

METHODS IN COMPUTATIONAL PHYSICS A: LABORATORY  
WINTER SEMESTER 2022/2023

---

## Likelihood analysis of $\gamma$ -ray flares in 3C 279

---

### ASSIGNMENT DESCRIPTION

All information required to perform this practical lab is provided hereafter. It can be completed with two sessions of approximately 3 hours, which shall be prepared autonomously. In summary, such a practical assignment will be organised as follows:

- Before the first session, the students should prepare themselves by closely reading this document.
- The first session will start with a short, preparatory test discussing the basics of the assignment. After this, the students will start working on Tasks 1 and 2.
- The second session will resume the experiment, focusing on Tasks 3 and 4.
- Each student should keep track of the procedures followed on a lab notebook (cuts made, codes, figures, and others). Such notebook can be formatted at the student's discretion (e.g. screen-shots on a text document would suffice), but should be provided at the end of every session.
- A detailed report describing the methods used and discussing the results must be provided within 2 weeks after the second session.

The final grade will be obtained from the **preparatory test** (20%), the **work performed** during the sessions (40%, including the **lab notebook**), and the **final report** (40%).

*For any doubt regarding this assignment, please feel free to contact the assignment's supervisor:  
[guillem.marti-devesa@uibk.ac.at](mailto:guillem.marti-devesa@uibk.ac.at)*

# 1 GAMMA-RAY ASTRONOMY FROM SPACE

Since magnetic fields deviate charged particles, neutral messengers are necessary to study cosmic rays – and therefore investigate particle acceleration in astrophysical sources. Subsequently,  $\gamma$ -rays (photons with energies larger than 1 MeV) can be used to indirectly track both Galactic and Extragalactic acceleration sites. While ground-based observatories can explore the very-high-energy regime (above 100 GeV) with the detection of extensive air showers, at lower energies it is necessary to go over the atmosphere to perform observations [1].

## 1.1 *The Fermi Large Area Telescope*

The Large Area Telescope (LAT) is the main instrument on board the *Fermi* satellite [2]. Its principal scientific objective is to conduct a long  $\gamma$ -ray survey of celestial sources in the so-called high-energy regime (0.1 – 100 GeV), although it can detect photons as energetic as 500 GeV. The LAT is designed in such a way that its field of view is approximately 2.4 sr – i.e.  $\frac{1}{5}$  of the whole sky! Therefore *Fermi* scans constantly the sky and, just in a few orbits around Earth ( $\sim 3$  hours), it obtains a full map of the sky at high-energies. But how does the LAT *actually* work?

Such instrument is basically a pair conversion detector in orbit (see Figure 1.1). The *Fermi*-LAT has 16 modules, each one of them consisting of 18 tungsten conversion foils interspersed

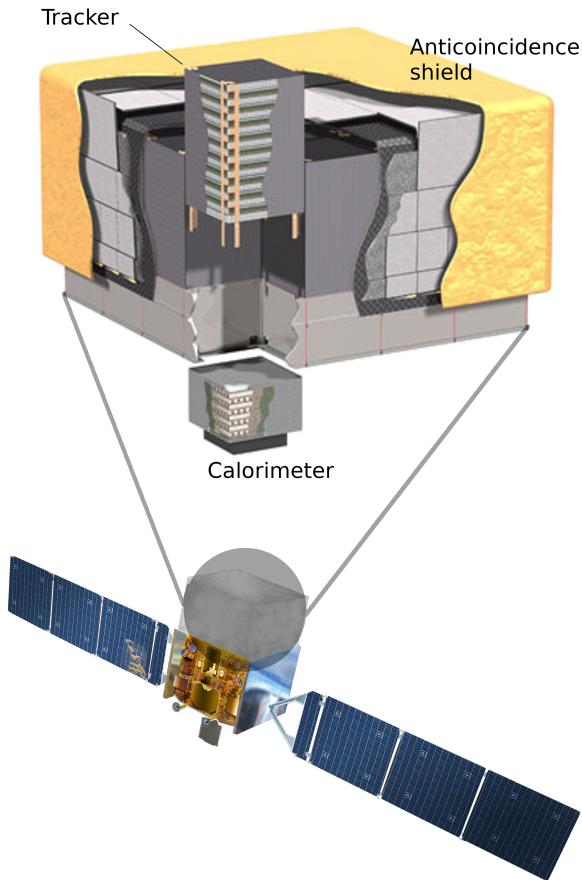


Figure 1.1: The LAT is the main instrument of the *Fermi* satellite. This scheme shows the three principal components of the detector: an anticoincidence shield covering 16 tracker modules, each one of them with its own calorimeter. Image credit: Adapted from NASA / Goddard Space Flight Center.

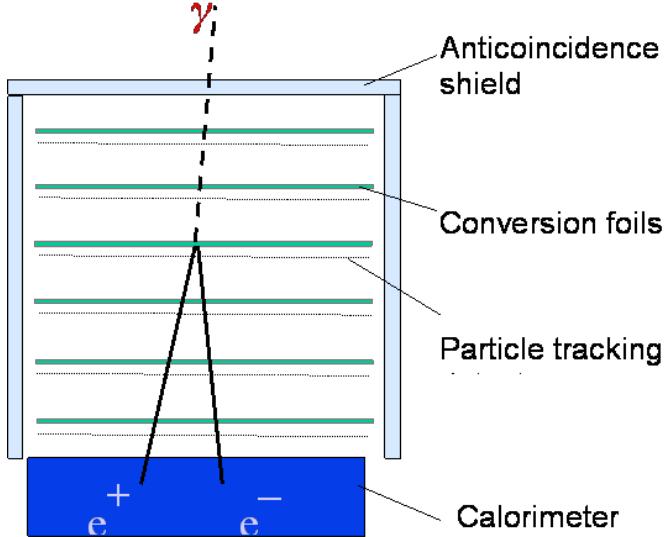


Figure 1.2: Schematic representation of the detection of a  $\gamma$ -ray by the LAT. Image credit: NASA / Fermi Science Support Center.

with 16 silicon-strip tracking detectors. When a  $\gamma$ -ray enters one of those modules, it interacts with one of the tungsten layers and forms an electron-positron pair (Figure 1.2). Such pair will scatter through the module, being their position tracked with the silicon layers. This allows the reconstruction of the direction of the original photon. When the electron and the positron reach the bottom of the module, they enter a CsI(Ti) calorimeter and hence their energy is measured.

Indeed such process allows the detection of  $\gamma$ -rays, but also cosmic rays! Those outnumber photons by a factor  $10^2 - 10^5$  and may produce particle showers within the instrument, thus constituting the main background source for the *Fermi*-LAT [3]. Therefore a detection of an event is not equivalent to the detection of a photon. In order to filter such background, one shall use the anticoincidence shield: 89 scintillator tiles covering the 16 tracker modules which can identify charged particles going through them. Events associated with charged particles are flagged by the on-board software as potential background.

## 1.2 LAT Instrument Response Functions (IRFs)

Before we described how does the *Fermi*-LAT detect individual events. Now we will briefly summarize the performance of the instrument as a function of the energy  $E$  and incidence angle  $\theta$ , among other parameters [4].

The IRFs relates the detected events to the real incoming photon flux, and will highly depend on the hardware design, the reconstruction algorithms and background rejections used. Consequently, depending on the goal of an analysis, the cuts might be slightly different and we can have several IRF sets (see Section 2). Nevertheless, they can be factorised in a few terms: the point-spread function (PSF), the effective area and the energy dispersion.

### 1.2.1 Effective Area

The first piece of the IRFs is the effective collecting area  $A_{eff}$ . This depends on the geometrical cross-section of the detector and its efficiency on the identification of the incident  $\gamma$ -rays. It is derived from Monte-Carlo simulations of the instrument:  $N_{MC}$   $\gamma$ -rays are generated uniformly in  $\log(E)$  and solid angle  $\Omega$ , being  $n_{i,j,k}$  the number of photons correctly identified in a

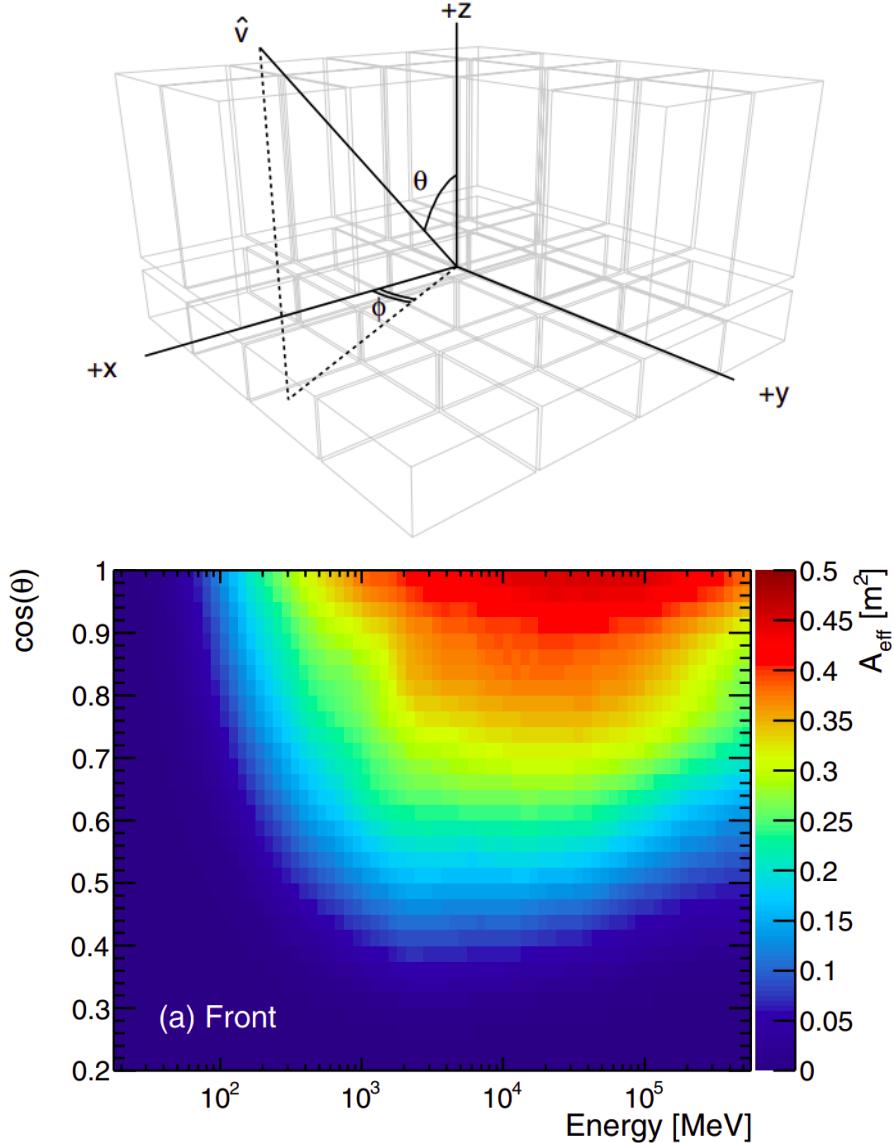


Figure 1.3: *Top:* Parametrisation of the simulated  $\gamma$ -rays in the Monte-Carlo simulations performed to characterise the IRFs. *Bottom:* Effective area for a FRONT selection and the old P7SOURCE\_V6 IRF averaged over  $\phi$ . Image credit: [4]

bin centred at  $E = E_i$ ,  $\theta = \theta_j$  and  $\phi = \phi_k$ :

$$A_{eff}(E_i, \theta_j, \phi_k) = [6m^2] \left( \frac{n_{i,j,k}}{N_{MC}} \right) \left( \frac{2\pi}{\Delta\Omega_{j,k}} \right) \times \left( \frac{\log E_{max} - \log E_{min}}{\log E_{max,i} - \log E_{min,i}} \right) \quad (1.1)$$

Effective area maps can therefore be generated depending on the selection and background cuts (see the example in Figure 1.3). Note that the exposure of any given direction in the sky  $\hat{v}$  at an energy  $E$  will be simply the integral over the time range of interest of the effective area:

$$\mathcal{E}(E, \hat{v}) = \int A_{eff}(E, \hat{v}(t)) dt \quad (1.2)$$

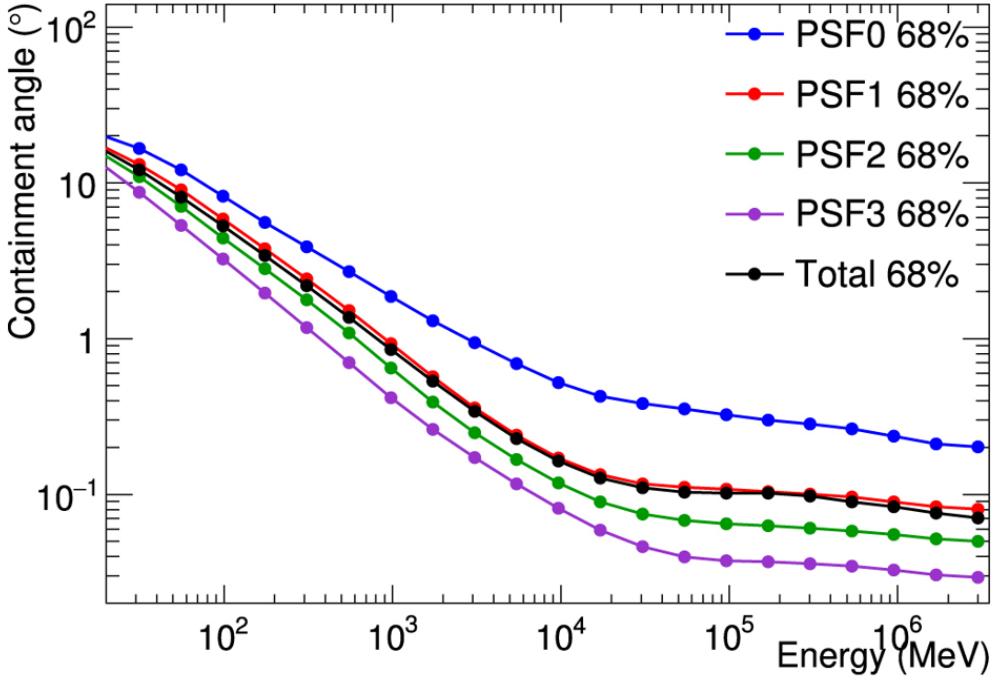


Figure 1.4: PSF from *Fermi* -LAT averaged over off-axis angle The different colors represent various data selections (see Section 2). Image credit: [5].

### 1.2.2 The PSF

A PSF is the probability distribution for the reconstructed direction of an event from a point-like  $\gamma$ -ray source. The LAT's PSF is limited by the scattering across the tracking modules and the bremsstrahlung production of the electron-positron pair, thus it worsens towards low energies – at 68% containment, the PSF is around  $5^\circ$  at 100 MeV and  $0.1^\circ$  at 10 GeV (Figure 1.4).

Analytically, a normalised PSF can be described as the composition of two different King-like profiles:

$$P(x, \sigma, \gamma) = f_{core} \cdot K(x, \sigma_{core}, \gamma_{core}) + (1 - f_{core}) \cdot K(x, \sigma_{tail}, \gamma_{tail}) \quad (1.3)$$

where

$$K(x, \sigma, \gamma) = \frac{1}{2\pi\sigma^2} \left(1 - \frac{1}{\gamma}\right) \left[1 + \frac{1}{2\gamma} \cdot \left(\frac{x}{\sigma}\right)^2\right]^{-\gamma} \quad (1.4)$$

Which is derived from the original King's profile function:

$$F_{king}(x, \sigma, \gamma) \propto \left[1 + \left(\frac{x}{\sigma}\right)^2\right]^{-\gamma} \quad (1.5)$$

where  $\sigma$  and  $\gamma$  are parameters which will depend on the IRF set and  $x$  is the rescaled angular deviation derived from  $\delta\hat{v}$ . Such scaling can be performed in energy dividing  $\delta\hat{v}$  by a factor:

$$S(E) = \sqrt{\left[c_0 \cdot \left(\frac{E}{100 \text{ MeV}}\right)^{-\beta}\right]^2 + c_1^2} \quad (1.6)$$

where  $c_0$ ,  $c_1$  and  $\beta$  are again parameters dependent on the set of IRFs used.

### 1.2.3 Energy dispersion

Finally, the energy dispersion will describe the differences between the measured energy of an event and its real energy. Such discrepancies are relatively small above 300 MeV (less than 5%), but are very relevant at lower energies (specially below 100 MeV). This is a similar phenomenon to what occurs with the PSF. A rescaled energy  $x$  can be defined as:

$$x = \frac{\delta E}{E \cdot S_D(E, \theta)} \quad (1.7)$$

where

$$S_D(E, \theta) = c_0(E)^2 + c_1(\cos\theta)^2 + c_2 \log E + c_3 \cos\theta + c_4 \log E \cos\theta + c_5 \quad (1.8)$$

The parameters  $c_{0-5}$  depend on the set of IRFs used and are again obtained from Monte-Carlo simulations. In analogy to what happens with the PSF, such scaled energy is a useful parameter for the description of the energy dispersion function. This function is obtained combining two asymmetric exponential power functions as:

$$D(x) = f \cdot g(x, \sigma_1, k_1, b_1, p_1) + (1 - f) \cdot g(x, \sigma_2, k_2, b_2, p_2) \quad (1.9)$$

where

$$g(x, \sigma, k, b, p) = \frac{p}{\sigma \Gamma\left(\frac{1}{p}\right)} \frac{k}{1+k^2} \begin{cases} \exp\left(-\frac{k^p}{\sigma^p}|x-b|^p\right), & \text{if } x-b \geq 0 \\ \exp\left(-\frac{1}{k^p \sigma^p}|x-b|^p\right), & \text{otherwise} \end{cases} \quad (1.10)$$

and  $\sigma$ ,  $k$ ,  $b$  and  $p$  are the deviation, skewness, bias and exponential index of those asymmetric exponential power functions, respectively.

## 1.3 Data Analysis Methods

Once the instrument is properly understood, one can try to obtain physical quantities from the events detected. Mainly, we will be interested in two things: the flux of  $\gamma$ -rays emitted by non-thermal sources, and how many of those  $\gamma$ -rays are detected as a function of the energy  $E$ . In order to try to obtain them you will use two different methods: aperture photometry and maximum-likelihood analysis.

### 1.3.1 Aperture photometry

Aperture photometry is the standard method to obtain a flux across astronomy. It is used by infra-red, optical and X-ray facilities and basically consists in defining a certain area of the sky, counting all photons within that region, correct for background from a different region of the sky and then divide by the exposure (see Figure 1.5).

In the case of  $\gamma$ -ray astronomy, data analyses are more accurately done using a likelihood analysis. Nevertheless, aperture photometry provides a useful, less computationally demanding method to derive fluxes – particularly for light curves. Unfortunately, no background subtraction is possible for *Fermi*-LAT data. The practical details of this method are described in Section 1.5.

### 1.3.2 Maximum-likelihood analysis

However, the most accurate and sensitive method to study  $\gamma$ -ray sources is a likelihood analysis, as backgrounds can be modelled out and source models can be applied. While aperture photometry (or even the qualitative exploration of the data) may suggest the presence of

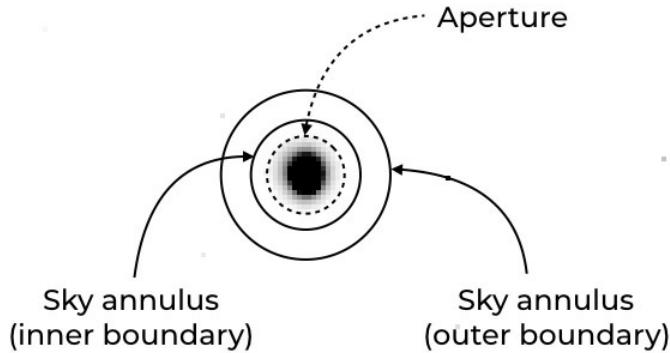


Figure 1.5: Aperture photometry from an optical image. The background region could be either circular or annular. Image credit: The Nieves Observatory.

sources, the inclusion of the IRFs is necessary in order to properly assess all proper corrections to the data [6].

Like for aperture photometry, a region of the sky has to be defined and use all photons within that region. We call it the region of interest (ROI), which should be significantly larger (double its size or more) than the PSF of the instrument at the energies that we want to study. To analyse LAT data (which basically consist of a list of events), we construct a likelihood function  $\mathcal{L}$  to estimate the probability that a certain model of the  $\gamma$ -ray sky truly represents the LAT data, and then maximise this likelihood to find the best fit model parameters. These parameters include the presence or not of sources, their positions and even their spectrum. There are two different ways of describing this  $\mathcal{L}$  function (and hence to perform the analysis): in an unbinned or a binned way. Here, we will discuss a binned likelihood function, since an unbinned method is computationally very expensive. The ROI is therefore divided in  $k$  bins in sky coordinates and in energy. Note that this has transformed our list of events in a data cuboid.

The likelihood function  $\mathcal{L}$  is then defined as

$$\mathcal{L} = \prod_k \frac{m_k^{d_k} \cdot e^{-m_k}}{d_k!} \quad (1.11)$$

$$\mathcal{L} = \sum_k d_k \log m_k - \sum_k m_k \quad (1.12)$$

where  $d_k$  and  $m_k$  are the number of detected and predicted counts in bin  $k$ , respectively. This number of predicted counts is obtained from convolving the flux expected from our prior knowledge of the  $\gamma$ -ray sources in the ROI and the IRFs.

Maximizing the likelihood function  $\mathcal{L}$ , one fits the different parameters of our model to the data. And how can we evaluate if the detection of a source is significant or not? A certain model included in  $\mathcal{L}_1$  is compared with the null-hypothesis  $\mathcal{L}_0$  (basically this means the underlying model except the parameter that we want to evaluate) via the test statistic  $TS$  defined as

$$TS = -2 \log \frac{\mathcal{L}_0}{\mathcal{L}_1} \quad (1.13)$$

Such  $TS$  follows a  $\chi^2$  distribution, whose probability density function can be expressed as

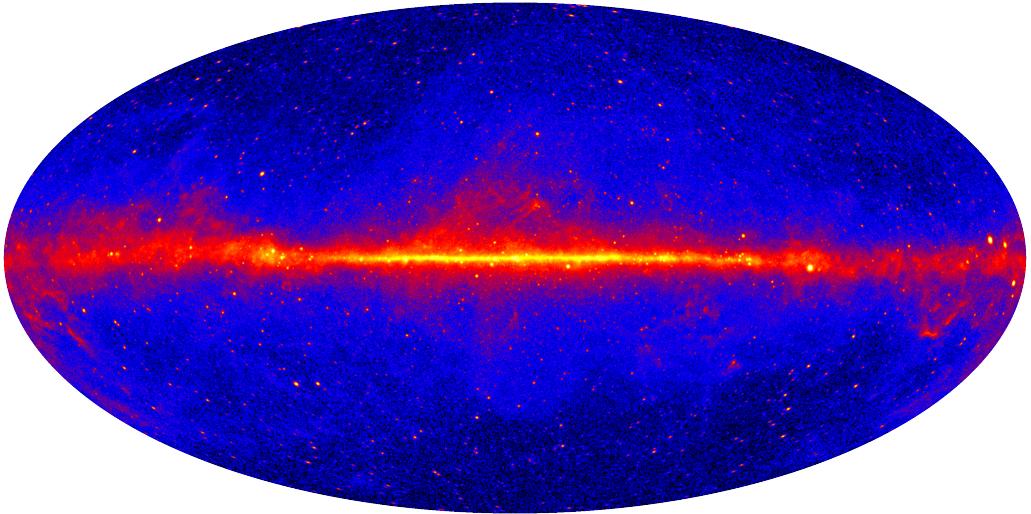


Figure 1.6: High-energy regime all-sky  $\gamma$ -ray emission as seen by the LAT. Besides the presence of point-like sources, the galactic diffuse emission can be seen by eye, as well the extragalactic isotropic diffuse emission. Image credit: *Fermi*-LAT Collaboration.

$$f(x, k) = \begin{cases} \frac{x^{\frac{k}{2}-1} \exp^{-\frac{x}{2}}}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})}, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1.14)$$

where  $k$  is the number of degrees of freedom. That means that for 1 degree of freedom,  $\sqrt{TS}$  will be the significance of having the hypothesis  $\mathcal{L}_1$  compared to the null-hypothesis (i.e. a detection of a source with  $TS = 25$  is equivalent to a detection at  $5\sigma$ ).

### 1.3.3 Diffuse background models

As shown in the previous section, the construction of a successful likelihood function  $\mathcal{L}$  demands a profound knowledge of the sources in the  $\gamma$ -ray sky. Independently of the coordinates that one decides to study, diffuse emission deserves special attention.

Leaving aside small, extended sources where diffusion of accelerated particles might occur, there are two main contributors to the diffuse  $\gamma$ -ray flux which will be always considered as background. Those are the diffuse galactic and the isotropic diffuse component<sup>1</sup>. The later one can be understood as the collective emission from unresolved extragalactic sources, while the former one arises from the interaction of cosmic-rays with the gas and dust in the Galactic plane (Figure 1.6).

Hence when a maximum-likelihood analysis is done, a map of the galactic diffuse emission is employed, and an isotropic background is considered which (of course) depends on the IRFs.

### 1.3.4 Source spectral models

Besides the diffuse emission, the ROI will contain many other sources, most of them point-like. As a prior for the analysis, one might consider using a catalogue of known sources, like the 4FGL [5]. But, how are they included in the model?

---

<sup>1</sup> <https://fermi.gsfc.nasa.gov/ssc/data/access/lat/BackgroundModels.html>

Point-like sources are convolved with the PSF to see how they will be seen in the sky, centred at the position of  $\gamma$ -ray excesses detected by the LAT. However, one needs to consider their spectral energy distributions (SEDs) so we can compute the number of expected counts in each energy bin. Here you will find a list of the main spectral models used in maximum-likelihood analyses:

- **Power law:** Simplest source model, used for most of the sources. It is seen as a straight line in log-scale plots.

$$\frac{dN}{dE} = N_0 \left( \frac{E}{E_0} \right)^{-\Gamma} \quad (1.15)$$

where  $N_0$  is the flux normalization,  $E_0$  the energy scale factor and  $\Gamma$  the spectral index.

- **Log-parabola:** Simplest curvature model, used to model simple blazars. It is seen as a parabola in log-scale plots.

$$\frac{dN}{dE} = N_0 \left( \frac{E}{E_0} \right)^{-\left(\alpha + \beta \log\left(\frac{E}{E_b}\right)\right)} \quad (1.16)$$

where  $N_0$  is the flux normalization,  $E_b$  the energy break and  $\alpha$  and  $\beta$  are the parameters controlling the curvature.

- **Broken power law:** Used to represent sources with two different components.

$$\frac{dN}{dE} = N_0 \times \begin{cases} \left(\frac{E}{E_b}\right)^{-\Gamma_1}, & \text{if } E < E_b \\ \left(\frac{E}{E_b}\right)^{-\Gamma_2}, & \text{otherwise} \end{cases} \quad (1.17)$$

where  $N_0$  is the flux normalization,  $E_b$  the energy break and  $\Gamma_1$  and  $\Gamma_2$  are the spectral indexes of the two components.

- **Power law with an exponential cut-off:** This model is used to characterise a sudden drop in the spectrum caused by the underling particles reaching a maximum energy.

$$\frac{dN}{dE} = N_0 \left( \frac{E}{E_0} \right)^{-\Gamma} \exp\left(-\frac{E}{E_c}\right)^\beta \quad (1.18)$$

where  $N_0$  is the flux normalization,  $E_0$  the energy scale factor,  $E_c$  the energy cut-off,  $\Gamma$  the spectral index and  $\beta$  the curvature parameter.

## 2 FERMI DATA PRODUCTS

Data from the *Fermi*-LAT consist of a list of events preprocessed in a fits file [7], which contains their directions, energies, arrival times and other information (Figure 2.1, top) – this list might be divided in several files, though. Furthermore, for an analysis a spacecraft file is necessary, which contains the position of the satellite in orbit during the time the data was acquired. These pre-processed files can be obtained from the *Fermi Science Support Center*<sup>2</sup> (FSSC), where one just needs to specify the coordinates of an object, how large will be the ROI, the observing times wanted and the energy range required (Figure 2.1, bottom).

---

<sup>2</sup><https://fermi.gsfc.nasa.gov/cgi-bin/ssc/LAT/LATDataQuery.cgi>

Table Browser for 1: L201009101439DF176D7534_PH00.fits												
	ENERGY	RA	DEC	L	B	THETA	PHI	ZENITH_ANGLE	EARTH_AZIM...	TIME	EVENT_ID	
1	590.543	67.8803	0.133458	195.106	-30.6071	41.8888	151.638	48.5547	249.371	5.6006689E8	4339307	▲
2	547.365	55.0982	0.051815	186.211	-41.1849	63.1824	233.89	35.5304	75.7802	5.600048E8	14669182	■
3	281.488	49.8361	1.81791	179.804	-44.1207	60.8929	226.955	36.1839	247.4	5.600230E8	5917486	
4	1356.01	50.304	2.34352	179.693	-43.4217	72.6127	222.913	59.2415	257.839	5.600234E8	6651547	
5	275.864	48.398	1.39581	178.886	-45.47	61.0633	223.771	36.8543	71.1625	5.600332E8	1303514	
6	707.401	55.4668	1.90553	184.585	-39.7557	57.985	241.48	22.2419	77.7234	5.600336E8	1975717	
7	1096.84	51.1198	1.12099	181.703	-43.6118	54.0678	236.884	4.70485	185.805	5.600396E8	1861393	
8	147.447	53.985	1.98829	183.283	-40.8573	61.2969	231.41	31.3661	241.944	5.600572E8	1356697	
9	446.638	47.3231	0.644879	178.635	-46.7695	58.9242	226.098	28.3126	241.413	5.600628E8	791287	
10	1138.76	46.3607	0.783348	177.522	-47.3778	60.2124	221.421	32.2373	72.9398	5.600790E8	12460719	
11	143.969	49.9287	0.608984	181.163	-44.8491	59.8733	228.623	26.1891	236.869	5.600856E8	1370746	
12	160.351	52.1412	0.338728	183.432	-43.3286	58.5965	232.986	16.6343	222.15	5.600969E8	924432	
13	120.858	46.1864	1.46025	176.625	-47.0347	62.3494	222.167	36.6323	245.832	5.601029E8	7933337	
14	318.872	47.3971	1.16443	178.154	-46.3619	54.9792	230.196	6.07529	191.658	5.601138E8	3467407	
15	221.476	54.8579	2.66803	183.317	-39.7518	59.1746	234.148	20.0644	229.585	5.601199E8	3164157	
16	1558.23	49.2273	0.57959	180.541	-45.3973	63.0836	225.633	32.9458	240.163	5.601200E8	3349248	
17	1966.91	47.0882	1.97836	176.994	-46.0265	67.1604	220.686	45.0336	249.571	5.601373E8	2084189	
18	959.609	54.7743	1.27441	184.671	-40.6902	60.6603	235.923	18.32	94.7717	5.601536E8	802969	
19	140.281	50.9476	1.36329	181.294	-43.585	57.8285	232.84	10.69	109.436	5.601537E8	957752	
20	255.878	52.0354	0.9030232	182.713	-43.0341	64.046	227.959	29.4838	234.174	5.601771E8	1664918	
21	153.724	47.8524	0.985487	178.792	-46.1493	65.8692	222.295	37.7683	242.65	5.601772E8	1883966	
22	311.101	51.7548	1.19268	182.192	-43.081	61.8747	229.357	22.6929	88.9556	5.601935E8	4215559	
23	168.937	49.1865	1.07226	179.978	-45.1016	58.8948	228.897	12.8642	105.234	5.601936E8	4434175	
24	224.63	50.992	1.88906	180.706	-43.2751	58.6672	231.486	11.7244	109.997	5.601936E8	4523238	
25	972.693	48.8818	1.44452	179.298	-45.0796	58.6612	228.1	13.4486	211.712	5.601939E8	4981288	
26	1055.73	49.1971	2.50251	178.5	-44.133	59.9071	226.605	22.0152	231.336	5.601941E8	5269865	
27	1463.76	52.6376	1.99731	182.128	-41.8877	62.2083	229.426	17.1206	209.683	5.602683E8	1293550	
28	2358.29	52.0573	1.58449	182.049	-42.5973	68.7308	220.734	36.7678	84.104	5.602789E8	3716053	
29	182.261	52.2147	1.60873	182.161	-42.4613	70.0069	224.346	38.8575	240.548	5.602858E8	4428965	▼

Total: 1,194,579    Visible: 1,194,579    Selected: 0

## LAT Photon, Event, and Spacecraft Data Query

Object name or coordinates:	<input type="text"/>
Coordinate system:	<input type="text" value="J2000"/>
Search radius (degrees):	<input type="text"/>
Observation dates:	<input type="text"/>
Time system:	<input type="text" value="MET"/>
Energy range (MeV):	<input type="text"/>
LAT data type:	<input type="text" value="Photon"/>
Spacecraft data:	<input checked="" type="checkbox"/>
<input type="button" value="Start Search"/> <input type="button" value="Reset"/>	

Figure 2.1: Top: Screenshot of a processed photon file from the FSSC from an arbitrary ROI (in this case, the region around the GRB 190114C). It contains more than  $10^6$  events! Bottom: Screenshot of the LAT Data Query system. Always remember to download the spacecraft file too. Image credit: *Fermi*-LAT Collaboration.

At this point, it is important to clarify that *Fermi*-LAT products usually employ a particular unit of time: the Mission Elapsed Time (MET). This is just the number of seconds since midnight (UTC) on January 1st 2001. For practical purposes, it might be useful to use the conversion utility xTime<sup>3</sup>.

As part of the event reconstruction process, LAT data classifies the events in different classes depending on their photon probability and quality reconstruction<sup>4</sup>. Each one of those classes is designed for different scientific purposes, and subdivided into event types based on individual event topologies (e.g. where within the instrument the pair conversion occurs). Therefore, any

<sup>3</sup><https://heasarc.gsfc.nasa.gov/cgi-bin/Tools/xTime/xTime.pl>

<sup>4</sup>[https://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone\\_Data/LAT\\_DP.html](https://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone_Data/LAT_DP.html)

combination of event class and event type will require a set of IRFs obtained for such selection.

However, one should note that this classification is not static: over the time, the calibration improves and the data is re-processed. Hence these classifications vary over the years. The latest data release is the Pass 8 (P8R3), motivated by the need to mitigate the impact of *ghost* events – instrumental pile-up away from the  $\gamma$ -ray shower. Here you will find the current event classes (Table 2.1) and types (Table 2.2). In Table 2.3, you will find a brief guide to decide when to use an event class or another. Note that in a single analysis, you can only use an event class and type.

Last but not least, atmospheric  $\gamma$ -rays from the Earth can be a significant source of background – also referred to as Earth Limb contamination<sup>5</sup>. The best way to account for them is having an additional cut on the maximum zenith angle of the incoming event. Given the topological parametrization underlying the event type classification, each one will have a different maximum zenith angle allowed to reject such background. Besides, its energy dependence has to be considered. A summary of the recommended cuts can be found on Table 2.4.

### 3 DATA ANALYSIS SOFTWARE

Analysis of *Fermi*-LAT data is typically done using the *Fermitoools*<sup>6</sup>, or via its python interface *fermipy*. While employing python scripts can be convenient, the direct usage of *Fermitoools* can be advantageous under some circumstances, since they are more versatile and compatible with the HEADAS FTOOLS<sup>7</sup> (check Figure 3.1 for a hierarchical description of the different tools included within *Fermitoools*). In this section, you will learn how to perform a standard analysis with both methods: command-line tools and python scripting.

#### 3.1 Binned analysis with *Fermitoools*

Here we will follow a simplified version of the tutorial available at the FSSC<sup>8</sup>. Remember that you will need the event data files, the spacecraft file and the background models!

In order to perform analysis, the first step will be generating a text file listing all necessary event files. This can be done with a single command line:

```
$ ls *_PH* > binned_events.txt
```

Then, we want to do a selection on that data according to our preferred cuts on the event class, energy range, time selection and others. For such purposes we can use the tool *gtselect*<sup>9</sup>, just calling it in the terminal as

```
$ gtselect evclass=128 evtype=3
```

where the *evclass* and *evtype* parameters provided should be already the ones specific to our analysis – in this case SOURCE data (*evclass*=128) with a FRONT+BACK selection (*evtype*=3). The tool will then ask for the input files, the desired output file and the rest of the data cuts to be applied:

<sup>5</sup>[https://fermi.gsfc.nasa.gov/ssc/data/analysis/LAT\\_caveats.html](https://fermi.gsfc.nasa.gov/ssc/data/analysis/LAT_caveats.html)

<sup>6</sup><https://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/overview.html>

<sup>7</sup><https://heasarc.gsfc.nasa.gov/docs/software/heasoft/>

<sup>8</sup>[https://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/binned\\_likelihood\\_tutorial.html](https://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/binned_likelihood_tutorial.html)

<sup>9</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtselect.txt>

<b>P8R3 IRF name</b>	<b>Event Class</b>	<b>Class Hierarchy</b>	<b>Description</b>
P8R3_SOURCEVETO_V3	2048	Standard	This class has the same background rate than the SOURCE class background rate up to 10 GeV but, above 50 GeV, its background rate is the same as the ULTRACLEANVETO one while having 15% more acceptance.
P8R3_ULTRACLEANVETO_V3	1024	Standard	This is the cleanest Pass 8 event class. Its background rate is 15-20% lower than the background rate of SOURCE class below 10 GeV, and 50% lower at 200 GeV. This class is recommended to check for CR-induced systematics as well as for studies of diffuse emission that require low levels of CR contamination.
P8R3_ULTRACLEAN_V3	512	Standard	This class has a background rate very similar to ULTRACLEANVETO.
P8R3_CLEAN_V3	256	Standard	This class is identical to SOURCE below 3 GeV. Above 3 GeV it has a 1.3-2 times lower background rate than SOURCE and is slightly more sensitive to hard spectrum sources at high galactic latitudes.
P8R3_SOURCE_V3	128	Standard	This event class has a residual background rate that is comparable to P7REP_SOURCE. This is the recommended class for most analyses and provides good sensitivity for analysis of point sources and moderately extended sources.
P8R3_TRANSIENT010_V3	64	Standard	Transient event class with background rate equal to one times the A10 IGRB reference spectrum.
P8R3_TRANSIENT020_V3	16	Standard	Transient event class with background rate equal to two times the A10 IGRB reference spectrum.
P8R3_TRANSIENT010E_V3	32	Standard	Extended version of the P8R3_TRANSIENT020 event class with a less restrictive fiducial cut on projected track length through the Calorimeter.
P8R3_TRANSIENT020E_V3	8	Standard	Extended version of the P8R3_TRANSIENT010 event class with a less restrictive fiducial cut on projected track length through the Calorimeter.
P8R3_TRANSIENT015S_V3	65536	Standard	Transient event class designed for analysis of prompt solar flares in which pileup activity may be present. This class has a background rate equal to 1.5 times the A10 reference spectrum.

Table 2.1: Event classes (evclass) from the P8R3 data release. Source: FSSC.

```

Input FT1 file[] @binned_events.txt
Output FT1 file[] gtselect_output_example.fits
RA for new search center (degrees) (0:360) [] INDEF
Dec for new search center (degrees) (-90:90) [] INDEF

```

<b>Event Type</b>	<b>evtype</b>	<b>Description</b>
FRONT	1	Events converting in the Front-section of the Tracker.
BACK	2	Events converting in the Back-section of the Tracker.
FRONT+BACK	3	Events converting in the Back and Front sections of the Tracker.
PSF0	4	First (worst) quartile in the quality of the reconstructed direction.
PSF1	8	Second quartile in the quality of the reconstructed direction.
PSF2	16	Third quartile in the quality of the reconstructed direction.
PSF3	32	Fourth (best) quartile in the quality of the reconstructed direction.
EDISP0	64	First (worst) quartile in the quality of the reconstructed energy.
EDISP1	128	Second quartile in the quality of the reconstructed energy.
EDISP2	256	Third quartile in the quality of the reconstructed energy.
EDISP3	512	Fourth (best) quartile in the quality of the reconstructed energy.

Table 2.2: Event types (evtype) from the P8R3 data release. Source: FSSC.

<b>Generic Event Class</b>	<b>Recommendation</b>
TRANSIENT0XX	Short Timescale Source Analysis ( $T < 1000$ s) for short duration GRB or SFRs without significant pileup activity.
TRANSIENT0XXS	Solar Flare Analysis during time periods with pileup activity.
SOURCE	Source Analysis on Timescales $> 1000$ s.
CLEAN	Diffuse Analysis.

Table 2.3: Recommendations for event classes.

<b>Event Type</b>	<b>E&gt;50 MeV</b>	<b>E&gt;100 MeV</b>	<b>E&gt;200 MeV</b>	<b>E&gt;300 MeV</b>	<b>E&gt;500 MeV</b>
FRONT+BACK and EDISP0-3	80	90	95	100	100
FRONT	85	95	100	100	100
BACK	75	85	90	95	100
PSF0	70	80	85	90	95
PSF1	75	85	95	100	100
PSF2	85	95	100	100	100
PSF3	90	100	100	100	100

Table 2.4: Recommendations for the cut on the maximum zenith angle (ZMAX; in degrees).

```

radius of new search region (degrees) (0:180) [] INDEF
start time (MET in s) (0:) [] INDEF
end time (MET in s) (0:) [] INDEF
lower energy limit (MeV) (0:) [] INDEF
upper energy limit (MeV) (0:) [] INDEF
maximum zenith angle value (degrees) (0:180) [] INDEF

```

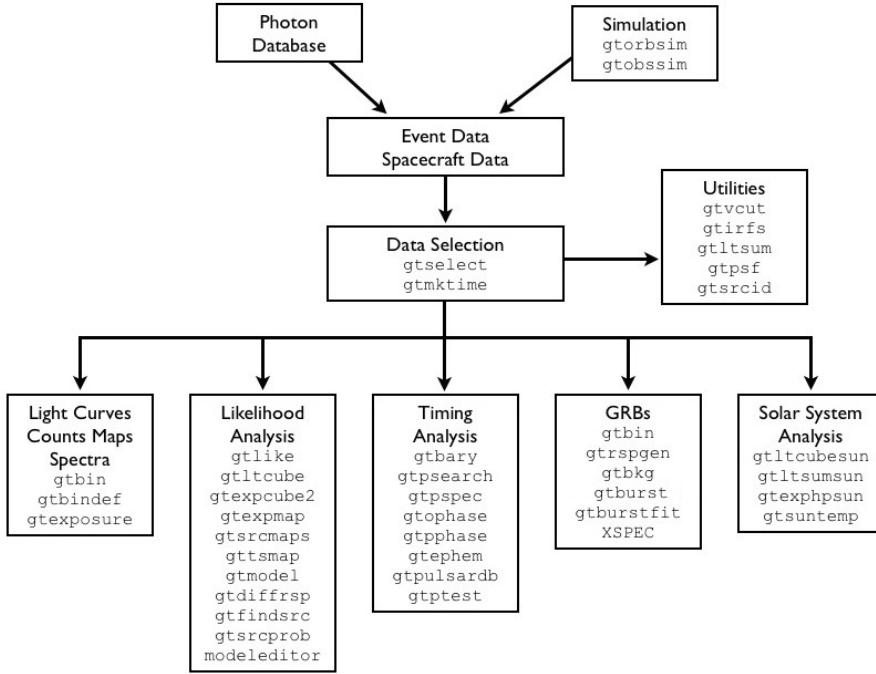


Figure 3.1: Hierarchical tree of the tools designed specifically for the analysis of LAT data. Image credit: FSSC.

Note that if a text file is provided as an input, it should be preceded by "@". The rest of the parameters should be selected at user's discretion according to the needs of the analysis. The output of the *gtselect* tool will be now a single fits file containing an event list with a selection in energy  $E$  and zenith angle  $\theta$  for a particular event class and type. However, the LAT might not be taking data always in science mode, and some spacecraft event might have affected the quality of the data. To ensure a good quality selection, a second step is necessary using the tool *gtmktime*<sup>10</sup>

\$ gtmktime

requesting additional information as

```

Spacecraft data file[] spacecraft_file_example.fits
Filter expression[] (DATA_QUAL>0)&&(LAT_CONFIG==1)
Apply ROI-based zenith angle cut[] no
Event data file[] gtselect_output_example.fits
Output event file name[] gtmktime_output_example.fits

```

where  $(\text{DATA\_QUAL}>0) \&\& (\text{LAT\_CONFIG}==1)$  ensures the selection to only contain events taken in normal science mode. No extra ROI-based zenith angle cut given that a more accurate maximum zenith angle was taken appropriately with *gtselect*.

Once the data selection is complete, in order to prepare a likelihood analysis we need to bin our data in a cuboid, as described previously. To do this, one can use the tool *gtbin*<sup>11</sup>. Such tool can be used to create simple count maps, light curves and others, but here we will focus

<sup>10</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtmktime.txt>

<sup>11</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtbin.txt>

on the creation of the cuboid. Again, after calling it in the terminal, the tool will request the parameters desired one by one:

```
$ gtbin
Type of output file (CCUBE|CMAP|LC|PHA1|PHA2|HEALPIX) [] CCUBE
Event data file name[] gtmktime_output_example.fits
Output file name[] example_ccube.fits
Spacecraft data file name[] NONE
Size of the X axis in pixels[]
Size of the Y axis in pixels[]
Image scale (in degrees/pixel) []
Coordinate system (CEL - celestial, GAL -galactic) (CEL|GAL) []
First coordinate of image center in degrees (RA or galactic l) []
Second coordinate of image center in degrees (DEC or galactic b) []
Rotation angle of image axis, in degrees[]
Projection method Projection method (OPTIONS) []
Algorithm for defining energy bins (FILE|LIN|LOG) []
Start value for first energy bin in MeV[]
Stop value for last energy bin in MeV[]
Number of logarithmically uniform energy bins[]
```

where the size of your cuboid has to be defined within the angular cone selected previously. The typical size of the bins (or pixels) in space coordinates will be the PSF at the highest energies of your data selection (see Figure 1.4), while in energy 8 or 10 bins per energy decade (i.e. in logarithm scale) will suffice. In order to have a good visualization, no rotation of the image is required, while a Hammer-Aitoff projection (AIT) is typically employed in *Fermi*-LAT publications. Do not worry if you got a "warning" for not using spacecraft file, it is not necessary for the cuboid. The output will then contain as many 2D images (count maps) as energy bins were created.

Until now, we only have cropped and binned our data. But before starting with the modelling and fitting, we need to compute the LAT's exposure across our ROI. In order to speed up this process, we can previously compute the live-time for each sky position and save it in a single file. In this way, we can re-use this information every time we want, without having to recompute it, which would be very computationally expensive. Such information can be obtained using the tool *gltcube*<sup>12</sup>

```
$ gltcube zmax=YOUR_MAXIMUM_ZENITH_ANGLE_VALUE
```

which only requires a few input parameters:

```
Event data file[] gtmktime_output_example.fits
Spacecraft data file[] spacecraft_file_example.fits
Output file[] ltcube_output_example.fits
Step size in cos(theta) (0.:1.) []
Pixel size (degrees) []
```

where a typical step size in  $\cos(\theta)$  can be  $\sim 0.025$ . Note that this function is creating a full-sky exposure map, therefore using the same pixel size from your ROI will take even longer. For this

---

<sup>12</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gltcube.txt>

step,  $1^\circ$  will suffice.

We therefore are ready to compute our exposure map with the tool `gtexpcube2`<sup>13</sup>. However, its size should be bigger than your ROI: LAT's PSF will make that photons from source outside your ROI might impact your analysis. Thus it is necessary to expand it a few degrees (e.g.  $\sim 10^\circ$ ), so the exposure is estimated for those sources too. These and other information will be required by the tool as

```
$ gtexpcube2

Livetime cube file[] ltcube_output_example.fits
Counts map file[] none
Output file name[] example_expcube_file.fits
Response functions to use[]
Size of the X axis in pixels[]
Size of the Y axis in pixels[]
Image scale (in degrees/pixel) []
Coordinate system (CEL - celestial, GAL -galactic) (CEL|GAL) []
First coordinate of image center in degrees (RA or galactic l) []
Second coordinate of image center in degrees (DEC or galactic b) []
Rotation angle of image axis, in degrees[]
Projection method []
Start energy (MeV) of first bin[]
Stop energy (MeV) of last bin[]
Number of logarithmically-spaced energy bins[]
```

where you should remember to use the IRFs according to the event class used (Table 2.1).

And finally this leads us to the need for a model of the  $\gamma$ -ray sky. As explained previously, a likelihood analysis will require a prior knowledge of the  $\gamma$ -ray sources within the ROI and beyond. The *Fermi tools* require such information in a very specifically formatted xml file, which can be done by hand using the `modeleditor` tool<sup>14</sup> or we can make use of the `make4FGLxml.py` script<sup>15</sup>. Here we will use the later: a python script using the raw 4FGL catalogue to create a reduced xml file particular to your ROI. You can run it in a single line,

```
$ python make4FGLxml.py file_catalogue.fit gtmktime_output_example.fits
```

but we (may) have to specify many parameters:

- -o: output file
- -G: path to Galactic diffuse model FITS file (see Section 1.3.4)
- -I: path to isotropic diffuse template file (see Section 1.3.4)
- -r: radius in degrees from ROI center beyond which sources will be fixed, default is selection radius
- -s: average significance below which all source parameters are fixed, default is  $5\sigma$

<sup>13</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtexpcube2.txt>

<sup>14</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/modeleditor.txt>

<sup>15</sup><https://fermi.gsfc.nasa.gov/ssc/data/analysis/user/make4FGLxml.py>

- -ER: extra radius beyond the ROI radius out to which sources will be included (sources outside the ROI will be fixed of course), default is 10 degrees

Although it would be ideal to fit all possible parameters, usually there are too many free parameters to gain a good spectral fit. Therefore it is a better option to revise these values so that only sources near your target, or very bright source, have all spectral parameters free. Farther away, you can fix the spectral form and free only the normalization parameter (or "prefactor").

Once the model is created, one should convolve its components with the IRFs. For such purpose the tool *gtsrcmaps*<sup>16</sup> will simplify our calculations just requiring the previously produced files

```
$ gtsrcmaps

Exposure hypercube file[] ltcube_output_example.fits
Counts map file[] example_ccube.fits
Source model file[] example_model.xml
Binned exposure map[] example_expcube_file.fits
Source maps output file[] example_srcmaps.fits
Response functions[] CALDB
```

where the *Fermitoools* will use their own standard calibration database (or CALDB), reading the IRFs from the files provided and using them accordingly. Hence, *gtsrcmaps* has taken each source spectrum in the xml model, multiplied it by the exposure at the source position, and convolved that exposure with the effective PSF. Although this process is not particularly time-consuming, it will require a substantial amount of memory.

Now you are ready to perform the maximum-likelihood fit. This can be done with the tool *gtlike*<sup>17</sup>:

```
$ gtlike
```

whose usage will benefit from the knowledge of a few hidden parameters:

- sfile: output fitted model
- ftol: relative fit's convergence tolerance. By default, it will be 0.01
- refit: whether to prompt to refit, re-reading the source model XML file before refitting. Default is "no"
- edisp: whether to consider energy dispersion in the fit. The default value is "no"
- psfcorr: whether to apply psf corrections for point source maps. Default is "yes"

Note that energy dispersion might be necessary depending on the energy cuts applied previously. When the energy dispersion correction is enabled, the *Fermitoools* will add two energy bins above and below the analysis energy range when evaluating the counts model. These additional bins are used to evaluate the contribution of events with true energies outside the range of your analysis. Such process should be applied to all sources except those that have already

---

<sup>16</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtsrcmaps.txt>

<sup>17</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtlike.txt>

been corrected for energy dispersion, or have been modelled already considering it. For the 4FGL standard sources, this only applies to the isotropic diffuse emission! Unfortunately, this requires modifying the model by hand adding `apply_edisp=false` to the xml file as in the following example:

```
<source name="isodiff" type="DiffuseSource">
<spectrum file="iso_IRF_version.txt" type="FileFunction" apply_edisp="false">
<parameter free="1" max="1000" min="1e-05" name="Normalization"
scale="1" value="1" />
</spectrum>
<spatialModel type="ConstantValue">
<parameter free="0" max="10.0" min="0.0" name="Value" scale="1.0" value="1.0"/>
</spatialModel>
</source>
```

In this example you can also see how free parameters are marked with `free="1"`, while the fixed ones will include `free="0"`. You can also check in this way that your sources where appropriately handled by the modelling script, or adjust them if you want. After calling the `gtlike` tool with your preferred hidden parameters, it will ask you some of the files previously generated

```
Statistic to use (BINNED|UNBINNED) []
Counts map file[] example_srcmaps.fits
Binned exposure map[] example_expcube_file.fits
Exposure hypercube file[] ltcube_output_example.fits
Source model file[] example_model.xml
Response functions to use[] CALDB
Optimizer []
```

where, given the largely multidimensional problem, we will also need specialized function solvers. Indeed, you can choose five different optimizer algorithms for `gtlike`: DRMNGB, DRMNFB, MINUIT, NEWMINUIT, and LBFGS. Most of them are well known solvers used in/by other packages, but adapted and introduced within the frame of the *Fermitoools*. Here you will find a brief description of each one adapted from the tool's documentation:

- **DRMNGB**<sup>18</sup>: This optimizer finds the local minima of a continuously differentiable function subject to simple upper and lower bound constraints. It uses a variant of Newton's method with a quasi-Newton Hessian updating method, and model/trust-region technique to aid convergence from poor starting values. The original code was in Fortran, but it has been converted in C++. It has some converge problems.
- **DRMNFB**<sup>19</sup>: Such optimizer uses many of the same subroutines as DRMNGB, but handles the derivative information differently and seems not to suffer from some of the convergence problems encountered with DRMNGB.
- **MINUIT**<sup>20</sup>: This is a well-known package from CERN designed for multi-parameter function solving. Originally in Fortran, it was also translated to C++. In the *Fermitoools*, only a few of MINUIT's capabilities are used. For example all variables are treated as bounded.

<sup>18</sup><https://netlib.sandia.gov/port//drmngb.f>

<sup>19</sup><https://netlib.sandia.gov/port//drmnfb.f>

<sup>20</sup><https://cdsweb.cern.ch/record/2296388/files/minuit.pdf>

No user interaction is allowed and only its MIGRAD algorithm is implemented.

- **NEWMINUIT**: It interfaces with an entirely new code of 'true' C++ designed in an object-oriented way, but it implements the optimization by using MINUIT. But NEWMINUIT uses only a few of MINUIT's features: The MIGRAD and HESSE algorithms. All variables are treated as bounded again. No user interaction is allowed. It has no limits on the number of free parameters, although a practical limit is often reached beyond which any fit is suspect or does not converge.
- **L-BFGS**<sup>21</sup> The last optimizer was also written in Fortran, but as the others, it was translated to C++. The "L" in the name means "limited memory". That means that the full approximate Hessian is not available.

In general, the fastest way to perform a parameters' estimation in the likelihood is using the DRMNGB or DRMNFB optimizer to find initial values and then use MINUIT or NEWMINUIT to find more accurate results. However, you should note that we are only fitting spectral parameters: the location of the sources remains fixed! Therefore, if you want to search for a new source, you should use the tool *gtfindsrc*<sup>22</sup>. Fortunately, this will not be necessary during this session.

If the fit successfully converges on a solution, the results will be saved in the output file indicated under the hidden parameter *sfile*. But to asses the quality of our results and refine it, we will employ the python package *fermipy*<sup>23</sup>.

### 3.2 Using fermipy

We will not follow any particular tutorial, but interested students may find some introductory notebooks on GitHub<sup>24</sup>.

Despite the benefits of directly using the *Fermitoools* through the Terminal, sometimes it might be convenient to use instead the python package *fermipy* [8]. This still uses the same *Fermitoools*, but reducing notably the parameters handled directly by the user and rather focusing on results' visualization. Here we will use the package to produce plots that will allow us to discuss (and quantify) the quality of our fit, and refit some parameters if required. Subsequently we will need two separate files: the python script itself (e.g. *not\_a\_virus.py*) and a configuration file (e.g. *config.yaml*). Let's focus first on the later.

The configuration (yaml<sup>25</sup>) file will contain all the parameters and paths needed by the package. Those are structured in hierarchical structure that keys parameters to a section (e.g. data, binning, ...). Thus the configuration file is actually a configuration dictionary! Such hierarchical structure can be easily identified in Listing 1, an example of a possible analysis on the Galactic Center. All required parameters and files appeared preciously in Section 3.1, therefore will not be described again here. As you can imagine, many other hidden parameters are not displayed, but hopefully they will not be necessary. Also note that if a necessary parameter is not provided, *fermipy* might either assume it as certain standard value, create it from scratch calling the *Fermitoools* if it is a file, or just break apart.

Once the configuration file is ready, we can start with python itself. Feel free to use a simple

<sup>21</sup><http://www.netlib.org/ampl/solvers/lbfgsb/README>

<sup>22</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtfindsrc.txt>

<sup>23</sup><https://fermipy.readthedocs.io/en/latest/index.html>

<sup>24</sup><https://github.com/fermiPy/fermipy-extra/tree/master/notebooks>

<sup>25</sup><https://yaml.org/>

script or a jupyter notebook<sup>26</sup>, whatever you prefer (using interactive python<sup>27</sup> is not recommended). In any case, you should import the *GTAnalysis* function from *fermipy* as

```
In [1]: from fermipy.gtanalysis import GTAnalysis
```

Listing 1: Example of a configuration file for an analysis of the Galactic Center.

```

1 data:
2     evfile : events.txt # If not in same folder, add paths!
3     scfile : SC00.fits
4     ltcube : ltcube_galactic_center.fits # Obtained from gtltcube
5
6 binning:
7     roiwidth    : 20.0
8     binsz       : 0.1 # Binning in degrees
9     binsperdec : 10 # Binning in energy (per log decade)
10    coordsys   : GAL
11
12 selection :
13     emin : 300
14     emax : 300000
15     zmax   : 90
16     evclass : 128 # SOURCE CLASS
17     evtype   : 3 # FRONT+BACK
18     tmin     : 239557414 # In MET
19     tmax     : 249557414
20     filter   : "DATA_QUAL==1 && LAT_CONFIG==1"
21     glon    : 0
22     glat    : 0
23
24 gtlike:
25     edisp : True # Energy
26     edisp_bins : -2 # Two extra bins to handle energy dispersion
27     irfs : 'P8R3_SOURCE_V3'
28     edisp_disable : ['isodiff'] # Only disable the isotropic diffuse
29     bexmap : 'exposure_galactic_center.fits'
30     srcmap : 'srcmap_galactic_center.fits'
31
32 model:
33     src_roiwidth : 30.0 # The model is larger than the ROI
34     galdiff   : 'gll_iem_v07.fits'
35     isodiff   : 'iso_P8R3_SOURCE_V3_v1.txt'
36     catalogs  : ['4FGL'] # Here you could use any xml model file

```

which immediately allows us to read the configuration file

---

<sup>26</sup><https://jupyter.org/>

<sup>27</sup><https://ipython.org/>

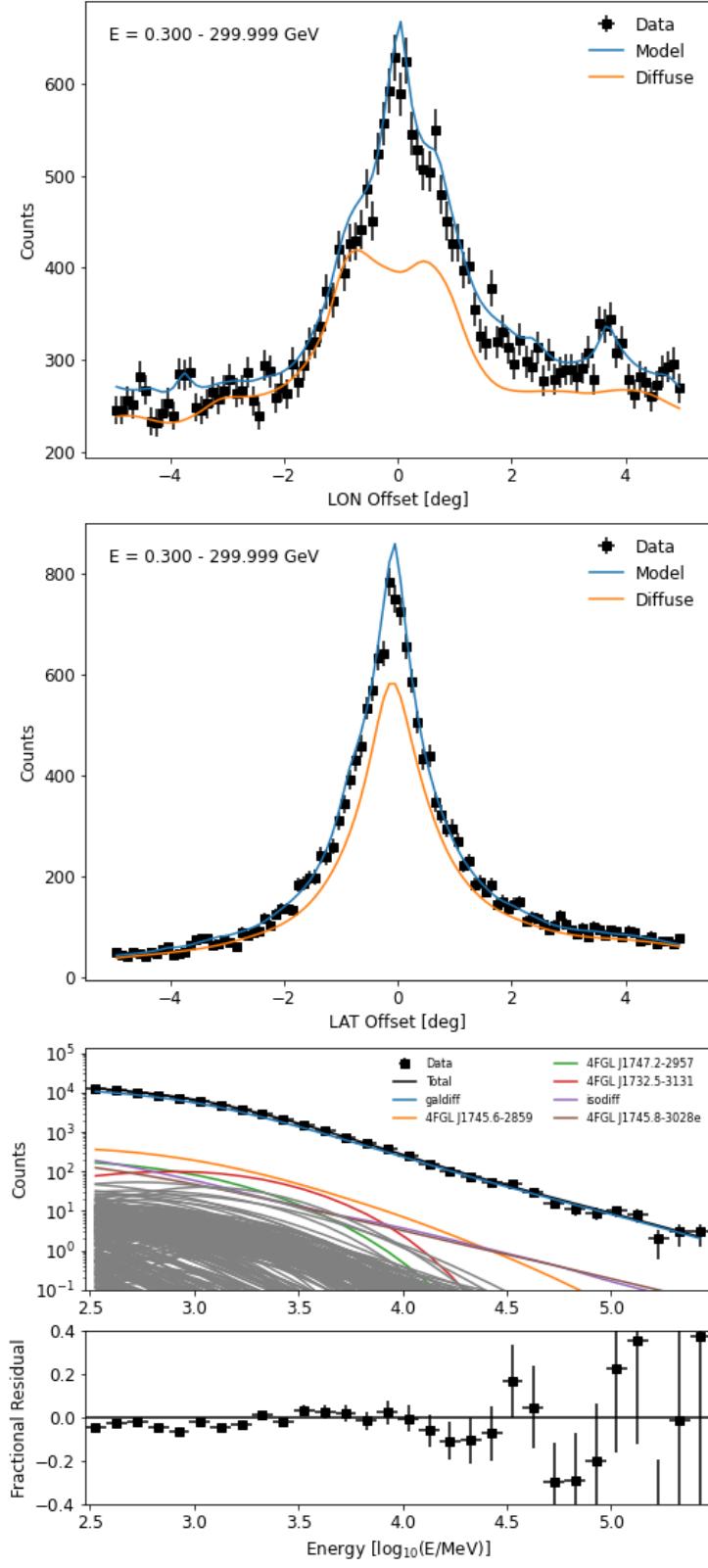


Figure 3.2: *Top:* Counts summed over latitude (i.e. projected on the longitude axis) for a ROI centred on the Galactic Center. *Middle:* Counts summed over longitude (i.e. projected on the latitude axis) for a ROI centred on the Galactic Center. *Bottom:* Sum of the spectrum of all sources in the ROI (in colour you will find the brightest ones).

```
In [2]: gta = GTAnalysis('boring_config_name.yaml')
```

and load it

```
In [3]: gta.setup()
```

That last line would create at once all exposure, live-time, source, or any other map needed for the analysis if we would not have them from Section 3.1. Therefore it is a very straightforward way of performing an analysis, although with way less versatility and a substantial amount of hidden parameters. Once the ROI is loaded, one can easily list all sources included in the model, their offset respect to the center, number of predicted counts and more with

```
In [4]: gta.print_model()
```

In order to start assessing the goodness of the fit, now we can produce a series of plots. We can save our current model as

```
In [5]: gta.write_roi('current_model', make_plots=True)
```

which, among others, will create plots with the sum over the ROI projected on longitude, latitude, and energy – see Figure 3.2 for an example of a fitted model to a ROI centred on the Galactic Center. In the particular case shown, you can see that the Galactic Plane’s diffuse emission is the brightest source, with a modelled photon flux almost compatible with the data. One can see small deviations here and there. But, are those real? Or expressed more appropriately, is any putative excess statistically significant? For demonstrative purposes, the source 4FGL J1745.6-2859 (i.e. a source in the inner Galactic region whose origin is attributed to Sgr A\* and its surroundings [9]) will be excluded from the model hereafter. Besides, given how crowded is the inner Galactic Plane, the plots shown will not correspond to the ROI, but a smaller region.

To properly characterize if there is a significant excess in our ROI, we can produce a *TS* map

```
In [6]: gta.tsmap('current_model', model={'SpatialModel' : 'PointSource', 'Index' : 2.0}, make_plots=True)
```

which computes the *TS* value of each pixel if a source modelled with a power law with spectral index  $\Gamma = 2$  is added. Two relevant plots are produced by *tsmap*, displayed in Figure 3.3. Both in the map and the *TS* distribution the excess above  $TS > 25$  is clear, implying the presence of a source not included in the model or that the fit converged in a local minimum. But then, how can we improve our fit?

We can free the parameters in our model with a single line with

```
In [7]: gta.free_source('4FGL J1745.6-2859')
```

or depending on the distance

```
In [8]: gta.free_sources(distance=5, pars='norm')
```

where the hidden parameter *pars* only frees the normalization of the sources (by default, all parameters are freed). We can then perform fit simply with

```
In [9]: gta.fit(optimizer='NEWMINUIT')
```

where any of the optimizers described in Section 3.1 can be used (by default, MINUIT will be used). Note also that parameters can be fixed again easily using

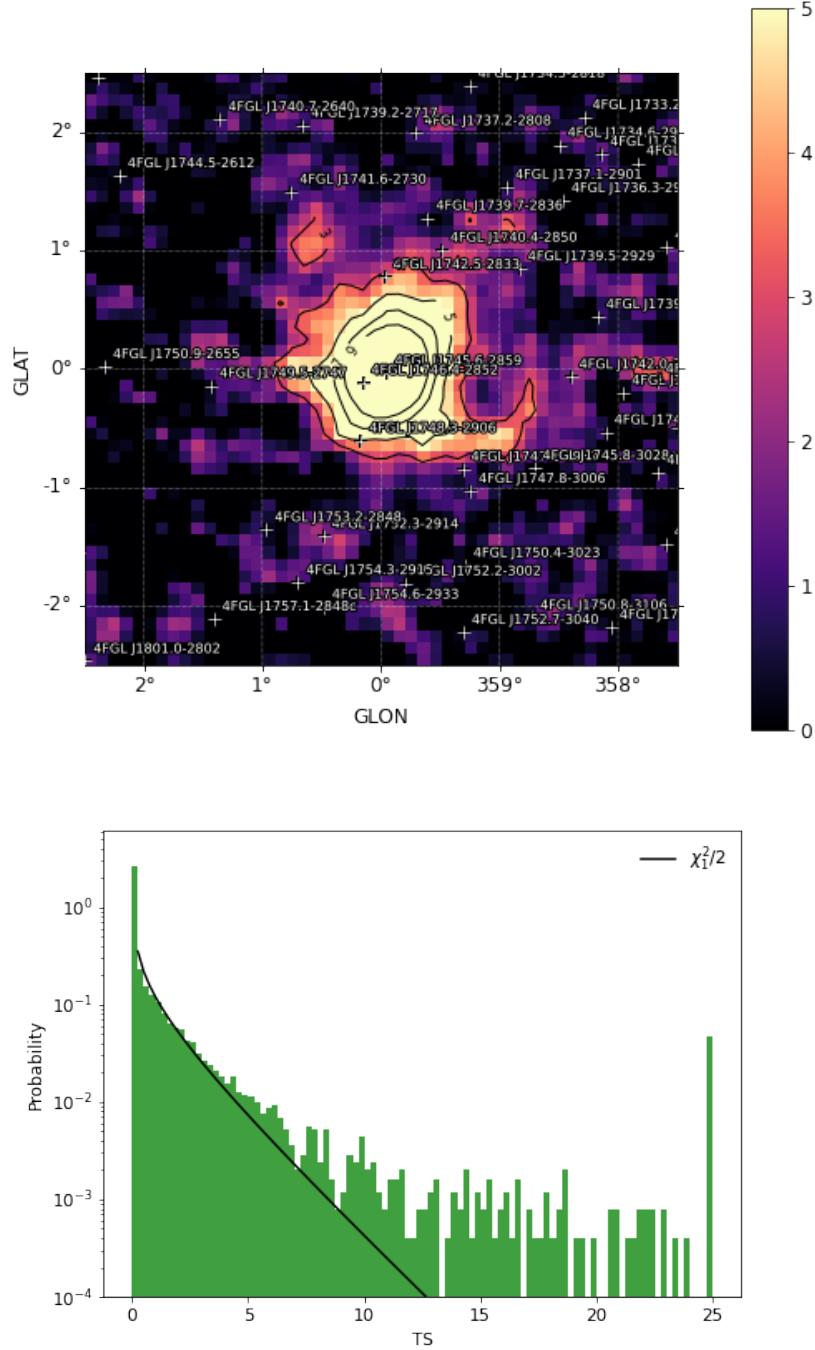


Figure 3.3: *Top:* Map of a ROI around the Galactic Center. The colour bar represents the square root of the  $TS$  value (i.e. the significance). The source 4FGL J1745.6-2859 is not included in the model, causing an excess. *Bottom:* Distribution of the  $TS$  values extracted from the sky map. If the model represents properly the ROI, the  $TS$  values should be distributed like a  $\chi^2$  distribution (black line).

```
In [10]: gta.free_sources(free=False)
```

To conclude with these notes on model optimization, new sources can also be added (or old ones deleted) on the fly or within the script with the following two lines:

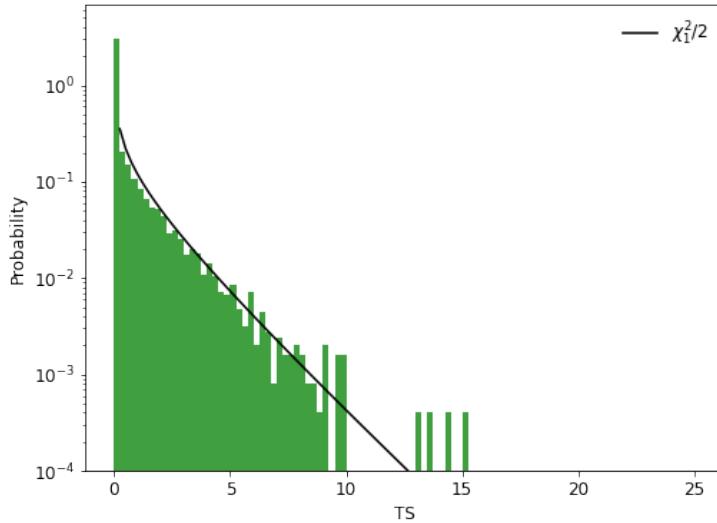
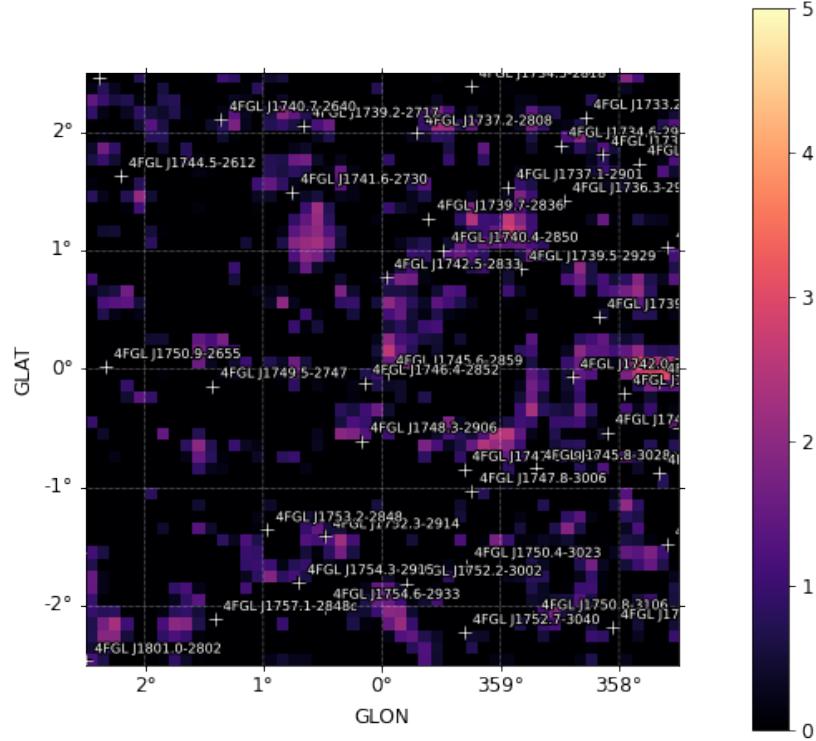


Figure 3.4: *Top*: Map of a ROI around the Galactic Center. The colour bar represents the square root of the  $TS$  value (i.e. the significance). The source 4FGL J1745.6-2859 is properly included in the model. *Bottom*: Distribution of the  $TS$  values extracted from the sky map. If the model represents properly the ROI, the  $TS$  values should be distributed like a  $\chi^2$  distribution (black line).

```
In [11]: gta.add_source('Sgr A*', {'glon' : 0.0, 'glat' : 0.0, 'SpectrumType' : 'PowerLaw', 'Index' : 2.0, 'Scale' : 1000, 'Prefactor' : 1e-11, 'SpatialModel' : 'PointSource' })

In [12]: gta.delete_source('Sgr A*')
```

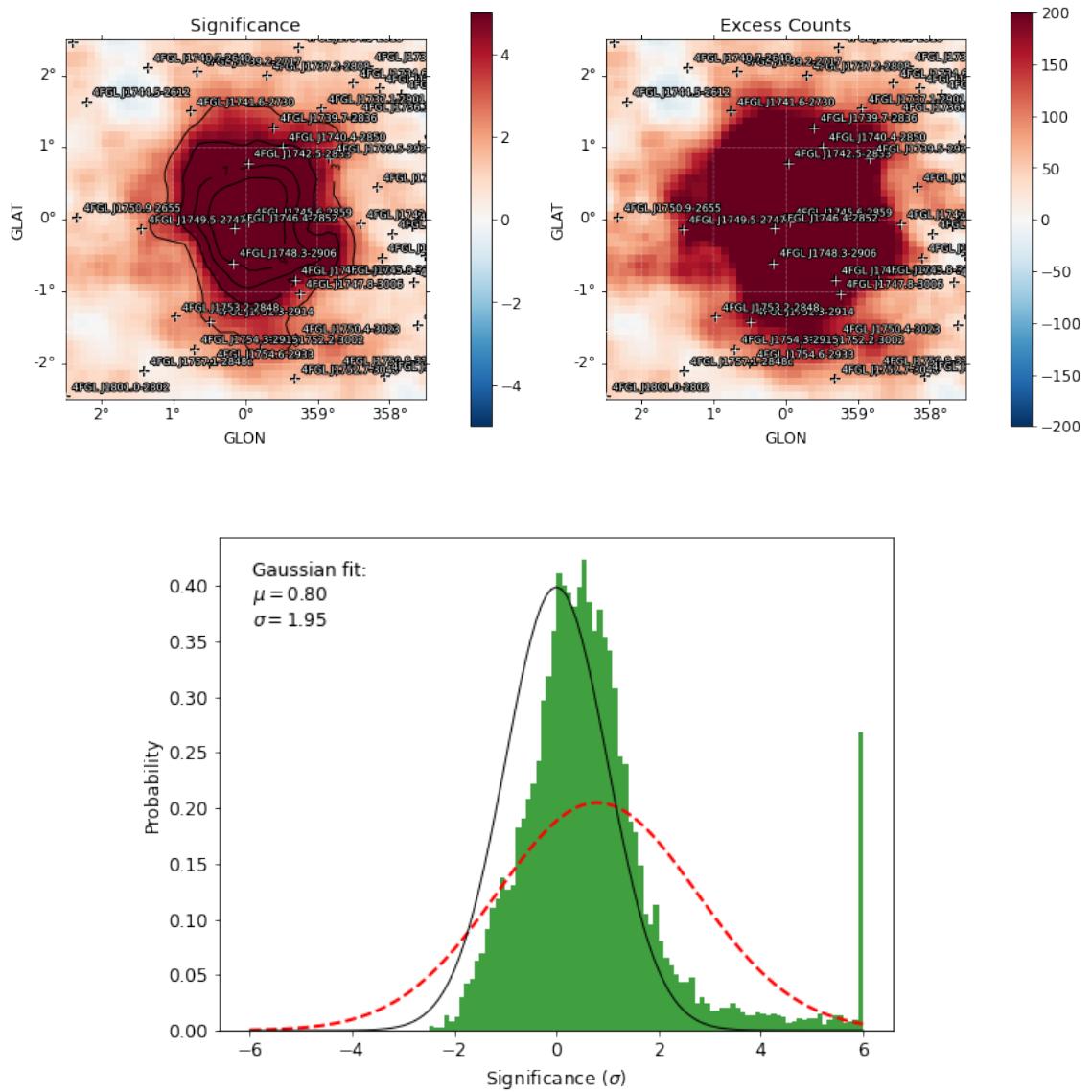


Figure 3.5: *Top:* Map of a ROI around the Galactic Center. The colour bar represents the significance of the deviation (*left*) or counts (*right*). The source 4FGL J1745.6-2859 is not included in the model, causing an excess. *Bottom:* Distribution of the significance values extracted from the sky map. If the model represents properly the ROI, significance values should be distributed like a Gaussian distribution (black line). The fit to the data is represented in red.

Therefore, with a few lines one can properly solve the presence of excesses. In Figure 3.4 the source 4FGL J1745.6-2859 is included again within the model, showing a clean ROI with an expected distribution of the  $TS$  values. However some might raise an important concern: what if there is no positive excess, but an over-subtraction of the model?

Although there are a few ways to do this (e.g. a PS map [10]), *fermpy* is able to produce residual maps, thus sensitive to both positive and negative deviations. Similarly to what is done with *tmap*, such residual map can be produced with

```
In [13]: gta.residmap('current_model', model={'SpatialModel': 'PointSource',
'Index' : 2.0}, make_plots=True)
```

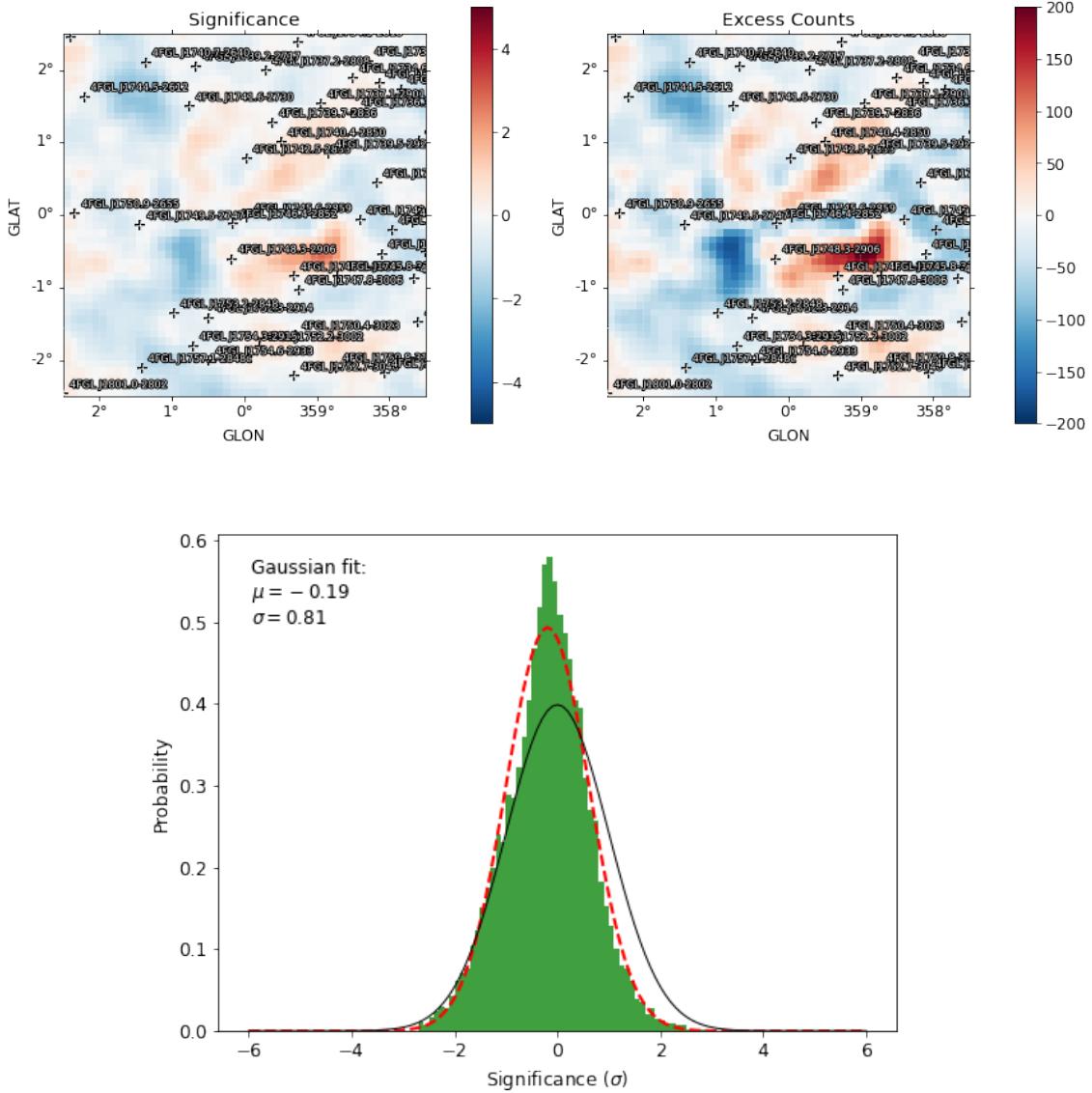


Figure 3.6: *Top*: Map of a ROI around the Galactic Center. The colour bar represents the significance of the deviation (left) or counts (right). The source 4FGL J1745.6-2859 is properly included in the model. *Bottom*: Distribution of the significance values extracted from the sky map. If the model represents properly the ROI, significance values should be distributed like a Gaussian distribution (black line). The fit to the data is represented in red.

again testing the assumption of having a source modelled with a power law with spectral index  $\Gamma = 2$  in each position. In Figure 3.5 one can see the excess from 4FGL J1745.6-2859 in the residual map, both in significance and counts. The distribution of those residuals should resemble

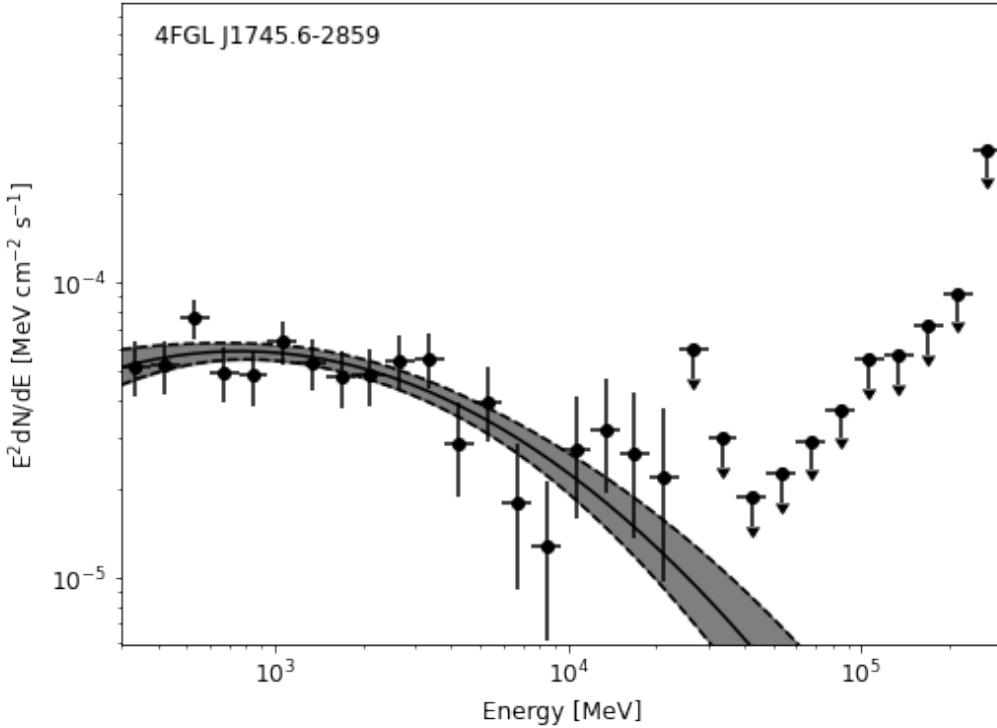


Figure 3.7: Spectral energy distribution of 4FGL J1745.6-2859, which is modelled by a Log-Parabola (see Section 1.3.4).

a Gaussian for pure white noise, therefore serving also as a critical check for the quality of the fit. Finally, Figure 3.6 shows how a clean residual should look like.

To conclude this section, the last thing we will need from *fermipy* is how to generate the spectral energy distribution of a source (see Figure 3.7). We can fit each energy bin of an energy source and its overall spectral model with

```
In [14]: gta.sed('4FGL J1745.6-2859', make_plots=True)
```

where via the hidden parameter `log_e_bins`, the user can provide a list of energy edges defining the bins, thus there is no need to use the ones from the overall likelihood analysis. You should note, however, that any other parameter from the overall fit should be fixed beforehand.

### 3.3 Aperture photometry

Before we have seen how to perform a maximum-likelihood analysis using the *Fermitools* directly or through *fermipy*. Sadly, if one wants to obtain a light curve of a certain object, such procedure should be repeated in each time bin, enlarging considerably the computational time required. To overcome this issue, here we will see how to use the *Fermitools* to obtain the time variation of any source's flux using aperture photometry. Thus this method provides a model independent measure of the flux for bright sources. However, a caveat should be raised beforehand: given the particularities of *Fermi*-LAT data, the adaptation of this method does not extract the background out like at other wavelengths (Figure 1.5).

Like for any standard analysis (Section 3.1), we start performing a data selection with `gtselect`,

but selecting photons only within  $1^\circ$ . Such cut allows to select mostly photons from our target, although some background contamination will be definitely included too. The data is also filtered using *gtmktime*, but the usage of *gtbin* differs significantly. Instead of using the CCUBE algorithm, we will use now the LC one:

```
Type of output file (CCUBE|CMAP|LC|PHA1|PHA2|HEALPIX) [] LC
Event data file name[]
Output file name[]
Spacecraft data file name[] NONE
Algorithm for defining time bins (FILE|LIN|SNR) [] LIN
Start value for first time bin in MET[]
Stop value for last time bin in MET[]
Width of linearly uniform time bins in seconds[]
```

where the bins can be previously defined in a file, be based on the signal-to-noise ratio or be equally spaced. The later case is recommended, thus a width for the bins is also required by the tool.

To proceed with aperture photometry, we also need the exposure. This can be computed for each bin using the *gtexposure*<sup>28</sup> tool as

```
$ gtexposure

Light curve file
Spacecraft file[]
Response functions[] CALDB
Source model XML file[] none
Photon index for spectral weighting[-2.1]
```

where the photon index of the source should be known from the likelihood analysis – otherwise,  $\Gamma = 2.1$  is used. Note that *gtexposure* requires  $-\Gamma$ , thus a negative value.

In this practical laboratory, there is no need to perform a barycentric correction with *gtbary*<sup>29</sup> given the extent of the bins. Therefore we have now all the ingredients to produce a light curve. Unfortunately, there is no automatic production of such plots with *Fermitoools* or *fermipy*. But a generic python script might serve us, just using standard packages like *NumPy* [11] and *Astropy* [12]. An example for such script is displayed in Listing 2.

Listing 2: Example of how to read a fits file and plot a light curve in python.

```
1 from astropy.table import Table as tb
2 import matplotlib.pyplot as plt
3
4 data = tb.read('/path/to/your/file.fits', format='fits')
5
6 rates = data['COUNTS'] / data['EXPOSURE']
7 err = data['ERROR'] / data['EXPOSURE']
8 time = data['TIME']
9 b = data['TIMEDEL'][0]/2 # bin size
10
11 plt.errorbar(time, rates, yerr=err, xerr=b, fmt='ko', label='Dataset')
```

<sup>28</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtexposure.txt>

<sup>29</sup><https://raw.githubusercontent.com/fermi-lat/fermitools-fhelp/master/gtbary.txt>

```

12 plt.xlabel('MET (s)')
13 plt.ylabel('Flux (ph/cm^2/s)')
14 plt.title('Example plot')
15 plt.legend()
16 plt.show()

```

## 4 BLAZARS

As you probably know, Active Galactic Nuclei (AGNs) are very luminous, distant galaxies powered by a central super-massive black hole accreting large amounts of gas from its surroundings. Blazars are a special class of AGNs where a relativistic jet points towards the observer (Figure 4.1). Within such jet, relativistic particles propagate with a bulk Lorentz factor  $\Gamma_{jet}$ , boosting their radiative emission. And, indeed, blazars radiate across all the electromagnetic spectrum, from radio to  $\gamma$ -rays.

### 4.1 High-energy emission in jets

Their spectral energy distribution has two bumps, the first one peaking near the optical range and the second one at  $\gamma$ -rays. The origin of the former is typically attributed to synchrotron radiation from relativistic electrons, while the later requires a bit more discussion.

One option can be that the same relativistic electrons also scatter low energy photons (e.g. radio photons) to higher energies via the inverse Compton (or IC) process. The option considered here, where the radiation field scattered is produced by the same particles, is commonly known as synchrotron self-Compton (or SSC). But another option is that a different particle population has something to do: relativistic protons. In that case, either synchrotron radiation from such particles or secondary particles created due to the collision of protons with the ambient medium could be responsible of the high-energy emission.

And how can we distinguish one model or another? Identifying *where* the putative electrons and protons might be accelerated. A first (and good) intuition can be to try mapping magnetic fields  $B$  with polarization studies. That's justified given that a large polarization implies a topologically ordered magnetic field within the jet or its compression in shear regions of the outflow. But  $\gamma$ -ray astronomy directly tracks the position of accelerated particles, and the variability of the high-energy component allows the characterization of the emitting region's size. Whatever process triggered the variation, it should be localized within a distance that particles can cover during the variability period. Thus motivating the identification (and study) of  $\gamma$ -ray flares.

### 4.2 Bayesian Blocks or how to identify a flare

From time to time, blazars enter flaring states with their luminosity changing drastically. If we want to identify them, we need to clearly discriminate between these flares and the normal (or quiescent) state – which, as we have seen, can be challenging for  $\gamma$ -ray astronomy due to the large statistical uncertainties. When do you define the start of a flare? And its beginning?

On one side, fitted functions are sometimes convenient in order to understand data, but they heavily rely on the underlying model. On the other side, the crude representation is susceptible to statistical fluctuations. In order to overcome this issue, a method commonly used for this purpose is Bayesian Blocks [14], a method seeking to find non-parametric representations of 1D data series, able to expose significant features of the original data. Here we will briefly

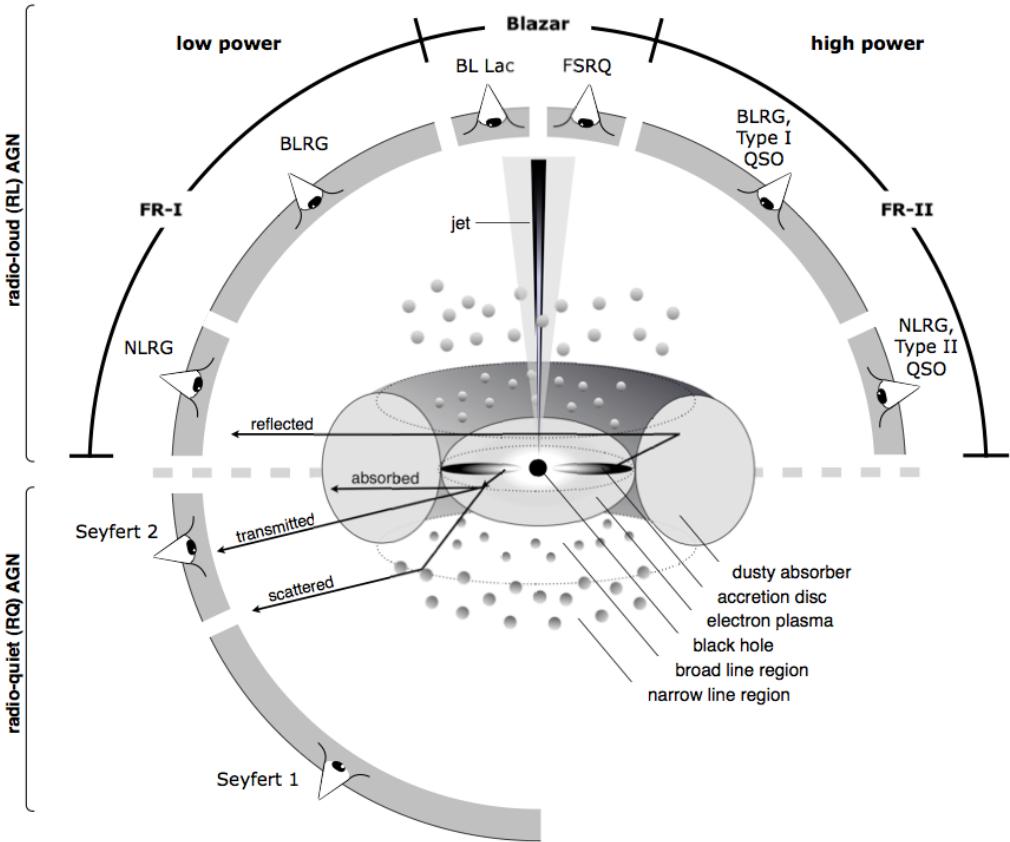


Figure 4.1: AGN unification scheme, where blazars can be identified as face-on systems. Image credit: [13].

discuss its main characteristics, focusing on time-binned data (although it can be applied also to event or measured data).

Consider  $N$  bins with the count-rates of our detector that last for a total time-lapse  $T$ . This is divided into  $N_b$  subintervals, called *blocks*, unequal on size. Each block will have a certain amplitude of its signal, and length, corresponding to how many bins does it include. Thus it is self-determined: nothing outside the block is needed to characterise it. Note that such segmentation produces a discontinuous signal. Nevertheless, this is an asset of the representation: we do not believe in the discontinuity of the signal, but we rather construct a tool to establish a collection of boundaries. We can call  $\mathcal{P}$  this partitions of the time interval  $T$  with various non-overlapping blocks.

The best representation is found optimizing a function (fitness  $F$ ) over all possible partitions, which will be the sum of the fitness of each block:

$$F(\mathcal{P}) = \sum_{k=1}^{N_n} f(B_k) \quad (4.1)$$

where many functions might be useful as a block fitness function  $f$ . As we have previously discussed, all information should be self-contained inside the bin itself. For a single bin, we can create a likelihood function given by the Poisson distribution as

$$\mathcal{L} = \frac{(\lambda e_n W_n)^{N_n} \cdot \exp(-\lambda e_n W_n)}{N_n!} \quad (4.2)$$

where  $N_n$  is the number of events in bin  $n$ ,  $\lambda$  is the actual event rate in counts per unit time,  $e_n$  is the average exposure over the bin and  $W_n$  is the width of the bin. We can now define a bin efficiency  $w_n = e_n N_n$  and construct the likelihood for the whole block  $k$  as

$$\mathcal{L}_k = \prod_{n=1}^{M_k} L_n = \lambda^{N_k} \cdot \exp(-\lambda w_k) \quad (4.3)$$

where  $M_k$  is the number of bins in block  $k$ . Finally, we actually need the log-likelihood function

$$\log \mathcal{L}_k = N_k \log \lambda - \lambda w \quad (4.4)$$

in order to satisfy the additive property required by Eq. 4.1.

Until this point, we have studied the blocks, but where is Bayesian statistics here? It is reasonable to assume that  $N_b \ll N$ , but how can we adopt a reasonable prior for  $N_b$ ? For this purpose, we can use for the probability distribution a geometric prior

$$P(N_b) = P_0 \gamma^{N_b} = \frac{1-\gamma}{1-\gamma^{N+1}} \gamma^{N_b} \quad (4.5)$$

where  $\gamma$  is a monotonic function smaller than 1, and  $P_0$  the normalisation constant. Therefore, the expected number of blocks will be

$$\langle N_b \rangle = P_0 \sum_{N_b=0}^N N_b \gamma^{N_b} = \frac{N \gamma^{N+1} + 1}{\gamma^{N+1} - 1} + \frac{1}{1-\gamma} \quad (4.6)$$

Luckily, the false positive rate  $p_0$  assuming a Poisson distribution of counts (like in Eq. 4.4) can also be related with  $\gamma$  via the expression

$$\log \gamma = 73.53 p_0 N^{-0.478} - 4 \quad (4.7)$$

thus providing a motivated prior depending on the significance that we will require to the studied features.

In order to use the Bayesian Blocks method, in this laboratory we will employ its implementation in *Astropy*<sup>30</sup>, so you do not have to reinvent the wheel (you can also find a MatLab implementation in [14]). Let's simulate a flaring source, which we will call "Kahless": a Gaussian-shaped signal with data scattered according to white noise (Figure 4.2). As we have seen, Bayesian Blocks will allow us to distinguish when the *start* and *end* of the flares can be defined with an arbitrary degree of certainty. First, we should import the package as:

```
In [15]: from astropy.stats import bayesian_blocks
```

and with a single line we can obtain the edges of the adapted histogram

```
In [16]: edges = bayesian_blocks(time, kahless_light, errors, fitness =
'measures', p0 = 0.05)
```

where  $p_0$  is the false alarm probability to compute the prior (in this case  $2\sigma$ ).

### 4.3 The case of 3C 279

The flat-spectrum radio quasar (FSRQ) 3C 279 ( $l = 305.10^\circ, b = 57.06^\circ; RA = 194.047^\circ, DEC = 5.789^\circ$ ) has been detected from radio wavelengths to TeV  $\gamma$ -rays [15]. Such blazar is at a redshift

---

<sup>30</sup>[https://docs.astropy.org/en/stable/api/astropy.stats.bayesian\\_blocks.html](https://docs.astropy.org/en/stable/api/astropy.stats.bayesian_blocks.html)

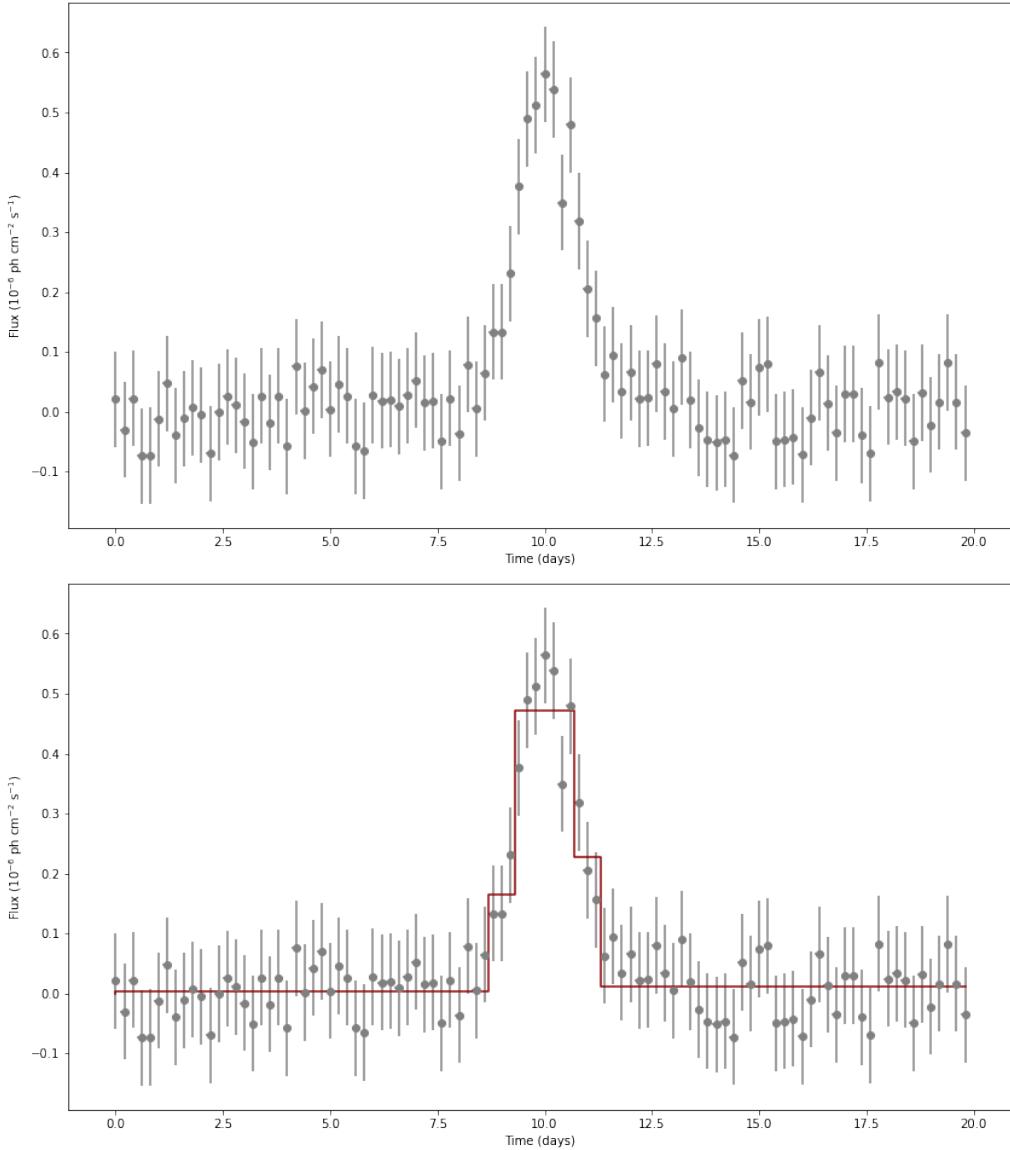


Figure 4.2: Light curve of the simulated flaring source Kahless. *Top:* Gaussian distribution convolved with white noise. *Bottom:* In red, Bayesian Block representation of the data.

of  $z = 0.536$ , harbouring a super-massive black hole of  $\sim 6 \cdot 10^8 M_\odot$  capable of launching a jet with  $\Gamma_{jet} \sim 16$ . Remember that a black hole's Schwarzschild radius will be

$$R_S = \frac{2GM_{SMBH}}{c^2} \quad (4.8)$$

Such source has displayed many  $\gamma$ -ray flares during the last decade, thus its has been deeply studied by many multi-wavelength observatories. In early 2009, the *Fermi*-LAT detected a GeV flare in 3C 279 coincident with a large change in the optical polarization as observed by the Triple Range Imager and Spectrograph (TRISPEC) instrument on the Kanata telescope [16]. Therefore, this correlation supports the idea that particle acceleration can be related with an alteration of the magnetic topology somewhere within the jet. But in which scales does this change occur? In a scenario of an emitting region co-moving with the jet plasma, it is possible to estimate the distance travelled by the emitting particles during the flare as

$$r \sim 5 \cdot 10^{17} \left[ \frac{\Delta t_{flare}}{1\text{day}} \right] \left[ \frac{\Gamma_{jet}}{15} \right]^2 \text{cm} \quad (4.9)$$

where  $\Delta t_{flare}$  is the duration of the flare. On the other hand, a lower limit can be set assuming that the emitting region does not propagate at all. Consequently, using

$$r \sim c \cdot \Delta t_{flare} \quad (4.10)$$

and Eq. 4.9, one can constrain the size of the emitting region in 3C 279 from  $\gamma$ -ray variability.

## 5 ASSIGNMENT TASKS

During the span of the two sessions, the students should complete the following tasks:

- **Exercise I:** Download 3C 279 data between January 1st 2009 (252460801 MET) and April 30th 2009 (262742402 MET) above 100 MeV. Perform a maximum-likelihood analysis on the data using the *Fermitools*.
- **Exercise II:** Employ *fermipy* to asses the quality of the fit.
- **Exercise III:** Produce a light-curve of the source using aperture photometry.
- **Exercise IV:** Use the Bayesian Block representation to determine the flaring and quiescent states of the blazar. If relativistic particles move at almost the speed of light, how large is the region where their energy is dissipated? Does this process occur close to the black hole's event horizon?
- **BONUS:** Does the spectral index of the blazar change between the flaring and quiescent states?

*Note: The bonus exercise is not mandatory and will only punctuate positively (max. +10%).*

## REFERENCES

- [1] B. Degrange and G. Fontaine. “Introduction to high-energy gamma-ray astronomy”. In: *Comptes Rendus Physique* 16.6-7 (2015), pp. 587–599. DOI: [10.1016/j.crhy.2015.07.003](https://doi.org/10.1016/j.crhy.2015.07.003).
- [2] W. B. Atwood et al. “The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission”. In: *Astrophysical Journal* 697.2 (2009), pp. 1071–1102. DOI: [10.1088/0004-637X/697/2/1071](https://doi.org/10.1088/0004-637X/697/2/1071).
- [3] M. Ajello et al. “Fermi Large Area Telescope Performance after 10 Years of Operation”. In: *Astrophysical Journal Supplement Series* 256.1, 12 (2021), p. 12. DOI: [10.3847/1538-4365/ac0ceb](https://doi.org/10.3847/1538-4365/ac0ceb).
- [4] M. Ackermann et al. “The Fermi Large Area Telescope on Orbit: Event Classification, Instrument Response Functions, and Calibration”. In: *Astrophysical Journal Supplement Series* 203.1, 4 (2012), p. 4. DOI: [10.1088/0067-0049/203/1/4](https://doi.org/10.1088/0067-0049/203/1/4).
- [5] S. Abdollahi et al. “Fermi Large Area Telescope Fourth Source Catalog”. In: *Astrophysical Journal Supplement Series* 247.1, 33 (2020), p. 33. DOI: [10.3847/1538-4365/ab6bcb](https://doi.org/10.3847/1538-4365/ab6bcb).

- [6] J. R. Mattox et al. “The Likelihood Analysis of EGRET Data”. In: *Astrophysical Journal* 461 (1996), p. 396. DOI: [10.1086/177068](https://doi.org/10.1086/177068).
- [7] W. D. Pence et al. “Definition of the Flexible Image Transport System (FITS), version 3.0”. In: *Astronomy & Astrophysics* 524, A42 (2010), A42. DOI: [10.1051/0004-6361/201015362](https://doi.org/10.1051/0004-6361/201015362).
- [8] M. Wood et al. “Fermipy: An open-source Python package for analysis of Fermi-LAT Data”. In: *35th International Cosmic Ray Conference (ICRC2017)*. Vol. 301. International Cosmic Ray Conference. 2017, p. 824.
- [9] M. Ajello et al. “Fermi-LAT Observations of High-Energy Gamma-Ray Emission toward the Galactic Center”. In: *Astrophysical Journal* 819.1, 44 (2016), p. 44. DOI: [10.3847/0004-637X/819/1/44](https://doi.org/10.3847/0004-637X/819/1/44).
- [10] P. Bruel. “A new method to perform data/model comparison in Fermi-LAT analysis”. In: *Astronomy & Astrophysics*, accepted (2021). DOI: [10.1051/0004-6361/202141553](https://doi.org/10.1051/0004-6361/202141553).
- [11] C. R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- [12] Astropy Collaboration et al. “The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package”. In: *Astronomical Journal* 156.3, 123 (2018), p. 123. DOI: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f).
- [13] V. Beckmann and C. R. Shrader. *Active Galactic Nuclei*. Wiley-VCH Verlag GmbH, 2012. DOI: [10.1002/9783527666829](https://doi.org/10.1002/9783527666829).
- [14] J. D. Scargle et al. “Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations”. In: *Astrophysical Journal* 764.2, 167 (2013), p. 167. DOI: [10.1088/0004-637X/764/2/167](https://doi.org/10.1088/0004-637X/764/2/167).
- [15] A. A. Abdo et al. “A change in the optical polarization associated with a  $\gamma$ -ray flare in the blazar 3C279”. In: *Nature* 463.7283 (2010), pp. 919–923. DOI: [10.1038/nature08841](https://doi.org/10.1038/nature08841).
- [16] M. Watanabe et al. “TRISPEC: A Simultaneous Optical and Near-Infrared Imager, Spectrograph, and Polarimeter”. In: *Publications of the Astronomical Society of the Pacific* 117.834 (2005), pp. 870–884. DOI: [10.1086/431914](https://doi.org/10.1086/431914).