<u>Hands-on session</u>

Fermi-LAT data Analysis with the Science tools

Particle and Astroparticle Physics School in Tirana 2020

Davide Serini
Email: davide.serini@uniba.it
davide.serini@ba.infn.it

FERMI-LAT Data: FITS files

- ***The Fermi-LAT*** **data are available for free on the FSSC (*Fermi Science Support Center*) website**
  - **Anybody can download the data, without restrictions!**
- **Data are stored in FITS files**
  - *FITS = Flexible Image Transport System*
- **Fits files can contain both images and tables (similar to excel files)**
  - **They have «*headers*» that describe the data field in each column/image with its unit**

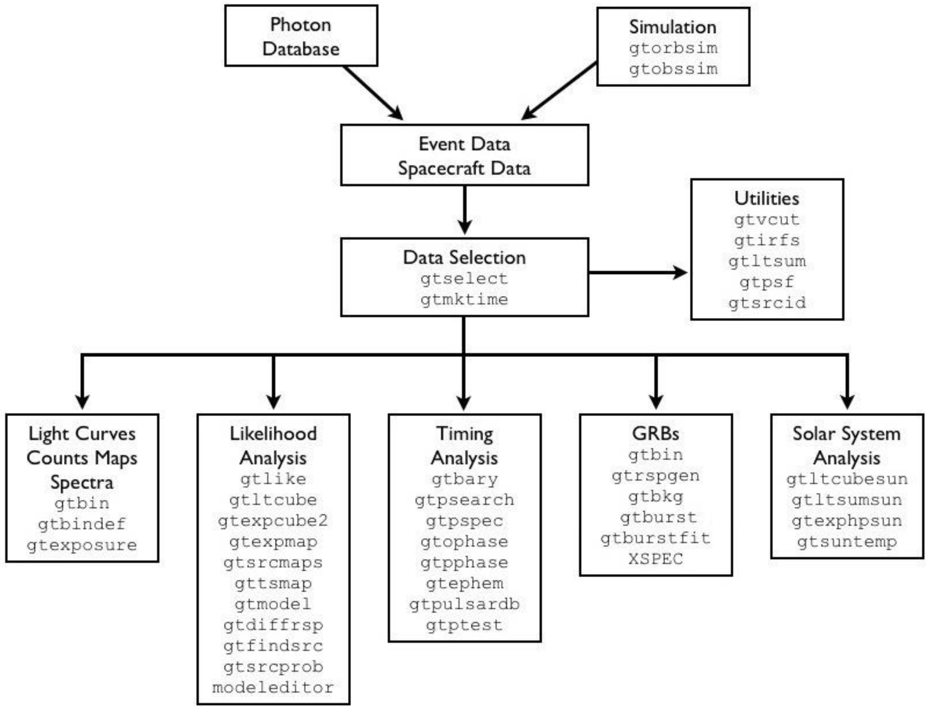
- **Two kind of fits files released:**
  - **FT1 files** (*Event files*): contain a table of data with variables are useful for scientific analysis (i.e. photon direction, energy, etc.)
  - **FT2 files** (*Spacecraft files*): contain a table of data related to the satellite attitude (e.g. pointing, position,...)
    - These data are recorded on 30s time intervals
    - These data are needed to know how a portion of the sky (i.e. a source) was observed at a given time

FERMI Software: Science Tools

- ***The Fermi-LAT collaboration has developed a software suite called «Science Tools» that consists of the basic tools necessary to analyze Fermi LAT data***

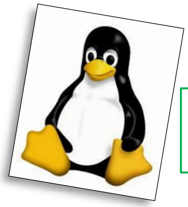
  - ✓ The Fermi Science tools are developed in a hierarchical structure in order to have all softwares that you need for any kind of analysis that you want to do

  - ✓ There are some general tools that are needed to prepare the data to your analysis and specific tools for accurate and precise purpose.


Photon Database
Simulation gtorbsim gtobssim
Event Data Spacecraft Data
Data Selection gtselect gtmktime
Utilities gtvcut gtirfs gtltsum gtpsf gtsrcid
Light Curves Counts Maps Spectra gtbin gtbindef gtexposure
Likelihood Analysis gtlike gtltcube gtexpcube2 gtexpmap gtsrcmaps gttsmap gtmodel gtdiffrsp gtfindsrc gtsrcprob modeleditor
Timing Analysis gtbary gtpsearch gtpspec gtophase gtpphase gtephem gtpulsardb gtptest
GRBs gtbin gtrspgen gtbkg gtburst gtburstfit XSPEC
Solar System Analysis gtltcubesun gtltsumsun gtexphpsun gtsuntemp

Linux O.S.

- **The Science tools are based on the *linux architecture* that is an open source operative system (like Windows O.S.) commonly used in the Scientific community (and not only .. for example also the Android O.S. is based on Linux architecture)**



Why do we use Linux?

- ✓ ***Free, open and customizable***: It is an open source operating system, so it is free to use and also free to study the source code.
- ✓ ***Compatibility:*** A large part of the scientific software is developed in order to run on Linux (both personal computers and large computing farms)
- ✓ ***Simplicity:*** It has simple visual and textual interfaces.

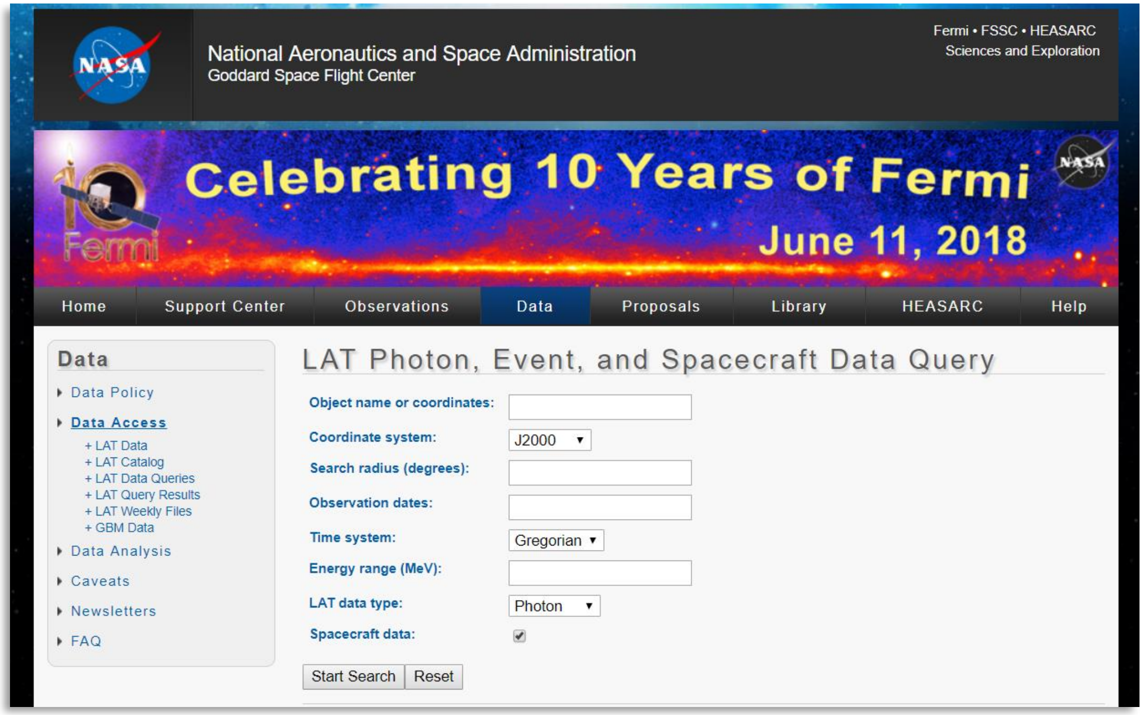
The pc that we are going to use have a distribution of Linux called Mint

The Fermi Science Support Center (FSSC)

All the photon Fermi data and analysis software are publicly available on the NASA website. You can freely download all the material we will use today!!!

Here you can select and download all the Fermi data that you want from here:

https://fermi.gsfc.nasa.gov/cgi-bin/ssc/LAT/LATDataQuery.cgi



LAT Photon, Event, and Spacecraft Data Query

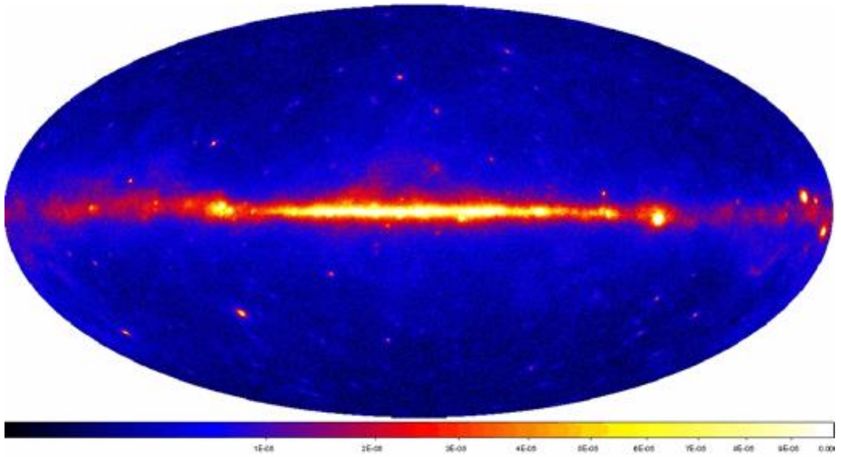
Object name or coordinates:	
Coordinate system:	J2000
Search radius (degrees):	
Observation dates:	
Time system:	Gregorian
Energy range (MeV):	
LAT data type:	Photon
Spacecraft data:	✓

Power of the Science tools: Explore the Fermi data

- **Using the Science tools you can create various kind of files for your analysis.**
  - ✓ **The files that we will create today are:**

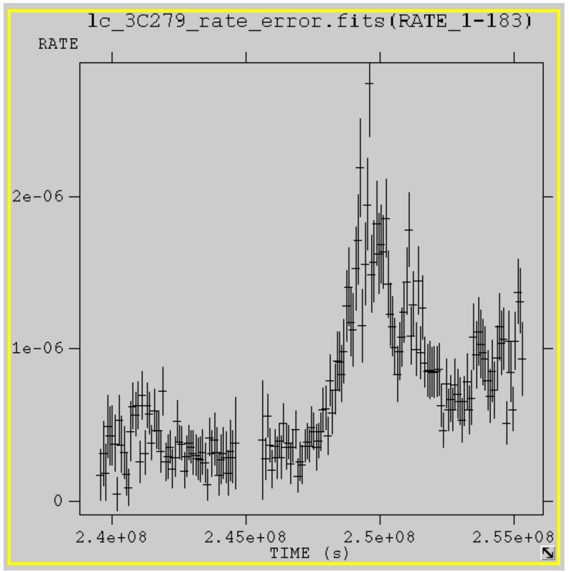
Count map:

- is a 2D plot of the observed photons from a sky region
- it is a "picture" of the gamma- ray sky at high energies.



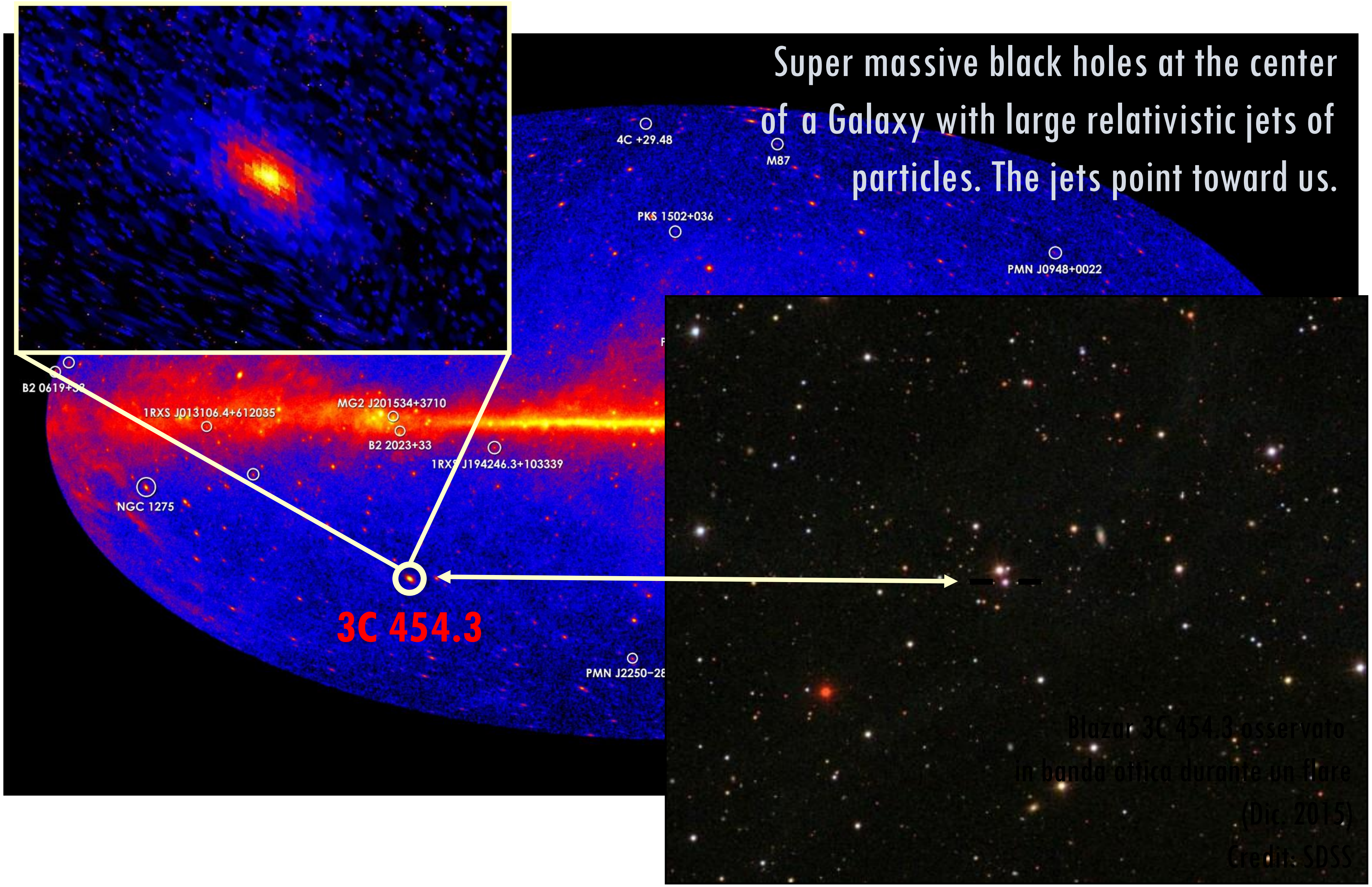
Light curve:

- is a graphical representation of the flux from a position in the sky as a function of time
- This kind of plot is useful to study variations of the luminosity of a source as a function of time



What we are going to analyze: the Blazar 3C 454.3

We are going to analyze an Active Galactic Nucleus (AGN).

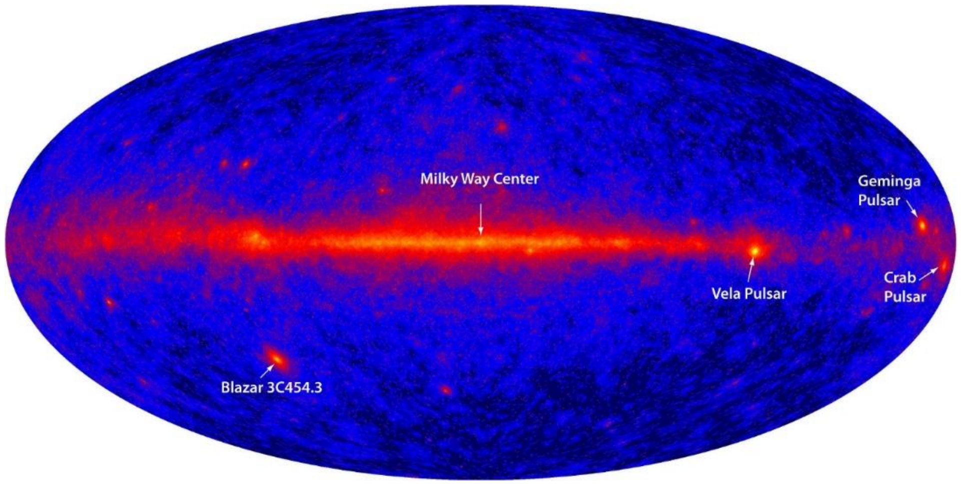


Super massive black holes at the center of a Galaxy with large relativistic jets of particles. The jets point toward us.

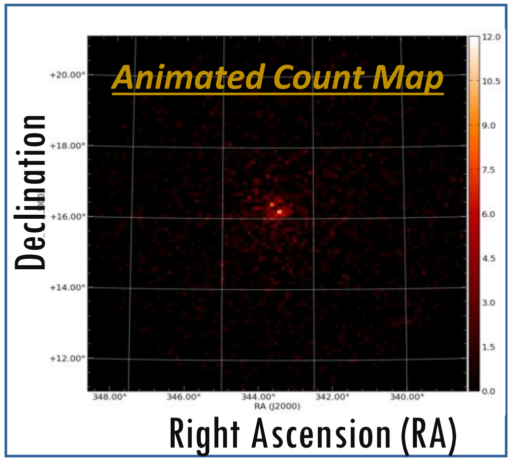
Blazar 3C 454.3 osservato in banda ottica durante un flare (Dic. 2015) Credit: SDSS

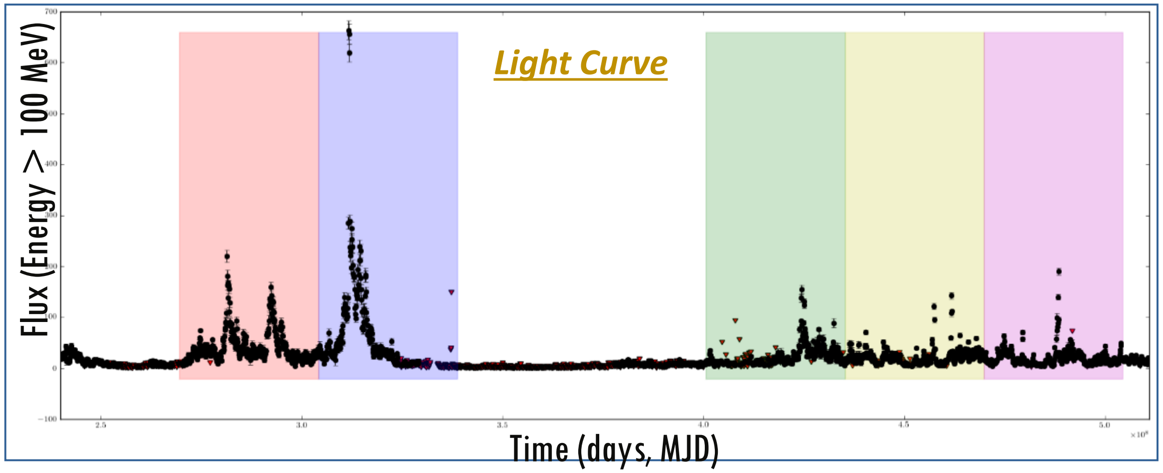
Blazar 3C 454.3 Analysis

3C 454.3 *is a blazar located away from the galactic plane. It is one of the brightest gamma ray sources in the sky, and is one of the most luminous astronomical object ever observed*



*The analysis that we will perform is called «**Aperture photometry**» and it is a part of the usual scientific AGN analysis*





Tools that we are going to use today…

- ***gtselect***: to select the events that we will consider in the analysis. We can select with respect to time, direction of arrival, energy…
- ***gtmktime***: to select the events depending on the status of the satellite (e.g. pointing direction, observing mode, etc..) and to determine the Good Time Intervals (GTIs), which are necessary to correctly calculate the "livetime" (time spent by the satellite to observe a given region of the sky)
- ***gtexposure***: to evaluate the "exposure", related to the time spent to observe a certain region of the sky
- ***gtbin***: to bin the file in 1D histograms (***light curves***) or 2D histograms (***count maps***)
- ***fcalc***: to execute calculations on fits files
- ***fv***: to show what is inside a fits file

How can we use these tools?

- ***We have implemented some python scripts that execute almost automatically the analysis routines by using the Science tools.***

What is Python?

- Python is an interpreted, high-level, general-purpose programming language.
- Its language constructs and object-oriented approach allow to make clear codes, very simple to interpret also for those are not expert programmers.

Using Python we have created some scripts that manage the same analysis routine commonly used in the scientific community, with the advantage to make it automatic.

- The same analysis can be executed by using command lines to run each tool that you need. In our case the python script will run for you the correct commands needed for the analysis and you just have to change the initial settings for your analysis (i.e. the files that you want to analyze, the galactic coordinates of the source, the time frame…)
- Using python to make this kind of routines is very common because you can generalize your routine for different analysis (i.e. if you want to analyze different sources..)

Python Scripts

- Using python to make this kind of routines is very common because you can generalize your routine for different analysis (i.e. if you want to analyze different sources..).

- The results of our analysis will be managed by the **`NumPy`** library functions
  - Numpy is the fundamental package for scientific computing with Python.

- We will also use python to make nice plots of our results.
  - Some python libraries as **`matplotlib`**, allow to create and customize your plot reaching the standard required in scientific pubblications.

- **`pyfits`** and **`os`** libraries will manage respectively the .fits files to be analyzed and the interaction with your operative system to run the Science tools.

Scripts that we will use…

time_converter.py: The time_converter python script takes a time in MET units (will we see what this unit is) as argument and converts it into a standard format.

MakeLightCurve.py: This script will set and run the routine to make our light curve that will be saved into a `.fits` file

PlotLightCurve.py: This script will read the `lightcurve.fits` file and create a plot of your light curve.

MakeCountMap.py:This script will set and run the routine to make our count map and it will plot it.

Conclusion

- Soon we will work on some real Fermi data
- We will take confidence with the Linux operative system.
- We will read and change some python scripts to make our create count maps and light curves.

Each PC will have its own dataset

- *At the end we will have a lightcurve of the whole dataset*

Initialization

We divided the dataset in 14 parts. Each group will analyze its dataset.
At the end we will merge your results.

Let's start

➢ In Linux O.S. most things are done by using the terminal.

- ✓ **Open a linux terminal clicking on**:
- ✓ type the command: ***pwd***
  - This command will show the folder in which you are
- ✓ Type the command cd Masterclass
  - This command will bring you in the folder where the Masterclass material is found
- ✓ *Then type the command:* ***source SetupScienceTool_v10r0p5.sh***
  - this command loads all we need for our analysis. In this way the PC will be able to run the dedicated software for the LAT data analysis, which will be used in the following sections.

Now the system is ready for the data analysis.

Linux basic operations

- **To enter in the folder in which the data will be analyzed you have to type the command: *cd Fermi***
  - ✓ The command "**cd**" allows to change folder.
  - ✓ Folders are arranged in a tree structure.
    - If you want to move one folder up the syntax is "**cd ..**".
    - If you want to move down you have to specify the name of the folder in which you want to move (e.g. **cd FolderName**) .

- **With the command *"cd Fermi"* you will move from the main folder** /home/Masterclass/ to the folder /home/Masterclass/Fermi/.

DataSET

- **In the table the initial and final times of each interval are listed.**
  - ✓ Times are expressed in **MET** (*Mission Elapsed Time*) **units**, i.e. in seconds from January 1st, 2009.
  - ✓ Our dataset covers about 3 years of observation starting from 1 th of January 2009 to 31 th December 2011.
- **Please take note of your start-stop time! We will need them for our analysis**

Start and stop times for each PC

PC 01	start	252460802	stop	259218516
PC 02	start	259218516	stop	265976230
PC 03	start	265976230	stop	272733944
PC 04	start	272733944	stop	279491658
PC 05	start	279491658	stop	286249372
PC 06	start	286249372	stop	293007086
PC 07	start	293007086	stop	299764800
PC 08	start	299764800	stop	306522514
PC 09	start	306522514	stop	313280228
PC 10	start	313280228	stop	320037942
PC 11	start	320037942	stop	326795656
PC 12	start	326795656	stop	333553370
PC 13	start	333553370	stop	340311084
PC 14	start	340311084	stop	347068798

Convert Time from MET units to standard format

- **Now you may want to convert the times of your interval from MET units into a standard format. To do that, just type the command:**

pyhton scripts/time_converter.py [TIME]

This command will run the python script time_converter.py which is in the folder /Fermi/scripts/ and which takes a time in MET units as argument and converts it into a standard format.

As an example, if you type:

```
>> python scripts/time_converter.py 435628803
```

you will get: 22/10/2014 00:00:03

Let’s start our analysis section

Now we are able to start our analysis with the FERMI Data.

- **First of all we can try to take a look at the data files.**
  - ✓ In the folder Data we have 2 .fits files that are:
    - The ***FT1 file*** *PHOTONS.fits* for the ***events***
    - The ***FT2 file*** *SC00.fits* for the ***spacecraft position informations***.
- To explore fits files we will use the program “fv”.

Let’s see the FT2 file. Type the command:

```
>> fv Data/SC00.fits
```

You can see a graphical window that allows us to read/explore the content of this file

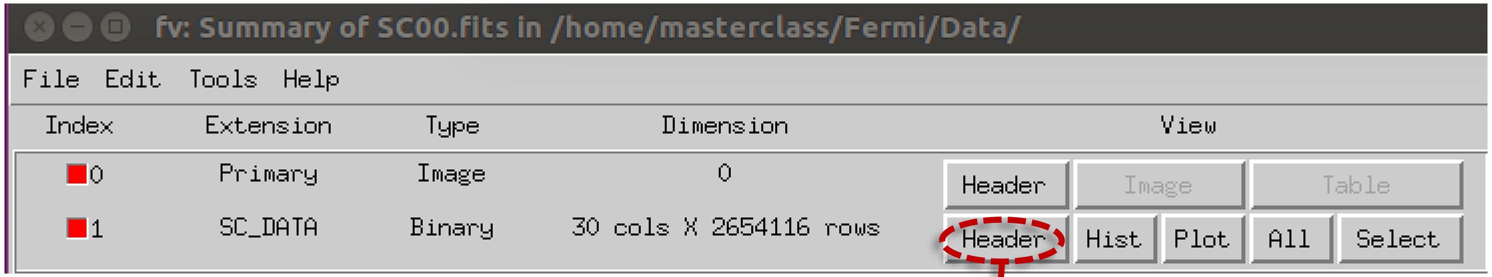
- **Fits files can contain both images and tables (as an excel file). They have «headers» that describe what his in each column/image with its unit.**
  - ✓ Take a look to the SC_DATA Header to understand what there is in our file

fv program: explore FT2 file (spacecraft)

- **Now we can explore the FT2 file (spacecraft) Type the command**

```
>> fv Data/SC00.fits
```

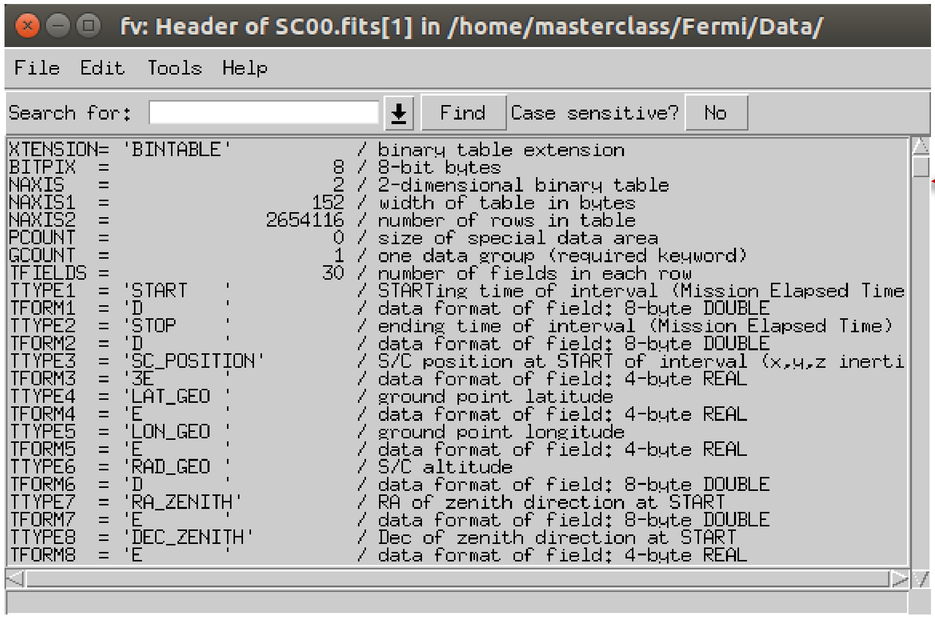
- **In this case you can see the Row «SC_DATA» (spacecraft Data) that contains a table with the satellite position each 30 s (e.g. pointing, position,...)**
  - ✓ We can take a look to the header file to see what there is in our ft2 file



fv: Summary of SC00.fits in /home/masterclass/Fermi/Data/

File Edit Tools Help

Index	Extension	Type	Dimension	View
🟥0	Primary	Image	0	Header Image Table
🟥1	SC_DATA	Binary	30 cols X 2654116 rows	Header Hist Plot All Select



fv: Header of SC00.fits[1] in /home/masterclass/Fermi/Data/

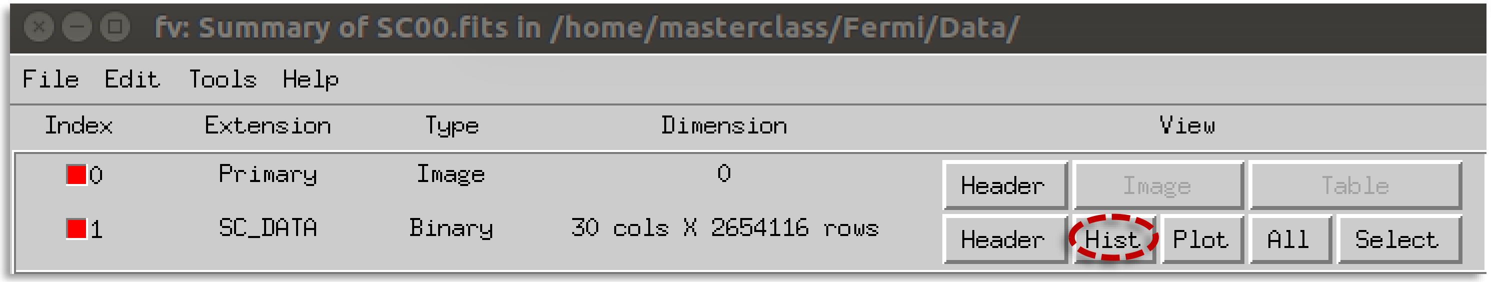
File Edit Tools Help

Search for: Find Case sensitive? No

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                  152 / width of table in bytes
NAXIS2  =              2654116 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                   30 / number of fields in each row
TTYPE1  = 'START   '           / STARTing time of interval (Mission Elapsed Time
TFORM1  = 'D       '           / data format of field: 8-byte DOUBLE
TTYPE2  = 'STOP    '           / ending time of interval (Mission Elapsed Time)
TFORM2  = 'D       '           / data format of field: 8-byte DOUBLE
TTYPE3  = 'SC_POSITION'        / S/C position at START of interval (x,y,z inerti
TFORM3  = '3E      '           / data format of field: 4-byte REAL
TTYPE4  = 'LAT_GEO '           / ground point latitude
TFORM4  = 'E       '           / data format of field: 4-byte REAL
TTYPE5  = 'LON_GEO '           / ground point longitude
TFORM5  = 'E       '           / data format of field: 4-byte REAL
TTYPE6  = 'RAD_GEO '           / S/C altitude
TFORM6  = 'D       '           / data format of field: 8-byte DOUBLE
TTYPE7  = 'RA_ZENITH'          / RA of zenith direction at START
TFORM7  = 'E       '           / data format of field: 8-byte DOUBLE
TTYPE8  = 'DEC_ZENITH'         / Dec of zenith direction at START
TFORM8  = 'E       '           / data format of field: 4-byte REAL
```

- ✓ We can take a look, for example to the ***Orbit*** of the spacecraft during all our time

fv program: Spacecraft Orbit


fv: Summary of SC00.fits in /home/masterclass/Fermi/Data/

File Edit Tools Help

Index	Extension	Type	Dimension	View
🟥0	Primary	Image	0	Header Image Table
🟥1	SC_DATA	Binary	30 cols X 2654116 rows	Header Hist Plot All Select

1. select Hist in the «***SC_DATA***» row and build in the same way as before, the **LON_GEO** vs **LAT_GEO** histogram that represents the *latitude* and the *longitude* of the spacecraft with respect to the ground.

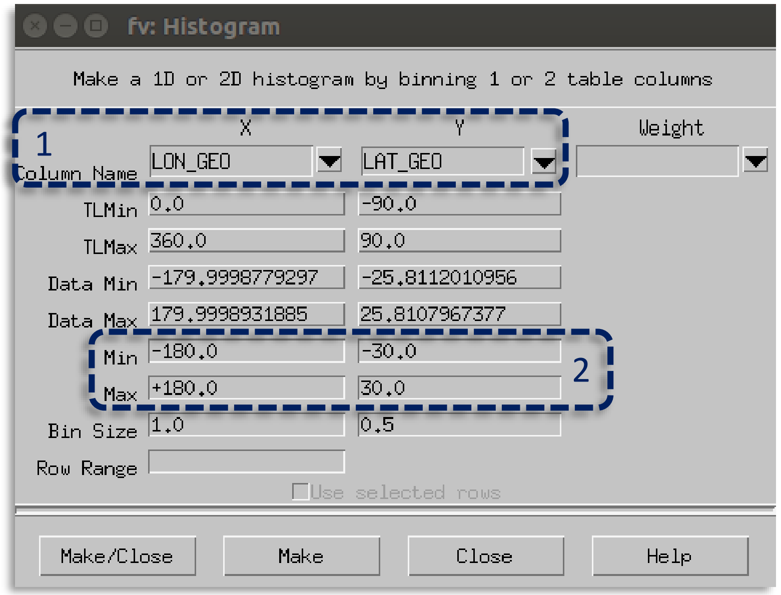
2. Fermi has an equatorial orbit with an inclination angle of 28° and a period of ~ 90 minutes. It means that the latitude of the spacecraft can vary of ± 28°.

So we can select: **for LON_GEO: X_min = -180° , X_max = +180°**

for LAT_GEO: Y_min = -30° , Y_max = +30°

The result is the overlap of all the orbits that the spacecraft has done during our 3 years of time-interval.

Orbit histogram

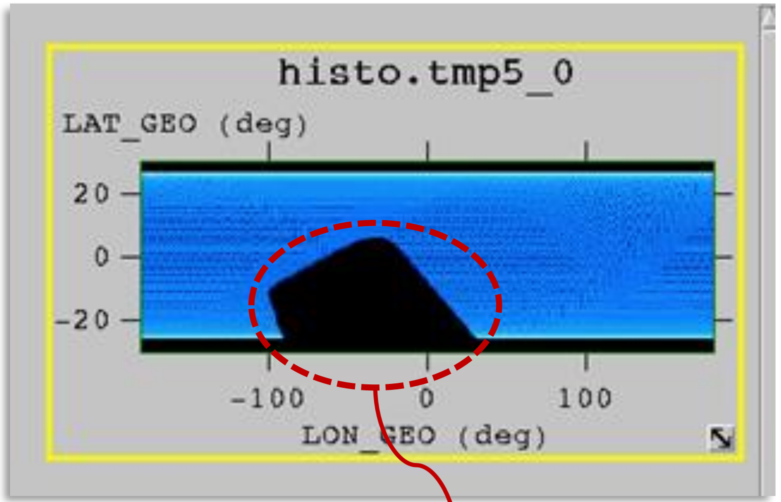


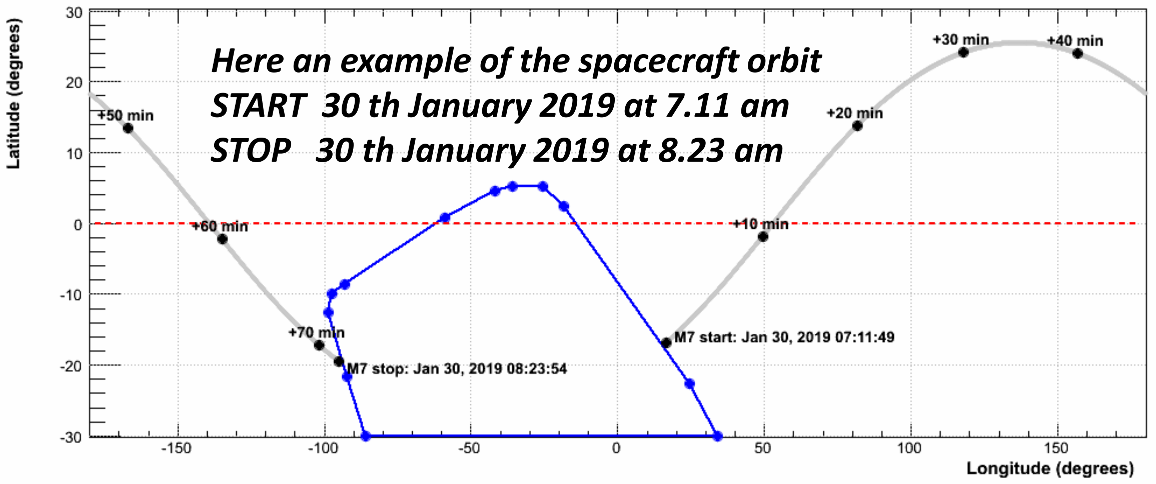
Make a 1D or 2D histogram by binning 1 or 2 table columns

	X	Y	Weight
Column Name	LON_GEO	LAT_GEO	
TLMin	0.0	-90.0	
TLMax	360.0	90.0	
Data Min	-179.9998779297	-25.8112010956	
Data Max	179.9998931885	25.8107967377	
Min	-180.0	-30.0	
Max	+180.0	30.0	
Bin Size	1.0	0.5	
Row Range			

☐ Use selected rows

Make/Close | Make | Close | Help





Here an example of the spacecraft orbit
START 30 th January 2019 at 7.11 am
STOP 30 th January 2019 at 8.23 am

<u>This is the SAA (*South Atlantic Anomaly*)</u>

The Earth magnetic field in this region leads to an increased flux of energetic particles and exposes orbiting satellites to higher-than-usual levels of radiation. In this region the data acquisition is turned OFF

MakeCountMap.py

While *fv* can be used to make quick plots when exploring the data, it doesn't automatically do all the things you would like when making data files for analysis.

- Now we can build a ***count map*** of the specific region to be analyzed by reading the ft1 file using the python script **`MakeCountMap.py`**
- The python script **`MakeCountMap.py`** will read all data contained in the PHOTONS.fits file and will make a nice count map using matplotlib functions.
- Before to run the code we need to open and editate it in order to select the time frame and the binning for the histrogram (like pixels in a picture!)
- To edit the code use the command

```
>>gedit MakeCountMap.py &
```

MakeCountMap.py (1)

```
import os
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import pyfits
```

Here we load all the libraries that we need to run the script

```
# Define input file names

ft1file = "Data/PHOTONS.fits"

# Define minimum and maximum times

met_start = 252846681
met_stop = 272846681

# Define binning for the map (in
right ascension and declination)

nra = 500
ndec = 500
```

Here we set all parameters we need to run gtbin

MakeCountMap.py (2)

```
# Define the histogram with the map

fig = plt.figure()
mymap,xedges,yedges = np.histogram2d(vra,vdec,bins=(nra,ndec))
ax = fig.add_subplot(111, title='Photon count
map',aspect='equal',xlim=xedges[[0,-1]],ylim=yedges[[0,-1]])

mesh = ax.pcolormesh(xedges, yedges, mymap, cmap=mpl.cm.hot)
mesh.set_clim(0,20)
cbar = fig.colorbar(mesh,ax=ax)
cbar.ax.set_ylabel('Counts')
plt.xlabel('Right ascension (degrees)')
plt.ylabel('Declination (degrees)')
plt.show()
fig.savefig(file_output)
```

This part of code define all settings for the plot (i.e. `plt.xlabel('Right ascension (degrees)’)` set the label string on the x_axis.
In particular, the command mesh.set_clim(0,20) set the minumum and maximum limits of the number of photons per bin.

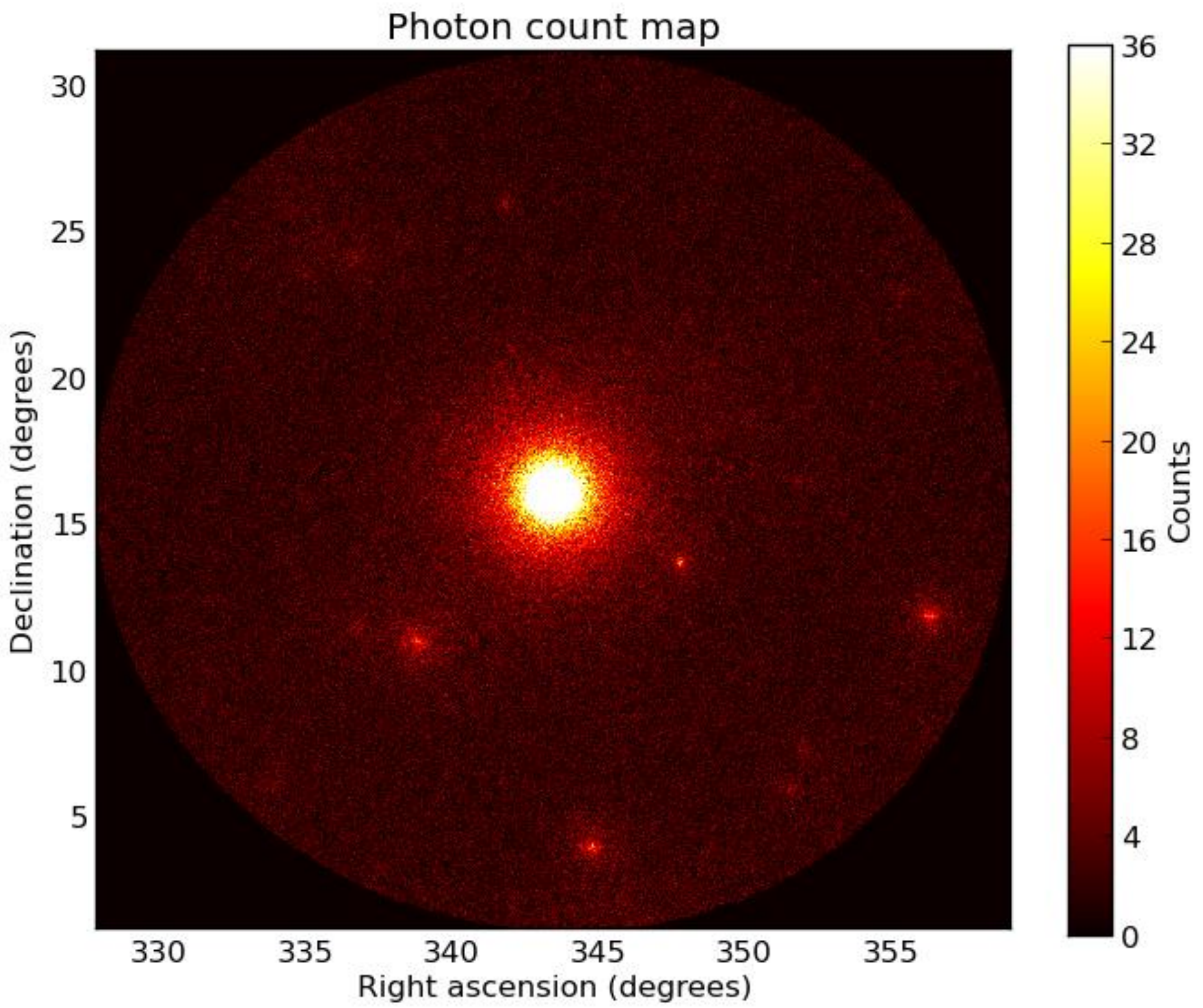
Run the script using the command

```
>>python MakeCountMap.py
```

and try to make some different plots by changing the upper limit in the set_clim setting.
This script will produce a .png file that you can see using the command

```
>>eog picture_name.png
```

Your Count Map



MakeLightCurve.py

This script run all the tools to make your light curve for your *Aperture photometry analysis*. This complete analysis require to run more than one Science tool in a dedicate routine.

1. **The first step to build your light curve is the selection of the data sample for your analysis.**
   - ✓ We will select photons within 1° from the source 3C454.3.

   *This will be done using the “**`gtselect`**” **software.***

2. **The next step is the selection of the good time intervals (GTIs) for the analysis.**
   - ➢ **GTIs are those time intervals where:**
     - ✓ the LAT was operating in “*standard*” conditions,
     - ✓ the source was within the field of view (FoV) of the instruments (and possibly well separated from the Sun)
     - ✓ the Earth Limb was outside the FoV.

   *The selection of GTIs is done working on the FT2 file with the software “**`gtmktime`**”.*

MakeLightCurve.py (1)

3. After the selection you need to divide your dataset in 1-day bins (86400 s)

*this operation is done by " `gtbin` " **software.***

4. The next step is to calculate the exposure of the source in the selected time interval.

✓ This tool computes the exposure (cm^2 s) associated with each time bin, allowing for a light curve in photons/s to be computed.

*this calculation is done with the **"`gtexposure`" software.***

5. Finally yon need to evaluate the rate as the ratio between the counts and the exposure and its error.

$$\Phi = \frac{counts(E,t)}{exposure(E,t)}$$

The error on the flux is evaluated as

$$\sigma_{\Phi} = \frac{\sigma_{counts}}{exposure}$$

*These evaluations are done by the software **"`fcalc`"** that execute operation among the columns in the .fits files and create a new column with the results.*

MakeLightCurve.py (2)

To recap..

The *Aperture photometry analysis* that we are going to perform in order to make our light curve, need the following steps:

$$\begin{tikzcd}
\text{gtselect} \arrow[r] & \text{gtmktime} \arrow[r] & \text{gtbin} \arrow[r] & \text{gtexposure} \arrow[r] & \text{fcalc}
\end{tikzcd}$$

gtselect	gtmktime	gtbin	gtexposure	fcalc
To select your Region of interest (RoI)	*To select data in the Good Time intervall (GTI)*	*To divide your time frame in bins (i.e. 1 bin = 1 day of observation)*	*To evaluate the exposure*	*To evaluate the rate of photons detected from your RoI and its error*

The script MakeLightCurve.py execute for you all these commands but you need to set some input values for your analysis

- As before, we need to open and editate it in order to set correctly all the imput values
- To editate the code use the command

```
>>gedit MakeLightCurve.py &
```

Light curve in your time interval

```
# Define source coordinates (right ascension and declination)
source_ra = 343.49
source_dec = 16.15

# Define times
met_start = 252846681
met_stop = 272846681

# Define time binning for the light curve (1 day = 86400 s)
delta_time = 86400

# Define minimum and maximum energies
emin = 100
emax = 300000

# Define angular radius (in degrees)
of the circular region around the source
rad = 1.0

# Define maximum zenith angle
zenmax = 90.0

# Define input file names
ft2file = "Data/SC00.fits"
ft1file = "Data/PHOTONS.fits"
```

PUT here your start and stop time in MET!!

To run the script:

```
>>python MakeLightCurve.py
```

PlotLightCurve.py

When the execution of `MakeLightCurve.py` finish, a *lightcurve.fits* will be produced.

- *In a similar way as before, we use the script **`PlotLightCurve.py`** to read the file lightcurve.fits and make a nice plot that represents the photon flux as a function of the time.*
  - Some of you will find an almost constant flux of photons, instead some others will find one or more peaks that correspond to a flare!
- As usual, before to run the script, we need to editate it to correctly set the input parameters.
- To editate the code use the command

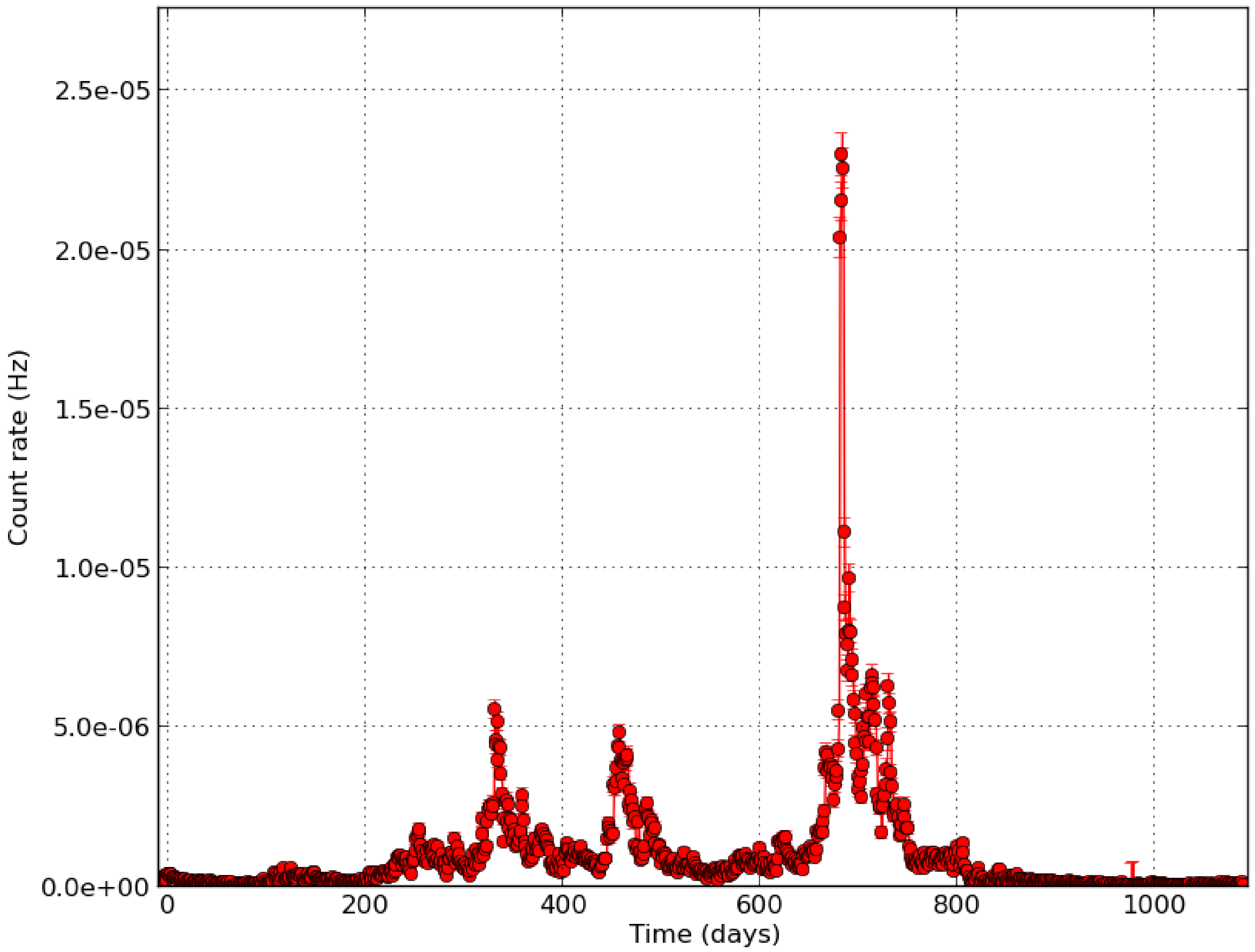
```
>>gedit PlotLightCurve.py &
```

```
# Useful variables
met_start = 252460802
```

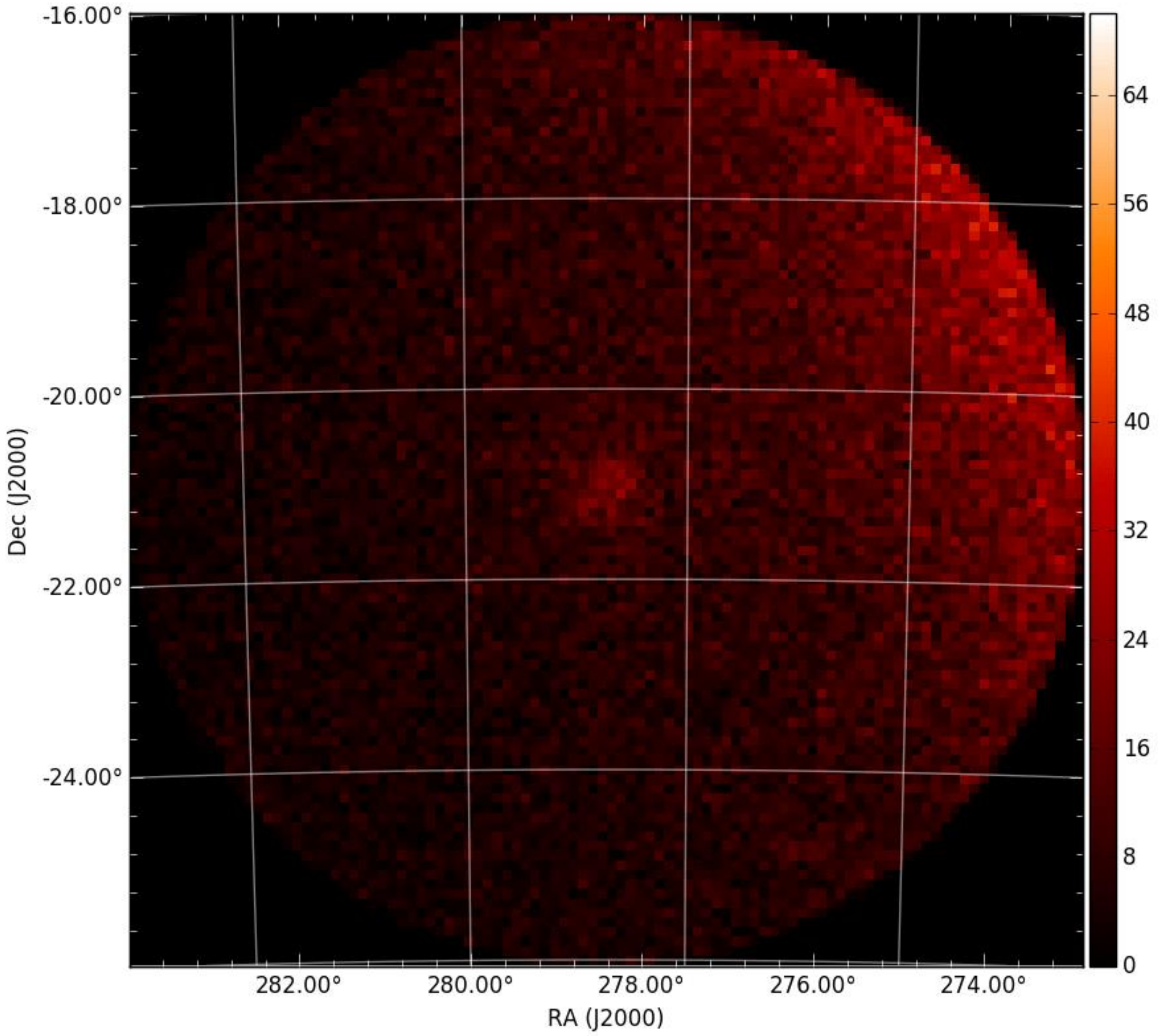
To run the script:

```
>>python MakeLightCurve.py
```

PlotLightCurve.py



3C 454.3 Animated gif



BACKUP

Let's start our analysis section

Now we are able to start our analysis with the FERMI Data.

- **First of all we can try to take a look at the data files.**
  - ✓ In the folder Data we have 2 .fits files that are:
    - The ***FT1 file*** *PHOTONS.fits* for the ***events***
    - The ***FT2 file*** *SC00.fits* for the ***spacecraft position informations***.
- To explore fits files we will use the program "fv".

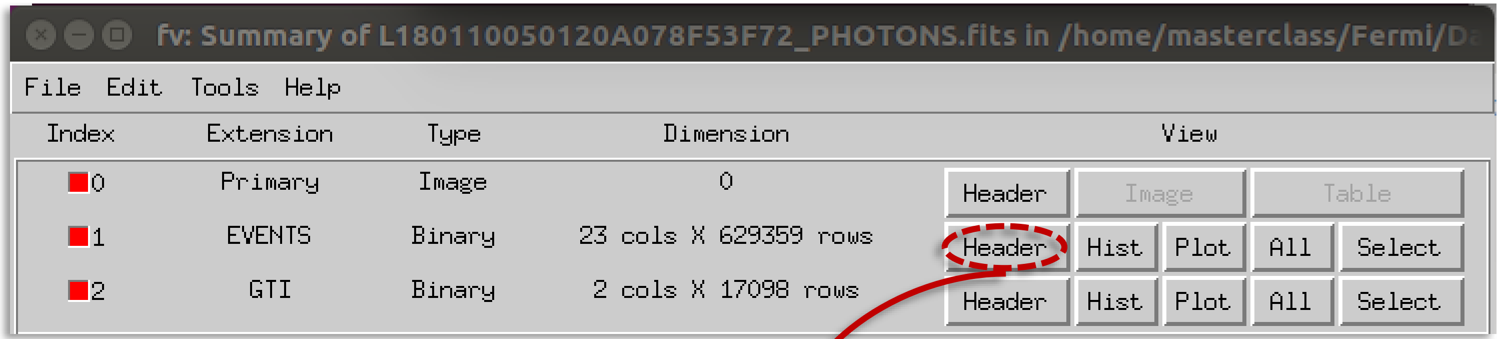
Let's start with the FT1 file. Type the command:

```
>> fv Data/PHOTONS.fits
```

You can see a graphical window that allows us to read/explore the content of this file

- **Fits files can contain both images and tables (as an excel file). They have «headers» that describe what his in each column/image with its unit.**
  - ✓ Take a look to the EVENTS Header to understand what there is in our file

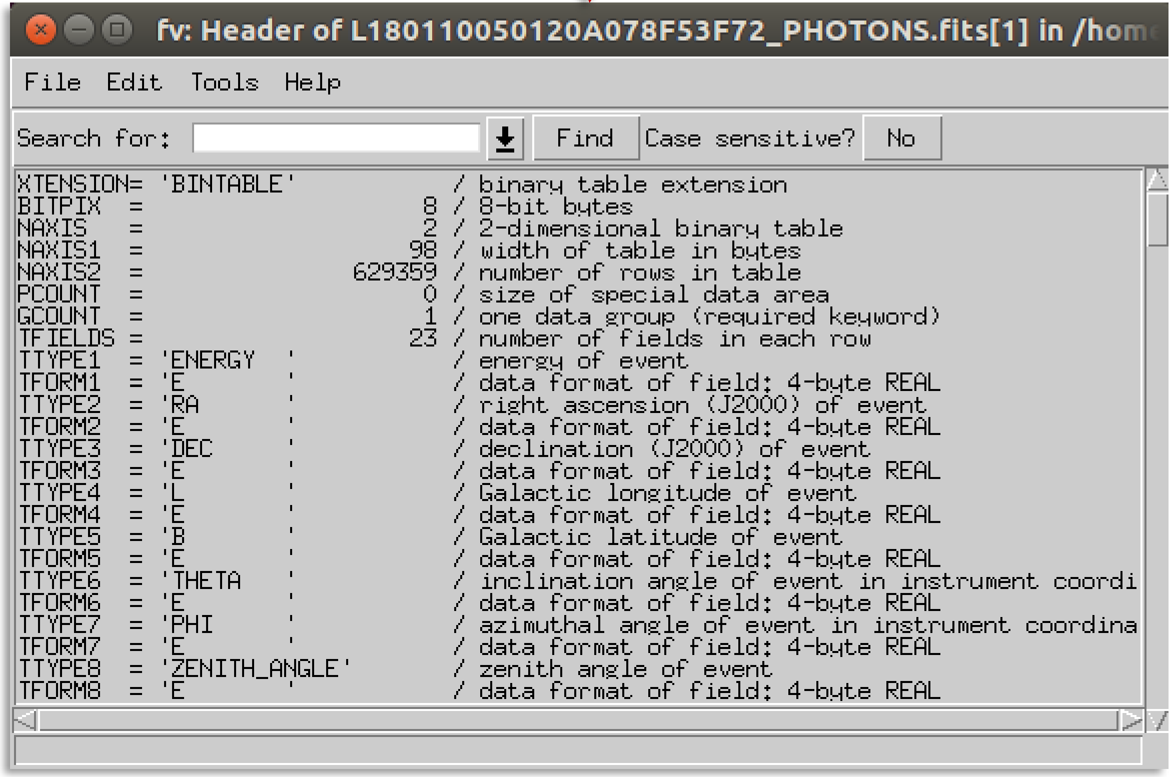
FT1 file: EVENTS Header



fv: Summary of L180110050120A078F53F72_PHOTONS.fits in /home/masterclass/Fermi/D

File Edit Tools Help

Index	Extension	Type	Dimension	View
🟥 0	Primary	Image	0	Header Image Table
🟥 1	EVENTS	Binary	23 cols X 629359 rows	Header Hist Plot All Select
🟥 2	GTI	Binary	2 cols X 17098 rows	Header Hist Plot All Select



fv: Header of L180110050120A078F53F72_PHOTONS.fits[1] in /home

File Edit Tools Help

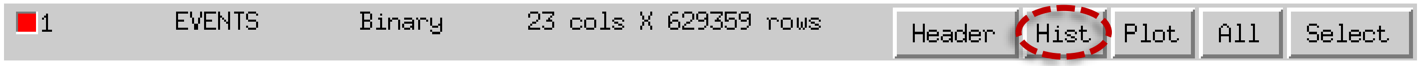
Search for: Find Case sensitive? No

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   98 / width of table in bytes
NAXIS2  =               629359 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                   23 / number of fields in each row
TTYPE1  = 'ENERGY  '           / energy of event
TFORM1  = 'E       '           / data format of field: 4-byte REAL
TTYPE2  = 'RA      '           / right ascension (J2000) of event
TFORM2  = 'E       '           / data format of field: 4-byte REAL
TTYPE3  = 'DEC     '           / declination (J2000) of event
TFORM3  = 'E       '           / data format of field: 4-byte REAL
TTYPE4  = 'L       '           / Galactic longitude of event
TFORM4  = 'E       '           / data format of field: 4-byte REAL
TTYPE5  = 'B       '           / Galactic latitude of event
TFORM5  = 'E       '           / data format of field: 4-byte REAL
TTYPE6  = 'THETA   '           / inclination angle of event in instrument coordi
TFORM6  = 'E       '           / data format of field: 4-byte REAL
TTYPE7  = 'PHI     '           / azimuthal angle of event in instrument coordina
TFORM7  = 'E       '           / data format of field: 4-byte REAL
TTYPE8  = 'ZENITH_ANGLE'       / zenith angle of event
TFORM8  = 'E       '           / data format of field: 4-byte REAL
```

fv program: explore FT1 file (data)

- ***fv program give us also the possibility to see histograms and plots.***
  - ✓ We can try to explore the FT1 file building a count map!

Select Hist **from EVENTS Row.**



```
■1          EVENTS        Binary        23 cols X 629359 rows     Header  Hist  Plot  All  Select
```

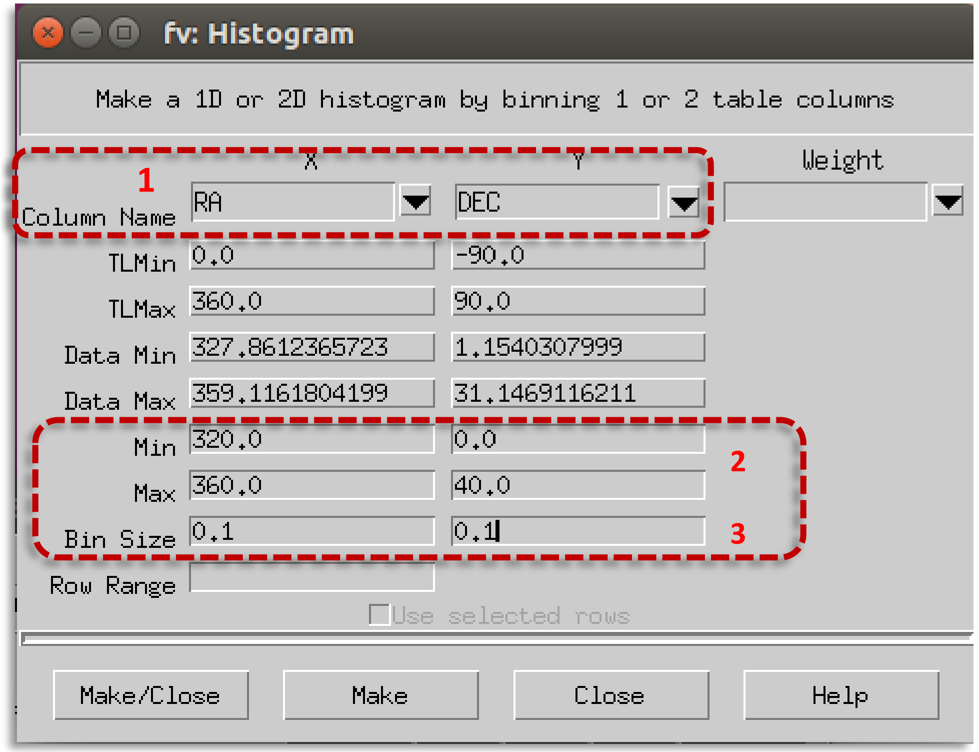
1. *Select* **RA** ***(Right Ascension, α)*** in X column and **DEC** (***Declination, δ***) in Y column.
2. Choose your Min and Max values in order to center more or less your Region of Interest (in our case the blazar **3C 454.3)**

   In our case RA = 343.49° and DEC = 16.15° so a good interval to build our COUNT MAP is:

   X_min = 320° , X_max = 360° and Y_min = 0° , Y_max = 40°
3. Finally choose a good binning for your histogram that can be 0.1, it means that we divide our range in intervals of 0.1°.

- ***After setting these information, click on*** Make/Close ***and you will see your first Count Map.***

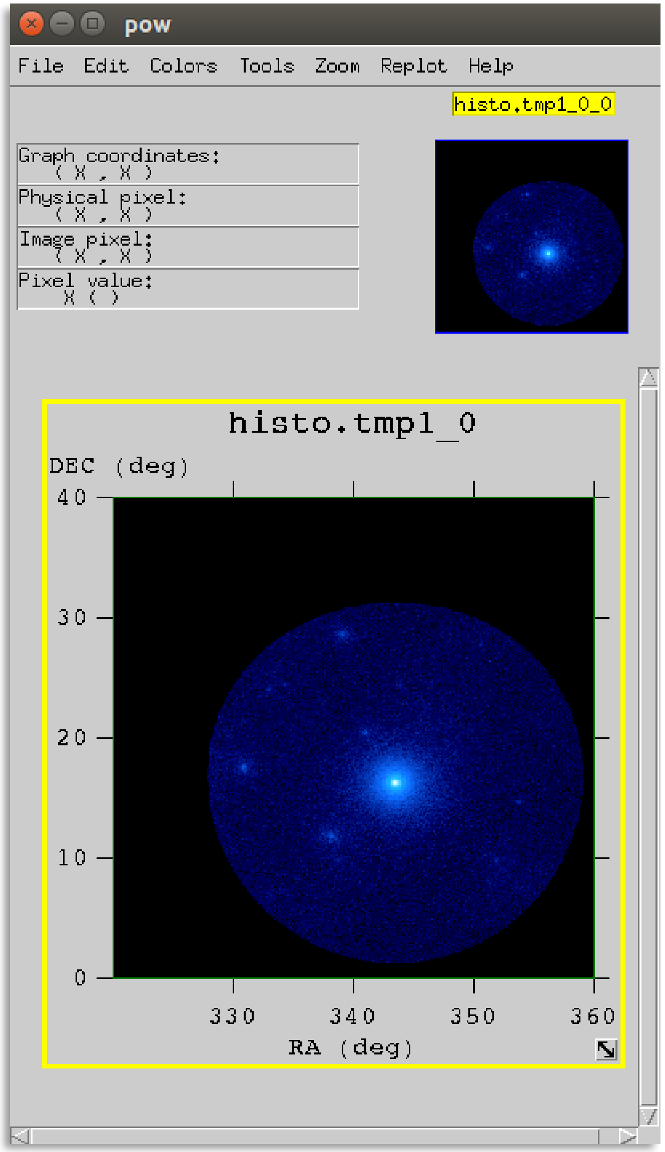
Count Map



fv: Histogram

Make a 1D or 2D histogram by binning 1 or 2 table columns

	X	Y	Weight
Column Name	RA	DEC	
TLMin	0.0	-90.0	
TLMax	360.0	90.0	
Data Min	327.8612365723	1.1540307999	
Data Max	359.1161804199	31.1469116211	
Min	320.0	0.0	
Max	360.0	40.0	
Bin Size	0.1	0.1	
Row Range			



To make your histogram nicer you can change the color scale from the section Colors (top) and choose the palette that you prefer

(for example Color➔Continuous➔cold)

The color gradient scale represents the number of counts (photons detected by FERMI) in each bin.
As you can see our source appears very bright.

Light curve in your time interval

- **The first step to build your light curve is the selection of the data sample for your analysis.**
  - ✓ We will select photons within 1° from the source 3C454.3.
  - ✓ This will be done using the "**gtselect**" software.

- **Type the command "gtselect" with the following inputs:**

```
>> gtselect
>> Input FT1 file[] Data/L180110050120A078F53F72_PHOTONS.fits
>> Output FT1 file[] temp_gtselect.fits
>> RA for new search center (degrees) (0:360) [343.490617] 343.49
>> Dec for new search center (degrees) (-90:90) [16.148211] 16.15
>> radius of new search region (degrees) (0:180) [1] 1
>> start time (MET in s) (0:) [435196803] 435628803
>> end time (MET in s) (0:) [469756804] 436752003
>> lower energy limit (MeV) (0:) [100] 100
>> upper energy limit (MeV) (0:) [200000] 200000
>> maximum zenith angle value (degrees) (0:180) [90] 90
>> Done.
```

PUT here your start and stop time in MET

- ***Our output will be: temp_gtselect.fits***

Gtmktime: Selection of GTIs

- **The next step is the selection of the good time intervals (GTIs) for the analysis.**
- **GTIs are those time intervals where:**
  - ✓ the LAT was operating in "*standard*" conditions,
  - ✓ the source was within the field of view (FoV) of the instruments (and possibly well separated from the Sun)
  - ✓ the Earth Limb was outside the FoV.

- **The selection of GTIs is done working on the FT2 file with the software "gtmktime". Type the command "gtmktime" on your laptop with the following inputs:**

These are our GTI conditions

```
>> gtmktime
>> Spacecraft data file[] Data/SC00.fits
>> Filter expression[] (DATA_QUAL==1) &&
     ABS(ROCK_ANGLE)<90 && (LAT_CONFIG==1) &&
     (angsep(RA_ZENITH,DEC_ZENITH,343.49,16.15)+1<105) &&
     (angsep(343.49,16.15,RA_SUN,DEC_SUN)>6) &&
     (angsep(343.49,16.15,RA_SCZ,DEC_SCZ)<180)
>> Apply ROI-based zenith angle cut[yes] yes
>> Event data file[] temp_gtselect.fits
>> Output event file name[] temp_gtmktime.fits
```

Light Curve with gtbin

- Now you are ready to build the light curve of the AGN 3C454.3 in your time interval.
- First of all you have to divide your dataset in 1-day bins (86400 s).
  - ✓ This will be done with the software "**gtbin**". Type "**gtbin**" with the following inputs:

```
>> gtbin
>> This is gtbin version ScienceTools-v10r0p5-fssc-20150518
>> Type of output file (CCUBE|CMAP|LC|PHA1|PHA2|HEALPIX) [CMAP] LC
>> Event data file name[] temp_gtmktime.fits
>> Output file name[] lightcurve.fits
>> Spacecraft data file name[] Data/SC00.fits
>> Algorithm for defining time bins (FILE|LIN|SNR) [LIN] LIN
>> Start value for first time bin in MET[] 435628803
>> Stop value for last time bin in MET[] 436752003
>> Width of linearly uniform time bins in seconds[86400] 86400
```

We want to create a Light Curve (LC)

PUT here your start and stop time in MET

1 day per bin

gtexposure

- **The next step is to calculate the exposure of the source in the selected time interval.**
- **This calculation is done with the "gtexposure" software.**
  - ✓ This tool computes the exposure (cm^2 s) associated with each time bin, allowing for a light curve in photons/s to be computed.
  - ✓ The calculation will take a few minutes. Type "**gtexposure**" on your laptop with the following inputs:

```
>> gtexposure
>> Light curve file[] lightcurve.fits
>> Spacecraft file[] Data/SC00.fits
>> Response functions[P8R2_SOURCE_V6] P8R2_SOURCE_V6
>> Source model XML file[none] none
>> Photon index for spectral weighting[-2.1] -2.1
```

- **The software "gtexposure" adds a column with the exposure to the file lightcurve.fits.**

fcalc

- Now you are ready to evaluate the rate as the ratio between the counts and the exposure.

$$\Phi = \frac{counts(E,t)}{exposure(E,t)}$$

- This is done by the software “fcalc”.

```
>> fcalc lightcurve.fits lightcurve_flux.fits RATE 'counts/exposure'
```

- The software “fcalc” adds a column to an input fits file (lightcurve.fits) and produces a new fits file (lightcurve_flux.fits).
    - The new column is “RATE” is evaluated as the ratio between the columns “COUNTS” and “EXPOSURE”.
- We need also to evaluate the error on the flux

$$\sigma_{\Phi} = \frac{\sigma_{counts}}{exposure}$$

- Using “fcalc” :

```
>> fcalc lightcurve_flux.fits lightcurve_flux_error.fits RATE_ERROR 'error/exposure'
```

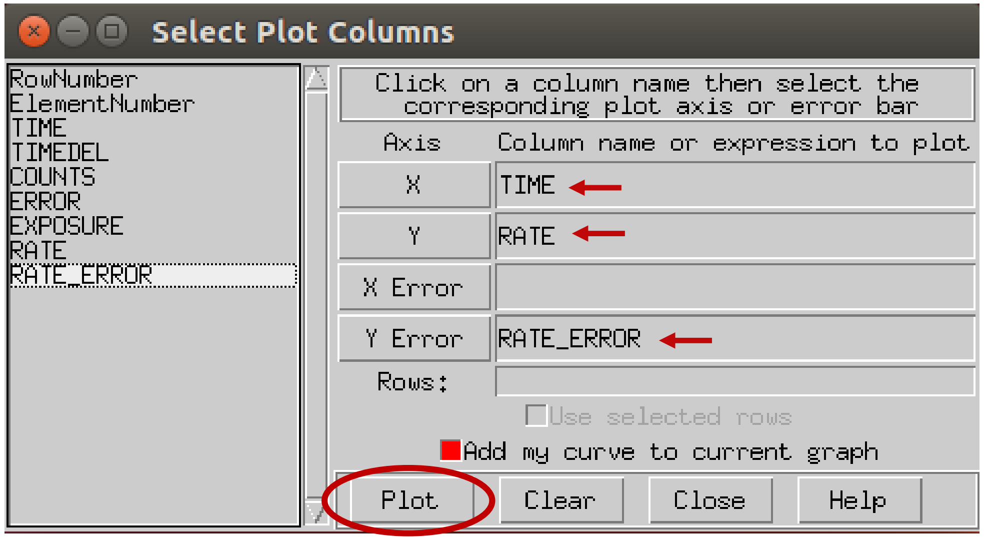
Your light curve

- Now you are ready to take a look at your light curve using the "fv" software.

```
>> fv lightcurve_flux_error.fits
```

- To produce the plot you have to select the variables (*TIME, RATE, RATE_ERROR*) in the left column and click on "X", "Y" and "Y ERROR" respectively.

- After you have made your selection, click on the "PLOT" button.


Select Plot Columns

```
RowNumber
ElementNumber
TIME
TIMEDEL
COUNTS
ERROR
EXPOSURE
RATE
RATE_ERROR
```

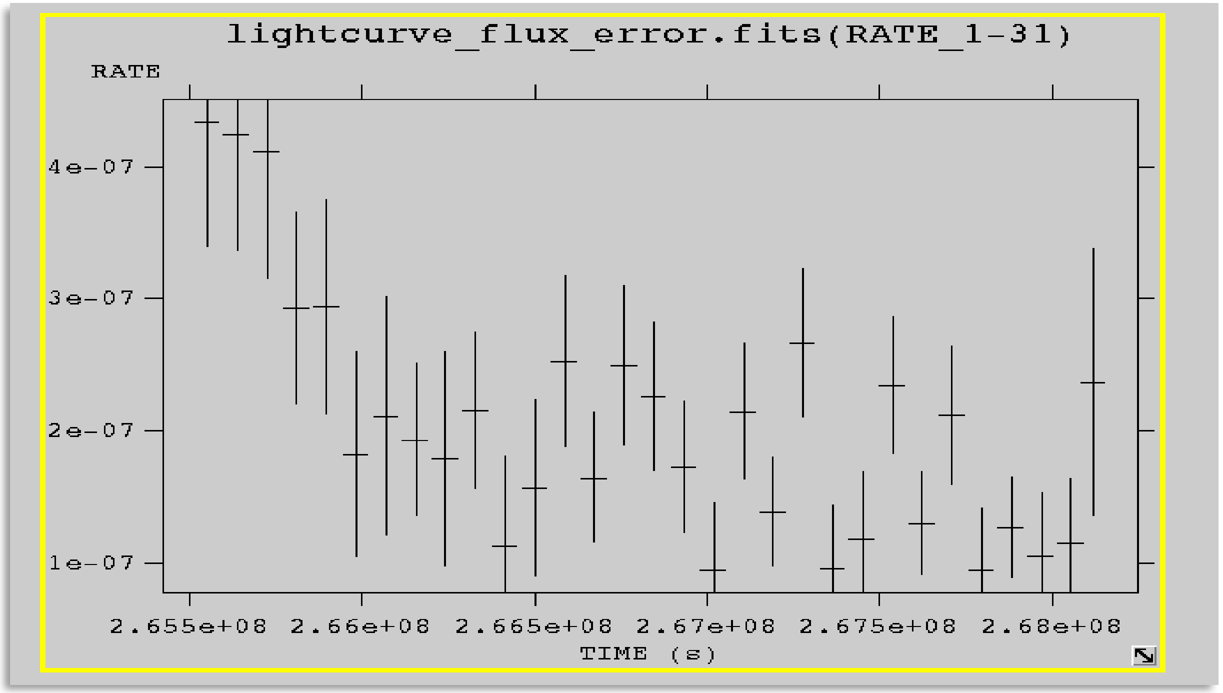
Click on a column name then select the corresponding plot axis or error bar

Axis	Column name or expression to plot
X	TIME
Y	RATE
X Error	
Y Error	RATE_ERROR
Rows:	

☐ Use selected rows

🟥 Add my curve to current graph

Merge all results



Now we can collect and merge all your results

- Copy your file on the pen drive and we will merge your spectrum in the spectra of the full time range using fmerge with the command:

```
>> fmerge "light_curve_1.fits light_curve_2.fits ..." light_curve_all.fits
```

- ✓ The column to be merged should be RATE RATE_ERROR TIME
- ✓ You can see the result using the fv software

gtbin

While *fv* can be used to make quick plots when exploring the data, it doesn't automatically do all the things you would like when making data files for analysis.

For this, you will need to use the *Fermi*-specific gtbin tool for analysis.

You can use gtbin to bin photon data into the following representations:

- Images (maps)
- Light curves
- Energy spectra
- …

➢ Now we can build a *count map* of the specific region to be analyzed using the "gtbin" command. You just have to type the command gtbin and answer the questions as indicated in the following slides.

Remember that for this program, as well as for the other programs which will be used in the following, the text within square brackets [ ] corresponds to the settings used when the program was ran for the last time, which is assumed as a default.

Be careful when answering the questions!

Count map with gtbin

Type **gtbin**

We want to create a Count MAP (CMAP)

```
>> gtbin
>> This is gtbin version ScienceTools-v10r0p5-fssc-20150518
>> Type of output file (CCUBE|CMAP|LC|PHA1|PHA2|HEALPIX) [] CMAP
>> Event data file name[] Data/L180110050120A078F53F72_PHOTONS.fits
>> Output file name[lc_3C279.fits] fulltime_cmap.fits
>> Spacecraft data file name[] Data/SC00.fits
>> Size of the X axis in pixels[100] 100
>> Size of the Y axis in pixels[100] 100
>> Image scale (in degrees/pixel)[0.1]  0.1
>> Coordinate system (CEL - celestial, GAL -galactic) (CEL|GAL) [CEL] CEL
>> First coordinate of image center in degrees (RA or galactic l)[128.5] 343.49
>> Second coordinate of image center in degrees (DEC or galactic b)[-45.8] 16.15
>> Rotation angle of image axis, in degrees[0] 0
>> Projection method e.g. AIT|ARC|CAR|GLS|MER|NCP|SIN|STG|TAN:[AIT] AIT
```

Coordinates of our source

In this way the "gtbin" program will produce a count map of the region to be analyzed, which will be saved in the file fulltime_cmap.fits.

This map can be seen with the "ds9" program, just typing the command:

```
>> ds9 fulltime_cmap.fits
```

Count Map in your time interval

- **Now you are able to use all Fermi tools do you need to do the Aperture photometry analysis like a real scientist.**
- **To recap we can try produce a count map in your time interval selecting the photons within 5° from our reference source.**
  - ✓ This can be done again with the "gtselect" software:

```
>> gtselect
>> Input FT1 file[] Data/L180110050120A078F53F72_PHOTONS.fits
>> Output FT1 file[temp_gtselect.fits] temp_gtselect_roi5.fits
>> RA for new search center (degrees) (0:360) [343.49] 343.49
>> Dec for new search center (degrees) (-90:90) [16.15] 16.15
>> radius of new search region (degrees) (0:180) [1] 5
>> start time (MET in s) (0:) [435628803] 435628803
>> end time (MET in s) (0:) [436752003] 436752003
>> lower energy limit (MeV) (0:) [100] 100
>> upper energy limit (MeV) (0:) [200000] 200000
>> maximum zenith angle value (degrees) (0:180) [90] 90
Done.
```

note here that the value is not 1 as before, but 5!

PUT here your start and stop time in MET

Count Map in your time intervall: gtbin

Now you can create a new map with the "gtbin" software:

We want to create a Count MAP (CMAP)

```
>> gtbin
>> This is gtbin version ScienceTools-v10r0p5-fssc-20150518
>> Type of output file (CCUBE|CMAP|LC|PHA1|PHA2|HEALPIX) [LC] CMAP
>> Event data file name[temp_gtmktime.fits] temp_gtselect_roi5.fits
>> Output file name[lightcurve.fits] cmap_laptop1.fits
>> Spacecraft data file name[] Data/SC00.fits
>> Size of the X axis in pixels[100] 100
>> Size of the Y axis in pixels[100] 100
>> Image scale (in degrees/pixel)[0.1] 0.1
>> Coordinate system (CEL - celestial, GAL -galactic) (CEL|GAL) [CEL] CEL
>> First coordinate of image center in degrees (RA or galactic l)[343.49] 343.49
>> Second coordinate of image center in degrees (DEC or galactic b)[16.15] 16.15
>> Rotation angle of image axis, in degrees[0] 0
>> Projection method e.g. AIT|ARC|CAR|GLS|MER|NCP|SIN|STG|TAN:[AIT] AIT
```

and you can see your map with ds9:

```
>> ds9 cmap_laptop1.fits
```