Gammapy: A Python package for gamma-ray astronomy

Axel Donath[7], Christoph Deil[14], Régis Terrier[3], Johannes King[23], Jose Enrique Ruiz[18], Quentin Remy[19], Léa Jouvin[30], Atreyee Sinha[27], Matthew Wood[12], Fabio Pintore[15], Manuel Paz Arribas[30], Laura Olivera[19], Luca Giunti[3], Bruno Khelifi[4], Ellis Owen[22], Brigitta Sipőcz[6], Olga Vorokh[30], Julien Lefaucheur[30], Fabio Acero[8], Thomas Robitaille[2], David Fidalgo[30], Jonathan D. Harris[30], Cosimo Nigro[16], Lars Mohrmann[19], Dirk Lennarz[1], Jalel Eddine Hajlaoui[3], Alexis de Almeida Coutinho[28], Matthias Wegenmat[30], Dimitri Papadopoulos[30], Maximilian Nöthe[26], Nachiketa Chakraborty[30], Michael Droettboom[21], Helen Poon[30], Arjun Voruganti[30], Jason Watson[9], Thomas Armstrong[30], Vikas Joshi[13], Erik Tollerud[25], Erik M. Bray[30], Domenico Tiziani[30], Gabriel Emery[30], Hubert Siejkowski[30], Kai Brügge[24], Luigi Tibaldo[17], Arpit Gogia[30], Ignacio Minaya[30], Marion Spir-Jacob[30], Yves Gallant[30], Andrew W. Chen[10], Roberta Zanin[5], Jean-Philippe Lenain[30], Larry Bradley[25], Kaori Nakashima[13], Anne Lemière[3], Mathieu de Bony[30], Matthew Craig[30], Lab Saha[7], Zé Vinicius[30], Kyle Barbary[30], Thomas Vuillaume[29], Adam Ginsburg[30], Daniel Morcuende[30], José Luis Contreras[30], Laura Vega Garcia[30], Oscar Blanch Bigas[30], Víctor Zabalza[30], Wolfgang Kerzendorf[20], Rolf Buehler[11], Sebastian Panny[30], Silvia Manconi[30], Stefan Klepser[11], Peter Deiml[30], Johannes Buchner[30], Hugo van Kemenade[30], Eric O. Lebigot[30], Benjamin Alan Weaver[30], Debanjan Bose[30], Rubén López-Coto[30], and Sam Carter[30]

(Affiliations can be found after the references)

February 13, 2022

ABSTRACT

Historically the data as well as analysis software in gamma-ray astronomy is proprietary to the experiments. With the future Cherenkov Telescope Array (CTA), which will be operated as an open gamma-ray observatory with public data, there is a corresponding need for open high-level analysis software. In this article we present the first major version v1.0 of Gammapy, a community-developed open-source Python package for gamma-ray astronomy. We present its general design and provide anoverview of the analysis methods and features it implements. Starting from event lists and a description of the specific instrument response functions (IRF) stored in open FITS based data formats, Gammapy implements . Thereby it handles the dependency of the IRFs with time, energy as well as position on the sky. It offers a variety of background estimation methods for spectral, spatial and spectro-morphological analysis. Counts, background and IRFs data are bundled in datasets and can be serialised, rebinned and stacked. Gammapy supports to model binned data usingPoisson maximum likelihood fitting. It comes with built-in spectral, spatial and temporal models as well as support for custom user models, to model e.g., energy dependent morphology of gamma-ray sources. Multiple datasets can be combined in a joint-likelihood approach to either handle time dependent IRFs, different classes of events or combination of data from multiple instruments. Gammapy also implements methods to estimate flux points, including likelihood profiles per energy bin, light curves as well as flux and signficance maps in energy bins. We further describe the generaldevelopment approach and how Gammapy integrates into ecosystem of other scientific and astronomical Python packages. We also present analysis examples with simulated CTA data and provide results of scientific validation analyses using data of existing instruments such as H.E.S.S. and Fermi-LAT .

Key words. Gamma rays: general - Astronomical instrumentation, methods and techniques - Methods: data analysis

1. Introduction

TODO: Axel and Regis write this...

Gamma-ray astronomy is a rather young field of research. By detecting and reconstructing arrival direction, time and energy of primary cosmic gamma-rays The gamma-ray sky is either observed by ground based instruments, driven by experiments with proprietary software often based on ROOT, because of the particle physics background. Such as HESS, Veritas or Magic.

The Cherenkov Telescope Array will be operated as an open observatory for the first time. Thus there is a need for open analysis software as well.

Once the primary photons are reconstructed the format of the data of all Gamma-ray instrumets can be brought into a common format. An effort is the gamma-astro data formats tbd (all contributors to the spec). The data format is based on FITS (Pence et al. 2010).

In recent years Python [1] has established as one of the standard programming languages for astronomy [2] as well as data sciences in general [3]. The success is mostly attributed to the simple and easy to learn syntax, the ability to act as a "glue" language between different programming lan-

[1] http://fits.gsfc.nasa.gov/
[2] Citation missing
[3] Citation missing

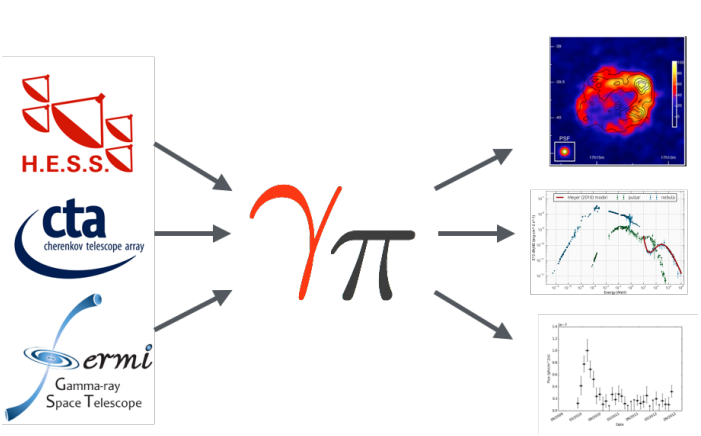

Fig. 1. Gammapy is a Python package for high-level gamma-ray data analysis. Using event lists, exposures and point spread functions as input you can use it to generate science results such as images, spectra, light curves or source catalogs. So far it has been used to simulate and analyse H.E.S.S., CTA and Fermi-LAT data, hopefully it will also be applied to e.g., VERITAS, MAGIC or HAWC data in the future.

guages, the rich eco-system of packages and the open and supportive community.

Astronomical data analysis software written in Python existed since 2000. e.g., sherpa (Refsdal et al. 2011, 2009), or for gamma-ray even PyFACT (Raue & Deil 2012).

The short-term success of Pythion lead to a prolifaration of packages, until Astropy (Astropy Collaboration et al. 2013) was created in 2012. Astropy is and Gammapy is a Python package for gamma-ray astronomy.

TODO: Figure 1: Data -> Gammapy -> Spectra etc with some details

Basic idea: build on Numpy and Astropy, use Python stack

TODO: Figure 2: Gammapy software stack

Here's a list of references I'd like to cite ... to be incorporated into the main text somewhere:

- Gammapy webpage[4]
- Naima[5] (Zabalza 2015)
- Gammapy use in science publications: (Owen et al. 2015), SNR shell, HGPS

* Gammapy – A Python package for gamma-ray astronomy * Gammapy – A prototype for the CTA science tools * Astropy: A community Python package for astronomy * THE ASTROPY PROJECT: BUILDING AN INCLUSIVE, OPEN-SCIENCE PROJECT AND STATUS OF THE V2.0 CORE PACKAGE * GammaLib and ctools * Fermipy proceedings * SunPy: Python for Solar Physics. An implementation for local correlation tracking *

2. Analysis Workflow Overview

3. Gammapy package

The Gammapy package is structured into multiple sub-packages which mostly follow the stages in the data reduction workflow.

[4] http://gammapy.org
[5] https://github.com/zblz/naima

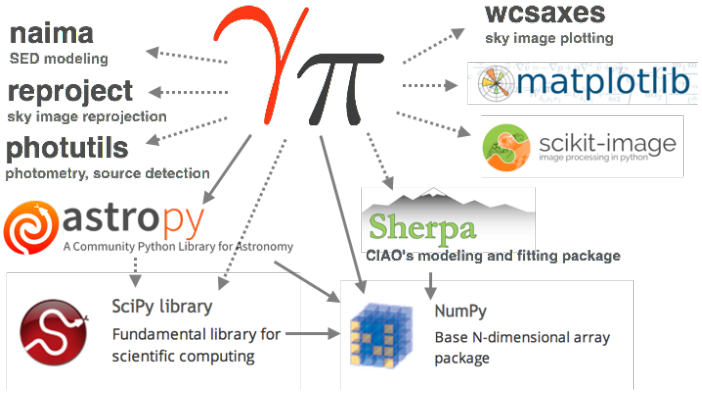

Fig. 2. The Gammapy stack. Required dependencies Numpy and Astropy are illustrated with solid arrows, optional dependencies (the rest) with dashed arrows.

3.1. Overview

Outline: * List typical analysis use cases * Can use from Python and Jupyter -> show Figure with Jupyter notebook here. * Gammapy code structure * How Numpy and Astropy is used

Figures: * Add a Figure showing dataflow in a typical application DL3 at the top, spectrum, map, lightcurve, fit results at the bottom. Mention major classes in between (DataStore, EventList, Map, MapMaker, MapFit, ...) * Probably not: Figure showing sub-packages and how they relate (gammapy.data and gammapy.irf at the base, then gammapy.maps, etc. * The code example Figure how to make a counts map, to explain how the package works.

TODO: How to sort the sub-packages? After data flow or alphabetically? What about maps?

3.2. gammapy.data

TODO: Cosimo Nigro The gammapy.data sub-package provides access to DL3 level data and observation handling.

3.3. gammapy.irf

TODO: Fabio Pintore IRF classes

3.4. gammapy.makers

TODO: Regis Terrier

The data reduction contains all tasks required to process and prepare data at the DL3 level for modeling and fitting. The gammapy.makers sub-package contains the various classes and functions required to do so. First, events are binned and IRFs are interpolated and projected onto the chosen analysis geometry. This produces counts, exposure, background, psf and energy dispersion maps. The MapDatasetMaker and SpectrumDatasetMaker are responsible for this task, see Fig 5.

Because the background models suffer from strong uncertainties it is required to correct them from the data themselves. Several techniques are commonly used in gamma-ray astronomy such as field-of-view background normalization or background measurement in reflected regions regions, see Berge et al. (2007). Specific Makers such

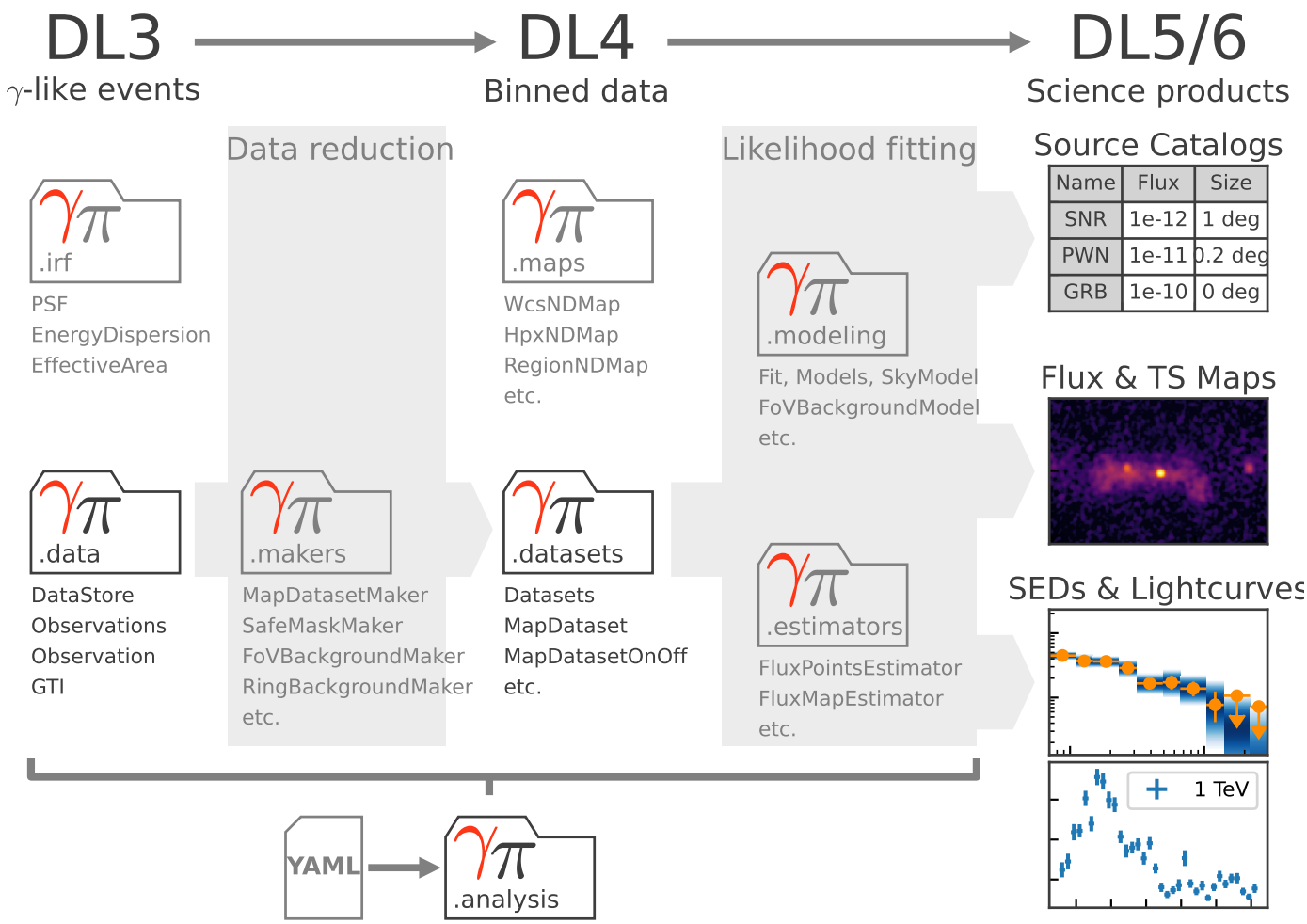

Fig. 3. Gammapy sub-package structure and data analysis workflow.

```python
from gammapy.data import DataStore

data_store = DataStore.from_dir("$GAMMAPY_DATA")
obs_ids = [1, 2, 3]
observations = data_store.get_observations(obs_ids)
```

Fig. 4. Using gammapy.data to access DL3 level data with DataStore

```python
from gammapy.makers import MapDatasetMaker, FoVBackground

maker = MapDatasetMaker()
bkg_maker = FoVBackgroundMaker()
mask_maker = SafeMaskMaker()

dataset = maker.run(dataset, observation)
dataset = mask_maker.run(dataset,observation)
dataset = bkg_maker.run(dataset,observation)
```

Fig. 5. Using gammapy.makers to reduce DL3 level data into a Dataset.

as the FoVBackgroundMaker or the ReflectedRegionsBackgroundMaker are in charge of this step.

Finally, to limit other sources of systematic uncertainties, a data validity domain is determined by the SafeMaskMaker. It can be used to limit the extent of the field of view used or to limit the energy range to e.g., a domain where the energy reconstruction bias is below a given value.

3.5. gammapy.datasets

TODO: Atreyee Sinha DL4 level data

Reduced data, IRFs, and models are bundled together to create a Data Level 4 (DL4). This is provide in the Dataset class which computes a likelihood to interface with the Fit class to perform a fitting. Corresponding to the different analysis types, five types of datasets are supported in gammapy (1). 3D data cubes (with two spatial and one energy) are represented by MapDatasets, whereas SpectrumDatasets contain binned energy information for a given on region. 2D images are a special case of MapDatasets, with only one bin in energy. The FluxPointsDataset is used to fit pre-computed flux points when no convolution with IRFs are needed.

Multiple datasets, or the same or different type, can be contained in a Datasets container. The Datasets class adds the likelihood from its constituent members to compute the full likelihood for a joint fitting.

Dataset Type	Data Type	Reduced IRFs	Geometry	Additional Quantities
MapDataset	counts	background, psf, edisp, exposure,	WcsGeom or RegionGeom	–
MapDatasetOnOff	counts	psf, edisp, exposure	WcsGeom	acceptance, acceptanc
SpectrumDataset	counts	background, edisp, exposure	RegionGeom	–
SpectrumDatasetOnOff	counts	edisp, exposure	RegionGeom	acceptance, acceptanc
FluxPointsDataset	flux	None	None	–

Table 1. Some caption.

```python
from astropy.coordinates import SkyCoord
from gammapy.maps import WcsGeom
from gammapy.datasets import MapDataset

skydir = SkyCoord("0d", "0d")
geom = WcsGeom.create(
        skydir=skydir, width="5 deg", binsz="0.2 deg"
)

dataset = MapDataset.create(
        geom=geom, name="my-dataset"
)
```

Fig. 6. Using gammapy.data to access DL3 level data with a DataStore

```python
from gammapy.modeling.models import (
        SkyModel,
        PowerLawSpectralModel,
        PointSpatialModel,
)

pwl = PowerLawSpectralModel()
point = PointSpatialModel()

model = SkyModel(
        spectral_model=pwl,
        spatial_model=point,
        name="my-model",
)
```

Fig. 7. Using gammapy.modeling.models

The total number of predicted counts for a dataset are computed as the sum of the predicted counts from all source components, plus the expected counts from the residual hadronic background. In case of normal datasets, the predicted counts from the hadronic background are computed directly from the model in reconstructed energy and spatial coordinates, whereas for OnOff datasets, the background is estimated from real off counts. The predicted counts from a source are obtained by forward folding with the IRFs, assuming that the IRFs are independent of one another.

3.6. gammapy.modeling

TODO: Quentin Remy Models and fitting

3.7. gammapy.stats

TODO: Regis Terrier Statistics methods

```python
from gammapy.datasets import MapDataset
from gammapy.estimators import TSMapEstimator
from astropy import units as u

dataset = MapDataset.read(
    "$GAMMAPY_DATA/cta-1dc-gc/cta-1dc-gc.fits.gz"
)

estimator = TSMapEstimator(
    energy_edges=[0.1, 1, 10] * u.TeV
)

maps = estimator.run(dataset)
```

Fig. 8. Using the TSMapEstimator from gammapy.estimators to compute a sqrt(TS) map.

3.8. gammapy.estimators

TODO: Axel Donath The gammapy.estimators sub-module features methods to compute flux points, light curves, flux maps, flux profiles from data.

The initial fine binning of MapDataset is grouped into larger bins.

Internal representation with a reference spectral model and an array of normalisation values given in energy, time and spatial bins.

No unfolding.

- Regrouping of dataset bins in time, energy etc.
- Reference spectral model scaled in energy bin
- "Forward folding" / "Backward folding": but there is no difference between the two, backwards folding is forward folding with one bin
- Likelihood profiles
- Uniform N-dimensional data structure
- Uniform API for plotting etc.
- FluxMaps and FluxPoints
- Serialisation to multiple formats, Astropy's Table and BinnedTimeSeries
- Additional quantities for debugging, such as predicted counts, fit convergence, sum of fit statistics
-

3.9. gammapy.analysis

TODO: Jose Enrique writes this... High level analysis API

3.10. gammapy.visualisation

TODO: Axel Donath Plotters etc.

```
from gammapy.catalog import SOURCE_CATALOGS

catalog = SOURCE_CATALOGS["3fhl"]
```

Fig. 9. Using gammapy.catalogs

3.11. gammapy.astro

TODO: Axel Donath Dark matter models, source population modelling

3.12. gammapy.catalog

TODO: Axel Donath Gamma-ray catalog access

3.13. gammapy.utils

Utility functions...

4. Applications

Each application example is a notebook in the online material: We could have one analysis as Python scripts instead of notebook in the online material. At the start of this section, point to gammapy-paper repo on Github and say that there's a Binder where people can try the examples online.

TODO: mention other application examples (joint Crab paper, HESS validation paper, HGPS, ...) here or in a subsection "other applications" at the end of this section?

4.1. Source detection

See Figure 10.
Ref: (Stewart 2009)

4.2. Multi instrument analysis

4.3. CTA simulation

CTA application example. 3D simulate and fit using public prod3 IRFs. AGN pop (Santiago), GRB pop (Thierry), PeVatrons?, GPS (Quentin) Diffuse emission + maybe a shell -> image and spectrum come out.

4.4. HESS

TODO: Catherine, Bruno

In September 2018 the HESS collaboration released a small subset of Gamma-ray data.

Maybe HESS Light curve using PKS flare from HESS

4.5. Fermi

Fermi: Galactic center, as in our notebook, same region as CTA.

5. Gammapy project

Open development, roadmap, communities, science tool aspect Infrastructure etc.

community driven vs. institutional driven

Validation and benchmarks? Validation as online appendix...

5.1. Development, testing

TODO: Jose Enrique writes this... -Github, pytest, CI, PIGs?

5.2. Documentation

- Notebooks

5.3. Software distribution and user support

- Pip, conda, versions, gammapy download

5.4. Community

TODO: Figure: Screenshot of Jupyter notebook or docs with notebook, could show the interactive maps view

```
m = Map.read("diffuse.fits")
m.plot_interactive()
```

6. Summary and Outlook

TODO: Axel and Regis write this...

Summary what we have in v0.9 and presented in this paper.

Roadmap to v1.0, about half a page.

Short conclusion: Gammapy has potential to be the Python package for gamma-ray astronomy.

Prospects for HAWC / SWGO? Or speak in general about water Cherenkov observatories...

Acknowledgements. Mention Christoph here? We would like to thank the Numpy, Scipy, IPython and Matplotlib communities for providing their packages which are invaluable to the development of Astropy. We thank the GitHub team for providing us with an excellent free development platform. We also are grateful to Read the Docs (https://readthedocs.org/), and Travis (https://www.travis-ci.org/) for providing free documentation hosting and testing respectively. Finally, we would like to thank all the Gammapy users that have provided feedback and submitted bug reports. TODO: copy over stuff from http://docs.gammapy.org/en/latest/about.html#thanks. TODO: add the ANR for Luca (and Atreyee in LUPM?)

References

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Berge, D., Funk, S., & Hinton, J. 2007, A&A, 466, 1219
Owen et al., E. 2015, PoS(SciNeGHE2014), 34 [arXiv:1506.02319]
Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., & Stobie, E. 2010, AAP, 524, A42+
Raue, M. & Deil, C. 2012, in American Institute of Physics Conference Series, Vol. 1505, American Institute of Physics Conference Series, ed. F. A. Aharonian, W. Hofmann, & F. M. Rieger, 789–792
Refsdal, B., Doe, S., Nguyen, D., & Siemiginowska, A. 2011, in 10th SciPy Conference, 4 – 10
Refsdal, B. L., Doe, S. M., Nguyen, D. T., et al. 2009, in 8th SciPy Conference, 51 – 57
Stewart, I. M. 2009, A&A, 495, 989
tbd (all contributors to the spec), A. ????, Data formats for gamma-ray astronomy : Version 1.0
Zabalza, V. 2015, ArXiv e-prints [arXiv:1509.03319]

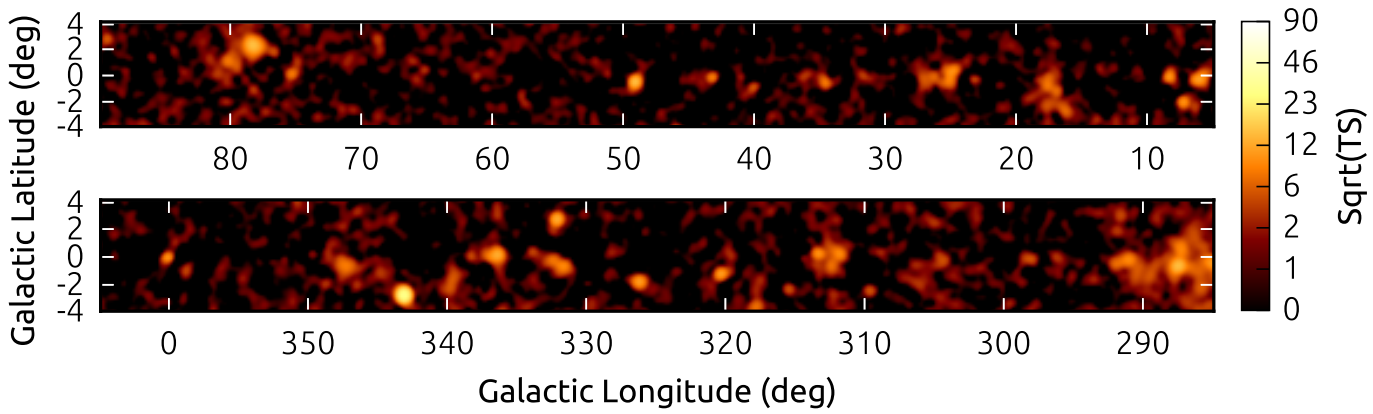

Fig. 10. Gammapy application example: A Fermi survey TS map of the inner Galactic plane region.

[1] Amazon, Atlanta GA, US
[2] Aperio Software, Insight House, Riverside Business Park, Stoney Common Road, Stansted, Essex, CM24 8PL, UK
[3] Astroparticle and Cosmology Laboratory CNRS, Université de Paris, 10 Rue Alice Domon et Léonie Duquet, 75013 Paris, France
[4] CNRS
[5] CTA Observatory gGmbH, Via Piero Gobetti 93/3 40129 Bologna, Italy
[6] California Institute of Technology, Pasadena CA, US
[7] Center for Astrophysics | Harvard & Smithsonian, CfA, 60 Garden St., 02138 Cambridge MA, US
[8] DAp/AIM, CEA-Saclay/CNRS, France
[9] DESY
[10] Department of Physics, University of the Free State, PO Box339, Bloemfontein 9300, South Africa
[11] Deutsches Elektronen-Synchrotron, DESY, Platanenallee 6 15738 Zeuthen, Germany
[12] Facebook, 1 Hacker Way, Menlo Park, CA 94025, US
[13] Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen Centre for Astroparticle Physics, Erwin-Rommel-Str. 1 91058 Erlangen, Germany
[14] HeidelbergCement AG, Berliner Straße 6, 69120 Heidelberg, Germany
[15] INAF/IASF, via Ugo La Malfa 153, 90146 Palermo, Italy
[16] Institut de Fisica d'altes Energies, IFAE, Edifici Cn, Campus UAB, 08193 Bellaterra (Barcelona), Spain
[17] Institute de Recherche en Astrophysique et Planetologie, IRAP,
[18] Instituto de Astrofísica de Andalucía - CSIC, Gta. de la Astronomía, 18008 Granada, Spain
[19] Max Planck Institut für Kernphysik, MPIK, Saupfercheckweg 1, 69117 Heidelberg, Germany
[20] Michigan State University, Department of Physics and Astronomy, 428 S Shaw Ln East Lansing, MI 48824, US
[21] Mozilla, Mountain View, CA
[22] National Tsing Hua University Institute of Astronomy, Hsinchu, Taiwan
[23] Pamyra
[24] Point 8 GmbH, Rheinlanddamm 201, 44139 Dortmund, Germany
[25] Space Telescope Science Institute, STScI, 3700 San Martin Drive, 21218 Baltimore MD, US
[26] TU Dortmund University, Otto-Hahn-Str. 4a 44227 Dortmund, Germany
[27] Universidad Complutense de Madrid EMFTEL, UCM, Plaza de la Ciencias, 28040 Madrid, Spain
[28] Universidade de São Paulo, Brazil
[29] Université Savoie Mont Blanc, CNRS, Laboratoire d'Annecyde Physique des Particules - IN2P3, 74000 Annecy, France
[30] unknown

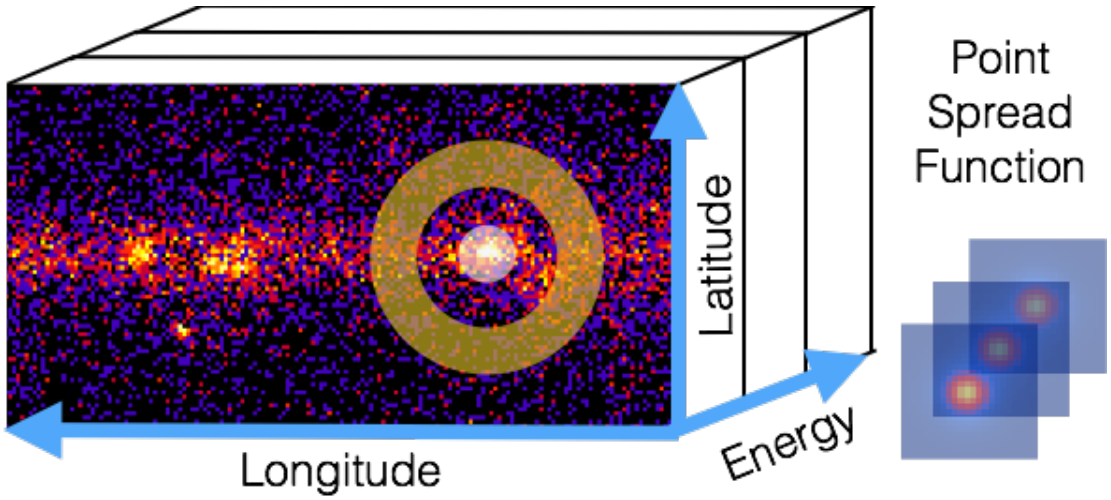

Fig. 11. Gammapy data model illustration. Binned analysis of lon-lat-energy cube data is supported via joint likelihood analysis of one image per energy bin. On-off-region based spectral analysis is supported as well.