# Gammapy – An open source Python package for gamma-ray astronomy

J. Lefaucheur[1], C. Deil[2], A. Donath[2], L. Jouvin[3], B. Khélifi[3] and J. King[2]
*[1]LUTH, [2]MPIK, [3]APC*

## Context

- The current experiments (H.E.S.S., MAGIC and VERITAS) using the Imaging Atmospheric Cherenkov Telescopes (IACTs) technic can detect gamma-rays above a few dozen of GeV
- Data and tools for their analysis are private in the IACT community. The upcoming of the open observatory Cherenkov Telescope Array (CTA) slowly begins to change the mindset of the community
- Gammapy can be used to measure source properties such as morphology, spectrum and variability, using event lists as well as instrument response function (IRF) by taking into account IACT analysis methods' specificities

## Gammapy

- Gammapy is built on the scientific Python stack and Astropy, optionally using Sherpa for modeling and fitting (see Figure 1)
- Our initial focus was to implement the common TeV analysis methods, i.e. using 2-dimensional sky images for source detection and morphology characterisation, followed by spectral analysis or light curve computation for a given source region. A cube-style analysis, with simultaneous spatial and spectral models of the gamma-ray emission, as well as background, is in development
- Additional developments and verification using data from existing Cherenkov telescope arrays, such as H.E.S.S. and MAGIC, as well as simulated CTA data is ongoing

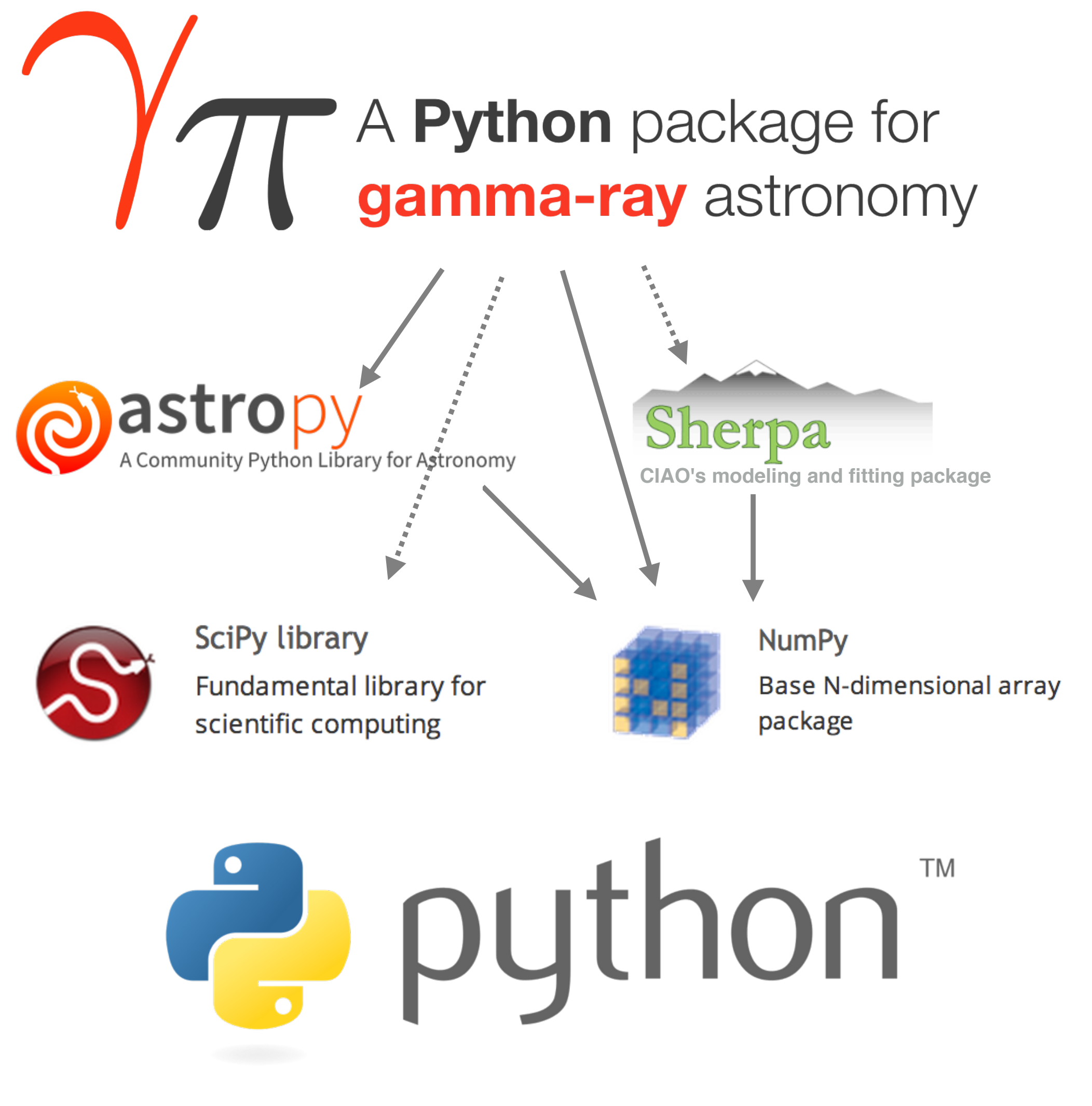

A Python package for gamma-ray astronomy
astropy
A Community Python Library for Astronomy
Sherpa
CIAO's modeling and fitting package
SciPy library
Fundamental library for scientific computing
NumPy
Base N-dimensional array package
python

*Figure 1: Gammapy is a Python package, built on Numpy and Astropy as core dependencies*

## Bibliography

- Berge et al., 2007, A&A 466, 1219–1229 (2007)

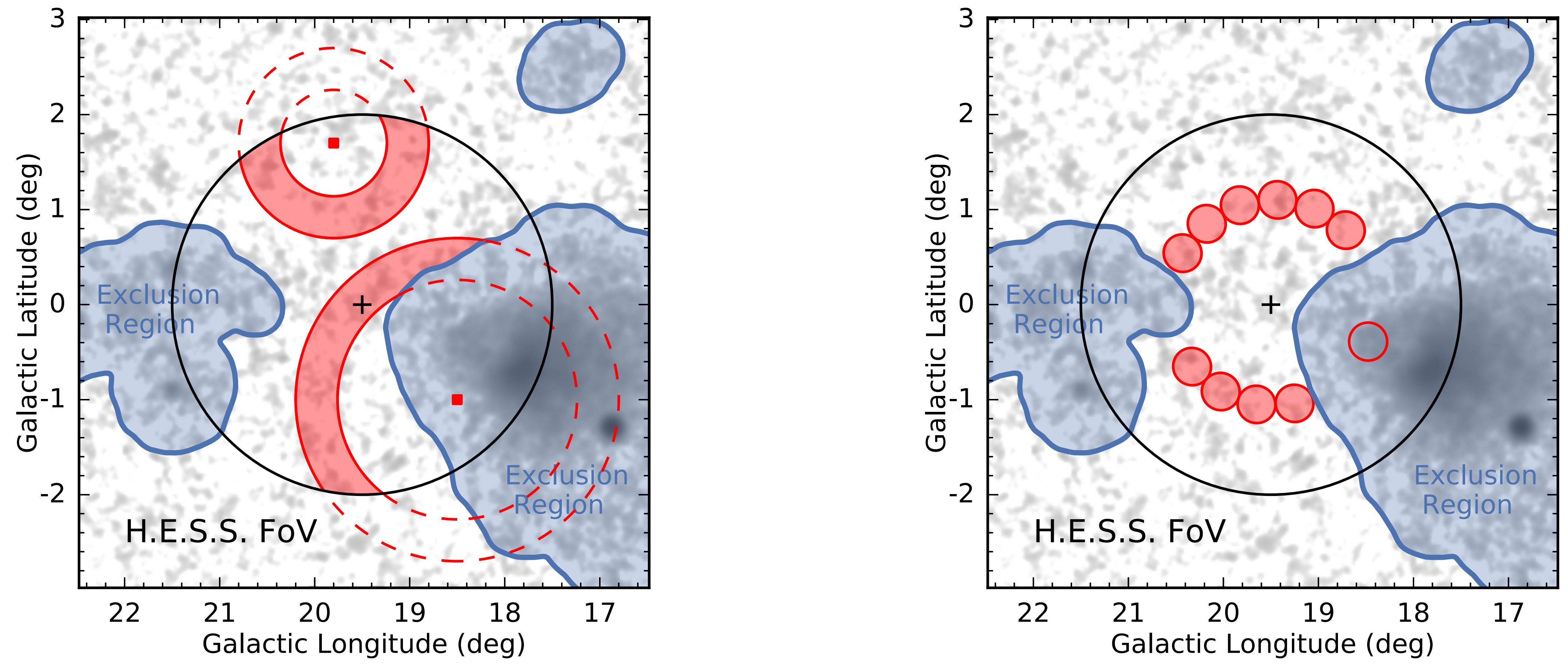


*Figure 2: Illustration of classical method used to estimate the background implemented in Gammapy. The center of the field of view (FOV) is indicated with a black cross. Left, ring-background model used to reconstruct the morphology of a source. A ring (filled red regions) is used around a trial position (red squares) to estimate the background contamination. Right, reflected-region-background model used to reconstruct the spectrum of a source. The OFF regions (filled red circles) are used to estimate the background in the ON region (empty red circle). See Berge et al. 2007 for a detailed discussion.*

## Behind the scene

- The following example code generates a counts image from an event list

```
"""Make a counts image with Gammapy."""
from gammapy.data import EventList
from gammapy.image import SkyImage
events = EventList.read(’events.fits’)
image = SkyImage.empty(
  nxpix=400, nypix=400, binsz=0.02,
  xref=83.6, yref=22.0,
  coordsys=’CEL’, proj=’TAN’,
)
image.fill_events(events)
image.write(’counts.fits’)
```

- The key point here is that all data are stored in Numpy arrays and processed efficiently via calls into existing C extensions in Numpy and Astropy. For instance, in the example, the EventList and SkyImage objects store coordinates and pixel data as Numpy arrays, respectively.
- Data processing routines such as image.fill(events) are based on Numpy histograms and calls into the CFITSIO and WCSLib C libraries.

## Project

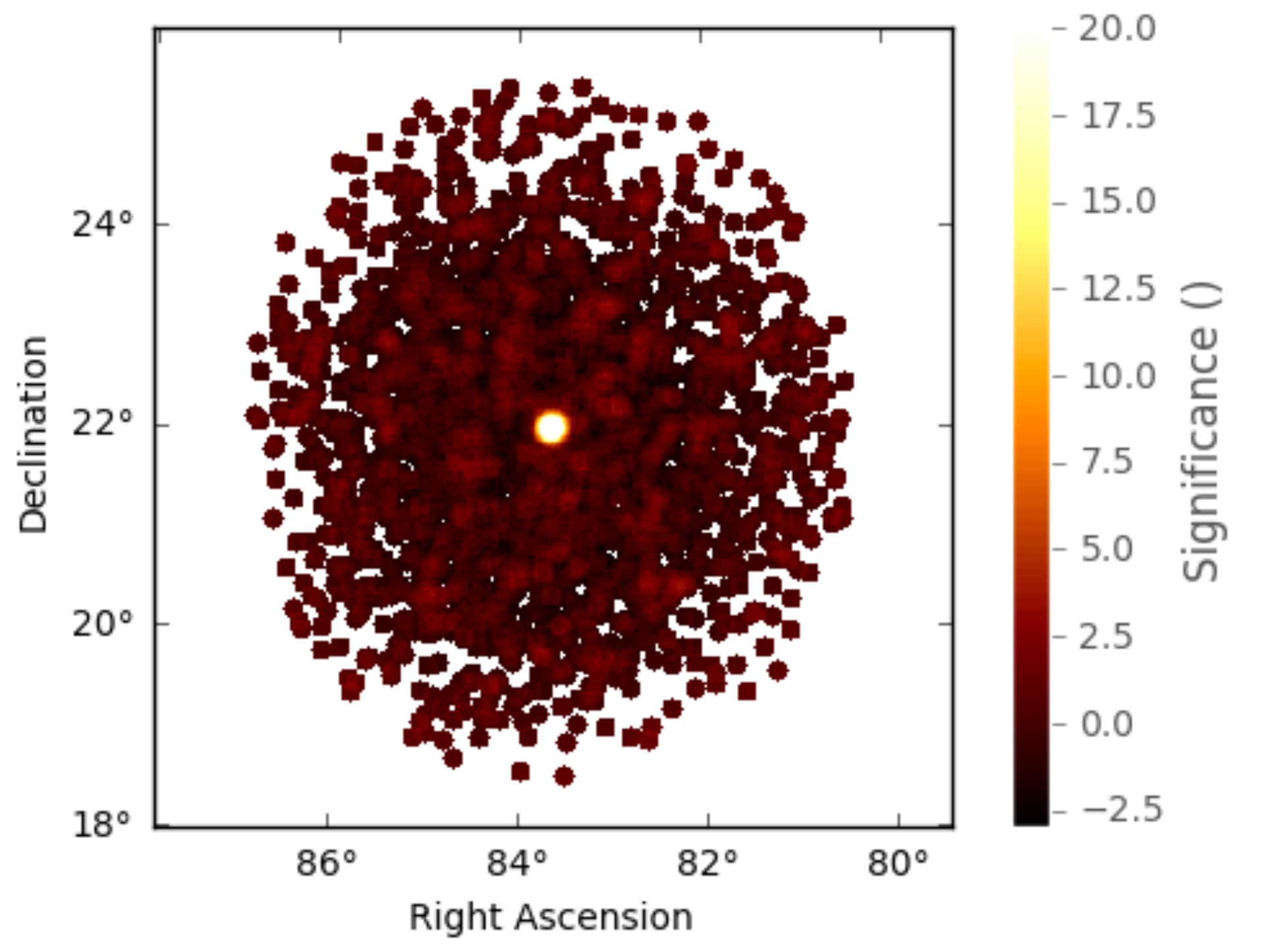
- Distributed and installed in the usual way for Python packages (pip, anaconda, Macports, etc.)
- Development happens on Github (extensive use of pull requests to discuss and review code, continuous integration of Travis-CI and Appveyor)
- Sphinx documentation and Jupyter notebooks for tutorials

## Links

- http://docs.gammapy.org/en/latest/
- https://github.com/gammapy/
- https://github.com/sherpa/

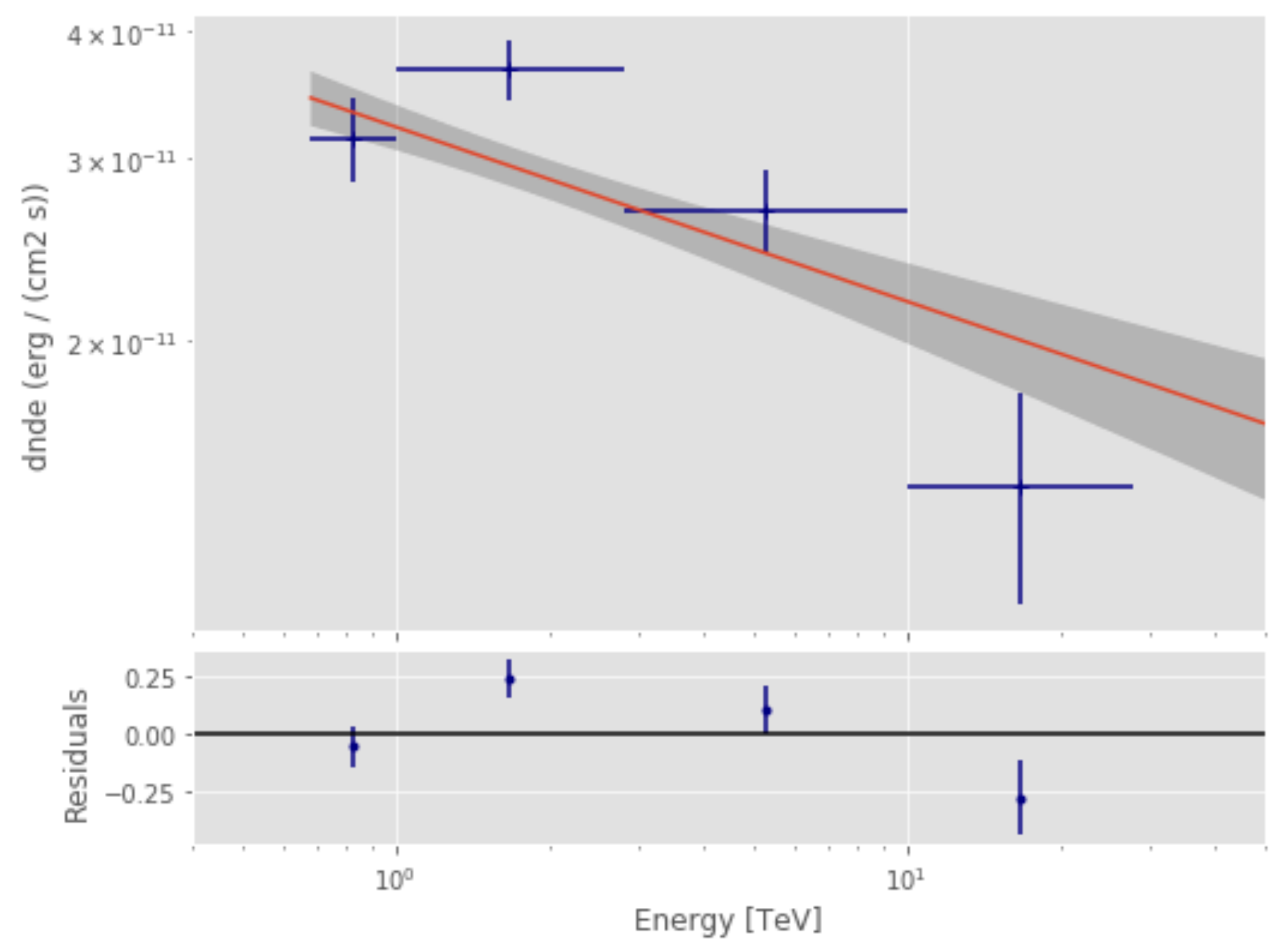
## Map example

- This figure shows an example of a significance map of simulated data for a point-like source at the position of the Crab Nebulae. Background is estimated with a ring-background model in each pixel (see Figure 2, left)



## Spectra example

- This figure shows an example of spectral reconstruction of simulated data with Sherpa. The background is estimated with reflected regions (see Figure 2, right). A forward-folding method is used (Cash statistics). The photon excess is compared to an expected excess computed with a power law hypothesis via a likelihood procedure to get the best fitted parameters of the model.


4 × 10⁻¹¹
3 × 10⁻¹¹
2 × 10⁻¹¹
dnde (erg / (cm2 s))
Residuals
10⁰
10¹
Energy [TeV]