Enrico Documentation

Release 0.1

The Enrico Developers

July 17, 2015

Contents

Hi, I am Enrico, and I will help you run your Fermi data analysis!

- Code: https://github.com/gammapy/enrico
- Issues: https://github.com/gammapy/enrico/issues
- Documentation: http://enrico.readthedocs.org/
- Mailing List: http://groups.google.com/group/gammapy_enrico
- Fermi Science Tools: http://fermi.gsfc.nasa.gov/ssc/

CHAPTER 1

Features

- Get your results easy and fast by using the enrico command line tools.
- Results are reproducible, because config files and logs are used.
- The enrico command line tools are just frontends for functions and classes in the enrico python package, so if you know the Fermi tools and some python it is easy for you to modify things to your needs.
- A *Graphical User Interface (GUI)* has been developed to simply the use of the package.

Enrico is based on a configuration file which contains all the setup for your analysis. For each enrico tool, you just have to type

```
enrico_'tool' Configuration.conf
```

CHAPTER 2

Quick setup

You need to have the Fermi ScienceTools installed on you machine.

The provided scripts enrico-init.sh will set up enrico for you. Before you need to define few global variables.

```
export FERMI_DIR=< location of your Fermi software installation >
export FERMI_DATA_DIR=< location of your Fermi weekly and preprocessed data >
# (not mandatory)
source $FERMI_DIR/fermi-init.sh

export ENRICO_DIR=< location of your Enrico software checkout >
source $ENRICO_DIR/enrico-init.sh
```

Sourcing init_enrico.sh will setup your environment. You can check your installation by typing

```
enrico_setupcheck
```

For more informations, go to *Setup*

CHAPTER 3

A simple analysis

After you have done the *Setup*, running an analysis is as simple as typing these commands:

```
enrico_config Myconfig.conf <answer a few questions like the position of your target>
```

It will create an file named Myconfig.conf. Be sure that it contains all your setup. After you should generate an xml file for the sky model :

```
enrico_xml Myconfig.conf
```

Now run the likelihood analysis for the full spectrum and data points:

```
enrico_sed Myconfig.conf
```

If you like, a light curve or a TS map:

```
enrico_lc Myconfig.conf
enrico_tsmap Myconfig.conf
```

Note that if you have a computing cluster at your disposal, these compute-time-intensive tasks can be run in parallel.

Finally at the end you should plot your results using:

```
enrico_plot_sed Myconfig.conf (plot the SED)
enrico_plot_lc Myconfig.conf (plot the lightcurve)
enrico_plot_tsmap Myconfig.conf (generate a fits file for the TS map)
```

For more information, see *Tutorial*

Table of Contents

4.1 Setup

The most difficult tasks is how to set up your analysis environment correctly. Once that's done running the analysis will be simple, because Enrico will help you.

You need to:

- Install the Fermi ScienceTools
- Install Enrico
- Install additional (optional) python packages
- Download Fermi data (photon and spacecraft files)
- Download the diffuse model and 2FGL catalog files

Here we give some instructions on how to do these steps, but to be honest, if you don't know how to install python packages (e.g. know what PYTHONPATH, setup.py and pip are), you might fail and will have to ask someone at your institute for help.

4.1.1 Install the Fermi ScienceTools

Download and install the Fermi Science Tools as described here.

Then do this to set up your shell (we are assuming *bash* throughout) for Fermi analysis:

```
export FERMI_DIR = <...>
source $FERMI_DIR/bin/init_fermi.sh
```

Check that you have access to the Fermi command line tools and python packages:

```
gtirfs
python
>>> import UnbinnedAnalysis
```

If there is no error message, the ST are installed. ST come with all the needed python packages (numpy, scipy) but you might want to install them.

4.1.2 Install Enrico

Get the enrico package

```
git clone https://github.com/gammapy/enrico.git
```

Make sure your PATH contains the enrico command line tools (in the scripts directory) and PYTHONPATH contain the enrico python package:

```
export ENRICO_DIR=< location of your Enrico software checkout >
source $ENRICO_DIR/enrico-init.sh
```

Run the following command to check the status of your analysis environment:

```
enrico_setupcheck
```

Build the documentation if you like:

```
cd doc
make html
firefox build/html/index.html
```

4.1.3 Install additional (optional) python packages

Note: You don't have to install all of the following packages, but if you do you'll have a much nicer and more powerful python environment.

configobj is used throughout and you really need it, other packages are optional or come with the ST

You'll get an *ImportError* with the name of the missing package once you try to use part of the code that relies on that package.

First of all you should install distribute and pip as discribed here, because pip makes it easy to install additional packages:

```
curl -O http://python-distribute.org/distribute_setup.py
python distribute_setup.py
curl -O https://raw.github.com/pypa/pip/master/contrib/get-pip.py
python get-pip.py
which pip (should be located in the Fermi software)
pip (should print a help message)
```

Next install ipython, which is a much nicer interactive python shell than the default python shell and configobj, which is a more powerful config file reader and is user by Enrico instead of the ConfigParser from the python standard library. *nose <http://readthedocs.org/docs/nose/en/latest/>__ is a python test runner, used e.g. by 'numpy.test()*. Sphinx is the python documentation generator and we also use it for this project:

```
pip install ipython
pip install configobj
pip install nose
pip install sphinx
```

Now update to a recent Numpy and Scipy. The Fermi tools ship with a very old Numpy (version 1.4.1) and no Scipy (even though scipy is used e.g. in *IntegralUpperLimits.py*.

```
pip install numpy
pip install scipy
```

Note: Numpy and Scipy have many C and Fortran extensions and compiling those can fail. In that case you have to download the packages and build them yourself, adjusting some build options to your system.

```
git clone https://github.com/numpy/numpy/
cd numpy
python setup.py build <options for your system here>
```

Finally install some nice and useful python packages:

- Kapteyn is great for working with coordinates and plotting images,
- ATpy has a nicer API for working with tables than pyfits
- uncertainties makes error propagation dead simple.

```
pip install http://www.astro.rug.nl/software/kapteyn-beta/kapteyn-2.1.1b9.tar.gz
pip install atpy
pip install uncertainties
```

4.1.4 Download Fermi data (photon and spacecraft files)

There are two options. If you are only analysing one ore two targets, you can download the data for these targets specifically from the FSSC dat server.

If you are doing many analyses or survey work, you should download the complete data set, i.e. one global spacecraft file and weekly photon files from the FSSC FTP server.

Actually Enrico will help you working with the weekly files. Just set the following environment variable to wherever you'd like the spacecraft file and weekly photon files to be:

```
FERMI_DATA = <somewhere with ~20 GB storage space>
```

Then running the following command will download the data in an incremental manner

```
enrico_download --download_data
```

This will run wget to update only the weekly files that are necessary and download a spacecraft file for the whole mission (~ 500 MB). There is no documented method to combine weekly spacecraft files.

Obviously you should share one software and data installation per institute and not hit the FSSC servers without need.

4.1.5 Download the diffuse model and 2FGL catalog files

The diffuse model and 2FGL catalog files can be downloaded from the FSSC

Enrico uses the following environment variables to find the catalog and diffuse model files

```
FERMI_CATALOG_DIR
FERMI_DIFFUSE_DIR
FERMI_DOWNLOAD_DIR
FERMI_PREPROCESSED_DIR
```

They are set automatically but you can change the default value and run the following command to download any missing files from the FFSC

```
enrico_download  --download_aux
```

This will also download the Template files for the analysis of extended sources.

4.1.6 Issues

- Building from source doesn't work on the MPIK cluster or on my Mac.
- Importing pyIrfLoader might fail if pyLikelihood hasn't been imported first. So if you ever see that error, look at the traceback where it happens and replace

```
>>> import pyIrfLoader
```

with

```
>>> import pyLikelihood
>>> import pyIrfLoader
```

4.2 Tutorial

This tutorial will walk you through the steps to repeat the analysis of the AGN PG1553+113 done in the official Fermi collaboration python tutorial and results published here.

We strongly recommend that you run this analysis once, so that you can be sure to produce correct results before analyzing your source.

Make a new directory now where you want all of your intermediate and final analysis results to be placed and go there. For illustration purposes we will assume in this tutorial this directory is *~/myanalysis*:

```
mkdir ~/myanalysis`~/myanalysis`
cd ~/myanalysis
```

First get the data as explain the the offical page. You should have this when typing *ls*

```
L110728114426E0D2F37E85_PH00.fits
L110728114426E0D2F37E85_PH01.fits
L110728114426E0D2F37E85_SC00.fits
```

Make an ascii will with the 'entire' path of the FT1 (PH) fits files. Let's assume this file is called *data.list* .

The analysis can be performed on disjoint time bins. An ascii file must be provided in the [Time]/file option. The file must be a 2 columns file with T_start and T_stop (in MJD, MET or JD). Each line is a time bin.

```
239557418 239657418
239667418 239757410
239957418 239967418
```

4.2.1 Folder management

Once you have decided where the results will be placed (here *~/myanalysis*), enrico will create some subfolders to store results and files.

- fits files (livetime cube, selected events, etc...) will be placed in the main folder (here *~/myanalysis*)
- The spectrum results (ascii files, plots) will be stored in a folder named Spectrum (here *~/myanalysis/Spectrum*)
- Data (fits files) and results from the generation of energy bin will be stored in a subfolder *Ebin#* where # is the number of bins you ask for (here *~/myanalysis/Ebin#*).
- Data (fits files) and results generated for the light curve will be placed in *Lightcurve#* where # is the number of bins you ask for (here *~/myanalysis/Lightcurve#*).
- TSmap data will be placed in a TSmap subfolder (here *~/myanalysis/TSmap*).

4.2.2 Make a config file

enrico uses configuration file to run analysis. The way it has been thought, you need only one configuration file per analysis. The drawback is that such file can be complicated to generated by hand or to understand (for a description see *Configuration Files*).

You can use the *enrico_config* tool to quickly make a config file called *myanalysis.conf*. It will ask you for the required options and copy the rest from a default config file *enrico/data/config/default.conf*:

Statement enclosed in [] are default, being there to help you.

```
$ enrico_config myanalysis.conf
Please provide the following required options [default] :
Output directory [~/myanalysis] :
Target Name : PG155+113
Right Ascension: 238.92935
Declination: 11.190102
Options are : PowerLaw, PowerLaw2, LogParabola, PLExpCutoff
Spectral Model [PowerLaw] : PowerLaw2
ROI Size [15] : 10
FT2 file [~/myanalysis/spacecraft.fits] : ~/myanalysis/L110728114426E0D2F37E85_SC00.fits
FT1 list of files [~/myanalysis/events.lis] : ~/myanalysis/data.list
tag [LAT_Analysis] : TestOfPG1553
Start time [239557418] : 239557417
End time [334165418] : 256970880
Emin [100] : 200
Emax [300000] :
```

Note :

- Always give the full path for the files
- We used the PowerLaw2 model as in the Fermi tutorial.
- Time is give in MET
- Energy is given in MeV
- ROI size is given in degrees

Now you can edit this config file by hand to make further adjustments. For more informations about the configuration file see *Configuration Files*

Note: If you know exactly how the analysis steps work you can also make adjustments later on. But we have not put in a gadzillion of checks for each step to make sure that parameters are consistent with previous steps, so it is best to only adjust parameters at the beginning.

4.2.3 Make a model xml file

The ST works using an sky model written in xml format. Often, this model is complicated to generate. You can run enrico_xml to make such model of the sky and store it into a xml file which will be used for the analysis.

```
$ enrico_xml myanalysis.conf
use the default location of the catalog
use the default catalog
Use the catalog :  /CATALOG_PATH/gll_psc_v06.fit
Add  12  sources in the ROI of  10.0  degrees
3  sources have free parameters inside  3.0  degrees
write the Xml file in  ~/myanalysis/PG155+113_PowerLaw2_model.xml
```

Note: Note that you give options for this step simply by mentioning the config file. For the *enrico_xml* tool, the relevant options are in the [space], [target] section. The out file is given by [file]/xml.

4.2.4 Get data

There are two possibilities:

- Download data by hand for this target.
- Use the weekly generated fits file available on the web

4.2.5 Run global fit

The gtlike tool implemented in the ST find the best-fit parameters by minimizing a likelihood function. Before running gtlike, the user must generate some intermediary files by using different tools. With enrico, all those steps are merged in one tool. To run the global fit just call :

```
$ enrico_sed myanalysis.conf
```

This will make all the steps for you (gtselect, gtmktime,gtltcube, etc...), produce all the required fits files and fit the data (gtlike). A file with the extension 'results' will be produced and where all the results will be stored.

If you want to refit the data because e.g. you changed the xml model, you are not force to regenerate the fits file. Only the gtlike tool should be recall. This is also possible with enrico. By changing the option [spectrum]/FitsGeneration from yes to no, enrico will be told to not generate the fits files and directly proceed to the fit.

if the found TS is below the value set in [UpperLimit]/TSlimit, then an upper limit is computed.

Note: For the *enrico_sed* tool, most of the relevant options in the [spectrum] section

You can use *enrico_testmodel* to compute the log(likelihood) of the models *POWERLAW*, *LogParabola* and *PLExpCutoff*. An ascii file is then produced in the Spectrum folder with the value of the log(likelihood) for each model. You can then use the Wilk's theorem to decide which model best describe the data.

4.2.6 Make flux points

Often, an SED is presented with point obtained by restricting the energy range and re-run a complete analysis.

To make flux points, again *enrico_sed* tool will be used. It will first run a global fit (see previous section) and if the option [Ebin]/NumEnergyBins is greater than 0, then at the end of the overall fit, enrico will run NumEnergyBins analysis by dividing the energy range.

Each analysis is the a proper analysis (it runs gtselect, gtmktime,gtltcube,..., gtlike), run by the same enrico tool than the full energy range analysis. If the TS found in the time bins is below [Ebin]/TSEnergyBins then an upper limits is computed.

Note: If a bin failed for some reason or the results are not good, you can rerun the analysis of the bin by calling *enrico_sed* and the config file of the bin (named SOURCE_NumBin.conf and in the subfolder Ebin#).

Note: Most of the relevant options in the [Ebin] section.

4.2.7 Make a light curve

A light curve is obtained by run the entire analysis chain into time bins. To make a light curve

```
$ enrico_lc myanalysis.conf
```

It will divide the time range in [LightCurve]/NLCbin bins and run a proper analysis. If the TS found in the time bins is below [LightCurve]/TSLightCurve then an upper limits is computed.

Note: Note that you give options for this step simply by mentioning the config file. For the *enrico_lc* tool, most of the relevant options are in the [LightCurve] section

---

It is also possible to make a folded LC using

```
$ enrico_foldedlc myanalysis.conf
```

This is designed for binary or periodic system. The user provide the starting point a period and the period lentgh as well as number of LC bins

4.2.8 Make a TS map

TS map are use to find new source in a ROI. They are produced by adding a spurious source on each point of a grid (pixel) and computing the TS of this source.

You can make a TS map using the tool *enrico_tsmap*. It will compute the TS in each bin of the ROI. You must have run *enrico_sed* before.

Note: This binning is based on the count map produced during the fit of the full energy range *enrico_sed*. The division of the ROI controlled by the option [space]/npix and [space]/npiy but cannot be change after having run *enrico_sed*.

In order to speed up the process, parallel computation can be used. Either each pixel can be a job by itself (option [TSMap]/method = pixel) or a job can regroup an entire row of pixel (option [TSMap]/method = row)

If you want of remove the source your are interested in from the TS map (i.e. froze its parameters to the best fit values) use [TSMap]/RemoveTarget = yes.

Note: For the *enrico_tsmap* tool, most of the relevant options are in the [TSMap] section

---

It can happend that some job failed of the obtain TS is not good (due to convergence problems) If a pixel (or a row) has failed, you can rerun it.

To re-run a single pixel, i.e. the pixel (49,4) :

```
enrico_tsmap myanalysis.conf 49 4
```

To re-run a row, ie row number 49 :

```
enrico_tsmap myanalysis.conf 49
```

4.2.9 Upper Limits

An upper limits is calculated if the Test Statistic (TS) of the source is below a certain limit set by the user. To set this limit for :

- *enrico_sed*, use the option [UpperLimits]/TSlimit.

- bins in energy, use the option [Ebin]/TSEnergyBins.
- *enrico_lc*, use the option [LightCurve]/TSLightCurve.

Two methods are available :

- The profile method, which look for a decrease of a certain amount of the likelihood function
- The integral method which compute the integral of the likelihood function as a function of a parameter to set the UL

Both implementations are provided by the ScienceTools and used by enrico.

Note: For upper limits, most of the relevant options are in the [UpperLimits] section

4.2.10 Plot results

Now, we want to plot the results of the analysis we performed. Some plots can be produced by enrico. Using the tools *enrico_plot_** allow to plot the results of your analysis.

The 1 sigma contour plot can be computed by *enrico_sed* if the option [Spectrum]/ResultPlots=yes. Then to plot it, call *enrico_plot_sed myanalysis.conf* which will make a SED with the 1 sigma contour and add the data points computed previously (section *Make flux points*).

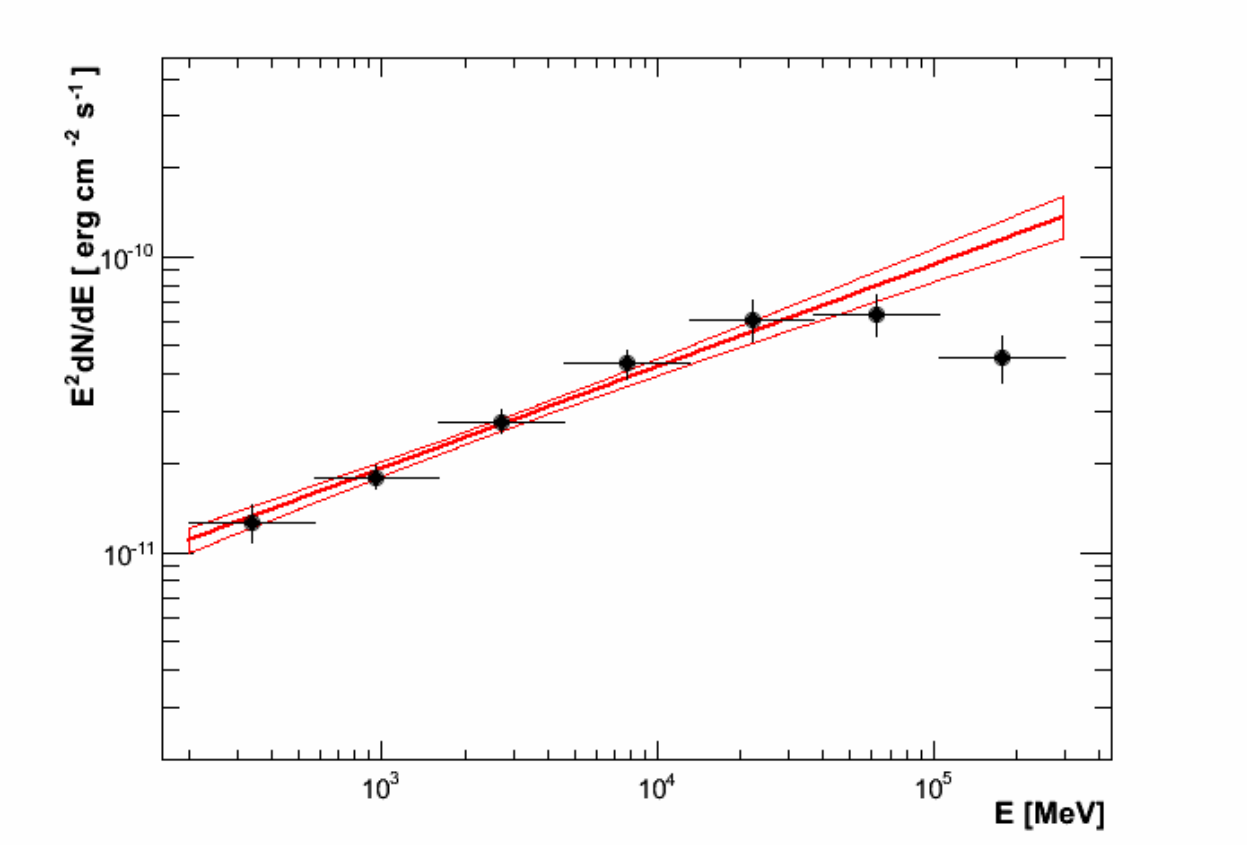


Fig. 4.1: SED of PG 1553+113

If you ran a binned analysis and with the option [Spectrum]/ResultPlots=yes then a model map is produced to compare with the real count map (see the section check results).

- The light curve can be plotted using *enrico_plot_lc myanalysis.conf* as well as diagnostic plot (TS vs time, Npred vs time, etc...)

```
Chi2 =  33.4499766302  NDF =  19
probability of being cst =  0.0213192240717

 Fvar =  0.17999761777  +/-  0.089820202452
```

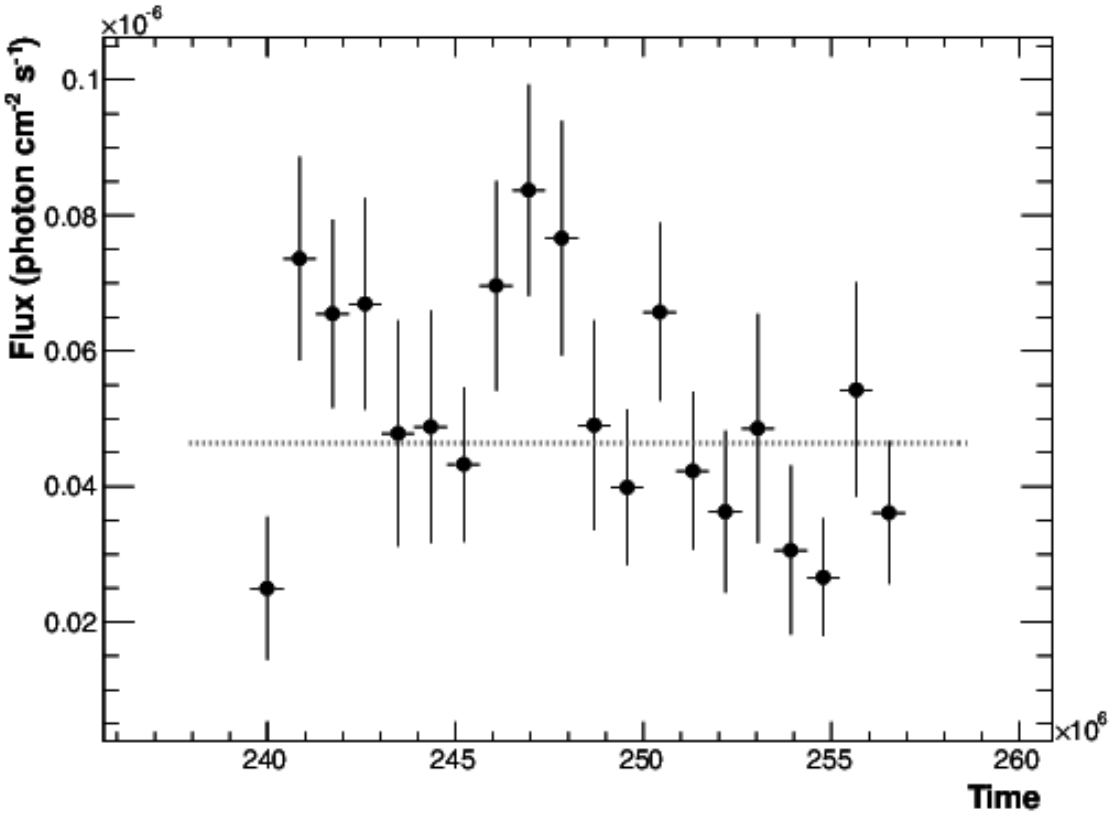


Fig. 4.2: Light curve of PG 1553+113. The dashed gray line is the results of a fit with a constant.

enrico also computes the variability index as described in the 2FGL catalog (see here).

- The TS map (see the section *Make a TS map*) can be plotted and save in a fits file using *enrico_plot_tsmap myanalysis.conf*. This will generate a fits file that can be plotted using the script *plotTSmap.py*

4.2.11 Check results

There is different way to check the quality of a results. First have a look the log file and loff for any error or warning messages. Enrico also produce maps that can use to check the results

Spectrum

- counts map, model map and subtract map.

These maps are use to visualize the ROI and check and see any misfitted sources. You can plot then using the script 'plotMaps.py'

- Counts Plot and Residuals. The points (# counts/bin) are the data, and the solide line is the source model. Dashed line is the sum of all other model and dotted line is the sum of both. Error bars on the points represent

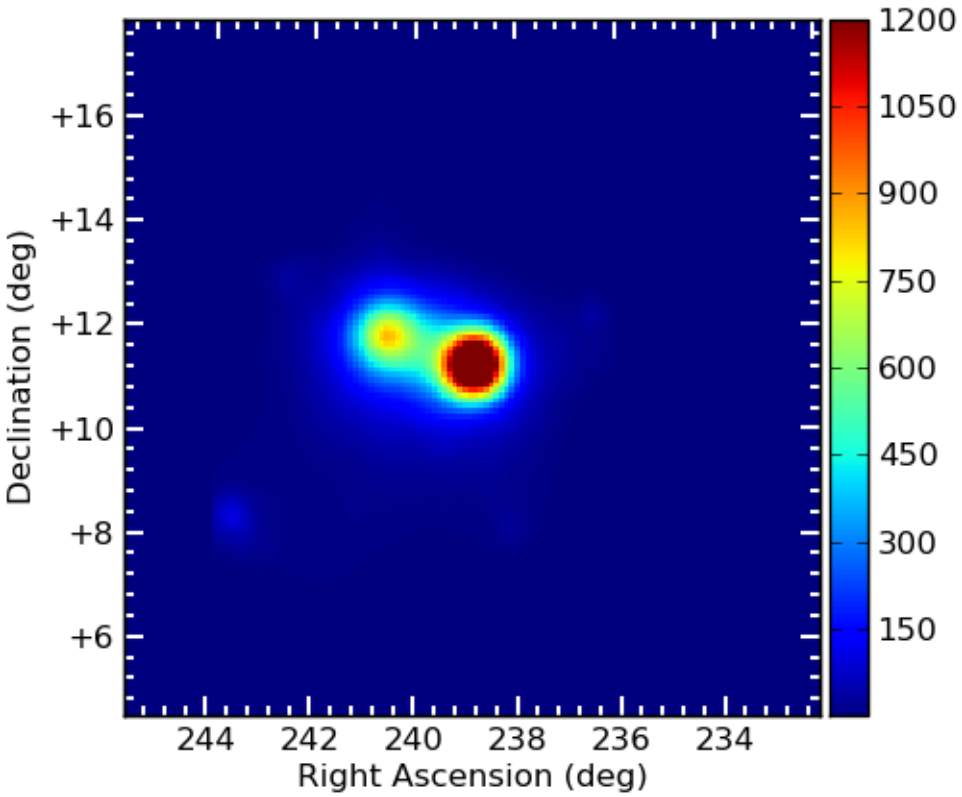
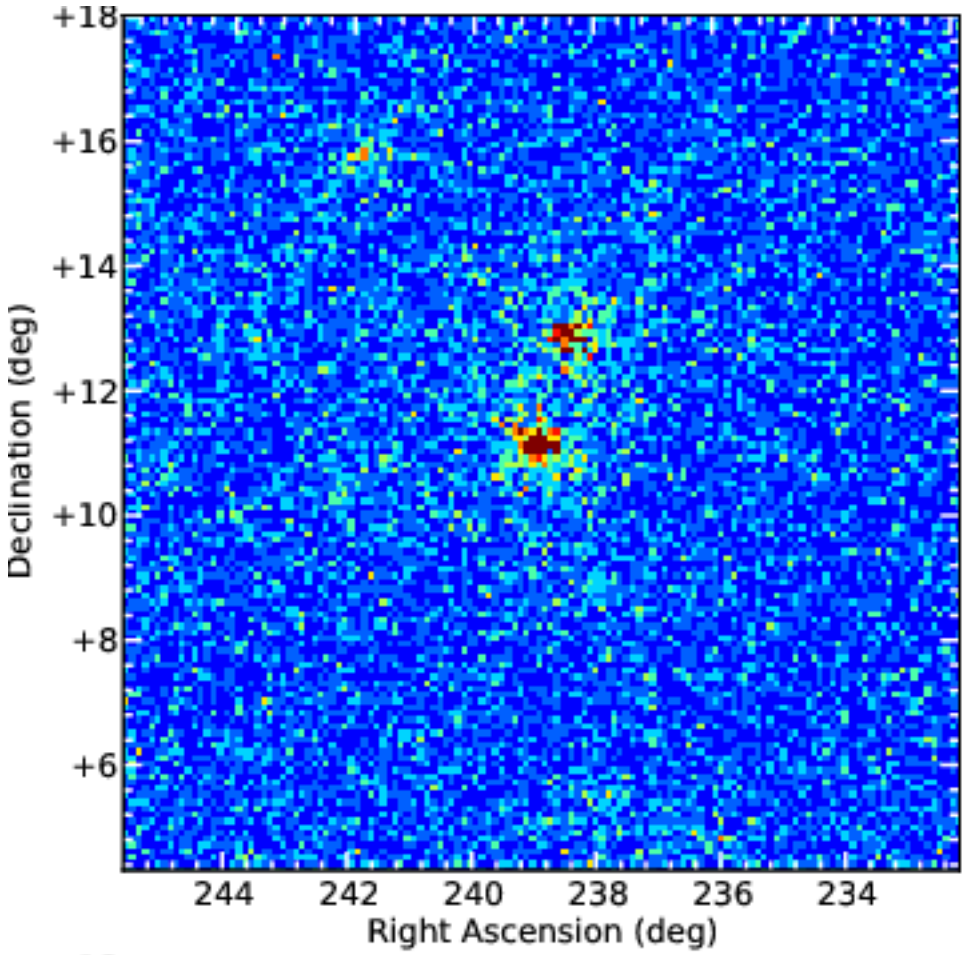
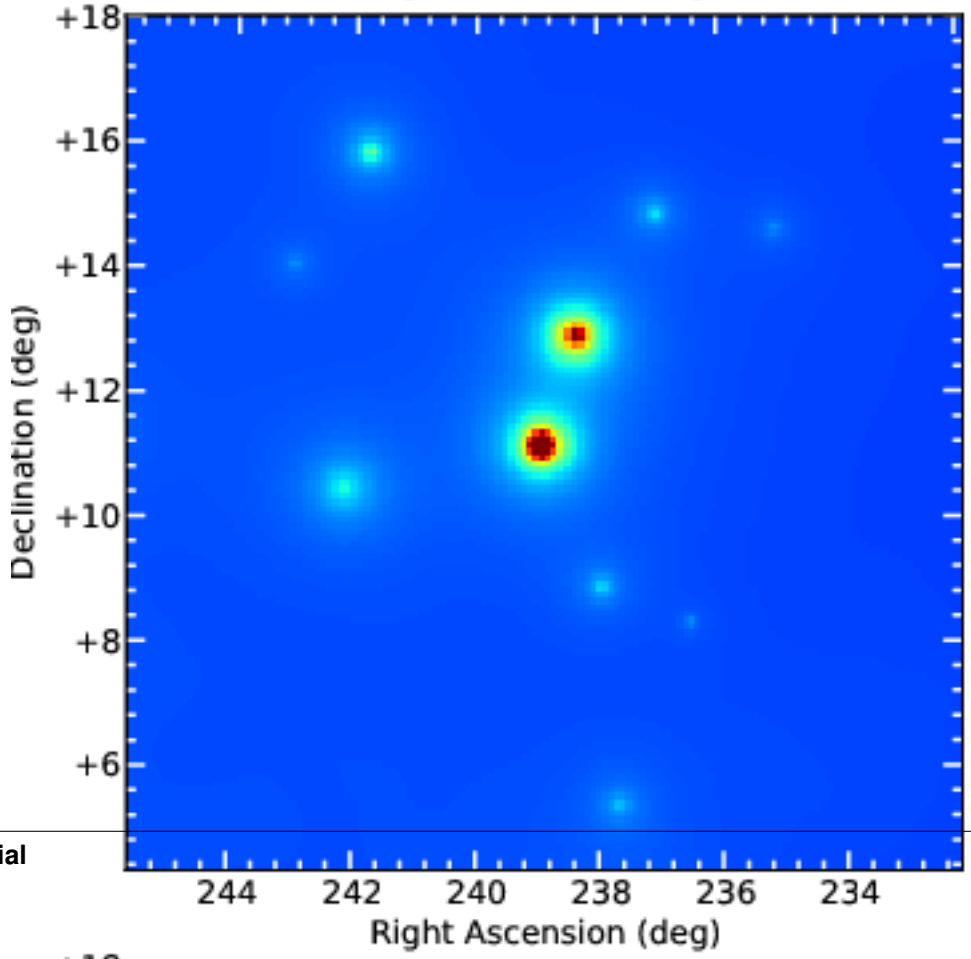


Fig. 4.3: TS Map of PG 1553+113.


+18
+16
+14
+12
+10
+8
+6
Declination (deg)
244
242
240
238
236
234
Right Ascension (deg)


+18
+16
+14
+12
+10
+8
+6
Declination (deg)
244
242
240
238
236
234
Right Ascension (deg)

sqrt(Nobs) in that band, where Nobs is the observed number of counts. The Residuals are computed between the sum model and the data.

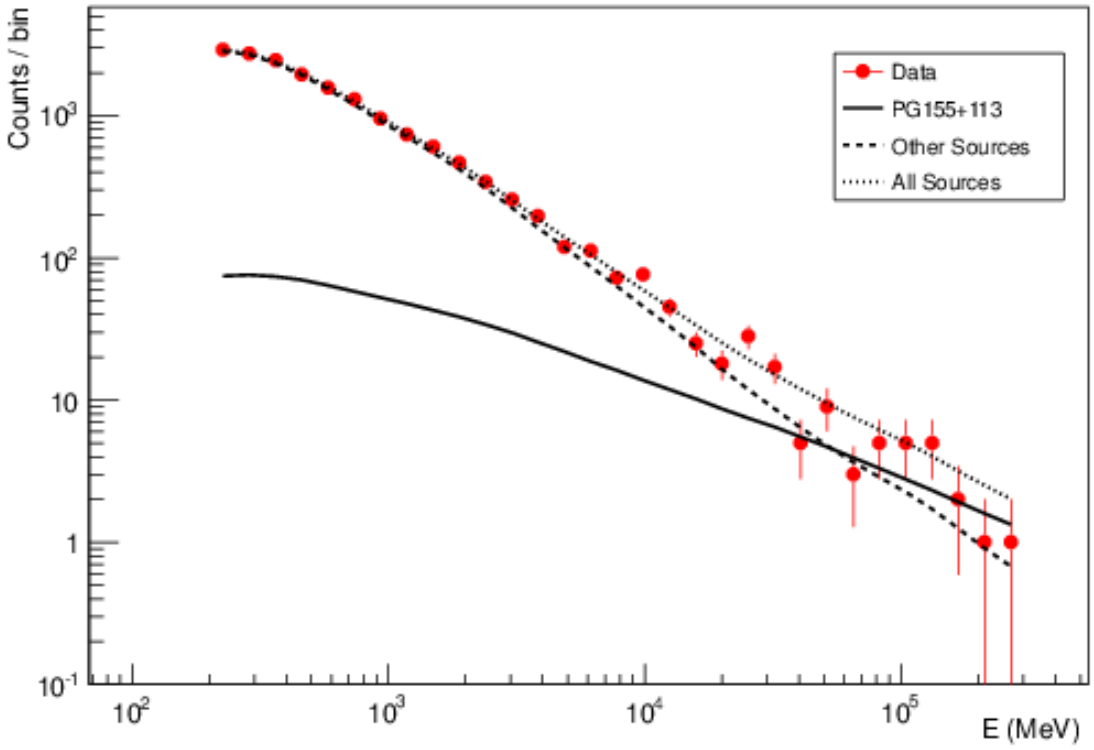


Fig. 4.5: Count plot of PG 1553+113

Light-curves

- The generation of light-curves might suffer from troubles, especially in the error bars computation. To check this, enrico plots the flux/dflux vs Npred/DNpred. If the errors are well computed the two variables are highly correlated.

4.2.12 Make Profile likelihood

The tool *enrico_scan* allows to make profile likelihood of the free parameters of the target. Each plot is save under the *scan* folder. Fits files must have been generated before.

4.2.13 Check different models

enrico_testmodel will run the gtlike analysis with different model and compute the loglikehihood value for each model. The user can then decide, base on a LRT which model best describe the data. Fits files must have been generated before.

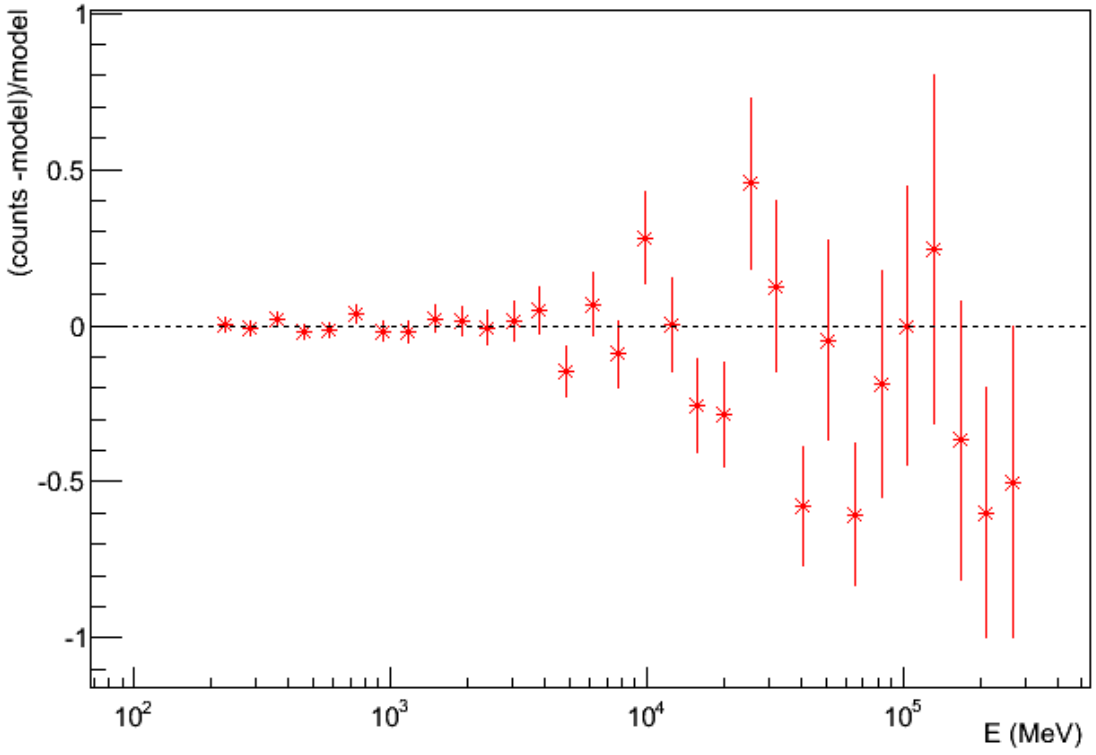


Fig. 4.6: Residuals plot of PG 1553+113

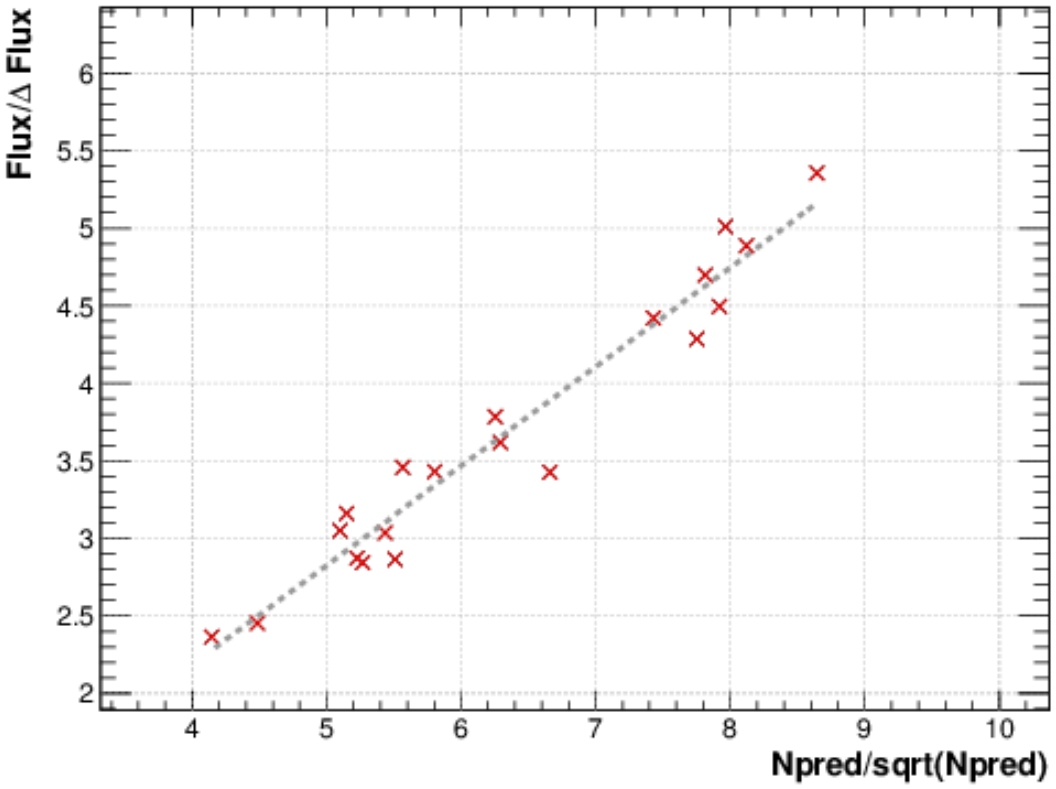


Fig. 4.7: flux/dflux vs Npred/DNpred plot of PG 1553+113. Red points are time-bins with TS> TSlimit, black (if any) are point for which an upper-limits on the flux was calculated and the points can be safely ignored in this plot. The gray dashed line is the fit with a linear function. to guide the eyes.

4.3 Configuration Files

4.3.1 Generalities

The config file is there to defined the parameters of your analysis. A config file is divided into sections. Each section is a logical group of option and there is some constants across the sections like names, functionalities.

This page will go through all the section, one by one. As for the *Tutorial*, we will asume that the working directory is *~/myanalysis*.

The *out* option gives the main folder where enrico will put the produced files and the results. The *verbose* option (default is 'yes') allow enrico to print information on the main output like TS, fluxes, predicted numbers of gammas, etc... Such values need computation time which can be saved by turning the option to 'no'. The *Submit* option, if turn to 'yes', will send the jobs to a cluster. *clobber* is a wrapper for the ST tool option and allow to overwrite existing output files is 'yes' (default)

```
out = ~/myanalysis
verbose = yes
Submit = no
clobber = no
```

4.3.2 Target

The target options gives the name, position and spectral model you want to use for the source you are interested in.

Note :

- Coordinate are in degrees.
- Available models are 'PowerLaw', 'PowerLaw2', 'LogParabola', 'PLExpCutoff'

The parameter *spectrum* is used for the generation of the sky model. Of course, you can change the model describing the spectrum of any sources (by hand...). All the models supported by the ST are described here. For now, the supported models by *enrico* are PowerLaw, PowerLaw2, LogParabola, PLExpCutoff, Generic. *Generic* is for the non-supported models. By supported, we mean that enrico can produce a xml file with such model and has some optional features. It is completely possible to run an analyze with enrico and a non-supported model.

```
[target]
   name = PG155+113
   ra = 238.92935
   dec = 11.190102
   spectrum = PowerLaw2
```

4.3.3 Space

The space option defined the properties of the region of interest (ROI). The names are the same as for the ScienceTools. The center of the ROI is xref, yref and the size is rad (in degrees). By default xref and yred are the target coordinates but this might be changed.

```
[space]
   xref = 238.92935
   yref = 11.190102
   rad = 10.0
   binsz = 0.1
   coordsys = CEL
```

```
   proj = AIT
   phibins = 0
```

Note: if you used binned analysis or for the generation of a TS map, the ROI will be divided in nxpix and nypix pixel with a bin size of binsz.

4.3.4 File

Here you defined where the FT2 and FT1 files are (this can be an ascii list). The xml option gives the location of the sky model. The *tag* is added to all files produced by enrico. When generating a config file, the value of spacecraft and event are set in such way that is point towards the the file downloaded with *enrico_download*

```
[file]
   spacecraft = ~/myanalysis/FT2.fits
   event = ~/myanalysis/data.list
   xml = ~/myanalysis/XML_model.xml
   tag = MyTag
```

4.3.5 Time

Start and stop time of your analysis in MET. The file option allow the analysis (Lc or SED) to be performed in disjoint time bins. This can be useful for e.g. MWL campaigns or non-constant time bins LC. The file must be an ascii file with 2 columns (start and stop) and each line is a time bin

```
[time]
   tmin = 239557417.0
   tmax = 256970880.0
   file = ""
   type = 'MET'
```

4.3.6 Energy

Minimal and maximal energy of your analysis in MeV. *enumbins_per_decade* is the number of bins per decade for the BINNED analysis chain.

```
[energy]
   emin = 200.0
   emax = 300000.0
   enumbins_per_decade = 10
```

4.3.7 Environ

Here are defined some directories. They are also defined as environment variables which can be over-writted using the configuration file.

```
[environ]
   # Analysis environment configuration
   # Can also be done via shell environment variables
   FERMI_DATA_DIR = ""
   FERMI_CATALOG_DIR = ""
   FERMI_CATALOG = ""
```

```
   FERMI_DIFFUSE_DIR = ""
   FERMI_PREPROCESSED_DIR = ""
```

4.3.8 Analysis

This part is used to defined how enrico should select the event. You can defined the event class (evclass : 1, 2 , etc..), the zenith angle cut (zmax) and the filter for gtmktime (filter). Also the IRFS used to describe the instrument are defined here (irfs).

Convtype is use to select either the front (0), back (1) or both (-1) events. If convtype =0 or 1, an ::FRONT of ::BACK is happened at the end of the irfs string automatically allowing to use the good IRFS.

```
[analysis]
   # General analysis options
   likelihood = binned
   evclass = 2
   zmax = 100.0
   roicut = no
   filter = DATA_QUAL==1&&LAT_CONFIG==1&&ABS(ROCK_ANGLE)<52
   irfs = P7REP_SOURCE_V15
   # if convtype =0 or 1, an ::FRONT of ::BACK is happend at the end of the irfs string automatically
   convtype = -1
```

4.3.9 fitting

Option for the minimizer. You can use MINUIT, NEWMINUIT, DRMGB, etc. ftol is the tolerance that the minimizer should reach.

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

4.3.10 model

This section is about the sky model generation. If you have set correctly you environment variables, then enrico is able to find the galactic and extragalactic model. If you want to use other model, you can specify here, their names and locations.

The 2FGL is used to find the source in the ROI. All the source with a significance greater than *min_significance* will be added. All sources within *max_radius* (in degrees) have their parameters free to vary in the fitting procedure. The other sources have their parameters frozen to the 2FGL value.

```
[model]
   # The following options determine the xml model
   diffuse_gal_dir = ""
   diffuse_iso_dir = ""
   diffuse_gal = gal_2yearp7v6_v0.fits
   diffuse_iso = iso_p7v6source.txt

   # user points sources for diffuse catalog sources
   point_only = True
   # freeze spectral parameters for weak and far away sources:
   min_significance = 4.0
   max_radius = 3.0
```

4.3.11 Spectrum

Options for *enrico_sed* which run all the ST tool to make an pointlike analysis.

- FitsGeneration, if yes, enrico will make all the steps before running gtlike and generated all the fits files needed. If the files have already been generated, change FitsGeneration to no and enrico will only run gtlike
- ResultPlots : Compute the SED (butterfly) and the model map (in the case of an binned analysis)
- FrozenSpectralIndex : froze the spectral index of the source (works for POWERLAW and POWERLAW2 models)
- SummedLike : you can use the summed likelihood method, then front and back event are treated separately and the likelihood which is minimized is the the sum of the front likelihood and back likelihood. This feature is provided by the ScienceTools.
- Submit : submit the job to a cluster or run it in the current shell.

```
[Spectrum]
   #Generates fits files or not?
   FitsGeneration = no
   #Generates plots (SED, model map)
   ResultPlots = yes
   #Freeze the spectral index of the source
   FrozenSpectralIndex = 0.0
   #Use the summed likelihood method
   SummedLike = no
```

4.3.12 UpperLimit

This section allows to set up the upper limit computation. During the computation, the spectral index of the source (it is assumed that a POWERLAW or POWERLAW2 model is used) is frozen to *SpectralIndex*. Two methods can be used, Profile of Integral, see the Fermi web site for more informations.

An upper limit, at the confidence level *cl*, is computed if the TS is below TSlimit. This hold only for *enrico_sed*

```
[UpperLimit]
   #Assumed Spectral index
   SpectralIndex = 1.5
   # UL method could be Profile or Integral (provided by the fermi collaboration)
   Method = Profile
   envelope = no
   #Compute an UL if the TS of the sources is <TSlimit
   TSlimit = 25.0
   # Confidence level for the Ul computation
   cl = 0.95
```

4.3.13 LightCurve

Option for enrico_lc which run an entire analysis in time bins and produce all the fits files needed to use gtlike.

- FitsGeneration, if yes, enrico will make all the steps before running gtlike and generated all the fits files needed. If the files have already been generated, change FitsGeneration to no and enrico will only run gtlike
- NLCbin : number of time bins
- MakeConfFile : enrico_lc will produce config file readable by enrico for each time bin. You can ask the tool to not do so, if you want to use/modify the config files.

- Submit : submit the job to a cluster or run it in the current shell.
- TSLightCurve : an upper limit is computed is the TS in a time bin is below this value.
- DiagnosticPlots : ask enrico_plot_lc to generate diagnostic plot (TS vs time, Npred vs flux ...)

```
[LightCurve]
   #Generates fits files or not?
   FitsGeneration = yes
   #Number of points for the LC
   NLCbin = 20
   MakeConfFile = no
   #Compute an UL if the TS of the sources is <TSLightCurve
   TSLightCurve = 9.0
   #Generates control plots
   DiagnosticPlots = yes
```

4.3.14 Folded LightCurve

This section is devoted to the folded LC. This is designed for binary system analysis.

- NLCbin : number of time bins
- epoch: Epoch of phase=0 in MJD, equal to tmin is 0
- Period: Orbital period in days

```
[FoldedLC]
   #Number of bins for the orbitally folded LC
   NLCbin = 10
   #Epoch of phase=0 in MJD, equal to tmin is 0
   epoch = 0
   #Orbital period in days
   Period = 10
```

4.3.15 Ebin

- FitsGeneration, if yes, enrico will make all the steps before running gtlike and generated all the fits files needed. If the files have already been generated, change FitsGeneration to no and enrico will only run gtlike
- NumEnergyBins : number of bins in energy
- TSEnergyBins : an upper limit is computed is the TS in an energy bin is below this value.
- Submit : submit the job to a cluster or run it in the current shell.

```
[Ebin]
   #Generates fits files or not?
   FitsGeneration = yes
   NumEnergyBins = 7
   #Compute an UL if the TS of the sources is <TSEnergyBins
   TSEnergyBins = 9
```

Option for enrico_tsmap

4.3.16 TSMap

This section is used to configured *enrico_tsmap* and *enrico_plot_tsmap*

- Re-Fit : use rerun gtlike in order to have the best fit parameters in your model.
- npix : number of pixels of you map. Remember that the TS map grid is based on the other maps (like count map) produced before and centred to the coordinates xref,yref.
- RemoveTarget : remove your source of interest form the map by freezing its parameters.
- Submit : submit the job to a cluster or run it in the current shell.

In order to speed up the process, parallel computation can be used. Either each pixel can be a job by itself (option [TSMap]/method = pixel) or a job can regroup an entire row of pixel (option [TSMap]/method = row)

```
[TSMap]
   #Re-fit before computing the TS map
   Re-Fit = no
   #Numbers of pixel in x and y
   npix = 10
   #Remove or not the target from the model
   RemoveTarget = yes
   #Generate the TS map pixel by pixel or by grouping the pixels by row.
   #(reduce the numbers of jobs but each job are longer)
   method = row
```

If a pixel (or a row) has failed you can rerun it. For the pixel 49,4 :

```
enrico_tsmap myanalysis.conf 49 4
```

For the entire row 49 :

```
enrico_tsmap myanalysis.conf 49
```

4.3.17 Finding the position of a source

This section is used to configured *enrico_findsrc*. It run the tool gtfindsource and update the file Roi_model.reg with the fitted position in red.

- FitsGeneration, if yes, enrico will make all the steps before running gtfindsource and generated all the fits files needed. If the files have already been generated, change FitsGeneration to no and enrico will only run gtfindsource
- Refit : re-run the optimizer before (use the option reopt)

```
[findsrc]
   #Generates fits files or not?
   FitsGeneration = option('yes', 'no', default='yes')
   #Reoptimize before
   Refit = option('yes', 'no', default='yes')
```

4.4 Graphical User Interface (GUI)

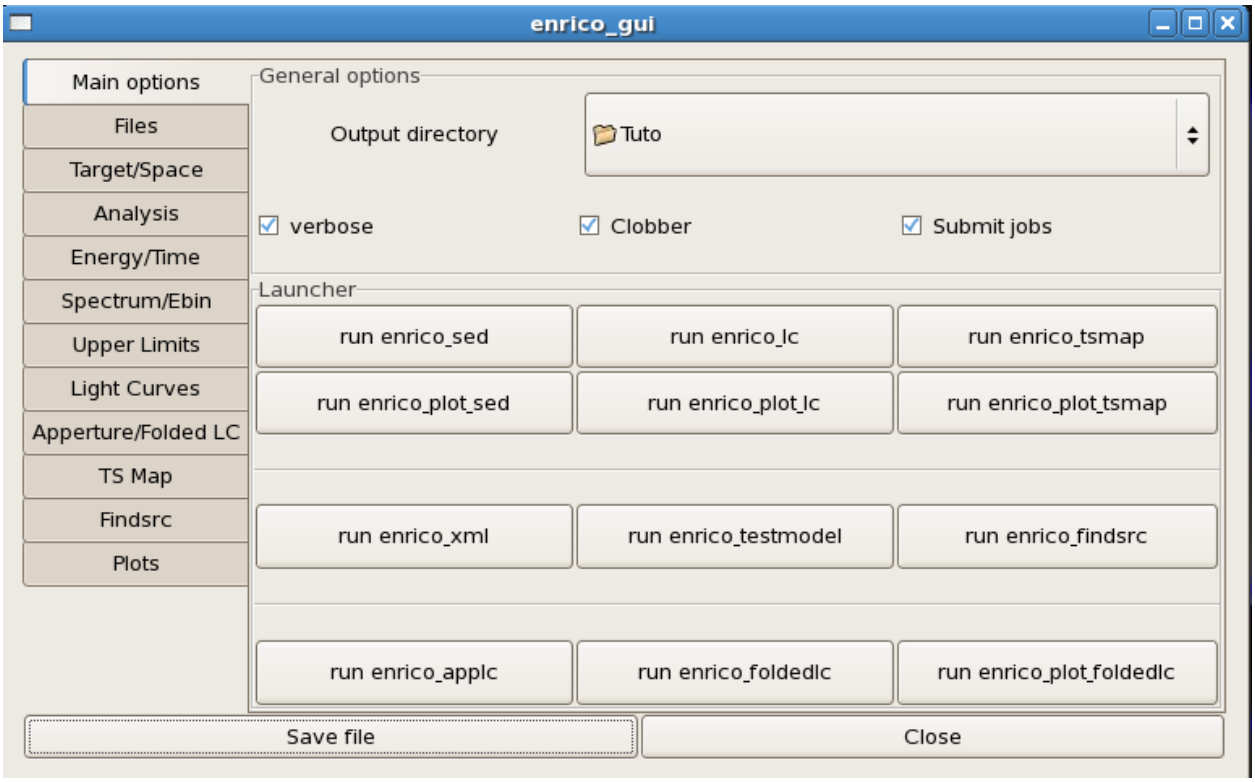
This page describe the windows of the *enrico* GUI. To run the GUI, type

```
enrico_'gui' Configuration.conf
```

The GUI aims to allow easy configuration file management and to run tools. Each page, arranged in tabs, roughly correspond to a section of the configuration file (see *Configuration Files*). At the bottom of the GUI are buttons to save the configuration file and close the GUI. The convention is such that a crossed box stand for a *yes* in the configuration file.

4.4.1 Main page

The first page is the Main page of the GUI. Here the main options can be defined and tools can be run using the buttons (run a tool save the configuration file on disk).



4.4.2 Files page

The second page manage the files definition (event, FT2 and xml) as well as the tag of your analysis.

4.4.3 Target/Space page

The target (name, position and spectral model) is defined is this page (first frame). The second frame defined the ROI (center (Xref, Yref), size). The *sync* button update the value Xref and Yref with the target position. It is then possible to have a ROI not centered on the target. The projection type and coordinate system is defined here also.

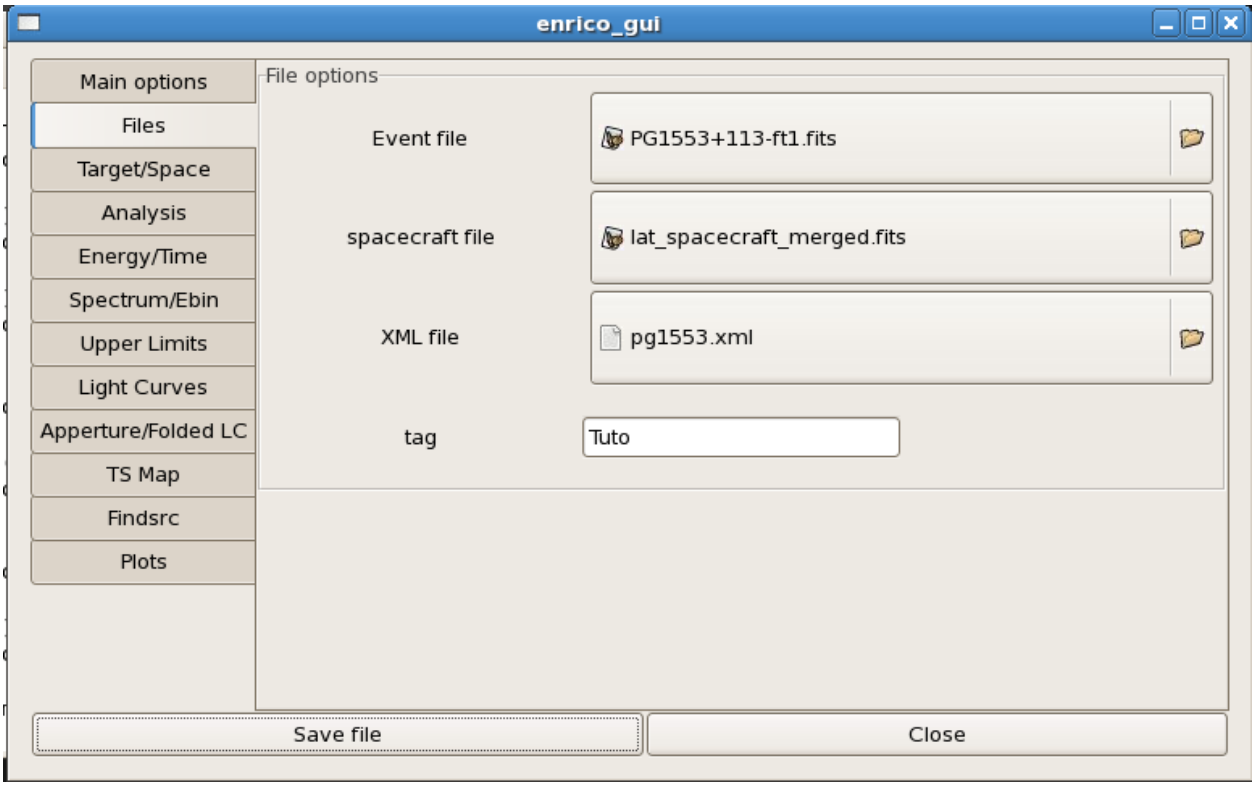
4.4.4 Analysis page

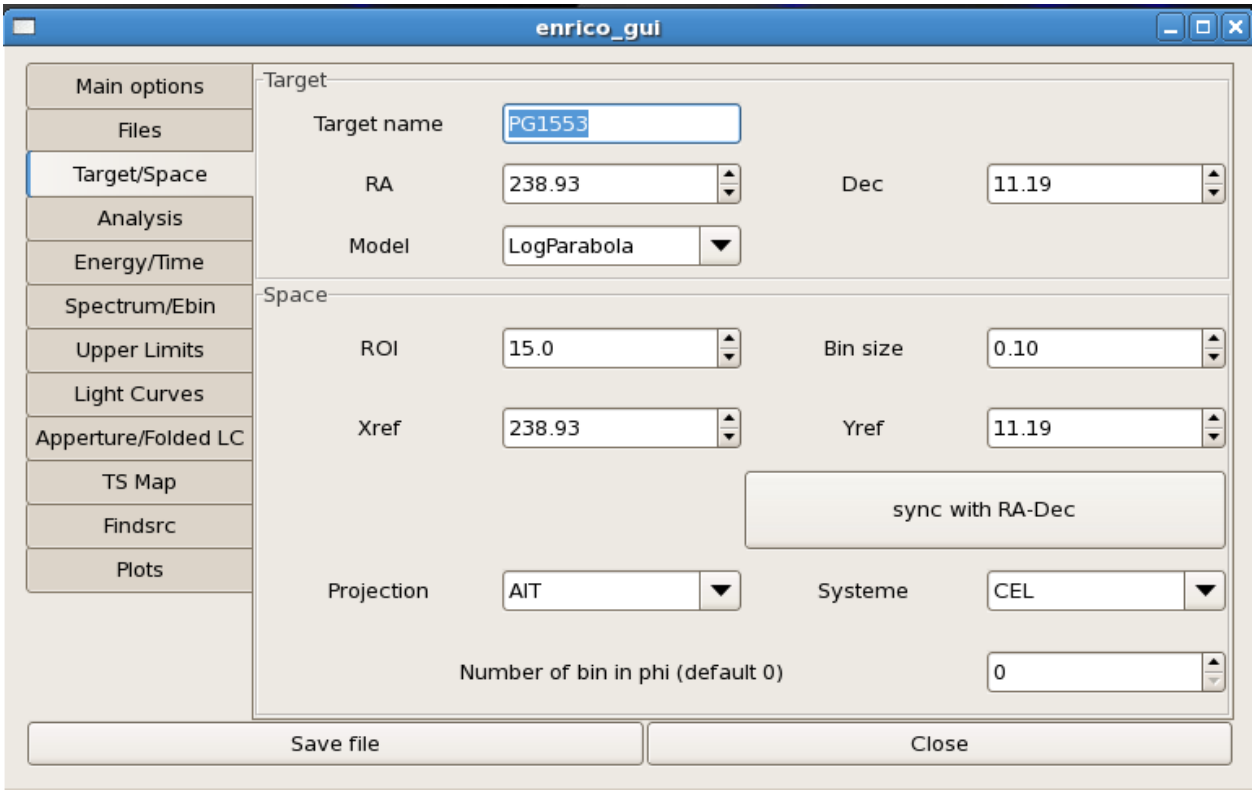
The analysis tab deal with the analysis chain (binned or unbinned), some cuts (zenith angle (zmax), filter for the GTI definition). The IRFs are also defined here.

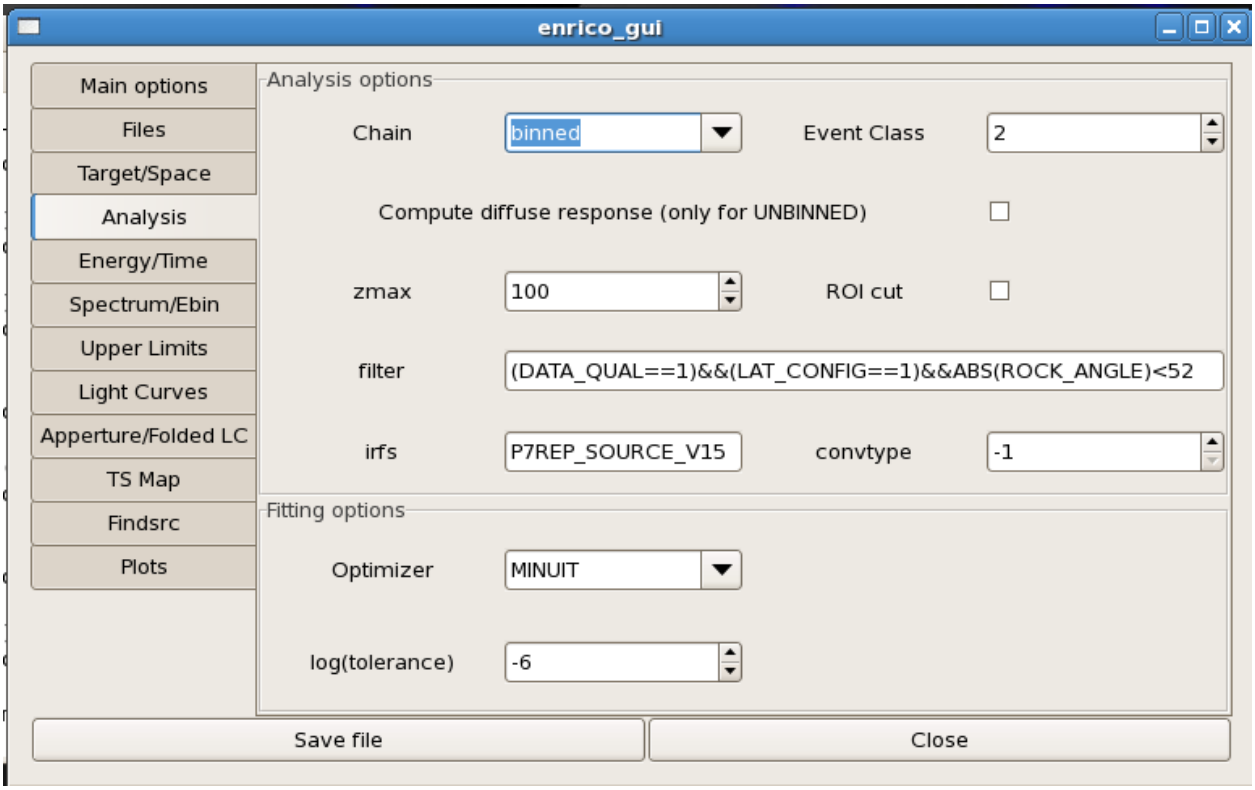
The fitting algorithm (MINUIT, NEWMINUIT, etc) and the tolerance are setup here.

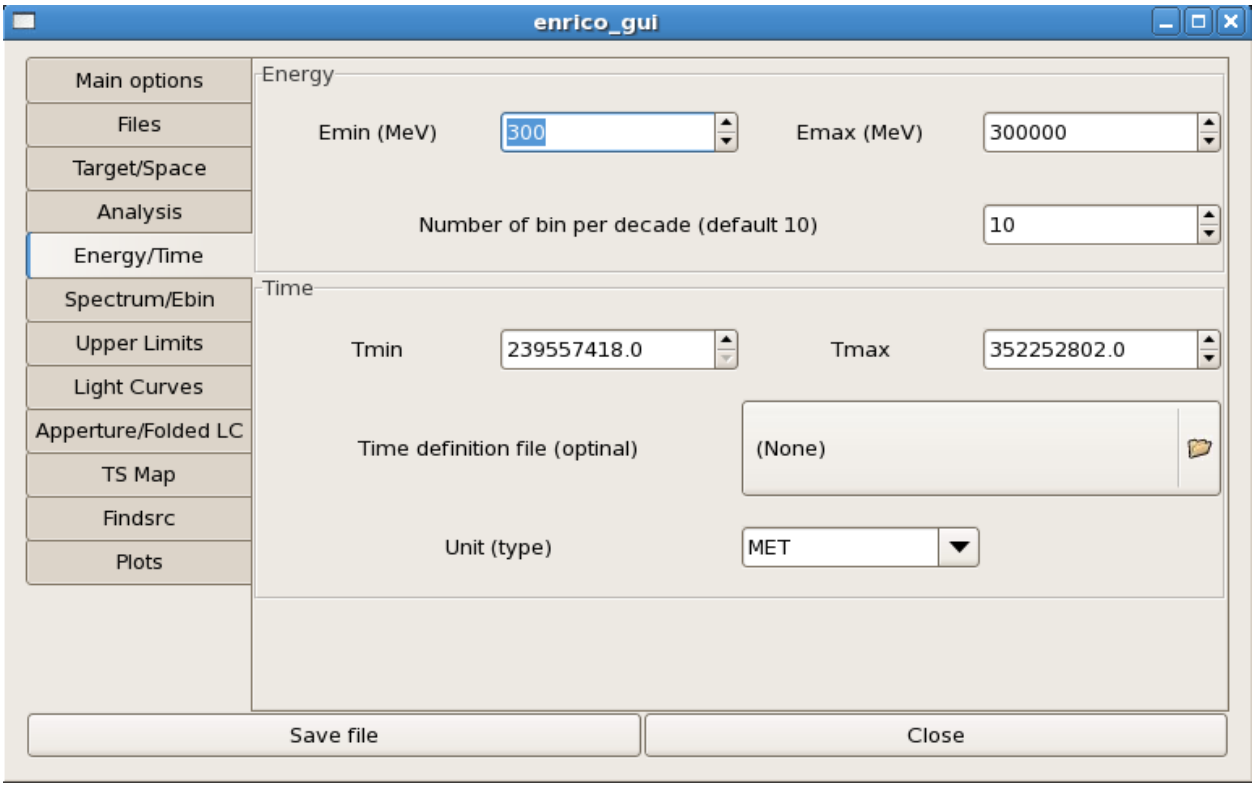
4.4.5 Energy/Time page

This page define the energy and time ranges.


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
File options
Event file
PG1553+113-ft1.fits
spacecraft file
lat_spacecraft_merged.fits
XML file
pg1553.xml
tag
Tuto
Save file
Close


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
Target
Target name
PG1553
RA
238.93
Dec
11.19
Model
LogParabola
Space
ROI
15.0
Bin size
0.10
Xref
238.93
Yref
11.19
sync with RA-Dec
Projection
AIT
Systeme
CEL
Number of bin in phi (default 0)
0
Save file
Close


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
Analysis options
Chain
binned
Event Class
2
Compute diffuse response (only for UNBINNED)
zmax
100
ROI cut
filter
(DATA_QUAL==1)&&(LAT_CONFIG==1)&&ABS(ROCK_ANGLE)<52
irfs
P7REP_SOURCE_V15
convtype
-1
Fitting options
Optimizer
MINUIT
log(tolerance)
-6
Save file
Close


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
Energy
Emin (MeV)
300
Emax (MeV)
300000
Number of bin per decade (default 10)
10
Time
Tmin
239557418.0
Tmax
352252802.0
Time definition file (optinal)
(None)
Unit (type)
MET
Save file
Close

4.4.6 Spectrum/Ebin page

This page is used to manage the spectrum and energy bins generation as in the configuration file. The buttons *Re-run Ebin* can be used to only rerun the bin (by running as many jobs as the number of bin)

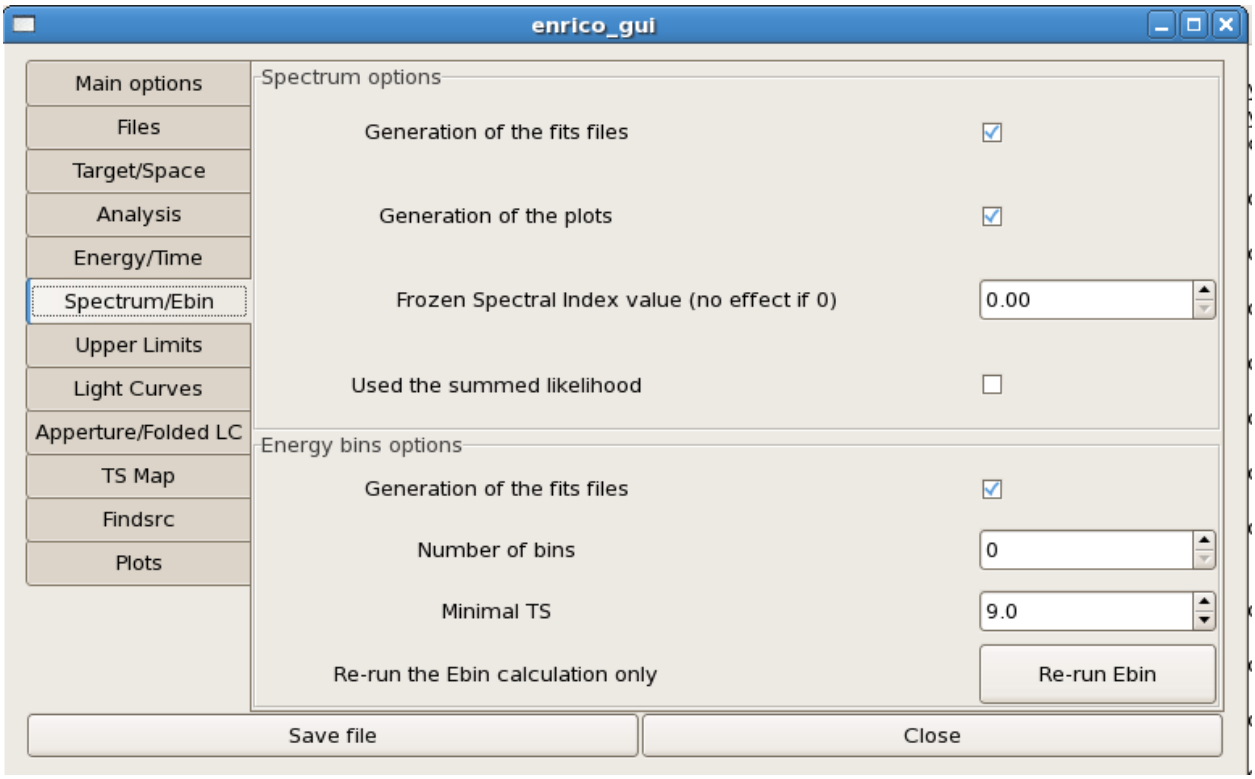


Fig. 4.8: Spectrum page of the GUI

4.4.7 Upper Limits page

This page allows the definition of the UL computation parameters : assumed index, minimal TS and confidence level

4.4.8 Light Curves page

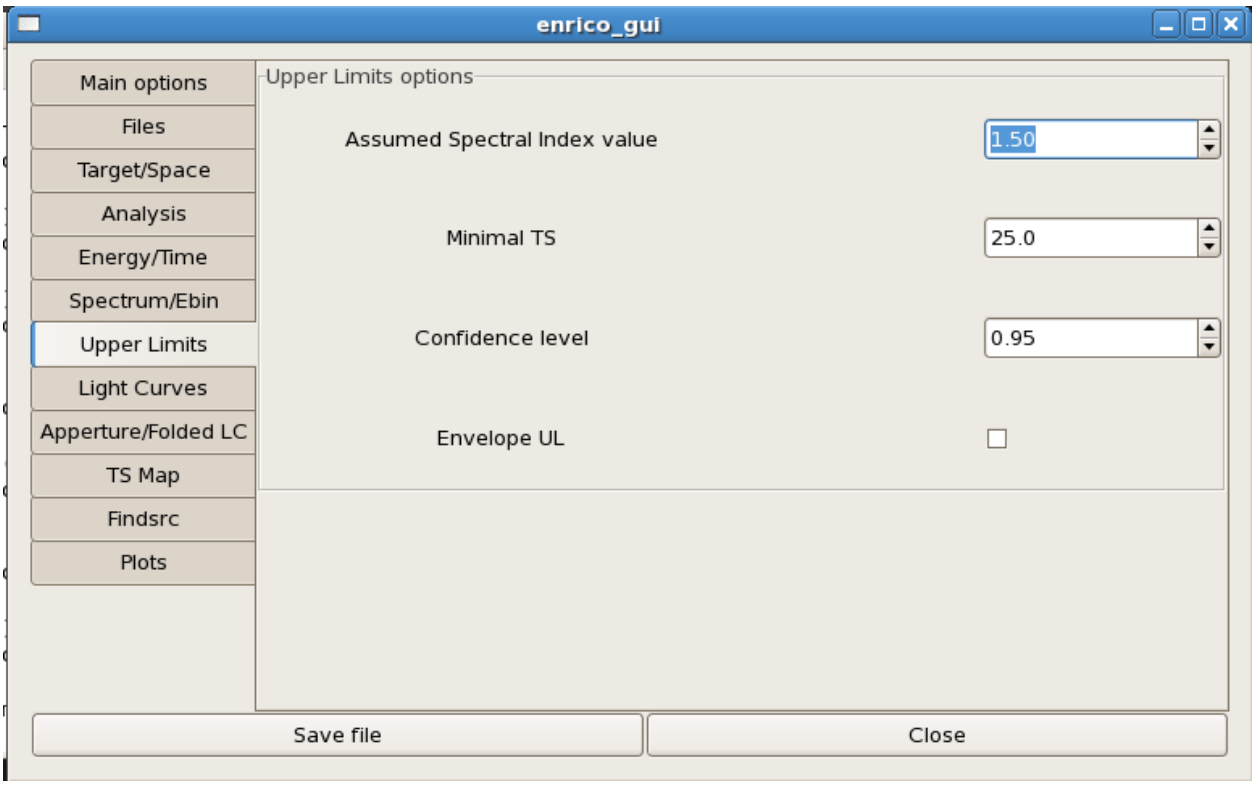
The light Curves are setup here.

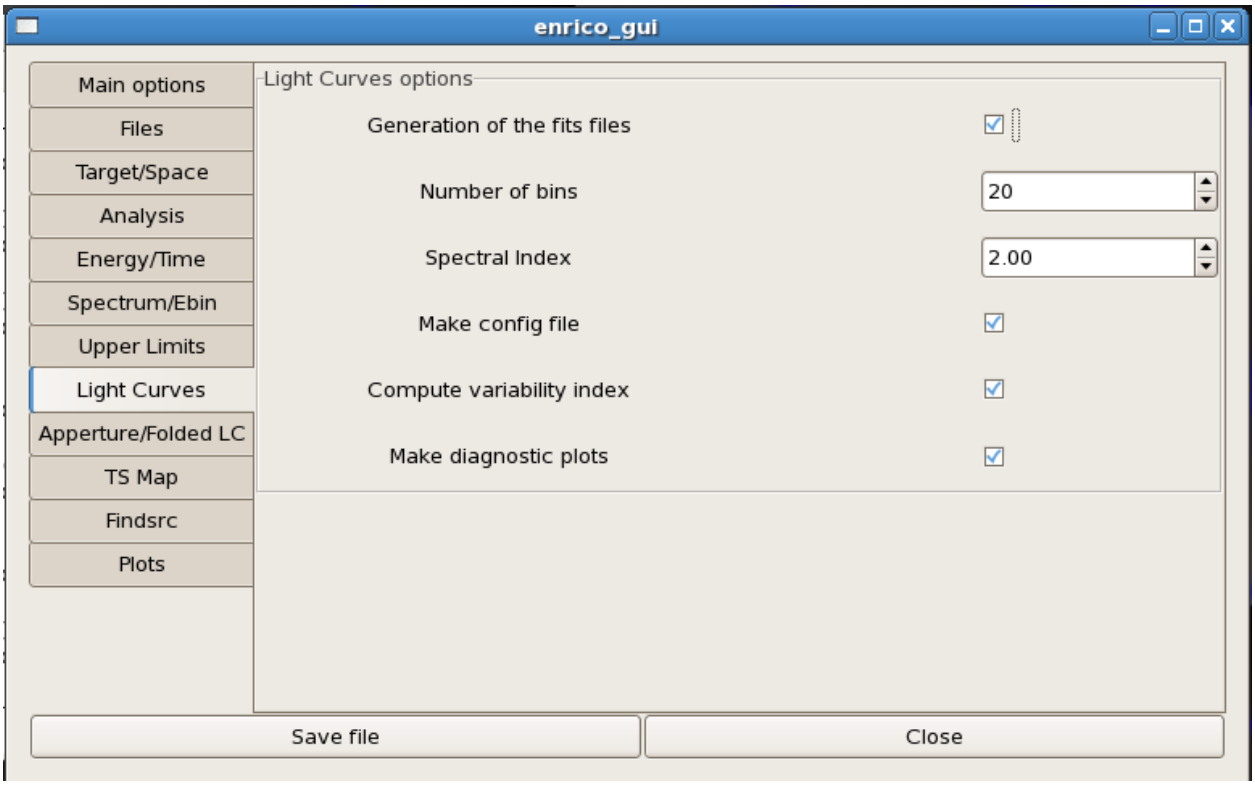
4.4.9 Aperture/FoldedLC page

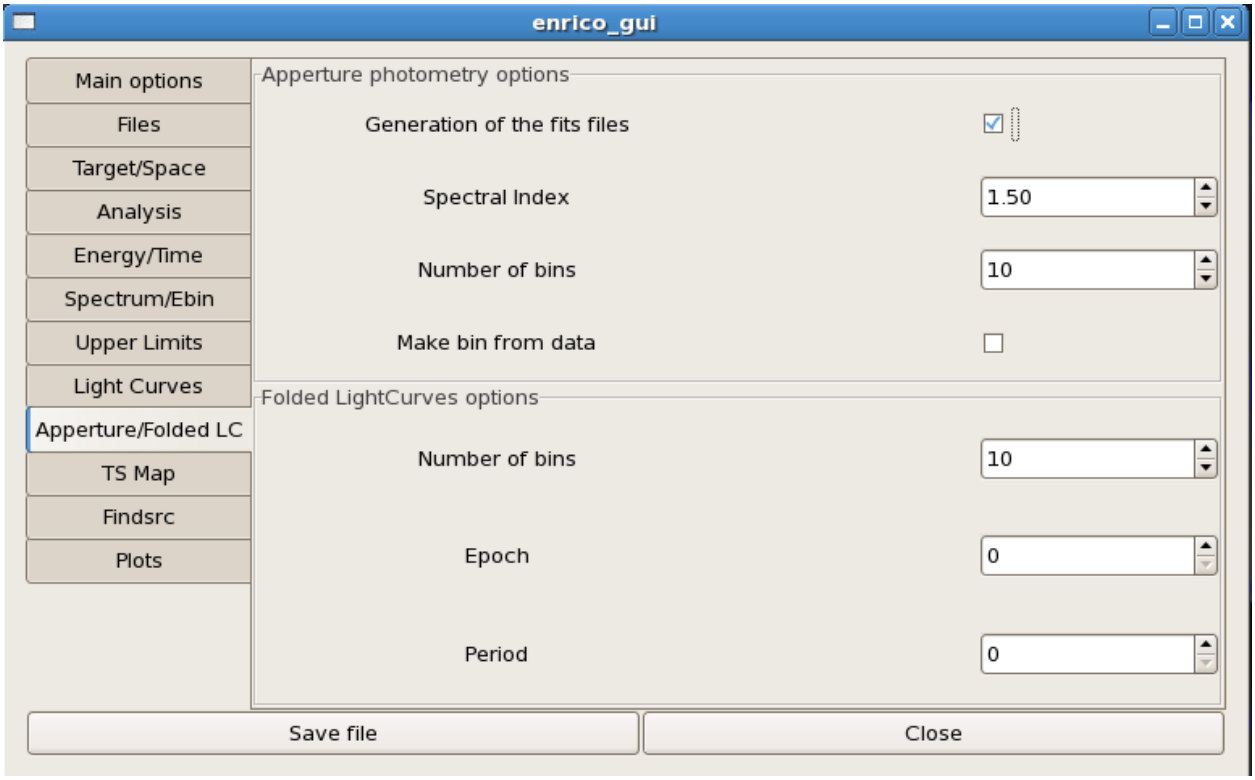
The first frame of the page is for the aperture LC and the second for the Folded LC.

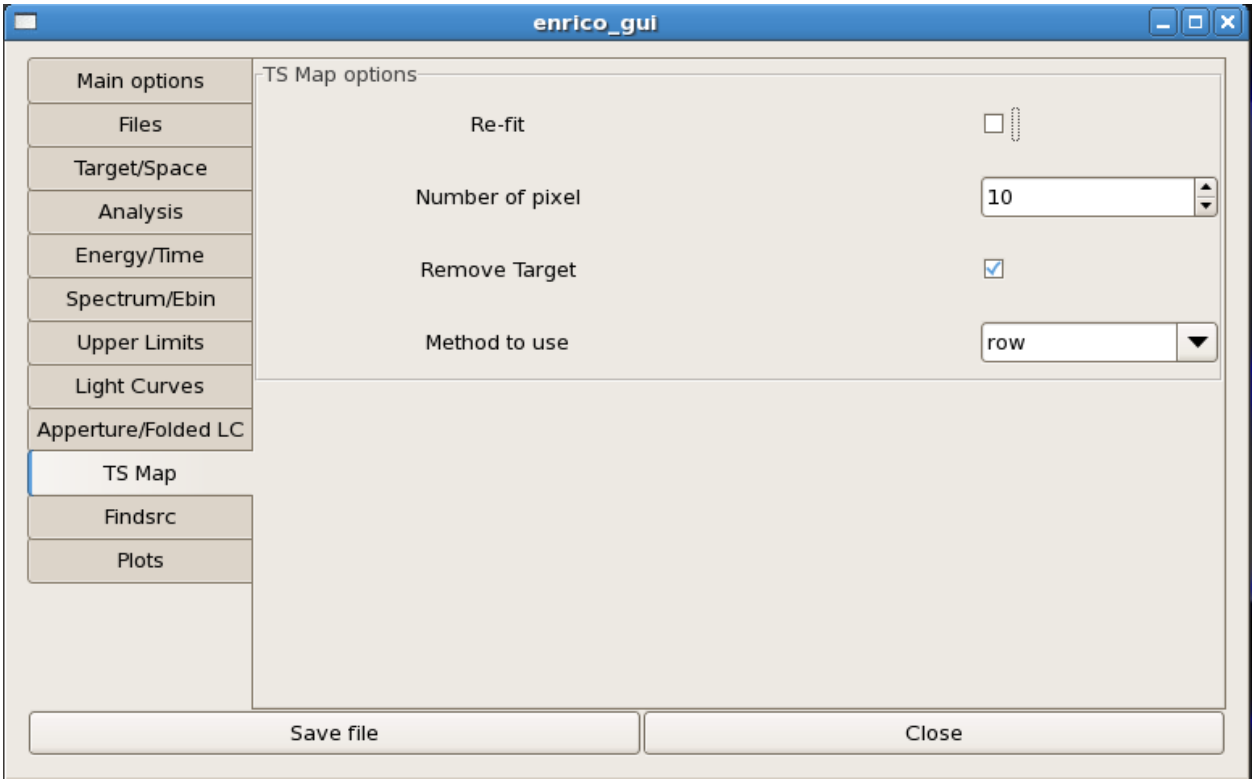
4.4.10 TS Map page

TS Map parameters are managed here


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
Upper Limits options
Assumed Spectral Index value
1.50
Minimal TS
25.0
Confidence level
0.95
Envelope UL
Save file
Close

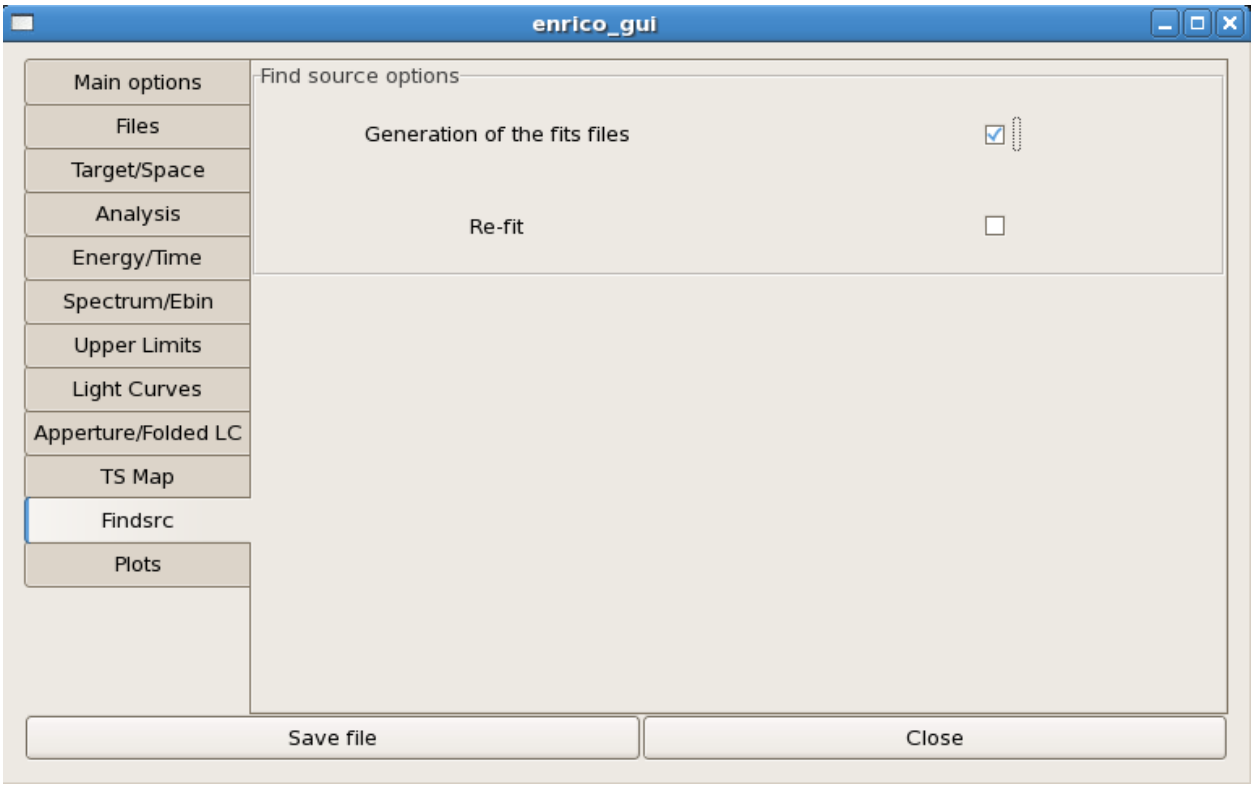

enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
Light Curves options
Generation of the fits files
Number of bins
20
Spectral Index
2.00
Make config file
Compute variability index
Make diagnostic plots
Save file
Close


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
Apperture photometry options
Generation of the fits files
Spectral Index
1.50
Number of bins
10
Make bin from data
Folded LightCurves options
Number of bins
10
Epoch
0
Period
0
Save file
Close


enrico_gui
Main options
Files
Target/Space
Analysis
Energy/Time
Spectrum/Ebin
Upper Limits
Light Curves
Apperture/Folded LC
TS Map
Findsrc
Plots
TS Map options
Re-fit
Number of pixel
10
Remove Target
Method to use
row
Save file
Close

4.4.11 Findsrc page

The findsrc tool parameters are managed here



4.4.12 Plots page

This page allow to draw the produced plots. Using the buttons, you have access to

- the SED and corresponding debug plots
- the LC and corresponding debug plots

If the plot has not been produced, Enrico Fermi picture is display.

4.5 Description of each tool

ScienceTools are described here

enrico_setupcheck : check your installation

enrico_download: download data and auxiliary files (backgrounds, catalog, etc...)

enrico_config: produce a configuration file

enrico_gui : run the GUI of enrico

enrico_xml: produce an xml file that is use to model the ROI using the 2FGL catalog.

enrico_sed: Run gtlike afer having produced all the need fits files is asked.

enrico_testmodel : compute the log(likelihood) of the models *POWERLAW*, *LogParabola* and *PLExpCutoff*. An ascii file is then produced in the Spectrum folder with the value of the log(likelihood) for each model. You can then use the Wilk's theorem to decide which model best describe the data.

enrico_lc: produce a light-curve by running gtlike in different time bins

enrico_foldedlc: produce a folded light-curve by running gtlike in different time bins. see the specific section in the config file

enrico_applc: produce a light-curve using the aperture photometry technique (see here)

enrico_tsmap : produce a TS map with or without the source of interest.

enrico_plot_sed: plot the SED resulting from enrico_sed

enrico_plot_lc: plot the LC resulting from enrico_lc

enrico_plot_tsmap: plot the TS Map resulting from enrico_tsmap

enrico_scan: make a profile likelihood for each free parameter of the target

enrico_findsrc: run gtfindsource

enrico_help: print help

4.6 Scripts

Few nice and helpful scripts are available. They required the additional python packages and are located under the script folder.

4.6.1 Plot Count Maps

The script is plotMaps.py and allows count, model and residuals map to be plotted. The user must provide a configuration file.

```
python plotMaps.py config [vmax]
```

Options:

- vmax is used to defined the max of the color scale

4.6.2 Plot TS Maps

The script is plotTSMaps.py. The user must provide a configuration file.

```
python plotTSMaps.py  config [vmax] [stretch]
```

Options:

- vmax is used to defined the max of the color scale
- stretch can be linear (default), log, sqrt, arcsinh or power

4.7 Developer Information

Any improvements to Enrico are welcome!

Please report bugs and contribute features using github.

Or even better (for us :-), fork us on github, make the improvement yourself and make a pull request for your change to be included in the main repo.

If you don't know how to use git and github, check out http://astropy.readthedocs.org/en/latest/development/

4.8 Enrico Developers

Enrico was initially written by David Sanchez and Christoph Deil in 2011.

You can find up to date lists of contributors here or here.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search