**easyFermi**: a graphical interface for performing *Fermi*-LAT data analyses

Raniere de Menezes\textsuperscript{1,2}\textsuperscript{*}

\textsuperscript{1}Universidade de São Paulo, Departamento de Astronomia, Rua do Matão, 1226, São Paulo, SP 05508-090, Brazil
\textsuperscript{2}Lehrstuhl für Astronomie, Universität Würzburg, Emil-Fischer-Strasse 31, 97074 Würzburg, Germany

*Submitted to Astronomy and Computing on 07/March/2022, accepted on 22/June/2022*

**Abstract**

Since its launch in 2008, the *Fermi* Large Area Telescope (LAT) allowed us to peek into the extremely energetic side of the Universe with unprecedented sensitivity and resolution. The tools available for analyzing *Fermi*-LAT data are the *Fermitools* and *Fermipy*, both of which can be scripted in Python and run via command lines in a terminal or in web-based interactive computing platforms. In this work, we are providing the community with *easyFermi*, an open-source user-friendly graphical interface for performing basic to intermediate analyses of *Fermi*-LAT data in the framework of *Fermipy*. With *easyFermi*, the user can quickly measure the γ-ray flux and photon index, build spectral energy distributions, light curves, test statistic maps, test for extended emission and even relocalize the coordinates of γ-ray sources. The tutorials for *easyFermi* are available on YouTube and GitHub, allowing the user to learn how to use *Fermi*-LAT data in about 10 min.

**Keywords**: methods: data analysis, methods: miscellaneous, gamma rays: observations

\textsuperscript{*}E-mail: raniere.m.menezes@gmail.com
1. Introduction

The Large Area Telescope (LAT) onboard the Fermi Gamma-ray Space Telescope (Fermi) is a pair-conversion telescope with imaging and spectroscopic capabilities, a large field-of-view of ∼ 2 sr, and sensitive to γ-rays in the energy range from ∼ 20 MeV to ∼ 1 TeV [1]. The Fermi-LAT performs an all-sky survey every ∼ 3 hours and, since its launch in 2008, has collected more than 13 years of data.

Once the events detected by the Fermi-LAT are classified, the collected γ-ray data are released to the public on the Fermi Science Support Center (FSSC) data server[2]. The users of these data can rely on FermiTools[3] and Fermipy[4] to perform their analyses, both tools relying on several online tutorials[5]. Fermipy, in particular, is a high-level Python package that facilitates the analysis of Fermi-LAT data in the framework of the Fermitools.

In this work we present easyFermi, an open-source graphical interface suited to perform Fermi-LAT data analyses of point-like and extended γ-ray sources. easyFermi facilitates the experience and can save a substantial amount of time for those doing γ-ray astronomy, especially to scientists just starting in the field of high-energy astrophysics. Here we provide an introduction to what is possible to do with easyFermi and how to use its graphical interface. Further tutorials can be found online on GitHub[6] and YouTube[7]. This paper is organized as follows. We describe the installation and setup processes of easyFermi in [2], and detail what is happening behind the scenes, especially the dependency of easyFermi on Fermipy, in [3]. In [4] we show some of the main data products of easyFermi and, finally, we conclude in [5].

2. Installation and setup

The current (and upcoming) release of easyFermi (i.e. V1.0.7) is available in the Python Package Index (PyPI) server[8] and GitHub, together with detailed instructions for installation and usage. The installation of easyFermi (V1.0.7) requires an existing installation of the Fermitools V2.0.8 and Fermipy V1.0.1 (Python 3 version) and works on Linux and Mac operational systems. The graphical interface will be maintained such that its installation will be compatible with the upcoming releases of the Fermitools and Fermipy.

In summary, once the Fermitools and Fermipy are installed (see the online tutorial in the easyFermi GitHub webpage), one can use the terminal to open the fermi environment with conda by typing:

```
$ conda activate fermi
```

And then simply type:

```
$ pip install easyFermi
```

To test if the installation is properly working, the user can type:

```
$ python
>> import easyFermi
```

The graphical interface of easyFermi should appear at this point (see Figure [1]) and the user can enter the desired configurations.

To exemplify the usage of the software, we will use the blazar PG 1553+113 as a test target. The first step is to download the spacecraft and photon data files from the FSSC server[9], as well as the Galactic (gll_ies_v07)
Figure 1: The main window of easyFermi.
and isotropic (iso_P8R3_SOURCE_V3_v1) diffuse emission models. The parameters that we adopt to download the Fermi-LAT data for PG 1553+113 are:

- Coordinates: 238.92935, 11.19010 (°)
- Search radius: 10 (°)
- Observation dates: 2008-08-04 15:43:36, 2009-08-04 00:00:00 (Gregorian)
- Energy range: 1000, 500000 (MeV)

For a standard point-source analysis of PG 1553+113, the user only needs to feed the empty boxes of easyFermi with the coordinates (i.e. 238.92935, 11.19010), desired energy range (1000, 500000), the directory where the spacecraft, photon and background files were downloaded, and the adopted time interval (from 2008-08-04 15:43:36 to 2009-08-04 00:00:00). For the users with some experience on Fermi-LAT data analysis, there are a set of advanced configurations that can also be controlled, but all of them are optional. Since PG 1553+113 is listed in 4FGL, the configuration step is finished. For targets not listed in 4FGL (or 3FGL), the user has to change the “Target cataloged” entry to “No” and insert a nickname for the target in the box “Target name/tag”.

The user can then simply choose what are the desired outputs by checking the Light curve, Spectral Energy Distribution (SED), Extension, Re-localize and test statistic (TS) map boxes. The results will then be saved as “.txt”, “.npy”, “.fits”, and “.pdf” (or “.png”) files in the selected output directory. If no output directory is chosen, easyFermi will save all files in the current working directory. The analysis will start after pressing “Go!”, however, the user can still make modifications on the fly, like activating or deactivating one of the boxes in the Science panel. For more complex analysis, we refer the user to the online tutorials.

3. Behind the scenes

The graphical interface and data analysis provided by easyFermi strongly depend on the Python packages PyQt5 and Fermipy, although requiring minimum maintenance. Once the user sets the desired configurations and start the process, easyFermi automatically organizes all these information with PyQt5 and communicates them to Fermipy, thereafter starting a binned likelihood analysis. This process generates several data subproducts, as a list of selected events, a counts cube, an exposure map, a livetime cube, and a source map.11

Before fitting the model to the data in the region of interest (ROI), we first optimize the ROI model with the Fermipy function optimize() to ensure that all parameters are close to their global likelihood maxima, and then look for uncataloged sources with the function find_sources() (this one can be disabled by the user). The model is then fitted to the data using the minimizer MINUIT and the main results for the target are saved in the file “Target_results.txt”. For the full set of results including all sources in the ROI, the user can access the files “Results.fits” or “Results.npy”.

3.1. Current spectral models

In easyFermi we use the spectral models from the Fermi-LAT third and fourth source catalogs [3FGL and 4FGL, respectively;3,2], namely i) the power-law model, defined by

\[
\frac{dN}{dE} = N_0 \left( \frac{E}{E_0} \right)^\gamma,
\]

where \(N_0\) is the normalization (in units of cm\(^{-2}\) s\(^{-1}\) MeV\(^{-1}\)), \(E\) is the photon energy, \(E_0\) is the pivot energy, and \(\gamma\) is the photon index; ii) the log-parabolic model \[9\] defined by

\[
\frac{dN}{dE} = N_0 \left( \frac{E}{E_0} \right)^{-\alpha - \beta \log(E/E_0)},
\]

11More details on these data subproducts here: https://fermi.gsfc.nasa.gov/ssc/data/access/lat/BackgroundModels.html

[3]
[2]
[9]
where $\alpha$ and $\beta$ are indexes describing the hardness and curvature of the spectrum; and iii) the power-law with a super exponential cutoff

$$\frac{dN}{dE} = N_0 \left( \frac{E}{E_0} \right)^\gamma \exp(-aE^b),$$

where $a$ and $b$ are the exponential factor and index describing the shape of the spectral cutoff. If the target is listed in one of the Fermi-LAT catalogs, easyFermi automatically uses the cataloged spectral model in the analysis, otherwise a new source with a power-law spectrum is added to the ROI model. The user can always change the spectral model for the target under the checkbox “Change model”. Depending on the feedback by the users of easyFermi, we can add more spectral models in the next software releases.

3.2. Automatic configuration

For those using the Standard mode of easyFermi, the following set of configurations apply.

- The ROI is defined as an $L \times L$ square with size depending on the adopted starting energy ($E_{\text{min}}$), i.e. $L = 15^\circ, 12^\circ$ or $10^\circ$ for $E_{\text{min}} < 500$ MeV, 500 MeV $\leq E_{\text{min}} < 1000$ MeV and $E_{\text{min}} \geq 1000$ MeV, respectively.

- For the same energy ranges defined in the previous item, we set the maximum zenith angle to $z_{\text{max}} = 90^\circ, 100^\circ$ or $105^\circ$.

- The classification and point spread function type for each event are filtered with $\text{evclass} = 128$ and $\text{evtype} = 3$, while the adopted instrument response function is P8R3_SOURCE_V3 and the dataset is divided in 8 bins per energy decade.

- The radius from the ROI center in which the parameters of the sources are allowed to vary during the fit is defined as $R_{\text{free}} = L/2$, where $L$ is the size of the ROI.

The experienced user can change all of these configurations by passing a customized configuration file to easyFermi under the selection of the “custom” button and by modifying the entries in the “Advanced configurations” box (see Figure 1).

3.3. Recovering the state of the analysis

A very useful characteristic of easyFermi is that it allows the user to recover the latest state of the analysis, such that the user can quit easyFermi and continue the exact same analysis some time later. Once the analysis is done, the state of the graphical interface is automatically saved in the file “GUIstatus.npy” in the output directory, and can easily be recovered by clicking on “Load GUI state” in the “Menu” button of the toolbar.

4. Main data products

The main results from easyFermi are the measurements of the flux and spectral shape for all sources in the ROI, which are saved in the files “Target_results.txt”, “Results.npy”, and “Results.fits”. Furthermore, the user can relocalize the target (the ROI is then updated with the new location of the target), compute a light curve, a $\gamma$-ray spectrum, look for extended emission and compute a TS map.

In Figures 2, 3, and 4 we show the $\gamma$-ray spectrum and light curves for PG 1553+113 built with the dataset described in §2. Upper-limits are displayed whenever an energy or time bin has $\text{TS} \leq 9$. For the extension, since PG 1553+113 is a point-like source, the peak in the delta log-likelihood shown in Figure 5 is centered at zero.

All of these results are saved in files named “SOURCE-NAME_task.fits” and “SOURCENAME_task.npy”, where “task” here can be “loc”, “lightcurve”, “SED”, or “extension”. These files can easily be accessed with numpy [8], astropy [4, 5] and TopCat [10]. Furthermore, easyFermi automatically plots the results as .pdf or .png figures labeled as, e.g. “Quickplot_task.pdf”.

4.1. Goodness of fit

The detection, flux determination and spectral modeling of Fermi-LAT sources with easyFermi is accom-
The results obtained with easyFermi are exactly the same as those obtained with Fermipy, since easyFermi is actually running Fermipy in the background. We tested this by performing several analyses (e.g., the light curves, SEDs and TS maps for ROIs centered on PG1551+113, 3C 273, 3C279, Omega Centauri, and the extended lobes of Centaurus A) with easyFermi and comparing the results with pure Fermipy analyses. In terms of performance, an analysis done with easyFermi takes basically as much CPU time and RAM as an analysis done directly with Fermipy on the terminal, however, easyFermi is slightly faster and uses less RAM than an analysis performed with Fermipy in web-based interactive computing platforms. To test if easyFermi is properly working, we recommend the user to follow the tutorial on GitHub/YouTube\textsuperscript{13}. Furthermore, the user can check if both Fermipy and easyFermi are working smoothly by simply typing

\begin{verbatim}
$ python
>> from fermipy.gtanalysis import GTAnalysis
>> import easyFermi
\end{verbatim}

If both modules are successfully imported, the installation is fine.

The maintenance of easyFermi will be done such that the users can ask for the repair of possible issues via GitHub and in a way that we guarantee its compatibility with the upcoming releases of the Fermitools and Fermipy.

\textsuperscript{12}https://www.youtube.com/channel/UCeLCfEoWasUKky6CPNN_opQ
\textsuperscript{13}https://github.com/ranieremenezes/easyFermi

5. Validation, performance, and maintenance

The γ-ray spectrum for PG 1553+113 fitted with a log-parabolic model.

Figure 2: The γ-ray spectrum for PG 1553+113 fitted with a log-parabolic model.
Figure 3: Photon flux light curve for PG 1553+113.

Figure 4: Energy flux light curve for PG 1553+113.
6. Conclusions

The easyFermi graphical interface is a user-friendly tool that allows astronomers from all niches to use Fermi-LAT $\gamma$-ray data. This tool is especially indicated for those scientists just starting in the field of high-energy astrophysics, and can be used for several goals like building a light curve, computing a $\gamma$-ray spectrum, or looking for extended $\gamma$-ray emission. Furthermore, easyFermi is meant to be simple. For more complex types of analyses, we refer the user to Fermipy and Fermitools.

The tutorials and source code for easyFermi can be found online\(^{14}\) and allow the user to learn how to use Fermi-LAT data in just a few minutes. Bug reports and proposals for new functionality should be made through the GitHub issue tracker. Although minimum maintenance is required, we aim to keep easyFermi updated and in synergy with Fermipy, hence new versions of the tool can be released in the future. We also plan to make a similar graphical interface for the Cherenkov Telescope Array\(^{17}\) and possibly for other Imaging Atmospheric Cherenkov Telescopes in the near future.

Acknowledgements

I would like to thank the anonymous referees for their suggestions and comments, as well as Alessandra Azzollini, Clodomir Vianna, Douglas Carlos, Fabio Cafardo, Kaori Nakashima, Lucas Costa Campos, Lucas Siconato, Raí Menezes, Rodrigo Lang, and Romana Grossova for installing, testing, and helping me with the development of easyFermi. Part of this project was also supported by the European Research Council for the ERC Starting grant MessMapp, under contract no. 949555.

Data Availability

All data used to exemplify the usage of easyFermi in this work can be found online in the Fermi-LAT data server\(^{15}\).

References

[1] Abdo, Aous A., Ackermann, M., Ajello, M. et al. 2009, ApJS, 183, 46.
[2] Abdollahi, S., Acero, F., Ackermann, M. et al. 2020, Fermi large area telescope fourth source catalog, ApJS, 247, 33.
[3] Acero, F., Ackermann, M., Ajello, M. et al. 2015, Fermi large area telescope third source catalog, ApJS, 218 (2), 23.
[4] Astropy Collaboration, Robitaille, Thomas P., Tollerud, Erik J., Greenfield, P. et al. 2013, Astropy: A community Python package for astronomy, Astronomy & Astrophysics, 558, A33.
[5] Astropy Collaboration, Price-Whelan, A.M., Sipőcz, B.M., Günther, H. M. et al. 2018, The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package, The Astronomical Journal, 156 (3), 123.

\(^{14}\)https://github.com/ranieremenezes/easyFermi

\(^{15}\)https://fermi.gsfc.nasa.gov/cgi-bin/ssc/LAT/LATDataQuery.cgi
[6] Atwood, W.B., Abdo, Aous A., Ackermann, M. et al. 2009, The large area telescope on the Fermi gamma-ray space telescope mission, ApJ, 697 (2), 1071.

[7] Bernlöhr, K., Barnacka, A., Becherini, Y. et al. 2013, Monte Carlo design studies for the Cherenkov telescope array, Astroparticle Physics, 43, 171.

[8] Harris, Charles R., Millman, K. Jarrod, van der Walt, Stéfan J. et al. 2020, Array programming with NumPy, Nature, 585, 7825.

[9] Massaro, E., Perri, M., Giommi, P., Nesci, R. 2004, Log-parabolic spectra and particle acceleration in the BL Lac object Mkn 421: Spectral analysis of the complete BeppoSAX wide band X-ray data set, Astronomy & Astrophysics, 413 (2), 489.

[10] Taylor, M.B. 2005, TOPCAT & STIL: Starlink Table/VOTable Processing Software, Astronomical Data Analysis Software and Systems XIV, Astronomical Society of the Pacific Conference Series, 347, 29.

[11] Wood, M., Caputo, R., Charles, E. et al. 2017, Fermipy: An open-source Python package for analysis of Fermi-LAT Data, 35th International Cosmic Ray Conference (ICRC2017), 301, 824.