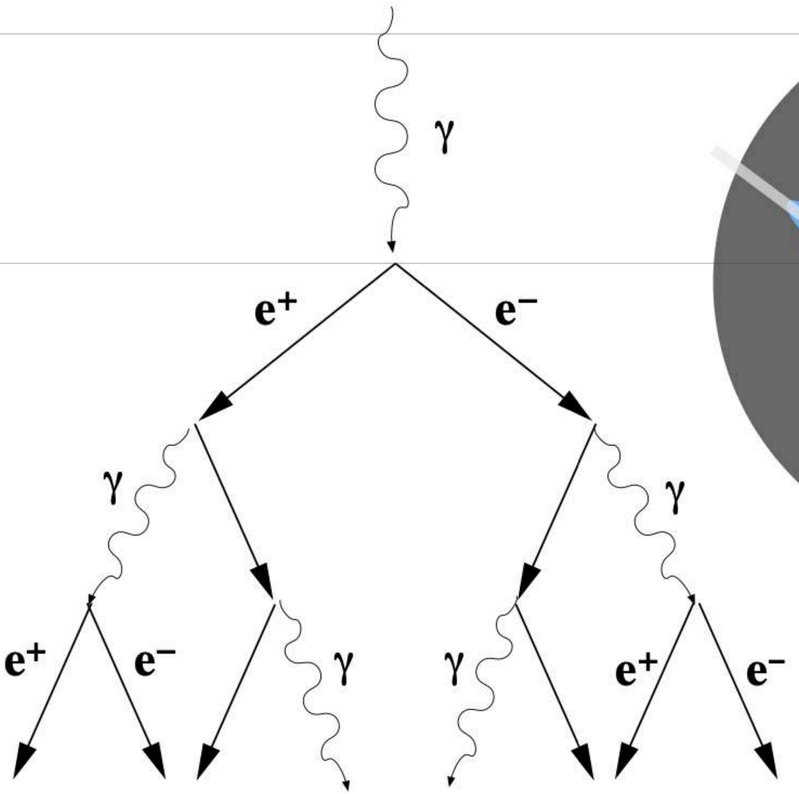
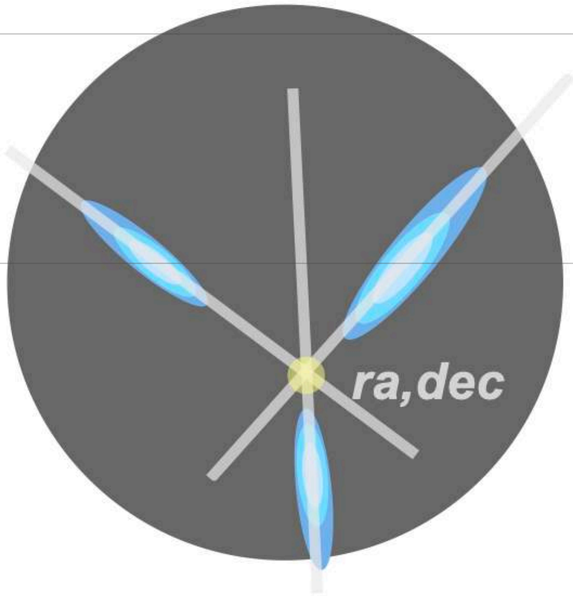
Atreyee Sinha
IPARCOS/UCM, Madrid
asinha@ucm.es

For the Gammapy-dev team

A short introduction to VHE gamma-ray detection techniques

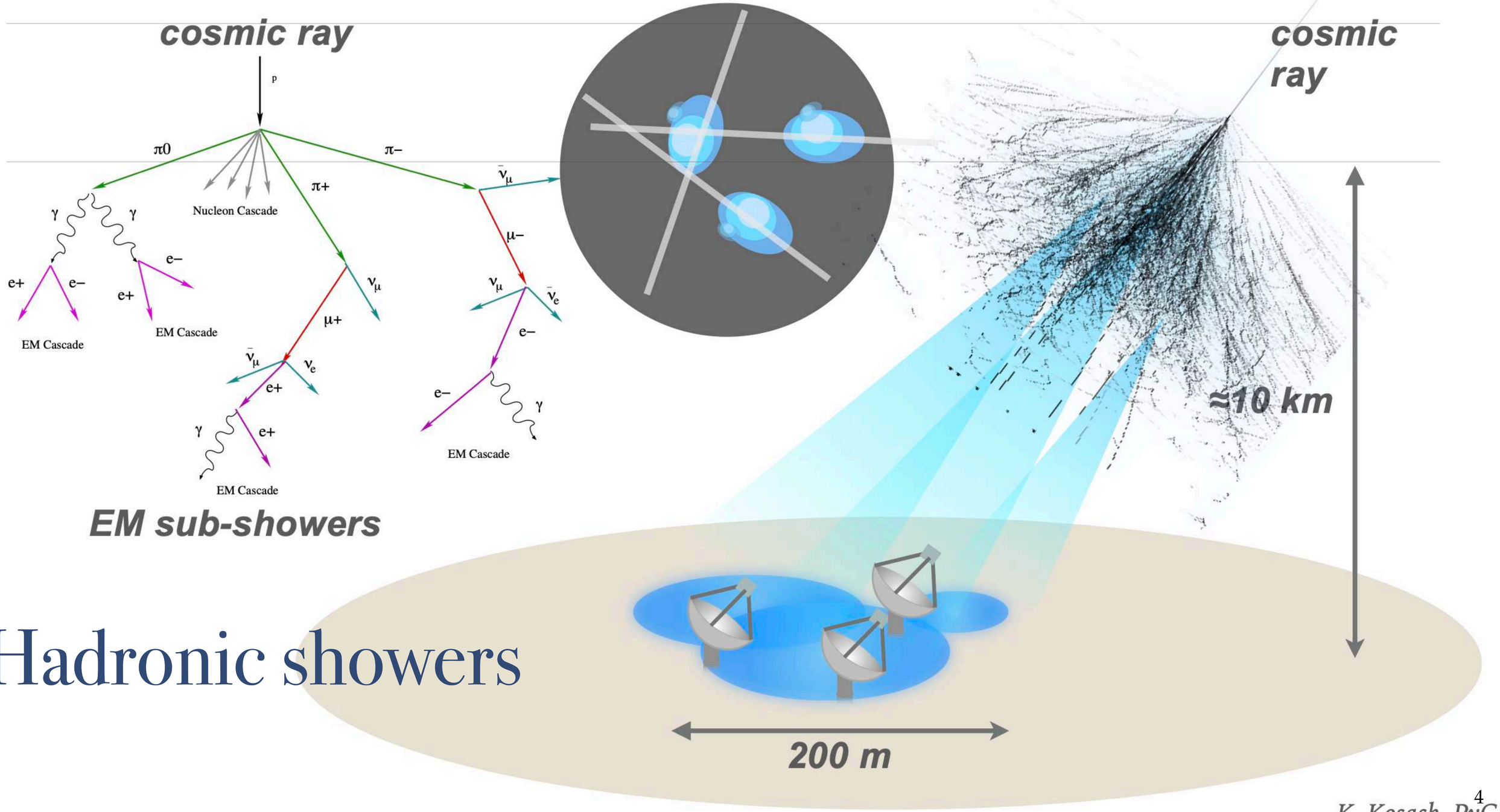




Photon showers

effective area *≈ size of light pool (small array)*

≈ size of array (large array)

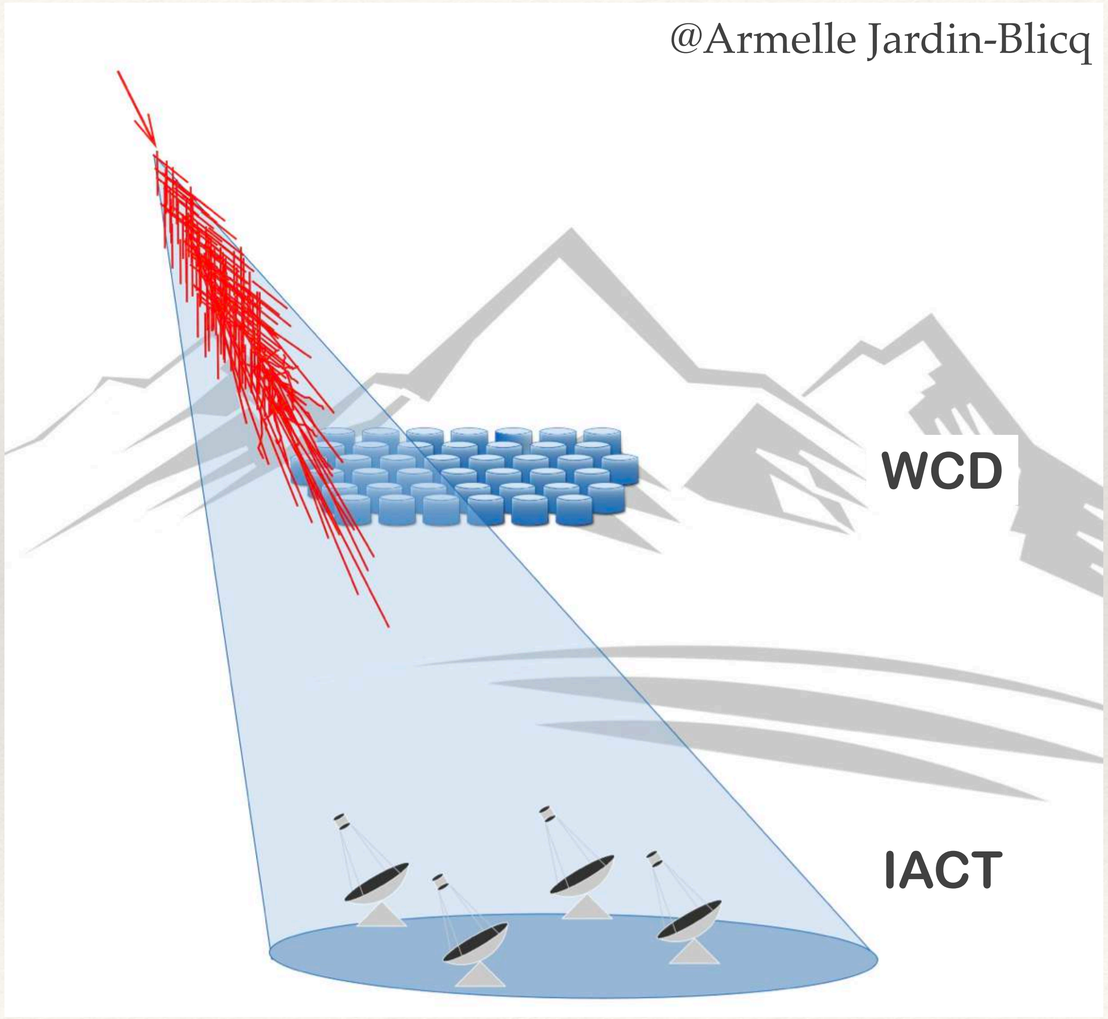


Hadronic showers

K. Kosack, PyG

Detection techniques

	IACT	WCD
Type	Pointing	All Sky
Energy threshold	~100 GeV	TeV
Energy resolution	~10%	40%
Angular Resolution	~0.1 deg	~0.5 deg
FoV	~5 deg	~ 90 deg
Duty Cycle	10%	~100%
Current instruments	H.E.S.S. , VERITAS, MAGIC, FACT	HAWC
Future Instruments	CTA	SWGO



Challenge: Atmosphere and Observation Condition

- Instrument response changes with
  - Gamma-ray **energy**
  - **Position** in Camera Field of View
  - **Zenith Angle** of observation (atmospheric thickness)
  - **Azimuth**: Earth's magnetic field orientation
  - **Telescopes triggered**
  - **Subarray choice**
  - Atmospheric Density profile
  - Optical **Night-Sky-Background** light level (Moon, Zodiacal light, Light pollution)
  - Detector Configuration
  - …

Challenge: Atmosphere and Observation Condition

- Instrument response changes with

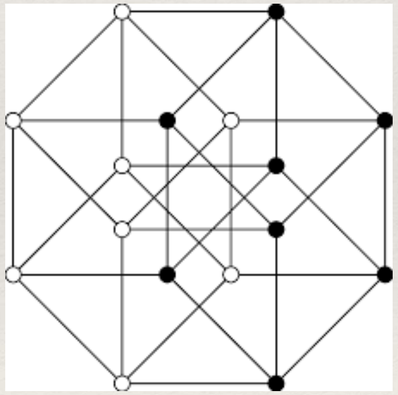
Change during an observation

- Gamma-ray **energy**
- **Position** in Camera Field of View
- **Zenith Angle** of observation (atmospheric thickness)
- **Azimuth**: Earth's magnetic field orientation
- **Telescopes triggered**

Change between observations

- **Subarray choice**
- Atmospheric Density profile
- Optical **Night-Sky-Background** light level (Moon, Zodiacal light, Light pollution)
- Detector Configuration
- …

- Potentially very high dimensional Instrument Response Functions
- Lots of custom simulations



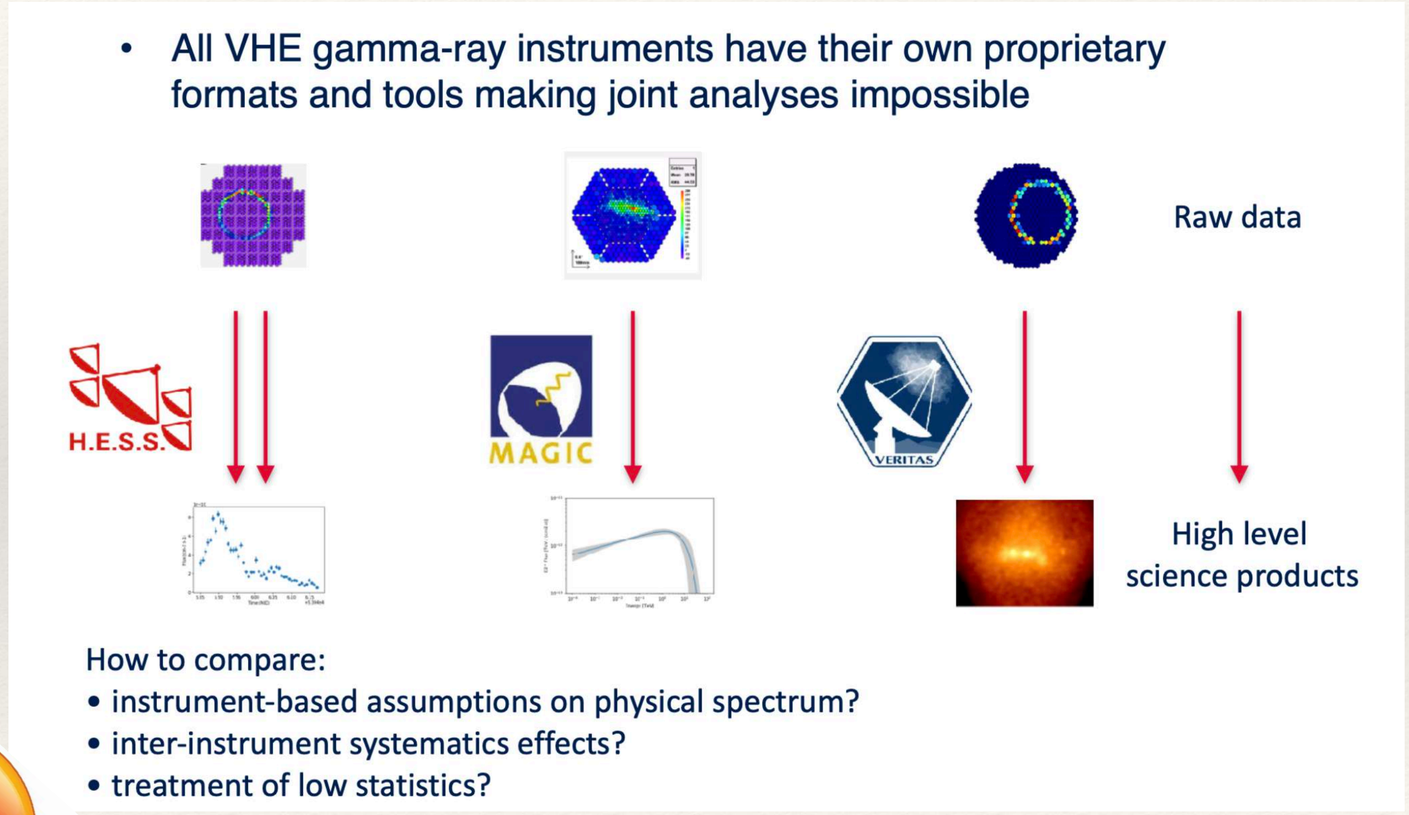
- Need good parametrisation and data model

Challenge: Strong residual hadronic background

- 10^4 hadronic triggers for 1 photon trigger
- ~98-99% rejected by present gamma-hadron separation techniques
- Still residual background is ~100 times of signal strength
  - **"90-99% of gamma-like particles are actually hadrons"**

Proprietary approaches

- Each telescope has its own data format and software
  - Multiple analysis chains within each collaboration
  - Cross checking analysis is a never-ending issue
- Combination of data from different experiments needs hacking into proprietary analysis s/w - terrible experience



Towards open VHE data analysis

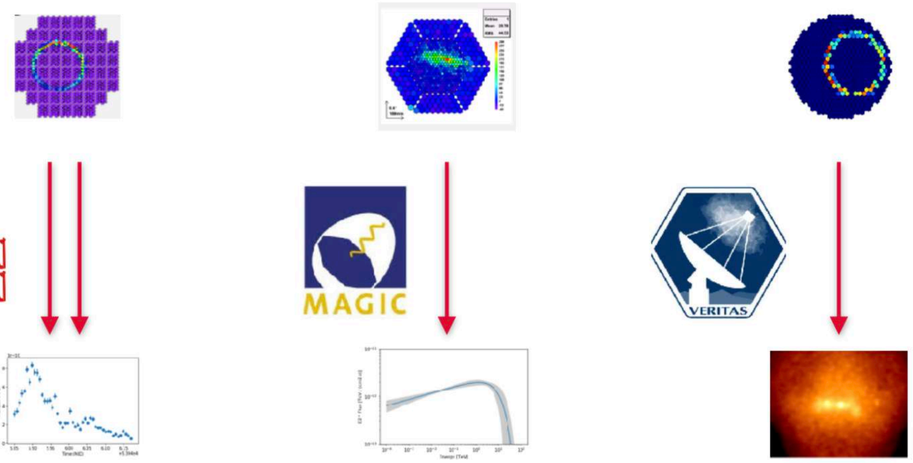
➡CTA will operate as **open observatory**

➡**Public legacy data release** of current instruments

➡Multi-instrument analysis necessary

◉VHE analysis needs common **open data formats** and **common open tools**

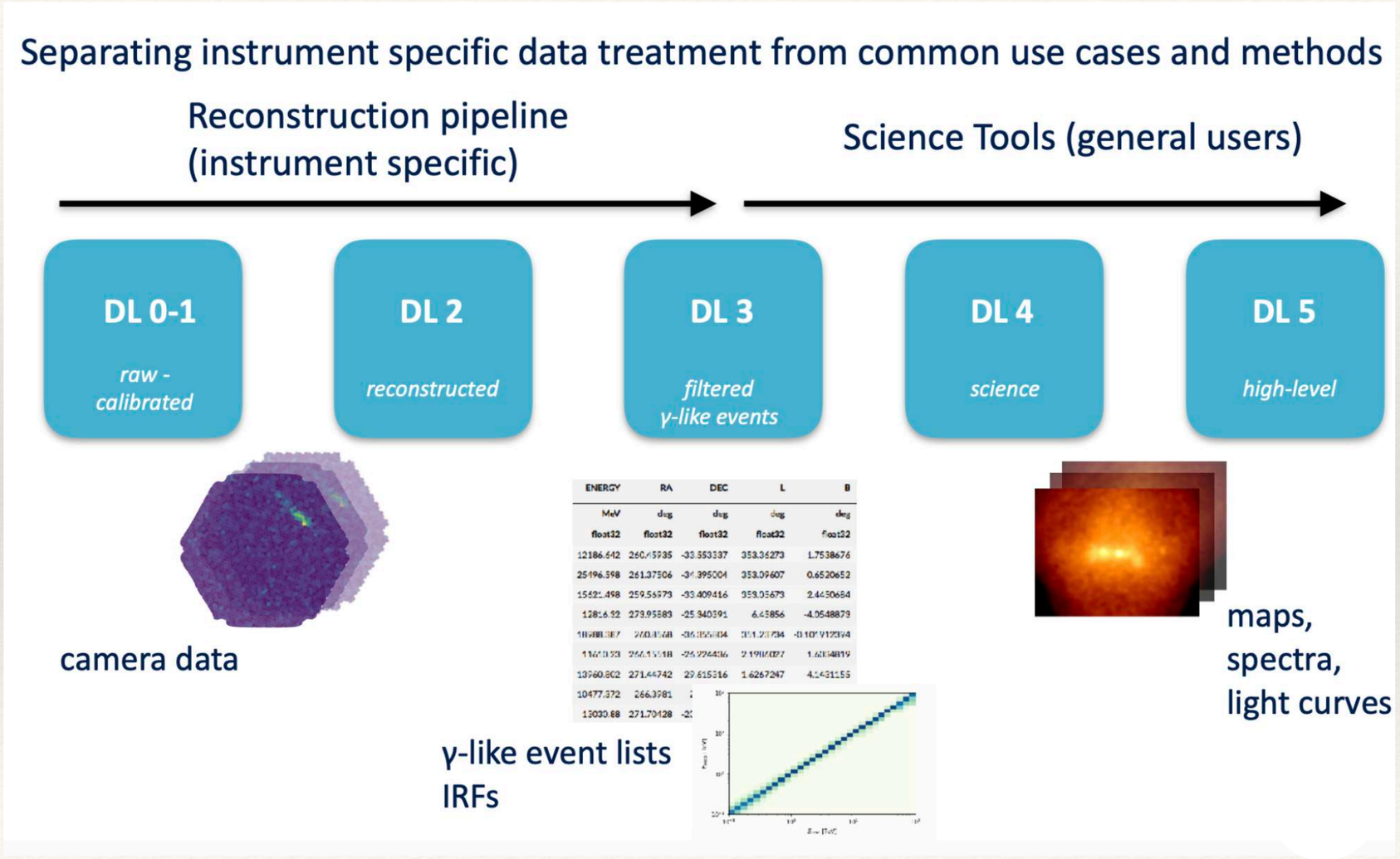
- All VHE gamma-ray instruments have their own proprietary formats and tools making joint analyses impossible



How to compare:
- instrument-based assumptions on physical spectrum?
- inter-instrument systematics effects?
- treatment of low statistics?

GammaAstroData formats & Gammapy

The GADF initiative



- Gamma Astro Data Formats (GADF)
- Based on the Fermi-LAT format Proposed in 2016
- Adopted by CTA
- H.E.S.S. DL3 DR1
- MAGIC data release

The Gammapy concept

A high level gamma-ray astronomy package based on common data formats

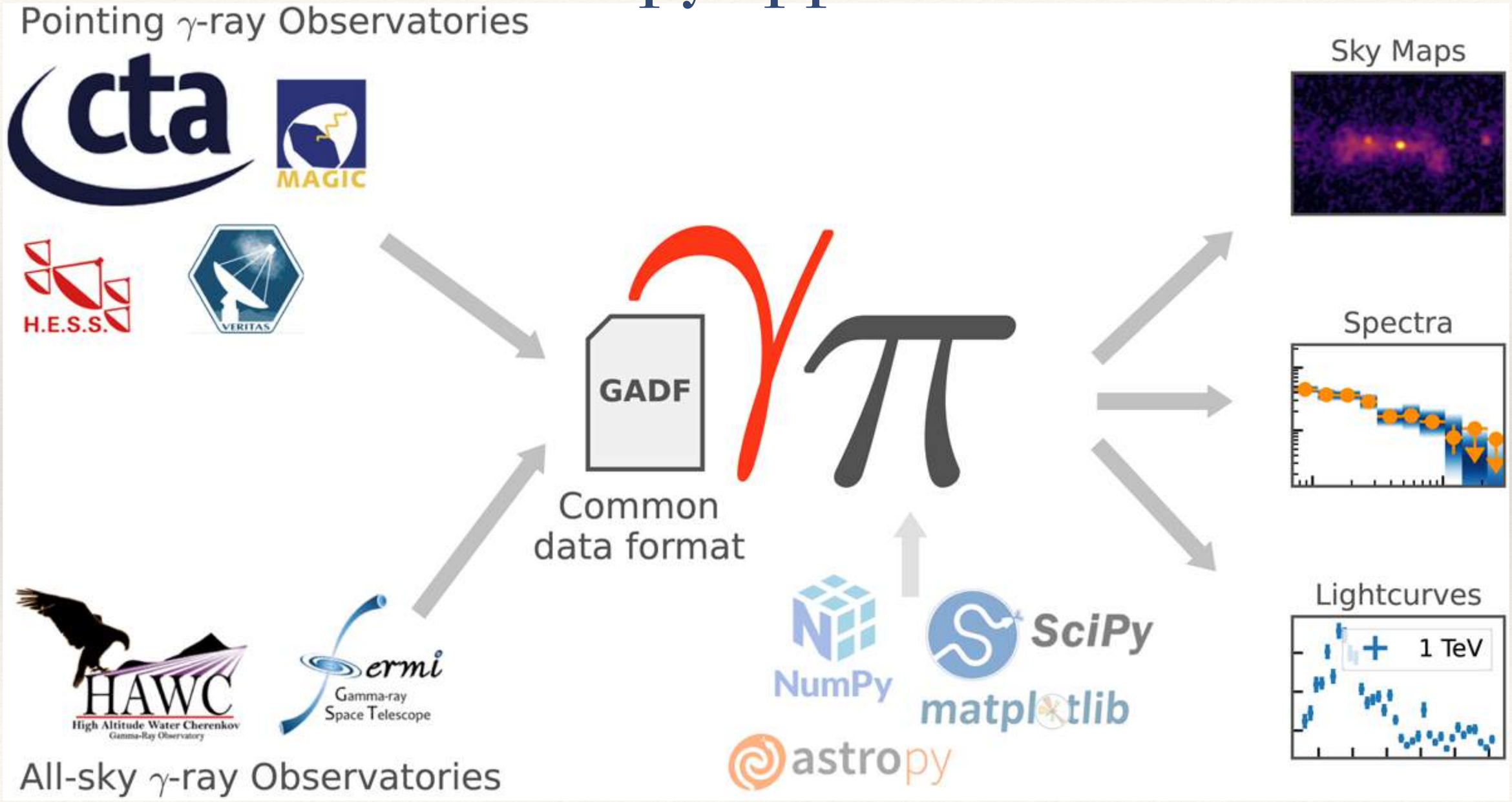
+

A flexible, open source, community driven python library

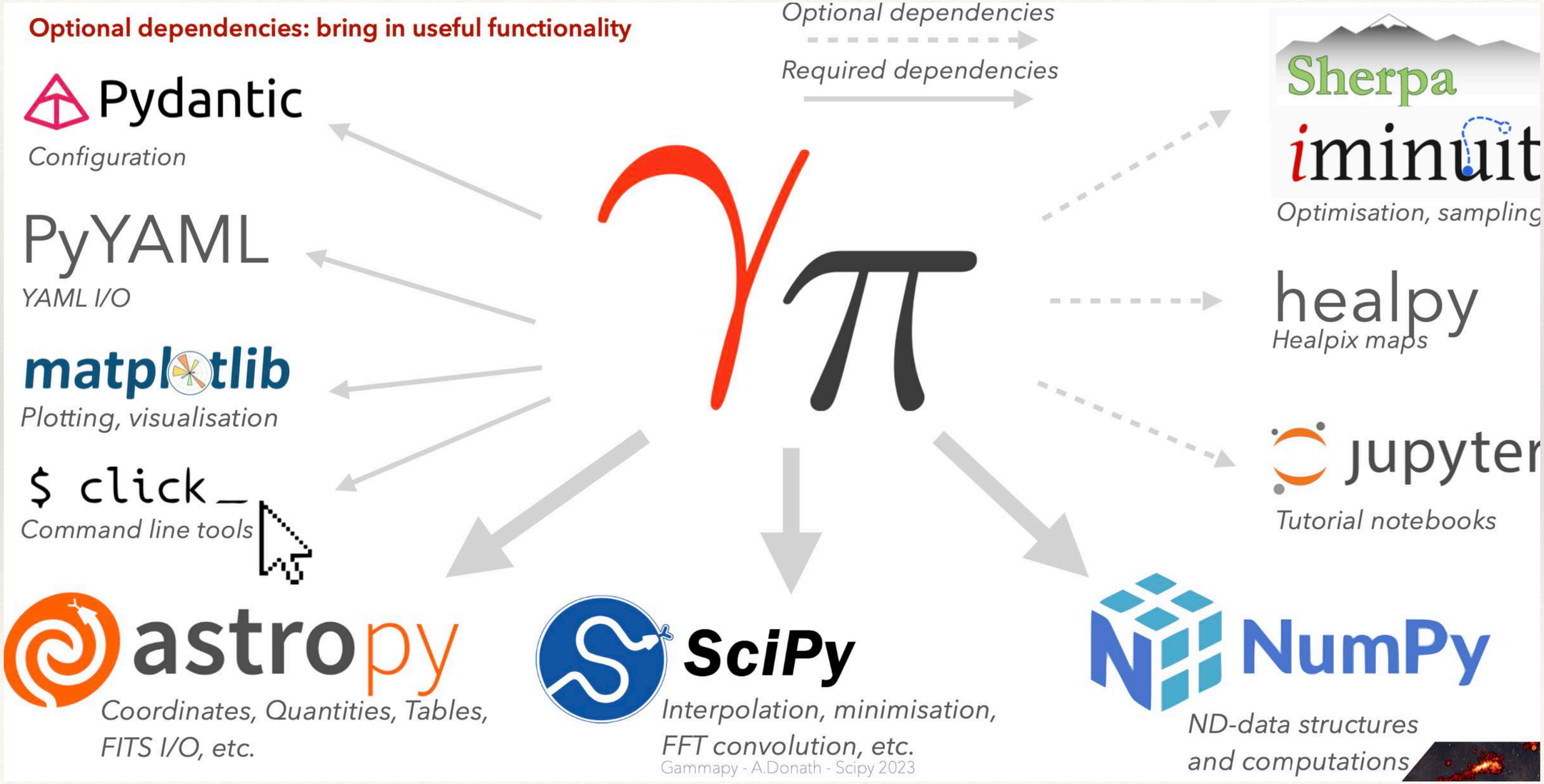
+

Science tools for the CTA

Gammapy approach

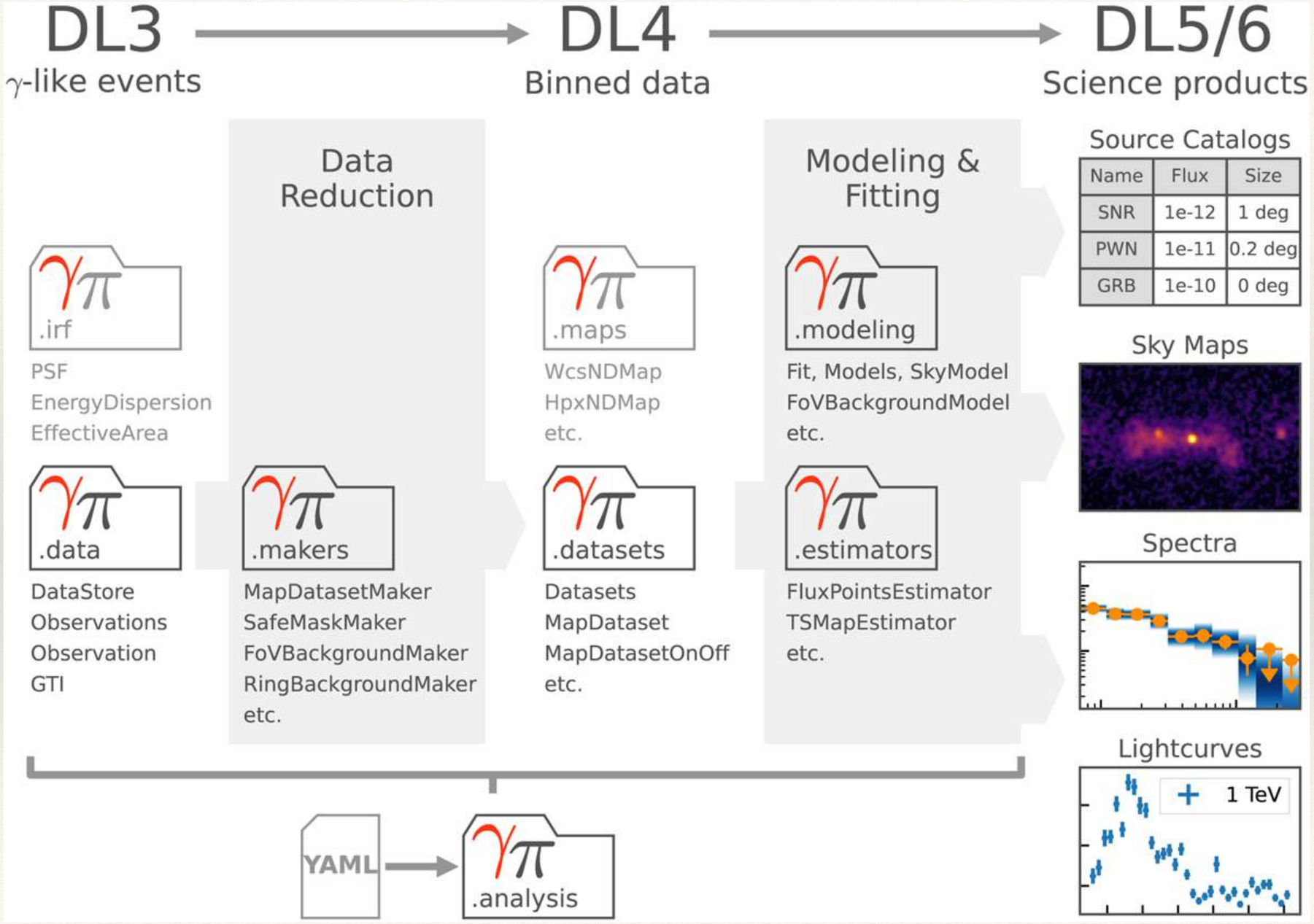


Dependencies



Internal workflow

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



Data reduction

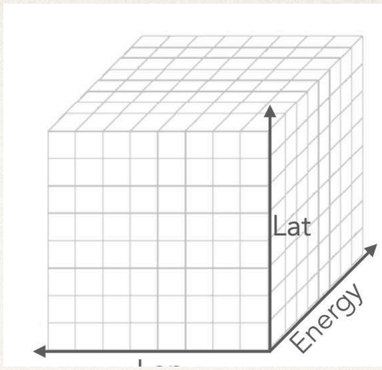
DL3 ⟶ DL4 ⇢ DL5

Dataset

- 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
...	...	...	...	...

3D analysis



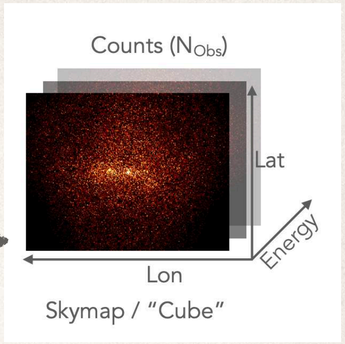
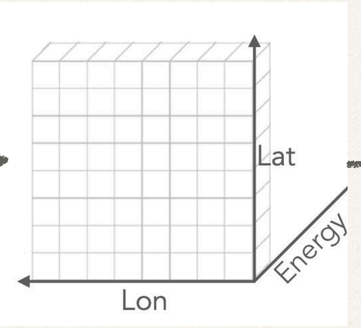
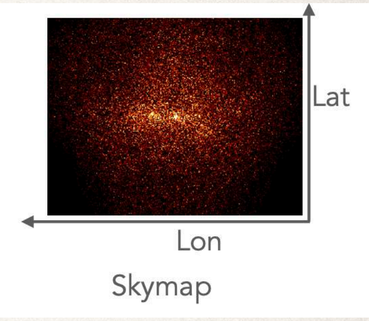
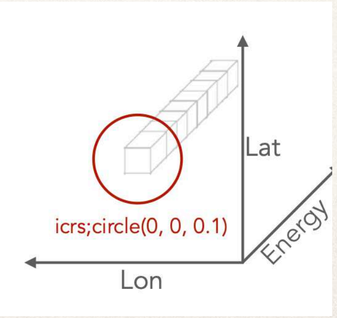


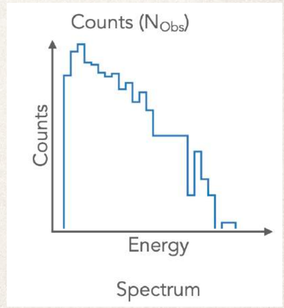
Image analysis





Spectral analysis analysis





Data fitting
DL4 ⟶ DL5

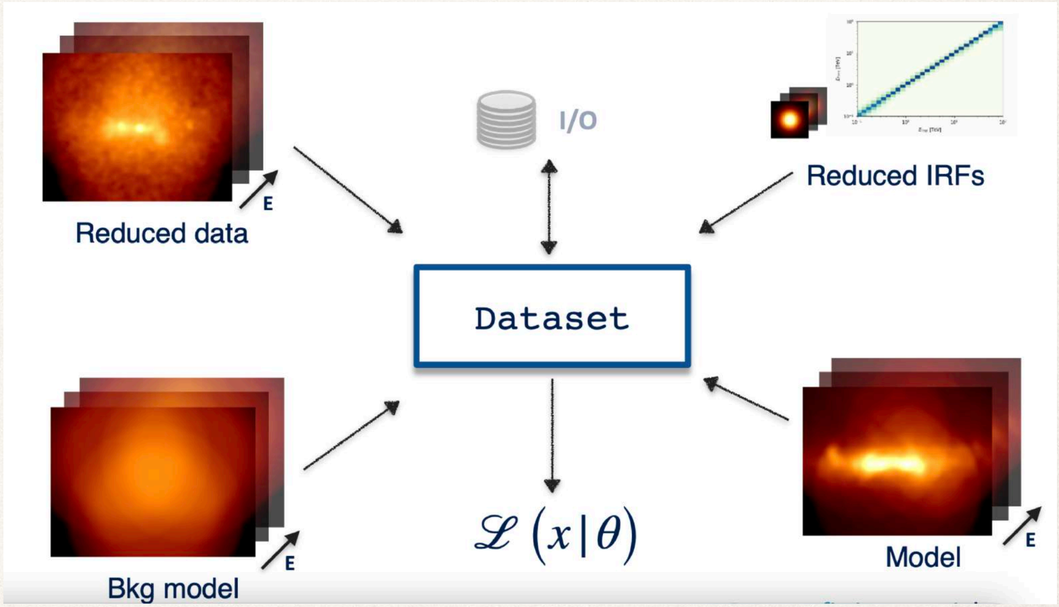
- Fitting on pre-computed datasets
  - eg: From HAWC, Fermi-LAT, OGIP files, etc
- Forward folding with maximum likelihood estimation

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

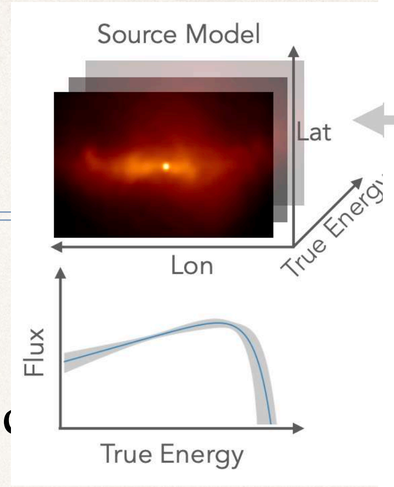
Cash: known background

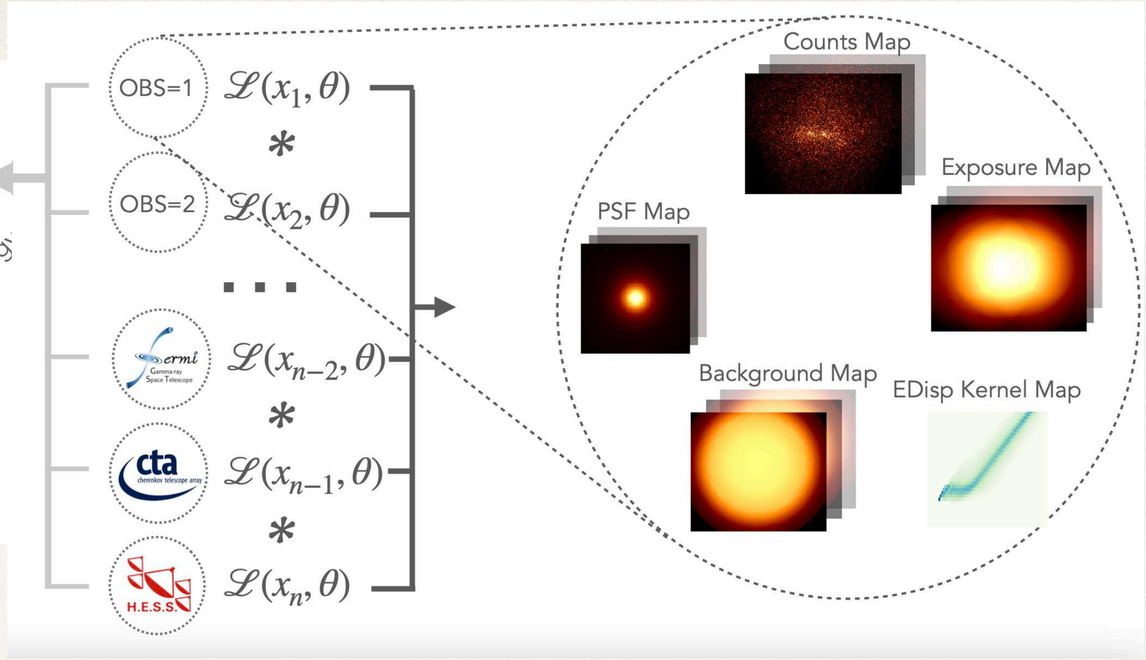
$$TS = -2\ logL = 2\Sigma\ (\ N * logN_{pred} - N_{pred}\ )$$

Wstat: counts with measured background

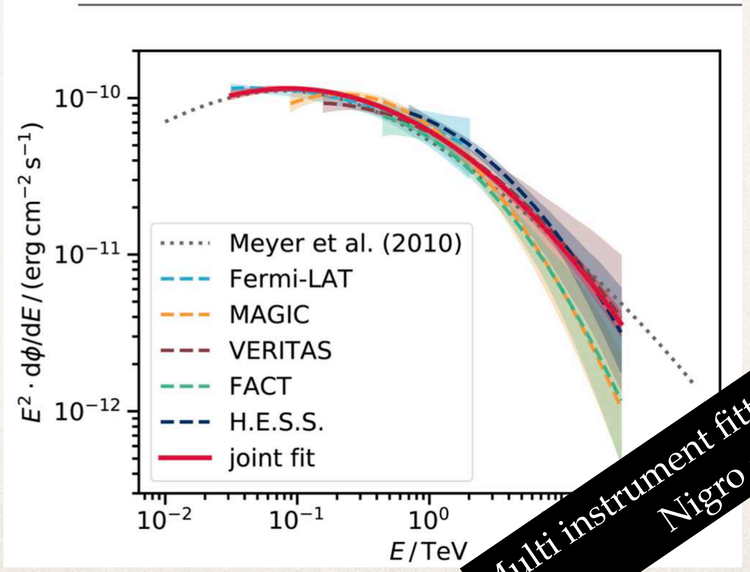


Joint likelihood





- Simultaneous fitting of various d
- 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

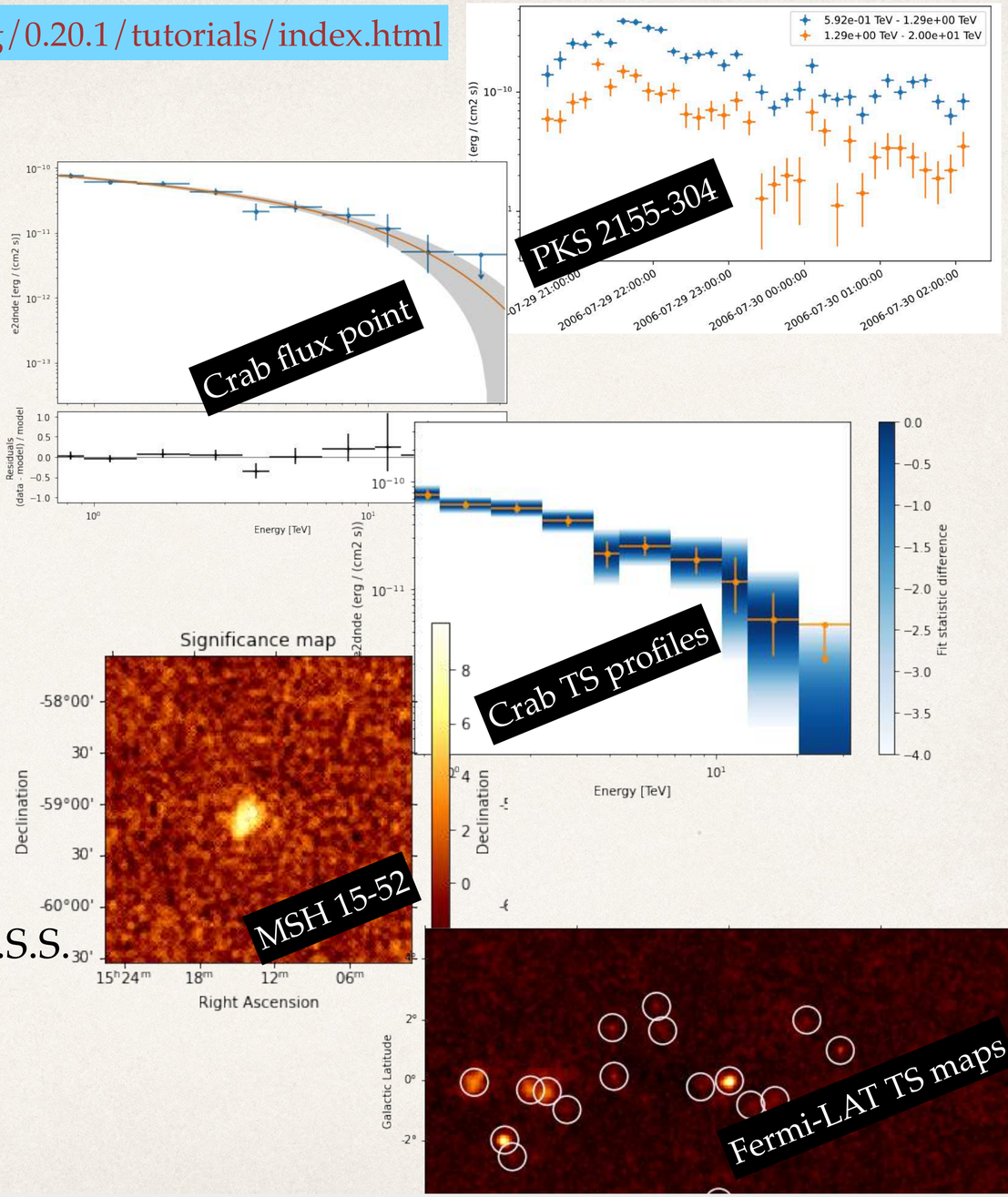


Multi instrument fitting of Crab Nebula
Nigro et al, 2019

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…



API & SubPackages

gammapy.data

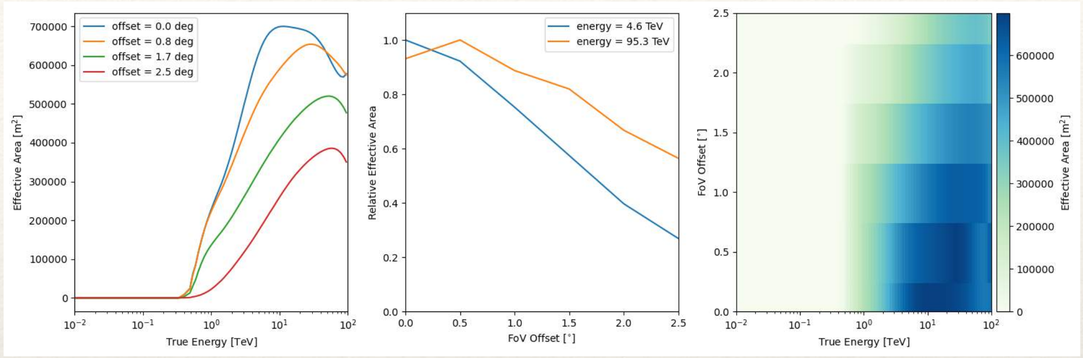
- Select, read and represent DL3 data in memory
- Compliant with GADF v0.2, v0.3
- `Observation`
  - `EventList`
  - Associated IRFs
- hdu-index-table: Link event list to associated IRFs
- obs-index-table: Selecting observation

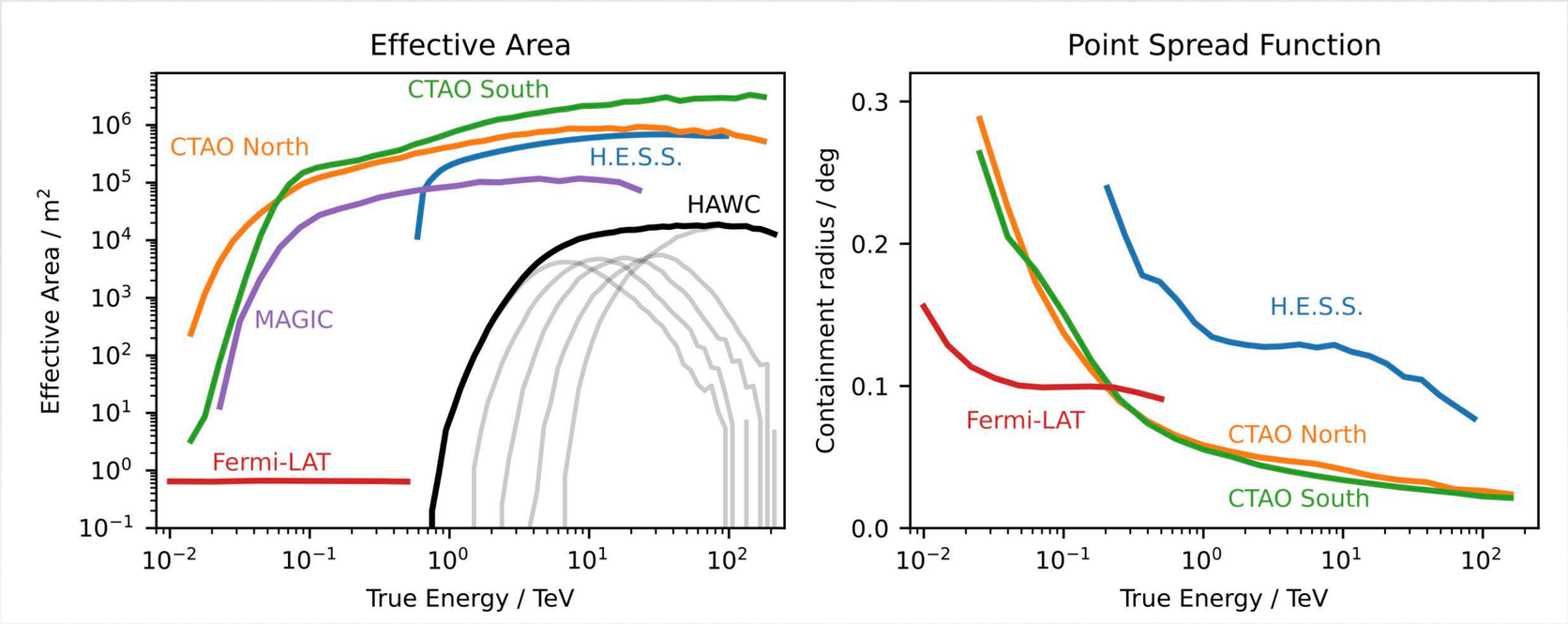
```python
from gammapy.data import DataStore

data_store = DataStore.from_dir(
    base_dir="$GAMMAPY_DATA/hess-dl3-dr1"
)

obs_ids = [23523, 23526, 23559, 23592]

observations = data_store.get_observations(
    obs_id=obs_ids, skip_missing=True
)

for obs in observations:
    print(f"Observation id: {obs.obs_id}")
    print(f"N events: {len(obs.events.table)}")
    print(f"Max. area: {obs.aeff.quantity.max()}")
```

OBS_ID	RA_PNT	DEC_PNT	...	TARGET_OFFSET	SAFE_ENERGY_LO	SAFE_ENERGY_HI
	deg	deg	...	deg	TeV	TeV
20136	228.6125	-58.771667	...	0.38821736	0.40738028	100.0
20137	228.6125	-59.771667	...	0.6156251	0.40738028	100.0
20151	228.6125	-58.771667	...	0.38821736	0.40738028	100.0
20275	187.27792	2.552389	...	--	0.33113113	100.0
20282	228.6125	-58.771667	...	0.38821736	0.40738028	100.0
20283	228.6125	-59.771667	...	0.6156251	0.3801894	100.0
...	...	...	...	...	...	...
33801	330.29538	-30.225555	...	0.4998021	0.8128305	100.0
47802	330.29538	-30.225555	...	0.4998021	0.61659503	100.0
47803	329.13797	-30.225555	...	0.5002569	0.43651584	100.0
47804	329.71667	-29.725555	...	0.500033	0.40738028	100.0
47827	330.29538	-30.225555	...	0.4998021	0.61659503	100.0
47828	329.13797	-30.225555	...	0.5002569	0.43651584	100.0
47829	329.71667	-30.725555	...	0.49996707	0.3801894	100.0

gammapy.irfs

- Storage container for DL3 & DL4 IRFs
- Interpolation, evaluation, serialisation
- Effective area
- PSF
- Energy dispersion
- Background



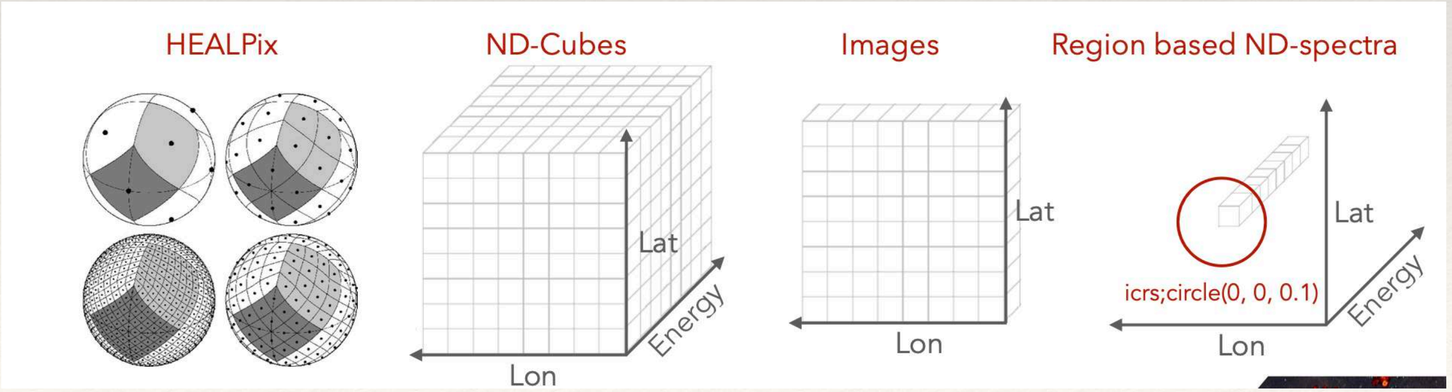


gammapy.maps

Internal dependency of Fermipy

- N-**dimensional coordinate aware data structures** for storing gamma-ray data, with arbitrary number of non-spatial dimensions
- Uniform API for WCS, HEALPix and region based pixelization schemes
- MapAxis - Contiguous axis
- TimeMapAxis - Non-contiguous axis
- LabelMapAxis - Independent values
- Bin edges/centers, interpolation schemes

Generalised container for data, Instrument Response, template models, etc…



gammapy.makers

- Reprojection of counts and IRFs
- Background estimation
  - A **run wise** estimation of the background necessary
  - Different methods supported
    - Reflected Background
    - Ring Background
    - **Field of View Background**

Traditional methods:
Measured off counts, WSTAT statistic

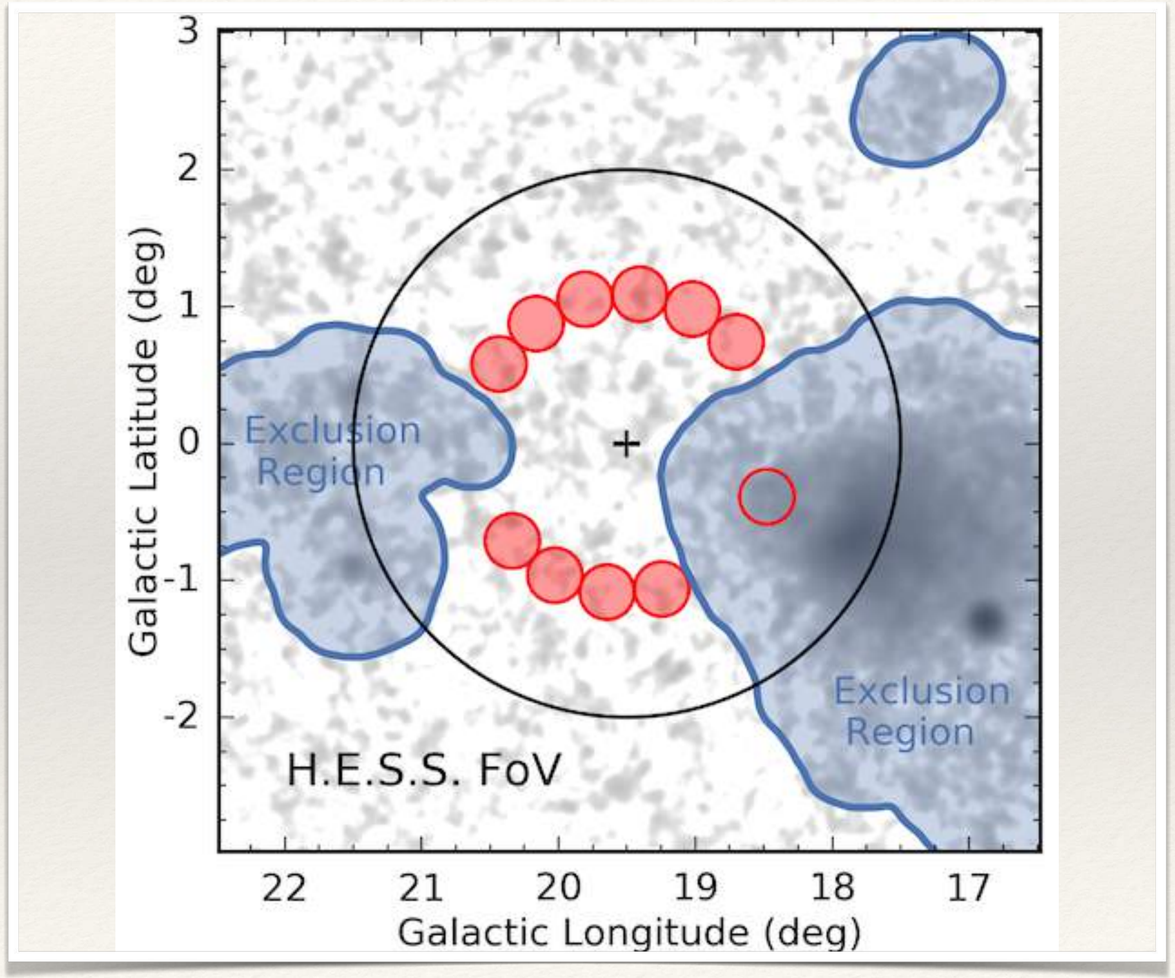
Novel implementation:
Background modelled simultaneous with source, Cash statistic

Reflected Background

`ReflectedRegionsBackgroundMaker`

- Assumption: Background is approximately purely radial in the field-of-view
- Valid of observations taken in Wobble Method
- A set of OFF counts is found in the observation, by rotating the ON region selected around the pointing position
- Valid for Spectral Analysis
- Can be adapted to use off regions in phase

`PhaseBackgroundMaker`

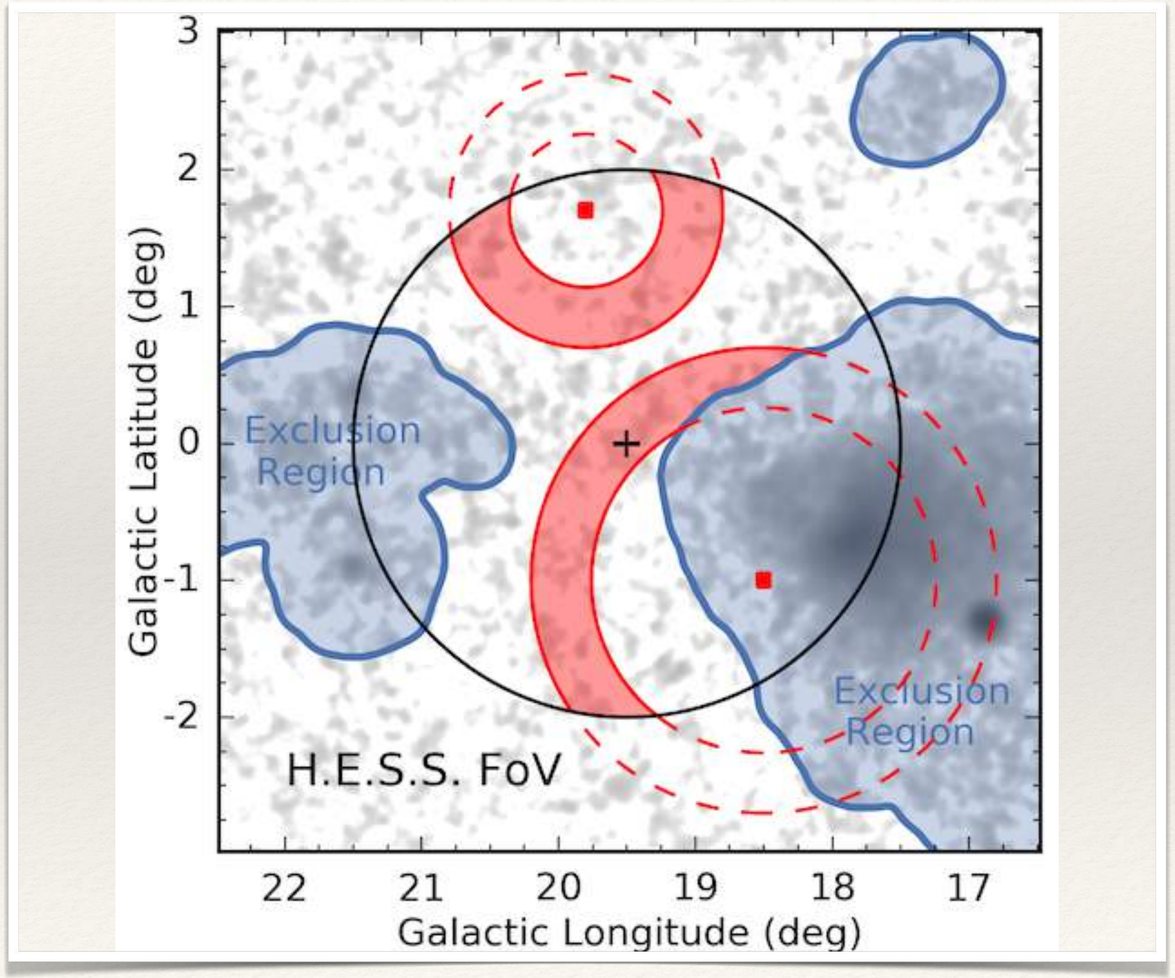


Ring Background

`RingBackgroundMaker`

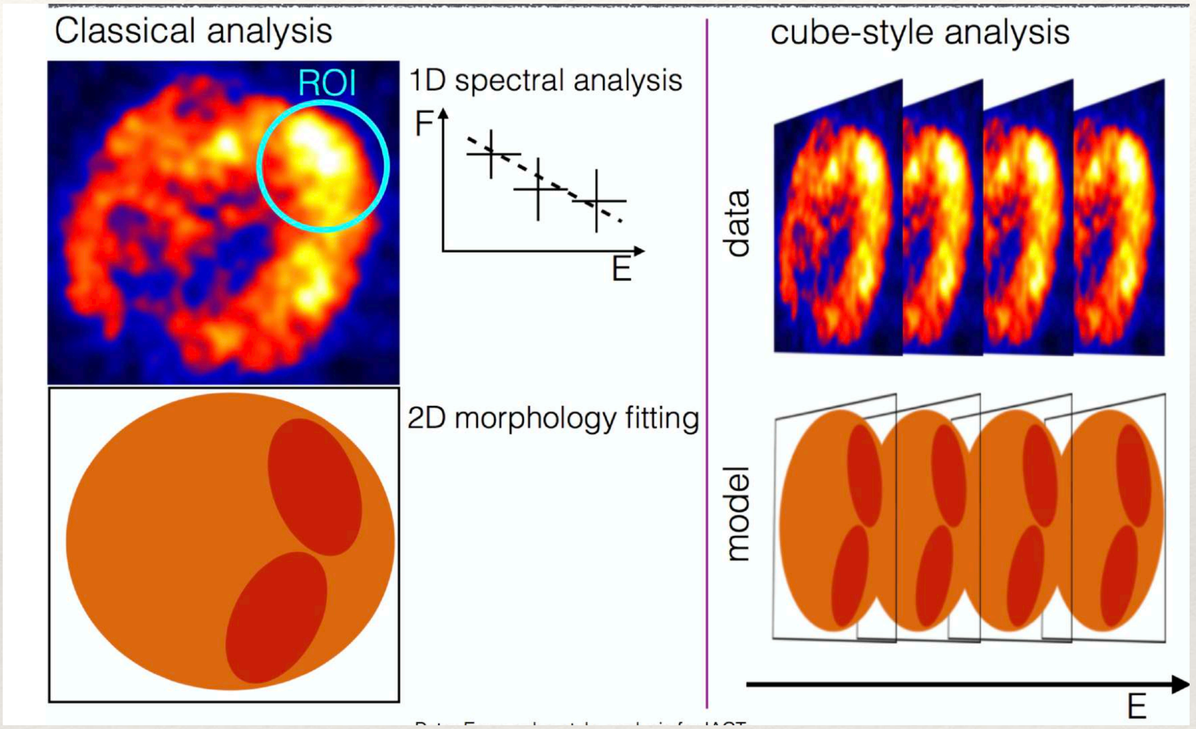
- Valid for classical 2D image analysis
- For a given pixel, OFF counts are estimated from a ring centered on the test position
- Acceptance model is necessary
- Variation - adaptive ring

`AdaptiveRingBackgroundMaker`



3D FoV Background

- A 3D likelihood analysis much more powerful than classical analysis
  - M***orphology*** and ***spectrum*** modelled simultaneously
  - Full ***multi-dimensional instrument response*** (PSF, ARF,
    EDISP) correctly taken into account
  - S***ensitivity gain*** for point-like and extended sources
  - Separation of multiple source components in crowded regions
  - Analysis of ***FoV-scale emission*** through background modelling instead of background subtraction



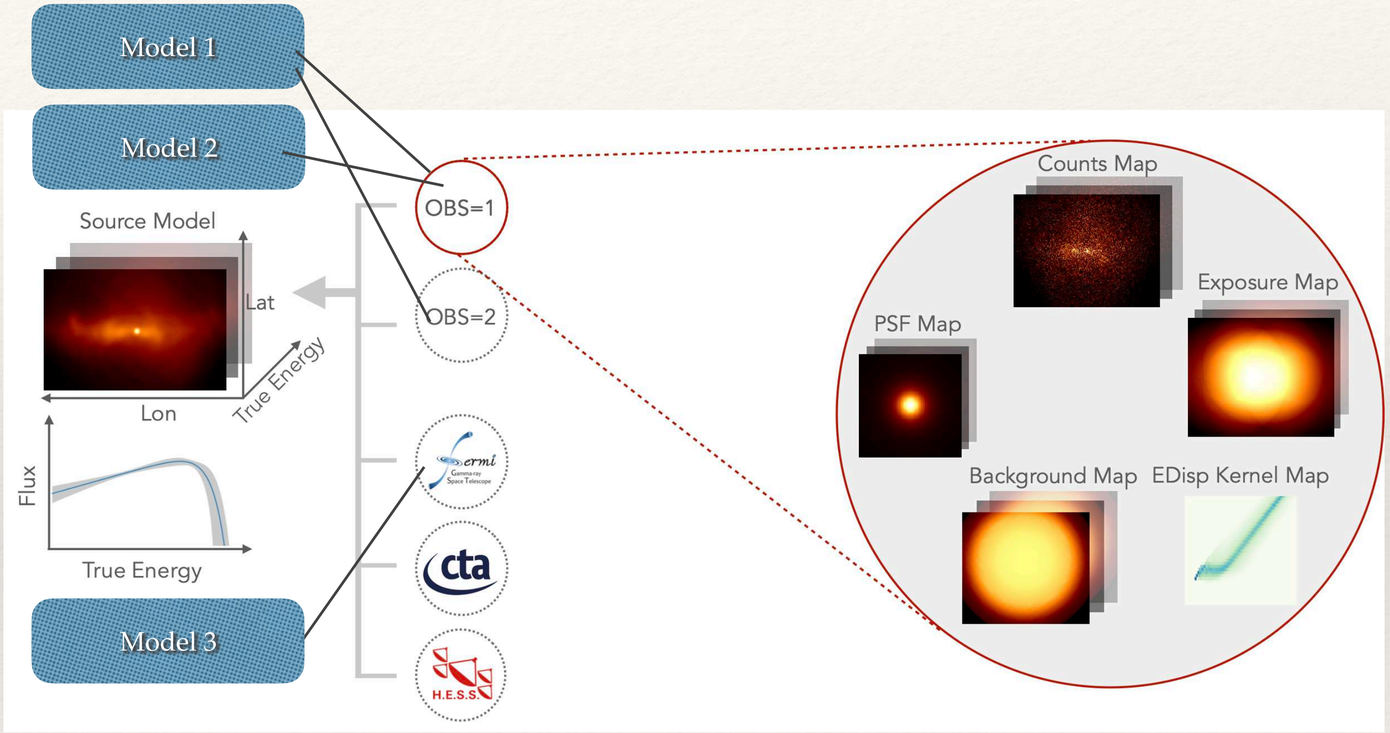
Standard analysis technique for Fermi-LAT

Complicated with IACT due to atmospheric uncertainties

`FoVBackgroundMaker`

gammapy.datasets

- Bundles binned data models & likelihood function
  - Sharing of models across datasets
- Interface to the Fit class
- Different types of datasets based upon analysis type & statistic
- Joint fitting between same/different types of datasets

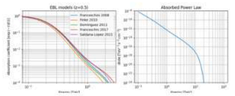


For creating Fermi-LAT datasets, see:
https://docs.gammapy.org/1.2/tutorials/data/fermi_lat.html#sphx-glr-tutorials-data-fermi-lat-py

An easier-to-use prototype: https://github.com/adonath/snakemake-workflow-fermi-lat

gammapy.modeling

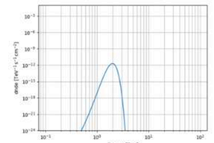
Spectral models



EBL absorbption spectral model

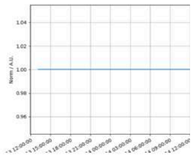
This model takes a constant value along the spectral range.

Constant spectral model

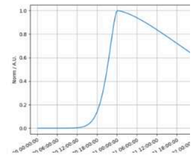


Gaussian spectral model

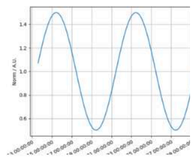
Temporal models



Constant temporal m

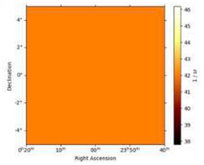


Generalized Gaussian tempora

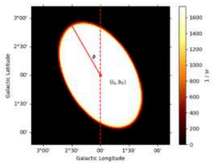


Sine temporal moc

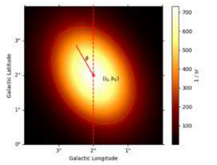
Spatial models



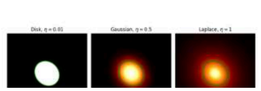
Constant spatial model



Disk spatial model

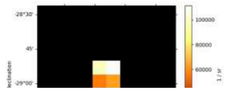


Gaussian spatial model

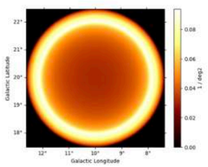


Generalized gaussian spatial model

Piecewise norm spatial model



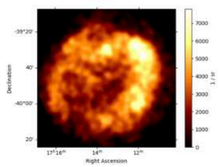
Point spatial model



Shell spatial model

This is a spatial model parametrizing a projected radiating shell.

Shell2 spatial model



Template spatial model

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

ALL models serialised to YAML format

```
components:
-   name: pwl-gauss-model
    type: SkyModel
    spectral:
        type: PowerLawSpectralModel
        parameters:
        -   name: index
                                                    ude
                                    value: 1.0e-12
            unit: cm-2 s-1 TeV-1
        -   name: reference
            value: 1.0
            unit: TeV
    spatial:
        type: GaussianSpatialModel
        frame: icrs
        parameters:
        -   name: lon_0
            value: 0.0
            unit: deg
        -   name: lat_0
            value: 0.0
            unit: deg
```

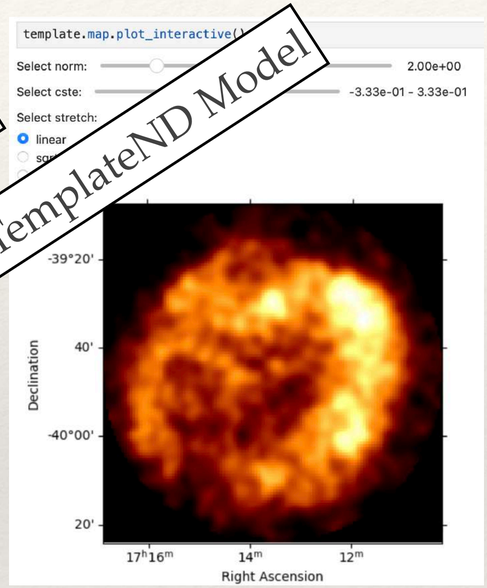
See Model Gallery: https://docs.gammapy.org/1.2/user-guide/model-gallery/index.html

Support for user contributed models

```python
from astropy.coordinates.angle_utilities import angular_separation
from gammapy.modeling.models import SpatialModel


class MyCustomGaussianModel(SpatialModel):
    """My custom Energy Dependent Gaussian model.

    Parameters
    ----------
    lon_0, lat_0 : `~astropy.coordinates.Angle`
        Center position
    sigma_1TeV : `~astropy.coordinates.Angle`
        Width of the Gaussian at 1 TeV
    sigma_10TeV : `~astropy.coordinates.Angle`
        Width of the Gaussian at 10 TeV

    """

    tag = "MyCustomGaussianModel"
    is_energy_dependent = True
    lon_0 = Parameter("lon_0", "0 deg")
    lat_0 = Parameter("lat_0", "0 deg", min=-90, max=90)

    sigma_1TeV = Parameter("sigma_1TeV", "2.0 deg", min=0)
    sigma_10TeV = Parameter("sigma_10TeV", "0.2 deg", min=0)

    @staticmethod
    def evaluate(lon, lat, energy, lon_0, lat_0, sigma_1TeV, sigma_10TeV):

        sep = angular_separation(lon, lat, lon_0, lat_0)

        # Compute sigma for the given energy using linear interpolation in log en
        sigma_nodes = u.Quantity([sigma_1TeV, sigma_10TeV])
        energy_nodes = [1, 10] * u.TeV
        log_s = np.log(sigma_nodes.to("deg").value)
        log_en = np.log(energy_nodes.to("TeV").value)
        log_e = np.log(energy.to("TeV").value)
        sigma = np.exp(np.interp(log_e, log_en, log_s)) * u.deg

        exponent = -0.5 * (sep / sigma) ** 2
        norm = 1 / (2 * np.pi * sigma**2)
        return norm * np.exp(exponent)
```

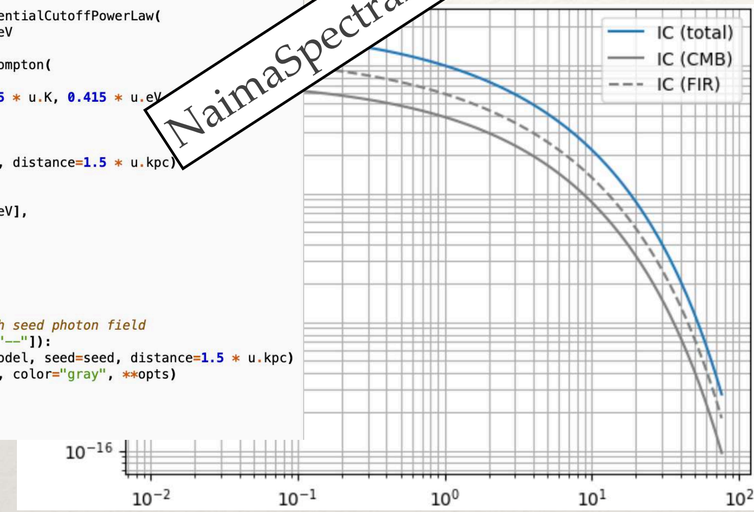
An analytic energy dependent Gaussian model

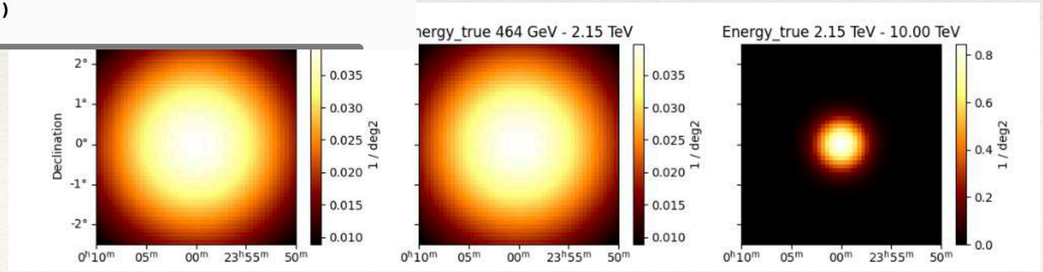
A TemplateND Model



NaimaSpectralModel

```python
from astropy import units as u
import matplotlib.pyplot as plt
import naima
from gammapy.modeling.models import Models, NaimaSpectralModel, SkyModel

particle_distribution = naima.models.ExponentialCutoffPowerLaw(
    1e30 / u.eV, 10 * u.TeV, 3.0, 30 * u.TeV
)
radiative_model = naima.radiative.InverseCompton(
    particle_distribution,
    seed_photon_fields=["CMB", ["FIR", 26.5 * u.K, 0.415 * u.eV]],
    Eemin=100 * u.GeV,
)

model = NaimaSpectralModel(radiative_model, distance=1.5 * u.kpc)

opts = {
    "energy_bounds": [10 * u.GeV, 80 * u.TeV],
    "sed_type": "e2dnde",
}

# Plot the total inverse Compton emission
model.plot(label="IC (total)", **opts)

# Plot the separate contributions from each seed photon field
for seed, ls in zip(["CMB", "FIR"], ["-", "--"]):
    model = NaimaSpectralModel(radiative_model, seed=seed, distance=1.5 * u.kpc)
    model.plot(label=f"IC ({seed})", ls=ls, color="gray", **opts)

plt.legend(loc="best")
plt.grid(which="both")
```

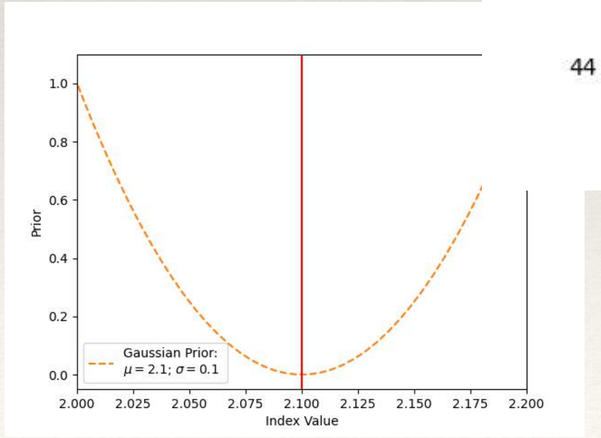


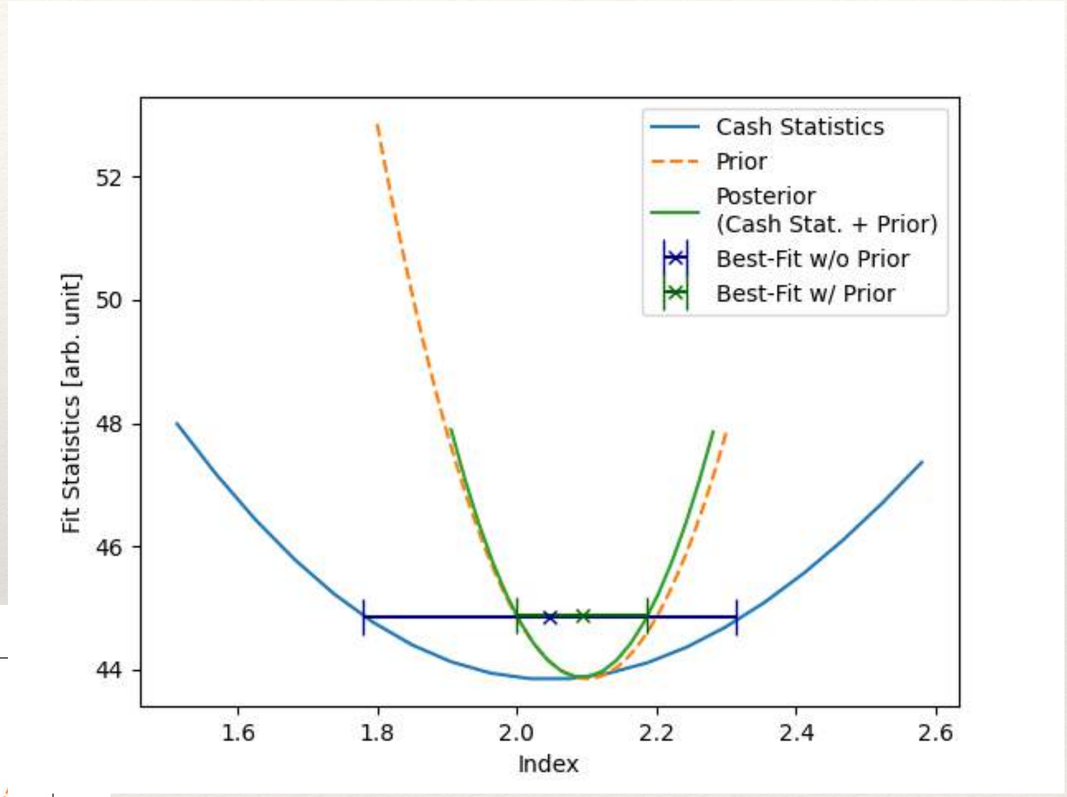


- Analytic models
- N-dim template models
- Wrappers over `naima` models

Adding priors on Parameters

- Prior: A probability density function of the model parameters
- Includes information about the parameters
- Added to the fit statistic to get the Posterior
- Possible to add Custom priors

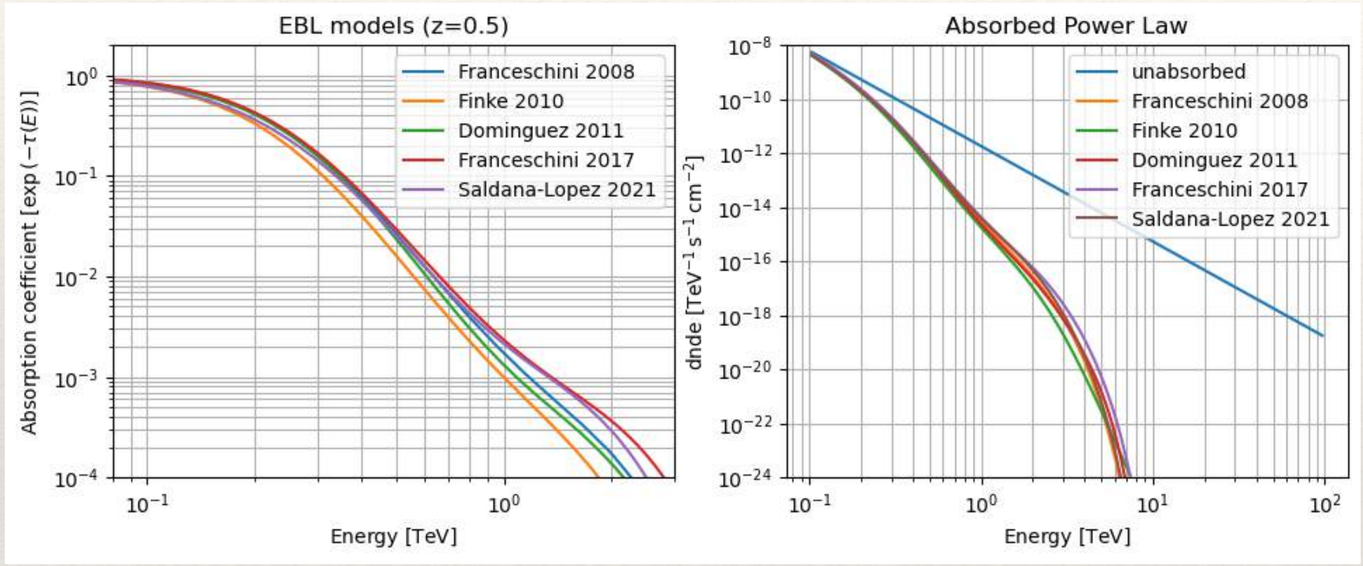




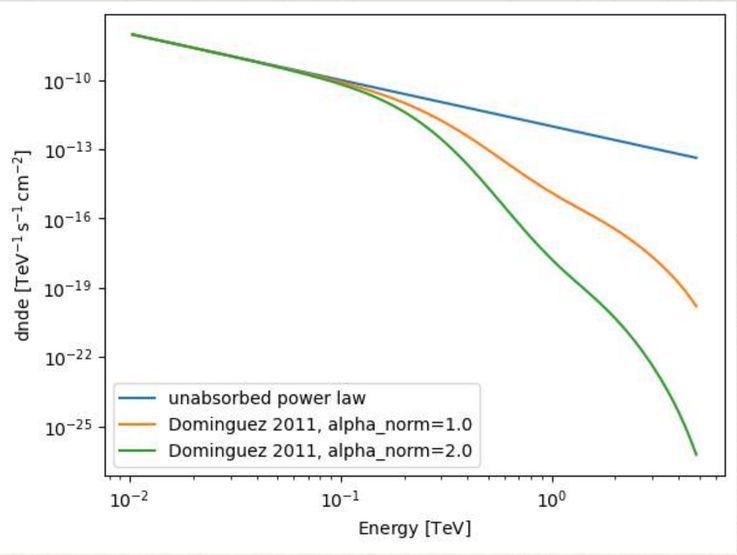
Example with a Gaussian prior

Built in EBL models

https://docs.gammapy.org/1.2/user-guide/model-gallery/spectral/plot_absorbed.html#absorption-spectral-model




```
: from gammapy.modeling.models import EBLAbsorptionNormSpectralModel, PowerLawSpectralModel
pwl = PowerLawSpectralModel()
pwl.plot(energy_bounds=[0.01, 5] * u.TeV, label="unabsorbed power law")
redshift = 0.5
dominguez = EBLAbsorptionNormSpectralModel.read_builtin("dominguez", redshift=redshift)
ax= (pwl*dominguez).plot(energy_bounds=[0.01, 5] * u.TeV, label="Dominguez 2011, alpha_norm=1.0")
dominguez.alpha_norm.value=2.0
(pwl*dominguez).plot(ax=ax, energy_bounds=[0.01, 5] * u.TeV, label="Dominguez 2011, alpha_norm=2.0")
plt.legend()
```



$$\exp\left(-\alpha \times \tau(E,z)\right)$$

Scalable norm

Predicted optical depth

- As NormSpectralModels correction
- Many models inbuilt

gammapy.estimators

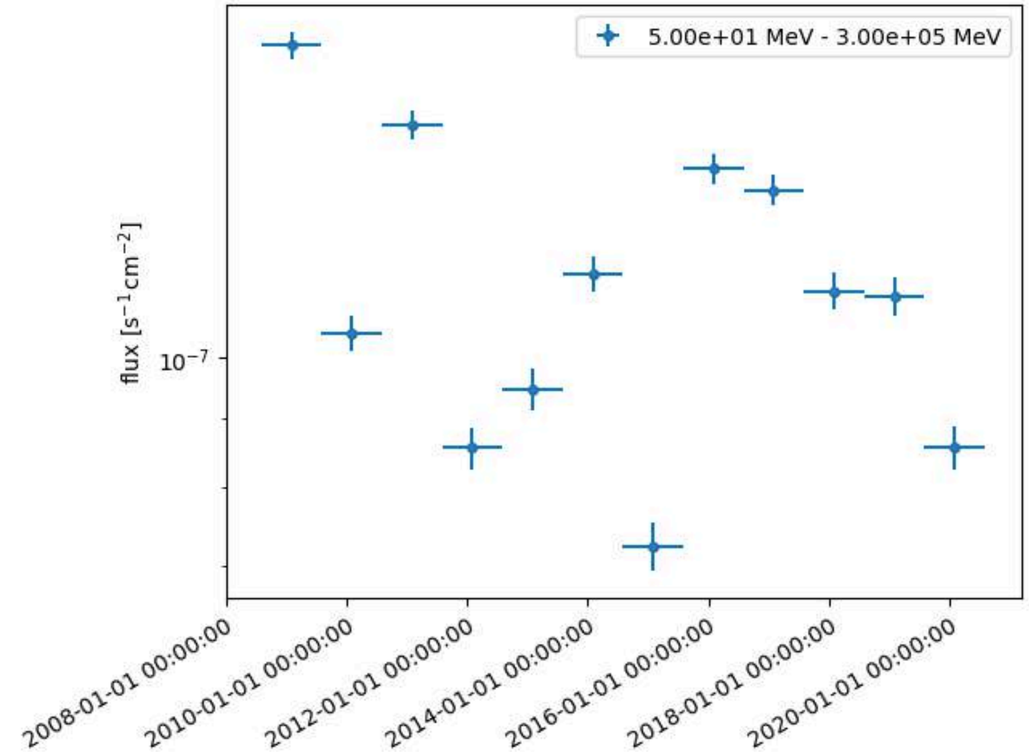
- Initial fine bins of a dataset are grouped to create new bins
- Multiplicative correction factor (norm) fitted to the spectral model in each bin
  - Stores the likelihood profile in each bin
    - Diagnostic and re-computation later
- Compute FluxMaps, FluxPoints, Lightcurves, FluxProfiles, etc
- Multiple serialisation formats supported
  - GADF-SED, Astropy table, Binned Time Series…

gammapy.catalog

```
from gammapy.catalog import SourceCatalog4FGL
catalog = SourceCatalog4FGL()
print(len(catalog.table))
src = catalog["PKS 2155-304"]
src.lightcurve().plot()
```

```
6659
```

```
<Axes: xlabel='Time [iso]', ylabel='flux [$\\mathrm{s^{-1}\\,cm^{-2}}$]'>
```



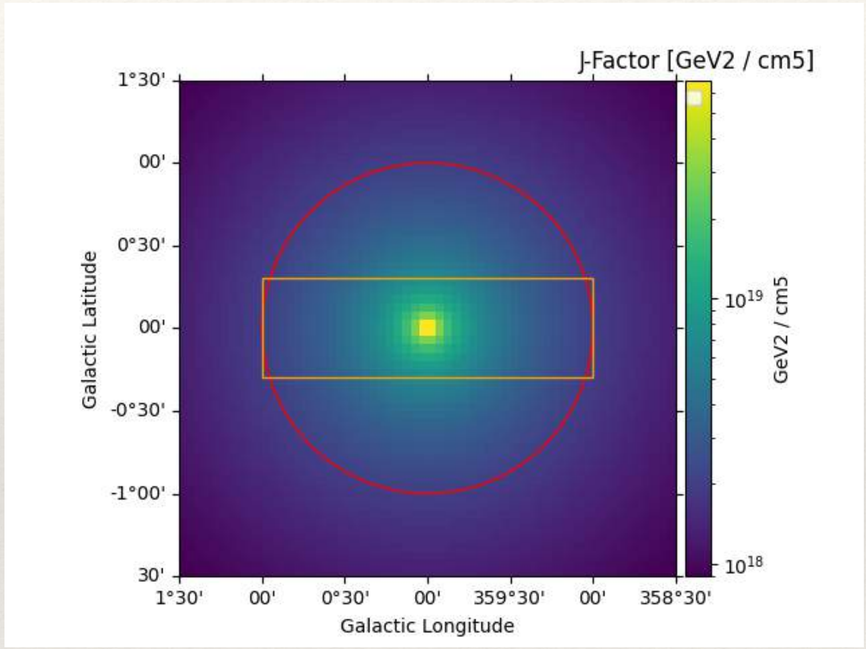
- Access to common catalogs
- Maps catalog information to gammapy objects
  - Easy to plot, access models, spectra, etc

- `hgps` / `SourceCatalogHGPS` - H.E.S.S. Galactic plane survey (HGPS)
- `gamma-cat` / `SourceCatalogGammaCat` - An open catalog of gamma-ray sources
- `3fgl` / `SourceCatalog3FGL` - LAT 4-year point source catalog
- `4fgl` / `SourceCatalog4FGL` - LAT 8-year point source catalog
- `2fhl` / `SourceCatalog2FHL` - LAT second high-energy source catalog
- `3fhl` / `SourceCatalog3FHL` - LAT third high-energy source catalog
- `2hwc` / `SourceCatalog2HWC` - 2HWC catalog from the HAWC observatory
- `3hwc` / `SourceCatalog3HWC` - 3HWC catalog from the HAWC observatory

https://docs.gammapy.org/1.2/tutorials/api/catalog.html#sphx-glr-tutorials-api-catalog-py

Additional packages

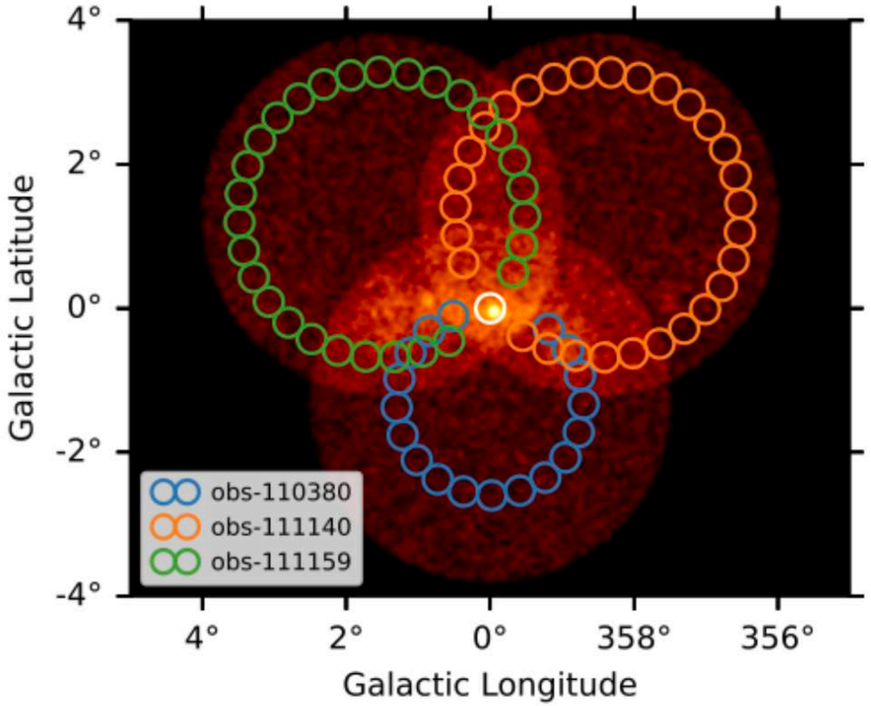
- gammapy.stats:
  - Statistical utility functions
- gammapy.astro:
  - utility functions for studying physical scenarios in high-energy astrophysics
- gammapy.visualization:
  - helper functions for plotting and visualizing analysis results and Gammapy data structures
- gammapy.analysis:
  - High level interface
  - python scripts, notebooks…
  - Automatize workflows driven by parameters declared in a configuration file in YAML format

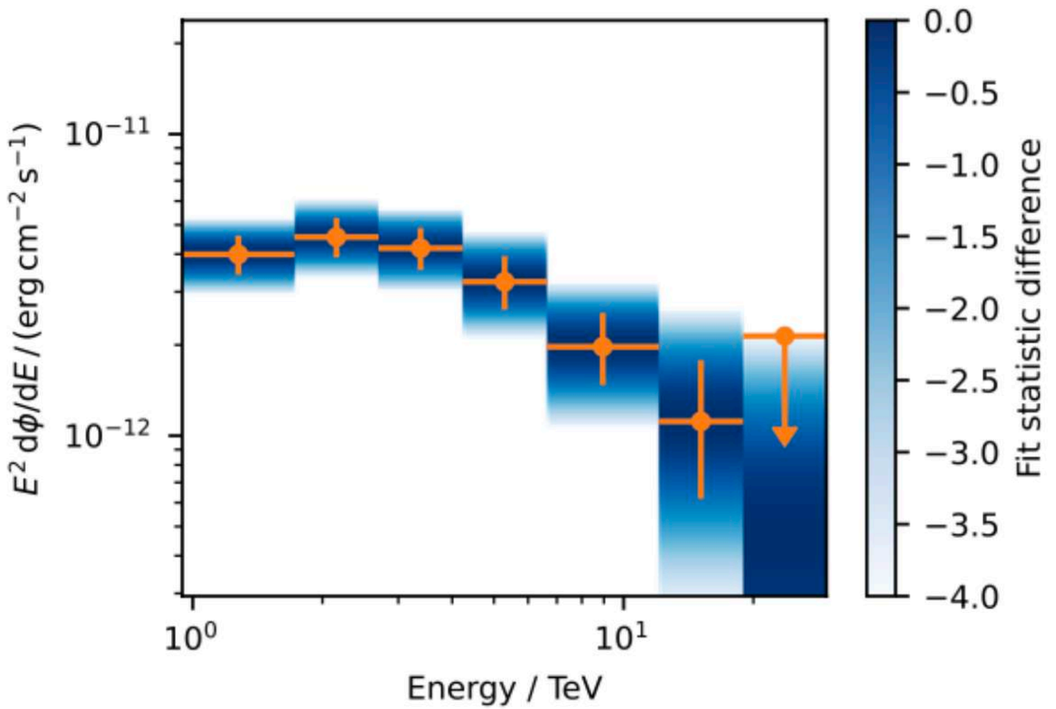


https://docs.gammapy.org/1.2/tutorials/api/astro_dark_matter.html#sphx-glr-tutorials-api-astro-dark-matter-py

Standard analysis examples

Example I. A 1D spectral analysis

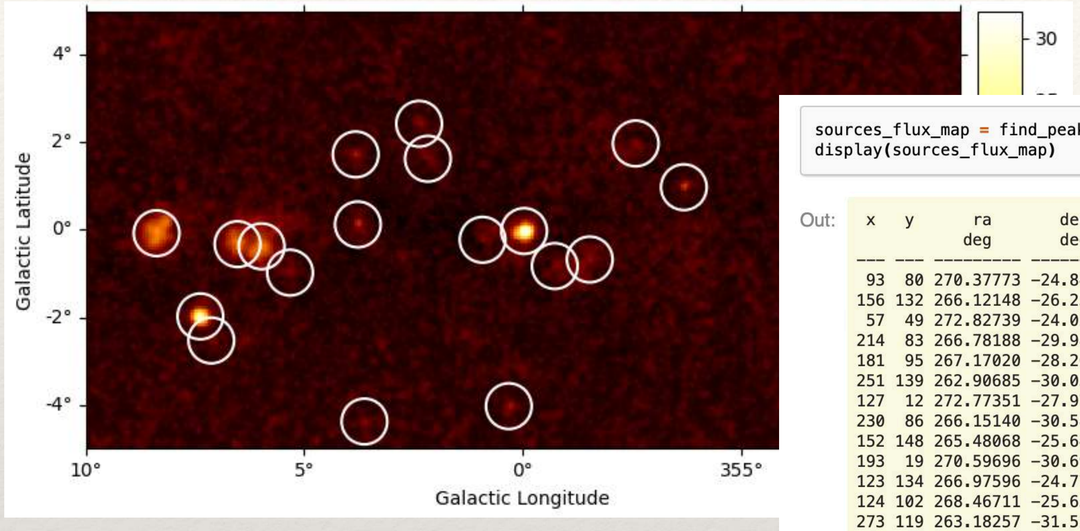




- This is a **"classical" gamma-ray analysis**: measuring a spectrum of a source and estimating the background from these ring-like arrangements of multiple regions.
- This uses simulated data from CTA and also exports the **likelihood per energy bin (**shown as the blue colored band). This is a lot more information then e.g. reporting only errors or upper limits for flux points

https://docs.gammapy.org/1.1/tutorials/analysis-1d/spectral_analysis.html#sphx-glr-tutorials-analysis-1d-spectral-analysis-py

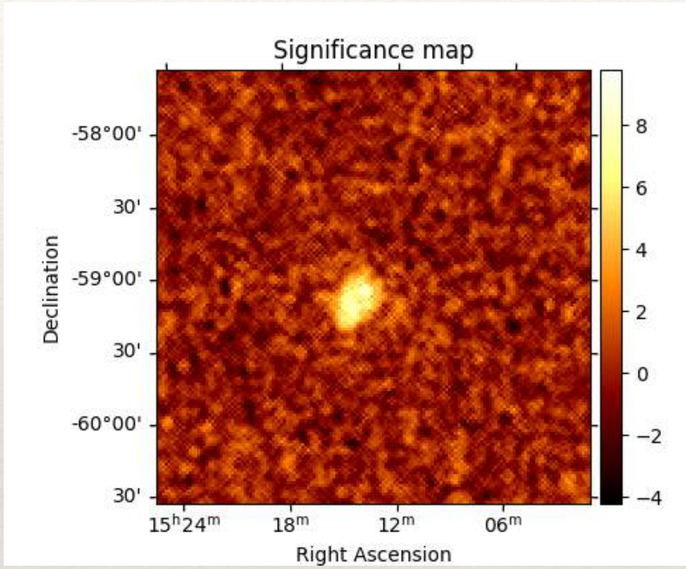
Example II: Building significance maps




```
sources_flux_map = find_peaks_in_flux_map(maps, threshold=5, min_distance="0.25 d
display(sources_flux_map)
```

Out:

x	y	ra	dec	...	stat_null	success	flux	flux_err
		deg	deg	...			1 / (cm2 s)	1 / (cm2 s)
93	80	270.37773	-24.84233	...	840.62111	True	8.167e-11	2.347e-11
156	132	266.12148	-26.23306	...	692.59451	True	5.923e-11	1.872e-11
57	49	272.82739	-24.02653	...	723.70546	True	5.614e-11	1.740e-11
214	83	266.78188	-29.98429	...	838.88781	True	9.765e-11	2.599e-11
181	95	267.17020	-28.26173	...	659.39224	True	1.295e-10	3.186e-11
251	139	262.90685	-30.05853	...	766.22149	True	6.708e-11	1.925e-11
127	12	272.77351	-27.97934	...	433.37517	True	4.133e-11	1.357e-11
230	86	266.15140	-30.58926	...	865.54892	True	1.287e-10	3.058e-11
152	148	265.48068	-25.64323	...	611.76473	True	7.129e-11	1.889e-11
193	19	270.59696	-30.69138	...	445.28958	True	6.653e-11	1.721e-11
123	134	266.97596	-24.77174	...	655.77218	True	9.271e-11	2.137e-11
124	102	268.46711	-25.63326	...	881.98258	True	1.714e-10	3.070e-11
273	119	263.18257	-31.52587	...	846.92490	True	1.774e-10	2.965e-11
80	92	270.15899	-23.98049	...	1093.46225	True	4.609e-10	5.312e-11
69	93	270.40919	-23.47797	...	1044.25763	True	4.586e-10	5.293e-11
32	98	271.16056	-21.74479	...	1036.42361	True	5.442e-10	5.833e-11
52	60	272.43197	-23.54282	...	1092.41995	True	6.023e-10	4.699e-11
200	99	266.41449	-28.97054	...	137.30287	True	1.424e-09	7.945e-11

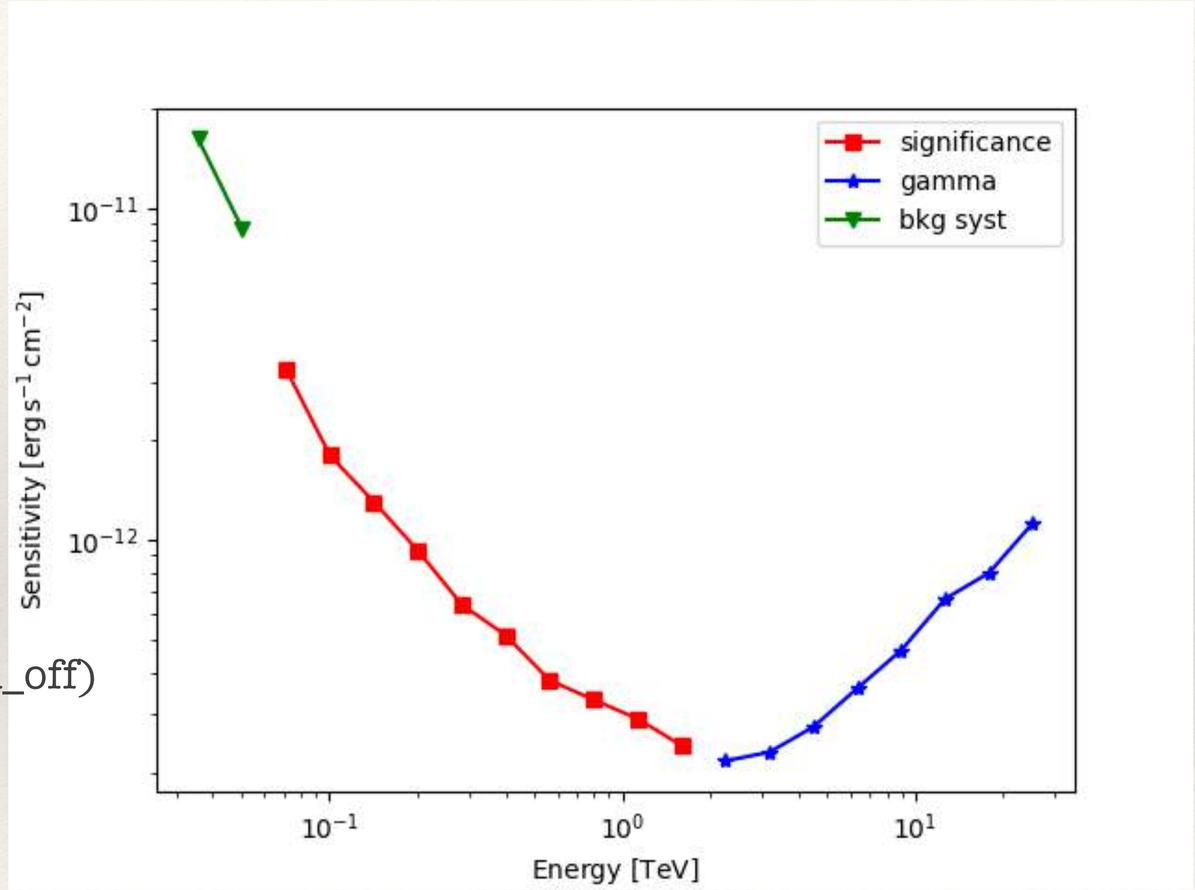


- TSMapEstimator : compares the likelihood function L optimized with and without a given source.
- ExcessMapEstimator: LiMa formalism, faster, no fitting

https://docs.gammapy.org/1.1/tutorials/analysis-2d/ring_background.html#sphx-glr-tutorials-analysis-2d-ring-background-py
https://docs.gammapy.org/1.1/tutorials/analysis-2d/detect.html#sphx-glr-tutorials-analysis-2d-detect-py

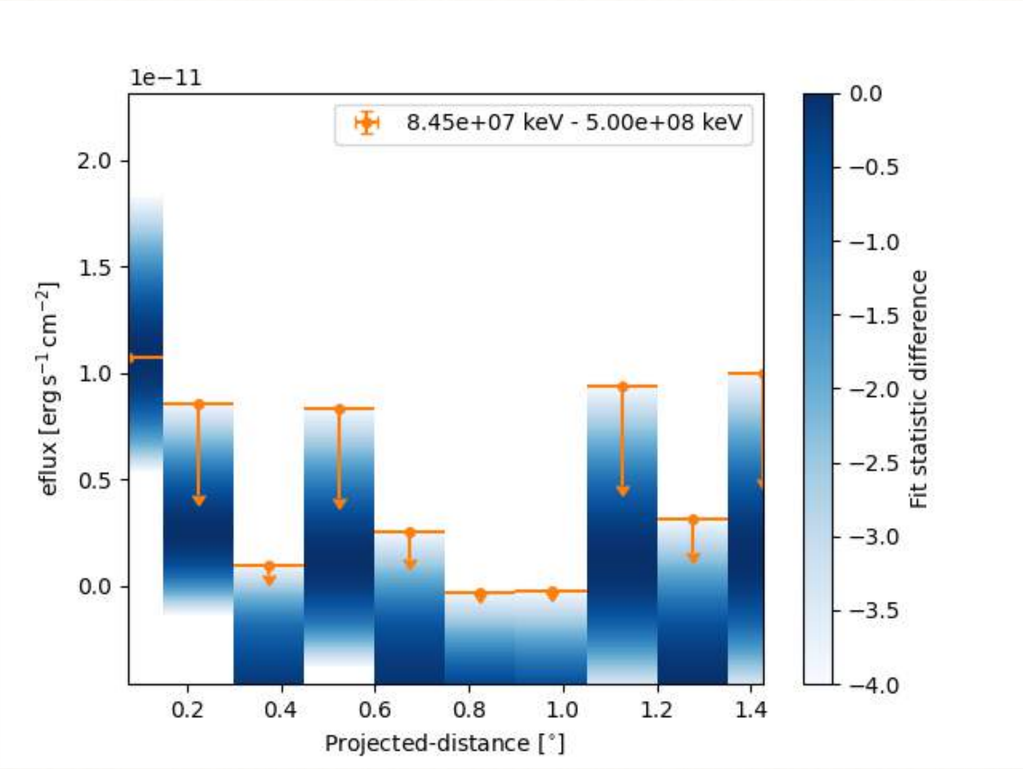
III: Compute sensitivity

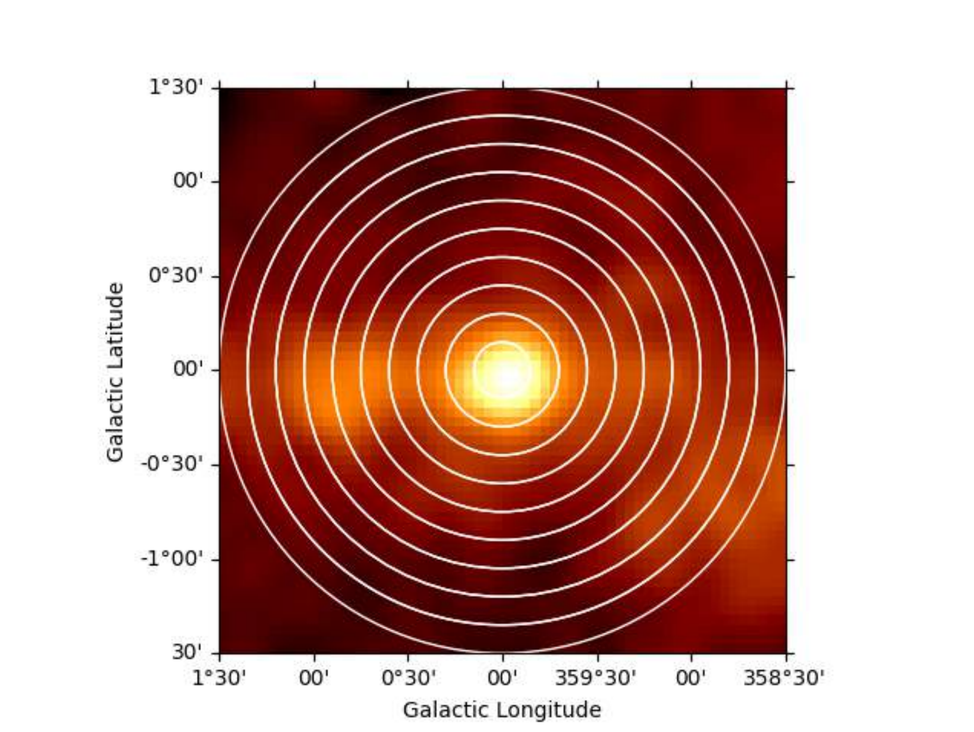
```
sensitivity_estimator = SensitivityEstimator(
    gamma_min=10,
    n_sigma=5,
    bkg_syst_fraction=0.05,
)
sensitivity_table = sensitivity_estimator.run(dataset_on_off)
```



https://docs.gammapy.org/1.1/tutorials/analysis-1d/cta_sensitivity.html#sphx-glr-tutorials-analysis-1d-cta-sensitivity-py

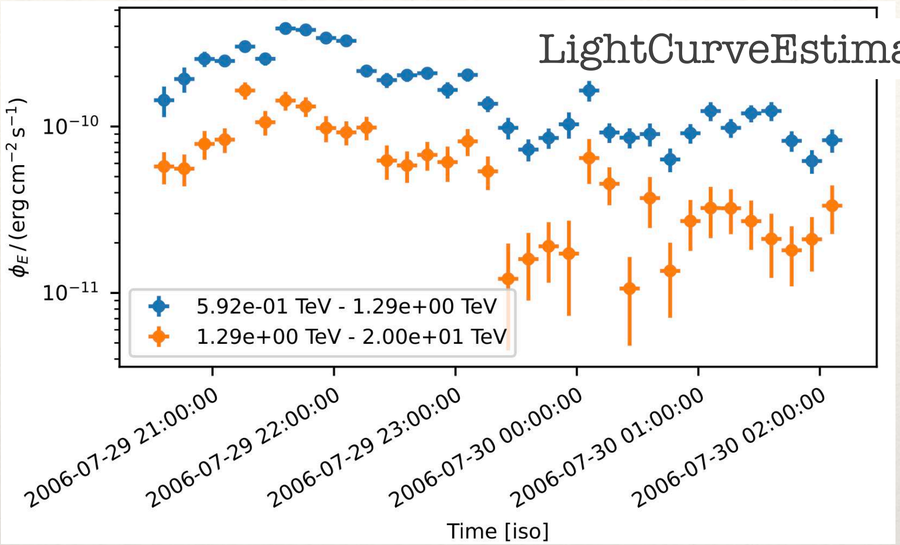
IV: Compute flux profiles





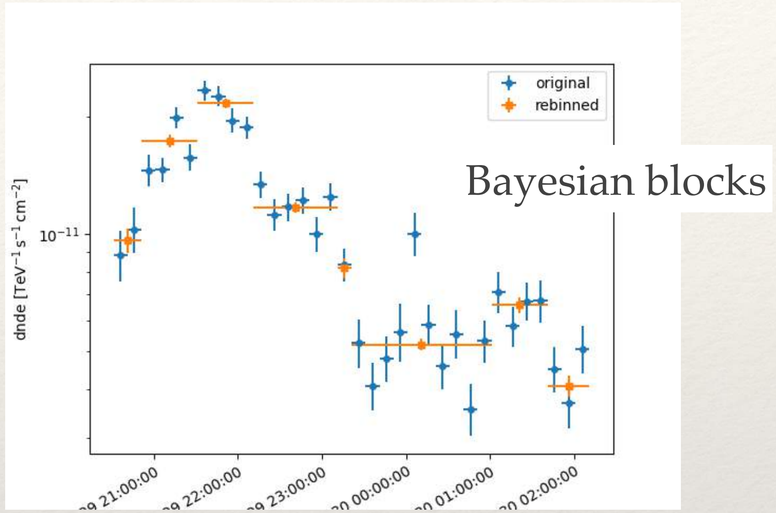
https://docs.gammapy.org/1.1/tutorials/analysis-3d/flux_profiles.html#sphx-glr-tutorials-analysis-3d-flux-profiles-py

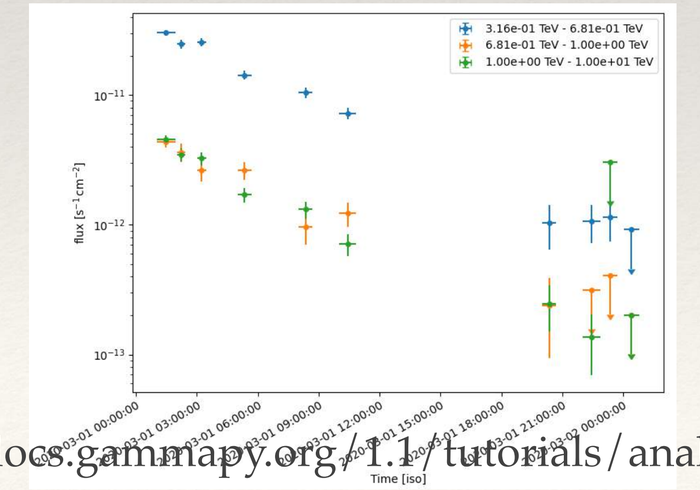
V: Temporal studies



Rebin using

likelihood profiles





Simultaneous

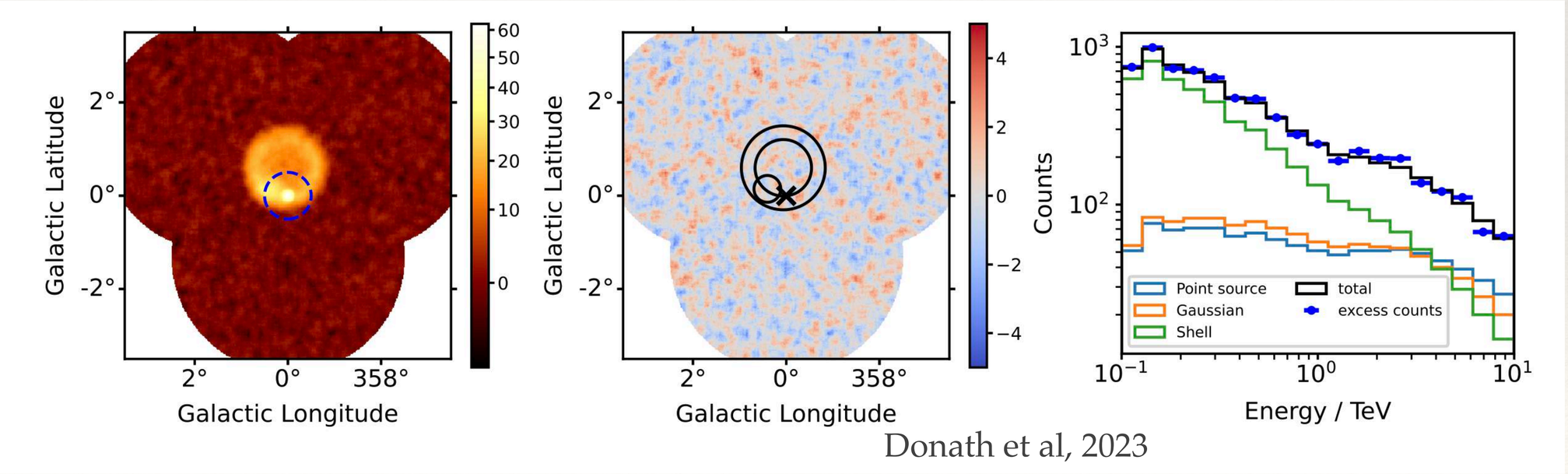
Spectro-temporal fitting

name	value	unit	error	frozen
---	---	---	---	---
index	2.9558E+00		3.131E-02	False
amplitude	1.0292E-11	cm-2 s-1 TeV-1	3.541E-13	False
reference	1E+00	TeV	0E+00	True
t0	5.8447E+00	h	2.002E-01	False
t_ref	5.8909E+04	d	0E+00	True

https://docs.gammapy.org/1.1/tutorials/analysis-time/light_curve_flare.html#sphx-glr-tutorials-analysis-time-light-curve-flare-py

VI: 3D simulation and analysis

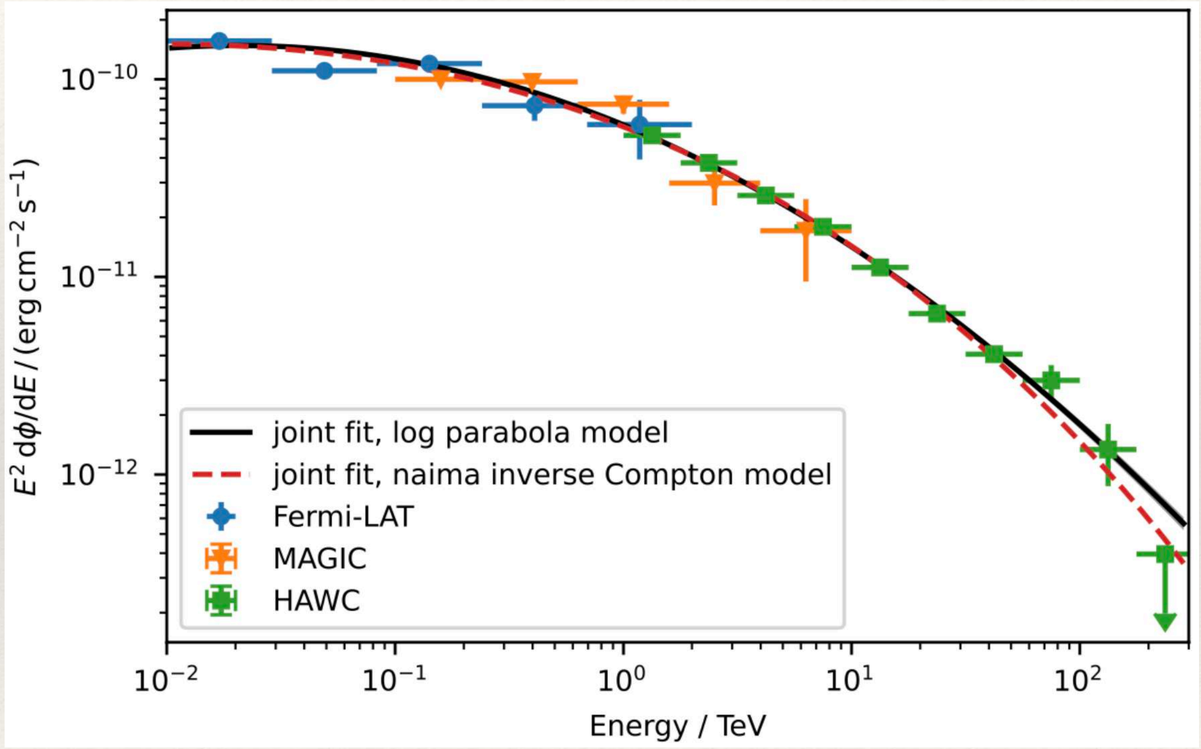
- Simultaneous fitting of multiple sources and background
- Allows to disentangle the contribution from overlapping sources to the same spatial region



Donath et al, 2023

VII: Multi instrument analysis

- A Spectral Fit combining different types of data
  - MAGIC DL3 data - point like 1D spectral analysis
  - Fermi-LAT DL4 data - full 3D analysis
  - HAWC DL5 data - precomputed flux points
- Fit physical models directly to reduced data (and not precomputed flux points)
- Displays the flexibility of gammapy

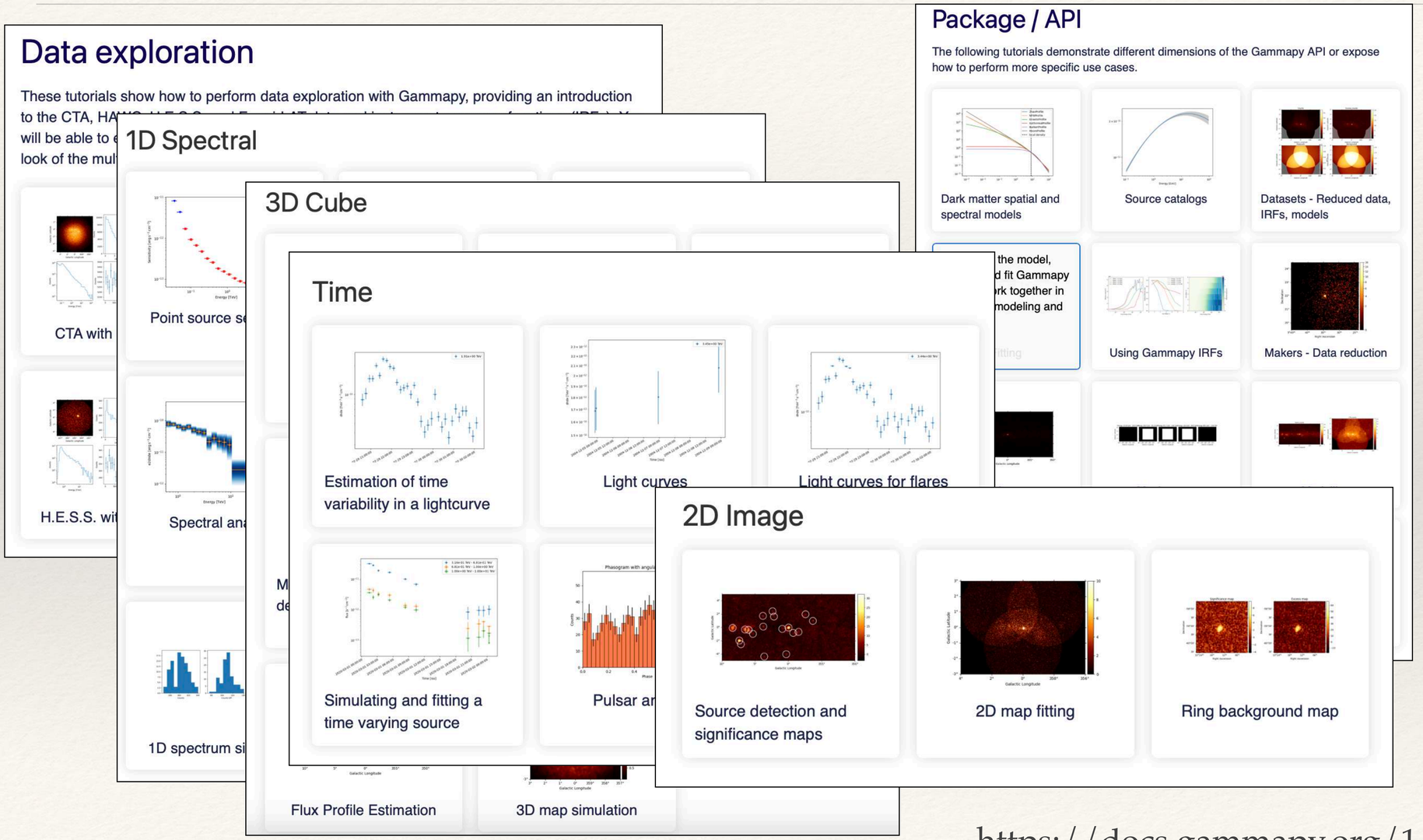


Docs and validation

Development & CI setup

- Hosted and **openly developed on GitHub**: https://github.com/gammapy/gammapy
- **GitHub actions** used to **run CI on each PR** and a release pipeline
- **codecov.io** used for **monitoring of code test coverage**
- **Sphinx** used to build the documentation: https://docs.gammapy.org/
- **pytest** for testing
- **black** code formatting used for **a consistent code format.**
- **Isort** to automatically **sort and format imports**
- **Flake8** to **check PEP8** standard

Tutorials

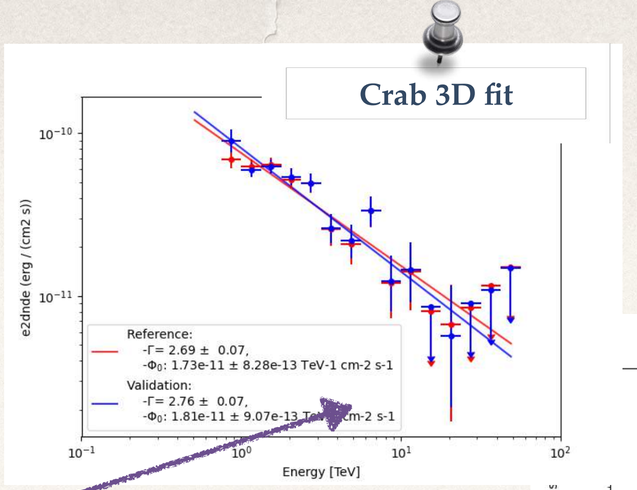


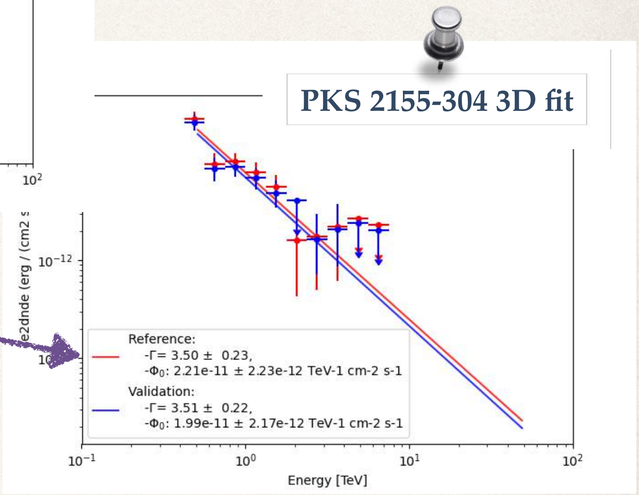
- Selection of most common analysis scenarios
- Specific API tutorials for advanced users
- Currently 38 notebooks

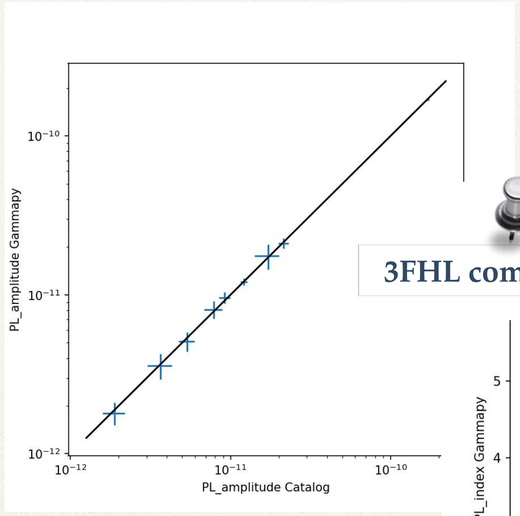
https://docs.gammapy.org/1.2/tutorials/index.html

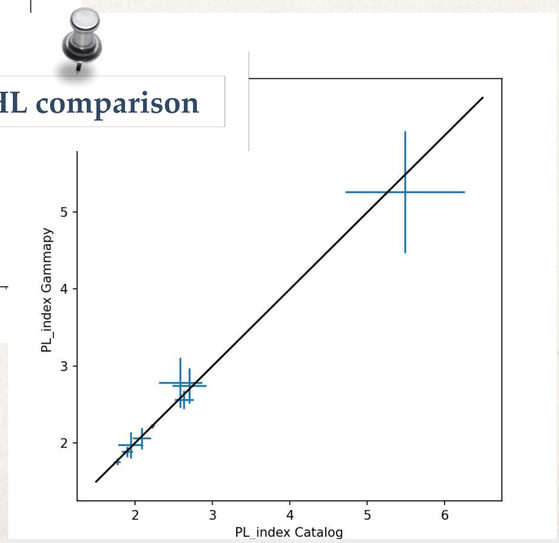
Gammapy validation

- 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







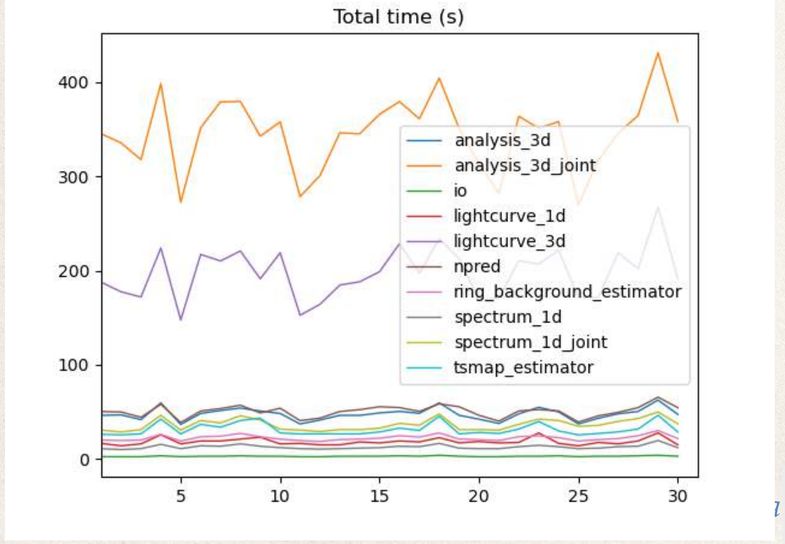


See: https://github.com/gammapy/gammapy-benchmarks/tree/master/validation

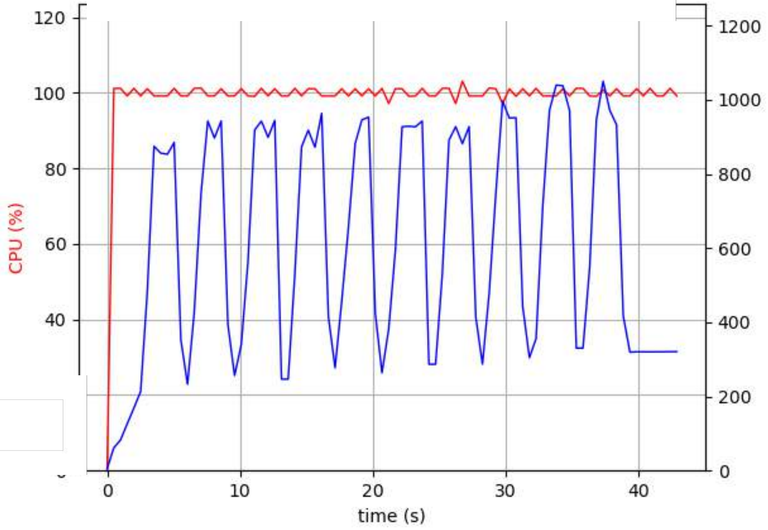
See: https://github.com/gammapy/gammapy-benchmarks/tree/master/benchmarks

Gammapy benchmarks

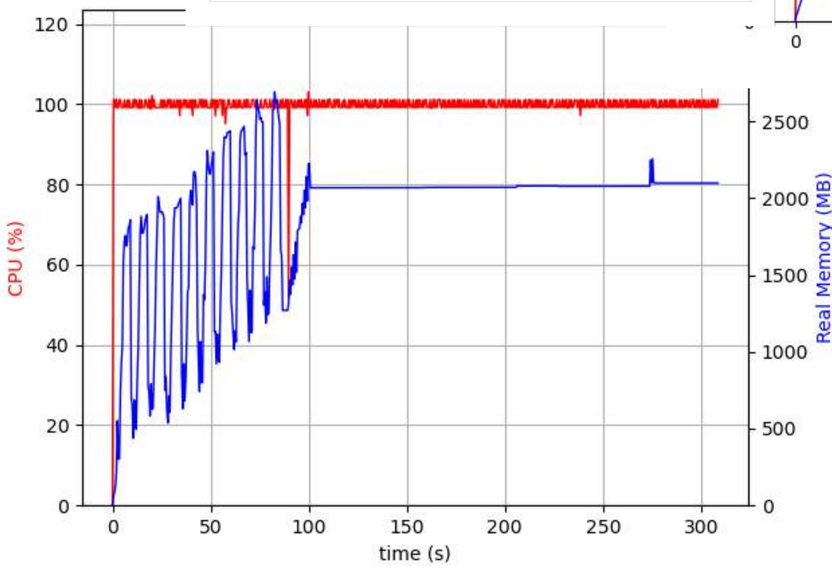
- Daily automatic monitoring of memory usage and computation time for most common use cases
- Computation time across different actions
  - Loading observations
  - Binning data and IRFs
  - Fitting
  - FluxPointEstimations
- Pinpoint PR significantly affecting the performance



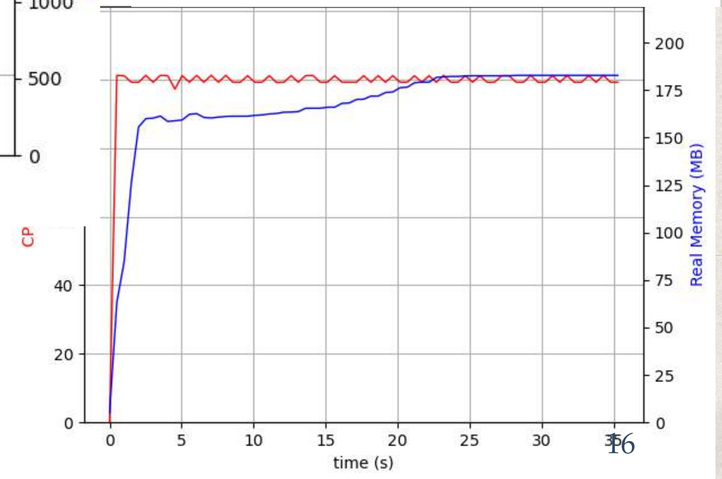
Stacked 3D analysis



Joint 3D analysis



Joint 1D analysis

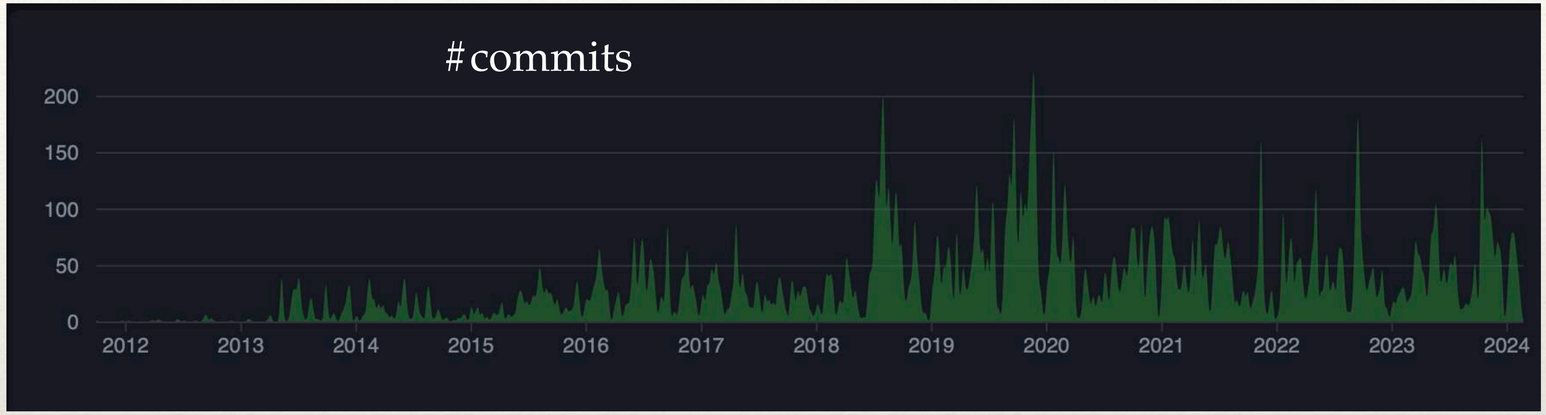


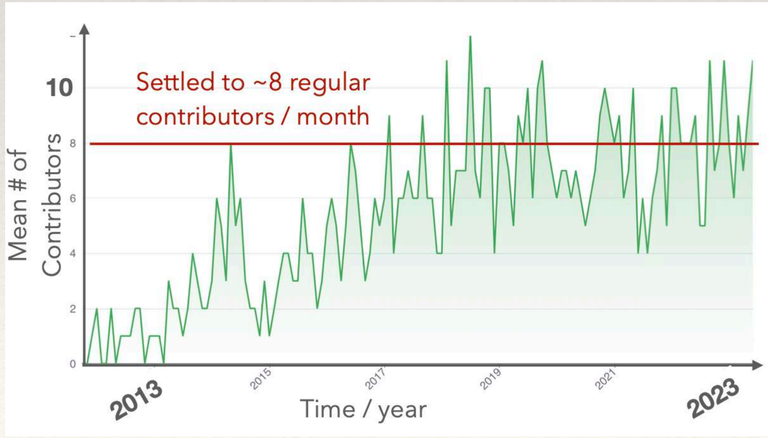
Project info

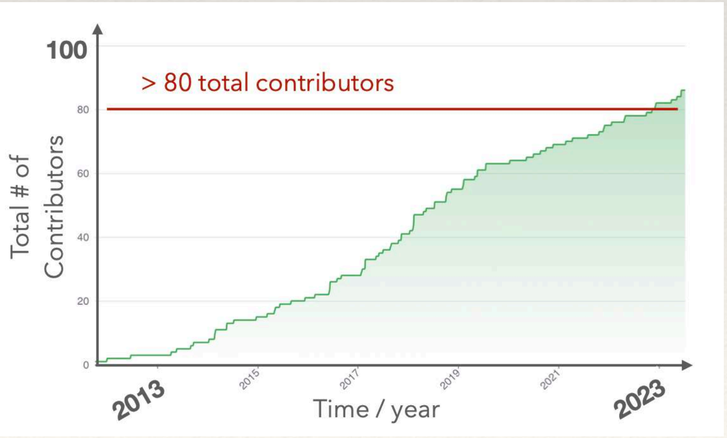
Development status

Versions

Version v1.2 10.5281/zenodo.10726484	Feb 29, 2024
Version v1.1 10.5281/zenodo.8033275	Jun 13, 2023
Version v1.0.1 10.5281/zenodo.7734804	Mar 14, 2023
Version 1.0 10.5281/zenodo.7311399	Nov 10, 2022
Version v0.20 10.5281/zenodo.6552377	May 13, 2022







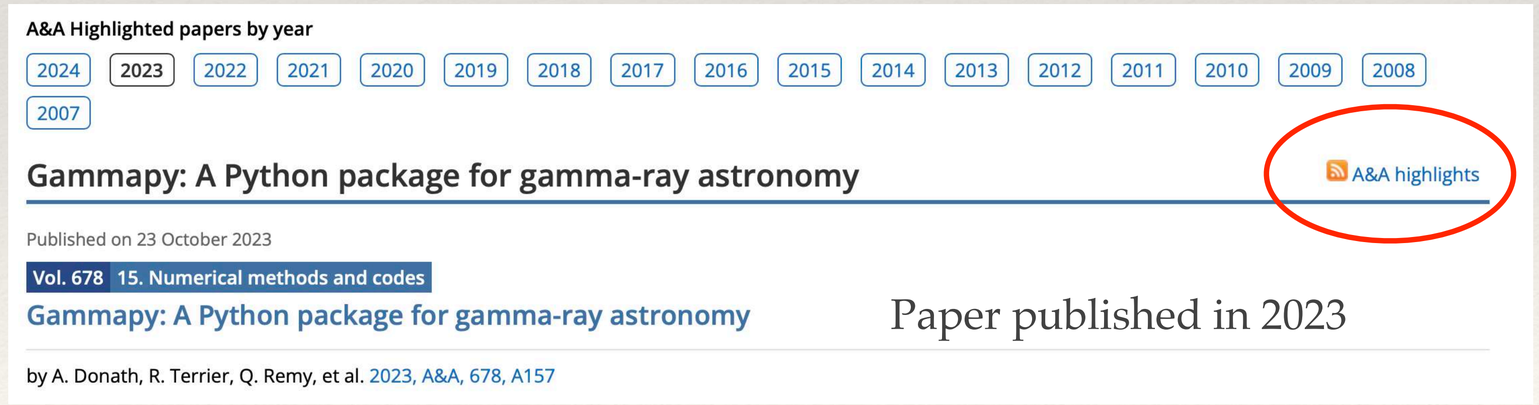
- ~50,000 lines of code
  - 34% API, 25% docs, 20% tests
- 20 minor releases
- LTS 1.0 in Nov, 2022
  - LTS release ~ 2yrs
  - Minor releases ~6 months
  - Bug fixes ~ as required
- Min 6 month deprecation cycle

Usage status

- Official tool within:
  - Library for the **CTA Science Analysis Tools**
  - Official analysis tool within **H.E.S.S.** and **MAGIC**
  - Proposed tool for **SWGO**

- Ongoing tests for
  - Propotype analyses for **HAWC** and **VERITAS**
  - Internal developments within **LHAASO**
  - Prototype analysis for **X-ray** (https://zenodo.org/records/7092736)

- Internal dependencies of
  - fermipy (gammapy.maps)
  - agnpy (gammapy.modeling)
  - Jetset



Paper published in 2023

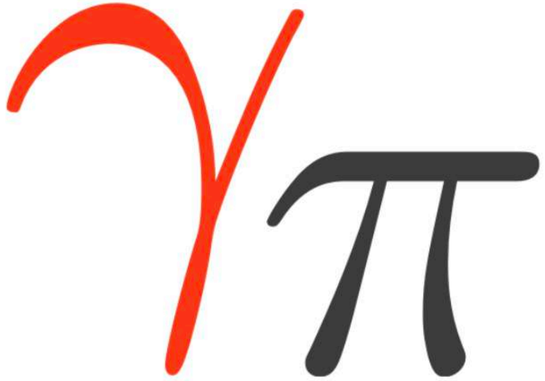
Project Organisation

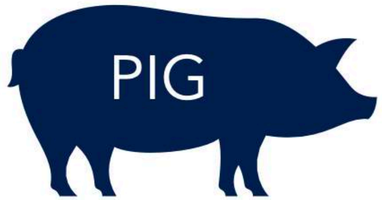
Project Managers
"Non-technical executive leads"

Lead Developers
"Technical executive leads"

Sub-package maintainers
"Lead the maintenance of sub-packages"

Contributors





We have **Proposals for Improving Gammapy** (PIG), which follow the idea of PEPs, NEPs etc.

Coordination Committee (CC)
"Promotes, coordinates and steers Gammapy developments"

Consists of representatives of the contributing institutions:

MAX-PLANCK-INSTITUT FÜR KERNPHYSIK HEIDELBERG

DESY.

APC, CNRS, FRANCE

irfu cea saclay

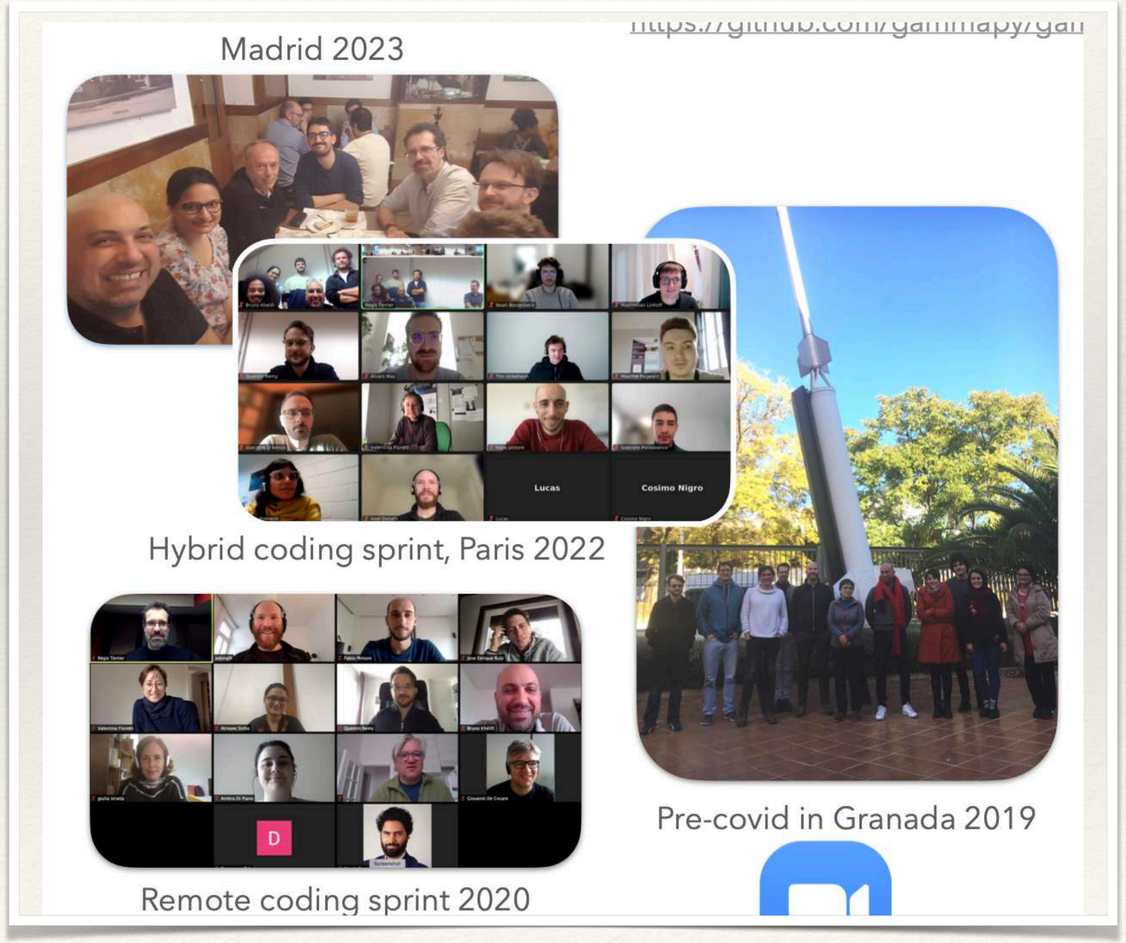
UNIVERSIDAD COMPLUTENSE MADRID

l'Observatoire de Paris

https://gammapy.org/team.html

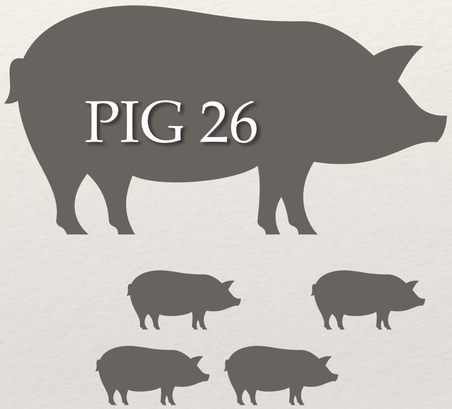
Contact Us!

- Install gammapy:
  - conda install -c conda-forge gammapy
- Docs:
  - https://docs.gammapy.org/1.2/
- Issues?
  - Slack: gammapy.slack.com (quick questions, immediate help)
  - GitHub issues: https://github.com/gammapy/gammapy/issues (feature requests & bug reports)



Future outlook

- Development of LTS versions coordinated via roadmap documents
- Roadmap for v2.0 includes
  - Solution for distributed computing and / or GPU support, `ray` / pytorch / jax
  - Metadata handling with pydantic
  - Separation of internal data format from GADF
  - Spectral Unfolding
  - Unbinned analysis
  - Support for event types/classes…
  - MCMC sampling, etc…



- Sub-packages with broader interest
  - gammapy.maps
  - gammapy.modeling.models
  - gammapy.catalog

Thank you!