

JULIUS-MAXIMILIANS-UNIVERSITÄT-WÜRZBURG

BACHELOR THESIS

---

# Rise and Decay Time of Blazars

---

*Author:*

Max Knopf

*Supervisor:*

Prof. Dr. Karl Mannheim



ASTRO WÜRZBURG



*A thesis submitted in fulfilment of the requirements  
for the degree of Bachelor of Science*

*in the group of*

Prof. Dr. Karl Mannheim

Chair of Astronomy



# Contents

<b>Zusammenfassung</b>	<b>1</b>
<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Active Galactic Nuclei . . . . .	3
1.1.1 Structure and Processes . . . . .	3
1.1.2 Classes of Active Galactic Nuclei . . . . .	5
1.1.2.1 Radio-Loud and Radio-Quiet Quasars . . . . .	5
1.1.2.2 Blazars . . . . .	6
1.1.3 Jets . . . . .	7
1.2 Gamma-Radiation . . . . .	10
1.2.1 Photomeson Production . . . . .	10
1.2.2 Synchrotron Radiation . . . . .	10
1.2.3 Inverse Compton Scattering . . . . .	11
1.2.4 Relativistic Beaming . . . . .	12
1.2.5 Detection-Methods . . . . .	12
1.3 Fermi Large Area Telescope . . . . .	13
1.3.1 Converter-Tracker . . . . .	13
1.3.2 Calorimeter . . . . .	14
1.3.3 Anticoincidence Detector . . . . .	14
1.3.4 Mode of Operation . . . . .	14
1.4 Goal of this Work . . . . .	14
<b>2 Observational Data and Methods</b>	<b>15</b>
2.1 Analysed Data and Coding Setup . . . . .	15
2.2 Bayesian-Block-Algorithm . . . . .	15
2.3 Dominance-Algorithm . . . . .	18
<b>3 Results</b>	<b>25</b>
<b>4 Conclusion and Outlook</b>	<b>29</b>
<b>Bibliography</b>	<b>30</b>
<b>Acknowledgements</b>	<b>34</b>
<b>Declaration of authorship</b>	<b>35</b>

<b>Appendices</b>	<b>37</b>
<b>A Python-Code</b>	<b>39</b>

# *Zusammenfassung*

Aktive Galaxienkerne (AGN) sind wegen ihrer hohen Variabilität in allen Spektren und ihrer außergewöhnlichen Leuchtkraft von allgemeinem Interesse in der Astronomie. Außerdem überstrahlen sie normale Galaxien um Längen. Bis jetzt wurden zahlreiche Modelle entwickelt und diskutiert, um deren Verhalten zu beschreiben, von Mechanismen die für die Entstehung von Jets verantwortlich sind bis hin zu verschiedenen Strahlungsprozessen, welche die beobachteten Emissionen von AGNs beschreiben. Die meisten AGN-Typen lassen sich über ihre Orientierung zum Beobachter beschreiben und werden in dem "Unification-Model" zusammengefasst, siehe [Urry and Padovani \(1995\)](#). Um mehr über diese Quellen und ihre zeitliche Variabilität in Erfahrung zu bringen werden die monatlich aufgezeichneten Lichtkurven im  $\gamma$ -Strahlen Spektrum mit Hilfe des Bayesian-Block-Algorithmus (BBA) und des entwickelten Dominanz-Algorithmus (DA) analysiert. Die Lichtkurven wurden mittels des "Fermi Gamma-ray Space Telescope" kurz Fermi/LAT aufgezeichnet. Dabei ist jede aufgezeichnete Messung mit einer Teststatistik behaftet, welche der Messung eine Gewichtung zuordnet. Unter Zuhilfenahme dieser Teststatistik werden die Lichtkurven mit einer 3 Sigma Konfidenzschwelle gefiltert, um deren Signifikanz festzustellen. Daraufhin wurde der BBA genutzt, um die Lichtkurve zu parametrisieren. Der DA wird auf die normale und die parametrisierte Lichtkurve jeder Quelle angewendet, um deren Extremstellen sowie die zugehörigen Anstiegs und Abstiegszeiten festzustellen. Die Evaluation der Resultate von fast 200 analysierten Lichtkurven zeigt, dass keine Tendenz in Bezug auf Abstieg und Anstieg vorliegt, bezogen auf ein lokales Maximum. Das beweist jedoch nicht, dass alle Lichtkurven symmetrisch sein müssen, da die Auswertung statistisch über alle Quellen durchgeführt wurde, das heißt einzelne Quellen können immer noch asymmetrisch ausgeprägt sein. Es lässt sich jedoch die Aussage treffen, dass im Allgemeinen keine Tendenz in dieser Ausprägung der Lichtkurven vorliegt.



# *Abstract*

Active Galactic Nuclei (AGN) are objects of general interest in astronomy due to their high variability in all of their spectra and their extraordinary brightness. They outshine normal galaxies by far. Up to this point, numerous models have been developed and discussed in order to explain their behaviour, from different mechanisms to describe the generation of jets to several radiative processes responsible for the emission observed by AGN's. Most of the different types of AGN's can be distinguished by their orientation to the observer and are summarised under the unification model, see [Urry and Padovani \(1995\)](#). In order to gain new knowledge about these sources and their variability their monthly binned light curves in the  $\gamma$ -ray spectrum are being analysed with the help of the Bayesian-Block-Algorithm (BBA) and the developed Dominance-Algorithm (DA). The light curves were observed with the Fermi Gamma-ray Space Telescope (Fermi/LAT), and every recorded measurement comes with a test-statistic. With the help of the test-statistic the sources were filtered with a 3-sigma confidence threshold to determine their significance. Then the BBA is employed to parameterise the light curves. From there on the DA is applied on the normal and the parameterised light curve to determine the extreme points and their respective rise and decay times. The evaluation of the results of almost 200 analysed sources shows that there is no tendency to either the rise or the decay time of an observed peak. However, this does not proof the point that all peaks must be symmetrical, since the evaluation was done statistically over all sources, single sources can still be asymmetric. Nevertheless, this shows that there is no bias in the overall shape of light curves.





# Chapter 1

## Introduction

### 1.1 Active Galactic Nuclei

According to [Schneider \(2014\)](#), active galactic nuclei (AGN), especially quasars, can be some of the brightest sources of radiation, even exceeding the luminosity of normal galaxies by a factor of thousand. The vast radiation bandwidths are a specific attribute of these AGN's indicating non-thermal radiation. AGN is just an umbrella term subsuming a multitude of different sources which nevertheless share a common build-up, a supermassive black hole (SMBH) resides in the middle of the nucleus and is surrounded by an accretion disk. Around this structure resides a torus which is able to absorb emitted radiation. If it is a radio-loud source a jet is also emitted.

Although it was previously mentioned that there is a multitude of different AGN's, both [Antonucci \(1993\)](#) and [Urry and Padovani \(1995\)](#) state that most of the AGN classifications are just resulting from the orientation of the source towards the observer and most of these classes can be subsumed under radio-loud and radio-quiet AGN's. This is referred to as the unification model. A summary of all classes and their orientation to the observer can be seen in figure [1.1](#).

#### 1.1.1 Structure and Processes

The SMBH is considered to be the central engine of the whole process and several arguments can be made in favour of this statement. [Schneider \(2014\)](#) focuses on the efficiency needed for the radiation at hand by comparing the efficiencies of nuclear fusion and accretion onto the black hole, he concludes that the efficiency of nuclear fusion is not high enough. [Tanaka et al. \(1995\)](#) argue via the relativistic effects caused by a SMBH on the radiation from the accretion disk and the ability to observe these effects.

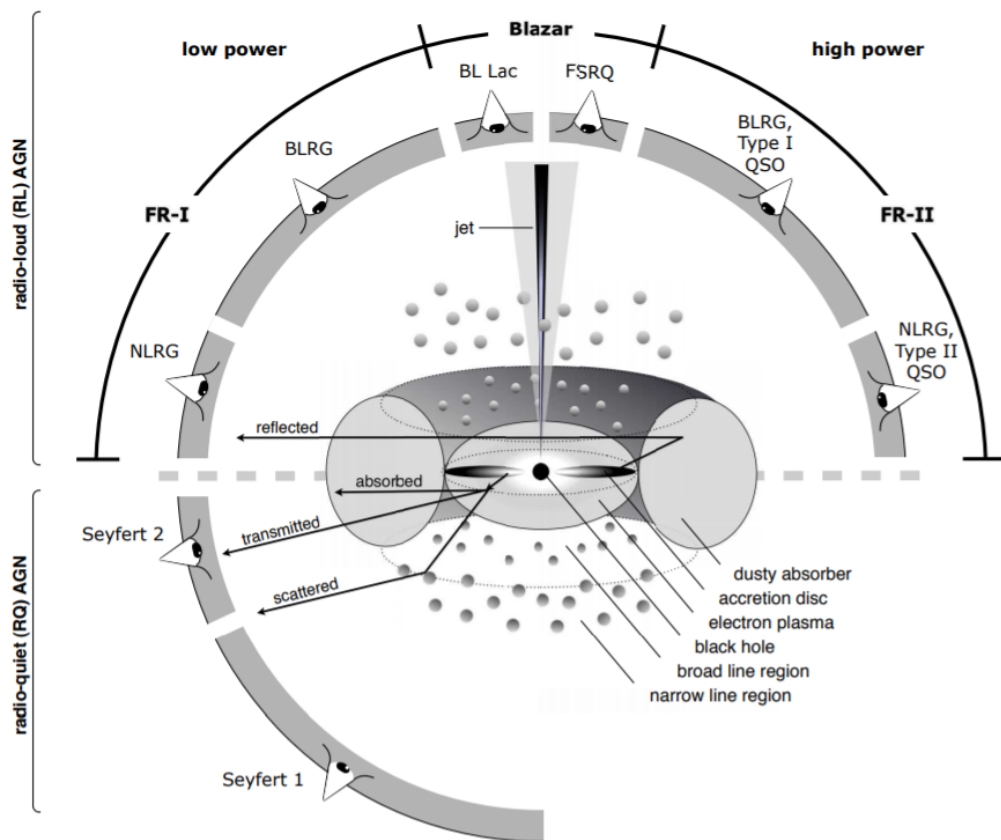


FIGURE 1.1: Representation of the unification model showing the different AGN classes according to their orientation to the observer. The upper part illustrates radio-loud galaxies and the lower part the radio-quiet galaxies. Credit: [Beckmann and Shrader \(2012\)](#)

Assuming that the energy arises from the gravitational force of the SMBH this gravitational pull accelerates the mass of the accretion disk onto the black hole in a circular motion. By doing so the potential energy is converted into kinetic energy. If the particles of the accretion disk simply fall into the SMBH they would not have enough time to radiate. However, since we expect the gas to have a finite angular momentum such a process is not possible. Thus, resulting in a disk perpendicular to the angular momentum. Considering Kepler's Laws, the angular velocity of the gas is dependent on the radius, this results in internal friction, converting the kinetic energy into heat and deceleration of the gas following an inward bound movement.

In the cases of FSRQ and BL Lac sources, the Blazars, there is also a collimated jet consisting of plasma emanating near the AGN in both directions and perpendicular to the accretion disk. According to [Schneider \(2014\)](#) the collimation follows from magnetic field lines which are anchored in the accretion disk. [Blandford and Payne \(1982\)](#) also discuss the accretion of matter with the help of a magnetic field. They state that a centrifugally induced outflow is possible if the poloidal component of the magnetic field

and the disk lie at an angle of maximum  $60^\circ$  to each other and that this outflow is then collimated by the toroidal components of the field. On the other hand, [Blandford and Znajek \(1977\)](#) presents an electromagnetic extraction of energy and angular momentum following from a rotating black hole which is threaded by a magnetic field. This model allows the acceleration of relativistic electrons without serious losses far away from the black hole.

The radiative processes in such a jet are often discussed and most of the time synchrotron radiation and inverse Compton scattering are mentioned as the main reason. However, [Blandford and Königl \(1979\)](#) discuss the possibility of relativistic shockwaves in jets resulting in doppler boosted fluxes. [Mannheim \(1993\)](#) adds to the relativistic idea with the hadronic concept of a proton Blazar, this concept makes the assumption that there can also be protons moving with relativistic velocities in the jet and resulting in photomeson production. This can give rise to other radiative processes and will be discussed in [1.2.1](#).

### 1.1.2 Classes of Active Galactic Nuclei

As discussed before in the beginning of [1.1](#), there are several different kinds of AGN. The most important ones for this thesis will be discussed in this section following mostly the explanation of [Beckmann and Shrader \(2012\)](#) if not stated otherwise.

#### 1.1.2.1 Radio-Loud and Radio-Quiet Quasars

Quasi-stellar radio sources (quasars) are the brightest types of AGN. They were first discovered via radio astronomy and are characterized by their blue star like appearance in the optical spectrum. They also exhibit a distinct redshift easily exceeding  $z = 2$ . However, only roughly 10% of quasars can be detected via their radio spectrum. The other 90% are the so-called radio-quiet quasars. As the name indicates the radio output of these sources is rather low, nevertheless, this does not imply that they are completely radio-silent. Since there is still a rest radio signal left, it is assumed that a small-scale jet with a vastly lower output than the usually observed ones powers the emission. The detection of compact nonrelativistic velocities in the Seyfert Galaxy NGC4151 by [Ulvestad et al. \(2005\)](#) supports this claim. An expected smooth transition between the two types of quasars has not yet been discovered. Because there is no real discrete distinction between radio-loud and radio-quiet the quietness is defined with the logarithmical quotient of the radio and the optic flux. If the condition  $R^* < 1$  is satisfied the source is radio-quiet.

$$R^* = \log \left( \frac{f_{radio}}{f_{optical}} \right) < 1 \quad (1.1)$$

The radio-loud quasars on the other hand feature a distinctive jet that is responsible for the radio emission which is probably produced with synchrotron emission and their host galaxies tend to be more massive.

### 1.1.2.2 Blazars

If the angle between the jet of an AGN and the point of view of an observer is small then they are called Blazars. They are characterised by a very high variability and emissions in the radio and x-ray domain and even higher energies (above 1TeV), e.g.  $\gamma$ -radiation. The flux variations can develop in the time scale of days or even as far as years. [Albert et al. \(2007\)](#) even show observed flux variations of Markarian 501 in a time scale of minutes. Blazars can be separated into two main groups BL lac and FSRQ objects, BL Lac's in contrast to FSRQ often show no broad or narrow emission lines and if they do, they are rather weak. This follows from the nonthermal continuous radiation which overshadows the emission lines. According to [Schneider \(2014\)](#) the emitted light in the optical spectrum is highly polarized, and if the BL Lac is quiescent the nonthermal radiation is weak enough so that the emission lines are observable. If that is not the case determining the redshift is nearly impossible.

The flat spectrum radio quasars (FSRQ) can be identified by their broad Balmer lines despite the nonthermal continuum and therefore it is significantly easier to determine their redshift. Another way to distinguish between BL Lac and FSRQ is by comparing the ratios of their Eddington luminosity, here the BL Lac's show lower ratios than the FSRQ.

[Urry and Padovani \(1995\)](#) mention that the extended radio distributions of high-redshift BL Lacs and FSRQ are distinguishable with a confidence level of 99,9%. This is not the only distinction they found. The spectrum of the X-ray emissions is also systematically different in which the spectral indices of FSRQ's are lower than the BL lac ones. In addition, the polarization structure of the BL Lac sources implies a magnetic field perpendicular to the jet axis, whereas the magnetic field of quasars resides parallel to the jet axis. [Figure 1.2](#) also shows a clear difference between the Mg II emission line luminosities between the two classes.

[Ghisellini et al. \(2017\)](#) discusses the bolometric luminosity of BL Lac and FSRQ, citing that the FSRQ's show approximately the same spectral energy distribution (SED) as the luminosity increases and consequently the X-ray spectra become harder for higher luminosities. BL Lacs on the other hand become more red for increasing luminosities

with a softer  $\gamma$ -ray slope and a larger dominance of the high energy peak. Figure 1.3 shows the original Blazar sequence indicating these features.

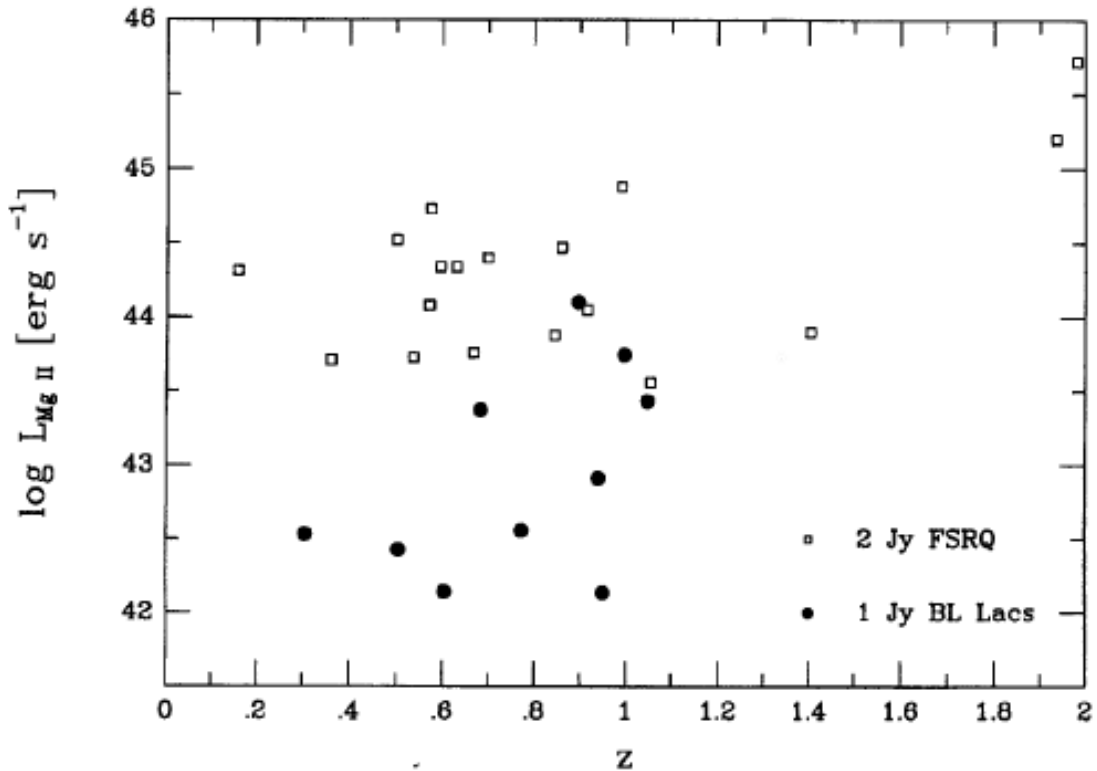


FIGURE 1.2: Mg II emission-line luminosities of 1 Jy BL Lacs (black circle) and 2 Jy FSRQ (open squares) plotted against their redshift. With some exceptions a clear separation of the two is observable. Credit: [Urry and Padovani \(1995\)](#)

### 1.1.3 Jets

The origin of jets is still uncertain, however, there are different models discussing the processes within a jet to explain the observed radiation. For instance [Blandford and Königl \(1979\)](#) and [Blandford and McKee \(1977\)](#) explain that the relativistic electrons needed for synchrotron and Compton processes have to be accelerated within the jet or otherwise it would require a vast external power. To achieve this, they propose relativistic shock waves travelling through the plasma possibly following from an unsteady velocity field or dense clouds getting accelerated by the flow. A cloud or a blob of gas entering a supersonic jet will lead to a bow shock. A lot of the bulk kinetic energy flux will then dissipate and a part of it may appear as ultra-relativistic electrons behind the shock. That process also amplifies frozen-in magnetic fields which favours synchrotron and inverse Compton radiation. The acceleration of the clouds inside of the jet may be the cause for observed flux variations.

Norman et al. (1982) argue that the supersonic jet propagates with high Mach numbers ( $M_b \approx \geq 6$ ). The jet structure is similar to the structure of supersonic laboratory jets. They are dependent on the ratio of the jet pressure and the ambient pressure. If the jet medium transitions from a high-pressure reservoir into a lower pressure medium it is called an underexpanded jet. After the gas leaves through the orifice it experiences so-called Prandtl-Meyer-expansions which lead to shocks and the Mach disk in the propagating medium. A representation of this can be found in figure 1.5. However, there are two major differences. The first one is that the observed jets of AGN's generate their own effective orifices and the second one is that in the rest frame of the Mach disk the intergalactic medium (IGM) flows with supersonic velocity towards the Mach disk.

Sokolov et al. (2004) also takes into consideration that a possible shock wave collides with the Mach disk and results in the creation of flares.

Konigl (1981) argues that relativistic jets take a big role in the production of high energy radiation such as X-ray and  $\gamma$ -radiation. The relativistic movement results in a Doppler-shift of the higher frequencies if the point of view is close to the symmetry-axis. This effect is called beaming.

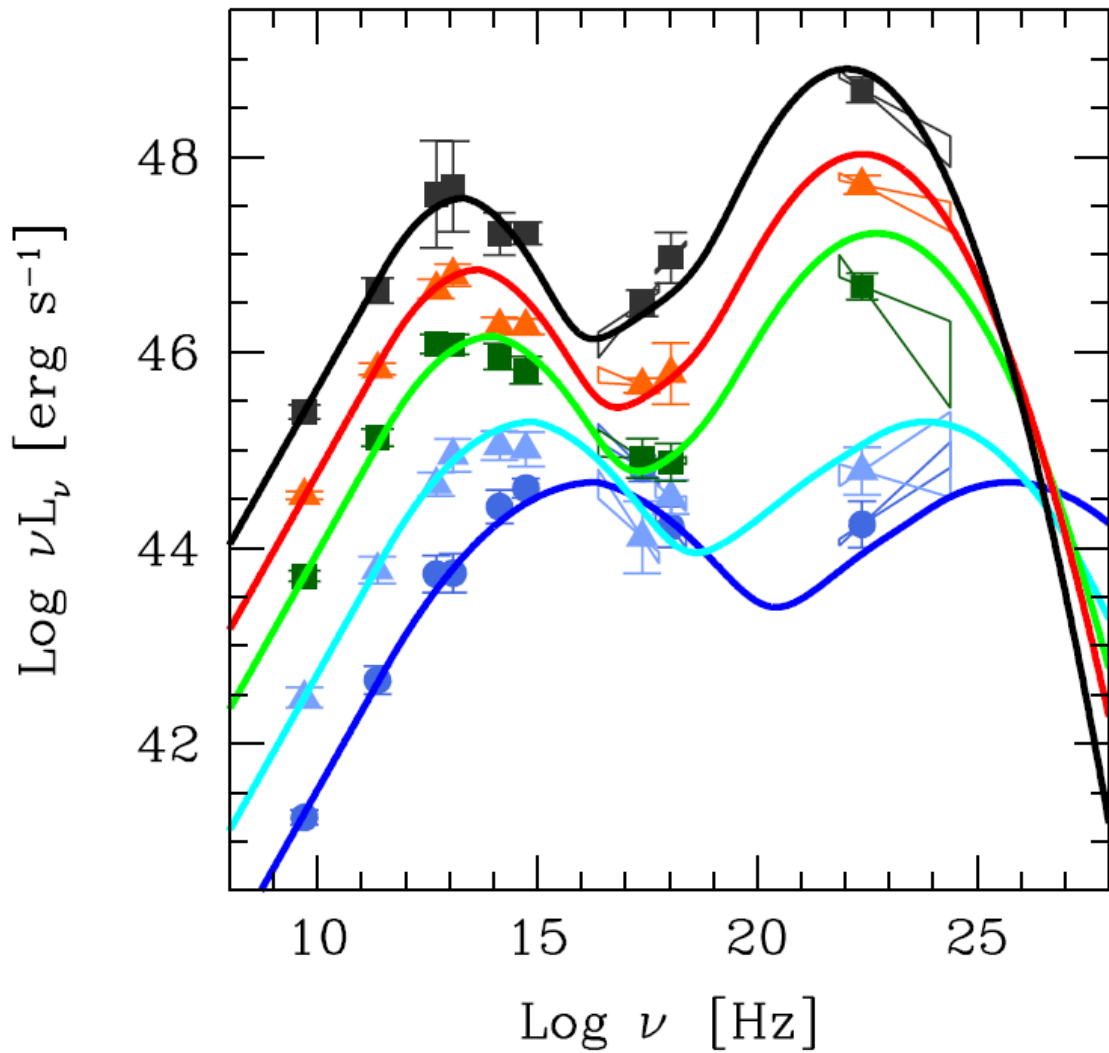


FIGURE 1.3: Blazar sequence consisting of 126 sources from radio and X-ray samples, showing a characteristic form with 2 peaks. Credit: [Ghisellini et al. \(2017\)](#)

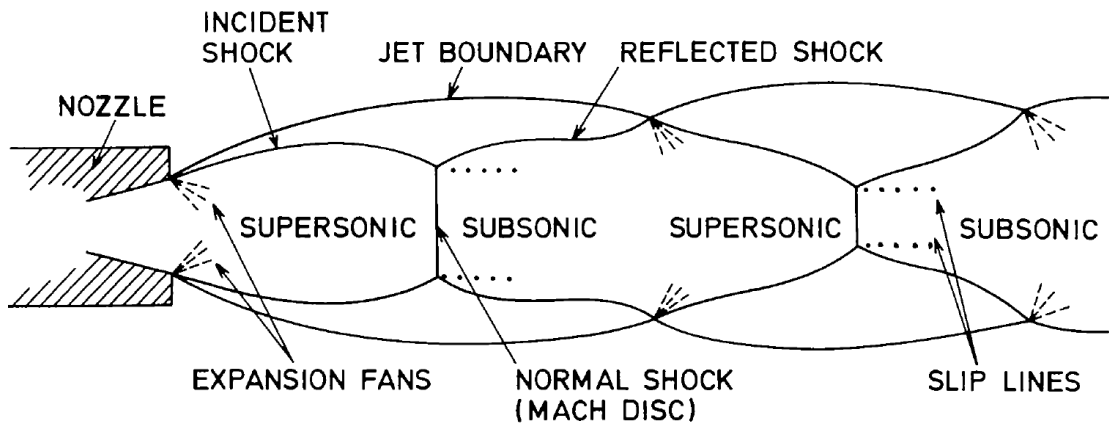


FIGURE 1.4: Structure of a supersonic jet with an underexpanded nozzle, the gas emanating from the reservoir behind the nozzle has a higher pressure than the surrounding medium. Credit: [Norman et al. \(1982\)](#)

## 1.2 Gamma-Radiation

Since the postulation of the  $\gamma$ -radiation in the 1950's numerous telescopes have been launched. However, the satellites SAS-2 and COS-B were the first to enable detailed observations of  $\gamma$ -ray-sources. Among other things they observed the first pulsars in gamma range and confirmed the gamma background radiation. In the year 2008 the Fermi Gamma-ray Space Telescope, with which the data discussed in this thesis was generated, was launched, see [Schneider \(2014\)](#).

Gamma-rays are defined as Photons with an Energy above 100 keV, see [Longair \(2011\)](#). The most common source of this radiation on earth is the radioactive decay. But these high energy photons are also emitted in space. These emissions originate from different processes, such as  $\pi^0$ -decay, inverse Compton scattering, synchrotron radiation and bremsstrahlung of ultra-relativistic charged particles.

### 1.2.1 Photomeson Production

The pion-decay is an elementary particle phenomenon which results from the inelastic collision of highly relativistic protons with soft photons or nonrelativistic protons. In the proton-proton case "[m]ore than 50% of the energy released goes into neutrinos, the remainder going into electrons or positrons and gamma rays[...]" [Sikora et al. \(1987\)](#). Additionally, one has to consider pair production and photo-pion production in the case of the proton-photon interaction. The photo-pion production may also be responsible for high-energy cut-offs in cosmic rays, see [Sikora et al. \(1987\)](#).

If the energy of a photon is high enough and it is in vicinity of a massive particle such as a nucleus, it is possible for the photon to convert into an electron positron pair, other particle antiparticle pairs are also possible, under the condition that the energy is higher than the rest mass of the produced pair. In this event, the energy is converted to mass according to Einstein's law and the nucleus receives repulsion. This has to happen near a massive particle so that momentum is conserved. The newly formed electron-positron-pair is now able to take part in other radiation processes such as annihilation or synchrotron-radiation, see [Beckmann and Shrader \(2012\)](#).

### 1.2.2 Synchrotron Radiation

Synchrotron radiation is emitted when charged particles, like the electrons mentioned beforehand, are accelerated in a magnetic field. The force produced by the field is applied perpendicular to the moving direction of the charged particle. This Process



can be deduced for a uniform and static magnetic field using the Lorentz-force. For relativistic purposes the Lorentz-factor also has to be considered.

$$\gamma = \frac{1}{\sqrt{1 - \frac{\mathbf{v} \cdot \mathbf{v}}{c^2}}} \quad (1.2)$$

Here the charge is given by  $q = Ze$ .

$$m \frac{d}{dt}(\gamma \mathbf{v}) = \frac{Ze}{c}(\mathbf{v} \times \mathbf{B}) \quad (1.3)$$

The left-hand side of the equation can be simplified by executing the derivative.

$$m\gamma \frac{d\mathbf{v}}{dt} = \frac{Ze}{c}(\mathbf{v} \times \mathbf{B}) \quad (1.4)$$

The velocity-vector of the particle can be split into its parallel  $\mathbf{v}_{\parallel}$  and perpendicular  $\mathbf{v}_{\perp}$  parts, respective to the magnetic field. It can be seen that the velocity parallel  $\mathbf{v}_{\parallel}$  to the field does not change. However, the perpendicular velocity  $\mathbf{v}_{\perp}$  results in an acceleration upright to plane of the field  $\mathbf{B}$  and the velocity  $\mathbf{v}$ , see [Longair \(2011\)](#).

$$\gamma m \frac{d\mathbf{v}}{dt} = ze v_{\perp} |\mathbf{B}| (\hat{e}_v \times \hat{e}_B) = ze |\mathbf{v}_{\perp}| |\mathbf{B}| \sin\theta (\hat{e}_v \times \hat{e}_B) \quad (1.5)$$

"Thus, the motion of the particle consists of a constant velocity along the magnetic field direction and circular motion with radius  $r$  about it, that is, a spiral path with constant pitch angle  $\theta$ ." [Longair \(2011\)](#). The overall luminosity of this process is given by the equation 1.6.

$$L = \frac{2Z^2 e^2}{3c^2} \gamma^4 \left[ \left( \frac{d\mathbf{v}_{\perp}}{dt} \right)^2 + \gamma^2 \left( \frac{d\mathbf{v}_{\parallel}}{dt} \right)^2 \right] \quad (1.6)$$

### 1.2.3 Inverse Compton Scattering

In contrast to the synchrotron radiation, inverse Compton scattering does not produce photons. However, it increases the photons energy by scattering processes. For example, if a low-energy/frequency photon hits an ultra-relativistic electron the photon is able to receive energy from the electron. In the frame of the relativistic electron the photon energy is considerably smaller than that of the electron. Because of this the process can be seen as Thomson scattering. This concludes that the energy of the photon does not change in the electrons frame. Transforming this back into the laboratory frame shows that the photon energy increases in proportion to the square of the Lorentz factor 1.2, see [Beckmann and Shrader \(2012\)](#).

### 1.2.4 Relativistic Beaming

The relativistic beaming, already mentioned in 1.1.3, that follows from superluminal motion is already linked to the relativistic jets and their  $\gamma$ -radiation observed in Blazars. This effect results in a blueshift of the emitted radiation and an increase of the radiated intensity, see Longair (2011). However, these observations can be explained by geometry and do not contradict with the postulates of special relativity. The following equations 1.7, 1.8 and explanation from Longair (2011) show that the transverse velocity, of an object emitted by the jet, can be larger than the speed of light ( $v_{\perp} > c$ ).

Let us assume that at point O an object is ejected at an angle  $\Theta$  to the line of sight of an observer at the distance D. The object holds a relativistic velocity. This happens at time  $t_0$  and the emitted signal arrives at the time  $t = \frac{D}{c}$ . Also beginning at  $t_0$  the object travels the distance  $vt_1$  and is observed at a projected distance  $vt_1\sin\Theta$ . The signal delivering this information to the observer arrives at the time

$$t_2 = t_1 + \frac{D - vt_1\cos\Theta}{c} \quad (1.7)$$

Since the signal now has to travel a shorter distance the transverse speed of the signal is

$$v_{\perp} = \frac{vt_1\sin\Theta}{t_2 - t_1} = \frac{vt_1\sin\Theta}{t_1 - \frac{vt_1\cos\Theta}{c}} = \frac{v\sin\Theta}{1 - \frac{v\cos\Theta}{c}} \quad (1.8)$$

By differentiating  $v_{\perp}$  for  $\Theta$  we find that the maximum transverse speed can be found at  $\Theta = \frac{v}{c}$ . This means that  $v_{\perp,max} = \gamma v$  where  $\gamma$  is the Lorentz-factor. This proves that  $v_{\perp} \geq c$  is true for high enough velocities( $v$ ).

### 1.2.5 Detection-Methods

There are several ways to detect the gamma radiation, one of the most common ones on earth is an ionisation chamber which is used in Geiger counters to detect ionizing radiation, e.g. from radioactive decay. To observe the radiation emanating from the beforehand discussed processes other methods have to be used. Since the atmosphere shields the earth from extraterrestrial gamma radiation direct observation of gamma-rays has to take place outside of the atmosphere. Indirect measurements of the photons are possible by observing air showers. These air showers originate from photons entering the atmosphere and colliding with particles resulting in more particles capable of scattering processes. There are two viable detection methods, either the generated particles are detected with particle detectors or the Cherenkov-radiation resulting from the higher velocity of the particles in comparison to the local light velocity, can be observed via optical telescopes such as the H.E.S.S and the M.A.G.I.C., see Schneider (2014).

The direct observation of the gamma radiation takes place on satellites like the Fermi/LAT, which will be discussed in the following section.

### 1.3 Fermi Large Area Telescope

The Fermi  $\gamma$ -ray Space Telescope was launched by NASA in 2008. Its purpose is to observe the sky for gamma-radiation and to measure direction, energy and arrival time of gamma-rays in a wide field of view and in an energy range from below 20 MeV to more than 300 GeV.

Since gamma-radiation can neither be reflected nor refracted, the measurement uses the pair-production of highly energetic photons near massive particles as discussed in 1.2.1. To do so a precision converter track, a calorimeter and an anticoincidence detector are used as measuring devices. Below the composition of the device will be briefly discussed, see [Atwood et al. \(2009\)](#).

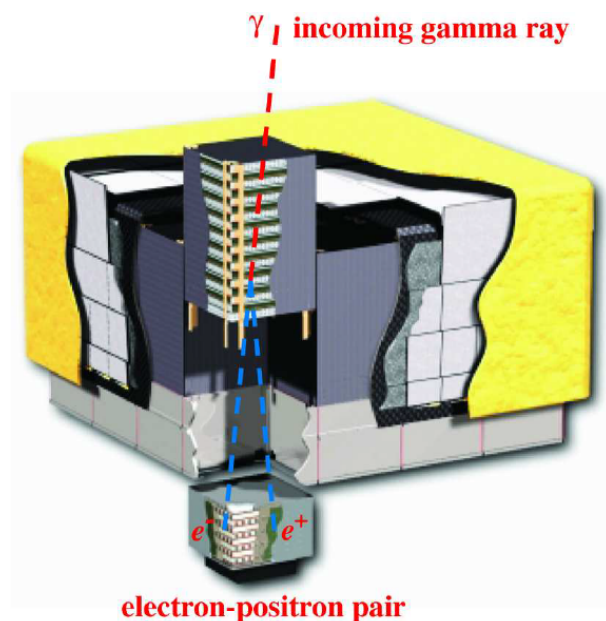


FIGURE 1.5: Schematic diagram of the Large Area Telescope. Credit: [Atwood et al. \(2009\)](#)

#### 1.3.1 Converter-Tracker

The converter-tracker is a device partially made of the high-Z material tungsten which is aimed to favour the pair production of the impacting  $\gamma$ -rays. After the pair conversion position sensitive detectors record the path of the charged particles through the tracker. With this recording the direction of the incoming photon can be reconstructed.

### 1.3.2 Calorimeter

The calorimeter serves two purposes, the first one is to measure the energy deposition of the electromagnetic particle showers following the photon impact and the second one is to map the shower development. The calorimeter is composed of caesium iodide which is used as a scintillator and photodiodes mounted on the end of the crystals to measure the scintillation light arriving at the end of the crystal.

### 1.3.3 Anticoincidence Detector

The  $\gamma$ -ray photons are not the only particles that can initiate a "detection"-process in the telescope. To suppress these false positives, produced by other particles, the anticoincidence detector (ACD) is used. It is supposed to detect charged particles with an efficiency of at least 0.9997. The ACD is composed of plastic scintillators the resulting light is then collected by wavelength shifting fibres which are coupled with two photomultipliers for redundancy. [Atwood et al. \(2009\)](#)

### 1.3.4 Mode of Operation

The Fermi/LAT is now continuously monitoring in an all sky scanning mode for over a decade while it orbits at an altitude of 565km and an inclination of  $25.5^\circ$ . Over this time it produced a vast amount of data including many different sources like galaxies, neutron stars, pulsars and AGN. The data collected of the AGN's is the quintessence of this bachelor's thesis and will be discussed thoroughly in the following chapters.

## 1.4 Goal of this Work

The question emerging from the observation of the Blazar-variability phenomenon is: Do the flux variations follow a particular pattern or are they random, i.e. of stochastic nature?

To solve this question the plan is to analyse the observed flux variations employing two statistical algorithms, accounting for the fact that the data were obtained with a flux-limited experiment.

The first is the Bayesian-Block-Algorithm developed by [Scargle et al. \(2013\)](#). The second is the Dominance-Algorithm which has been developed in this thesis project.

## Chapter 2

# Observational Data and Methods

### 2.1 Analysed Data and Coding Setup

The Data gained from the Fermi-spacecraft is processed by the instrument teams and then listed in the fits-files. These fits-files include all important data concerning the sources. In this case the used file consists of over two thousand light curves with monthly binning over 10 years and an amplitude over 1GeV. Every light curve consists of 119 measurements and quality of every single data point is defined by a test-statistic value associated with that data point. <sup>1</sup>

This test-statistic value is used to filter the pool of light curves for robust sources and is set so that a data point is accepted as significant enough if its test-statistic is above 9 which resembles a 3-sigma confidence threshold. To be used in the evaluation at least 50% of the light curves data points have to be accepted as significant. Following from this only 184 out of 2278 sources are selected for the final analysis. All necessary data is stored in an attached repository DVD.

All of the code and graphics were produced using the python (3.6.7) language and corresponding software packages such as numpy (1.16.2), astropy (3.0.5), matplotlib (3.0.2). [Virtanen et al. \(2019\)](#)

### 2.2 Bayesian-Block-Algorithm

The Bayesian-Block-Algorithm (BBA) discussed in [Scargle et al. \(2013\)](#) was used to parameterise the light curves in order to find real peaks in the data. Hereinafter this algorithm and its implementation in the rise and decay analysis will be discussed. An

---

<sup>1</sup>[https://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone\\_Likelihood/Likelihood\\_overview.html](https://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone_Likelihood/Likelihood_overview.html) [12.2.2020]

example of the Bayesian blocks on a light curve can be seen in figure 2.1.

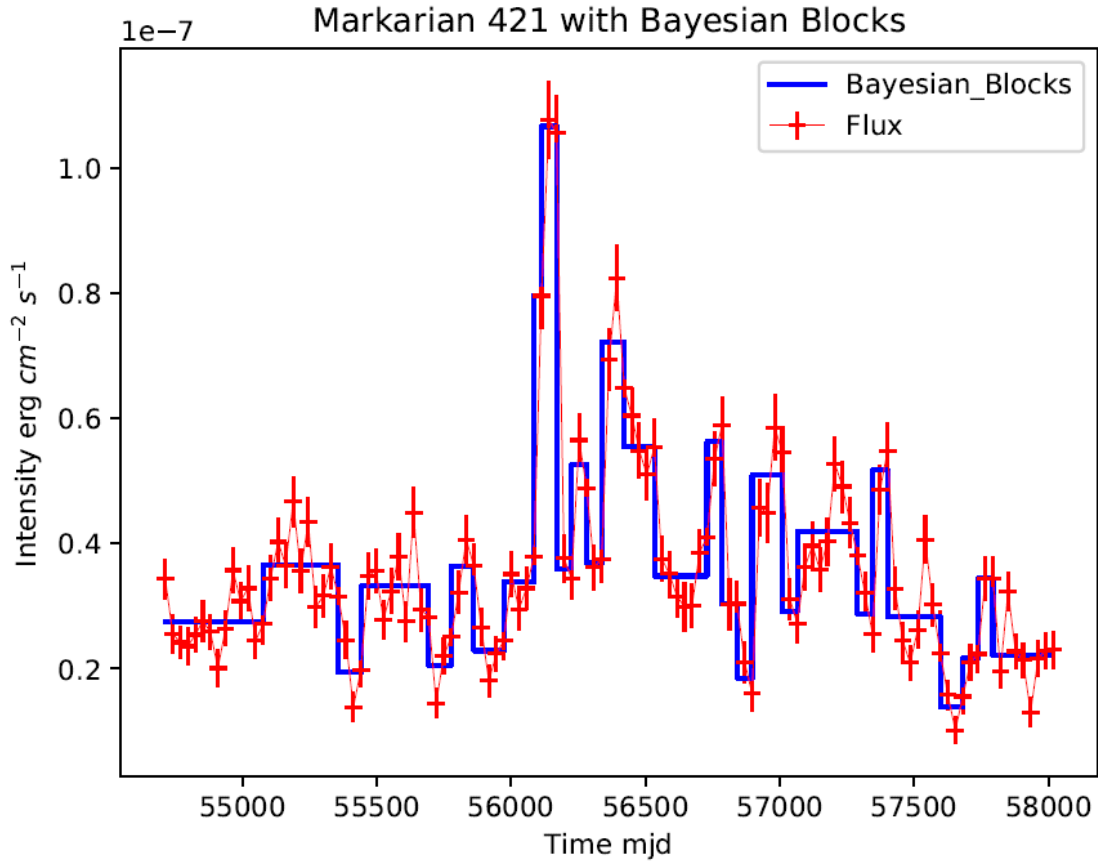


FIGURE 2.1: Light curve of Markarian 421 with a monthly binned data set over 10 years and Bayesian block parameterisation. The vertical extent of the flux points represents the error.

The orientation of the algorithm lies on a nonparametric analysis of time series data and it is able to analyse the data retrospectively and in real time. This happens with so-called change point detection which separates the data into a number of blocks, beginning and ending at a change-point. [Scargle et al. \(2013\)](#) defines, in regard to time series literature, a change point as a point at which a statistical model undergoes an abrupt transition. Because of the fact that a block always starts and ends with a change point there cannot be any overlapping. Within these blocks, which are defined over a time period, the intensity of the light curve is set to be constant in range of its errors, this is called a piecewise constant model. However, the beginnings and endings of a data set have to be treated differently, this will also be the case for the DA, the first point of time cannot be a free parameter, since it always has to be the start point of the first block and the first change point is the start of the second block. If the last data point is considered to be a change point it will result in a single data block. In the case of gaps in the data, a light curve is simply treated as if the data point after the gap follows right after the one before the gap. The expected number of blocks is defined via a geometric

prior, using only a single parameter  $\gamma$ .

$$P(N_{blocks}) = P_0 \gamma^{N_{blocks}} \quad (2.1)$$

After normalising 2.1 the expected number of blocks follows as 2.2

$$\langle N_{blocks} \rangle = P_0 \sum_{N_{blocks}=0}^N N_{blocks} \gamma^{N_{blocks}} = \frac{N\gamma^{N+1} + 1}{\gamma^{N+1} - 1} + \frac{1}{1 - \gamma} \quad (2.2)$$

In conclusion,  $\gamma$  can be considered a free parameter whose alteration can differ the structure of the Bayesian blocks.

This piecewise constant modelling is done by finding the optimal partition of a given time interval. Partitions are a summary of the number of blocks  $N_{blocks}$  and the corresponding edges defined by the indices of the data cells. In this case every cell consists of one measurement. To be able to use the BBA in the python environment it has to be imported via `astropy`. However, this implementation only generates the edges of the Bayesian blocks. In order to plot and analyse the light curves the intensity of the curve between two edges has to be determined and is then added to an array. For every block 3 different values were appended into separate arrays, the mean value, the minimum value and the maximum value. Due to the fact that the BBA parameterises the light curve every occurring turning point of the parameterisation is of significance. In order to find the maxima and minima and their rise and decay times, the dominance algorithm with a threshold of 1 will be used. That simply implies that every found turning point is accepted as one, meaning that every block surrounded by one left and one right of lower intensity is a maximum and every block surrounded by higher ones is a minimum, see Figure 2.2. The turning point is then defined as the middle of the block, if its width is uneven the middle will be rounded down, this follows from the discreteness of the data. The direction of the rounding is simply by choice and has no impact on the statistics, it simply shifts it one to the right or to the left. Following from this the rise and decay times arise from the distance between the previously defined maxima and minima. Self-explanatory the rise time is from minimum to maximum and the decay time vice versa. There is still a problem with the endings and the beginnings of a light curve, since we cannot know what was before and after the measurement it is not possible to determine the rise and decay times at the edges properly. A possible solution would be to neglect the first and the last peak in order to get rid of the flawed data, but by doing so light curves with only one peak would then also be neglected. The code to this explanation can be found in the appendix A.

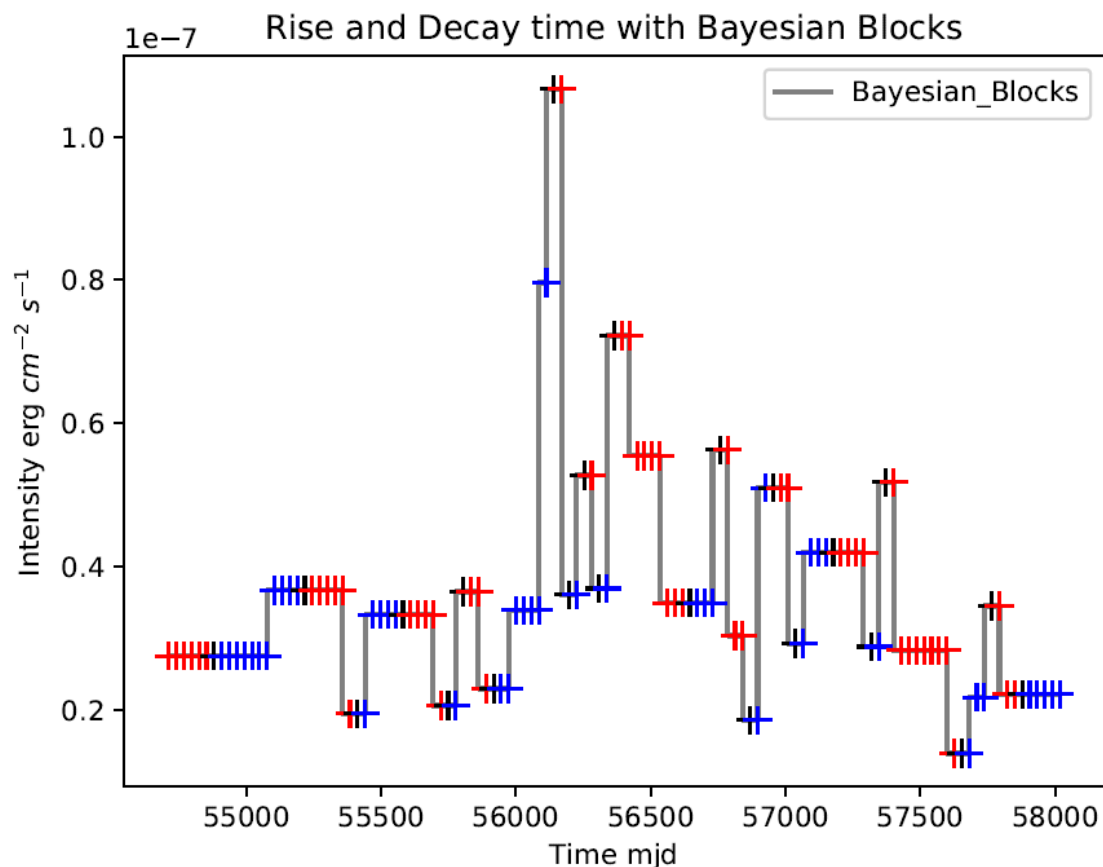


FIGURE 2.2: Bayesian block parameterisation of the Markarian 421 light curve and peaks defined by the dominance algorithm. The peaks are coloured in black, the rise in blue and the decay in red. The normal data points of the light curve have been neglected in this plot for better visibility of the rise and decay times.

## 2.3 Dominance-Algorithm

The Dominance-Algorithm (DA) is inspired by the topographic isolation of mountains used in geography. The DA analyses every data point and calculates the number of steps it takes to get to a measurement with higher intensity, the corresponding code snippet is discussed in listing A.1.

```

1 | lefts = []
2 | rights = []
3 | for i in range(len(data)):
4 |     if len(np.where(data[0:i]>data[i])[0]) == 0:
5 |         left_dominance = i-1
6 |     elif np.max(np.where(data[0:i]>data[i])[0]) == i-1:
7 |         left_dominance = 0
8 |     else:
9 |         left_dominance = i-np.max(np.where(data[0:i]>data[i])[0])
10 | if len(np.where(data[i+1:]>data[i])[0]+i+1) == 0:
11 |     right_dominance = (119-i)
12 | elif np.min(np.where(data[i+1:]>data[i])[0]+i+1) == i+1:
13 |     right_dominance = 0

```



```

14 |     else:
15 |         right_dominance = np.min(np.where(data[i+1:]>data[i])[0]+i+1)-(i+1)
16 |         lefts.append(left_dominance)
17 |         rights.append(right_dominance)

```

LISTING 2.1: This code snippet elaborates the process of finding the distance to the next point with a higher intensity. Here data is a list of all intensities of a light curve. The np.where() commands are used to determine the dominance for left and right and all of it happens in an "in-range"-function to analyse all data points. The results are then appended into the corresponding list.

By doing so every point in time gets a dedicated distance value for left and right, the lower one of both then determines the dominance. Because it will always take at least one step to get to the next measurement and also the time resolution is discrete, 1 is the least possible dominance. Following from this a threshold under which a maximum will not be accepted as one can easily be given. Through simple mirroring of the light curve on the time-axis the DA can also be used to find minima with little to no extra effort. After determining the maxima, this information is used to generate an array with the same length as the intensity array. This array consists of two different values ones and threes, the numbers were arbitrarily chosen. The threes simply hold the information at which point a maximum resides in the array. Following from this, the array is used to define the intervals between two maxima. These Intervals are then searched for their minimum. The array is then adapted and the minimum-information is imbedded as zeros. With the help of this array a second one is constructed following the same number code but with the extra information of rise and decay by defining all rise-points as ones and all decay-times as twos. That gives us an array with all the necessary data as underlying structure to the normal intensity array. Using this structure, it's not only possible to easily colour code plots of the light curves but also to determine the rise and decay times of their corresponding maximum. This can also be seen in figure 2.3.

From there on it is the same procedure as in the section before, the rise time is the interval from minimum to maximum and the decay time the interval from maximum to minimum, and is calculated via the array mentioned in the passage before. In the end the 3 most vital values of a light curve are saved in a 3-dimensional numpy array. The first dimension is for the maximum and its value sitting at the right position in the time interval. The second one gives us the rise and sits at the same position as its corresponding maximum and the last does the same but for the decay. This is mostly done by using the numpy.where function and can be seen in the listing A.2.

```

1 | final_data = np.zeros((3,119))
2 | #final_data[0] == maxima
3 | #final_data[1] == rise
4 | #final_data[2] == decay
5 | for i in range(len(np.where(rise_decay_time == 3)[0])):
6 |

```

```

7  final_data[0][np.where(rise_decay_time==3)[0][i]]=data[np.where(rise_decay_time
   ==3)[0][i]]
8
9  final_data[1][np.where(rise_decay_time==3)[0][i]]=np.where(rise_decay_time==3)
   [0][i]-
10 np.where(rise_decay_time==0)[0][np.argmax(np.where(np.where(rise_decay_time==0)
   [0]<
11 np.where(rise_decay_time==3)[0][i])[0])]
12
13 final_data[2][np.where(rise_decay_time==3)[0][i]]=np.where(rise_decay_time==0)
   [0]
14 [(np.where(np.where(rise_decay_time==0)[0]>
15 np.where(rise_decay_time==3)[0][i])[0])[0]]-np.where(rise_decay_time==3)[0][i]

```

LISTING 2.2: This code snippet elaborates the process of calculating the rise and decay and assigning them to their maximum. This is done via the `rise_decay_time` array which holds the information to which group a point belongs to in a number code. With the help of `numpy.where` functions the flux value is assigned to its maximum and the rise and decay are calculated by finding the distance to the surrounding minima.

However, there is still a problem with the beginning and the end of plots. If the first turning point of a light curve is a local maximum the rise time can not be properly identified. This is due to the fact that we cannot know how the source behaved before the measurement. The same problem occurs if the last turning point is also a maximum, in this case, however, the decay time cannot be identified. There are two possible ways to solve this, the first one is to neglect the maximum and its rise and decay all together and the second one is to accept the data as the minimal possible rise or decay time of such a maximum. The second one was chosen due to the fact that there are only small data samples with monthly binning. If e.g. a source with only one peak occurs the source in its entirety would be neglected. An example of the DA can be seen in figure 2.3.

The code for this algorithm can be found in the appendix A.

The beforehand discussed threshold gives a parameter to neglect possible unwanted maxima, this helps filtering out only supposed significant turning points because some may be little peaks in the rise or the decay time of vast outbursts. The alteration of the limit may change the amount of turning points and their ascents and descents in single sources, but in summary it can be stated that all of the given data the statistical outcome stays the same with a sole exception. If the limit is set to high, for example half of the width of a light curve, there is a major chance that the algorithm will find no peak at all. This results from the edges of our data set, the DA cuts off at the borders and defines this distance as the dominance. In conclusion the threshold was set to allow every point with a dominance higher or equal 6 as a maximum ( $th \geq 6$ ). However, this also leads to the problem of peaks residing too near to the edge of a light curve. If a maximum is less than 6 "steps" away from the edge, the DA is not able to identify it

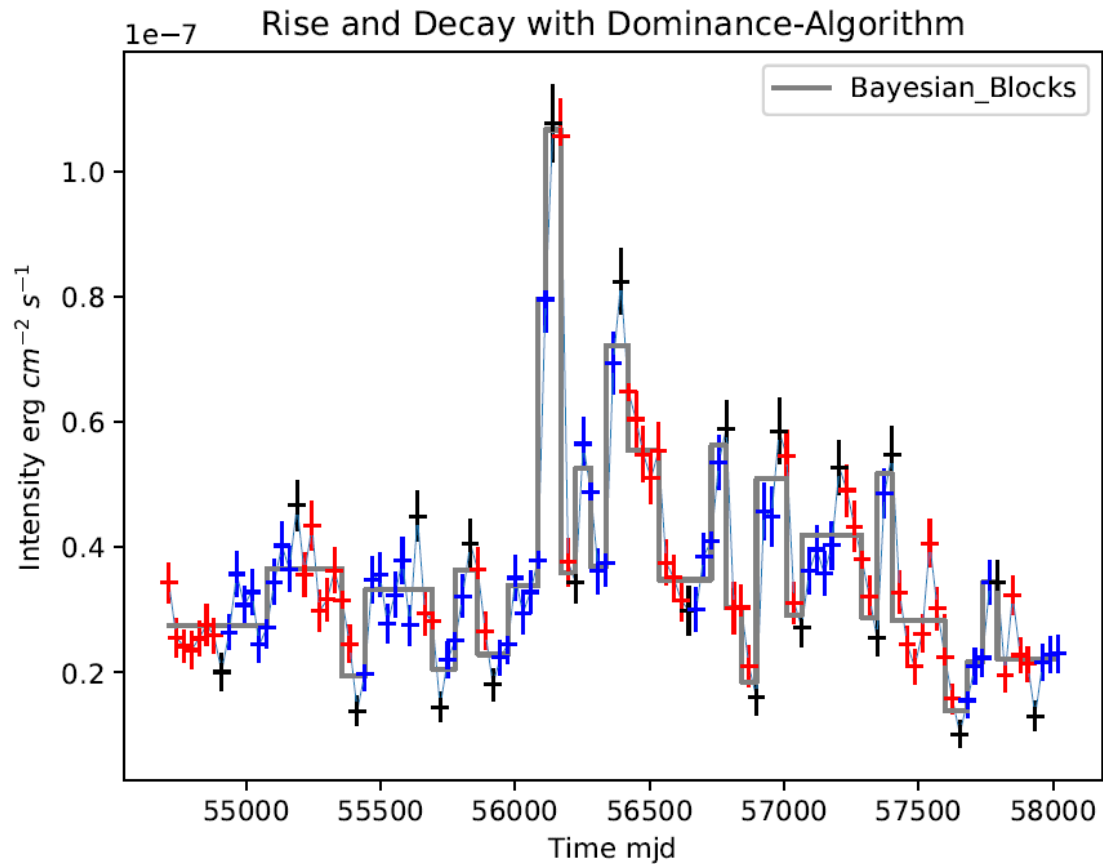


FIGURE 2.3: Light curve of Markarian 421 with monthly binned data set over 10 years and a graphical implementation of the dominance-algorithm. Here the red markings represent the decay and the blue markings the rise. The corresponding extreme points are black. The grey lines are Bayesian block parameterisation. The vertical extent of the flux points represents the error.

since it can only count the dominance until it hits the edge, an example of this can be seen in Figure 2.4. This will be thoroughly discussed in the upcoming section. Figure 2.5 shows the threshold and the dominance of every data point.

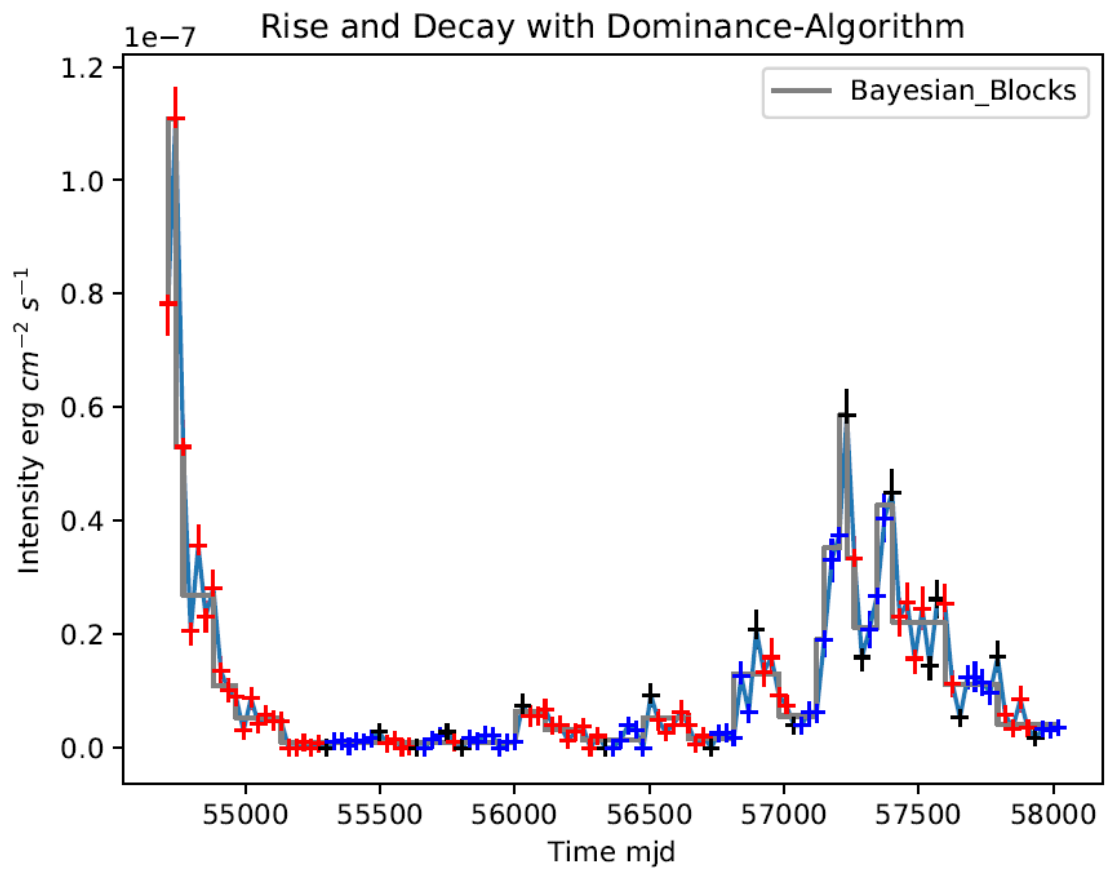


FIGURE 2.4: Dominance of source 3FGL\_J0238.6+1636 clearly shows the edge problems of the DA, since the peak cannot be identified as one because of its closeness to the edge. The colour coding is the same as 2.3.

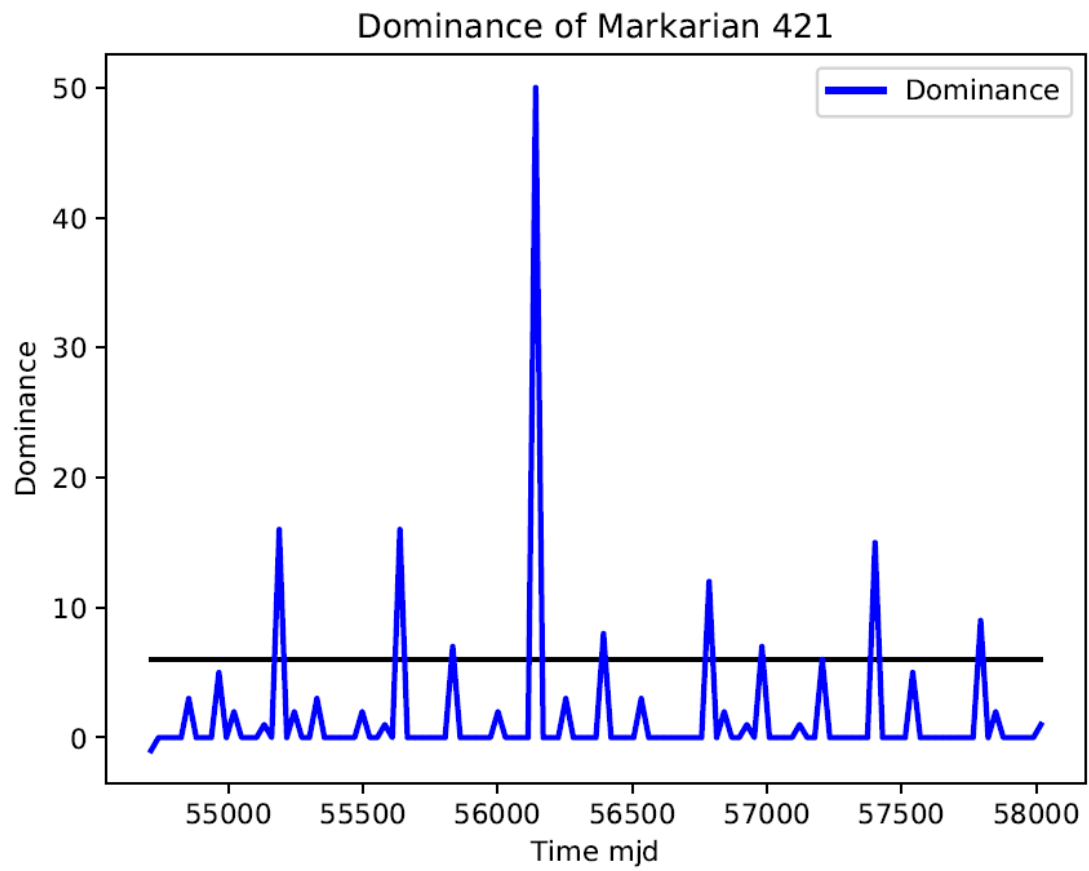


FIGURE 2.5: Dominance of Markarian 421 plotted against the time. The black line symbolises the threshold ( $th \geq 6$ ).



## Chapter 3

# Results

The light curves of all sources are analysed with the two algorithms and the corresponding rise and decay times are illustrated with histograms. Just by looking at the plots in figure 3.1 and 3.2 the resemblance of rise and decay from the DA and from the BBA strikes out. To check this assumption a Kolmogorov-Smirnov-test (KS-test) is deployed to see if the probability measures of both histograms coincide, see [Hartung et al. \(1993\)](#). In both cases the percentaged acceptance outnumbers the 5% threshold by far. Hence, we can assume that the rise and decay time both follow the same statistic, leading to the conclusion that most flares should be symmetric or that at least there should be no bias regarding their shape. However, this only refers to the width of a flare and not to the shape of it. Even though the BBA and DA histograms slightly resemble each other, their KS-test shows clearly that they do not follow the same statistic. The rise and decay may still be different in shape. Also, one could argue that this algorithm does not take into account that a flare might also be in a constant high or low and the algorithm only considers point like peaks. Assuming that the peak also has a width would complicate the algorithm and would surely result in different rise and decay times. The remaining question is, would that change the statistics or the symmetry of said times?

This assumption is partially in agreement with the findings of [Sokolov et al. \(2004\)](#). They modelled multifrequency Blazar emissions with rapid variability. According to their paper the main amount of emission is produced through synchrotron radiation (SR). The higher energy radiation occurs through inverse Compton scattering of the (SR). They claim that the main reason for flares is the collision between a relativistic shock wave travelling through the jet and a stationary feature in the jet. This clash leads to forward and reverse shocks. They also found out that their simulated high-frequency light curves peak at about half of the time it takes the shock to traverse the jet and all of them appear to be symmetric for a wider range of frequencies. This symmetry apparently directly follows from the geometry of the excitation region along the shock

front.

[Bhatta and Dhital \(2019\)](#) also found symmetric behaviour in 20 Blazars, they analysed weekly binned data and computed the percental rise and decay rates per day and came to the same conclusion, that the rates are symmetric. They got to that conclusion by also deploying the KS-test.



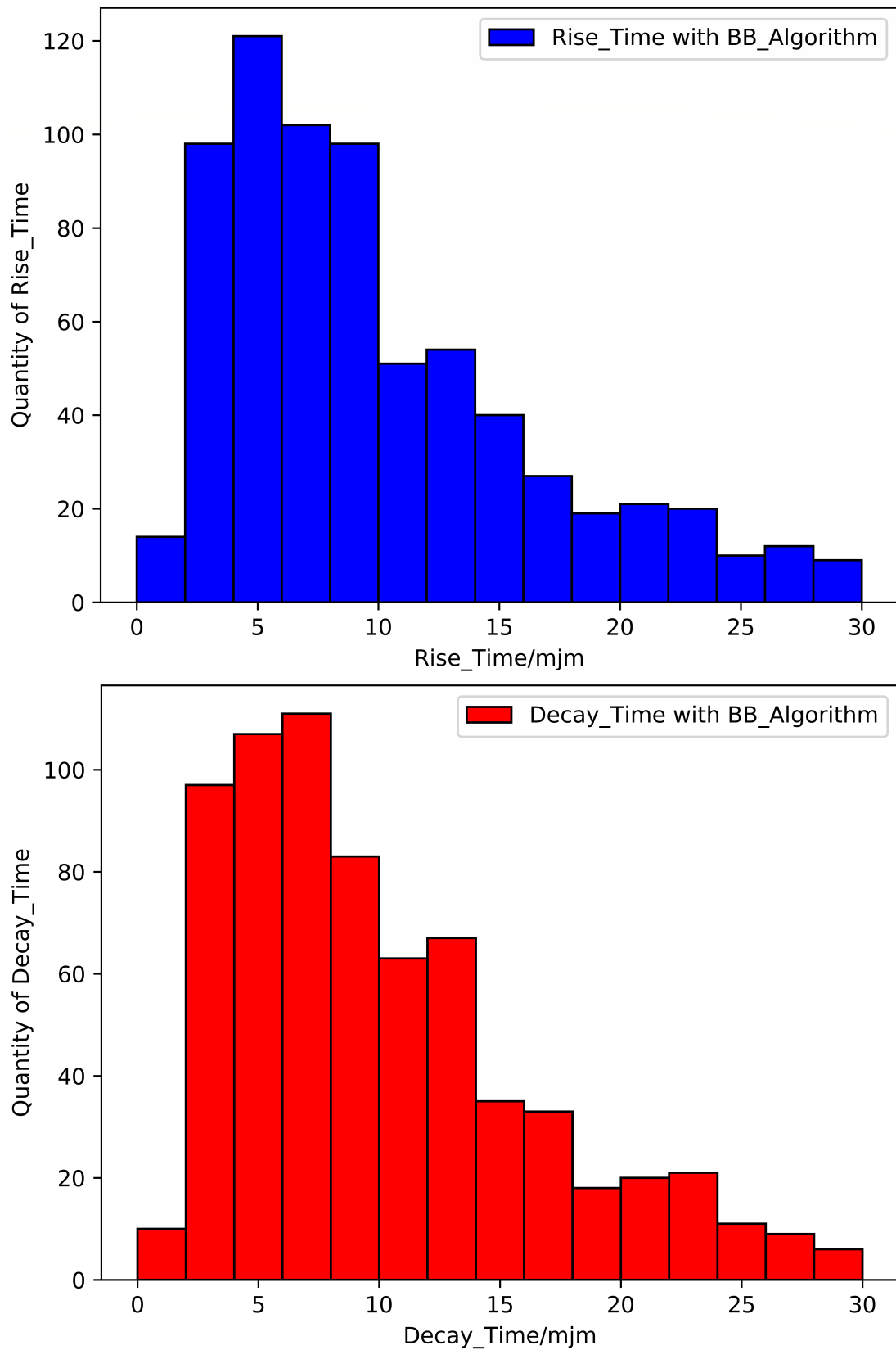


FIGURE 3.1: Histograms of all rise and decay times determined by an application of the dominance algorithm on the Bayesian-block parameterisation. The KS-Test for the two histograms yields  $p = 0,774$ , concluding that the two statistics are not distinguishable. This makes a strong argument about the bias of the flares, meaning that there might be none.

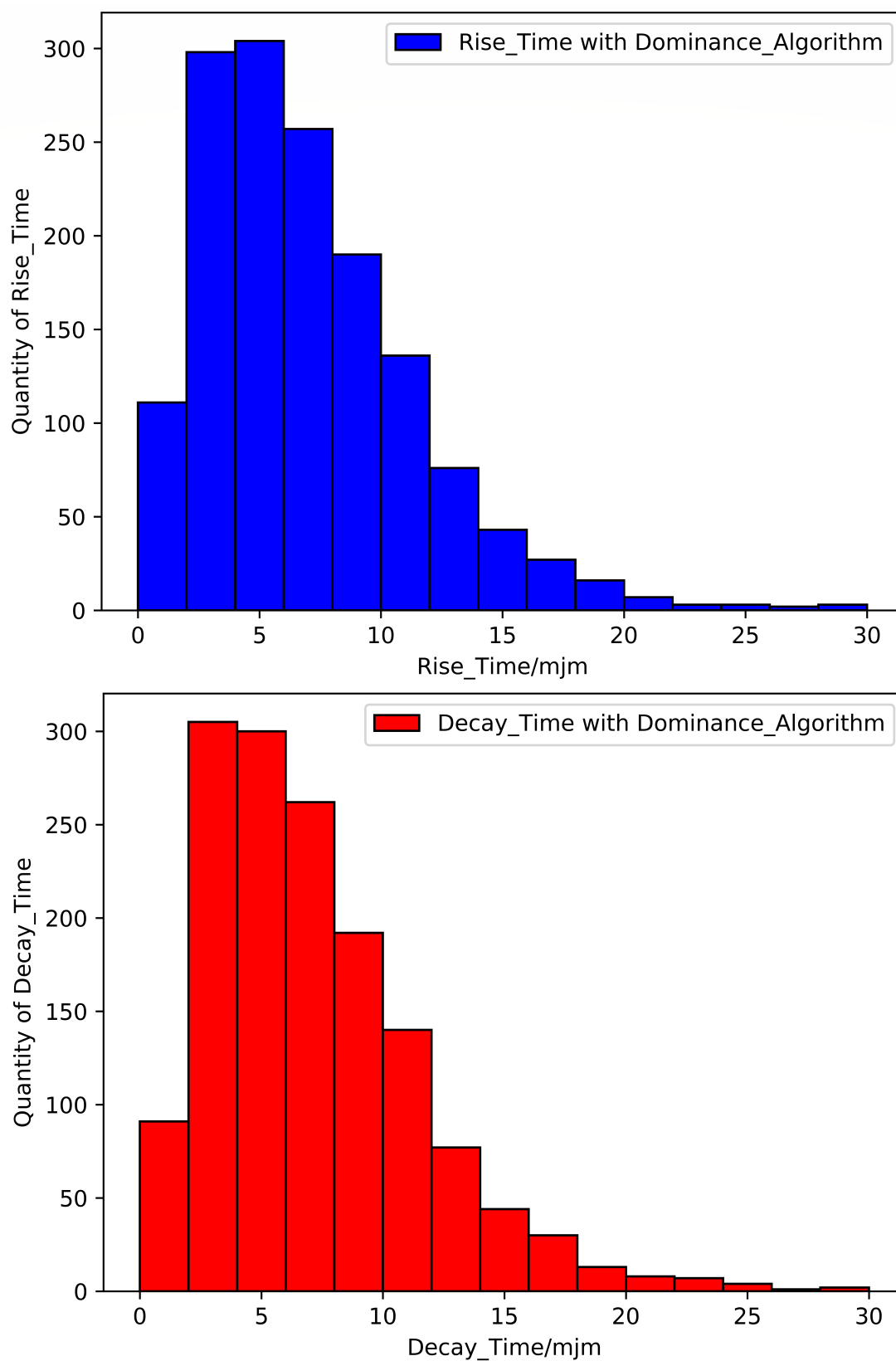


FIGURE 3.2: Histograms of all rise and decay times determined by an application of the dominance algorithm on the unaltered light curve. The KS-Test for the two histograms yields  $p = 0,796$ , concluding that the two statistics are not distinguishable. This makes a strong argument about the bias of the flares, meaning that there might be none.

## Chapter 4

# Conclusion and Outlook

The focus of this Thesis lies in the statistical properties of the rise and decay times of the flux variations in the gamma-ray light curves of Blazars. The light curves of more than two thousand sources observed with the Fermi/LAT were analysed. The observation took place over a time interval of ten years and every flux measurement was averaged over one month in this interval. Each of these measurements comes with a test-statistic value describing the significance. With the help of this parameter the pool of sources was filtered for their overall significance. This shrinks the quantity of sources approximately by a factor of ten. The light curves were then analysed with the Bayesian-Block-Algorithm (BBA) and the Dominance-Algorithm (DA). The BBA parameterises the light curve and the DA checks the curve for extreme points. The DA is employed on the parameterised and the normal curve, the only adjustable parameter in this algorithm is a threshold value defining a dominance value as a threshold for the acceptance of a peak as a maximum. Via the determined maxima the respective rise and decay times are defined. The statistical analysis of rise and decay shows that there is no bias concerning the shape of a light curve. This result is consistent for the normal light curves and for the parameterised ones.

However, this method can still be improved, since the DA, so to speak, only deals in extremes. There are only maxima, minima and rise and decay times, this leaves no room for possible non active intervals or constant intervals in a maximum. If it is assumed that a flaring state has two different states, a non-active one, where the flux is the lowest, and an active one, where the flux is the highest, it cannot be neglected that there is a high chance rise and decay times do not exactly begin and end at the first encountered extreme point. If such intervals are existent, they are simply ignored. Also, the identification of rise and decay times at the edges of light curves is not properly possible. The effect of this problem can be reduced with wider data sets, e.g. daily binned light curves. If non-active and highly active interval states are considered the

rise and decay problem could be diminished. A higher resolution of the data could also help reduce edge effects. The main problem is that the DA can not work intrinsically on every set of light curves, because all follows from the arbitrarily chosen threshold at the beginning of the algorithm.

# Bibliography

Albert, J., Aliu, E., Anderhub, H., Antoranz, P., Armada, A., Baixeras, C., Barrio, J. A., Bartko, H., Bastieri, D., Becker, J. K., Bednarek, W., Berger, K., Bigongiari, C., Biland, A., Bock, R. K., Bordas, P., Bosch-Ramon, V., Bretz, T., Britvitch, I., Camara, M., Carmona, E., Chilingarian, A., Coarasa, J. A., Commichau, S., Contreras, J. L., Cortina, J., Costado, M. T., Curtef, V., Danielyan, V., Dazzi, F., De Angelis, A., Delgado, C., de los Reyes, R., De Lotto, B., Domingo-Santamaría, E., Dorner, D., Doro, M., Errando, M., Fagiolini, M., Ferenc, D., Fernández, E., Firpo, R., Flix, J., Fonseca, M. V., Font, L., Fuchs, M., Galante, N., García-López, R. J., Garczarczyk, M., Gaug, M., Giller, M., Goebel, F., Hakobyan, D., Hayashida, M., Hengstebeck, T., Herrero, A., Höhne, D., Hose, J., Hrupec, D., Hsu, C. C., Jacon, P., Jogler, T., Kosyra, R., Kranich, D., Kritzer, R., Laille, A., Lindfors, E., Lombardi, S., Longo, F., López, J., López, M., Lorenz, E., Majumdar, P., Maneva, G., Mannheim, K., Mansutti, O., Mariotti, M., Martínez, M., Mazin, D., Merck, C., Meucci, M., Meyer, M., Miranda, J. M., Mirzoyan, R., Mizobuchi, S., Moralejo, A., Nieto, D., Nilsson, K., Ninkovic, J., Oña-Wilhelmi, E., Otte, N., Oya, I., Paneque, D., Panniello, M., Paoletti, R., Paredes, J. M., Pasanen, M., Pascoli, D., Pauss, F., Pegna, R., Persic, M., Peruzzo, L., Piccioli, A., Prandini, E., Puchades, N., Raymers, A., Rhode, W., Ribó, M., Rico, J., Rissi, M., Robert, A., Rügamer, S., Saggion, A., Saito, T., Sánchez, A., Sartori, P., Scalzotto, V., Scapin, V., Schmitt, R., Schweizer, T., Shayduk, M., Shinozaki, K., Shore, S. N., Sidro, N., Sillanpää, A., Sobczynska, D., Stamerra, A., Stark, L. S., Takalo, L., Tavecchio, F., Temnikov, P., Tescaro, D., Teshima, M., Torres, D. F., Turini, N., Vankov, H., Vitale, V., Wagner, R. M., Wibig, T., Wittek, W., Zandanel, F., Zanin, R., and Zapatero, J. (2007). Variable Very High Energy  $\gamma$ -Ray Emission from Markarian 501. *The Astrophysical Journal*, 669(2):862–883.

Antonucci, R. (1993). Unified models for active galactic nuclei and quasars. *Annual Review of Astronomy and Astrophysics*, 31:473–521.

Atwood, W. B., Abdo, A. A., Ackermann, M., Althouse, W., Anderson, B., Axelsson, M., Baldini, L., Ballet, J., Band, D. L., Barbiellini, G., and et al. (2009). The large

- area telescope on the fermi gamma-ray space telescope mission. *The Astrophysical Journal*, 697(2):1071–1102.
- Beckmann, V. and Shrader, C. (2012). *Active Galactic Nuclei*. Active Galactic Nuclei. Wiley.
- Bhatta, G. and Dhital, N. (2019). Nature of  $\gamma$ -ray variability in blazars. *arXiv e-prints*, page arXiv:1911.08198.
- Blandford, R. D. and Königl, A. (1979). Relativistic jets as compact radio sources. *The Astrophysical Journal*, 232:34–48.
- Blandford, R. D. and McKee, C. F. (1977). Radiation from relativistic blast waves in quasars and active galactic nuclei. *Monthly Notices of the Royal Astronomical Society*, 180:343–371.
- Blandford, R. D. and Payne, D. G. (1982). Hydromagnetic flows from accretion disks and the production of radio jets. *Monthly Notices of the Royal Astronomical Society*, 199:883–903.
- Blandford, R. D. and Znajek, R. L. (1977). Electromagnetic extraction of energy from Kerr black holes. *Monthly Notices of the Royal Astronomical Society*, 179:433–456.
- Ghisellini, G., Righi, C., Costamante, L., and Tavecchio, F. (2017). The fermi blazar sequence. *Monthly Notices of the Royal Astronomical Society*, 469(1):255–266.
- Hartung, J., Elpelt, B., and Klösener, K. (1993). *Statistik: Lehr- und Handbuch der angewandten Statistik*. Oldenbourg.
- Königl, A. (1981). Relativistic jets as X-ray and gamma-ray sources. *The Astrophysical Journal*, 243:700–709.
- Longair, M. (2011). *High Energy Astrophysics*. Cambridge University Press.
- Mannheim, K. (1993). The proton blazar. *Astronomy and Astrophysics*, 269:67–76.
- Norman, M. L., Winkler, K. H. A., Smarr, L., and Smith, M. D. (1982). Structure and dynamics of supersonic jets. *Astronomy and Astrophysics*, 113:285–302.
- Scargle, J. D., Norris, J. P., Jackson, B., and Chiang, J. (2013). The Bayesian Block Algorithm. *arXiv e-prints*, page arXiv:1304.2818.
- Schneider, P. (2014). *Extragalactic Astronomy and Cosmology: An Introduction*. Springer Berlin Heidelberg.

- Sikora, M., Kirk, J. G., Begelman, M. C., and Schneider, P. (1987). Electron Injection by Relativistic Protons in Active Galactic Nuclei. *Astrophysical Journal, Letters*, 320:L81.
- Sokolov, A., Marscher, A. P., and McHardy, I. M. (2004). Synchrotron Self-Compton Model for Rapid Nonthermal Flares in Blazars with Frequency-dependent Time Lags. *The Astrophysical Journal*, 613(2):725–746.
- Tanaka, Y., Nandra, K., Fabian, A. C., Inoue, H., Otani, C., Dotani, T., Hayashida, K., Iwasawa, K., Kii, T., Kunieda, H., Makino, F., and Matsuoka, M. (1995). Gravitationally redshifted emission implying an accretion disk and massive black hole in the active galaxy MCG-6-30-15. *Nature*, 375(6533):659–661.
- Ulvestad, J. S., Wong, D. S., Taylor, G. B., Gallimore, J. F., and Mundell, C. G. (2005). VLBA Identification of the Milliarcsecond Active Nucleus in the Seyfert Galaxy NGC 4151. *Astronomical Journal*, 130(3):936–944.
- Urry, C. M. and Padovani, P. (1995). Unified Schemes for Radio-Loud Active Galactic Nuclei. *Publications of the Astronomical Society of the Pacific*, 107:803.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . . (2019). SciPy 1.0—Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, page arXiv:1907.10121.

## *Acknowledgements*

I want to thank Dr. Karl Mannheim for giving me the opportunity to work on this fascinating subject. I'd also like to thank Dr. Sara Buson for providing me with the Data set from the Fermi/LAT without whom this thesis would not have been possible. This bachelor thesis would not have been possible without the help of Paul Ray Burd. He introduced me to this subject and always supported me with valuable advice and provided me with guidance when it was needed.

I'd also like to thank my fellow students in the group of Dr. Mannheim's group for the active discussions of light curves which broadened my understanding and my perspective of the matter.

At last I want to thank the people closest to me, my mother Jutta, my father Claus, my Brother Felix and my love Marlene. Without their unconditional support none of this would have been possible and I am deeply grateful for that.



# Declaration of authorship

I, Max Knopf, declare that this thesis titled, 'Rise and Decay Time of Blazars' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

---

Date:

---



# Appendices



# Appendix A

## Python-Code

```
1 import sys
2 import numpy as np
3 from numpy import inf
4 from astropy.io import fits
5 from astropy.time import Time
6 import matplotlib.pyplot as plt
7 import matplotlib.colors as colors
8 import astropy.stats.bayesian_blocks as bblocks
9 from matplotlib.backends.backend_pdf import PdfPages
10
11 df = fits.open('MonthlyLC1GeV_cleaned_fixSpec_sourceListAll2283.fits')
12
13
14 #Bayesian-Blocks
15
16 def time(s):
17     return (df[1].data[s][3]-df[1].data[s][2])/2+df[1].data[s][2];
18
19 widths = []
20 dominance_rise = []
21 dominance_decay = []
22 bb_rise = []
23 bb_decay = []
24
25 for s in range(len(df[1].data)):
26
27     ts = []
28     test_statistic = df[1].data[s][23]
29
30     for i in range(len(test_statistic)):
31         if test_statistic[i] <= 9:
32             ts.append(test_statistic[i])
33
34     if len(ts) < 60:
35
36
```

```

37     edges=bblocks(t=time(s), x=df[1].data[s][51], sigma=df[1].data[s][44],
fitness='measures')
38         #print(edges)
39
40     position=[]
41     for j in range(len(edges)):
42         position.append(np.min(np.where(time(s)>=edges[j])))
43     position=np.array(position)
44
45     mean_values=[]
46     mini_values=[]
47     maxi_values=[]
48     width_test =[]
49
50     for j in range(len(position)-1):
51
52         if j == len(position)-2:
53             mean_values.append(np.mean(df[1].data[s][51][position[j]:]))
54             mini_values.append(np.min(df[1].data[s][51][position[j]:]))
55             maxi_values.append(np.max(df[1].data[s][51][position[j]:]))
56         else:
57             mean_values.append(np.mean(df[1].data[s][51][position[j]:position
[j+1]]))
58             mini_values.append(np.min(df[1].data[s][51][position[j]:position[
j+1]]))
59             maxi_values.append(np.max(df[1].data[s][51][position[j]:position[
j+1]]))
60             widths.append(position[j+1]-position[j])
61             width_test.append(position[j+1]-position[j])
62
63     mean_values=np.array(mean_values)
64     mini_values=np.array(mini_values)
65     maxi_values=np.array(maxi_values)
66
67     mean=np.zeros(119)
68     mini=np.zeros(119)
69     maxi=np.zeros(119)
70     for k in range(len(mean_values)):
71         if k == len(mean_values)-1:
72             mean[position[k]:] = mean_values[k]
73             mini[position[k]:] = mini_values[k]
74             maxi[position[k]:] = maxi_values[k]
75         else:
76             mean[position[k]:position[k+1]] = mean_values[k]
77             mini[position[k]:position[k+1]] = mini_values[k]
78             maxi[position[k]:position[k+1]] = maxi_values[k]
79
80     bb_lefts = []           # left dominances
81     bb_rights = []        # right dominances
82     for i in range(len(mean_values)):
83         if len(np.where(mean_values[0:i]>mean_values[i])[0]) == 0:
84             if i-1 == 0:
85                 left_dominance = 1
86             else:
87                 left_dominance = i-1

```

```

88         elif np.max(np.where(mean_values[0:i]>mean_values[i])[0]) == i-1:
89             left_dominance = 0
90         else:
91             left_dominance = i - np.max(np.where(mean_values[0:i]>mean_values
92 [i])[0])
93         if len(np.where(mean_values[i+1:]>mean_values[i])[0] + i+1) == 0:
94             right_dominance = (len(mean_values)-i)
95         elif np.min(np.where(mean_values[i+1:]>mean_values[i])[0] + i+1) == i
96 +1:
97             right_dominance = 0
98         else:
99             right_dominance = np.min(np.where(mean_values[i+1:]>mean_values[i
100 ])[0] + i+1) - (i+1)
101
102         bb_lefts.append(left_dominance)
103         bb_rights.append(right_dominance)
104
105     # determine maxima from lefts and rights
106     bb_max_min = []
107     for i in range(len(mean_values)):
108         if bb_lefts[i] >= 1 and bb_rights[i] >= 1:
109             bb_max_min.append(3)
110         else:
111             bb_max_min.append(1)
112
113     # finding minima between two maxima
114     # max = 3
115     # min = 0
116
117     bb_max_min = np.array(bb_max_min)
118     bb_find_max = np.zeros((2,119))
119     for i in range(len(np.where(bb_max_min == 3)[0])):
120         bb_find_max[0][np.where(bb_max_min == 3)[0][i]] = 3
121         bb_find_max[1][np.where(bb_max_min == 3)[0][i]] = np.where(
122 bb_max_min == 3)[0][i]
123     bb_max_id = []
124     bb_interval_between_max = []
125
126     for i in range(len(bb_find_max[0])):
127         if i == 0:
128             bb_max_id.append(0)
129         elif not bb_find_max[1][i] == 0:
130             bb_max_id.append(bb_find_max[1][i])
131         bb_max_id.append(len(mean_values))
132     for i in range(len(bb_max_id)-1):
133         bb_interval_between_max = mean_values[int(bb_max_id[i]):int(
134 bb_max_id[i+1])]
135         bb_max_min[int(bb_max_id[i]+np.argmin(bb_interval_between_max))]
136 = 0
137
138     bb_rise_decay = np.zeros((3,119))
139     for i in range(len(bb_rise_decay[0])):
140         if bb_rise_decay[0][i] == 0:
141             bb_rise_decay[0][i] = 1

```

```

137
138     for i in range(len(bb_max_min)):
139         if bb_max_min[i] == 3:
140             bb_rise_decay[0][int(np rint((np.sum(width_test[0:i]))+(
width_test[i]/2)))-1] = 3
141         elif bb_max_min[i] == 0:
142             bb_rise_decay[0][int(np rint((np.sum(width_test[0:i]))+(
width_test[i]/2)))-1] = 0
143
144     bb_rise_decay_time = []
145     rise_decay_bb = -1
146     # 1 = rise
147     # 2 = decay
148     # 0 = min
149     # 3 = max
150     for i in range(len(bb_rise_decay[0])):
151         if bb_rise_decay[0][i] == 0:
152             rise_decay_bb = 1
153             bb_rise_decay_time.append(0)
154         elif bb_rise_decay[0][i] == 3:
155             rise_decay_bb = 2
156             bb_rise_decay_time.append(3)
157         else:
158             bb_rise_decay_time.append(rise_decay_bb)
159
160
161     for i in range(len(bb_rise_decay_time)):
162         if bb_rise_decay_time[i] == 0:
163             bb_rise_decay_time[0:i] = [2]*(i)
164             break
165         elif bb_rise_decay_time[i] == 3:
166             bb_rise_decay_time[0:i] = [1]*(i)
167             break
168     bb_rise_decay_time = np.array(bb_rise_decay_time)
169
170     bb_final_data = np.zeros((3,119))
171     # bb_final_data[0] == maxima
172     # bb_final_data[1] == rise
173     # bb_final_data[2] == decay
174     for i in range(len(np.where(bb_rise_decay_time == 3)[0])):
175         if len(np.where(np.where(bb_rise_decay_time == 0)[0]<np.where(
bb_rise_decay_time == 3)[0][i])[0]) == 0:
176             bb_final_data[1][np.where(bb_rise_decay_time == 3)[0][i]] = 0
177         else:
178             bb_final_data[0][np.where(bb_rise_decay_time == 3)[0][i]] = mean[
np.where(bb_rise_decay_time == 3)[0][i]]
179             bb_final_data[1][np.where(bb_rise_decay_time == 3)[0][i]] = np.
where(bb_rise_decay_time == 3)[0][i]-np.where(bb_rise_decay_time == 0)[0][np.
argmax(np.where(np.where(bb_rise_decay_time == 0)[0] < np.where(
bb_rise_decay_time == 3)[0][i])[0])]
180             bb_final_data[2][np.where(bb_rise_decay_time == 3)[0][i]] = np.
where(bb_rise_decay_time == 0)[0][(np.where(np.where(bb_rise_decay_time == 0)
[0] > np.where(bb_rise_decay_time == 3)[0][i])[0])[0]]-np.where(
bb_rise_decay_time == 3)[0][i]
181     for i in range(len(bb_final_data[0])):

```



```

182         if not bb_final_data[1][i] == 0:
183             bb_rise.append(bb_final_data[1][i])
184         if not bb_final_data[2][i] == 0:
185             bb_decay.append(bb_final_data[2][i])
186
187     # Dominance-Algorithm
188
189     data = df[1].data[s][51]
190
191     lefts = []           # left dominances
192     rights = []        # right dominances
193
194     for i in range(len(data)):
195         if len(np.where(data[0:i]>data[i])[0]) == 0:
196             left_dominance = i-1
197         elif np.max(np.where(data[0:i]>data[i])[0]) == i-1:
198             left_dominance = 0
199         else:
200             left_dominance = i - np.max(np.where(data[0:i]>data[i])[0])
201
202         if len(np.where(data[i+1:]>data[i])[0] + i+1) == 0:
203             right_dominance = (119-i)
204         elif np.min(np.where(data[i+1:]>data[i])[0] + i+1) == i+1:
205             right_dominance = 0
206         else:
207             right_dominance = np.min(np.where(data[i+1:]>data[i])[0] + i+1) -
208 (i+1)
209
210         lefts.append(left_dominance)
211         rights.append(right_dominance)
212
213     # determine maxima from lefts and rights
214
215     max_min = []
216     for i in range(len(data)):
217         if lefts[i] >= 6 and rights[i] >= 6:
218             max_min.append(3)
219         else:
220             max_min.append(1)
221
222     # finding minima between two maxima
223     # max = 3
224     # min = 0
225
226     max_min = np.array(max_min)
227     find_max = np.zeros((2,119))
228     for i in range(len(np.where(max_min == 3)[0])):
229         find_max[0][(np.where(max_min == 3)[0][i])] = 3
230         find_max[1][(np.where(max_min == 3)[0][i])] = np.where(max_min == 3)
231 [0][i]
232     max_id = []
233     interval_between_max = []
234
235     for i in range(len(find_max[0])):

```

```

235         if i == 0:
236             max_id.append(0)
237         elif not find_max[1][i] == 0:
238             max_id.append(find_max[1][i])
239     max_id.append(len(data))
240     for i in range(len(max_id)-1):
241         interval_between_max = data[int(max_id[i]):int(max_id[i+1])]
242         max_min[int(max_id[i]+np.argmax(interval_between_max))] = 0
243     rise_decay_time = []
244     rise_decay = -1
245     # 1 = rise
246     # 2 = decay
247     # 0 = min
248     # 3 = max
249     for i in range(len(max_min)):
250         if max_min[i] == 0:
251             rise_decay = 1
252             rise_decay_time.append(0)
253         elif max_min[i] == 3:
254             rise_decay = 2
255             rise_decay_time.append(3)
256         else:
257             rise_decay_time.append(rise_decay)
258
259
260     for i in range(len(rise_decay_time)):
261         if rise_decay_time[i] == 0:
262             rise_decay_time[0:i] = [2]*(i)
263             break
264         elif rise_decay_time[i] == 3:
265             rise_decay_time[0:i] = [1]*(i)
266             break
267     rise_decay_time = np.array(rise_decay_time)
268     final_data = np.zeros((3,119))
269     # final_data[0] == maxima
270     # final_data[1] == rise
271     # final_data[2] == decay
272     for i in range(len(np.where(rise_decay_time == 3)[0])):
273         final_data[0][np.where(rise_decay_time == 3)[0][i]] = data[np.
where(rise_decay_time == 3)[0][i]]
274         final_data[1][np.where(rise_decay_time == 3)[0][i]] = np.where(
rise_decay_time == 3)[0][i]-np.where(rise_decay_time == 0)[0][np.argmax(np.
where(np.where(rise_decay_time == 0)[0] < np.where(rise_decay_time == 3)[0][i
]) [0])]
275         final_data[2][np.where(rise_decay_time == 3)[0][i]] = np.where(
rise_decay_time == 0)[0][np.where(np.where(rise_decay_time == 0)[0] > np.
where(rise_decay_time == 3)[0][i])[0]][0]-np.where(rise_decay_time == 3)[0][
i]
276
277     for i in range(len(final_data[0])):
278         if not final_data[1][i] == 0:
279             dominance_rise.append(final_data[1][i])
280         if not final_data[2][i] == 0:
281             dominance_decay.append(final_data[2][i])
282
283     #Plots

```

```

283
284     with PdfPages('./Rise_Decay/' + df[1].data[s][56] + '_' + str(s) +
    _ts_filtered.pdf') as pdf:
285
286         #Plot zur Dominanz
287
288         plt.plot((df[1].data[s][3]-df[1].data[s][2])/2+df[1].data[s][2], data)
289         plt.plot((df[1].data[s][3]-df[1].data[s][2])/2+df[1].data[s][2], mean,
    label='Bayesian_Blocks', c = 'grey')
290         for i in range(len(data)):
291             if rise_decay_time[i] == 1:
292                 plt.errorbar(x = (df[1].data[s][3][i]-df[1].data[s][2][i])/2+
    df[1].data[s][2][i], y = df[1].data[s][51][i], yerr=df[1].data[s][44][i],
    ecolord='B', elinewidth=1, linewidth=0.5, marker='+', c = 'B')
293                 elif rise_decay_time[i] == 2:
294                     plt.errorbar(x = (df[1].data[s][3][i]-df[1].data[s][2][i])/2+
    df[1].data[s][2][i], y = df[1].data[s][51][i], yerr=df[1].data[s][44][i],
    ecolord='R', elinewidth=1, linewidth=0.5, marker='+', c = 'R')
295                 else:
296                     plt.errorbar(x = (df[1].data[s][3][i]-df[1].data[s][2][i])/2+
    df[1].data[s][2][i], y = data[i], yerr=df[1].data[s][44][i], ecolord='k',
    elinewidth=1, linewidth=0.5, marker='+', c = 'k')
297                     plt.title('Rise and Decay with Dominance-Algorithm')
298                     plt.ylabel('amplitude erg/s')
299                     plt.xlabel('time mjd')
300                     plt.legend()
301                     pdf.savefig()
302                     plt.close()
303
304         #Plot zu BB
305         plt.errorbar(x = time(s), y = mean, linewidth=2, label='
    Bayesian_Blocks', marker='_', c = 'c')
306         plt.errorbar(x = time(s), y = df[1].data[s][51], yerr=df[1].data[s
    ][44], ecolord='grey', elinewidth=1, label='Flux', linewidth=0.1, marker='+', c =
    'grey')
307         for i in range(len(mean)):
308             if bb_rise_decay_time[i] == 3:
309                 plt.errorbar(x = time(s)[i], y = mean[i], ms=10, marker='+', c
    = 'black')
310             elif bb_rise_decay_time[i] == 0:
311                 plt.errorbar(x = time(s)[i], y = mean[i], ms=10, marker='+', c
    = 'black')
312             elif bb_rise_decay_time[i] == 2:
313                 plt.errorbar(x = time(s)[i], y = mean[i], ms=10, marker='+', c
    = 'red')
314             elif bb_rise_decay_time[i] == 1:
315                 plt.errorbar(x = time(s)[i], y = mean[i], ms=10, marker='+', c
    = 'blue')
316         plt.title('Rise and Decay with Bayesian_Blocks')
317         plt.ylabel('amplitude erg/s')
318         plt.xlabel('time mjd')
319         plt.legend()
320         pdf.savefig()
321         plt.close()
322

```

```

323
324 dominance_rise = np.array(dominance_rise)
325 np.save('dominance_rise', dominance_rise)
326 dominance_decay = np.array(dominance_decay)
327 np.save('dominance_decay', dominance_decay)
328 bb_rise = np.array(bb_rise)
329 np.save('bb_rise', bb_rise)
330 bb_decay = np.array(bb_decay)
331 np.save('bb_decay', bb_decay)

```

LISTING A.1: This code shows the process of finding the distance to the next point with a higher intensity. Data is a list of all intensities of a source. The `np.where()` commands are used to determine the dominance for left and right and all of it happens in an "in-range"-function to analyse all data points. The rise and decay of both DA and BBA are saved in numpy arrays see code [A.2](#).

```

1 df = fits.open('MonthlyLC1GeV_cleaned_fixSpec_sourceListAll12283.fits')
2
3 bb_decay = np.load('bb_decay.npy')
4 bb_rise = np.load('bb_rise.npy')
5 dominance_rise = np.load('dominance_rise.npy')
6 dominance_decay = np.load('dominance_decay.npy')
7
8 plt.hist(bb_rise, bins = np.arange(0,30+2,2), color = 'blue', label = 'Rise_Time
   with BB_Algorithm', edgecolor = 'k')
9 plt.ylabel('Quantity of Rise_Time')
10 plt.xlabel('Rise_Time/mjm')
11 plt.legend()
12 plt.savefig('Rise_Time with BB_Algorithm.png', dpi=1000)
13 plt.close()
14
15 plt.hist(bb_decay, bins = np.arange(0,30+2,2), color = 'red', label = 'Decay_Time
   with BB_Algorithm', edgecolor = 'k')
16 plt.ylabel('Quantity of Decay_Time')
17 plt.xlabel('Decay_Time/mjm')
18 plt.legend()
19 plt.savefig('Decay_Time with BB_Algorithm.png', dpi=1000)
20 plt.close()
21
22 plt.hist(dominance_rise, bins = np.arange(0,30+2,2), color = 'blue', label = '
   Rise_Time with Dominance_Algorithm', edgecolor = 'k')
23 plt.ylabel('Quantity of Rise_Time')
24 plt.xlabel('Rise_Time/mjm')
25 plt.legend()
26 plt.savefig('Rise_Time with Dominance_Algorithm.png', dpi=1000)
27 plt.close()
28
29 plt.hist(dominance_decay, bins = np.arange(0,30+2,2), color = 'red', label = '
   Decay_Time with Dominance_Algorithm', edgecolor = 'k')
30 plt.ylabel('Quantity of Decay_Time')
31 plt.xlabel('Decay_Time/mjm')
32 plt.legend()
33 plt.savefig('Decay_Time with Dominance_Algorithm.png', dpi=1000)
34 plt.close()

```

```
35 |  
36 | print(ks(dominance_rise, dominance_decay))  
37 | print(ks(bb_rise, bb_decay))  
38 | print(ks(bb_decay, dominance_decay))  
39 | print(ks(bb_rise, dominance_rise))
```

LISTING A.2: In this code the final data saved in code [A.1](#) is tested with the KS-Test and the histograms are plotted.