Gammapy: An open source python package for gamma-ray astronomy

Atreyee Sinha

IPARCOS/UCM, Madrid

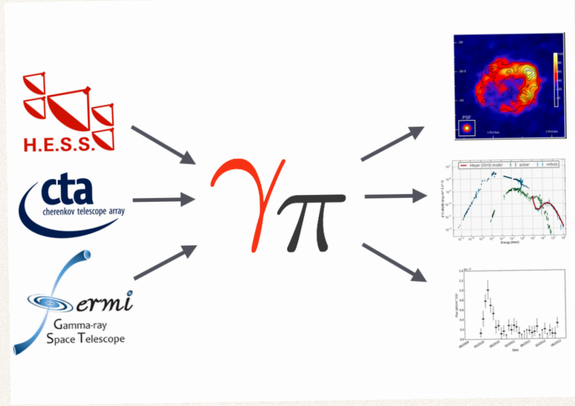
R. Terrier, A. Donath, B. Khelifi, L. Giunti, L. O. Nieto, M. Noethe, C. Nigro et al for the Gammapy dev team

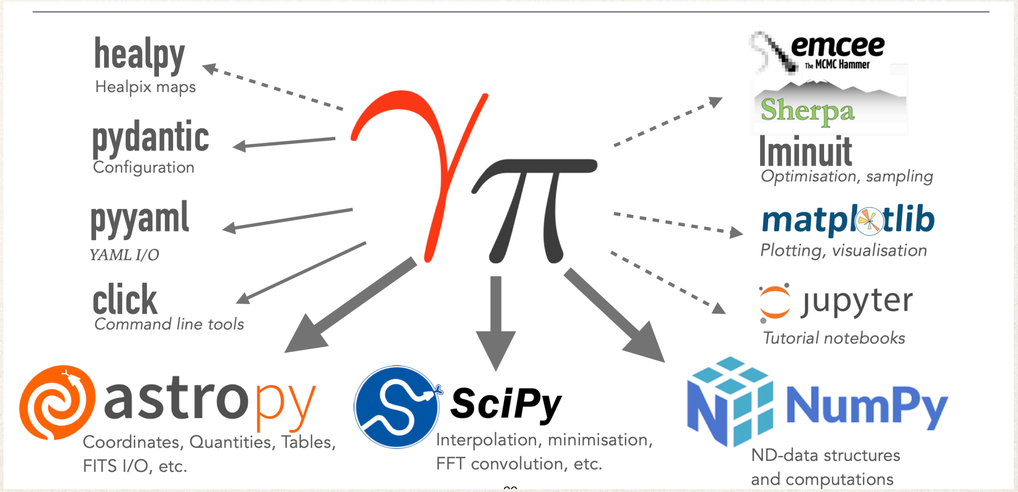
7th Heidelberg International Symposium on High Energy Gamma-Ray Astronomy, Barcelona, Spain, July, 2022

See: https://gammapy.org

The gammapy concept

- Flexible, open source, community driven python library
- Embedded in the python ecosystem
- Based on common data formats (defined in the GADF) See: https://gamma-astro-data-formats.readthedocs.io/en/v0.2/
  - Library for the CTA science tools (and used by many other experiments like MAGIC, H.E.S.S., etc)

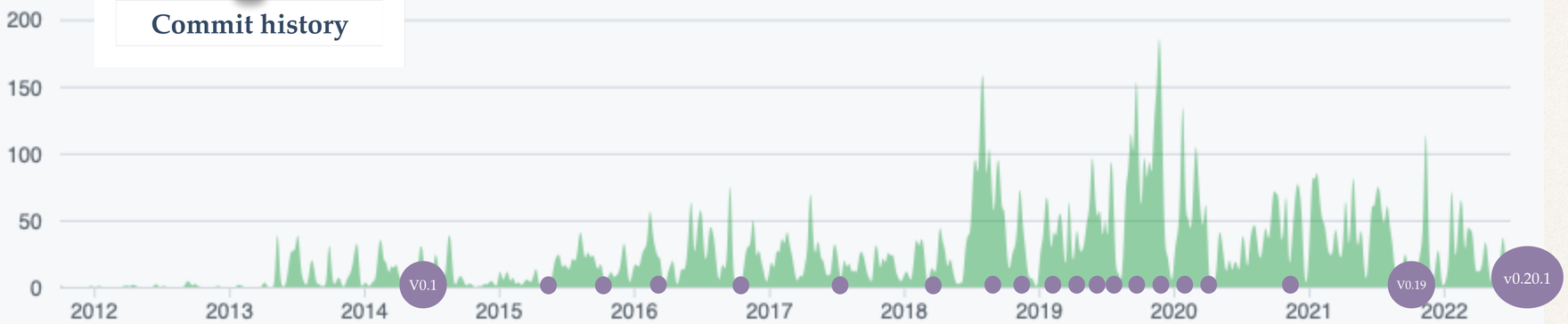




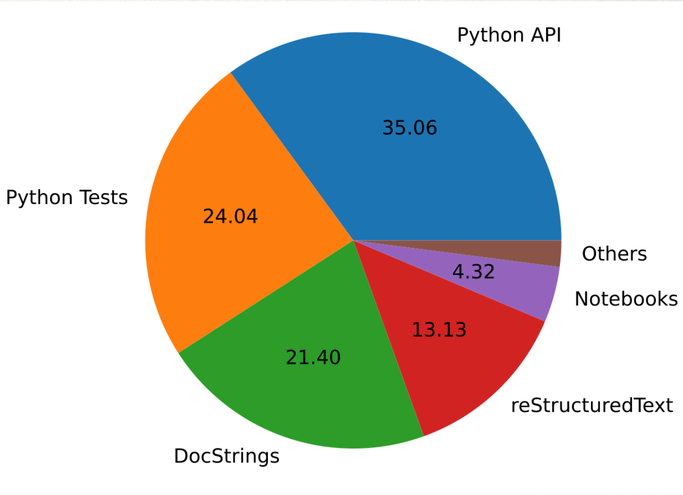
Development history

Total Lines :	93,063	Code Lines :	54,420	Percent Code Lines :	58.5%
Number of Languages :	3	Total Comment Lines :	20,155	Percent Comment Lines :	21.7%
		Total Blank Lines :	18,488	Percent Blank Lines :	19.9%

Commit history



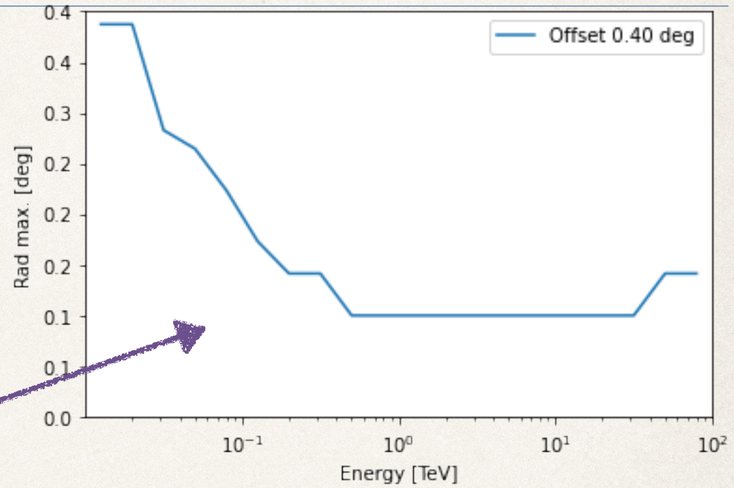
# of contributors

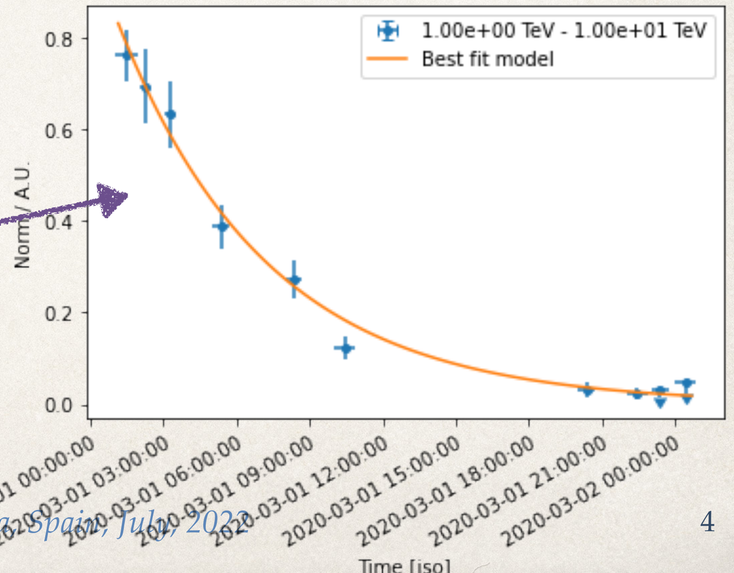


For detailed list, see:
https://docs.gammapy.org/0.20.1/release-notes/v0.20.1.html

What's new in the latest version?

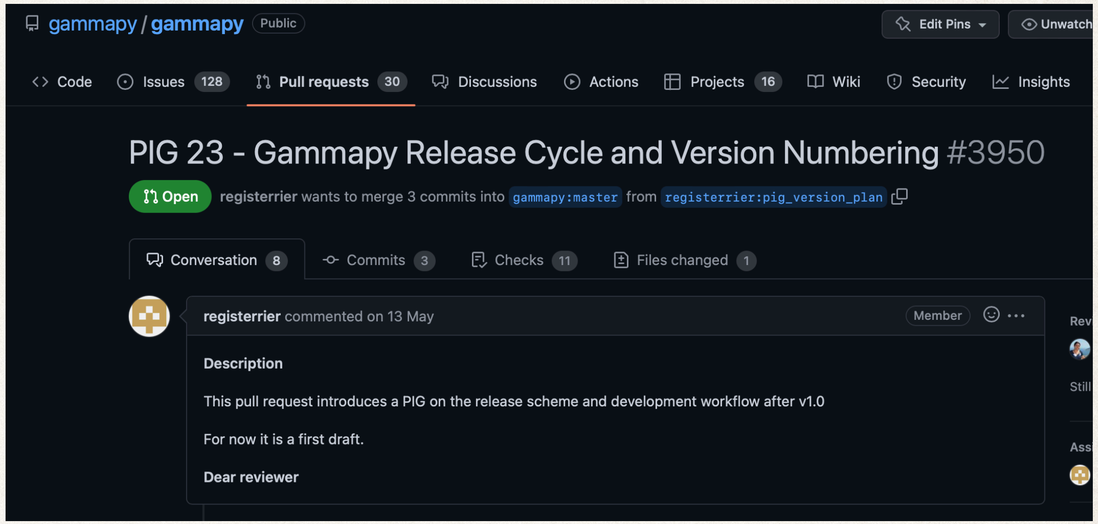
- Features:
  - A new documentation theme
  - Updated dependencies
  - Support for energy dependent on-region spectral extraction:
    - Used by the MAGIC and the LST-1 for point-like analysis
  - Support for fitting temporal models to light curves
- Bug fixes…





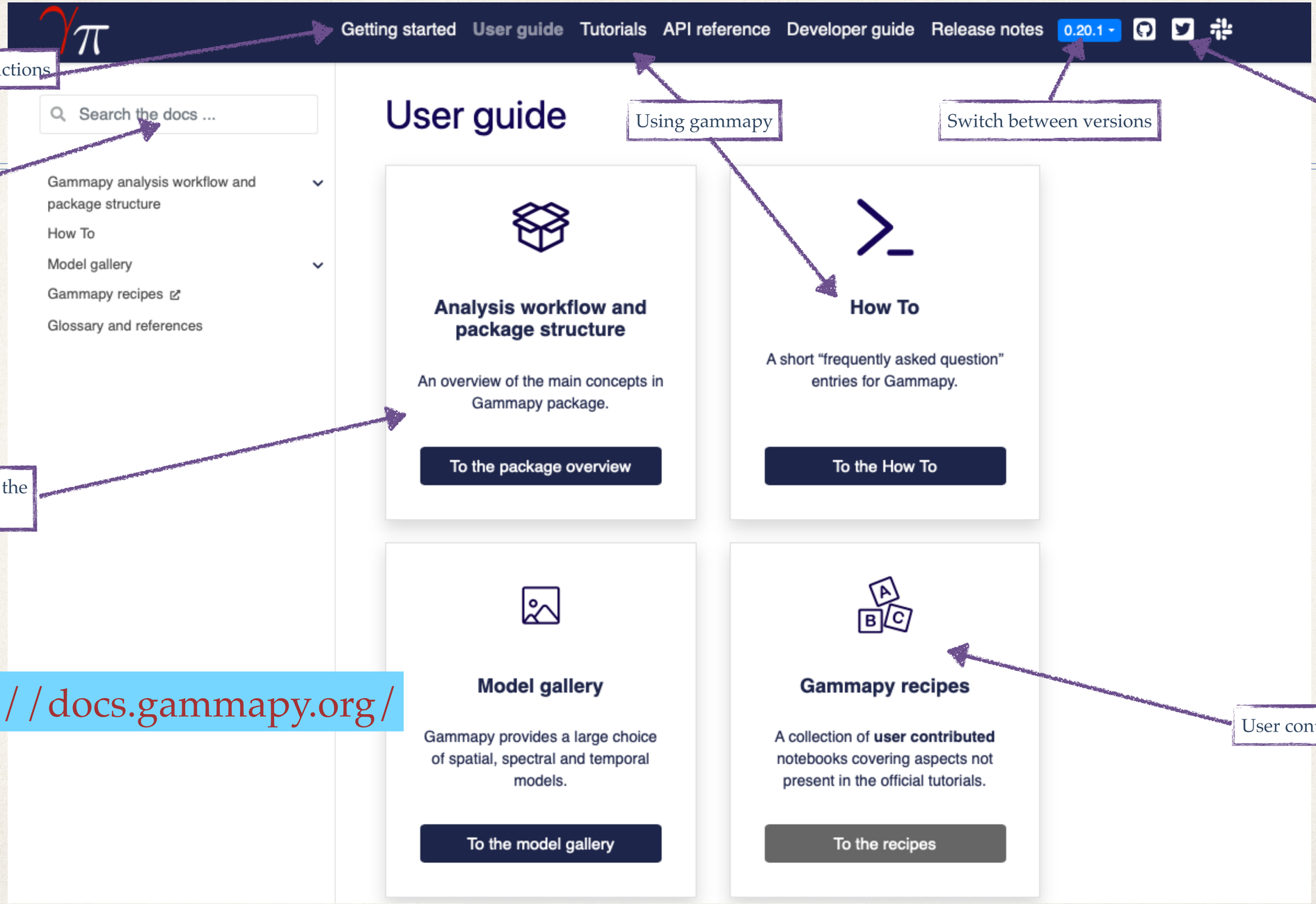
Towards a long term stable release

- Present development: Regular feature releases with added functionalities and bug fixes
  - Not backwards compatible
  - Urgent cases: Bug fix releases within ~2 week
- Future plans
  - A stable API with relevant bug fix releases
  - Feature releases at regular intervals
  - Requires planning and dedicated maintenance
  - **LTS version 1.0** scheduled for release later this year



PIGs: Proposals for Improvement of Gammapy

Getting started with gammapy



Installation instructions

Search box

Using gammapy

Switch between versions

Links to get in touch

Explanations on how the Package works

User contributed notebooks

See: https://docs.gammapy.org/

How To for quick look ups

- HowTos for specific use cases
- Links directly to the code block in the tutorials
- We welcome your suggestions!

Detailed workflow in the tutorials

Tutorials

This page lists the Gammapy tutorials that are available as Jupyter notebooks. You can read them here, or execute them using a temporary cloud server in Binder.

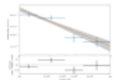
To execute them locally, you have to first install Gammapy locally (see Installation) and download the tutorial notebooks and example datasets (see Getting started). Once Gammapy is installed, remember that you can always use `gammapy info` to check your setup.

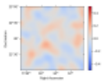
Gammapy is a Python package built on Numpy and Astropy, so to use it effectively, you have to learn the basics. Many good free resources are available, e.g. A Whirlwind tour of Python, the Python data science handbook and the Astropy Hands-On Tutorial.

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.

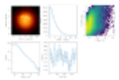

High level interface


Low level API

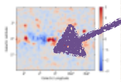

Data structures

Data exploration

These three tutorials show how to perform data exploration with Gammapy, providing an introduction to the CTA, 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

H.E.S.S. with Gammapy


Fermi-LAT with Gammapy

Data analysis

The following set of tutorials are devoted to data analysis, and grouped according to the specific covered use cases in spectral analysis and flux fitting, image and cube analysis modelling and fitting, as well as time-dependent analysis with light-curves.

- Most common analysis recipes
  - Spectral fitting
  - Source detections
  - Temporal Modelling
  - 3D FoV analysis
  - Simulations

- Getting started
  - Instrument specific data handling
  - Underlying data structures
  - A full analysis workflow

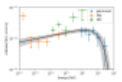
Data analysis

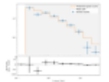
The following set of tutorials are devoted to data analysis, and grouped according to the specific covered use cases in spectral analysis and flux fitting, image and cube analysis modelling and fitting, as well as time-dependent analysis with light-curves.

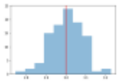
1D Spectral

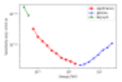

Spectral analysis


Spectral analysis with energy-dependent directional cuts

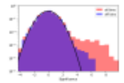

Flux point fitting

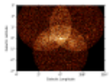

Spectral analysis of extended sources

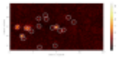

1D spectrum simulation


Point source sensitivity

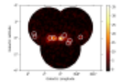
2D Image

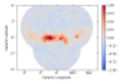

Ring background map

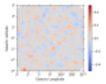

2D map fitting

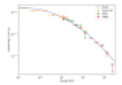

Source detection and significance maps

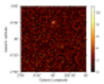
3D Cube

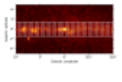

Basic image exploration and fitting


3D detailed analysis


3D map simulation


Multi instrument joint 3D and 1D analysis


Event sampling

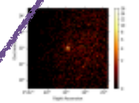

Flux Profile Estimation

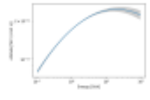
- Learn how to use the general API
  - Go beyond the tutorials
  - Understand the underlying API
  - Exploit Gammapy flexibility!
  - Test fit diagnostics
  - Add your own models
  - —

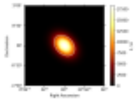

Pulsar analysis

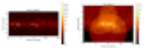
Package / API

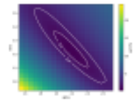
The following tutorials demonstrate different dimensions of the Gammapy API or expose how to perform more specific use cases.

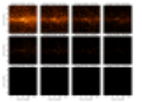

Makers - Data reduction

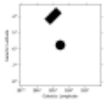

Source catalogs

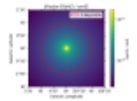

Models

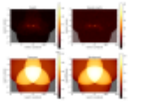

Modelling


Fitting


Maps


Mask maps


Dark matter spatial and spectral models


Datasets - Reduced data, IRFs, models

Scripts

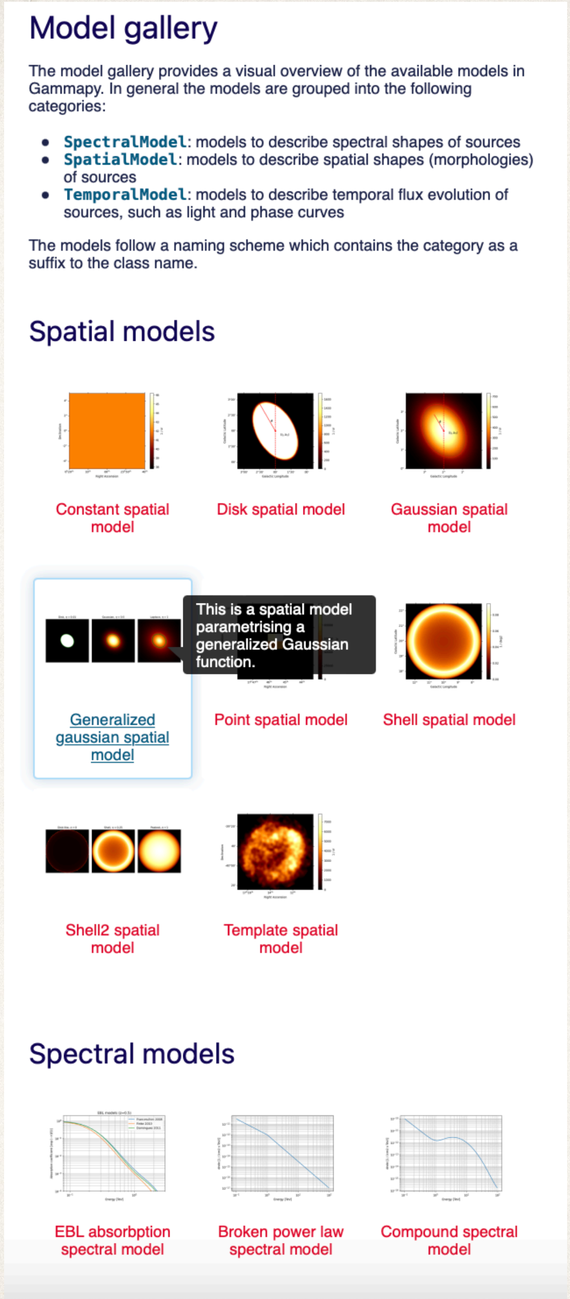
For interactive use, IPython and Jupyter are great, and most Gammapy examples use those. However, for long-running, non-interactive tasks like data reduction or survey maps, you might prefer a Python script.

The following example shows how to run Gammapy within a Python script.

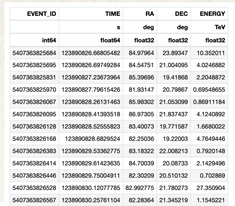
- Survey map

Models gallery

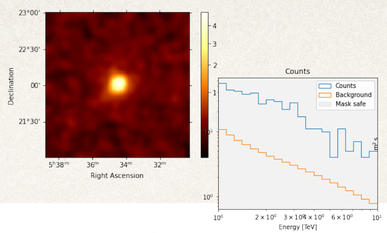
- A variety of inbuilt models in the Models Gallery
- Spatial, Temporal and Spectral Models
- Easy to add your own custom models

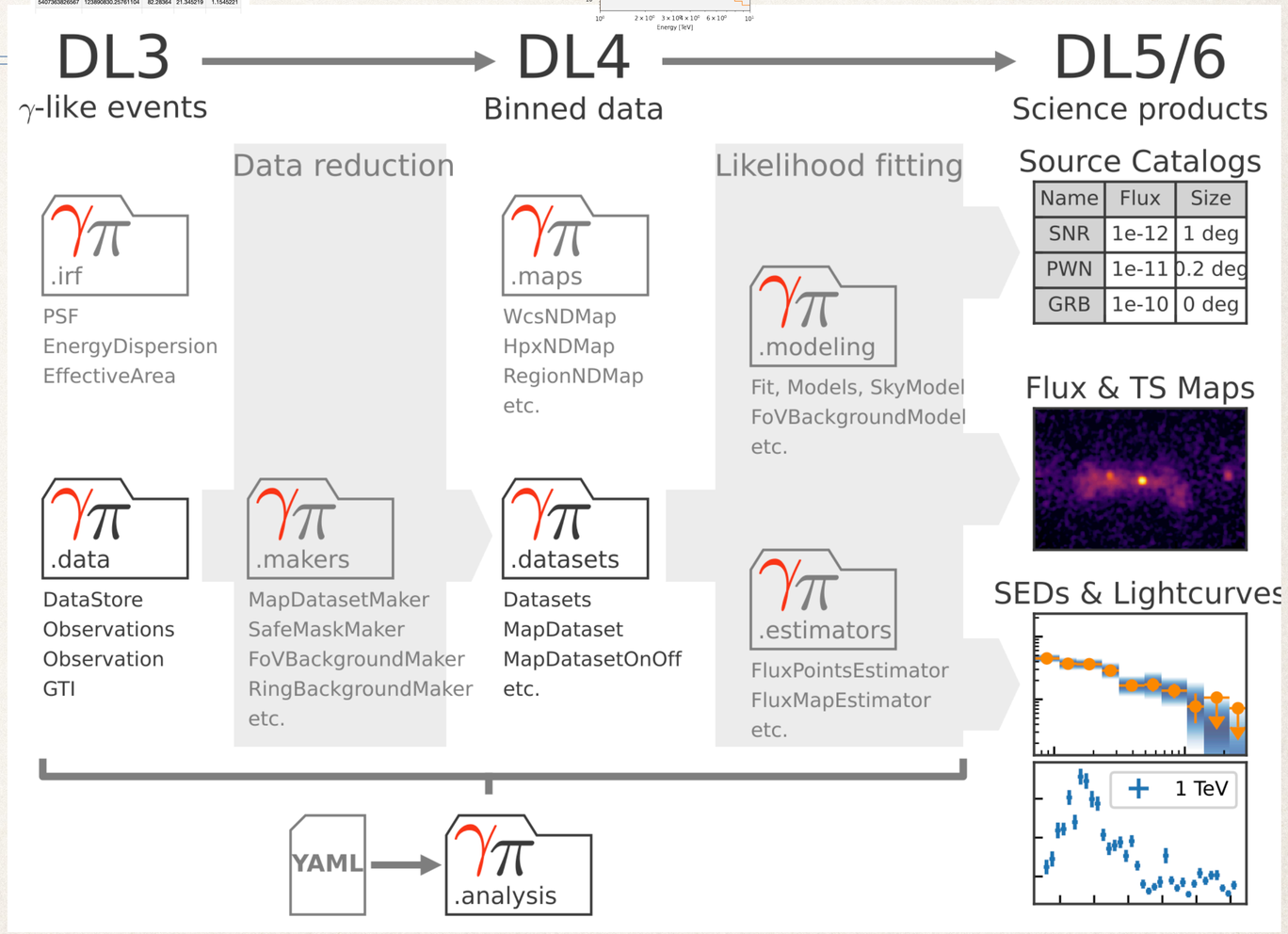


Gammapy internal workflow



EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
5407363825684	123890826.66805482	84.97964	23.86947	10.352011
5407363825695	123890826.69749284	84.54751	21.004095	4.0246882
5407363825831	123890827.23673964	85.39696	19.41868	2.2048872
5407363825970	123890827.79615426	81.83147	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
5407363826414	123890829.61423635	84.70039	20.08733	2.1429496
5407363826446	123890829.75004911	82.30209	20.510132	0.702869
5407363826528	123890830.12077785	82.992775	21.782073	27.350904
5407363826567	123890830.25761104	82.28364	21.345219	1.1545221





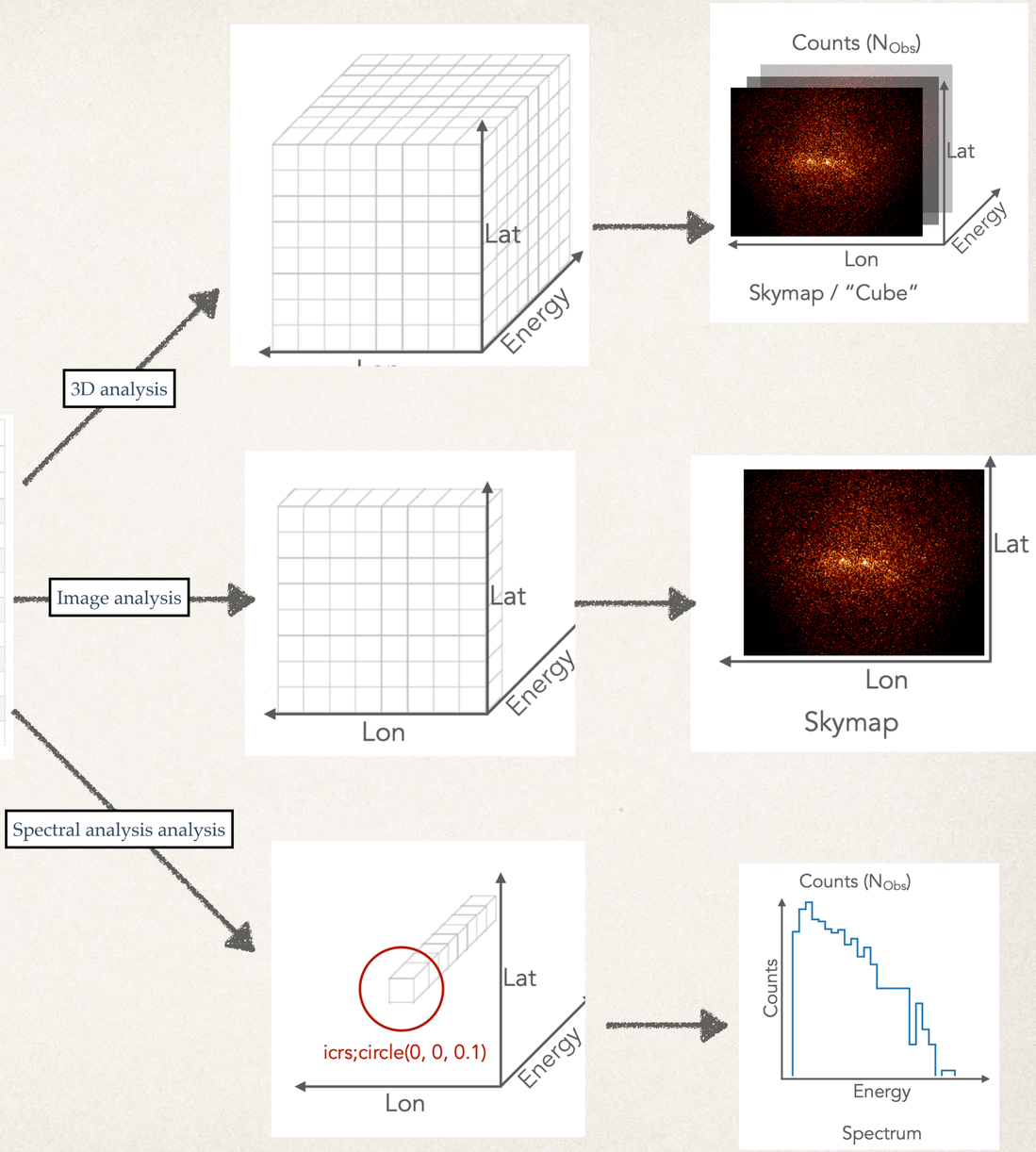
- 2 step workflow
  - Data reduction (DL3 - DL4)
  - Modelling and fitting (DL5 - DL5)

Data reduction

DL3 ⟶ DL4 ⋯⋯▸ DL5

- Bin events (and IRFs) into n-dim sky maps
  - Apply event selections (time, offset, etc)
  - Spatial and energy binning
- Generalised case: 3D maps
  - Image analysis: cube with one energy bin
  - Spectral analysis: Cube with one spatial bin

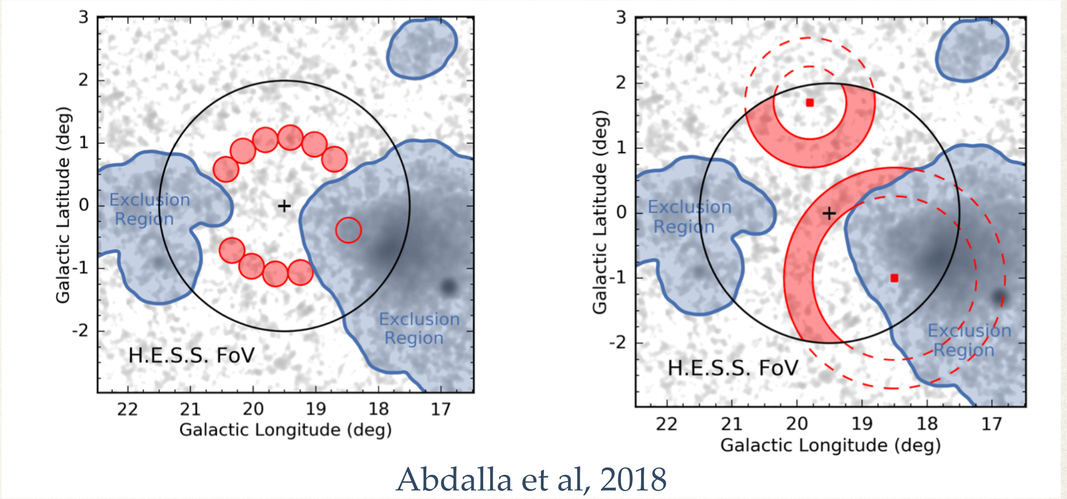
EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
5407363825684	123890826.66805482	84.97964	23.89347	10.352011
5407363825695	123890826.69749284	84.54751	21.004095	4.0246882
5407363825831	123890827.23673964	85.39696	19.41868	2.2048872
5407363825970	123890827.79615426	81.93147	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
...	...	...	...	...



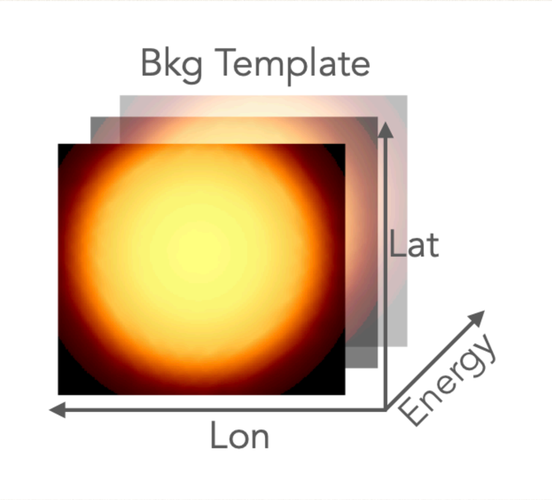
Data reduction

DL3 ⟶ DL4 ⇢ DL5

- Choose the background estimation algorithm
  - Reflected background
  - Ring background
  - FoV background to be modelled simultaneously with the source
    - ➡ Need full enclosure IRFs, with available background models!
- Choice of background exclusion regions
- Appropriate choice of statistics



Abdalla et al, 2018



Data fitting
DL4 ⟶ DL5

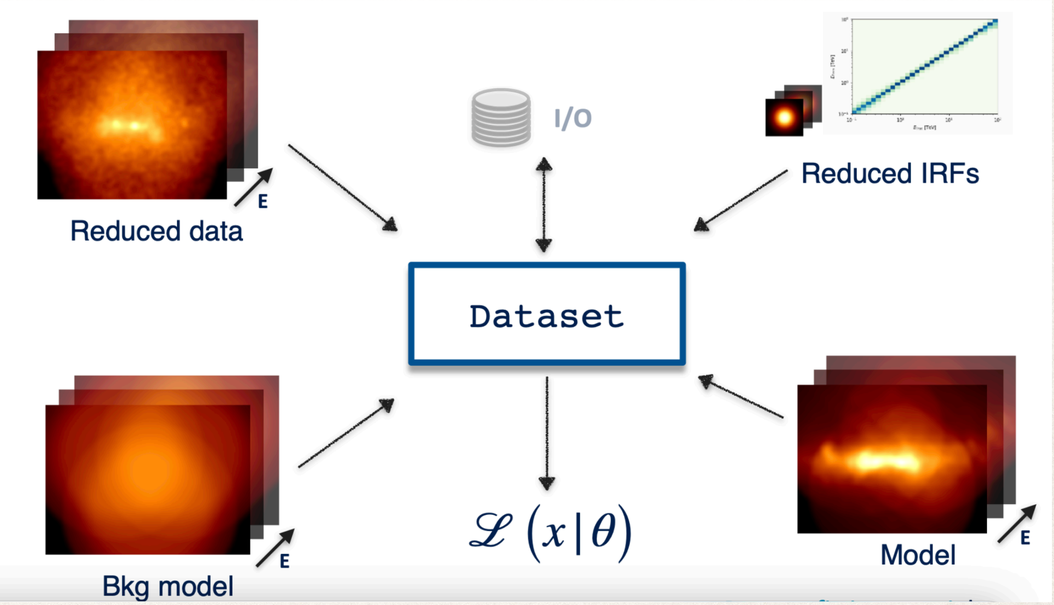
- Fitting on pre-computed datasets
  - eg: From HAWC, Fermi-LAT, OGIP files, etc
- Forward folding with maximum likelihood estimation

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

Cash: known background

$$TS = -2\ logL = 2\Sigma\ (\ N * logN_{pred} - N_{pred}\ )$$

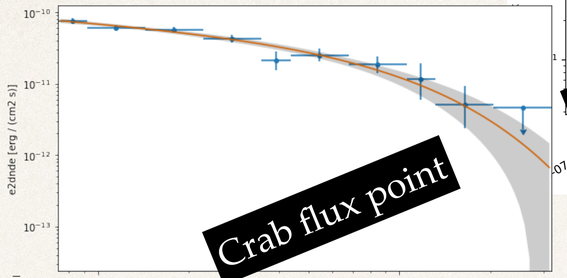
Wstat: counts with measured background

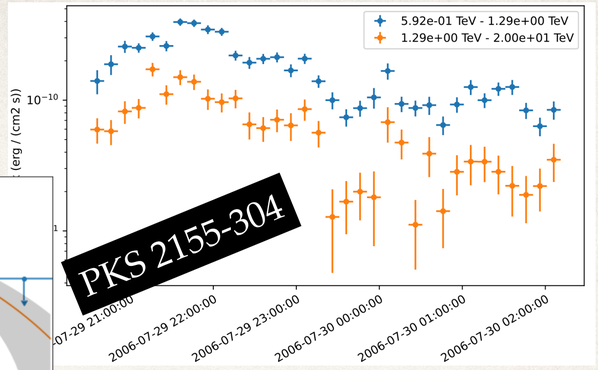


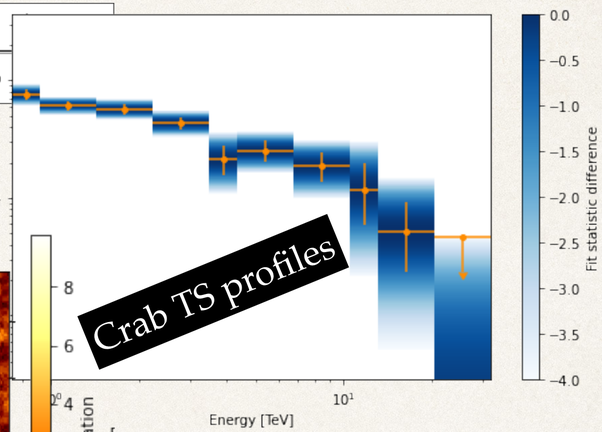
End products: DL5 and DL6

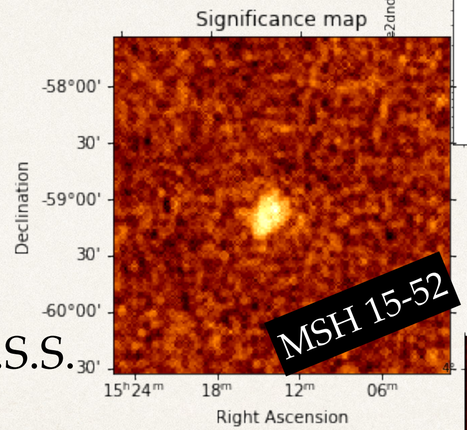
See: https://docs.gammapy.org/0.20.1/tutorials/index.html

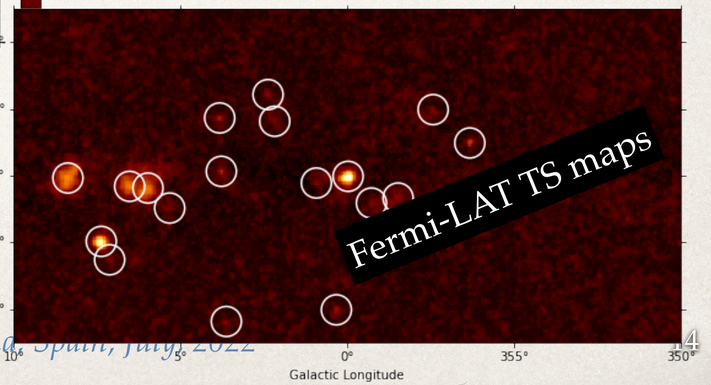
- DL5: Flux points, light curves and flux/TS maps
- Possible to fit DL5 data
  - Eg: published flux points, lightcurves
  - Chi2 statistics used
- DL6: catalogs
  - Support provide for common catalogs: Fermi 4FGL, H.E.S.S. galactic plane survey, HAWC catalog, etc
  - Create your own catalogs…





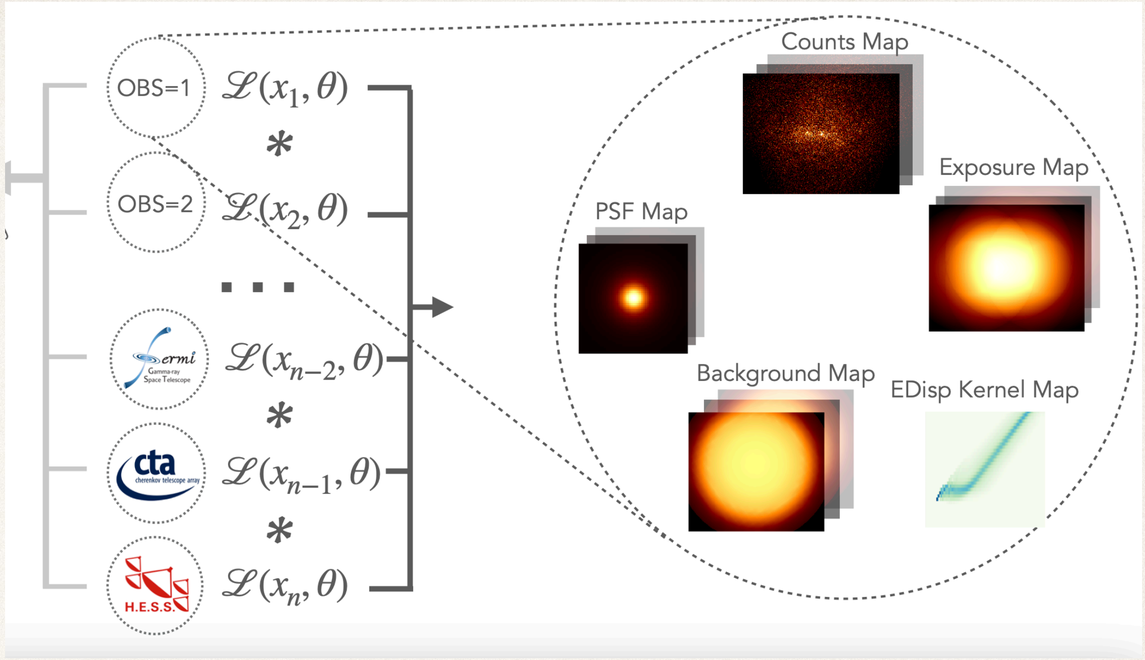


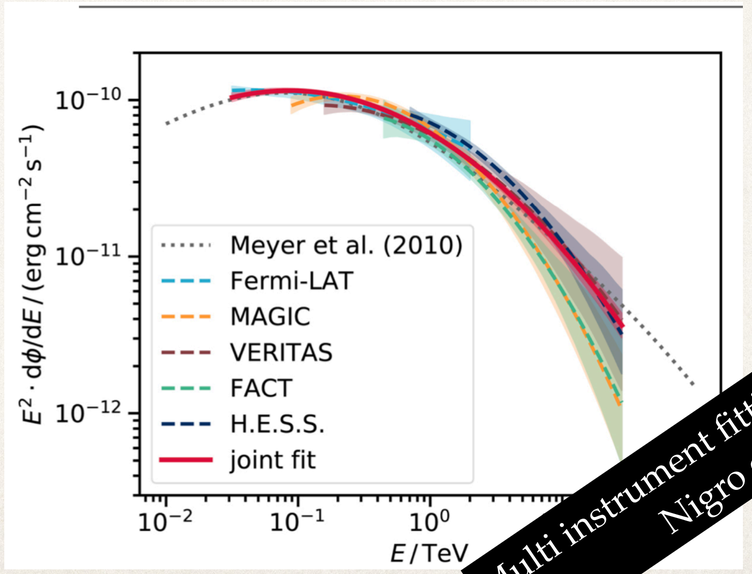




Joint likelihood

- Simultaneous fitting of various datasets
- Likelihood evaluated per dataset, individual likelihoods combined to get global likelihood
- May come from the same or different instruments
- Possible to combine DL4 and DL5 data



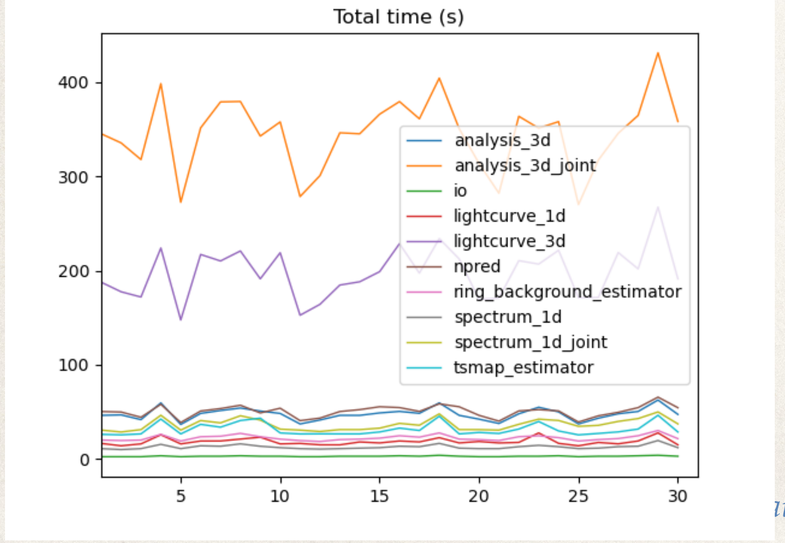


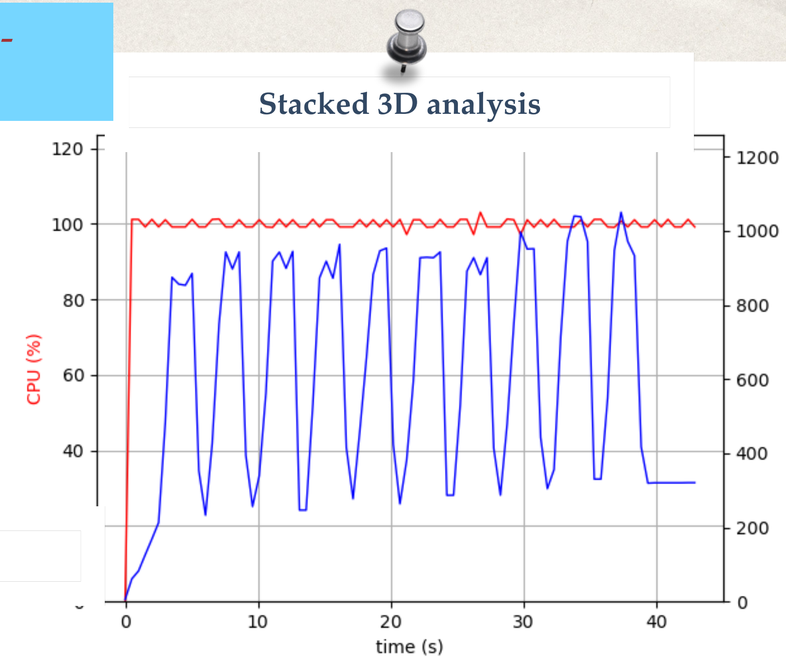
Multi instrument fitting of Crab Nebula
Nigro et al, 2019

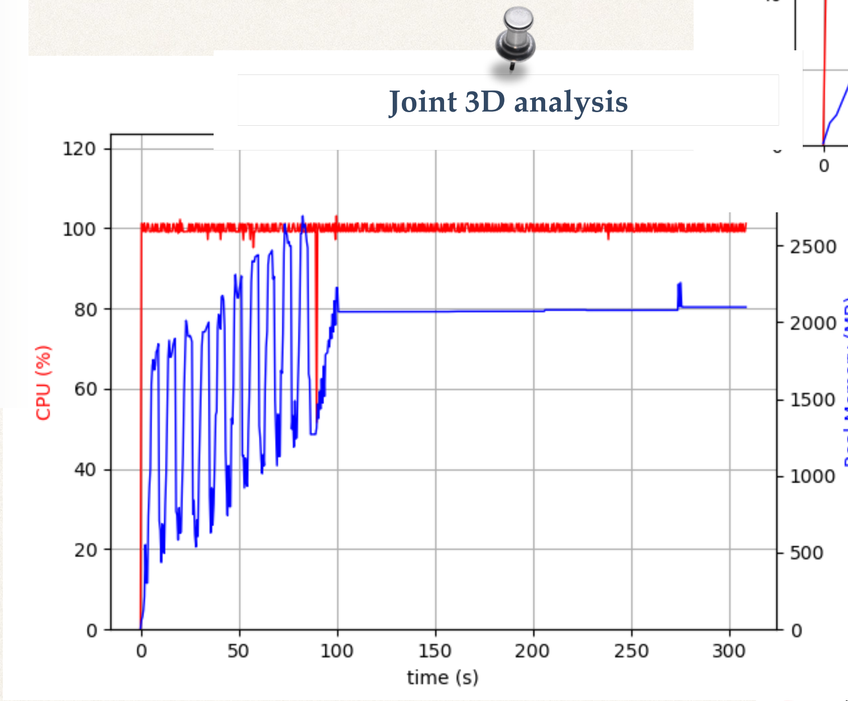
See: https://github.com/gammapy/gammapy-benchmarks/tree/master/benchmarks

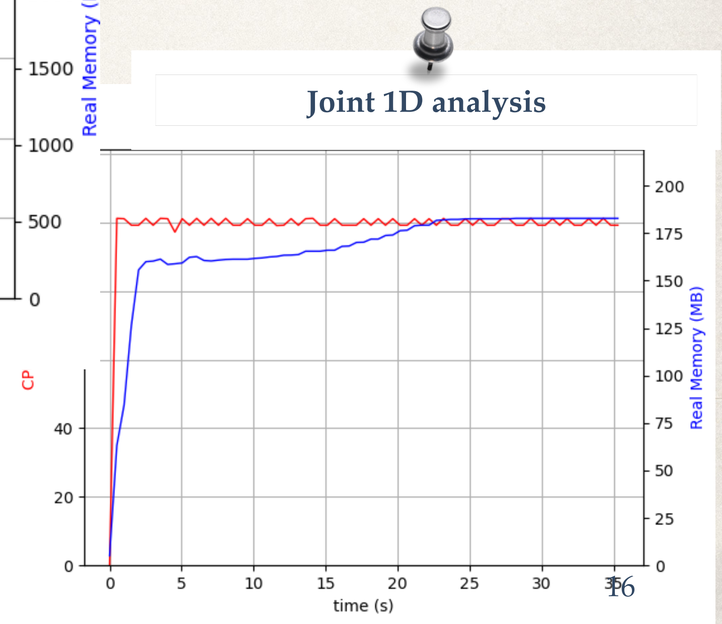
Gammapy benchmarks

- Daily automatic monitoring of memory usage and computation time for most common use cases
- Computation time across different actions
  - Loading observations
  - Binning data and IRFs
  - Fitting
  - FluxPointEstimations
- Pinpoint PR significantly affecting the performance



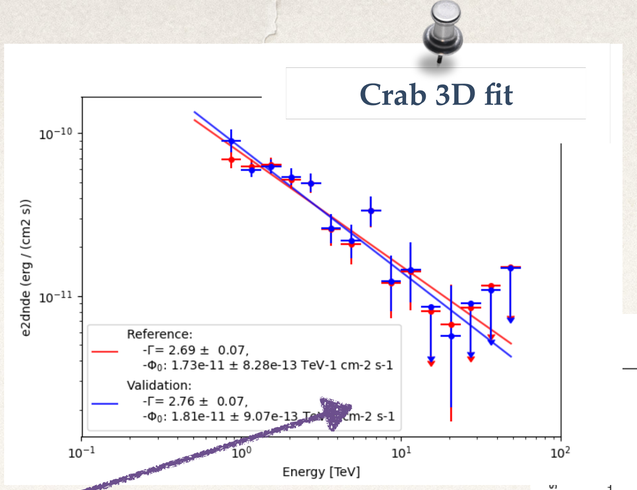


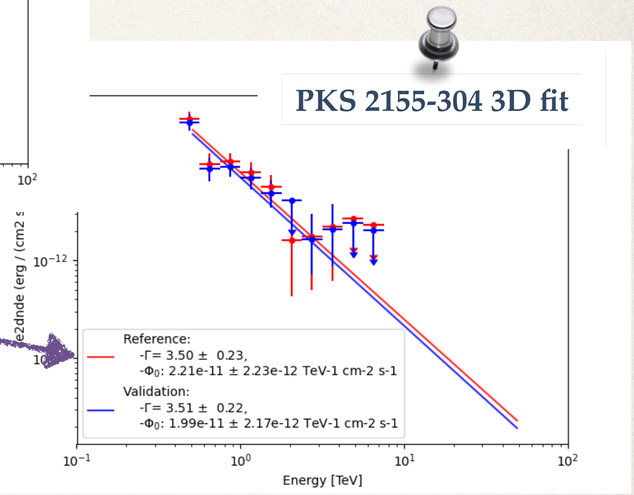


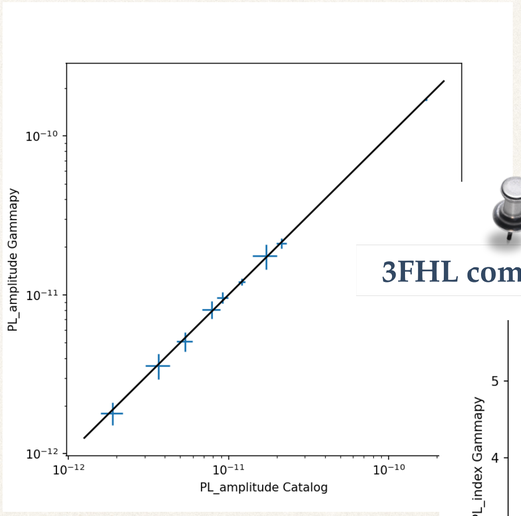


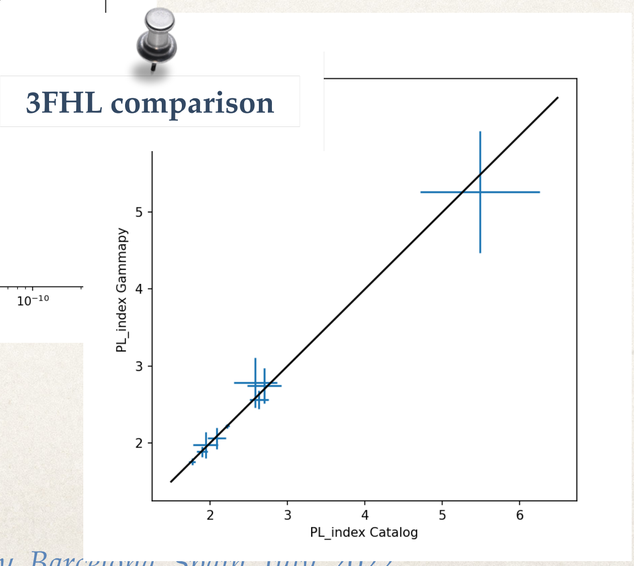
Gammapy validation

- Analysis of a list of science cases before every gammapy release
  - H.E.S.S. DL3 DR1 results
  - CTA DC1 results
  - Joint Crab validation paper
  - …
- Ensure the stability of results
- Compare against results from the Fermi ST
  - Subset of the 3FHL paper









See: https://github.com/gammapy/gammapy-benchmarks/tree/master/validation

Get in touch!!!

Gammapy at this symposium: 11+ contributions

- Follow the discussions on GitHub: https://github.com/gammapy/gammapy/discussions
- Join us on slack: gammapy.slack.com
- User calls on zoom
- Developer calls every Friday afternoon, and regular coding sprints/co-working weeks

Thanks and stay tuned!!!