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 Cherenkov Telescope Array (CTA) will observe the sky in very-high-energy gamma-ray light soon
- Astronomers will have access to CTA high-level data
- Gammapy can be used to measure source properties such as morphology, spectra and light curves, using event lists as well as instrument response function (IRF) and auxiliary information as input

Gammapy

- Gammapy is built on the scientific Python stack and Astropy, optionally using Sherpa or other packages 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 (Figure 2). A 3-dimensional analysis with a simultaneous spatial and spectral models of the gamma-ray emission, as well as background is in development
- Further 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

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

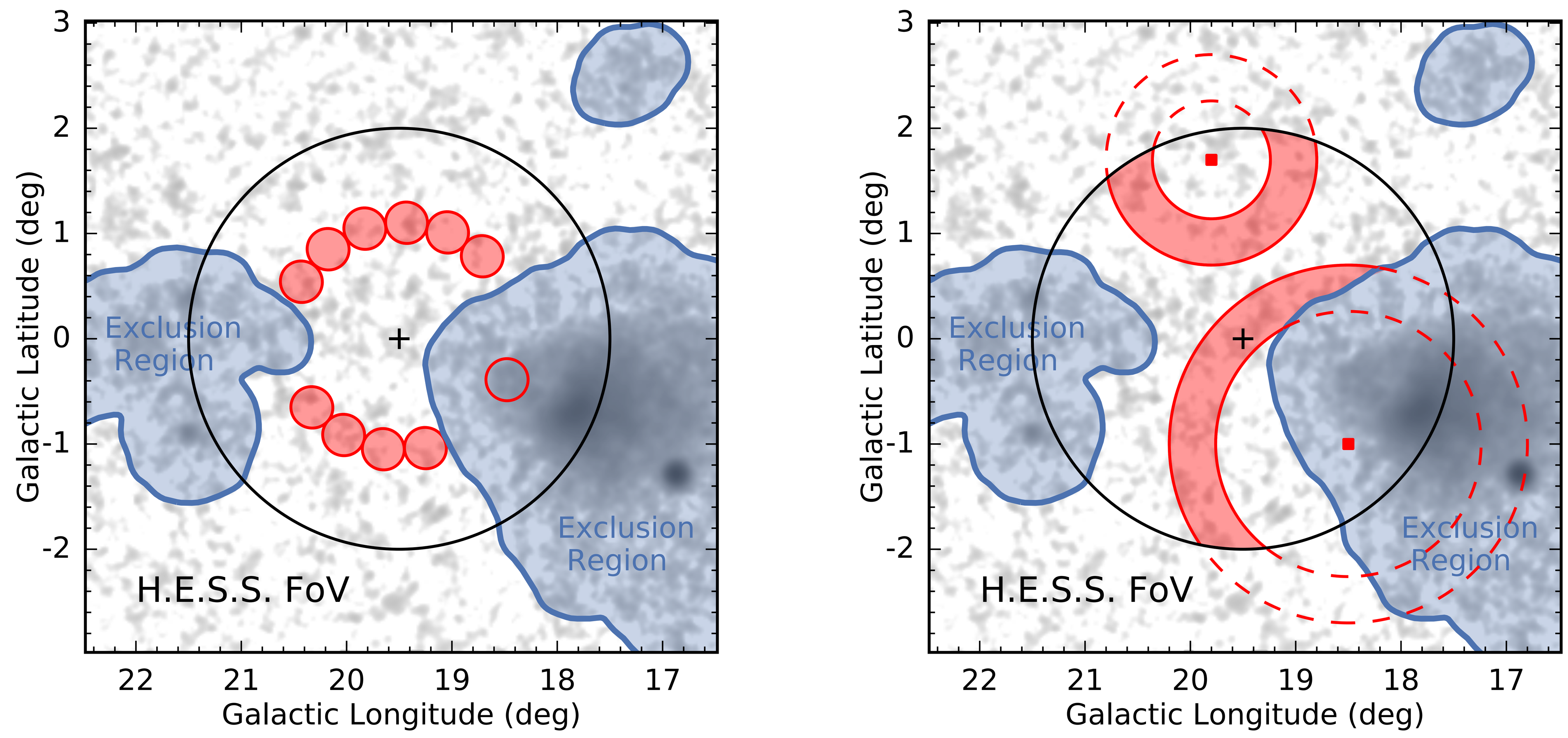


Figure 2: Illustration of classical methods to estimate the background implemented in Gammapy used in very high energy. Left, reflected regions to deal with spectral reconstruction. Right, ring background model to reconstruct the morphology of a source.

Behind the scene

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

```python
"""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')
```

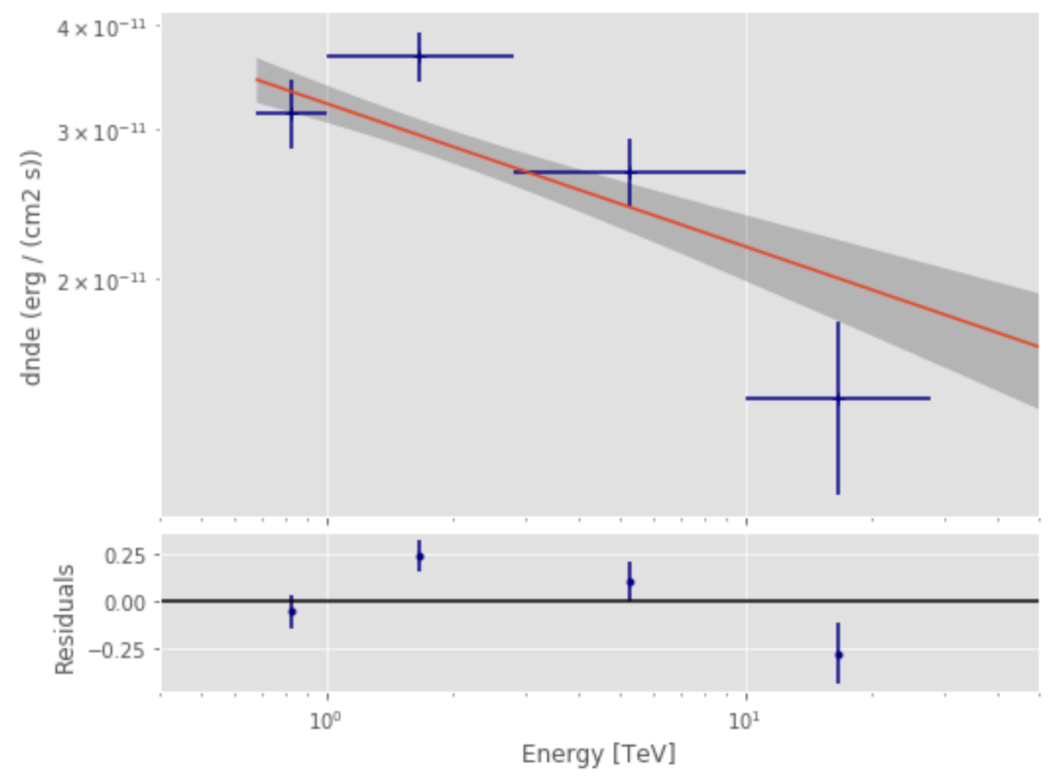
- The key point here is that all data are stored in Numpy arrays and processed efficiently via calls into existing C extensions in Numpy and Astropy. For instance, in the example, the EventList and SkyImage objects store coordinates and pixel data as Numpy arrays, respectively.
- Data processing routines 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

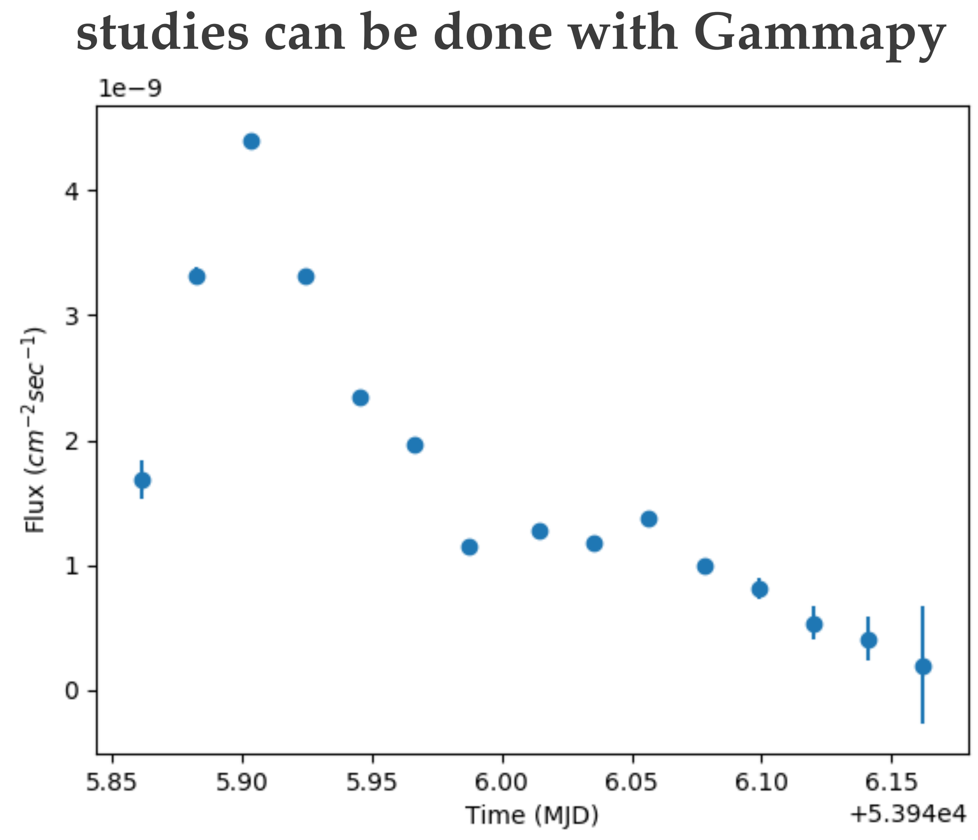
Spectra example

- This figure illustrates how data can be fitted using a full forward folding method, thanks to Sherpa



Light curve example

- This figure illustrates how variability studies can be done with Gammapy



Links

- http://docs.gammapy.org/en/latest/
- https://github.com/gammapy
- https://nbviewer.jupyter.org/github/gammapy/gammapy-extra/blob/master/index.ipynb