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

Hands-on VHE gamma-ray analysis with gammapy

Kirsty Feijen (APC)

Adelaide Workshop

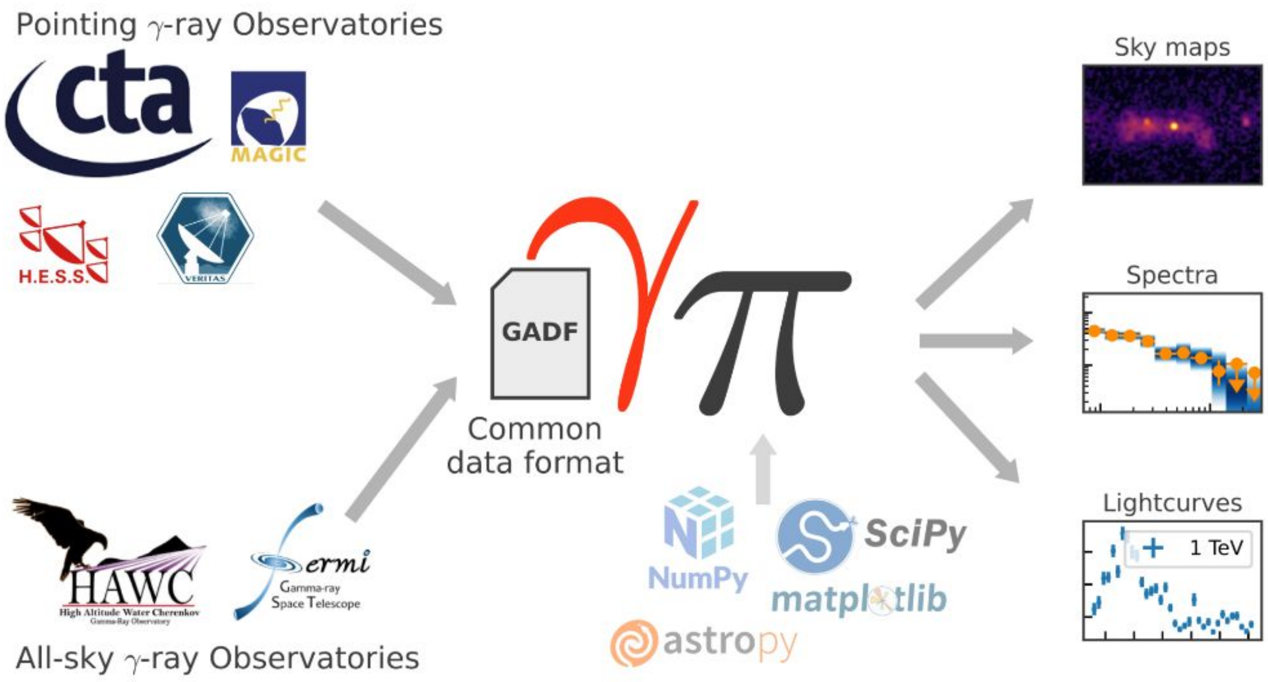
~~February~~ May 2024

https://github.com/Astro-Kirsty/gammapy_Adelaide_2024/

Who am I?

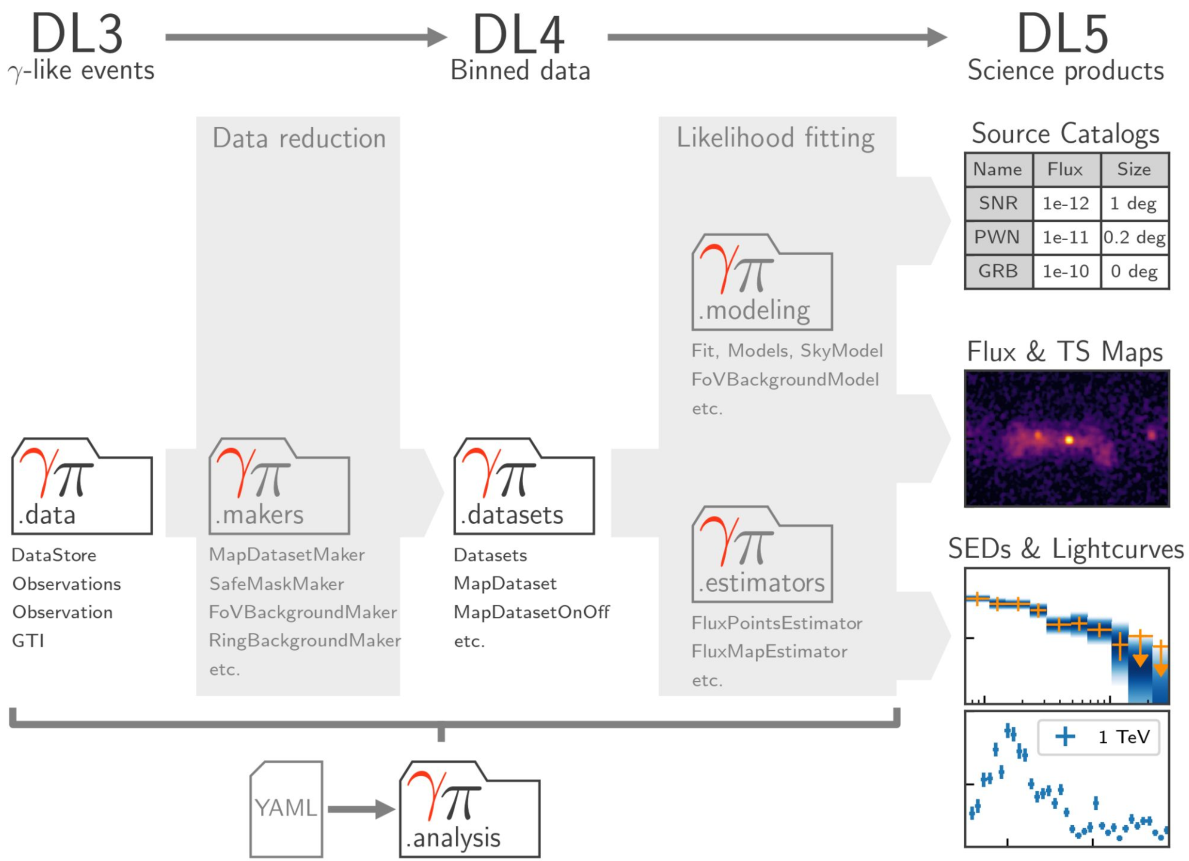
- **Previously:** PhD at the University of Adelaide with Sabrina Einecke and Gavin Rowell
- **Currently:** Post-doc at APC (Astroparticule et Cosmologie) lab in Paris, France
- I work in HESS and CTA with a research focus on galactic TeV gamma-ray sources
- I'm a maintainer of the python package gammapy

Gammapy overview



- v1.0 (Long Term Stable release) Nov 10th 2022
  - v1.0.2 (2nd bug release) Dec 6th 2023
- v1.1 (feature release) June 13th 2023
- v1.2 (feature release) Feb 29th 2024

Gammapy overview



2-step analysis procedure

- Data reduction (DL3 to DL4)
- Data modeling/fitting (DL4 to DL5)

Getting started: installation

Quickstart installation with conda/mamba

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

See Getting started

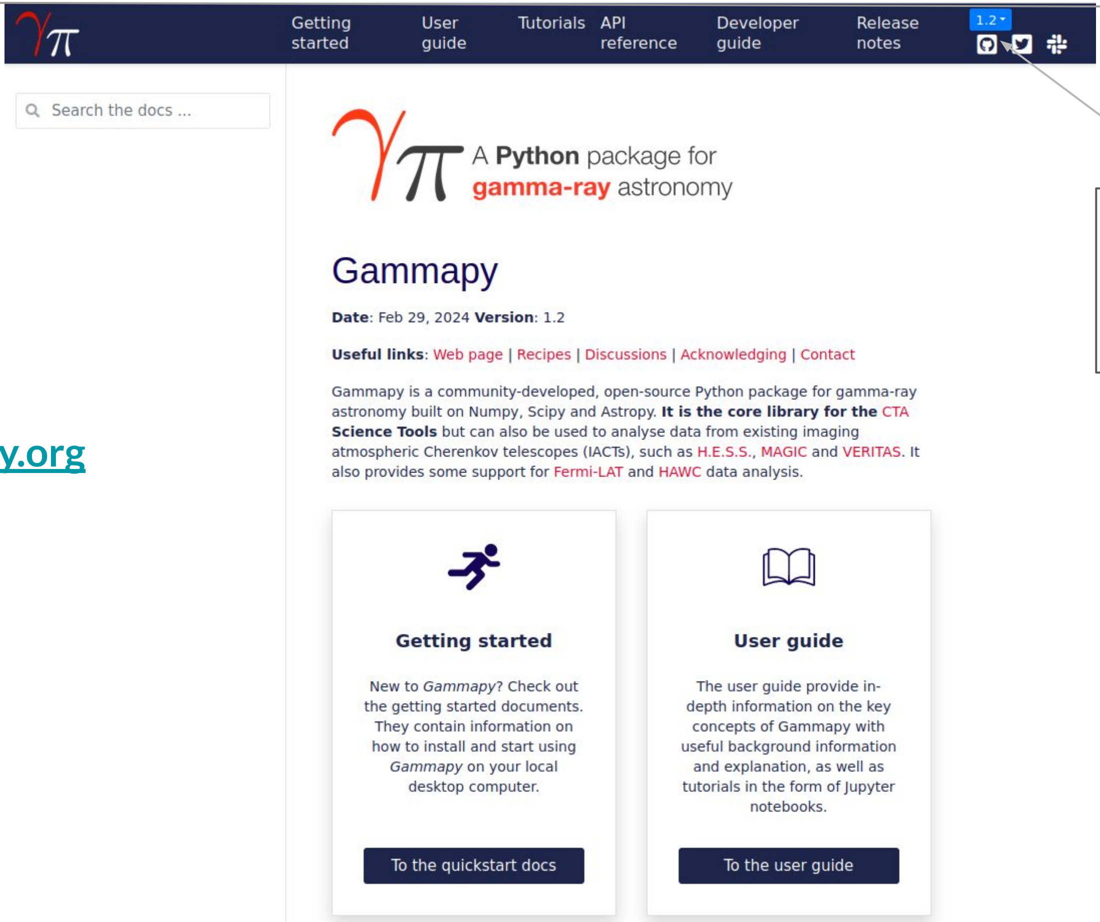
Installation with pip

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

install all dependencies

For this workshop, the other instructions are found on the GitHub

Getting started: documentation



Switch between versions

See docs.gammapy.org

Getting helping, reporting issues

How to provide feedback/get help

- #help channel on gammapy.slack
- #gammapy channel on hesschat.slack (if you are a HESS member)
- GitHub discussion, in particular the help category

How to report issues and bugs or request a new feature

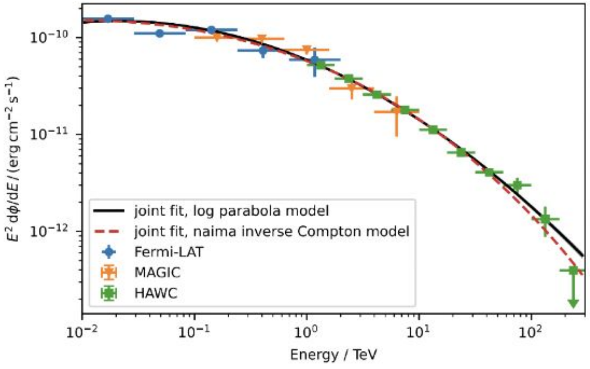
- GitHub issues page (requires a GitHub account)

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 (Fridays 2pm CET)
- Open a Pull Request (PR) on the GitHub gammapy repo

Recent activities

- Gammapy v1.0 paper is published (A. Donath, R. Terrier, Q. Remy, et al. [2023, A&A, 678, A157](#))
  - Fully reproducible code and figures. See [GitHub repo](#) and [pdf file](#)

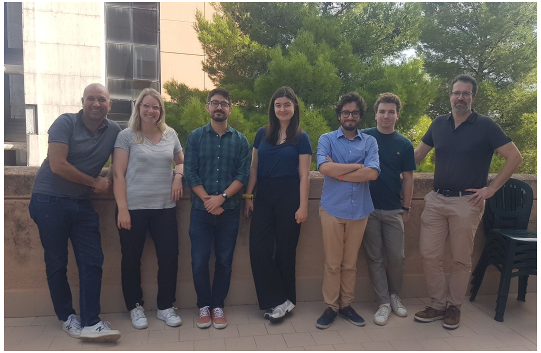


Multi-instrument joint fit of the Crab with a log parabola and inverse Compton models

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

- Most recent coding sprint in [Palermo October 2023](#)
  - 8 people on-site, 5 online



Gammapy & CTA

- v1.2 was released February 2024

Gammapy team involved in SDC effort

- Event types are supported
  - Distributed as two different data stores
  - Stacked or joint analyses are possible
- Missing features for SDC have been added:
  - Time dependent spectral models in v1.1
  - Metadata containers to support CTAO data model
- SDC will be important to prepare first SAT release

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

Introduction

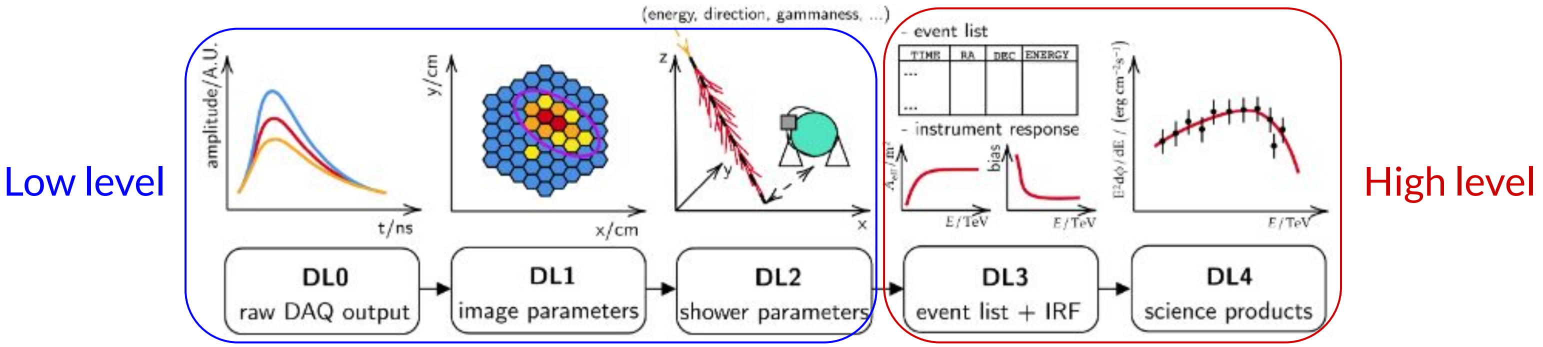
Kirsty Feijen (APC)

Adelaide Workshop

February 2024

Gamma-ray data analysis

- Low level analysis involves:
    - Calibrating the data
    - Applying cuts to reject hadrons
    - Reconstructing showers
    - Final product are event lists
- High level analysis utilises the event lists to produce a number of interesting products including, but not limited to
- We typically call these products data level (DL) 3-5

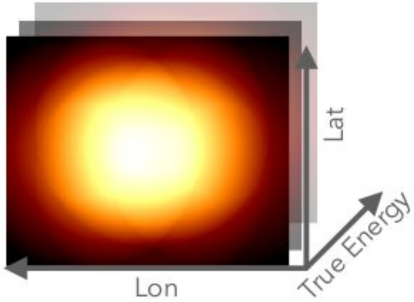


https://ui.adsabs.harvard.edu/abs/2021Univ....7..374N/abstract

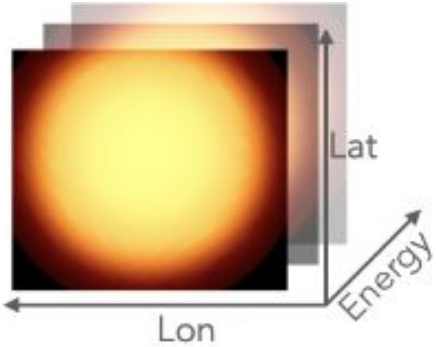
Data model/Instrument Response

- There are a number of different IRFs
  - Effective area, PSF, Energy dispersion, Background
- DL3 IRFs are reprojected onto the target geometry

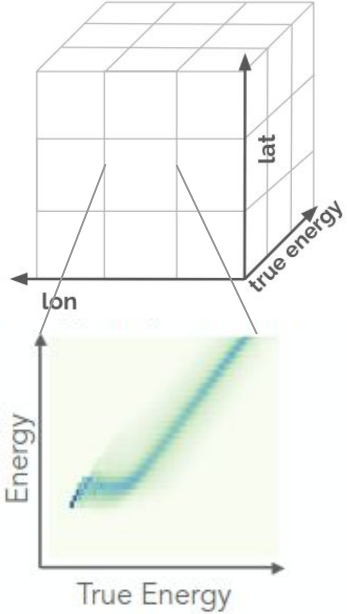
Exposure



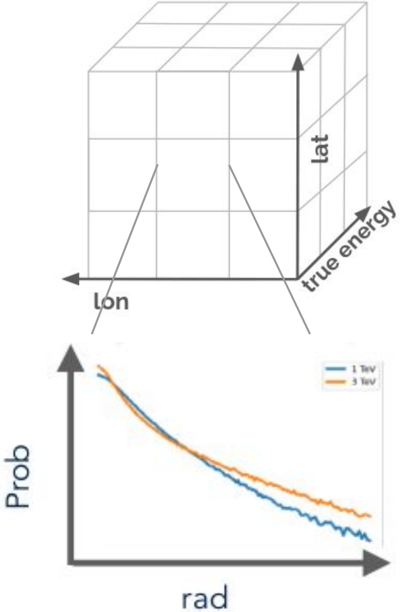
Background template



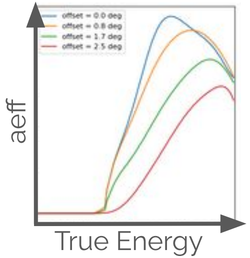
Energy Dispersion



PSF

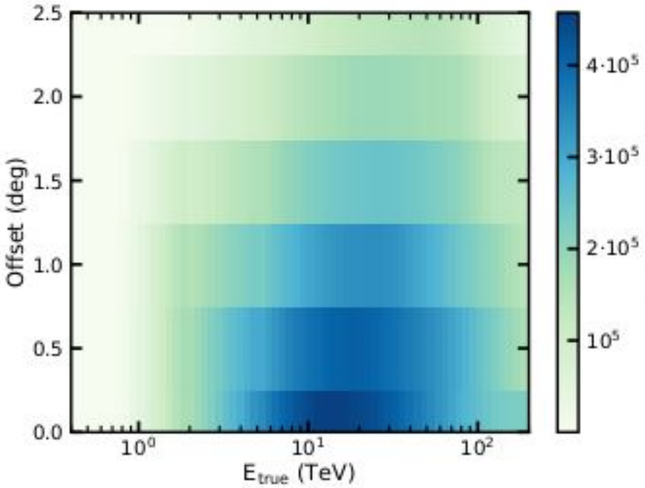


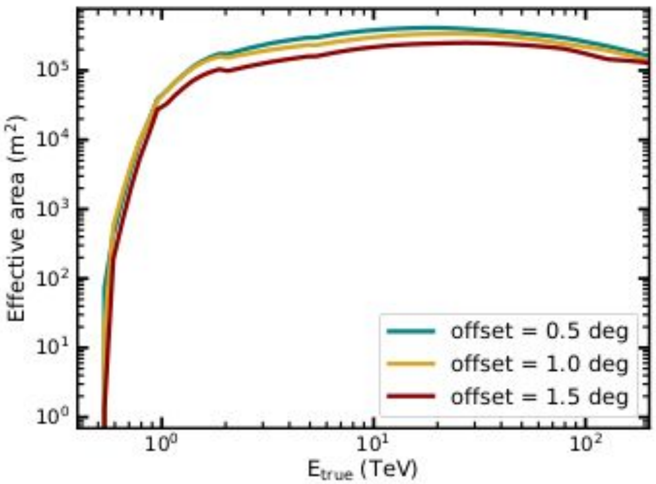
Effective Area



Effective area

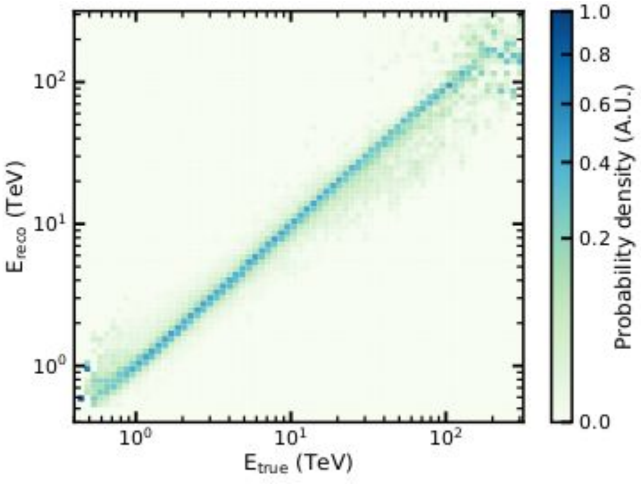
- The collection area of the detector, depends on:
  - Energy of the photons
  - Offset from pointing position
  - Observation zenith angle
  - Optical efficiency of telescopes
- Typically combined with observation time to derive the effective exposure

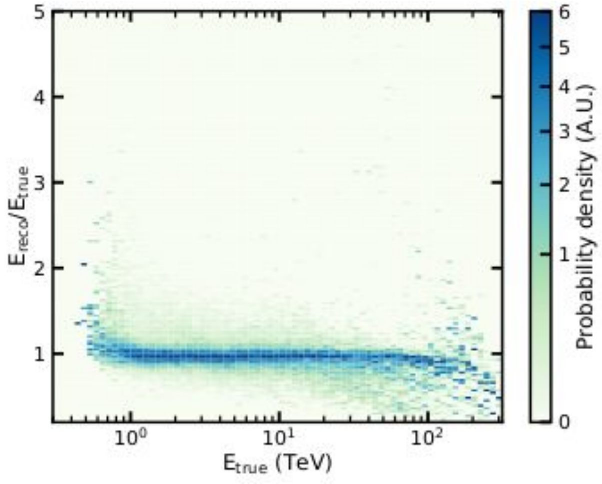


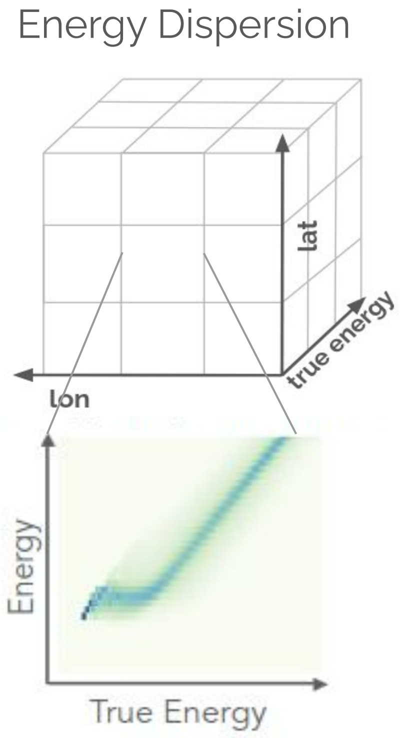


Energy dispersion

- Described by the probability that a gamma-ray with energy Etrue will be reconstructed with energy Ereco

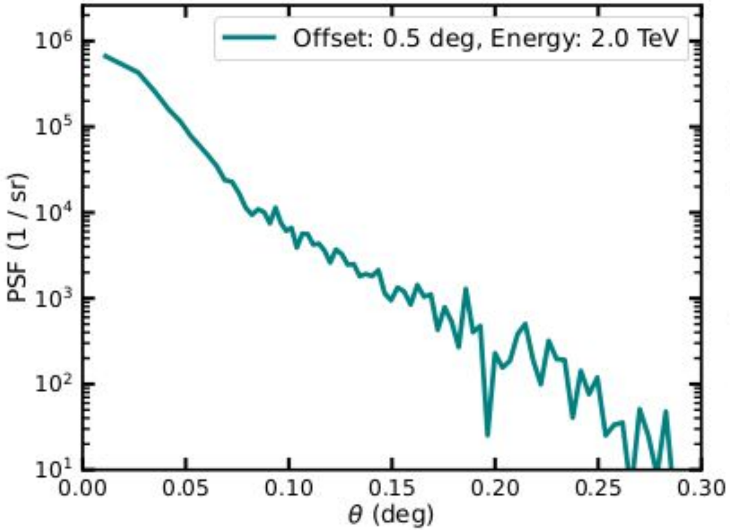


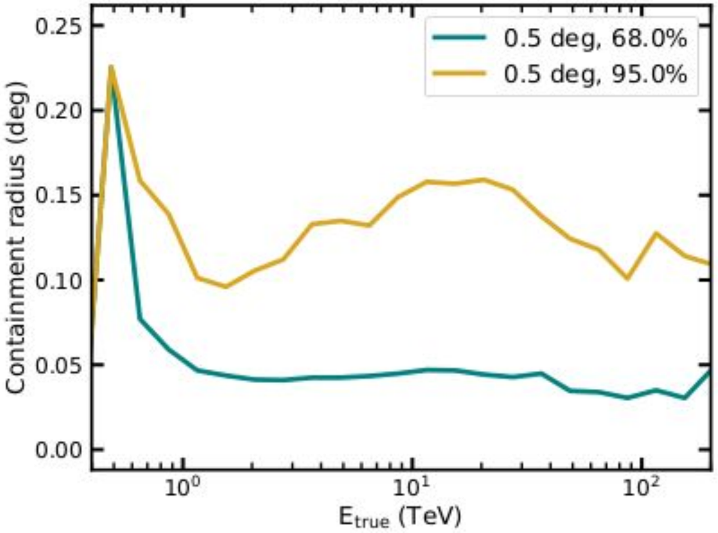




Point Spread Functions

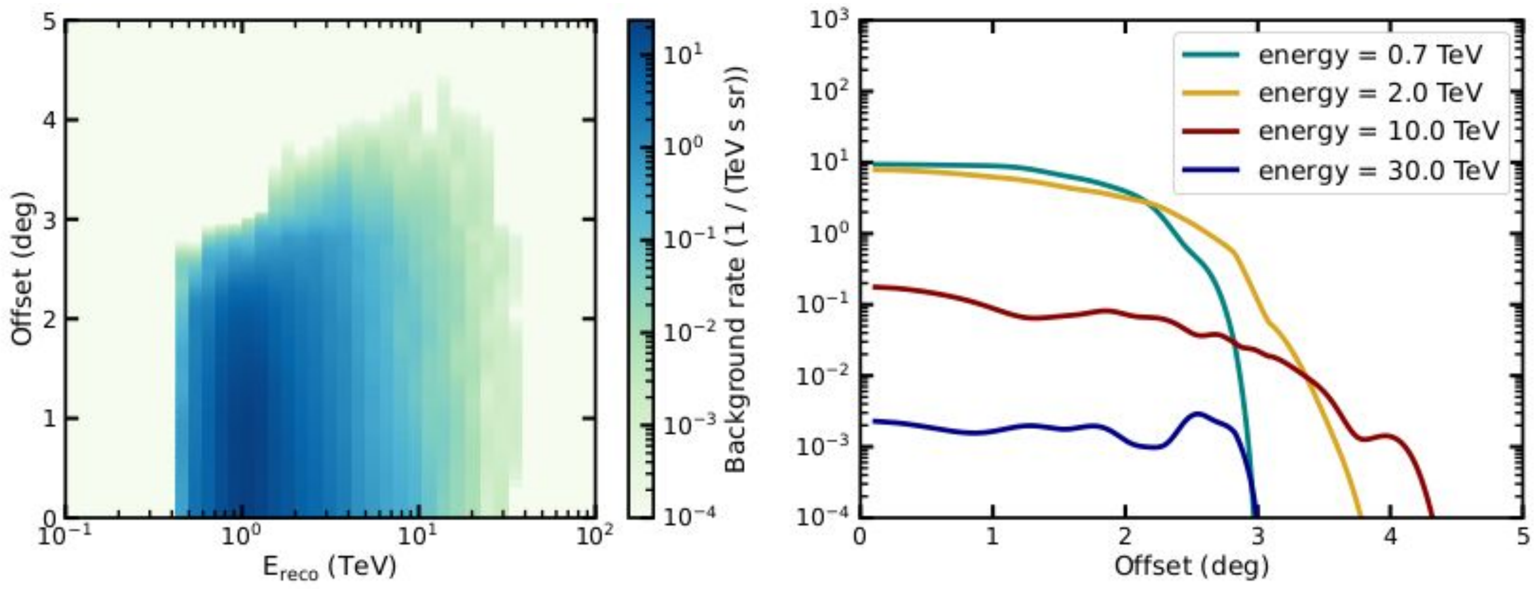
- Represents the reconstruction accuracy in the direction of incident gamma-ray
- Typically summarised by containment radius or angular distance to a fraction of the signal





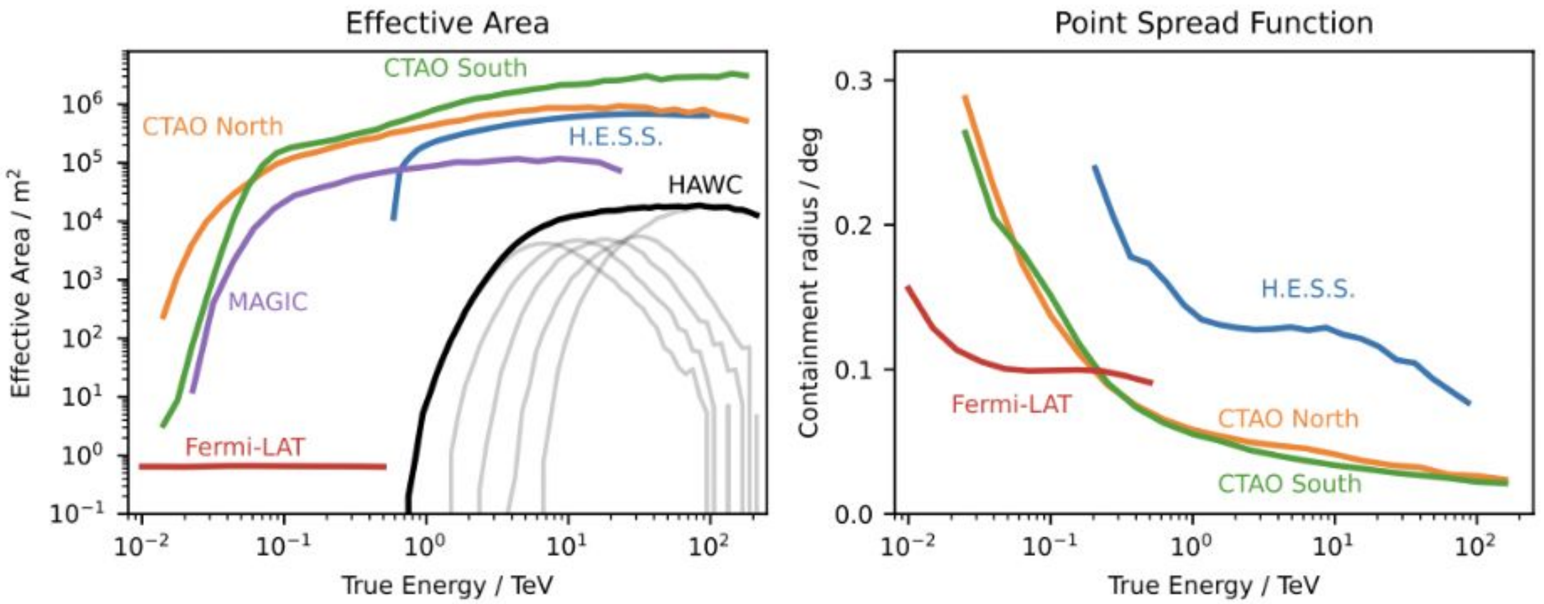
Background

- Represents the expected remaining hadronic background after gamma-hadron separation due to misclassified events
- Depends on the reconstructed energy and the offset from the pointing position



IRF visualisation

- Users can read and visualise IRFs which have been provided by various instruments and telescopes including CTA
  - We will see how to extract IRFs in this workshop



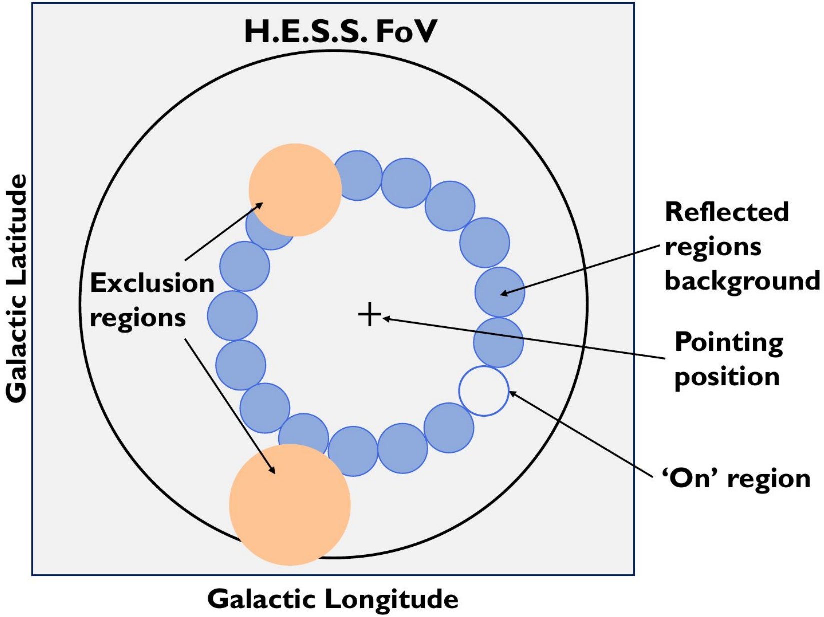
https://github.com/gammapy/gammapy-v1.0-paper/blob/main/src/figures/irfs.py

DL3 - DL4 : data reduction

1. Select and retrieve relevant observations
2. Define the reduced dataset geometry
    a. Is the analysis 1D (spectral only) or 3D?
    b. Define target binning and projection
3. Initialize the data reduction methods (makers)
    a. Data and IRF projection
    b. Background estimation
    c. Safe Mask determination
4. Loop over selected observations
    a. Apply makers to produce reduced datasets
    b. Combine them for stacked or joint analysis

1D analysis technique

- For each observation, the ‘on’ region is where gamma-ray sources are expected and the ‘off’ region contains no gamma-ray source emission
- The ‘off’ region has the same shape, size and offset from the pointing position as the ‘on’ region
- This technique is used for spectral analysis, as the spatial dimension is lost when grouping the pixels inside the ‘on’ region



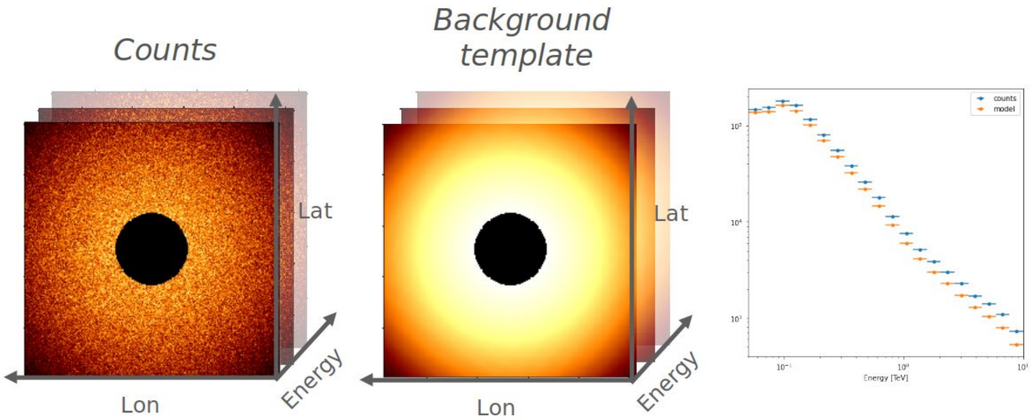
3D analysis technique

- Advanced method to estimate the background is through a 3D model
- Construct a model of acceptance, and then predict the background over the field of view (FoV) as a function of energy
- Runwise FoV background is adjusted to the counts measured outside the exclusion regions, implement a specific spectra model
  - Corrects for effects that are not taken into account during the FoV background model construction
- Below shows normalisation and tilt varies in the spectral shape for the model

$$\mathcal{B}(E) \longrightarrow \mathcal{B}(E) \times \mathtt{norm} \times \left(\frac{E}{1\,\mathrm{TeV}}\right)^{-\mathtt{tilt}}$$

Correcting background

- Background model provides the shape of the expected hadronic background and average rate
- Run-to-run variation in trigger rate, means the background needs to be corrected to the conditions of each observation run
- In general, we ensure obtain the number of counts where no sources are expected to adjust the background level
  - Normalized in regions devoid of signal



Safe Mask

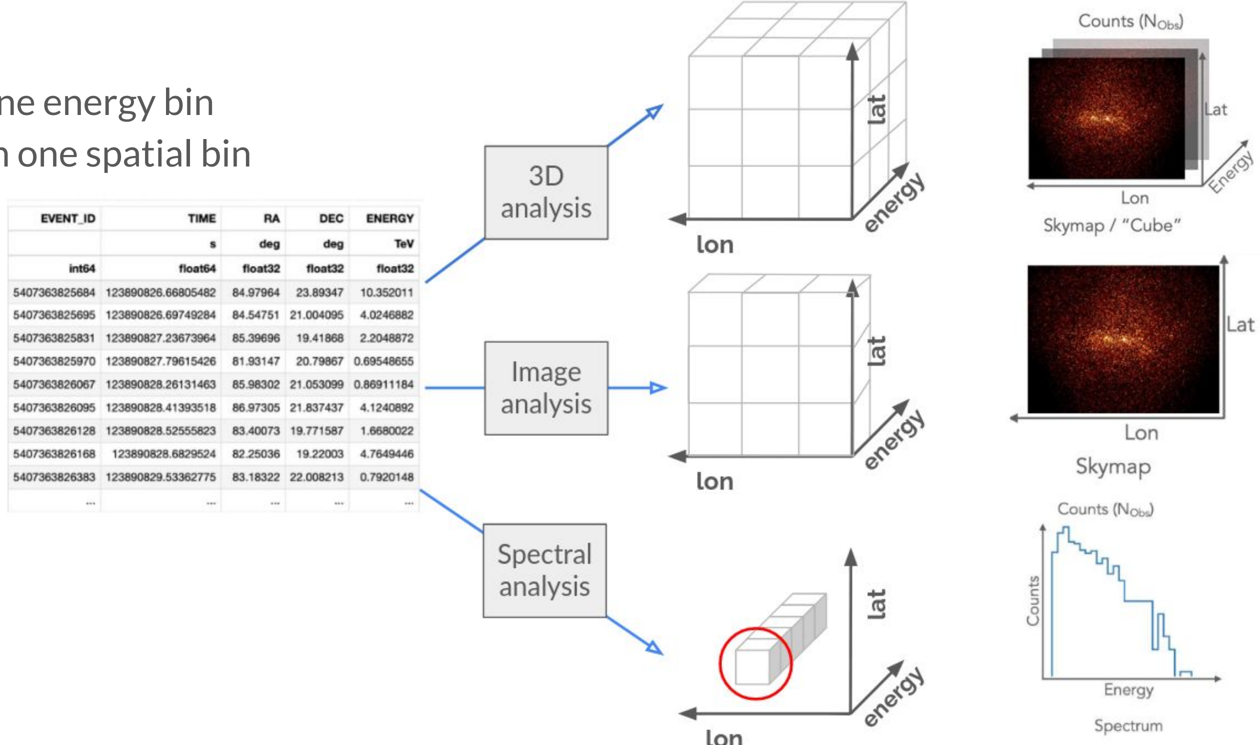
- We have various parameters that can affect our data reduction, so we must introduce a ‘Safe Mask’ – to ensure we account for this
- Two parameters that affect this are the zenith angle and offset angle

Offset angle – angle between the pointing position and the location of the source

- Maximum is typically taken to be ~2.2°
- IRF gets worse at large offsets from the pointing direction
  - Between 2° and 2.5° the instrument response is affected by systematics
  - For example, PSF asymmetries are known to arise at offsets >2°. This contradicts the assumption of radial symmetry, currently implemented in Gammapy PSF, which would potentially lead to biased fit results.

Data reduction summary DL3 → DL4

- 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 fitting DL4 → DL5

- For modeling and fitting, Gammapy relies on forward-folding:
  - Measured counts N is compared to predicted counts Npred

$$N_{\mathrm{pred}}(p, E) = \sum_{S} E_{disp}\left[PSF \star \left(expo \times \Phi_S(p_t, E_t)\right)\right] + N_{\mathrm{bkg}}(p, E)$$

- Model parameter estimation is performed through maximum likelihood technique
  - Cash statistics is used for counts data with a known background

$$TS = -2\log L = 2\sum\left(N \log N_{\mathrm{pred}} - N_{\mathrm{pred}}\right)$$

  - Wstat statistics is used for counts data with a measured background

Time for us to put this into practice!