Fermipy: An open-source Python package for analysis of Fermi-LAT Data

Matthew Wood[1], Regina Caputo[2], Eric Charles[1], Mattia Di Mauro[1], Jeffrey Magill[3], and Jeremy Perkins[2] on behalf of the Fermi-LAT Collaboration
[1]SLAC/KIPAC, [2]NASA/GSFC, [3]UMD - College Park

Abstract: *Fermipy* is an open-source python framework that facilitates analysis of data collected by the *Fermi* Large Area Telescope (LAT). *Fermipy* is built on the *Fermi ScienceTools*, the publicly available software suite provided by NASA for the LAT mission. *Fermipy* provides a high-level interface for analyzing LAT data in a simple and reproducible way. The current feature set includes methods for extracting spectral energy distributions and lightcurves, generating test statistic maps, finding new source candidates, and fitting source position and extension. *Fermipy* leverages functionality from other scientific python packages including NumPy, SciPy, Matplotlib, and Astropy and is organized as a community-developed package following an open-source development model. We review the current functionality of *Fermipy* and plans for future development.

Fermipy Package

Fermipy is a python package built on the *Fermi Science Tools* that facilitates high-level analysis of LAT data. *Fermipy* is organized around an open-source development model:

- Leverages large and growing ecosystem of open-source python libraries (Numpy, Scipy, Astropy, Matplotlib)
- Developed on GitHub [1]
- Easily installed with pip or conda package managers
- Community input and contributions accepted via Pull Request and Github Issues

See the online documentation [2] for a complete description of the package as well as tutorials.

Analysis Workflow

To perform an analysis, a user composes a YAML configuration file and python script. The analysis is controlled with an analysis state object that manages the data and model preparation and provides a set of high-level methods. A basic analysis of a source can typically be performed with a 10-20 line python script. Analysis results can be serialized to FITS or numpy files for subsequent post-processing or visualization.

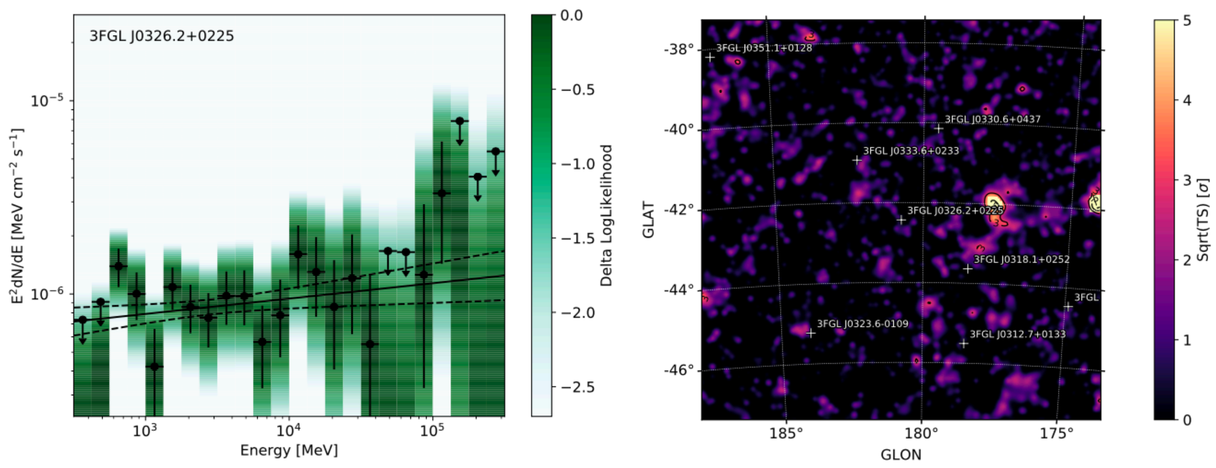
```
# Initialize the analysis object
from fermipy.gtanalysis import GTAnalysis
gta = GTAnalysis('config.yaml')
# Setup the analysis
gta.setup()
# Optimize spectral parameters
gta.optimize()
# Localize a source of interest
loc = gta.localize('Mkn421', make_plots=True)
# Extract an SED for a source of interest
sed = gta.sed('Mkn421', make_plots=True)
# Generate a TS map of the region
tsmap = gta.tsmap()
# Save the current analysis state to a file
gta.write_roi('fit0')
```

Example of a python script using *Fermipy* to an analyze the gamma-ray source Mkn 421.

Advanced Features

Fermipy provides a number of methods for more complex analysis tasks:

- Extracting Spectral Energy Distributions (sed)
- Finding new source candidates (find_sources)
- Localizing a source (localize)
- Fitting spatial extension (extension)
- Extracting a lightcurve (lightcurve)
- Generating Test Statistic (TS) and residual maps (tsmap and residmap)



Left: Spectral Energy distribution generated with the **sed** method. **Right:** Test Statistic map generated with the **tsmap** method.

Conclusions and Future Outlook

Building on the excellent ecosystem of open-source python scientific libraries, *Fermipy* provides a powerful and flexible framework for LAT data analysis. Although the feature set is now relatively mature, development of *Fermipy* is still active. Future releases will focus on the following areas:

- Support for analyzing data pixelized with HEALPix.
- Improved integration with other open-source gamma-ray analysis software such as *Gammapy* [3], *3ML* [4], and *Gammalib/ctools* [5]
- Functionality for propagating systematic uncertainties associated with the instrument response functions (effective area and PSF).
- Better standardization of FITS output formats through the *gamma-astro-data-formats* effort [6].

We aim to make *Fermipy* a community-supported effort and to this end we welcome input and contributions from all members of the LAT scientific community.

References

[1] https://github.com/fermiPy/fermipy
[2] http://fermipy.readthedocs.org/
[3] C. Deil et al. , *Gammapy – A prototype for the CTA Science Tools,* July, 2017.
[4] G. Vianello et al., *The Multi-Mission Maximum Likelihood framework (3ML)*, July, 2015, 1507.08343.
[5] J. Knödlseder et al., *GammaLib and ctools. A software framework for the analysis of astronomical gamma-ray data*, A&A 593, Aug., 2016, A1, 1606.00393.
[6] https://gamma-astro-data-formats.readthedocs.io

Acknowledgements

The *Fermi*-LAT Collaboration acknowledges support for LAT development, operation and data analysis from NASA and DOE (United States), CEA/Irfu and IN2P3/CNRS (France), ASI and INFN (Italy), MEXT, KEK, and JAXA (Japan), and the K.A. Wallenberg Foundation, the Swedish Research Council and the National Space Board (Sweden). Science analysis support in the operations phase from INAF (Italy) and CNES (France) is also gratefully acknowledged.