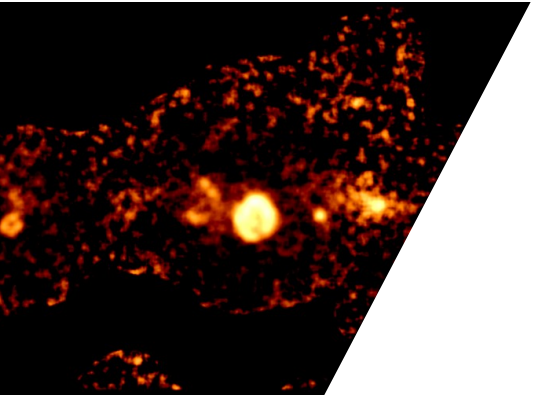


Light-curve extraction and time-variability estimation

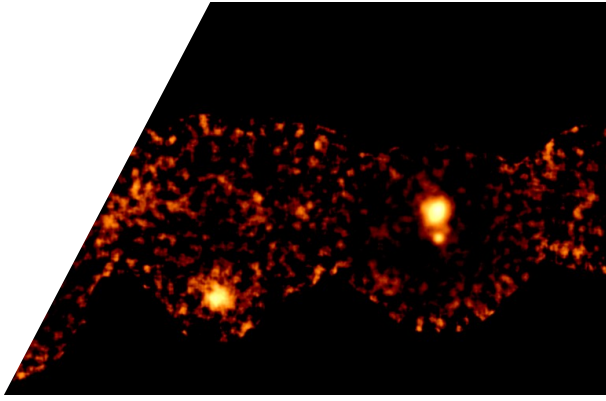
Joint likelihood and multi-instrument analysis



Gammapy `Dataset` structure allows heterogeneous data fitting. See joint fit tutorial



Joint multi-instrument analyses



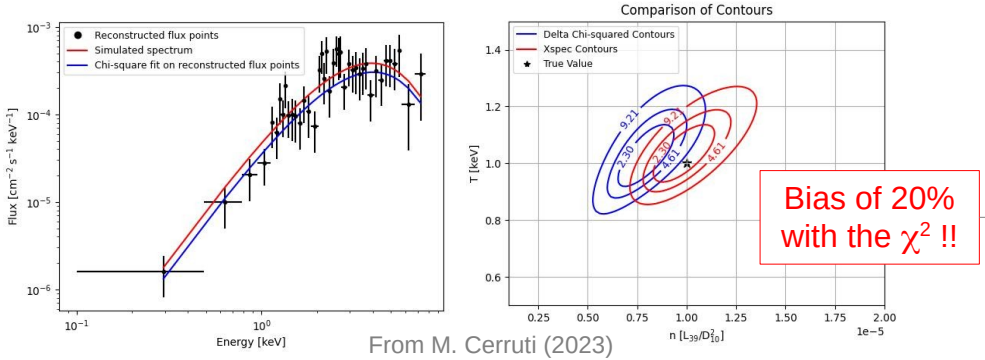
Challenges for multi-instrument analyses

Rigorous data analysis

- Need **correct handling of statistics**
  - In contrary to basic χ^2 fit on flux points!

Ex: simulation of
a BB seen by Swift
with XSPEC

Humphrey *et al.*, 2009 **ApJ** 693 822



From M. Cerruti (2023)

- And **inter-instrument systematics**
  - E.g. use of priors or parameters or IRFs

Readability of the IRFs

- IRFs from HE→UHE instruments can be factorised in the same manner
- Need of the use of standard formats! (GADF → VODF)

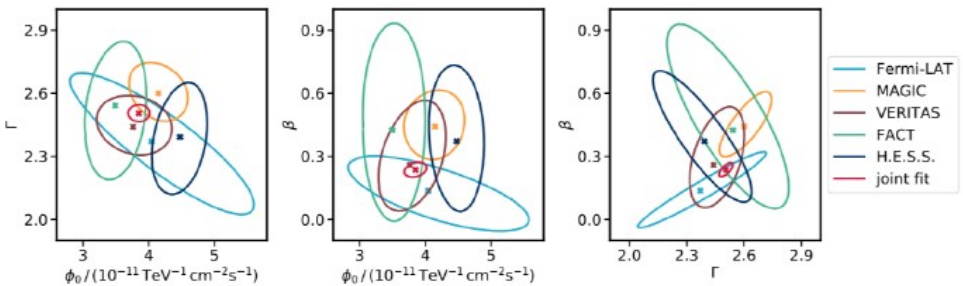
Use of 3D analysis

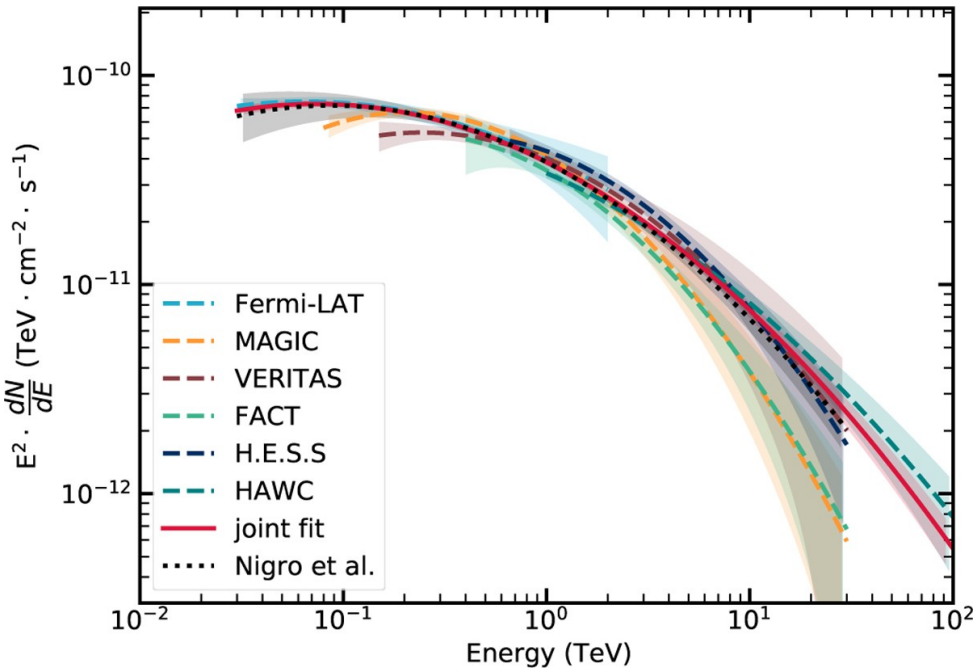
An example: from HE to UHE γ-rays

Joint point-like 1D spectral analysis of the Crab nebula

- 6 different instruments over 3.5 decades in energy
  - Simple log-parabola & physical inverse Compton model
  - Modelling of some systematic uncertainties
- Fully reproducible analysis

Nigro et al. 2019
+ HAWC: Albert et al. 2022





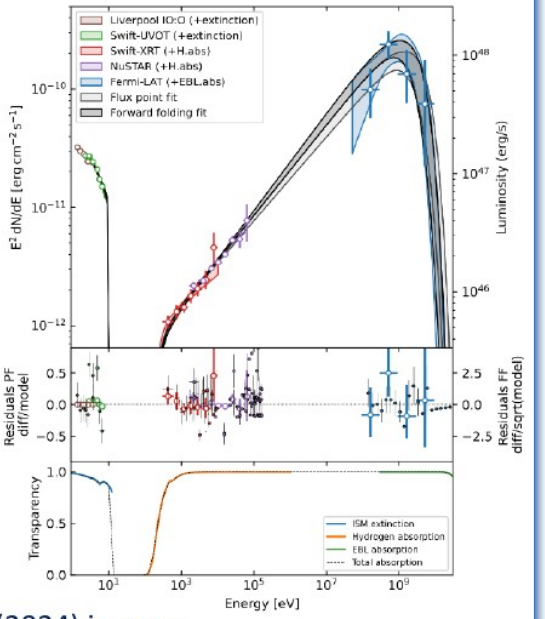
Other example: from optical to GeV

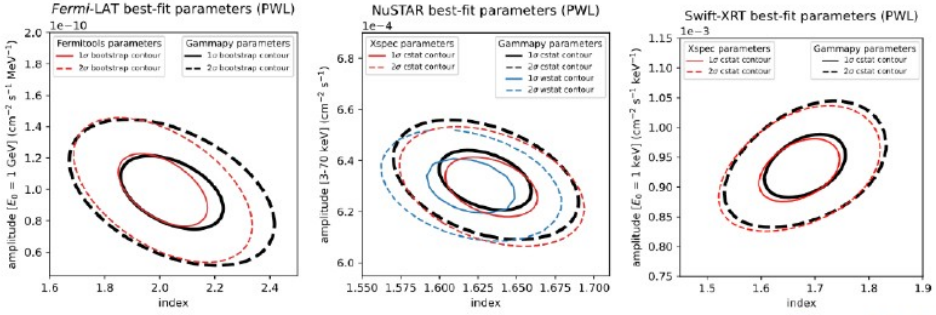
Multi-instrument analysis over 10 decades

- Joint forward fitting fit from eV to 10^{10} eV with Gammapy:
  Liverpool OT, Swift-UVOT, Swift-XRT, NUSTAR, Fermi-LAT

- Flux points lose some stat. information (e.g UL)
- Full forward fit provides more accurate results
- Gammapy facilitates the distribution and reproducibility of the results

OP 313 campaign





From M. Nievas et al (2024) in prep.

From R. Terrier, Gamma2024

An other one: from X-rays to VHE γ-rays

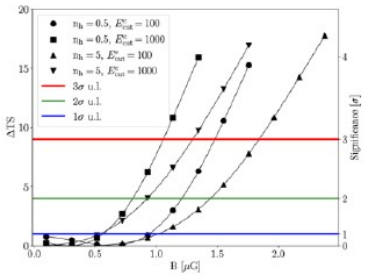
Multi-instrument analysis examples

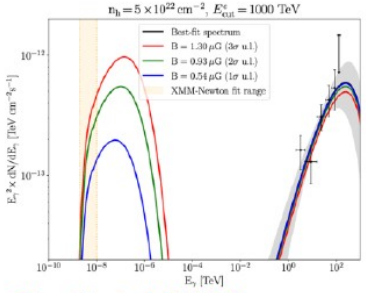
Joint X-ray and γ-ray fits

First approach:

- Read OGIP spectra (1D DL4) produced by X-ray telescopes and fit

gammapyXray package DOI 10.5281/zenodo.7092736



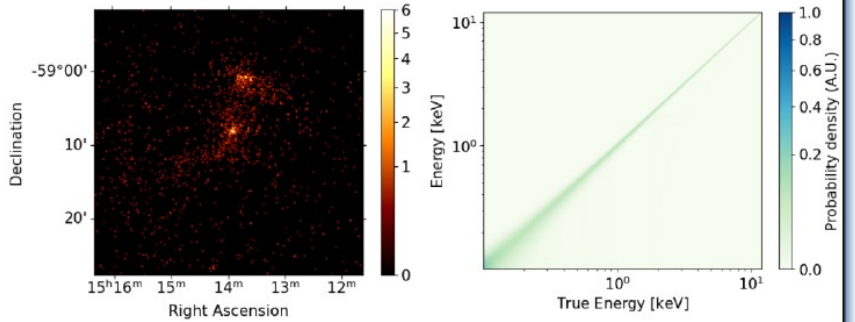


Giunti et al (2022)

Second approach:

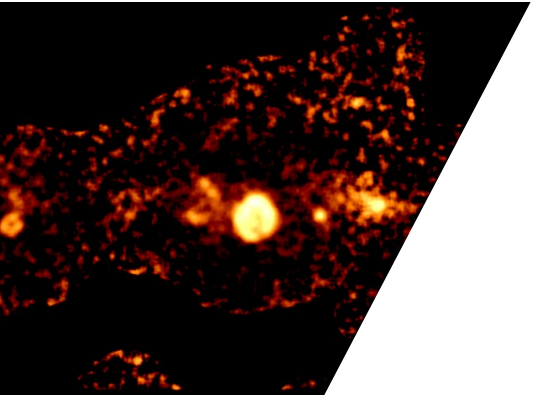
- Read X-ray events, IRFs and create 3D DL4 dataset

See e-ROSITA converter in K. Egg poster

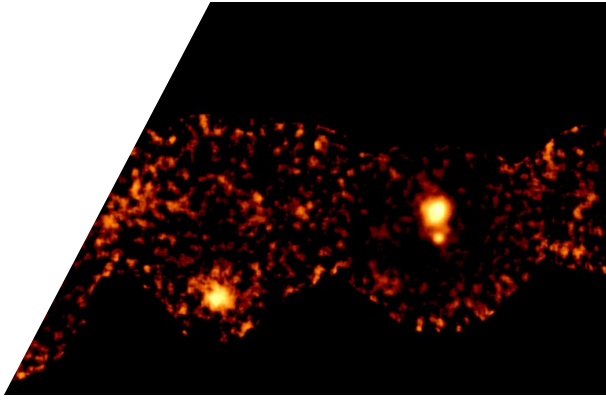


19

From R. Terrier, Gamma2024



An Open Science project



Open research software

Recognition and valorization

- Each release is a **real publication**
  - According to Open Science recommendations on the science evaluation
  - DOI and SWHID
- Transparent Gammapy **Authorship Policy**

Long-term archive

- On the universal archive **Software Heritage** (sustained by UNESCO)

Open science activities

- VHE standards : → creation of the “High Energy IG” [IVOA, Malta, Nov’24]
- VHE data format : GADF →
- Support of any open project that can be affiliated to Gammapy

Dissemination and outreach

Presentations, hands-on sessions

Material available here and here

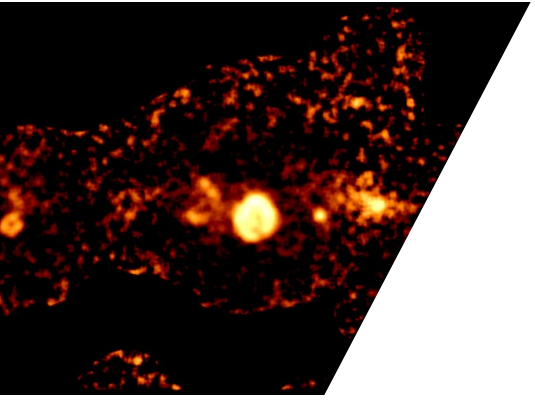
Schools

- ORP school on Multi-messenger Astrophysics, Durham University, Sept 2-6
- CTAO school, La Palma June 22 – 29
- MPIK-CDY school on the future of γ-ray astronomy, MPI-K HD, June 25-July 3
- IFSC school, São Marcos Feb./March

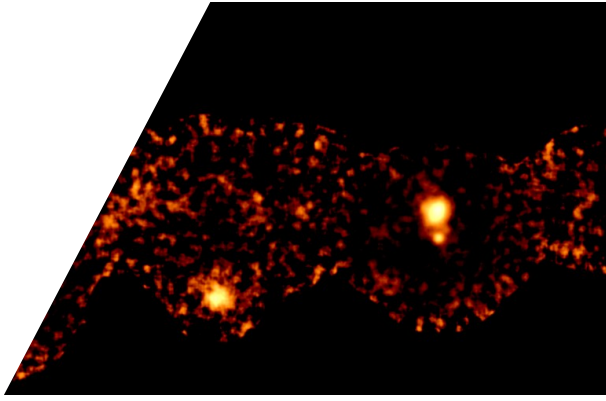
Advertisement: Cherenkov Astronomy Data School (CADS)
beginner and advanced hands-on sessions
Observatoire de Paris, October 14-18 2024, more on https://indico.obspm.fr/event/2480/

Within the collaborations

- Dedicated support and training



Conclusions



Towards the LTS v2.0 and beyond

Towards ML & MM data analysis

- UHE (HAWC → SWGO)
- X-ray handling improvement
- Neutrino processing
- Unbinned likelihood analysis

Better respect of s/w standards

- FAIR4RS principles (towards reproducibility)
- Some IVOA standards (e.g. Provenance)

Improved performance

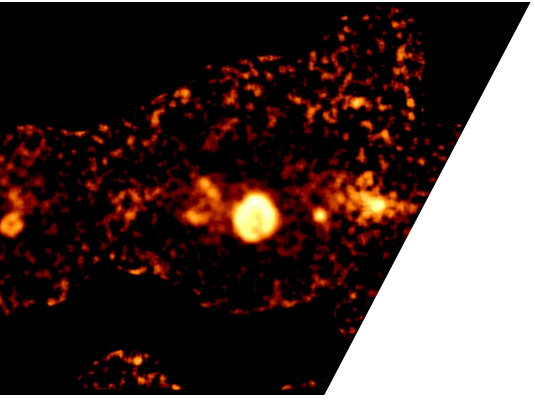
- Computing time (optimisation, `JAX`?)
- Memory (caching, sparse array, etc)

Get involved
in the adventure

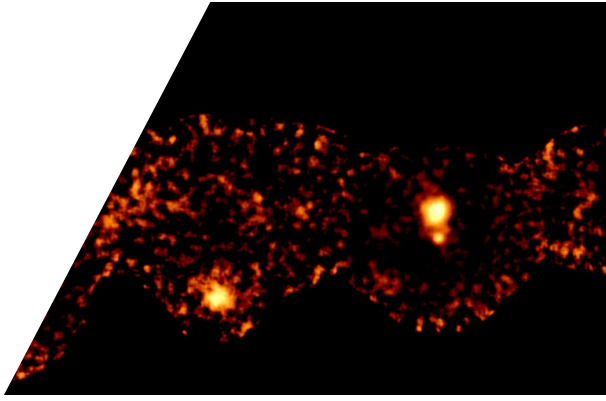
2 post-doc positions
be announced soon

Thank you for your attention

RICAP-24



Back-up slides



Project Organization

Projects managers
non-technical executive lead

Lead developers
technical executive leads

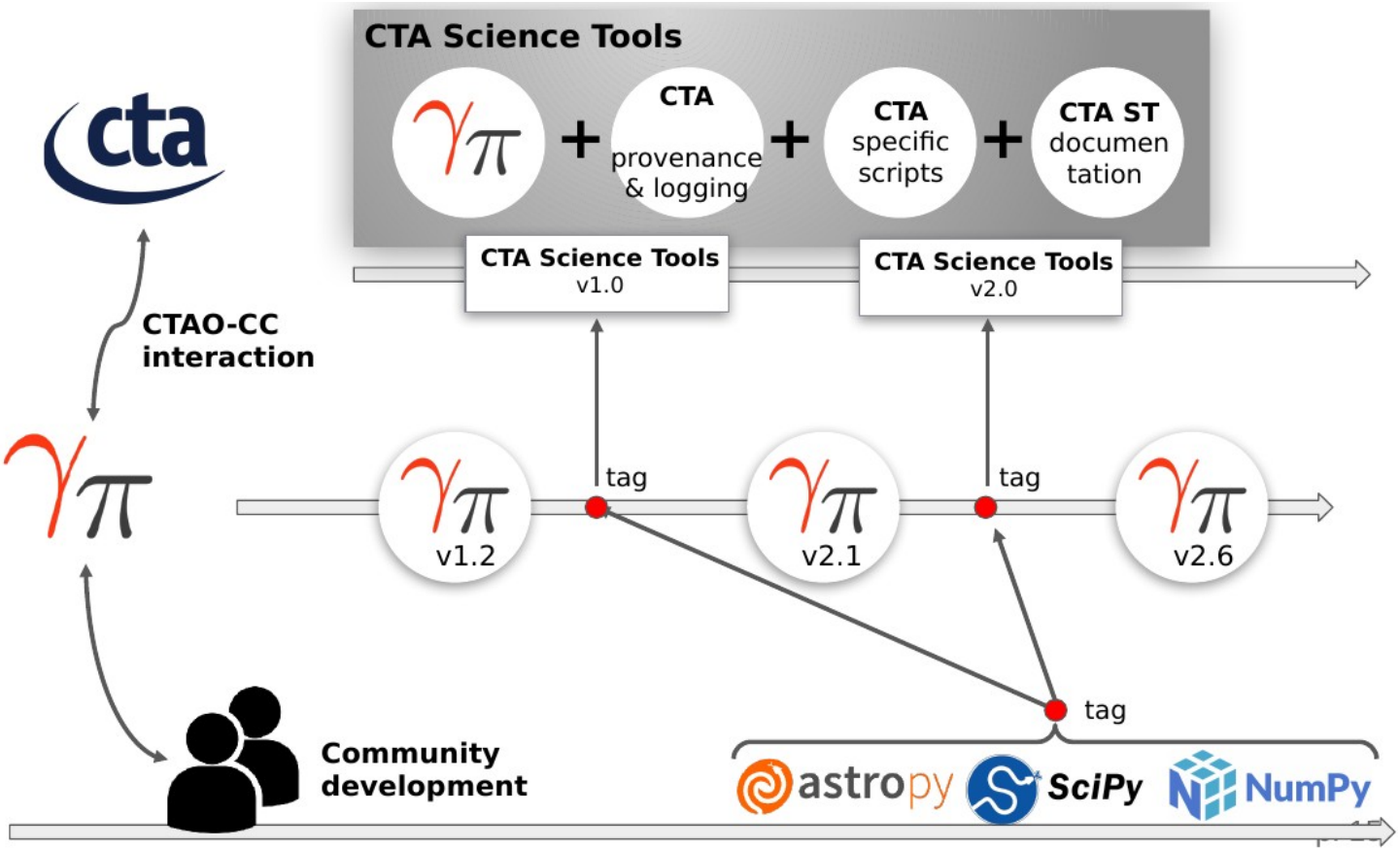
Sub-packages maintainers
core developer devoted to the maintenance of some sub-packages

Contributors
>80 individual contributors from various collaborations and beyond

Coordination Committee
« Promotes, coordinates and steers Gammapy developments »

See https://gammapy.org/team.html

Gammapy and CTAO SAT



Towards the LTS v2.0: features selection

- Parameter prior can now be defined on parameters and the associated log-prior is added to the total statistics during fitting
- Added timing studies utility functions for light curves, see tutorial
- Preliminary support for asymmetric IRFs. See tutorial
- Energy dependent ON-region size for 1D spectral analysis
- And many more...

- More to come in v1.3 (in October!) and beyond
  - Improved support for event types and joint analyses
  - Multi-parameters priors and spectral unfolding
  - Lightcurve simulation and PSD study tools
  - ...

Poisson Log-Likelihood

Common algorithms for the libraries: Poisson Log-Likelihood

"Cash statistics": summed over all "bins"

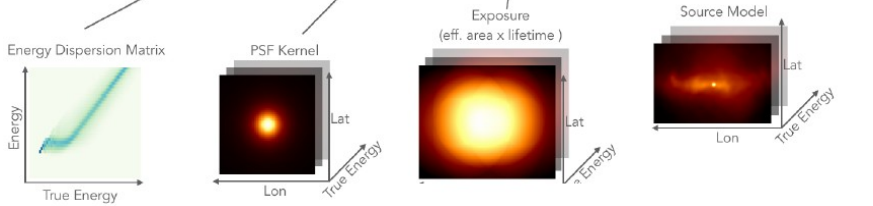
$$\mathscr{C} = 2\sum_i N^i_{Pred} - N^i_{Obs} \cdot \log N^i_{Pred}$$

i: spectral channels or 3D voxels

$$N_{Pred} = N_{Bkg} + \sum_{Src} N_{Pred,Src}$$

$$N_{Pred,Src} = \mathrm{EDISP}_{Src}(\mathrm{PSF}_{Src}(\mathscr{E}_{Src} \cdot f_{Src}))$$

$$f_{Src} = f_{Spectral}(E) \cdot f_{Spatial}(E,l,b) \cdot f_{Temporal}(t)$$



- Bins in the spectral, spatial, temporal domain

→ Need of a "global" background model template with "correction parameters"

→ Need of the "signal" IRFs and source models

Most of the time,
Identical factorization of the IRFs
for X-rays → UHE & neutrino exp.

FAIR4RS principles

As with the FAIR Guiding Principles, the FAIR4RS Principles (2022) are intended to be aspirational. The application of the FAIR4RS Principles is the responsibility of the owners (who are often the creators) of the software, not the users.

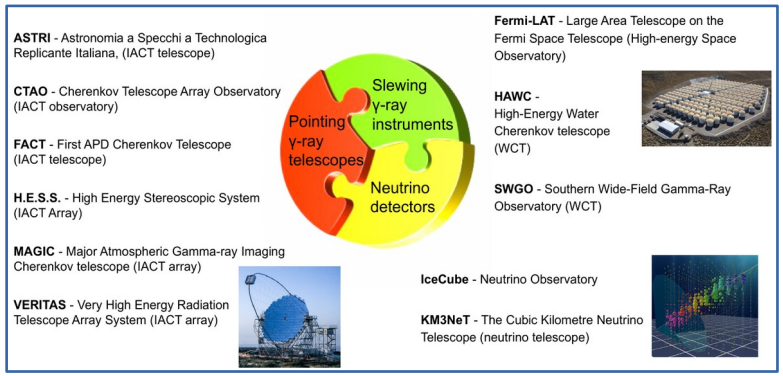
- **F** Software, and its associated metadata, is easy for both humans and machines to find
- **A** Software, and its metadata, is retrievable via standardized protocols
- **I** Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.
- **R** Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software)

VHE data format: VODF

Open Initiative ‘Very-high-energy Open Data Format’ (link)

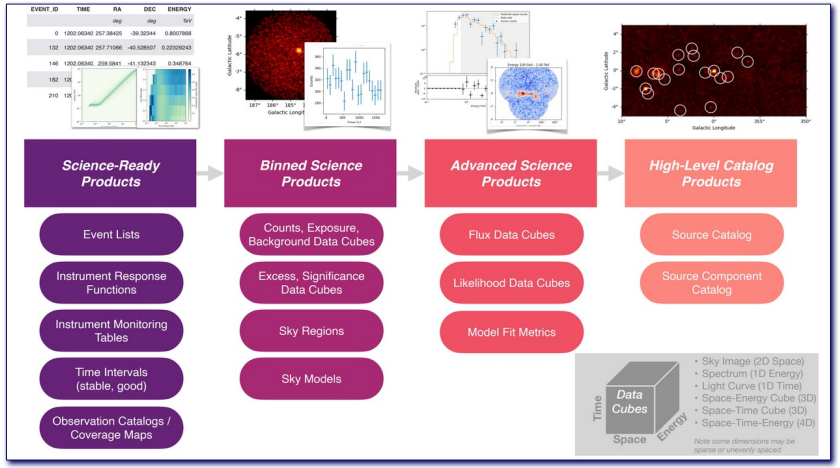
- Aims to format VHE data (gamma and neutrino)

- Officially supported by 11 experiments



- Structured with a project organization

Coordination Committee, Conveners: R. Zanin, B. Khélifi
Lead Editors: K. Kosack, L. Olivera-Nieto, J. Schnabel



Khélifi, B., *et al.*, Proc. of 38th ICRC (2023)

```
/*****************************************************************************
/* FITS FILE:
/*    VODF Level-1 Event Data
/*
/* EXTENSIONS SUMMARY:
/*  IDX  NAME               VER CLASS                TYPE
/*  ---  -----------------  --- -------------------  ----------------
/*    0. EVENTS               0 OGIP.EVENTS          [TableExtension]
/*    1. SOI                  0                      [TableExtension]
/*****************************************************************************

/ ############################################################################
/ HDU: EVENTS
/ DESCRIPTION:
/    VODF Level 1 Event List
/ ############################################################################
XTENSION = BINTABLE
EXTNAME  = EVENTS
EXTVER   = 0
```

Getting help / giving feedback - And get involved!!

- **Where/How to interact** with dev team and experienced users, provide feedback, get help:
  - gammapy.slack
    - In particular: #help channel
  - GitHub discussions
    - help category
  - GitHub issues to report bugs or feature requests

- **To get involved**
  - Development guide
  - The dev calls: each Friday at 14h CEST!
  - The hands-on sessions and schools, the recipes
  - Etc

Getting the software

- **Quickstart installation with conda (ex: LTS)**

```
curl -O https://gammapy.org/download/install/gammapy-1.0.2-environment.yml
conda env create -f gammapy-1.0.2-environment.yml
conda activate gammapy-1.0.2
```

- **Installation with pip**

```
pip install gammapy
```

install all dependencies

- **Download tutorials & associated data**

```
gammapy download notebooks
gammapy download datasets
export GAMMAPY_DATA=$PWD/gammapy-datasets
```

datasets are now versioned

See: