Enrico : a Python package to simplify Fermi-LAT analysis

D.A. SANCHEZ[1], C. DEIL[2],

[1] *Laboratoire d'Annecy-le-Vieux de Physique des Particules, Université de Savoie, CNRS/IN2P3, F-74941 Annecy-le-Vieux, France*
[2] *Max-Planck-Institut für Kernphysik, P.O. Box 103980, D 69029 Heidelberg, Germany*

david.sanchez@lapp.in2p3.fr

Abstract: With the advent of the Large Array Telescope (LAT) on board the Fermi satellite, a new window on the Universe has been opened. Publicly available, the Fermi-LAT data come together with an analysis software named ScienceTools (ST, http://fermi.gsfc.nasa.gov/ssc/data/analysis/software/) which can be run through a Python interface. Nevertheless, for the user, the ST can be hard to run and imply several steps. Users already contributed with scripts for a specific task but no tool allowing a complete analysis is currently available.
We present a Python package called `Enrico`, designed to facilitate the data analysis. Using only configuration files and front end tools from the command line, the user can easily perform/reproduce an entire Fermi analysis and make plots for publications. It also include new features like debug plots, pipeline execution on one or several CPUs, downloading of the Fermi data or the generation of a sky model from the Fermi catalogue.
`Enrico` is an open-source project currently available for download at `https://github.com/gammapy/enrico`.

Keywords: *Fermi*, Software, Analysis tool

1 Introduction

One year after the launch of the *Fermi* satellite, the data gathered by the large Area Telescope (LAT) together with the software (ScienceTools, ST) were publicly released. Today more that 4 years of data were taken and a lot of discoveries were made.

LAT data can be analysed using the ST which are a collection of command line tools. The use of these tools requires some expertise and detailed tutorials [1] have been written by the LAT team.

While very complete, the ST lack several important features, e.g. spectrum and light-curve generation, etc..., and can be difficult to configure, hard to use on CPU clusters to take advantage of the availability of several computation cores. Based on the ST Python interface, the `Enrico` package implements such features and simplifies the data analysis.

2 Features

The main features of this package can be summarised as follows:

- It uses the ST to analyse the LAT data (`FITS` file production, minimisation, etc...). The `Enrico` command line tools are just front-ends for functions and classes in the `Enrico` Python package.
- Analysis is simplified, `Enrico` generates `Xml` model files, `FITS` files and paper-quality plots with a few commands.
- Results are reproducible, because configuration files and logs are used.
- Good default options that are suitable for most analyses.
- Control plots were added to ensure the reliability of the analysis.

Written in Python, the code is portable and does not rely on the CPU architecture nor on the ST version. Documentation, tutorial and instructions for installation are available at `http://enrico.readthedocs.org`.

The `Enrico` package includes a job submission module allowing some parallelisation of the tasks, mainly for spectra (energy bins) and light curves (time bins). Currently the MPIK[2] and LAPP[3] clusters are supported but others can easily be added by the user.

3 A simple analysis

3.1 Prepare and run your analysis

Few steps need to be done before actually running the *Fermi* analysis. The command-line tools work with a configuration file that needs to be generated. This is done by: `> enrico_config PKS2155.conf` where `PKS2155.conf` is the name of the configuration files that will be created. For this exemple, the blazar PKS 2155-304 has been analysed using data from MET=239557418 to MET=271093418 with an energy range from 100 MeV to 300 GeV.

Few questions will be asked about the analysis to perform (source name, position, time, energy range, etc...). The *Fermi* data analysis uses a sky model written in an xml file. The package reads the 2FGL catalogue [?] to generate a coresponding xml model using the tool :

```
> enrico_xml PKS2155.conf
```

The produced xml file contains all the sources with the user-defined region of interest plus 10 degrees. Only sources within 3 degrees around the source have their parameters free to vary. Several spectral models are supported (PowerLaw, PowerLaw2, LogParabola, PLExpCutoff, Generic). At

1. see `http://fermi.gsfc.nasa.gov/ssc/data/analysis/`
2. `http://www.mpi-hd.mpg.de/`
3. `http://lapp.in2p3.fr/`

this point, the likelihood analysis implemented in `gtlike`, can be simply performed with the command :

`enrico_sed PKS2155.conf`

A full spectrum and also data points (see Fig. **??**) are then computed. The covariance matrix obtained during the minimisation process is then used to compute the 68% error contour also called butterfly. If the source is not detected (i.e. below the user-defined TS), an upper limit is computed.

Lightcurves (Fig. **??**) and TS maps can be produced by `Enrico` using :

`> enrico_lc PKS2155.conf`

`> enrico_tsmap PKS2155.conf`

The previous commands run several jobs either in parallel by submitting them to a cluster or sequentially one after the other. A single job is very similar to a spectrum calculation. `FITS` files that are need by `gtlike` are produced and the minimisation is performed either in time bins (light-curves) or in space bins (TS map). For each tool, the generation of the `FITS` files can be skipped if they have already been produced in a previsous analysis. This allows to save CPU time in the cases of recomputation of the best-fit values after a change in the sky model for exemple.

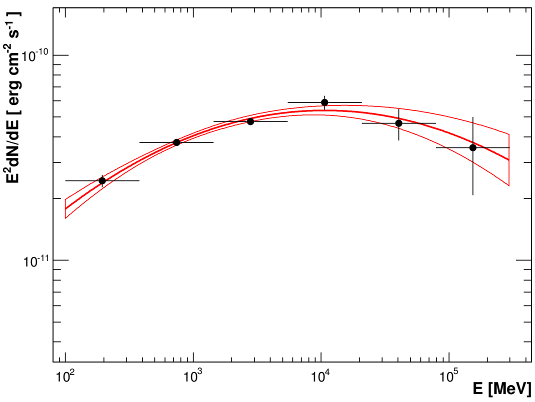

Fig. 1: Spectral energy distribution of the Blazar PKS 2155-304 as obtained with `Enrico` and the tools `enrico_sed` and `enrico_plot_sed`.

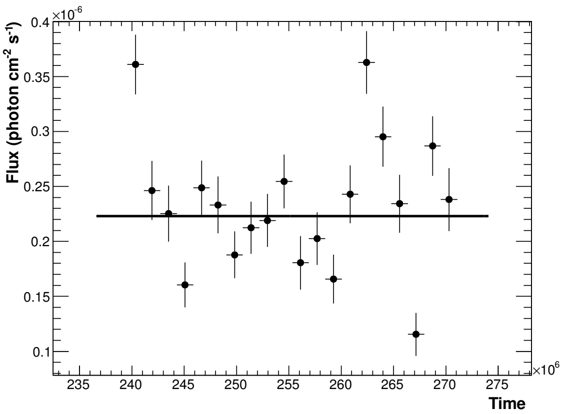

Fig. 2: light-curve of the Blazar PKS 2155-304 as obtained with `Enrico` and the tools `enrico_lc` and `enrico_plot_lc`.

```
[target]
        name = PKS2155
        ra = 329.7
        dec = -30.2
        spectrum = PowerLaw

[space]
        xref = 329.7
        yref = -30.2
        rad = 8.0
        srcpix = 120
        nxpix = 200
        nypix = 200
        nlong = 120.0
        nlat = 120.0
        binsz = 0.1
        coordsys = CEL
        proj = AIT
        phibins = 0.0

[time]
        tmin = 239557418.0
        tmax = 271093418.0

[energy]
        emin = 100.0
        emax = 300000.0
        enumbins_per_decade = 10

[analysis]

        likelihood = binned
        evclass = 2
        ComputeDiffrsp = yes
        zmax = 100.0
        roicut = no
        filter = DATA_QUAL==1&&LAT_CONFIG==1&&ABS(ROCK_ANGLE)<52
        irfs = P7SOURCE_V6
        convtype = -1

[fitting]
        optimizer = MINUIT
        ftol = 1e-06
```

Fig. 3: Exemple of a configuration file for `Enrico`.

`Enrico` has a module to produce paper-quality plots, saved in `eps`, `.C` (ROOT format) and `png`:

`> enrico_plot_sed PKS2155.conf`

`> enrico_plot_lc PKS2155.conf`

`> enrico_plot_tsmap PKS2155.conf`

It is also posible to compute the log-likelihood value of several spectral models assumed for the source of interest. Tested models are PowerLaw, LogParabola and PLExpCutoff. The tool is called by :

`> enrico_testmodel PKS2155.conf`

3.2 Configuration file

The configuration file is a `ascii` file automaticaly generated by the package but that can be edited by the user. An exemple is presented in Fig. **??**.

A configuration file is divided in sections. It contains simple sections (e.g. **energy**, **time**) in which the user defines cuts applied for all analysis. More specific sections exist and are used by only one tool (e.g. section **spectrum** for `enrico_sed`).

3.3 Check your results

While in principle straightforward, the ST might be tricky to use and the results of a *Fermi* data analysis should be checked. `Enrico` proposes few control plots in order to ensure the reliability of an analysis.

For the spectrum, a count plot (Fig. **??**) and a residual plot are produced. For the light-curves, a plot, $N_{\rm pred}/\sqrt{N_{\rm pred}}$ vs Flux/Δ Flux (Fig. **??**), is made to unsure the good computation of the errors since the two values must be correlated. Count map, model map and residual map (Fig. **??**) are also automaticaly produced and can be used to see if a source should be added to the sky model or have its paramters free.

4 Conclusions

`Enrico` is a powerful tool written in `Python` to run the *Fermi* ScienceTools. With the addition of debug plots and submission of several job on cluster of CPUs, rapid, robust and reproducible analyses can be performed easily. The implementation of automatic download of data from the FSSC data server for a given sky region, ener-

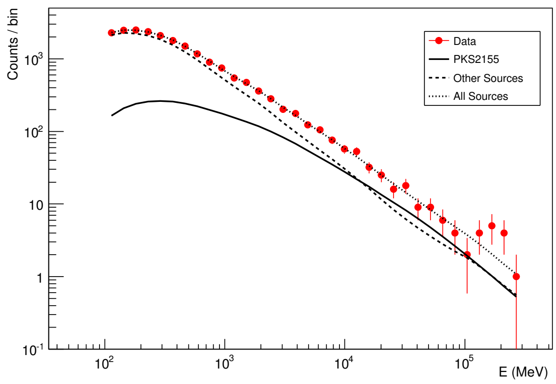

Fig. 4: `Counts plot` showing the number of count per bins and the different model components (Source, other objects and total) as fitted by `gtlike`.

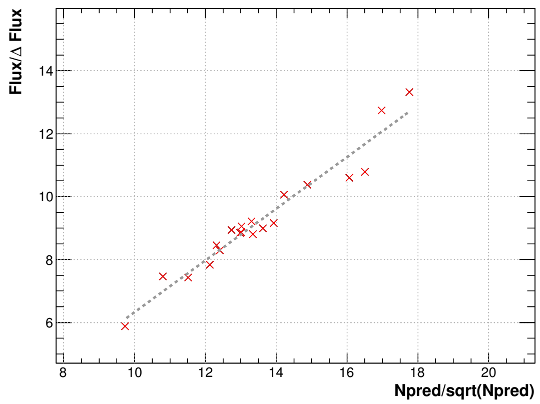

Fig. 5: Plot ($N_{\rm pred}/\sqrt{N_{\rm pred}}$ vs Flux/Δ Flux) used to decide whether the errors in the light-curves are well computed.

gy band and time range is being implemented as part of the `astroquery.fermi` module (see `https://github.com/astropy/astroquery`), which is a Python package to access public astronomy data services on the web.

We invite the reader to try `Enrico` for *Fermi* LAT data analysis. Start by downloading it from `https://github.com/gammapy/enrico` and browsing the documentation at `http://enrico.readthedocs.org`.

References

[1] Nolan, P. L., Abdo, A. A., Ackermann, M., et al. 2012, ApJS, 199, 31

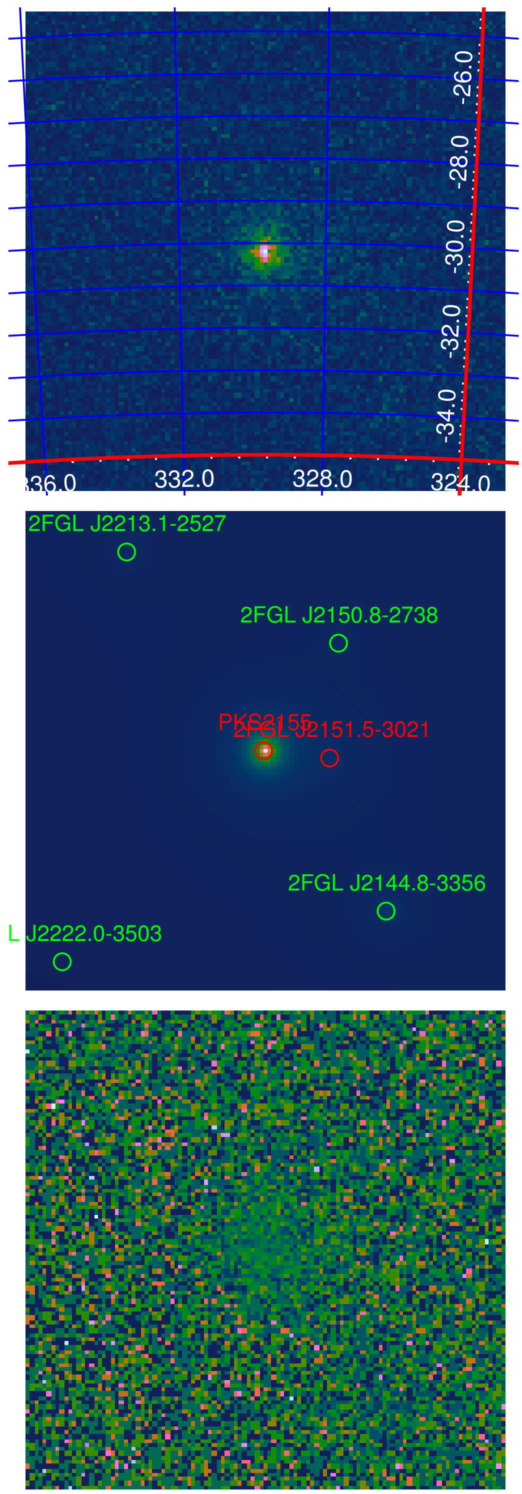

Fig. 6: From top to bottom: Counts map, Model map and residuals. The first two maps are produced with the ST (with `gtbin`and `gtmodelmap`) and the last is produced by `Enrico` to allow the user to check the sky model.