GRAVITATIONAL WAVES RESEARCH AT TOROS UTRGV

PAMELA IVONNE LARA

Master's Program in Physics

APPROVED:

Jorge A. López, Ph.D., Chair

Mario Diaz, Ph.D, Co-Chair

Marian Manciu, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

DEDICATION

I dedicate this thesis to my parents, Rene and Gladys Lara, and to Nathaniel Cowsert, my beloved nephew who left us too soon.

GRAVITATIONAL WAVES RESEARCH AT TOROS UTRGV

by

PAMELA IVONNE LARA, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Physics

THE UNIVERSITY OF TEXAS AT EL PASO

August 2018

ACKNOWLEDGEMENTS

I am deeply indebted to my advisor, Dr. Jorge A. Lopez for his guidance and support throughout my time at UTEP.

I also thank profusely my secondary advisor, Dr. Mario Diaz from UTRGV, for accepting me into TOROS at UTRGV.

I also acknowledge support from my TOROS UTRGV teammates, Dr. Martin Beroiz, Richard Camuccio, Moises Castillo, and Juan Garcia; I thank all of them for welcoming me into the team and for the Python lessons.

Finally, I thank UTEP for a hell of a ride.

ABSTRACT

The detection of gravitational waves (GW) has directly opened a new era in the observation of cosmic events. One hundred years after its theoretical prediction we find ourselves immerged in the multi-messenger study of the signals at the root of gravitational wave detection. The electromagnetic (EM) counterpart to GW is the optical portion of that signal and the main objective in the organization of TOROS Collaboration: finding and studying kilonovas, the name given by Metzger (Metzger et al, 2010), to the EM counterpart to gravitational waves.

In order for TOROS to find kilonovas, it needed to create a python language pipeline. One of the steps of the pipeline was the design of an image calibration method script, also written in python language: Imageredux. In this thesis the author gives an account of her part on the design of such script and its application to the field of HAT-P-27 (images captured on the night of June 22, 2017), as a test subject. Also, and account of TOROS Collaboration detection of the kilonova SSS17a found as a counterpart to GW170817 is given.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Gravitational waves are vibrations in space-time, the material from which the universe is made. In 1916, Albert Einstein recognized that, according to his General Theory of Relativity, the most violent bodies of the cosmos release part of their mass in the form of energy through these waves. He determined that the solutions of the linearized equations to weak field are transverse waves moving at the speed of light (Manreza, 2017). For many years it was thought that these space-time waves were just a theoretical result and they were never going to be able to be detected. However, on September 14 of the year 2015 the first direct detection of a gravitational wave was made.

In general, a wave is an oscillation that transports energy. The waves can be mechanical if the oscillation propagates in a medium, or electromagnetic if they are oscillations of electric and magnetic fields due to the accelerated movement of electric charges. In particular, gravitational waves are produced by accelerated movement of masses, which cause fluctuations in the curvature of space-time, that is, they are periodic deformations of space-time that affect the relative distances between points of space. Gravitational waves are obtained as solutions of Einstein's equations

$$G_{\mu\nu} = kT_{\mu\nu} \tag{1}$$

where the left side is the Einstein tensor, which is a derivative function of the metric tensor ($g_{\mu\nu}$), which describes the curvature of space-time. In the right side is the energy-momentum tensor, which describes the distribution of matter and $k = \frac{8\pi G}{c^4}$ is a constant dependent of the gravitational constant G and speed of the light c. In general, equations (1) are a set of ten non-linear equations partial derivatives, whit known solutions for some cases. To obtain gravitational waves Einstein made some approximations. First, he considered the solution of equations in the

vacuum, that is, he took $T_{\mu\nu} = 0$, and ignored the non-linear effects, so that the metric could be written as $g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu,}$ being $\eta_{\mu\nu}$ the metric of the flat space and $h_{\mu\nu}$ a small disturbance to this space ($|h_{\mu\nu}| \ll 1$). Thus, equations (1) are reduced to

$$\Box h_{\mu\nu} = 0, \tag{2}$$

where $\Box = -\frac{\partial^2}{\partial t^2} + \nabla^2$ is D'Alembert operator. Equation (2) is a wave equation whose typical solution (in the case of a wave that travels in the z-direction), has the shape

$$h_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_x & 0 \\ 0 & h_x & h_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

With $h_+ \; and \; h_x$ the only non-zero components that correspond to the possible polarizations ($+ \; y \; \times$) of the wave, respectively. The effect of the passage of this wave on a ring of test masses is to contract and expand the distances in directions orthogonal, as shown in Figure 1.2. Therefore, we can characterize gravitational waves as waves traveling at the speed of light with two polarizations and whose amplitude is determined by the magnitude dimensionless $h = 2\frac{\Delta L}{L}$ where L is the length between two test masses and ΔL are the variations of that length. Unlike electromagnetic waves, for which the dominant term in the emission comes from the dipolar moment, for the gravitational waves the dipole moment is zero (a consequence of there being no negative and positive masses), which is why the quadrupole term dominates. In this way the amplitude of gravitational waves is proportional to the second derivative of the quadrupole of

mass of the source. This implies that there are no distributions of spherically symmetrical matter that emit gravitational waves (Saulson, 1994).

While all the accelerated movements of asymmetric mass emit gravitational waves, being the gravitational interaction very weak, there is the need for large masses performing very fast movements to generate disturbances in the space-time for them to be measurable. This means that the main sources of gravitational waves (Manreza, 2017), with potential to be detectable are:

- Binary systems of black holes, neutron stars, or a combination of both: as the orbital radius of the binary system decreases as both compact objects approach, gravitational waves are emitted.
- Explosions of stars in the form of Supernovas: if the explosion is asymmetric, large amounts of matter are ejected at high speeds, which produces gravitational radiation.
- Rotating neutron stars: although in general the rotating neutron stars are objects spherically symmetrical, in some cases the star may be deformed and therefore emits gravitational radiation.
- Inflation: many cosmological models postulate that at very early stages of the evolution of the universe, this expanded in an accelerated way. If the expansion was asymmetrical, it could have emitted gravitational waves.

1.1 BRIEF HISTORY OF GRAVITATIONAL WAVES DETECTION

A background history on the topic needs to start with the French mathematician Henri Poincaré who, on July 5 of 1905, published “Sur la Dynamique d’ l’electron” (On the dynamics of electrons), and where he presented his thoughts about relativity. On this paper, Poincaré suggested that “gravity was transmitted through a gravitational wave” (Cervantes-Cota, 2015). A decade later Einstein’s General Theory of Relativity was published.

For Einstein, gravity was not a force of attraction between object, a Newtonian postulate, but a distortion in the fabric of space and time; as one of his later collaborators nicely put it *"Spacetime tells matter how to move; matter tells spacetime how to curve." (*Wheeler*, 2000).* Because Einstein himself was not very assured that gravitational waves could be detected and because of the amount of polemic awakened by the subject, he continued working, either to prove their existence or to finally declare that they were not plausible. During the 1930s, Einstein and Nathan Rosen, one of his collaborators, revealed a gravitational wave solution that did not correspond to a physically possible situation.

On June 1, 1936, Einstein and Rosen presented the paper (to the journal Physical Review), *Are There Gravitational Waves?*, however, they later withdrew it (Daniel, 2005). Suddenly it appeared that gravitational waves did not exist. Corresponding with his friend Max Born, Einstein wrote “Together with a young collaborator, I arrive at the interesting result that gravitational waves do not exist, though they have been a certainty to the first approximation” (Born-Einstein, 2005). Finding calculation errors in his work with Rosen, Einstein made the needed corrections to the galley proof and rewrote his paper under the title *On Gravitational*

Waves (Daniel, 2005), and kept his beliefs on the existence of gravitational waves (Cervantes-Cota, 2015).

Putting it in fundamental terms, the problem was in the coordinate system chosen by Einstein. A contemporary of his, the young British theoretical physicist Felix A. E. Pirani solved this problem. In 1956 Pirani published the article titled *On the Physical Significance of the Riemann tensor* (Pirani, 1956). In his paper, Pirani "settled on the notion that an observable effect of a passing gravitational wave is the undulating separation between two test masses in space (something gravitational physicists called "geodesic deviation" or "tidal deviation")" (Larson, 2016).

A milestone in relativity-gravitational wave's duo can be found in the 1957 Chapel Hill Conferences. After 19 months after the passing of Albert Einstein, 44 scientists met in North

Caroline for this critical conference. At this point, all physicist agreed on the fundamental need to show the effects that any gravitational phenomena would have in nature. One of the topics

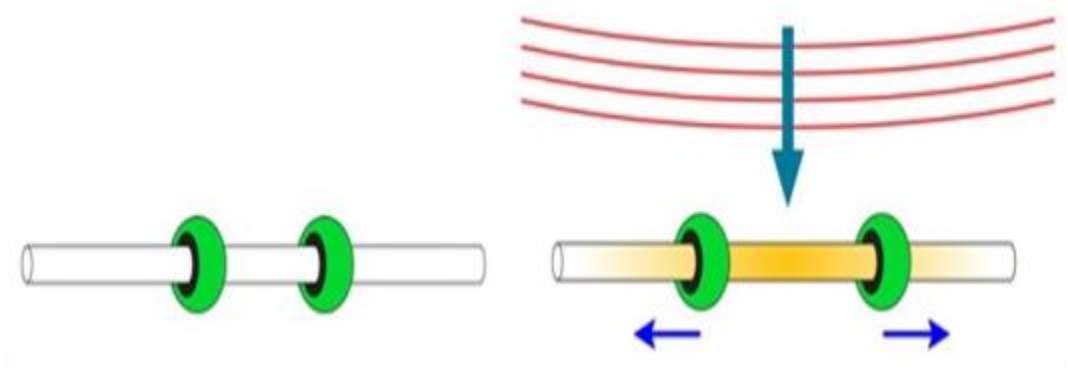

Figure 1.1: Sticky bead argument illustration

discussed there was whether or not a gravitational wave would carry energy. Among the attendees was Richard Feynman. Feynman expanded on Pirani's idea with his thought experiment "sticky bead argument." He imagined two freely-moving beads connected to a rod as shown in figure 1.1 (Chaudhuri, 2017). When the direction of wave propagation is perpendicular to the direction of the rod, the tidal wave generated (the longitudinal stresses, namely alternative compressions and extensions), will produce the motion of the beads relative to the stick. In the presence of friction, this relative motion will result in the production of heat (Chaudhuri, 2017). 'I think it is easy to see that if gravitational waves can be created, they can carry energy and can do work" (DeWitt, 2011). This heating due to friction represents the transmitted energy from the gravitational wave to the bar. When Feynman arguments were added to the arguments of other participants, a general consent was reached about the existence of gravitational waves.

A later point of discussion was about measuring the masses' displacement produced by the traveling wave. How can we measure the length of the displacement if any measuring tool we use would also be affected by the wave? Would be like trying to measure a distance using a rubber band. Years later this was solved by measuring, instead, the time taken by the light to travel the displaced distance since the speed of light is constant and unperturbed by the gravitational wave (Chaudhuri, 2017).

1.2 GRAVITATIONAL WAVES DETECTION METHODS

The first attempts to detect gravitational waves came from another 1957 Chapel Hill Conference attendee, the American physicist Joseph Weber. Weber decided to build a contraption capable of detecting gravitational radiation. After much research, he opted for a cylindrical bar, an antenna like resonance-mass detector of 2m in length by 0.5m in diameter made of aluminum, and monitored by piezoelectric transducers. Based on the knowledge that any solid has a natural oscillation frequency (mainly depending on the solid's size), if the bar was to be perturbed (by a small bump for example), it would respond with an oscillation, that would reach its maximum when the perturbation is attuned to the natural frequency of oscillation of the instrument. If a gravitational wave, with the same oscillation frequency as of the cylinder, traveled through the bar, this would vibrate with a quantifiable amplitude.

Figure 1.2 illustrates this effect on a ring of particles which is the foundation for resonant-mass gravitational wave detectors. The cylindrical bars would oscillate longitudinally as a gravitational wave travels through them and in a frequency resonant with the wave (Magalhães, 2005). Joseph Weber hoped to be able to measure these vibrations. Little did he know this was an impossible task for his experimental settings since the magnitude of the amplitude turned out to be of the order of 10^{-21}m (Saulson, 1994).

The significant limitations Weber encountered were the background noise (seismic waves for example), and not knowing the order of frequency of the gravitational waves. To attack these problems he built several cylinders of different sizes hoping to match that of the waves. In 1969 he reported that a couple of his bars had detected similar signals, “Six detectors have been operating. Four of them are aluminum cylinders of 153-cm length tuned to a narrow band of frequencies ($\Delta\omega = 0.1\, rad\, sec^{-1}$) near 1660 Hz. The bandwidth is adjustable. Piezoelectric

crystals bonded to their surfaces couple the normal-mode oscillations to an electromagnetic degree of freedom." (Weber, 1969). His results inspired other scientists to recreate his experiment.

After several of these efforts failed to produce acceptable results, the scientific community despaired in their hopes to confirm Einstein's predictions. Then in 1974 came the work of Hulse and Taylor (Hulse and Taylor, 1975). Theirs was not a direct detection but a confirmation of the existence of gravitational waves. For decades, they studied the pulses we receive from PSR B1913+163, a neutron star in a binary neutron star (BNS) system. The slow rate of period decrease of the pulses for this pulsar matched precisely the rate at which they were losing energy in gravitational waves, as predicted by General Relativity.

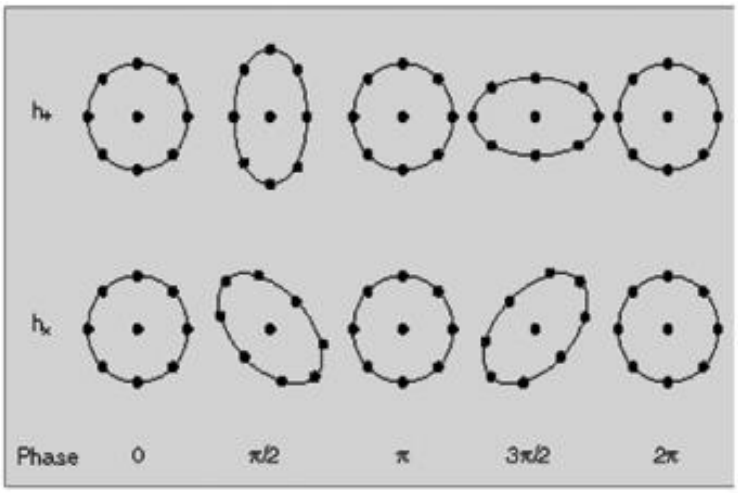

Figure 1.2: Resonance-mass detector effect on stationary particles. When a gravitational wave passes through a ring of particles, it changes their position depending on the wave's polarization. Top line, wave with polarization "+". Bottom line, wave with polarization "×".

The discovery and analysis awarded them the 1993 Nobel Prize in Physics "for the discovery of a new type of pulsar, a discovery that has opened up new possibilities for the study

of gravitation." This vital work also settled the debate on whether gravitational waves could carry energy with them. The decrease in period rate was proof that the energy was being radiated away as gravitational waves.

Other indirect methods of GW detection include the Pulsar Timing Array (PTA) technique. PTA searches for correlated delays in the pulse arrival times for an array of preselected millisecond pulsars. These delays can then be analyzed in search for signatures of passing GW. This type of projects aims to detect very low-frequency GW, from pHz to μHz. Potential sources for this very low-frequency signals are mergers of supermassive black hole binaries at the center of galaxies (Beroiz, 2017).

To look for a more direct method of observation, the scientific community started to delve into the principles of interferometry. Interferometry is a measuring method based on the interference property of merging waves (generally electromagnetic waves, radio waves, or sound waves). Interference is an undulatory phenomenon manifested when two or more waves propagate into the same region of space, and their contribution is added (constructively or destructively). When referring to electromagnetic waves, one could measure the distance, amid two separated objects for example, by determining the time the light takes in round-trip travel between the two objects. The Michelson and Morley experiment was the first to put to use the round-trip idea in their quest to identify the ether drift, or the relative motion between the light as it travels through a medium –ether-, (Michelson and Morley, 1887). They carried their experiment using what came to be known as the Michelson Interferometer.

A Michelson Interferometer is composed of a light source, a beam splitter (half silvered mirror), and two mirrors positioned at a distance with an 90° angle separation. When a light beam travels through the half-silvered mirror (which is a partially reflecting mirror), it splits into

two rays with different optical trajectories (the two mirrors); the rays bounce back recombining at the beam splitter and arriving at a detector (Figure 1.3). The alteration in the trajectory of the two rays generates a phase difference that forms an interference fringe pattern. Immediately after, the pattern is analyzed in the detector to evaluate the characteristics of the wave, the properties of the material, or the displacement of one of the orthogonal mirrors (depending on the measurements one wants to make). The same principles can be applied to detect gravitational waves. Gravitational waves interact with matter, compressing it in one direction, and stretching it in a plane perpendicular to the plane of compression.

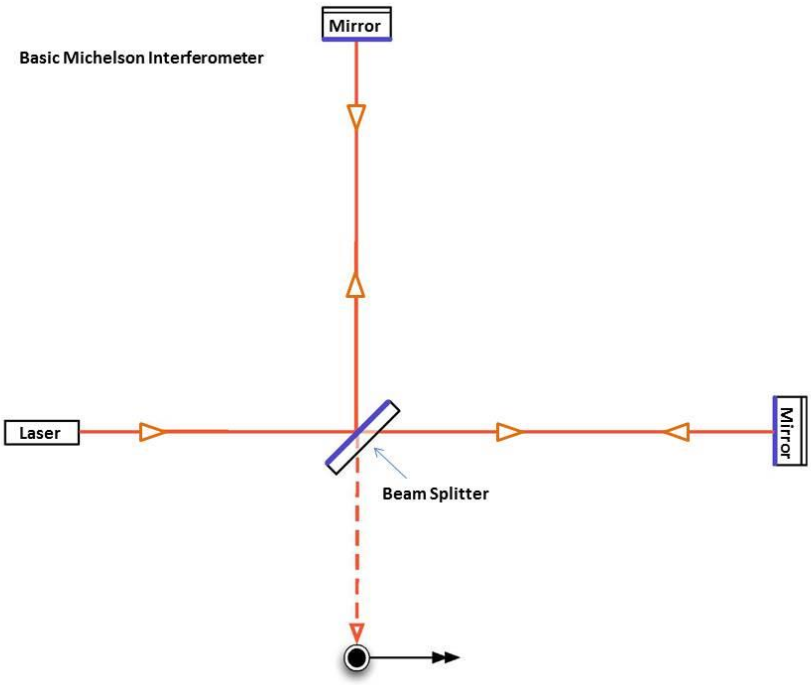

Figure 1.3: Basic Michelson interferometer configuration (image from the ligo-caltech web page)

The idea of using an interferometer to detect gravitational waves was not exclusive to one individual, but it occurred to several researchers and at around the same time. In 1962, two

Russian scientists, Mikhail Gertsenshtein and V.I. Pustovoit devised the notion of an interferometric detector of gravitational waves. Others were Joseph Weber, his student Robert Forward also in the 1960's, and Rainer Weiss in the 70's (Saulson, 1994).

Also in the late 1960's and early 1970's, scientists from the Massachusetts Institute of Technology (MIT), and the California Institute of Technology (Caltech) begun to consider the interferometer as a mean to detect gravitational waves. In 1980, the National Science Foundation (NSF), subsidized the manufacture of two prototype interferometers, one at MIT and the other at Caltech. As the prototypes presented successful results, in 1990 the NSF funded the creation of LIGO, Laser Interferometer Gravitational-Wave Observatory.

LIGO consisted of two interferometers (Figure 1.4), built in the US, one in Hanford (H1), Washington, and the other in Livingston, Louisiana (L1). These are laser interferometric systems with two perpendicular arms of 4000 m in length and 1.2 m in diameter. Two of the most critical characteristics of LIGO's interferometer are: 1) the constant vacuum system (8 times greater than the space vacuum at a pressure of $1x10^{-9}$ of earth atmosphere) which helps to ensure that the laser beams travel in a straight line inside the arms. Slight changes in temperature (refractive index temperature dependency), or even air currents, may curve the laser beam path making the vacuum system an essential part of a precise interferometer. 2) Internal and external seismic isolation system that consists of tiny magnets attached to the back of each mirror. If the mirrors move, electromagnets create a magnetic field that pushes or pull the magnets until the mirrors are in their original position. The LIGO's seismic isolation system insulate the instruments and beam from earth vibration and tidal disruption from the sun and the moon (LSC).

In addition to the Earth-based gravitational waves interferometers, there are plans to construct a progressive space-based detector. eLISA (the Evolved Laser Interferometer Space Antenna) is a

proposed gravitational wave interferometer that would trail earth's orbit. Initially a NASA project, it is now a European Space Agency (ESA) mission. eLISA will consist of three separate spacecraft orbiting in an equilateral triangle configuration separated 2.5 million kilometers from each other. After the successful launch of the "LISA Pathfinder" spacecraft, a test operation to assess the technologies needed for the final mission, the final launch of eLISA is planned for 2034. LISA will detect gravitational waves in the low-frequency band of the spectrum, roughly from the MHz to about 1 Hz in range. At this frequency range, binary neutron stars will comprise much of the signals it detects (Beroiz, 2017). Like LIGO, Virgo is a large interferometer built by an international collaboration to detect gravitational waves. It is located in Santo Stefano a Macerata, Cascina, Italy. The Virgo collaboration consists of over 280 physicists and engineers from 20 different European research groups. Schematically Virgo and LIGO are very similar. Virgo consists of 2 arms 3 km long and arranged in L shape. Virgo and LIGO, forming in 2007 the LIGO and Virgo Collaboration[1], have agreed to share and jointly analyze the data recorded by their detectors and to jointly publish their results.

1.3 LIGO-VIRGO COLLABORATION AND DETECTIONS

In 2007, the LIGO Scientific Collaboration (LSC) and Virgo united efforts in an umbrella collaboration named the LIGO-Virgo Collaboration (LVC). Almost ten years later, in February 2016, and at the centenary anniversary of Albert Einstein's first paper on gravitational waves (Einstein, 1916), the LIGO-Virgo collaboration announced the first-ever direct detection of a gravitational wave -GW150914-. With this finding, gravitational waves (GW), have leaped the

[1] https://dcc.ligo.org/LIGO-M060038/public

theoretical realm into the area of observable phenomena and the headlines of the 21st century astrophysical discoveries. When the detection of GW150914 was made, only two of LIGO interferometers were functioning as Virgo has been upgraded to Advanced Virgo. The improvements to Virgo will see a significant reduction in the localization errors with sensitivity ten times greater than that of the Initial Virgo (Virgo+).

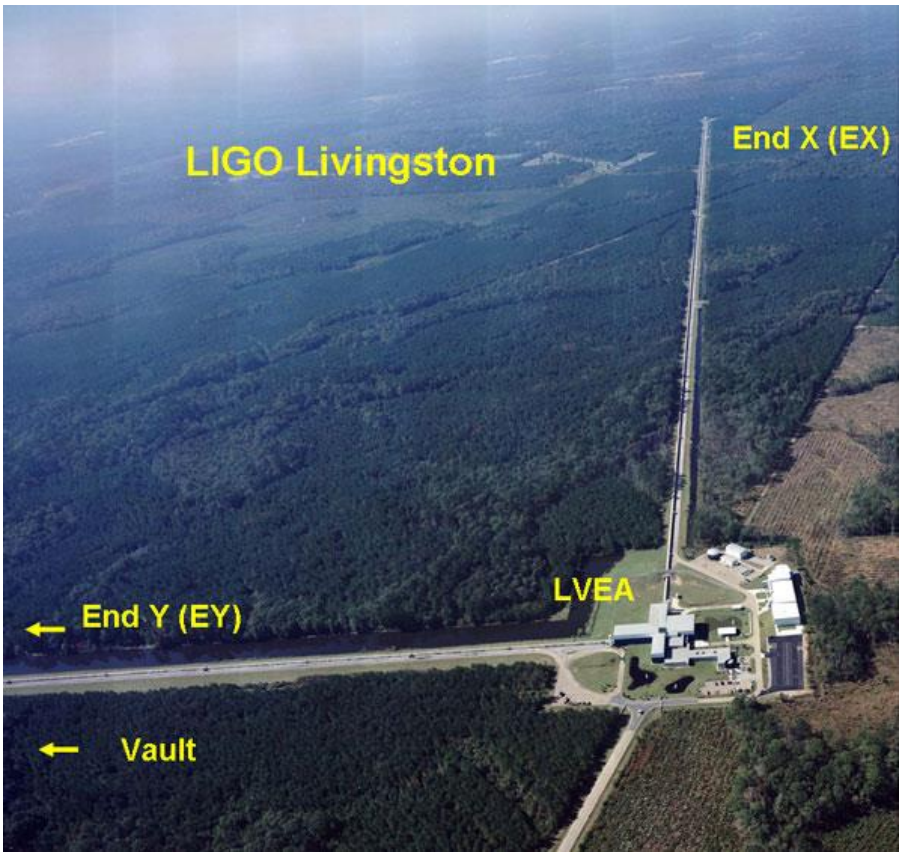

Figure 1.4: LIGO interferometer aerial view (courtesy of LSC web page)

LIGO has also been through upgrades since it started working: Initial LIGO, capable of detecting GW in the frequency band 40-700 Hz and with a strain amplitude as small as one part in 1021; Enhanced LIGO, with twice the sensitivity of Initial LIGO; and now Advanced LIGO

(aLIGO), which sensitivity is ten times greater than the Initial and it has a widened frequency range to 10 Hz (known as the seismic wall). Through the several planned runs, aLIGO will continue to improve until it reaches its sensitivity goal by 2020.

The expected sensitivity for each of the runs is presented in Table 1.1 (Abbott, 2016a); (Singer, 2014).

Table 1.1: LVC Observation Runs Projection

Epoch	Estimated Run Duration	$E_{GW} = 10^{-2} M_{\odot} c^2$ Burst-Range (Mpc)		BNS Range (Mpc)		Number Of BNS Detection	% BNS within 5 deg^2
		LIGO	Virgo	LIGO	Virgo		
2015	Three months	40-60	---	40-80	---	0.0004-3	---
2016-17	Six months	60-75	20-40	80-120	20-60	0.006-20	2
2017-18	Nine months	75-90	40-50	120-170	60-85	0.04-100	1-2
2019+	(per year)	105	40-70	200	65-130	0.2-200	3-8
2022+(India)	(per year)	105	80	200	130	0.4-400	17

1.3.1 GWS DETECTION

The aLIGO direct detection of GW150914 on September 14, 2015, at 09:50:45 UT, was entirely unexpected as it appeared in the two interferometers detectors while a testing run was underway (Abbott, 2016a). Later data analysis revealed that the GW was produced by the merging of two black holes in a binary system (BBH). This detection was of monumental importance as it opened a new era in gravitational waves physics (Diaz, 2016).

Figure 1.5 shows, Top: Estimated gravitational-wave strain amplitude from GW150914 projected onto H1" (LIGO Hanford) and bBottom: The Keplerian effective black hole separation

in units of Schwarzschild radii ($R_s = 2GM/c^2$) and the effective relative velocity given by the post-Newtonian, $^{v}/_{c} = (GM\pi f/c^3)^{1/3}$, where f is the gravitational-wave frequency calculated with numerical relativity and M is the total mass" (Abbott, 2016b). The detection of GWs proved very fructiferous for the initial two runs of the LIGO/Virgo collaboration. On October 12, 2015, LSC and LVC informed the scientific community about the candidate astrophysical event observed, LVT151012.

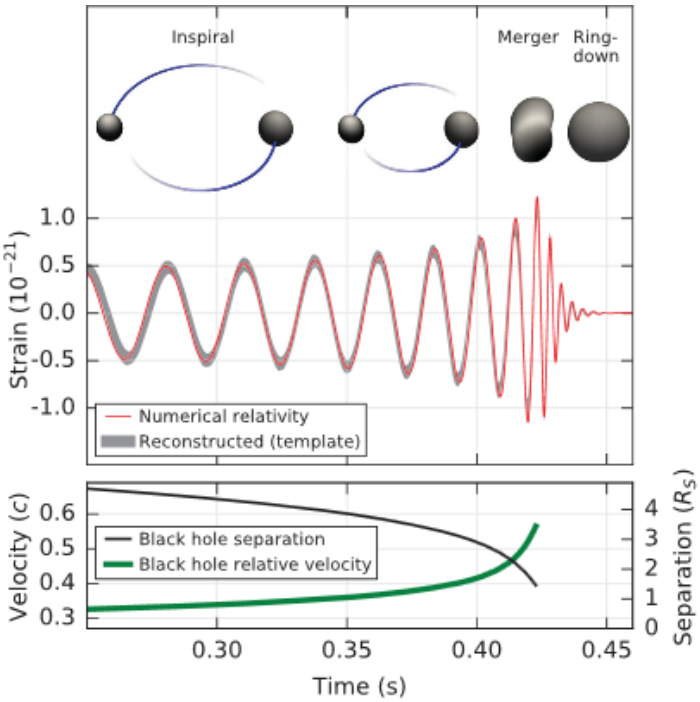

Figure 1.5: Illustration of the first detected gravitational wave GW150914

On December 26, 2015, another BBH merger was observed by LIGO, GW151226 (Abbot, 2016c). These were followed by the observation of three more binary black hole mergers: a 50-Solar Mass BBH, GW170104, on January of 2017 (Abbott, 2017a), a 19-Solar Mass BBH merger, GW170608, on June of 2017 (Abbott, 2017b); and GW170814a three-

detector observation on August of 2017, (advanced Virgo and the two Advanced LIGO detectors) (Abbott, 2017c).

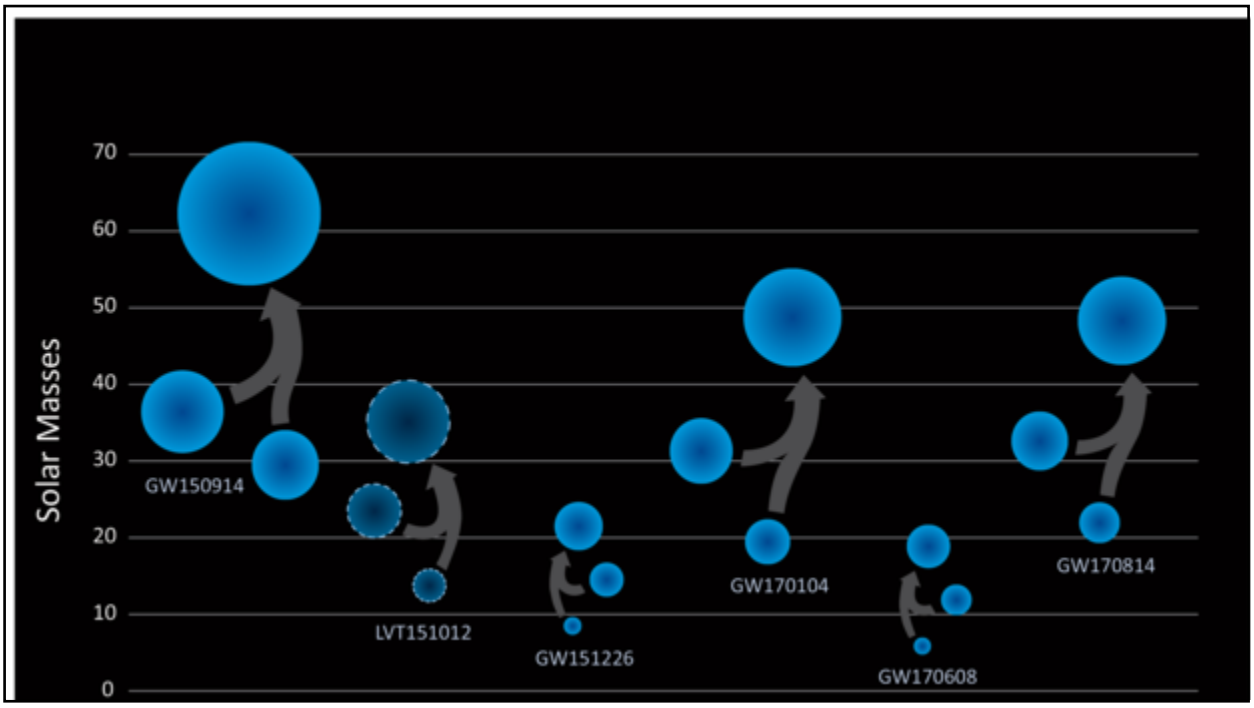

Figure 1.6: Known Black-Holes mergers as of November, 2017

Figure 1.6 shows a graphic representation of the detected mergers as of November of 2017. The graphic illustrates the binary components and the resulting black hole, in solar masses (LIGO Caltech, 2017).

Finally, on August 17 of 2017, the LIGO Hanford detector captured a GW which source was the merging of two neutron stars (BNS). This coalescence, dubbed GW170817, opened the door for the researchers of electromagnetic radiation (EM), counterparts to GW. Also, this first EM finding endorsed the hunt for the almost mythological merger remnant called a Kilonova. This EM counterpart is believed to shine about a thousand times brighter than a Nova, hence the name, and can help narrow the localization of the GWs to a few arcseconds. Models of

Kilonovae afterglows peak about a day after the GW emission (Metzger, 2010)) and can last for about a week (Beroiz, 2017).

1.4 THE MULTI-MESSENGER SIGNAL GW170817-GRB170817A-SSS17A

In February 2016 the announcement of the first detection of gravitational waves was made, then two more detections, until last August 17 at 8:41, the LIGO Gravitational Wave Observatory in the United States and the Virgo Observatory in Italy, detected a fourth gravitational wave (GW170817) and two seconds later the NASA Fermi space telescope and the Integral of the European Space Agency, observed in unison a burst of gamma rays: the most powerful type of explosion in the universe after the Big Bang (gravitational wave GW170817, along with a parallel short gamma-ray burst, GRB170817A and an optical transient source, Swope Supernova Survey 17a, SSS17a). These observatories determined the point of the sky from which the signals came and launched international alerts: one had to search for a needle among 50 possible galaxies. That same day, seventy observatories (Figure 1.7a), both on land and in space, pointed to a small area of the sky in the constellation of Hydra. The Observatory of Las Campanas in Chile, was the first to capture the event in the optic in the galaxy NGC 4993, then the Chandra X-ray satellite, more observatories captured it in the infrared and the large antennas in Hawaii and New Mexico detected it in the form of radio waves: The multi-messenger astronomy was born. The term multi-messenger in astronomy refers to the synchronized observation and interpretation of diverse "messenger" signals. There are four of these "messengers" that come from outside of our solar system: electromagnetic radiation, neutrinos, cosmic rays, and now we have to add gravitational waves. In this case, these four signals originated from the same event, though through different astrophysical processes, and are

bound to reveal a huge amount of information pertaining to their source (Bartos, 2017). In Figure 1.7b are represented the data results from the many telescopes that captured the GW170817 in all areas of the spectrum.

A peculiar characteristic of this signal, and one that the differentiated it from those previously captured, was its duration. The previous signs lasted approximately a fraction of a second, while this signal lasted about 100 seconds, which was instrumental in reducing uncertainties in the location of the source (Manreza, 2017). The analysis of the form of the signal also showed that it corresponded with the collision of objects less massive than the previous ones.

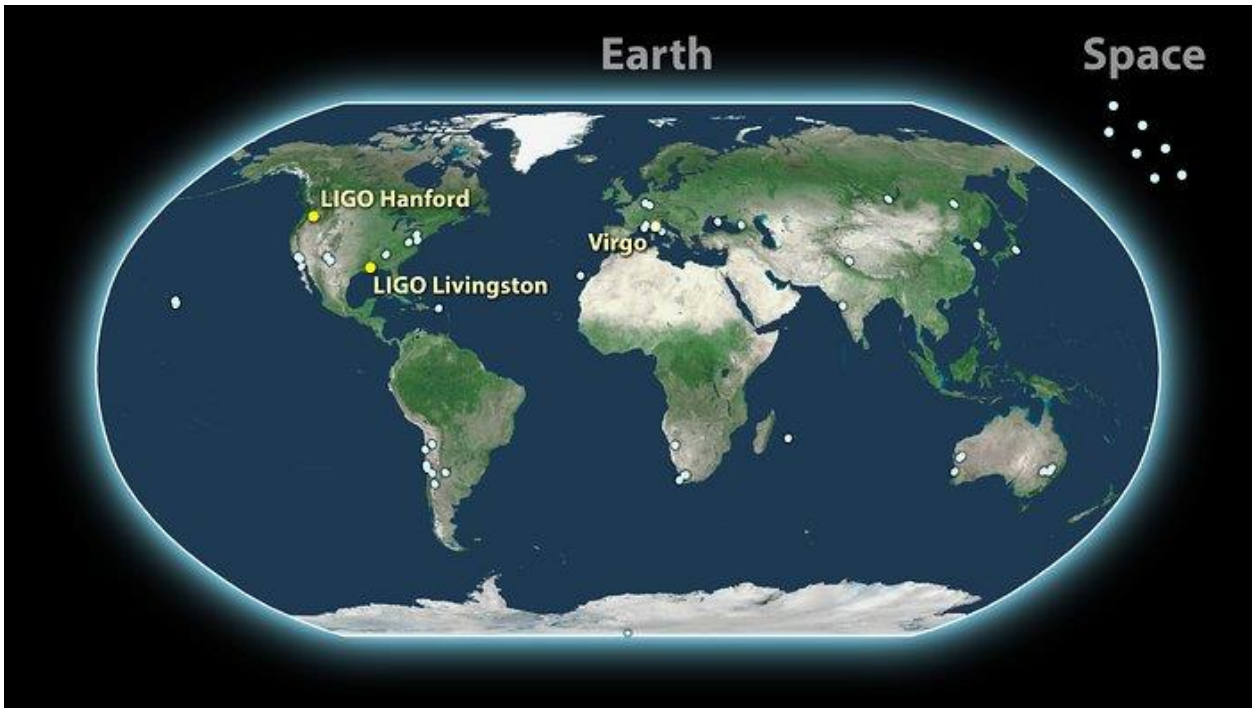

Figure 1.7a: Map of the approximately 70 observatories that detected the gravitational wave event called GW170817.

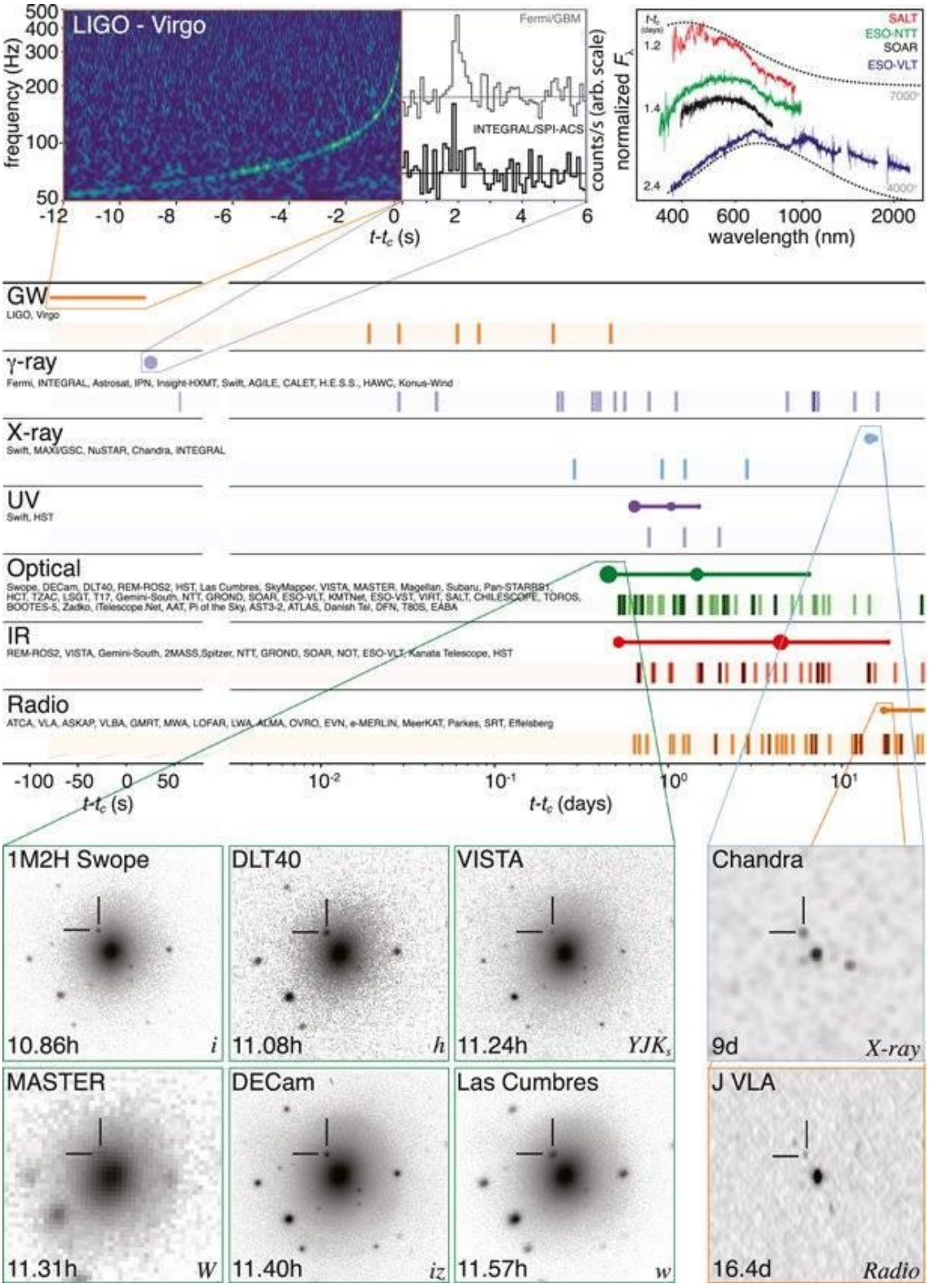

Figure 1.7b: Multi-messenger Astronomy. Observations made by 70 Observatories (on land and in space). Credits: ESO + NSF.

The masses were estimated at 1.1 $M_{\odot}$ and 1.6 $M_{\odot}$, which makes them compatible with neutron star masses. Also, by studying the amplitude of the signal it was possible to determine the approximate distance from the source at about 130 million light years with a 30% margin of error, therefore this event occurred at a distance closer to the Earth than the black holes' collisions.

The theoretical models had already predicted that the collision of two neutron stars should emit gravitational waves and gamma rays as well as a considerable amount of other emissions in the form of jets in all the bands of the electromagnetic spectrum, with which this observation comes to validate for the first time, said theoretical predictions. The gamma rays emitted were in the form of short flashes (short gamma ray bursts, GR), and it was called GRB170817A. This observation allowed solving one of the great enigmas of the high-energy astrophysics: precisely where this type of radiation came from. Observation of the signal GW170817 followed at two seconds by gamma rays, allowed checking also that gravitational waves travel at the speed of light (Manreza, 2017). With the collision of these neutron stars it was possible to prove, for the first time, the hypothesis of the appearance of a kilonova, 1000 times brighter than a Nova and its relationship with the important discovery of gold and platinum. This detection has served to confirm the theories that related the origin of the heavy elements of the periodic table with these convulsive events. They are the precious metals dust of neutron stars (Manreza, 2017).

On the other hand, the observation has provided a new way of measuring the Hubble constant -the rate of the expansion of the universe as a function of the distance from Earth-. This measurement can be made because knowing the amplitude and polarization of the gravitational wave it is possible to determine the distance to source. It was also possible to determine the

redshift of the light of the galaxy of origin and the speed at which it moves away from the Earth. Based in these measurements, the Hubble constant was calculated[2] ($70\ kms^{-1}\ Mpc^{-1}$), and the result agrees with measurements previously made with other techniques (Manreza, 2017).

Although the signal was detected by LIGO, Virgo played a role, because thanks to this detector it was possible to triangulate the location of the source in a small region of the sky, which allowed the other observatories to know where to point their telescopes to study the signal in all bands of the electromagnetic spectrum. With this a new window to the observation of the universe has been opened. Figure 1.8 shows the August 17, 2017 event discovered by the Laser Interferometer Gravitational-Wave Observatory (LIGO) and the Virgo Interferometer. Both coincided that the gravitational waves were produced by the collision between two neutron stars. "Within 12 hours observatories had identified the source of the event within the lenticular galaxy NGC 4993, shown in this image gathered with the NASA/ESA Hubble Space Telescope. The associated stellar flare, a kilonova, is clearly visible in the Hubble observations. This is the first time the optical counterpart of a gravitational wave event was observed. Hubble observed the kilonova gradually fading over the course of six days, as shown in these observations taken in between 22 and 28 August (insets). (NASA and ESA)." (Hubble News, 2017)

[2] https://www.ligo.org/science/Publication-GW170817Hubble/index.php

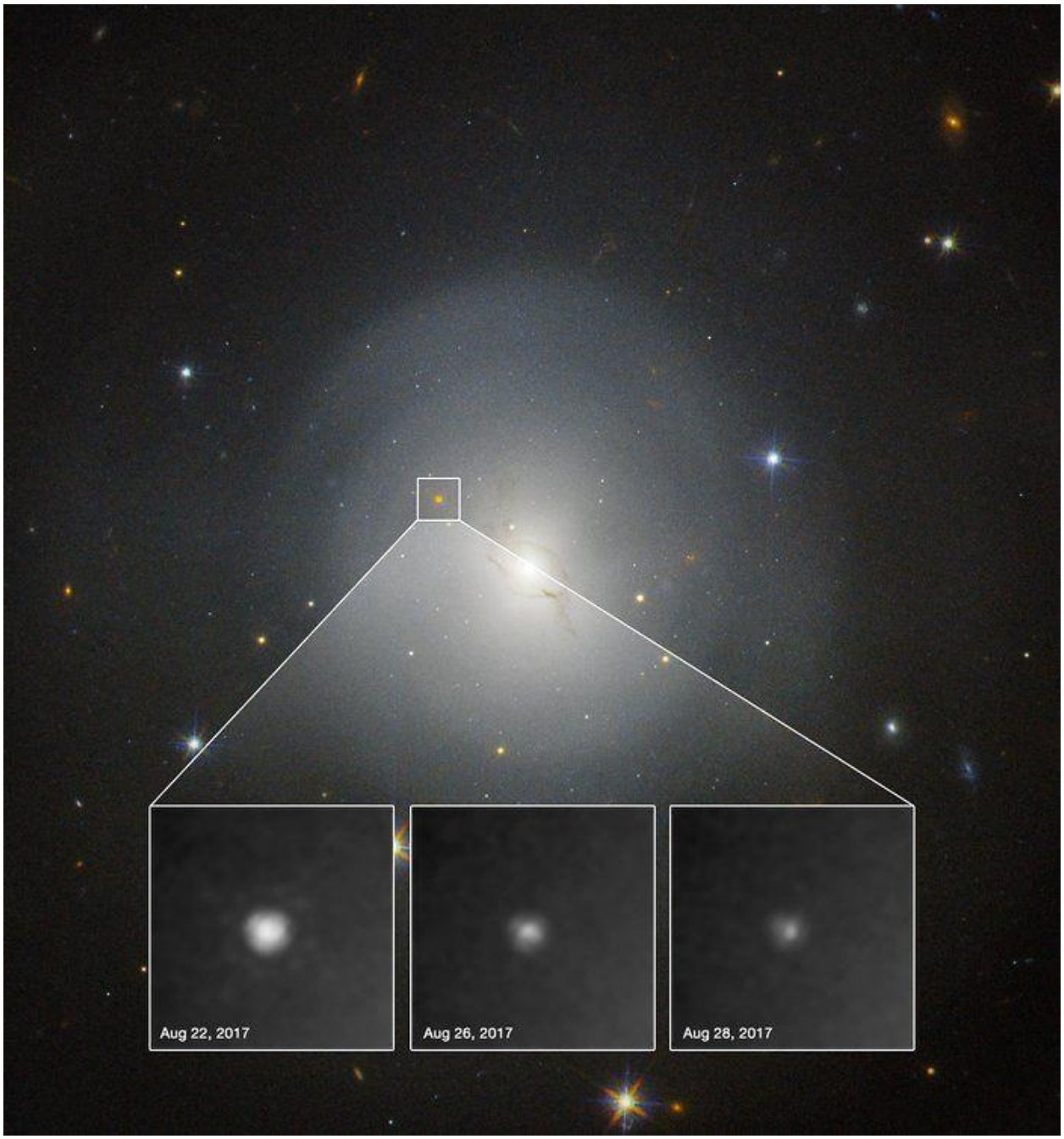

Figure 1.8: Kilonova Electromagnetic counterpart to the GW170817 originated at the lenticular galaxy NGC 4993 courtesy of Hubble space telescope

CHAPTER 2: SEARCH FOR EM COUNTERPART TO GW

2.1 THE TOROS COLLABORATION

In December of 2015, LVC extended an “Open call for partnership for the EM identification and follow-up of GW candidate events” (LIGO-M1300550, 2015). The LIGO and VIRGO call was to sign a Memoranda of Understanding (MOU) between LVC and independent astronomers/astronomy groups to establish a follow-up system to GW candidate detection resulting in EM emission. The MOUs meant to be an individual agreement of partnership in the program concerning LVC and the astronomers, but also a sharing of information among the partners. The MOU represented a framework of what the enterprise would include, a background on LIGO and VIRGO program, with a time-table observation schedule, application instructions, eligibility conditions, and an overview of the GW-EM follow-up program, which included a guiding policy, the sharing of alerts, and a guideline to publications.

“The baseline plan is to publish the GW discovery paper standalone, followed by companion papers for the results of the EM follow-ups. However, if the science calls for it, we may mutually agree to opt for a joint publication.” (LIGO-M1300550, 2015). There was especially a confidentiality agreement. The agreement recognized the sharing of information among the scientists in a sort of "bubble." Inside of it the exchange of information flowed freely to make the most out of the separate observations, however, all data and analysis belonged to the individual groups and their property needed to be firmly respected. The collaborators also agreed to maintain an unrestricted availability of the information to the public through an embargoed of information related to the ID number, date, and time of a gravitational wave event and its EM counterpart was put on place. The embargo was to be lifted after the end of each separate run.

One of the groups that signed the MOU, and the one this researched is involved with, was TOROS.

In 2011, scientists from the Center for Gravitational Wave Astronomy (CGWA) at The University of Texas at Brownsville (UTB), the Observatorio Astronomico de Cordoba (OAC), and Instituto de Astronomia Teorica y Experimental (IATE) (the last two in Argentina) established the TOROS project (Benacquista, 2014) to scan likely areas of the localization uncertainty sky map of the GW trigger, looking for possible optical counterparts. TOROS stands for Transient Optical Robotic Observatory of the South, and its original project consisted in the construction of an observatory site in Cordon Macon, a mountaintop at 4637 m above sea level, in the Andes mountain range in the north of Argentina (Beroiz, 2017).

The planned TOROS telescope would have a 0.6 m aperture and a 9.85 sq. Deg. Field of view. The idea is to operate on three fronts: follow up of GW triggers; follow up of gamma-ray burst triggers from Fermi, Swift, or other missions; and baseline imaging of the entire surveyable area. The project includes the creation of a data reduction and processing pipeline for transient detection and a database for the dissemination of catalogs and triggers. While waiting for the proper funds to develop the necessary instrumentation, other institutions requested to be added to our efforts. This only extended the capabilities of TOROS to an assortment of telescopes instead of the single Argentinean instrument as was first intended. However the changes, TOROS remain single-minded in the pursuit of its objective: the observation of GW events that include the merging of at least one neutron star since only such events are thoughts to produce a kilonova counterpart (Beroiz, 2017). The entire TOROS's procedure can be summarized into the creation of its pipeline. The pipeline is written entirely in Python.

Python language was chosen for its "user-friendly" characteristics. This language has become the go-to programming language in many of the scientific fields, including astronomy. For years astronomers collected a series of software packages written by and for astronomers, this software collection is called IRAF. IRAF corresponds to the acronym Image Reduction and Analysis Facility. This system provided a broad set of tools for the analysis and processing of astronomical images. It was developed in the mid-1980s by the National Optical Astronomy Observatory (NOAO) in Tucson, Arizona. It became popular in the astronomical community among other things because it was selected as the System on which the data reduction and analysis programs of the Space Telescope (HST), was based on. In 2008, the HST started to incorporate python through the creation of the Pyraf package, which used the IRAF tasks in the Python language (Sosa, 2015).

TOROS wanted to avoid the use of pyraf in its pipeline since its scientists preferred a cleaner approach through a stand-alone python script. The python language incorporates IRAF tasks by importing astronomy packages. One of the most complete of these packages is astropy. Astropy is a Python library for astronomers that includes many standard features in astronomy. Until several years ago, numerous independent astronomical libraries existed with specific functionalities such as asciidata (catalogs), APLpy (astronomical graphics) or pyfits (images and FITS tables), etc., that were maintained by developers with their syntax. In 2013 the first open version of astropy was published. This first version groups, in the same programming interface application and with common classes, libraries for reading catalogs, coordinates, dates and time, and more. In February 2018, a new astropy version -version 3.0- was published. However, version 2.0 continues to be the variety with long-term support (until the end of 2019) and therefore the best for programs that need unique stability like pipelines and commonly used

applications for Python 2 users, since astropy v3 is only available for Python 3. Python, along with tools like git (and the GitHub repository website) and virtual environments, make Python projects reproducible and installable in almost any Unix-like system with minimum effort. The TOROS project itself spun the creation of two new modules: astroalign and ois to align stellar images and perform image subtraction respectively (Beroiz, 2017).

2.2 TOROS PIPELINE

Being the automatization (robotization) of its telescope TOROS project's final goal, creating a software pipeline seems essential as the need exists to make the processing of images as automatic as possible, this means, taking out of the equation the human factor. From the moment an alert is triggered to the confirmation –or rejection- of a GW event, several steps and stages built TOROS pipeline. Dr. Martin Beroiz, one of TOROS scientists, created a sequenced flowchart representation of such steps shown Figure 2.1. At each individual step, one can find a python script which main purpose is to connect one stage to the next. A summary of the steps is given next.

— LVC Alert System: The Virtual Observatory Event file (VOE), is the GW detection notice distributed by LIGO-Virgo Collaboration after it has received a detection alert from the Gamma-Ray burst Coordinate Network (GCN) and the Transient Astronomy Network. The VOE notice, using the VOEvent (a simple transaction processing council, TCP, based protocol for transporting VOEvent messages from authors, through brokers, to subscribers), Transport Protocol, is delivered to predefine static IPs and ports of the

participating observatories searching for EM counterparts (TOROS registered two IPs, one at UTRGV in Texas, one in Cordoba, Argentina, but only the Texas IP is operating).

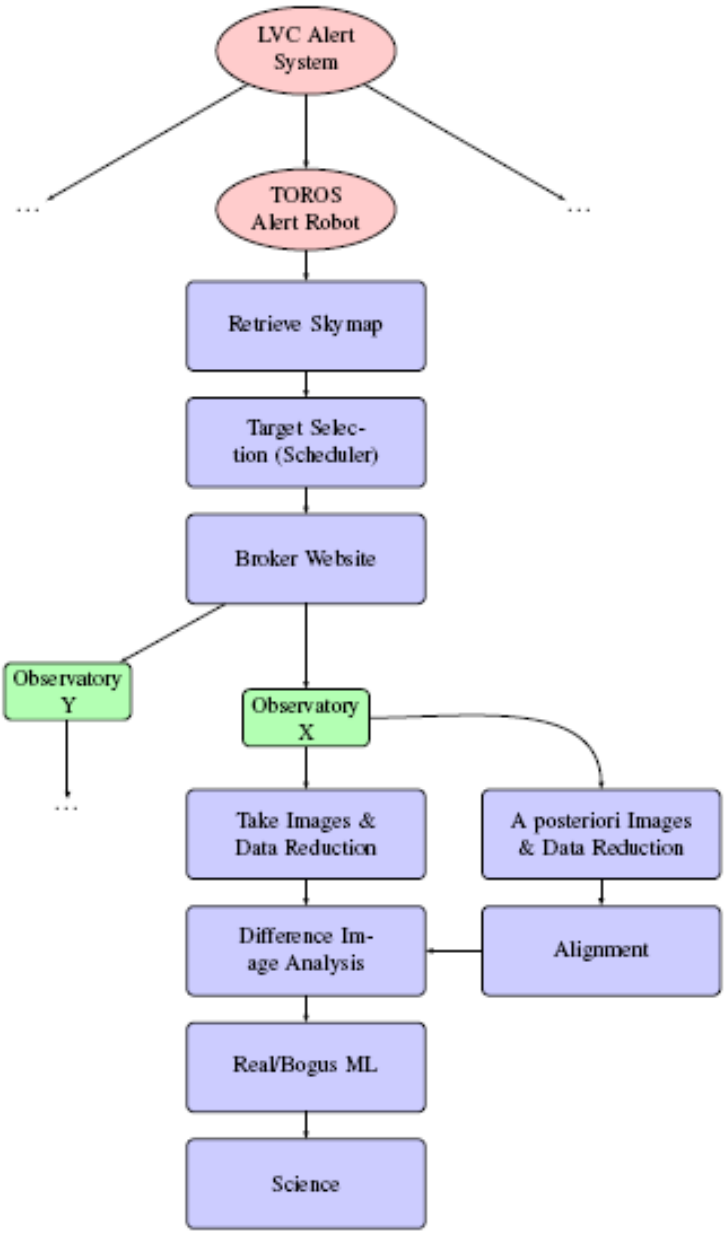

Figure 2.1: TOROS pipeline sequenced flowchart.

— TOROS Alert Robot: TOROS receives the VOEvent by a continuously-running Python script that listens on the port. This port is managed through a receiver robot that is hosted in a virtual machine server provided by the University of Texas Rio Grande Valley (UTRGV) IT Services. The listener script then takes four steps: first, it creates an alert email to send to individual collaborators at each TOROS partner telescope; second, it parses the XML file. Since the VOEvent comes in the Extensible Markup Language (XML) format, "the structured text is parsed with a Python native package to scavenge the relevant information from within the nested hierarchy." (Beroiz, 2017). Table 2.1 lists some of the information contained in the VOEvent; third, the listener script sends a second email containing the most significant information from the VOEvent and the skymaps recovered from the provided URL and its corresponding PNG file (see Table 2.1); fourth and last step, it generates a list of targets of interest to be followed-up at each observatory and upload them to the broker website.

Skymaps: Through triangulation LIGO determines the origin of the signal. Since LIGO uses its two detectors, the results will contain a degeneracy separated by 180°, and it may determine the origin to be in two separated regions. Obviously, a third detector would be needed to break the degeneracy and to cut in a third the localization uncertainty area (Singer, 2014) Figure 2.2 shows the finalized LIGO skymaps of six GW events. As soon as an equivalent detection by the operating detectors is made, a quick sky localization is prepared with BAYESTAR (Singer and Price, 2016)). "BAYESTAR is a rapid Bayesian position reconstruction code that will produce accurate sky maps in less than a minute after any BNS merger detection". This is accomplished by relying on the equivalent

detection triggers: the phases, times of arrival and amplitudes, along with prior probability distributions of source parameters (Beroiz, 2017).

Table 2.1: Part of information provided by a voevent notice (Beroiz, 2017)

GraceID	Identification in the Grace Database
Alert Type	VOEvent alert type. It could be any of "Initial" or "Update"
SKYMAP_URL_FITS_BASIC	Provides the URL of the location probability map for LIGO/Virgo GW event detection (FITS format)
SKYMAP_URL_PNG_BASIC	Provides the URL of the location probability map for LIGO/Virgo GW event detection (PNG format)
ProbHasNS	Probability that at least one object is the binary is less than 3 solar masses
ProbHasRenmant	Probability that there is matter in the surroundings of the central object
ISOTime	The time of the event in ISO format

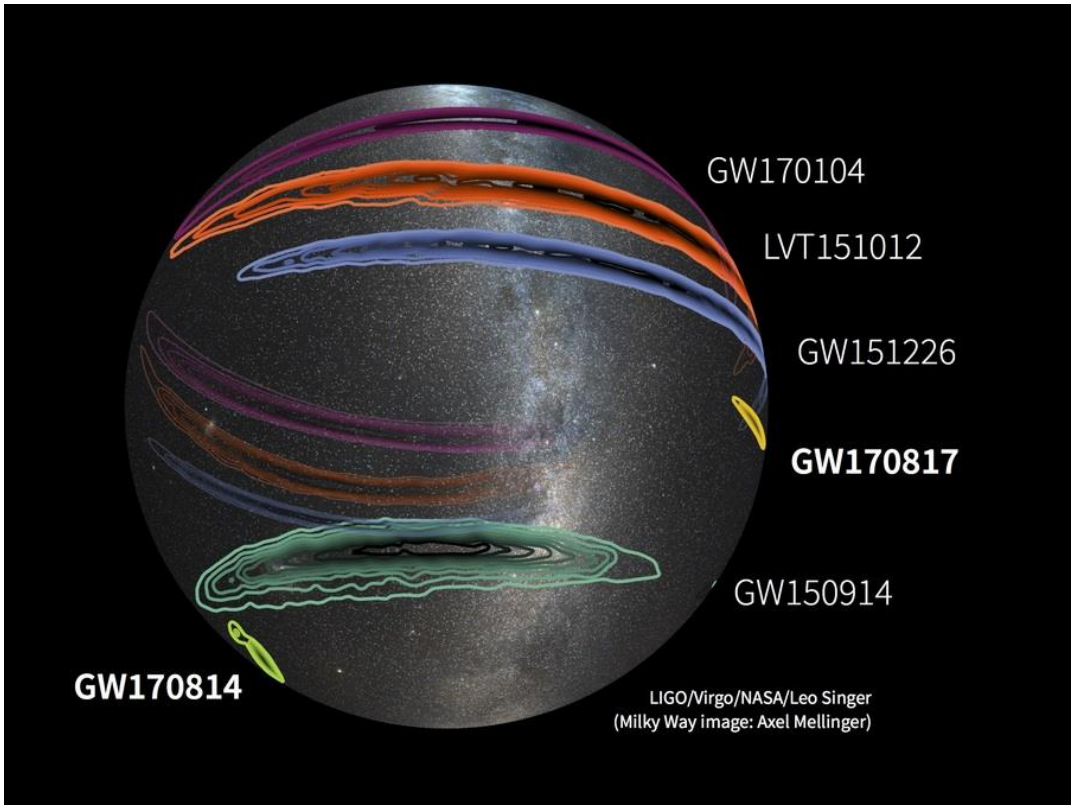

Figure 2.2: Skymap of GW event. Credit of LIGO/Virgo/NASA/Leo Singer (Milky Way image: Axel Mellinger).

— Target Selection: a separate module imported from the alert receiver robot is in charge of the target selection. This separation permits independent runs, preventing escalating errors in case of system failures. The skymaps attached to the alert usually cover about 200 square degrees for the 60% confidence region (Singer, 2014). Adopting additional strategies for target selections is essential if the instruments available cannot perform a wide-sky search to maximize the chances of discovery. This has been the case for TOROS telescopes. Our tactics include making a prioritizing list of objects of interest to accomplish the best results in the shortest time possible. Using galaxy catalogs significantly assists in the search for a signal point of origin. The importance of catalogs was first mentioned in White et al. 2011, where they presented the Gravitational Waves

Galaxy Catalog (GWGC). The GWGC catalog is a homogeneous list of 53,255 galaxies within 100Mpc. It is a compilation from 4 different catalogs: an updated version of the Tully Nearby Galaxy Catalog, the Catalog of Neighboring Galaxies, the V8k catalogue and HyperLEDA. GWGC contains information on sky position, distance, blue magnitude, major and minor diameters, position angle, and galaxy type. Also included in the catalog are 150 Milky Way globular clusters. Different authors agree that the GWGC is a more comprehensible catalog –within 100 Mpc– due to their use of more up-to-date input catalogs and the fact that they don't make a blue luminosity cut. Concerning the effectiveness of using catalogs, "Hanna et al. (Hanna, 2014), found that the use of galaxy catalogs can improve success rates by ~10% to a factor of 4 relative to follow-up strategies that do not utilize this kind of catalogs." (Beroiz, 2017).

— The Broker: After a list of targets is generated, each of the telescopes working with TOROS are able to select those targets that fall inside of their corresponding fields of view. This is handled using "a broker website written in Python using the Django[3] web framework and a light database on Sqlite 3." The database manages all the observatories information such as telescope details, observatory ephemeris, and log-in identifiers (website[4] hosted in UT RGV servers); it also manages information for the GW Galaxy Catalog. The website creates and controls specific list of targets tailored to each observatory thus preventing observation duplicity. The targets are then marked off the list as the observatories complete their observations. The website is also able to output a

[3] https://www.djangoproject.com/

[4] http://toros.utrgv.edu/winnow/broker

circular draft of all the targets observed by each telescope to be submitted to the GCN/TAN website for the rest of the community (Beroiz, 2017).

— Data Reduction: Though all participating telescopes manage their data reduction pipe line systems, the UTRGV TOROS group has implemented its own Python data reduction script called Imageredux. The particulars about this script and a detailed explanation about data reduction in given in the Section 2.3.1

— Optimal Subtraction: is the method of improving a given image by comparing it to another image, called a background or reference image, and then looking for the differences between them. This method is very useful when the process of capturing images cannot guarantee that all frames are taken with the same intensity of light. To make a suitable subtraction of the two images, one has to match the frames to be exactly the same seeing. Image Difference, also known as Optimal Subtraction or Differential Photometry, is a technique to find a convolution kernel that best describes (in some minimization sense) the change in point spread function between the images. The idea is to degrade the good seeing image —our reference image— to match the seeing of our second image. Finding the proper convolution kernel can be a delicate operation and there's plenty of literature on the subject. Methods range from PSF modeling through common Gaussian profiles, to unmodeled PSF's, to Information Theory, and Fourier Domain (Beroiz, 2017)

2.3 PROCEDURE

2.3.1 IMAGE REDUCTION

A digital image is an array of square pixels arranged in columns and rows. The base of a digital images is the number of photons falling on a light sensor. These photons will generate photoelectrons in separated photosites (pixels) of the sensor, and the output from each photosite is proportional to the number of photoelectrons detected during the exposure of the sensor to light. The number of photos is not proportional to the exposure time (also called integration time), since their arrival to the sensor is random. Under good conditions –good seeing conditions- the signal may consist of a few hundred photons/minute (Richard Berry. 20015). This randomness means that photons obey Poisson statistics. The statistical fluctuations in the average number of photons (over a long time interval), μ, have a standard deviation $\sigma = \sqrt{\mu}$. Thus, the signal to noise ratio (SNR), is given by the equation

$$S/N = {}^{\mu}/_{\sigma} = \sqrt{\mu} \quad (4)$$

Then, the statement $\mu \pm \sqrt{\mu}$ will indicate a normal distribution with the characteristic bell-shaped curve. Obviously "…the more photons in a signal, the better the signal quality" (Berry, 2017). Thus, the integration time plays an important role in the quality of an image. Usually this integration time will be assigned according to various factors including the brightness of the celestial object one wants to capture.

Astronomical data is characterized by the presence of noise which corrupts the information conveying signal. It produces unwanted "effects such as artifacts, unrealistic edges, unseen lines, corners, blurred objects and disturbs background scenes" (Boyat and Joshi, 2015). This noise may be originated in the seeing area, atmospheric conditions, cosmic occurrences, and mechanical flaws.

The first step for processing the images is to prepare the BIAS, DARKS, and FLAT frames.

A BIAS image (Figure 2.5b), is the combination of zero-time exposures (closed shutter). Said image contains the background and noise read from the detector present in all the images taken. They are parasitic signals mingled with the desired signal (called Light image or just image). All BIAS frames are added creating a Master Bias which will be subtracted from the Light images (same capture date).

The DARK frames (Figure 2.6), are images taken with an equivalent exposure time as the Light image frames; also a DARK frame can be created with the same exposure as the highest image frame exposure time. The shutter is kept closed so the frames will contain the electronic noise generated by the electrons freed by thermal energy (not by photoelectric effect). This type of current is called thermal current. All DARK frames will be added, by integration time, creating a Master Dark that will be subtracted from the Light images frames by integration time. It is often the case that BIAS and DARK frames are interchangeable and the presence of one makes unnecessary the need of the other (Villicaña, 2010).

The FLAT frame (Figure 2.7b), is an image with (ideally) the same exposure time as the images we need to process, illuminated as uniformly as possible. Flat frames correct for dirt or dust in the telescope system, vignetting and internal reflections. Since each pixel functions with its own somewhat different efficiency, the final image will not represent the true light spreading but one distorted by unequal efficiencies. Flats will correct this by taking an image of a uniform source.

Telescopes use either a “dome flat”, which is typically an image of a white screen installed on the dome of the telescope or “sky flats” which are images of the twilight sky (Beroiz, 2017). FLATS need to be combines, by filter used, creating a Master Flat that then is divided from the Light image. The Master Flat is often normalized, but we skipped the normalization and obtained exactly the same quality. In general, these steps can be summarized as

$$S = \frac{L - B}{F - B}$$

Where S represent all reduced frames, L is the Light frame, B is the Master Bias (or Master Dark), and F is the Master Flat.

As an example, the star HAT-P-27-WASP-40 (HATP27). Figure 2.3 shows a picture of this celestial object, at the center, obtained from the SIMBAD Astronomical Database using Centre de données astronomiques de Strasbourg (CDS) displayer. HATP27 is 204(±14) pc from our solar system and it is located between the constellations of Bootes and Virgo in the celestial globe. Using the Dr. Christina V. Torres Observatory (former Nompuewenu Observatory, in Brownsville, Texas), several images, Light (unfiltered), BIAS, DARK, FLATS, were taken on the night of June 22 of 2017. The telescope is a Meade 16 LX200 GPS. All frames (Figure 2.4a to Figure 2.8), were processed using IRAF package CCDPROC-imred-ccdred: imcombine, biascombine, zerocombine, and flatcombine, and flat mean normalization using Imarith module (Figure 2.7a). The resulting image (Figure 2.8) is here displayed by SAOImage DS9 which is an astronomical imaging and data visualization application. We repeated the process using the TOROS UTRGV reduction python script Imageredux to compare the results and to test the efficacy of the script that will be used as part of the TOROS Pipeline.

Figures 2.4a shows what is called a raw image, which is an unreduced image, zscale parameter, and inverted colormap. The main lens of our telescope has some dirt in it, the "space-worm", here we can clearly see it in the focused area viewing window (orange arrow). It is expected that the space-worm will disappear once the images are reduced. Figure 2.4b shows the target's celestial coordinate in Right Ascension (RA) and Declination (Dec)

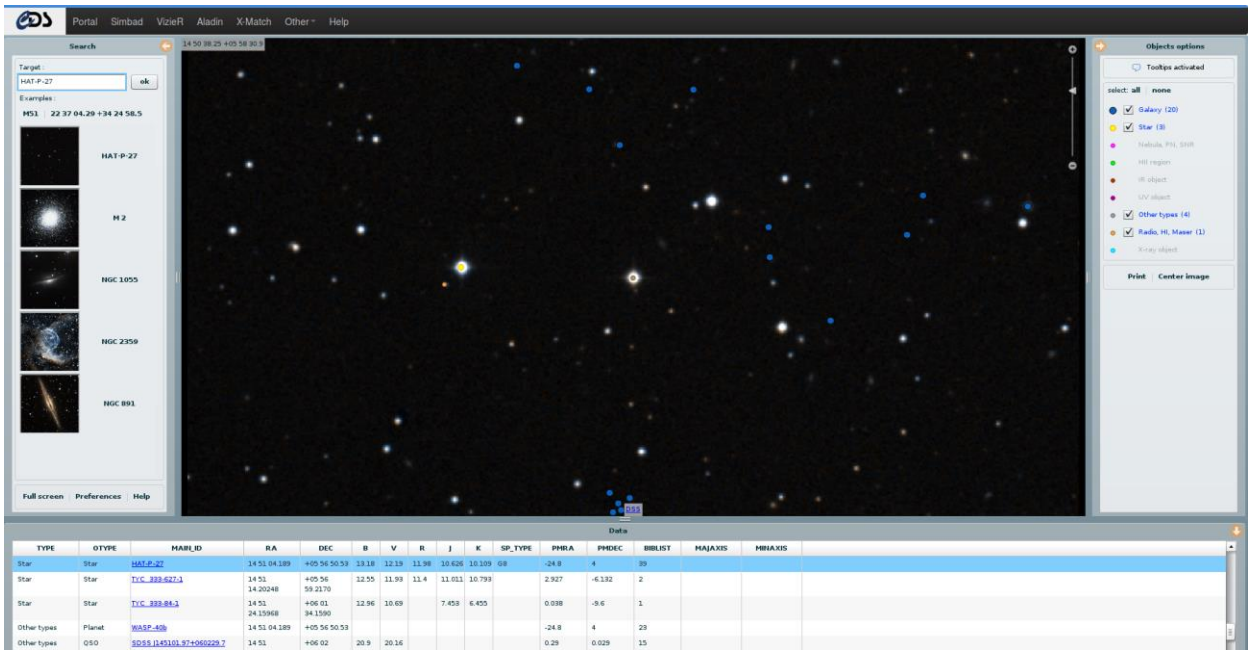

Figure 2.3: hat-p-27-wasp-40 from Simbad/CDS astronomical database

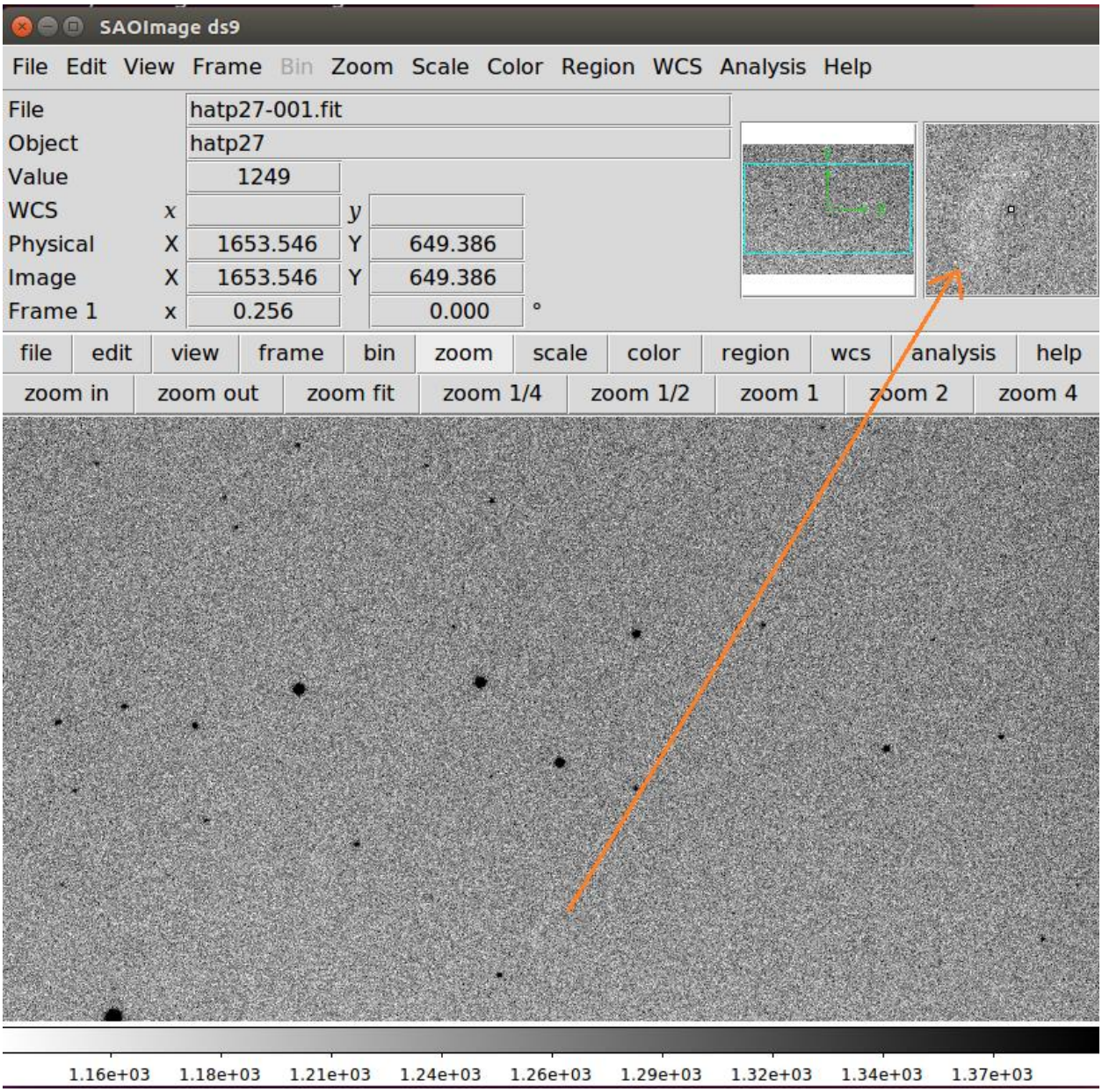

Figure 2.4a: Hatp-27, image oo1, displayed in ZScale, inverted color. Unfiltered raw image

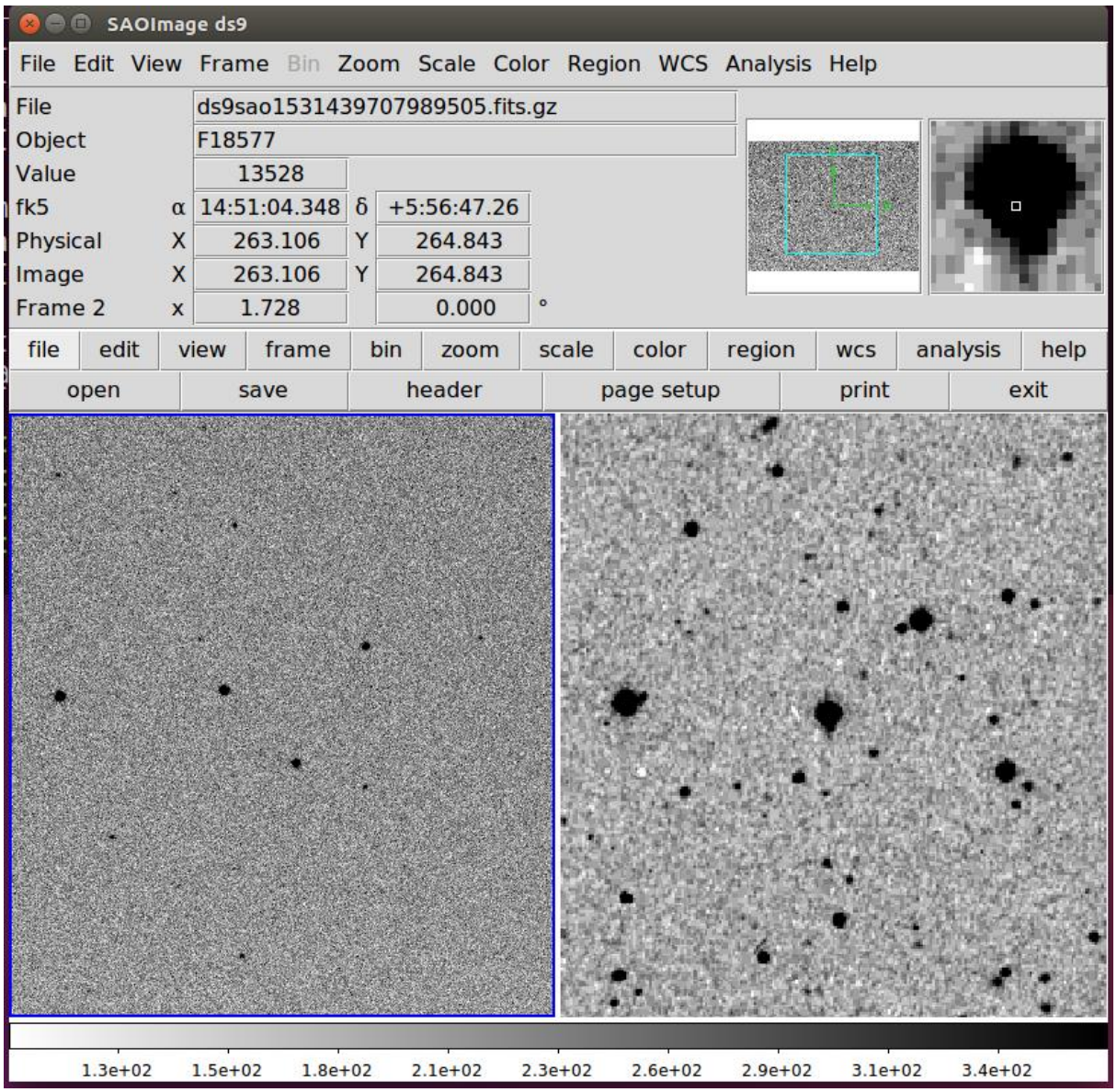

Figure 2.4b: Celestial Coordinates for target HATP27

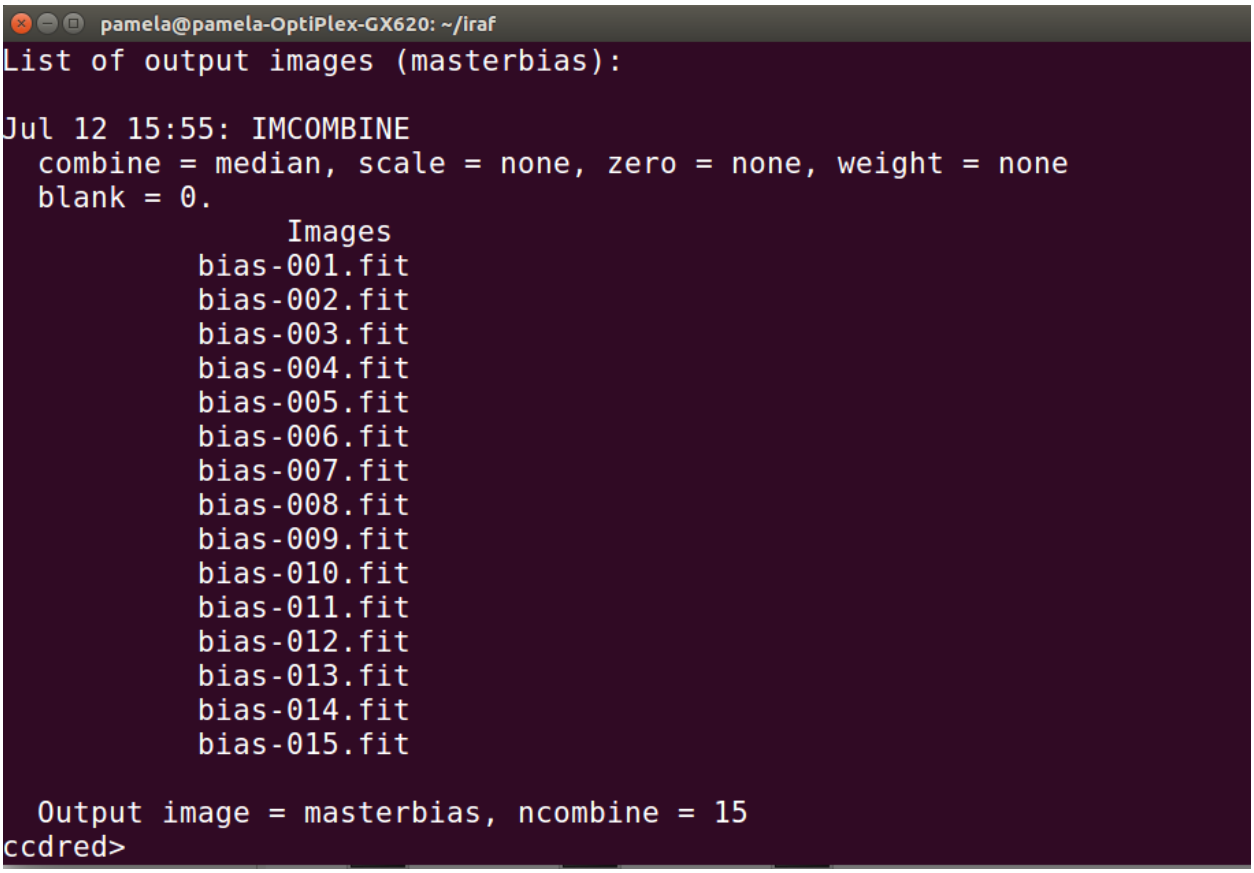

Figure 2.5a: imcombine iraf-ccdproc module from a Linux terminal to create a masterbias. Similar output for zerocombine and flatcombine

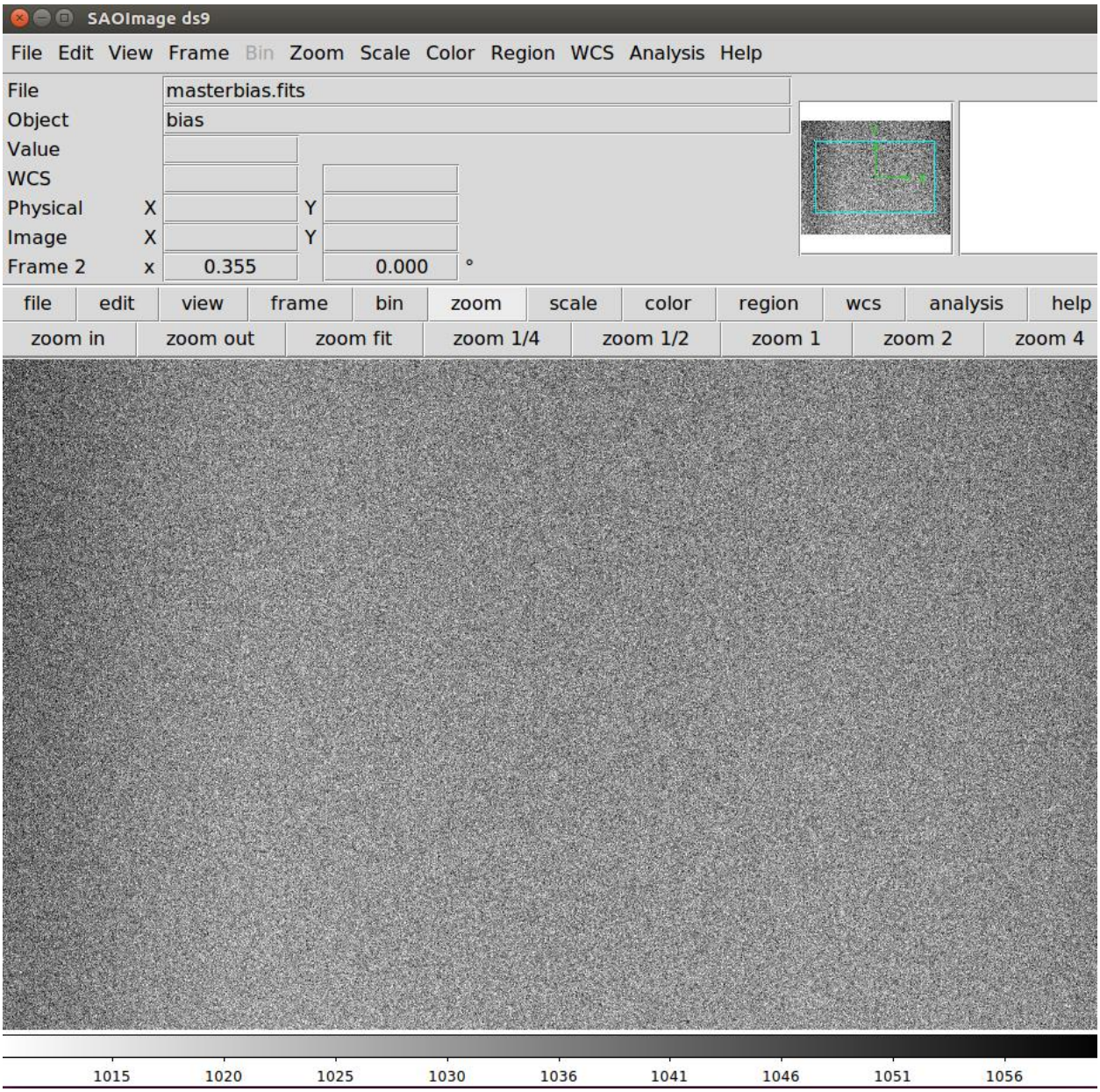

Figure 2.5b: Masterbias to reduce hatp27, ds9 display. Created on June 22, 2017

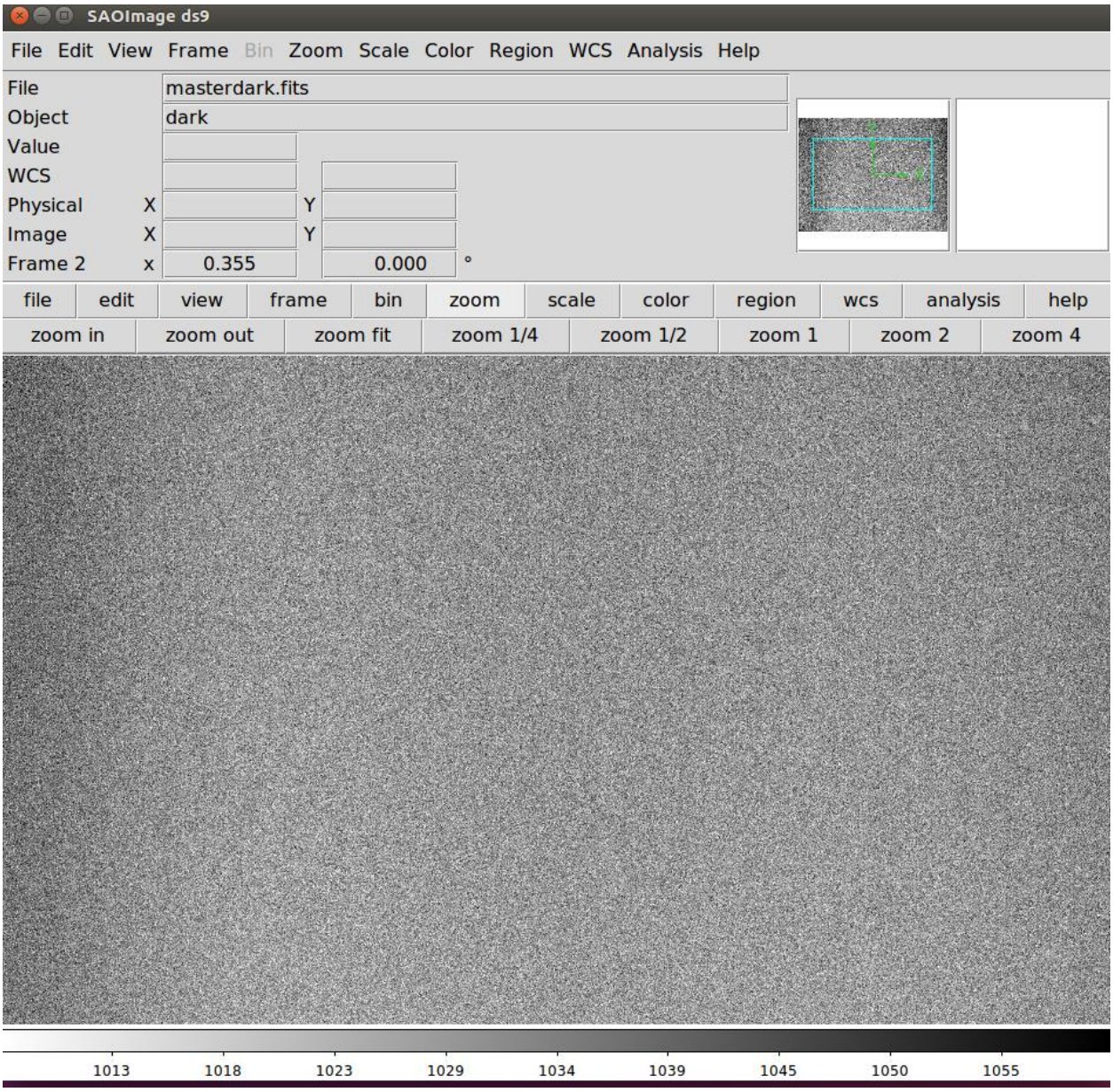

Figure 2.6: Master dark to reduce the HATp27, ds9 DISPlay. Created on June 22 of 2017

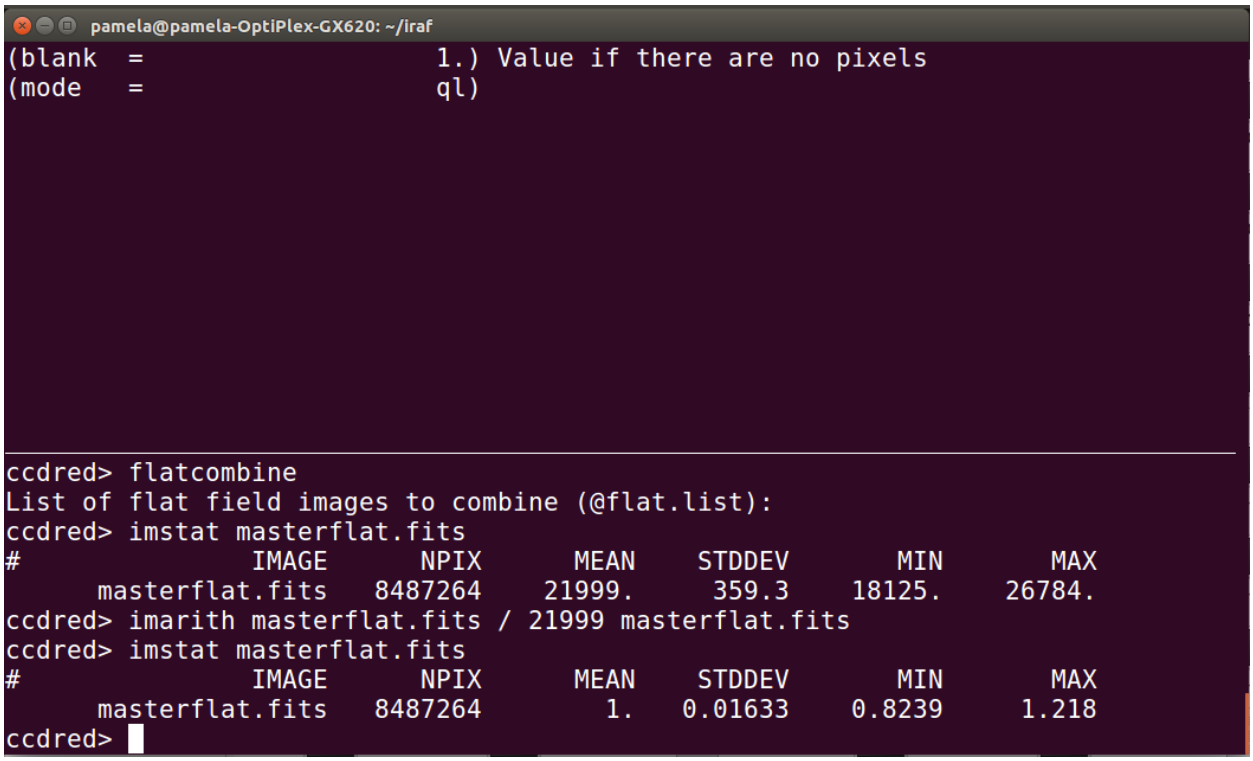

Figure 2.7a: normalization of masterflat using mean pixel value

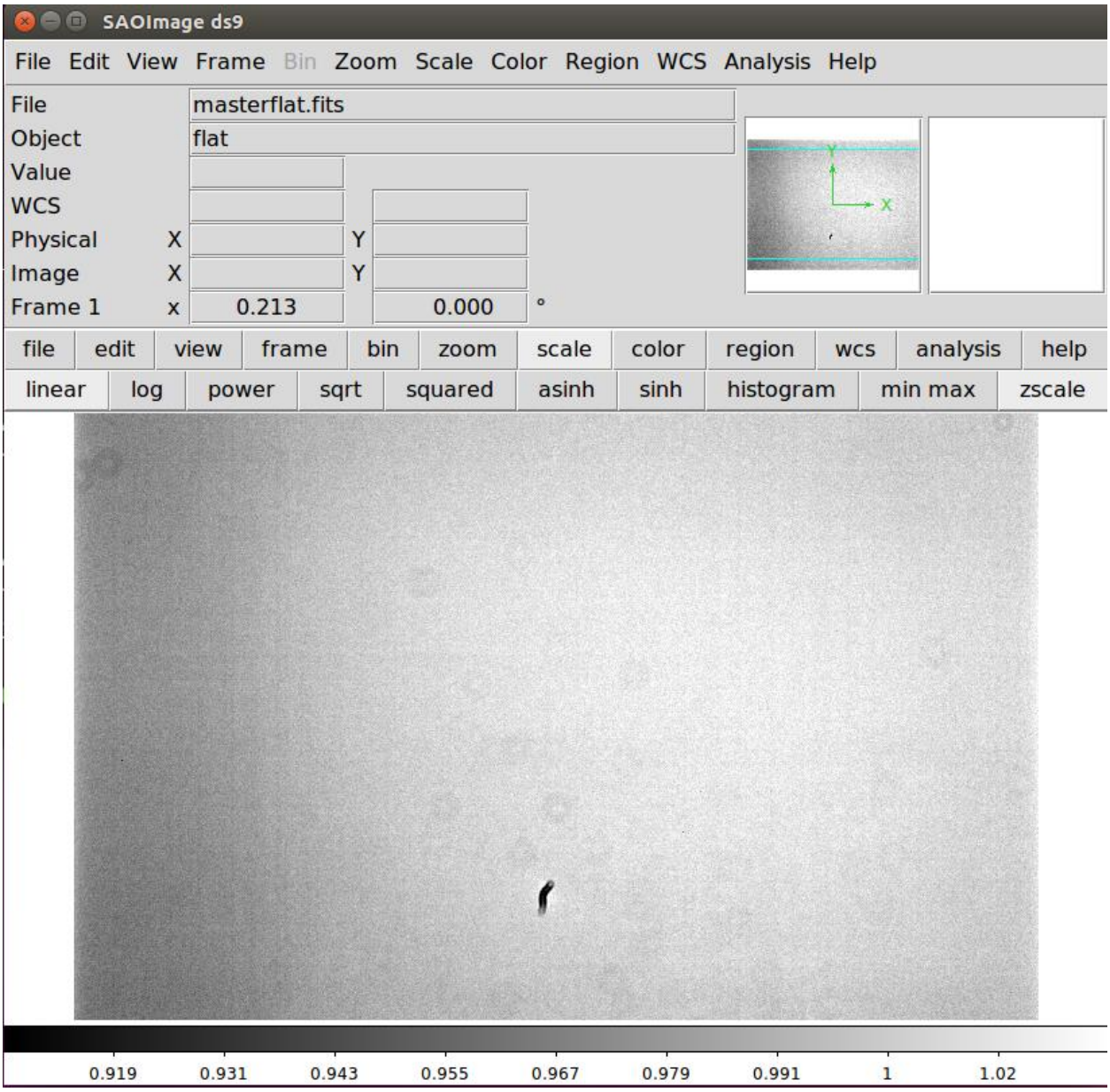

Figure 2.7b: atmospheric flat image used to reduce hatp27. Donut shaped errors are dust deposited on the telescope lens. Here also the space-worm. Images captured on June of 2017

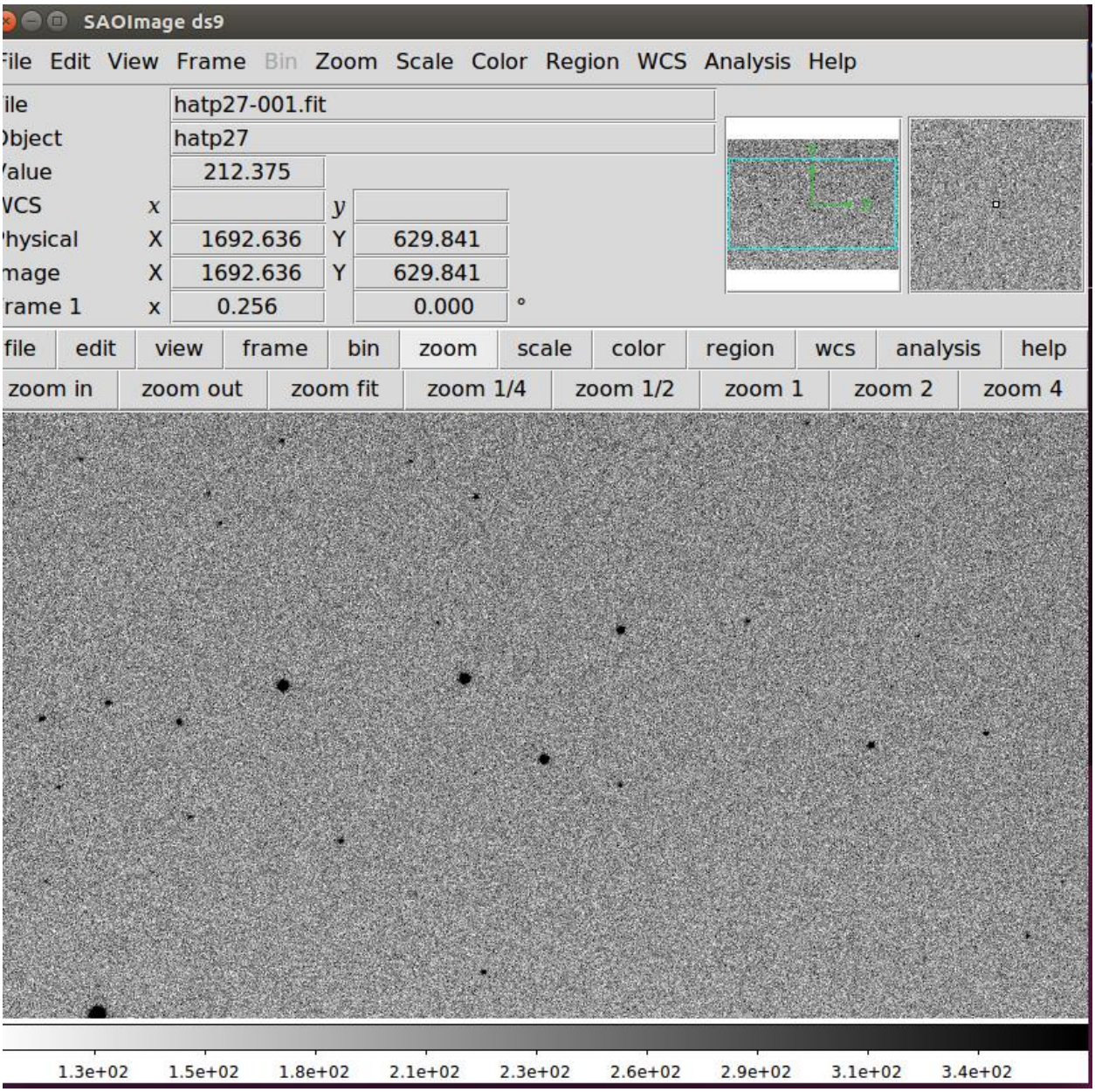

Figure 2.8: Reduced HATP27. No space-worm can be seeing after the images were processed

2.3.2 IRAF

The first round of reduction were done using IRAF. IRAF corresponds to the abbreviations Image Reduction and Analysis Facility, and as it is expected of that name, it provides a wide set of tools for the analysis and the processing of astronomical images. It was

developed in the mid-1980s by the National Optical Astronomy Observatory (NOAO) in Tucson, Arizona. It has become popular amid the astronomical community among other things because it has been selected as the system on which the data reduction and analysis programs of the Space Telescope (HST) are based. Its architecture allows external packages to be added with ease. In fact there are specific application packages that are distributed separately such as STSDAS AND TABLES (developed in the STScI, with many utilities, including tools to process the HST data), PROS (specific for X-ray data)

The IRAF system needs a root directory, it is usually launched from the same directory, and most will create a subdirectory under its main and call it for example “iraf” / home / username / iraf. Once placed in this directory we execute the instruction "mkiraf". This command will ask us what type of terminal we are going to use, and that in general it will be xgterm. In the meantime, two files will be copied: login.cl and loginuser.cl, which contain definitions to start the system. It is convenient to take a look at the loginuser.cl file and check that the variable imdir is set to HDR $ pixels / (set imdir = HDR $ pixels /). A subdirectory named uparm will also be created that will contain the last values of the parameters of each program that has been used in IRAF. In this researcher’s case, IRAF was access using a Linux machine, version 16.4.1. To launch IRAG we use CL,Command Language, and it turns out to be the operating system to access all the facilities of the IRAF system. As the system is started, we will see information about the version, a welcome message and it will also tell us the packages available in our environment. The commands used to perform certain tasks within the cl are called programs, and are grouped into packages according to their functionality ("tasks" and "packages" in the IRAF jargon). In turn, packages are usually organized in other packages of a higher level. A list of the

main packages that make up the core of the IRAF system appears below followed by a few basic image operations we used (Acosta, 2001):

dataio - programs to perform the conversion to different data formats.

images - utilities for the general treatment of images (arithmetic operations, filters, and combinations, among others)

language - the command language itself.

lists - processed lists (lists of names, files, etc.)

noao - contains the majority of programs used in the treatment of CCD data, analysis and calibration of both spectroscopy and photometry.

plot - graphic programs for images of any dimension, manipulation of graphic files ("metacode files").

proto - prototype or developing programs.

obsolete - old programs that due to their great use are maintained.

tv - utilities for image viewing and control.

system - system utilities (copy, delete, rename files),

stsdas - programs developed by the STScI

utilities - miscellany of programs.

Statistics - you can get information about the distribution of values in an image using imstatistic or just imstat,

cl> imstatistic * .imh \

>>> fields = "image, npix, mean, midpt, mode, min, max, stdev" format +

Creation - imexpr is used to create new images according to algebraic expressions defined by us and with the desired dimensions:

cl> imexpr "(I * 2 + J * 10) * 12" res dims = "1024,512"

Arithmetic - to perform binary operations between images, imarith is used. For example:

cl> imarith image1 - image2 res

Combining images - imcombine is a program used to combine any number of images, including an option to project a higher-dimensional image over another with fewer (for example, from a cube to a two-dimensional image). It is very versatile and has many possibilities. The main parameters and their options are described below:

combine = "average | median" # the result of the combination can be the average or the median.

reject = "none, minmax, sigclip, pclip, crreject" # you can discard values that meet certain conditions, maximum and / or minimum values, exceed a typical deviation factor, or using a certain algorithm

masktype = "none, goodvalue, badvalue" # type of mask to use

offsets = "wcs | none | grid" # images can be moved between them before combining them.

scale = "none, mode, media, mean, spouse" # each image can be scaled by a value based on the mean, or the median, etc.

zero = "none, mode, mean, medium" # you can add a value based on fashion, or the average, or the median, to each image.

lthreshold, hthreshold = limits # images are combined by discarding values that are within a selected range.

Processing images with IRAF can be done step by step individually, or it can be done through CCDPROC. The ccdproc program is responsible for performing most of the processing of CCD images, from subtracting the zero level (darks or bias), to correcting flat-field and bad pixels. The first step through ccdproc is to cut the images, subtract a constant zero level for each image, and in its case, also subtract the Zero image that contains the spatial structure of the zero level. In Figure 2.9 shows the list of the ccdproc parameters to do this processing. Using the *epar* module all combinations parameters may be changed to the specific need of the each observatory and image processing.

```
cc> lpar ccdproc
       images = "@allimages.lst" List of CCD images to correct
      (output = "c//@allimages.lst") List of output CCD images
     (ccdtype = " ")             CCD image type to correct
   (max_cache = 0)               Maximum image caching memory (in Mbytes)
      (noproc = no)              List processing steps only?\n
      (fixpix = no)              Fix bad CCD lines and columns?
    (overscan = yes)             Apply overscan strip correction?
         (trim = yes)            Trim the image?
     (zerocor = no)              Apply zero level correction?
     (darkcor = no)              Apply dark count correction?
     (flatcor = no)              Apply flat field correction?
    (illumcor = no)              Apply illumination correction?
   (fringecor = no)              Apply fringe correction?
     (readcor = no)              Convert zero level image to readout correction?
     (scancor = no)              Convert flat field image to scan correction?\n
    (readaxis = "line")          Read out axis (column|line)
     (fixfile = "")              File describing the bad lines and columns
     (biassec = "[1035:1050,6:1024]") Overscan strip image section
     (trimsec = "[3:1024,6:1024]") Trim data section
        (zero = "")              Zero level calibration image
        (dark = "")              Dark count calibration image
        (flat = "flat_19")       Flat field images
       (illum = "")              Illumination correction images
      (fringe = "")              Fringe correction images
  (minreplace = 1.)              Minimum flat field value
    (scantype = "shortscan")     Scan type (shortscan|longscan)
       (nscan = 1)               Number of short scan lines\n
 (interactive = no)              Fit overscan interactively?
    (function = "legendre")      Fitting function
       (order = 1)               Number of polynomial terms or spline pieces
      (sample = "*")             Sample points to fit
    (naverage = 1)               Number of sample points to combine
    (niterate = 2)               Number of rejection iterations
  (low_reject = 3.)              Low sigma rejection factor
 (high_reject = 3.)              High sigma rejection factor
        (grow = 0.)              Rejection growing radius
        (mode = "ql")
```

Figure 2.9: CCDPROC parameters

2.3.3 IMAGEREDUX

TOROS at UTRGV Imageredux, image processing script, is written in python and meant to be part of a fully autonomous pipeline that will rapidly calibrate –also known as reducing or processing- images after one of TOROS telescopes or associated telescopes capture the images pertaining to GW triggers that fall inside of their diverse fields of view. The main contributors to the creation of this script were, at the time, students of the University of Texas Rio Grande

Valley, UTRGV. Figure 2.10 shows the header of Imageredux script with the name of the collaborators and a copyright message.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""
imageredux - Script to reduce images for LIGO counterpart searches.

Copyright (c) 2017-2018, Martin Beroiz, Richard Camuccio, Juan Garcia,
Pamela Lara, Moises Castillo, Adam Zadrożny

All rights reserved.
"""
```

Figure 2.10: imageredux script team

Most of the work was done during the summer of 2017 but addendums and updates continued through 2018. Dr. Martin Beroiz was the key member in the creation of the script. The Imageredux project was under the supervision of Dr. Mario Diaz, director of the Center for Gravitational Waves Astronomy based at UTRGV and TOROS. Using GitHub[5] as the project repository the team members were able to do most of their collaborations (Figure 2.11). In order to build the script several packages (Figure 2.12). This packages included some IRAF tasks that all already proven to do the job. The following are the three packages imported that are astronomy related:

[5] https://github.com/toros-astro

— Importing Astropy: Astropy is a Python library for astronomy that includes many common features in astronomy. Until several years ago, numerous independent astronomical bookstores existed with specific functionalities such as asciidata (catalogs), APLpy (astronomical graphics) or pyfits (images and FITS tables), etc., which were maintained by developers with their own syntax. In 2013, the first public version of astropy was published, grouped in the same API and with common classes, libraries for reading catalogs, coordinates, dates and time, and much more. In February 2018 the version 3.0 of astropy was published, but the version 2.0 is still the version with long-term support (until the end of 2019) and therefore the patched for programs that need special stability such as pipelines or applications of common use and for Python 2 users, since astropy v3 is only available for Python 3 (van Rossum, 2001). There are still no astropy modules for some tasks such as photometry or spectroscopy, but there are the astropy affiliated packages, which do not yet meet all the specifications or are not stable or mature enough to be official packages, but they are usable. Besides offering the main astronomical utilities in the same package, astropy has the advantage of offering a common framework on which to create new functionalities using the common classes that astropy offers.

— Importing numpy: One of the most important Python modules is Numpy. The origin of Numpy is mainly due to software designer Jim Hugunin who designed the Numeric module to provide Python with calculation capabilities similar to those of other softwares such as MATLAB. Subsequently, Numeric improved by incorporating new functionalities born what we know today as Numpy. Numpy is in charge of adding all

the mathematical and vectorial capacity to Python making it possible to operate with any numeric data or array. It incorporates operations as basic as addition or multiplication or other more complex as the Fourier transform or linear algebra. It also incorporates tools that allow us to incorporate source code from other programming languages such as C / C ++ or Fortran, which significantly increases its compatibility and implementation. In order to use this module, if we have installed python (x, y), the first thing we have to do is download the latest version from the official site http://www.scipy.org/Download. In case our operating system is Linux it is probably installed inside Python. If not, we must install it by the usual method. For more information and other operating systems consult the official installation notes. To load the module in Python and to be able to call it to form more comfortable the use of the alias np is of extended use (van Rossum, 2001)

— Importing ccdproc: Ccdproc is is an joined package for the AstroPy package for basic data reductions of CCD images. "The ccdproc package provides many of the necessary tools for processing of ccd images built on a framework to provide error propagation and bad pixel tracking throughout the reduction process." (van Rossum, 2001). The ccdproc package delivers:

- An image class, CCData that includes an uncertainty for the data, units and methods for performing arithmetic with images including the propagation of uncertainties.
- A set of functions performing common CCD data reduction steps (e.g. dark subtraction, flat field correction) with a flexible mechanism for logging reduction steps in the image metadata.

- A function for reprojecting an image onto another WCS, useful for stacking science images. The actual reprojection is done by the reproject package.
- A class for combining and/or clipping images, Combiner, and associated functions.
- A class, ImageFileCollection, for working with a directory of images.

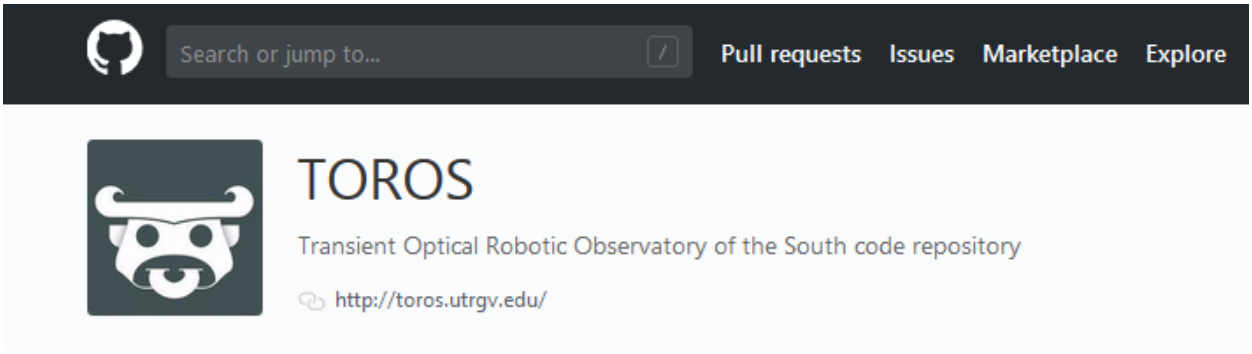

Figure 2.11: TOROS at GitHub website repository service

```
15  from astropy import units as u
16  import numpy as np
17  import ccdproc
18  import glob
19  import os
20  from pathlib import Path
21  from astropy.table import Table, Column
22  import logging
23
```

Figure 2.12: Packages and modules imported to Imageredux

The general tree structure of Imageredux is very simple and straighformard. As the program runs output prints will be made to a log.txt file. This is a big difference with IRAF. This script doesn't need to create master frames in order to process the images. All masters are a

CCData type object so they will remain active as the program runs but will not be created as separated frames. Figure 2.13 shown the directory tree structure and Figure 2.14 is an excerpt from the log.txt that indicates the steps followed by the scrip. Finally a message appears telling the user that Imageredux has finished running. The whole process is clean and efficient and the processed image is as good as an IRAF processed. Figure 2.15, displayed using SAO image DS9 shows calibrated HATP27 frame.

imageredux

Script to reduce images for LIGO counterpart searches.

Usage:

```
$ python imageredux -i path/to/images -o path/to/output/dir
```

Type `python -h` for help on command line arguments.

The script will assume the following directory tree structure:

```
root_path
    |---20170113
        |--- bias
        |    |--- *bias*.fits
        |    |--- ...
        |
        |--- dark
        |    |--- *dark*.fits
        |    |--- ...
        |
        |--- flat
        |    |--- *flat*.fits
        |    |--- ...
        |
        |--- object1
        |    |--- *.fits
        |    |--- ...
        |
        |--- object2
        |    |--- *.fits
        |    |--- ...
        |
   ...
```

Where `root_path` is the input dir for command line argument `-i`. Intermediate files will be saved to a `calib_files` folder in the `-o` argument path.

Figure 2.13: Imageredux directory tree structure

```
INFO - Combining darks
INFO - Writing master dark to disk
INFO - Combining flats
INFO - Subtracting dark from flat
INFO - Writing master flat to disk
```

Figure 2.14: Imageredux log.txt information file

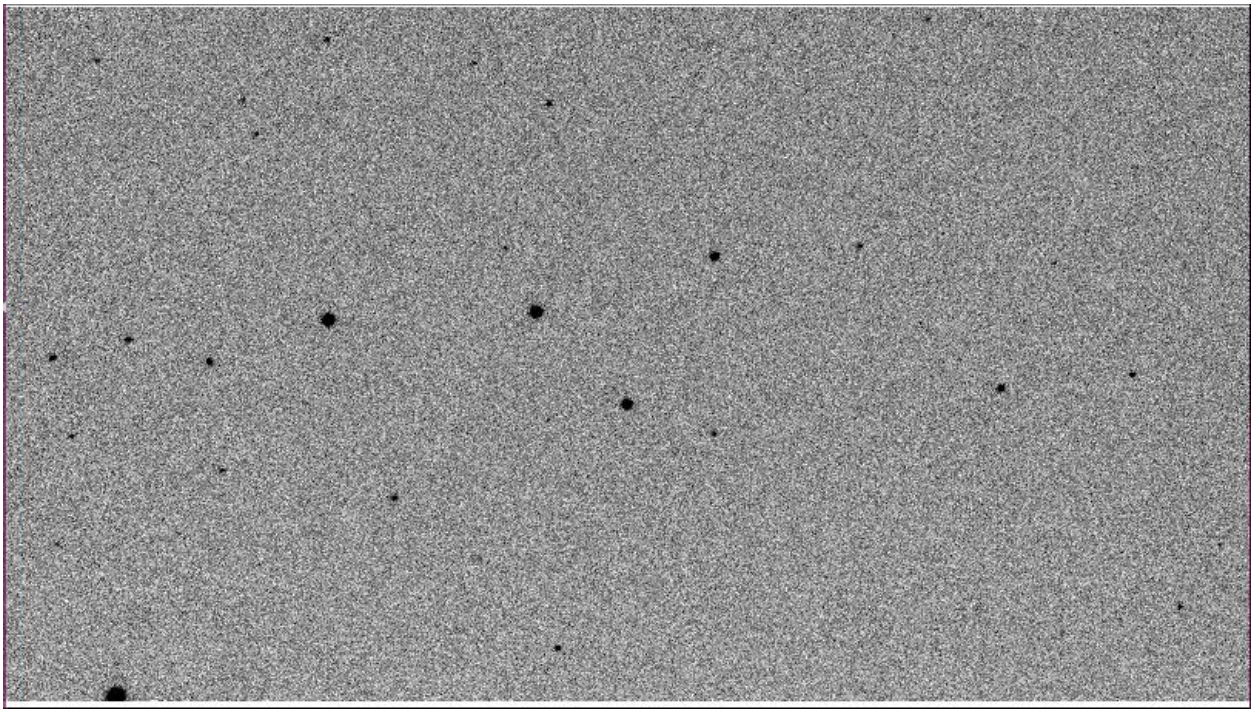

Figure 2.15: A calibrated HATP27 frame after Imageredux is completed

2.4 RESULTS

2.4.1 TOROS OBSERVATION OF SSS17A

Only a few hours following the trigger that gave birth to GW170817, the TOROS group was searching the skies hopping to find an electromagnetic counterpart to the GW. To do this 26 nearby galaxies inside of the region of interest were surveyed during August 17 and August 18.

Two observatories, TOROS affiliated collaboration groups, worked in the search. One of them was the T80-South telescope which is a 0.826-meter telescope optimized for robotic operation situated in Cerro Toloto, Chile; the other is a Meade LX200 16-inch telescope equipped with a SBIG STF 8300 camera, located in Tolar Grande, Argentina (Diaz, Garcia, Macri, 2017). It turned out that indeed there was an EM counterpart to the gravitational signal. The point of origin was determined to be near the lenticular galaxy NGC 4993, in the Hydra Constellation. As the transient optical source, Swope Supernova Survey 2017a (SSS17a) was confirmed, TOROS focused all effort in observing it. The efforts payed off as on the night of August 18, 2017, the T80-South found the source (around 35 hours after the GW trigger). 46 one-minute exposure images were captured spanned over three filters, 16 in g, 15 in r, and 15 in i. The time lapse of the observation were over the course of 80 minutes. Then, on August 19, and around 59 hours after the GW trigger, 88 one-minute unfiltered images were again capture but this time from the 1.54m telescope located at the Estación Astrofísica de Bosque Alegre (EABA) with an "Apogee ALTA F16 camera equipped with a KAF-16083 sensor, consisting of 4096×4096 pixels with a plate scale of 0.″24/pix that yields a FoV of 17′on a side." (Diaz, 2017a). Figure 2.16 through Figure 2.19 show two single images taken by the T80-South telescope on the nights of August 18 and August 19 respectively. Also an excerpt of their header is shown that display some of the ephemeris of each image. We can see that the calibration was done in-situ using CCDPROC IRAF package. Figure 2.20 shows a "color composite of a small subsection ($9'.5$ on a side) of the T80_S field of view…centered on the transient." (Diaz. 2017b). Photometry (which can be defined as the study of the luminous variations of a celestial object), was performed using IRAF CCDPROC packages and modules. A time-series point spread function photometry was carried

out as well. Some of the programs used were DAOPHOT/ALLSTAR, and ALLFRAME provided by Peter B. Stetson (photometry and astrometry software).

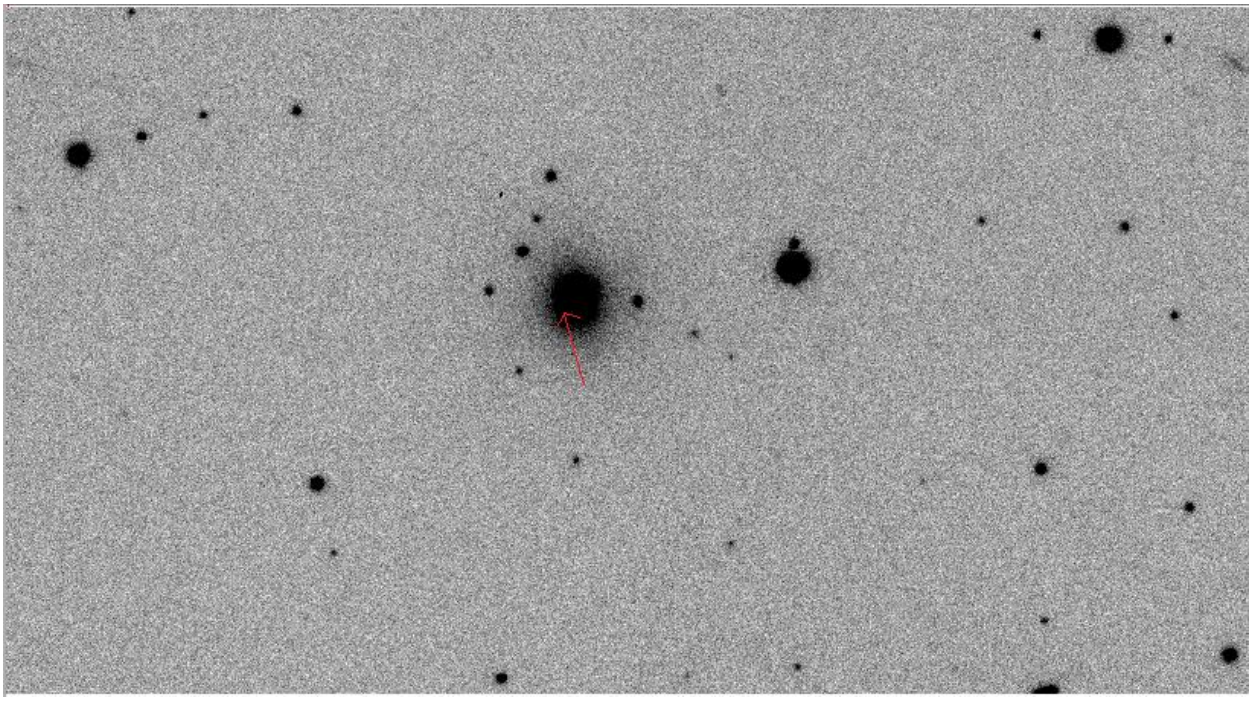

Figure 2.16: Calibrated image of the SSS17a field. Red arrow indicate the signal origin inside NGC 4993. Image in the r filter captured on August 18, 2017. t80Cam. Courtesy of Dr. Nilo Castellon

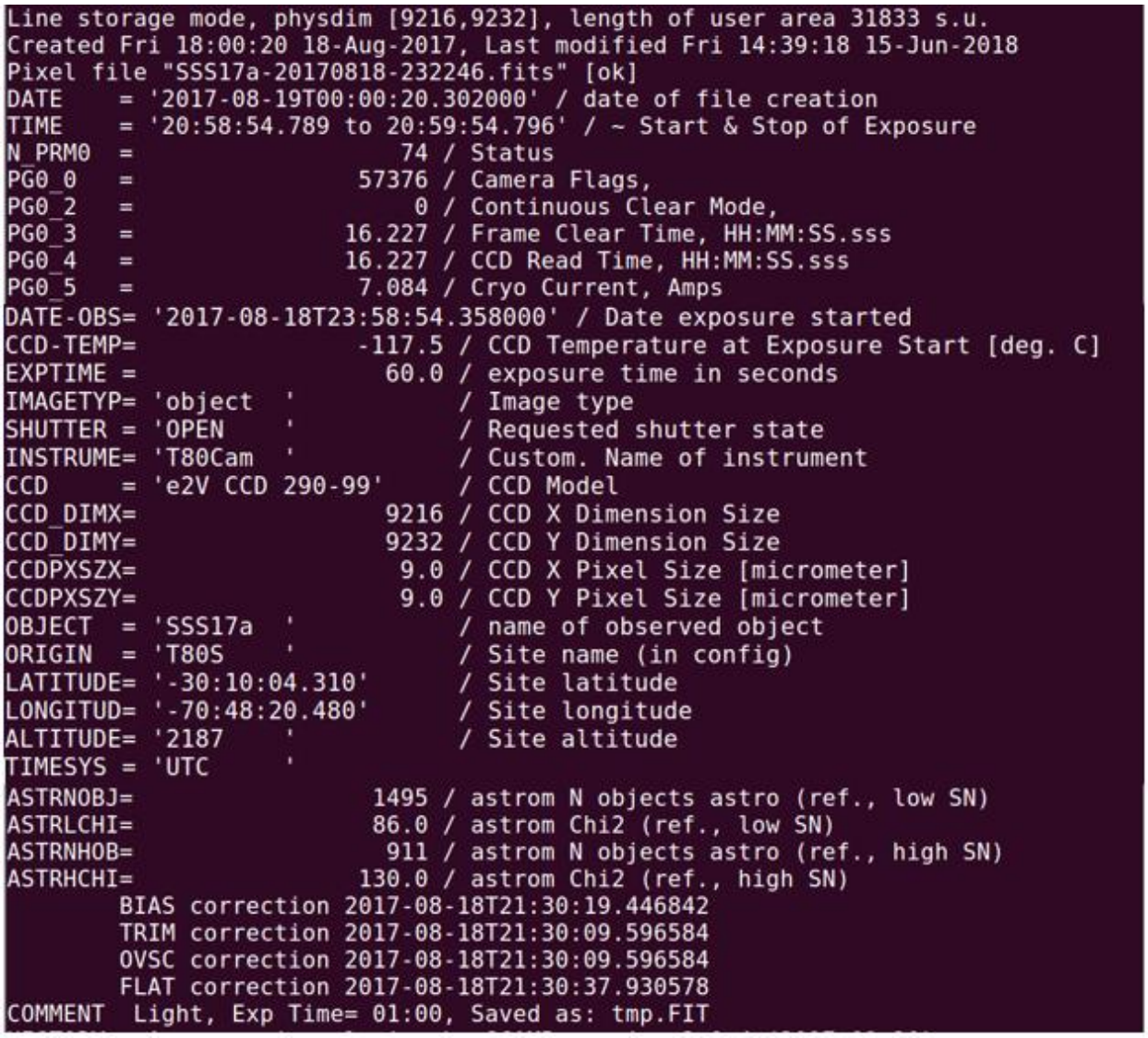

Figure 2.17: Ephemeris excerpt from header (imhead IRAF module) of SSS17a field. R filter. August 18, 2017, T80Cam. Courtesy of Dr. Nilo Castellon

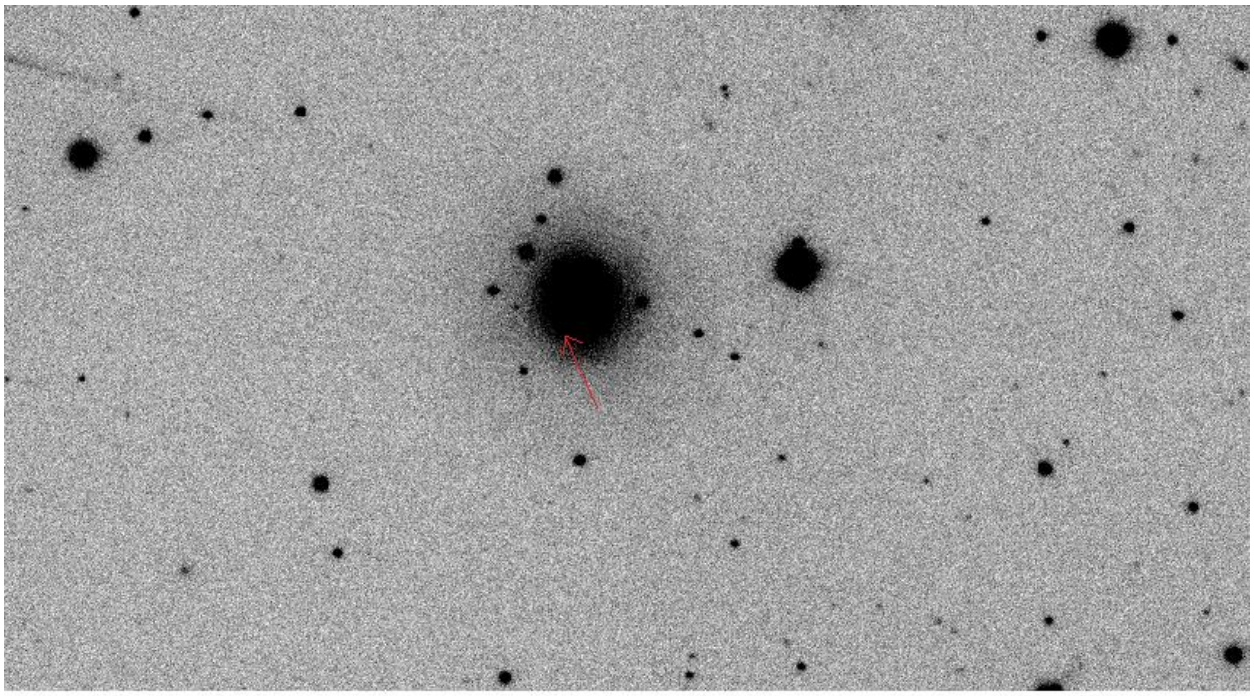

Figure 2.18: Calibrated image of the SSS17a field. Red arrow indicate the signal origin inside NGC 4993. Unfiltered image captured on August 19, 2017. t80Cam. Courtesy of Dr. Nilo Castellon

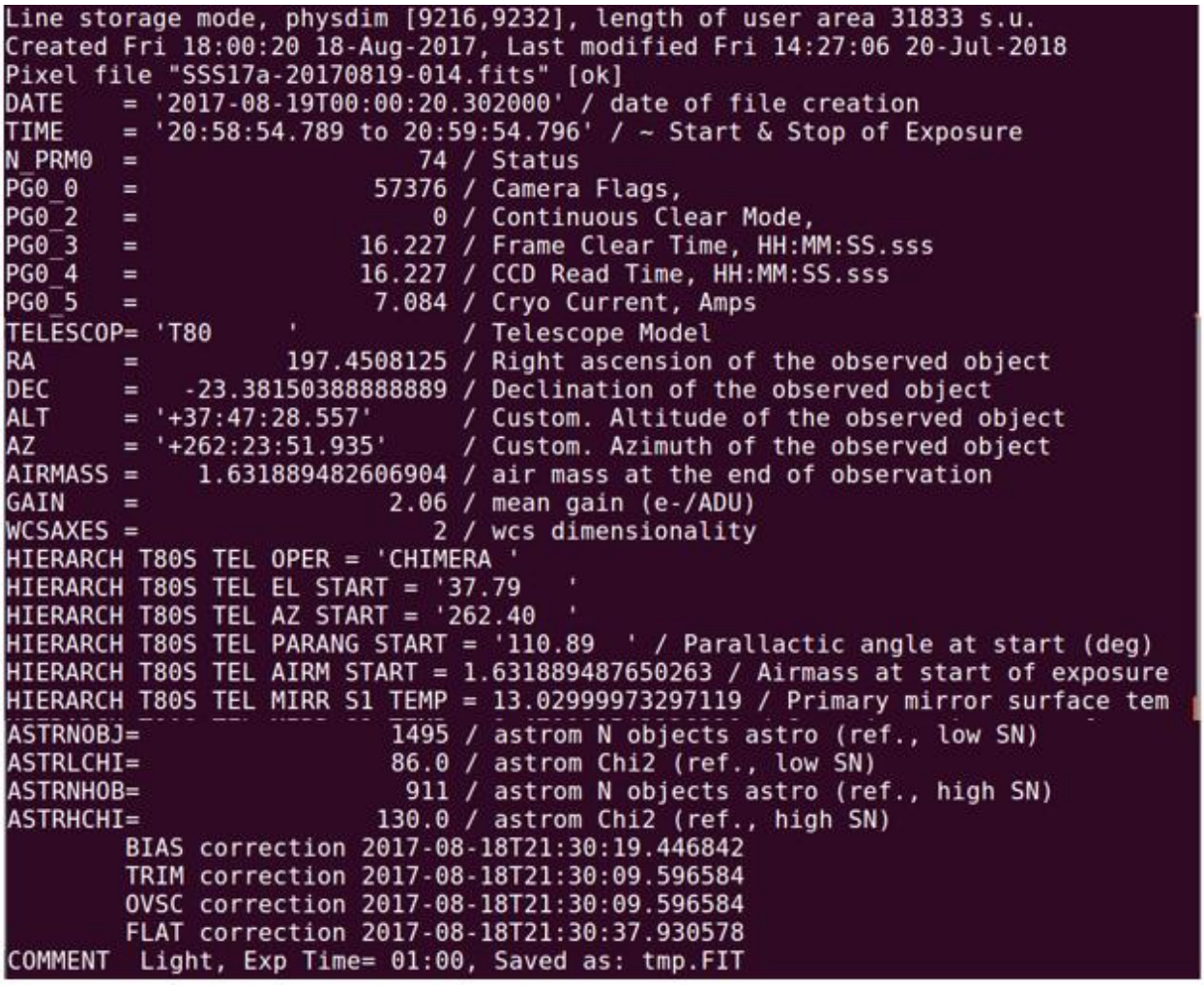

Figure 2.19: Ephemeris excerpt from header (imhead IRAF module) of SSS17a field. Unfiltered image. August 19, 2017, T80Cam. Courtesy of Dr. Nilo Castellon

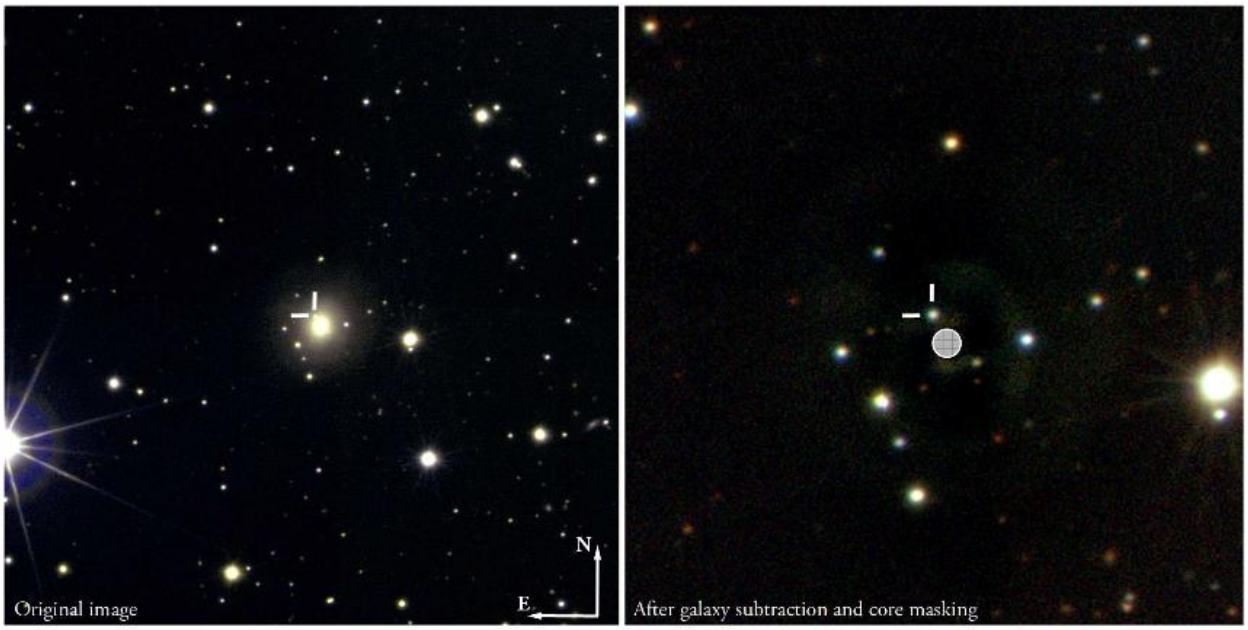

Figure 2.20: "Left: pseudo-color image of a small subsection (9.′5 on a side) of the F.o.V. of T80S, centered on the transient. Intensity scaling is logarithmic in order to better display the light distribution of the host galaxy. Right: 3 × zoom into the residual image after host galaxy subtraction and core masking (hatched circle)." (Diaz, 2017c).

DAOPHOT is an acronyms that describes a set of software written by Peter Stetson (HIA/NRC Canada), intended to do stellar photometry on digital images. "It's specifically aimed at doing high quality photometry on crowded fields" (Harris, 2002). It was written in evolving phases by Stetson from the mid-1980 through the 1990's. Though many other codes for digital photometry were developed after Stetson, DAOPHOT remains highly used and praised. A complete version of DAOPHOT and its companion code ALLSTAR is embedded within IRAF and can be learned through the usual IRAF protocols. For SSS17a photometry these software were used "to detect sources in each image with a significance of 4σvor greater, identify bright, isolated stars, and determine the PSFs. We then used ALLSTAR to obtain preliminary PSF photometry for all of the sources. Next, we used DAOMATCH and DAOMASTER to derive robust geometric transformations between all of the images obtained with a given telescope and

filter (Diaz, 2017). The images were processed, stacked, and evaluated following IRAF protocols.

On October of 2017 TOROS published a scientific paper in The Astrophysical Journal Letter of The American Astronomical Society giving an account of their observation of this first EM counterpart to a GW; the article was title *Observation of the First Electromagnetic Counterpart to a Gravitational-wave Source by the TOROS Collaboration*. The leading author was Dr. Mario Diaz joined by scientist from more than twenty different organization and groups. The following are their results (excerpt), taken from the publication (Diaz, 2017c):

> "Our absolute magnitudes and colors are inconsistent with the predictions of a "red kilonova" model containing dynamical ejecta (0.01 M⊙, v = 0.2c) rich in lanthanides (dotted lines). Our r-band luminosity is in fairly good agreement with the prediction of a "blue kilonova" model with a "wind" (0.01M⊙, v = 0.05c) free of lanthanides (Ye = 0.3 variant in T17, solid lines), but the predicted g – I color of this model at t= 1.5 days does not match the observations. A similar "wind model" (0.01M⊙, v = 0.05c), in which the ejecta contain a small amount of lanthanides (Ye = 0.25 variant in T17, dashed lines), matches the observed colors fairly well, but the predicted luminosities are somewhat lower than observed. Regardless, it appears that this event may have been a "blue kilonova" (Metzger, 2017) relatively free of lanthanides. Intriguingly, there is a low-significance (∼0.1 mag) modulation in theT80S light curve residuals across all bands. If confirmed by others, such a correlated departure from a linear decay may be useful for discriminating among models.

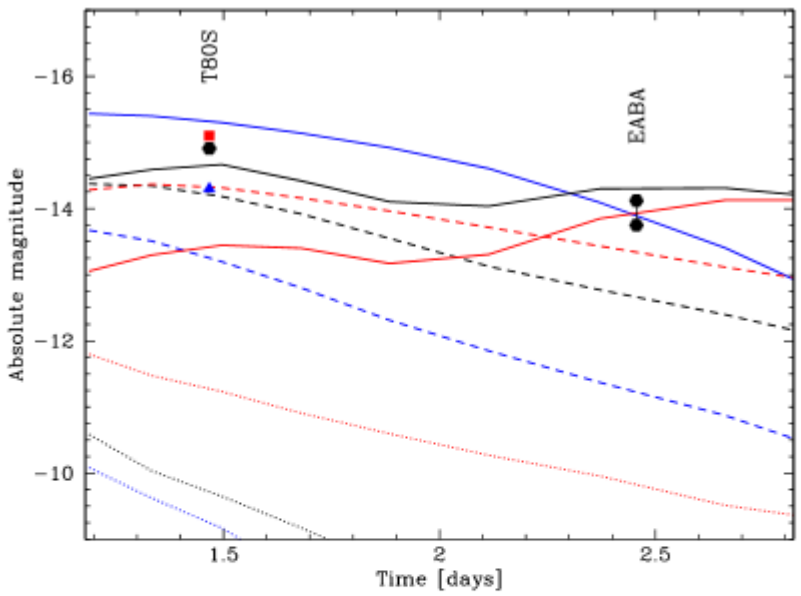

Figure 4. Comparison of our photometry (g: blue triangle; r: black hexagons; i: red square) adjusted to $D = 38$ Mpc with models from Tanaka et al. (2017) plotted using the same color scheme. The dotted lines represent a "red kilonova" model with dynamical ejecta rich in lanthanides. The dashed and solid lines represent "blue kilonova" wind models with decreasing amounts of lanthanides. The measurement uncertainties are smaller than the size of the symbols. The two possible r values at 2.456 days are discussed in §3.

As seen in Fig.4, the light curves predicted by the two "blue" kilonova models exhibit quite different behavior at the time of our observations."

CHAPTER 3: CONCLUSIONS

TOROS UTRGV successfully implemented a calibration python script –Imageredux-. Through comparing the results with accepted methodologies, specifically comparing the results obtained through Imageredux calibration with the calibration over the same images using IRAF packages, we are confident that the script works as desired. For example, where IRAF needed several steps to accomplish a calibration phase, Imageredux was able to perform the whole reduction process by pressing one console key; where IRAF could only be launched from a Linux or Macintosh machine, Imageredux could be accessed from any platform; and most importantly, though Imageredux can function independently, it can be incorporated to our pipeline without any difficulty unlike the IRAF reduction method.

During the time that Imageredux was been created, LIGO sent out a GW trigger –GW170817-. Our associated telescopes in Chile and Argentina were able to detect the EM counterpart –SSS17a- and were able to confirm that the event belonged to a binary neutron star merging-red kilonova-.At TOROS UTRGV we were not able to apply Imagerux reduction method to the images obtained by the associated telescopes because all SSS17a images were calibrated in-situ.

As future work we need to mention that the creation of Imageredux represents only a step in the TOROS pipeline. Other scrips are needed to align –astroaling-, attach celestial coordinates, test real/bogus injections into the system, and to produce qualitative results. TOROS UTRGV students and assessors are working with The Coral Project, in GitHub repository. Coral "is a collaborative effort to improve community on news sites through open-source software."

Future students will need to concentrate into adjoining Imageredux to the pipeline, developing the next generation telescope for the Dr. Christina V. Torres Observatory, and preparing for LIGO 03 data run (projected for early 2019).

REFERENCES

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016a, Living Reviews in Relativity, 19, 1.

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016b, PhRvL., 116, 061102.

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2016c, PhRvL., 116, 241103.

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017a, PhRvL., 118, 221101.

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017b, arXiv:1711.05578 [astro-ph.HE].

Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017c, PhRvL., 119, 141101.

Acosta Pulido, José A. *Reduccion de Datos Astronomicos Usando IRAF*. Diss. Instotuto de Astrofisica de Canarias, 2001.

Bartos, Imre and Kowalski, Marek. *Multimessenger Astronomy*. Bristol: IOP Publishing, 2017.

Beroiz, Martin. *Optical Counterparts to Gravitational Waves.* Diss. University of Texas Rio Grande Valley, 2017.

Berry, Richard and Burnell, James. *The Handbook of Astronomical Image Processing*. Richmond: Willmann-Bell, Inc, 2017.

Boyat, Ajay K., and Joshi, Brijendra K. "A Review Paper: Noise Models in Digital Image Preocessing." Signal & Image Processing: Am International Journal (SIPLJ) 6.2 (2015): 64.

Cervantes-Cota, Jorge L. *et al*., "A Brief History of Gravitational Waves." *Cornell University Library*. ArXiv:1609.09400 [physics.hist-ph], 26 September 2016. Web. July 2018.

Chaudhuri, Asis K. “Gravitational Wave for a Pedestrian.” *Variable Energy Cyclotron Centre*. Kolkata 700.064 (2016):1-10.

Daniel, K. “Einstein Versus the Physical Review.” *Phys. Today* 58 (2005): 43-48.

DeWitt, Cecile M., and Rickles, Dean. “The Role of Gravitation in Physics: Report from the 1957 Chapel Hill Conference.” Edition Open Access. (2011): 271.

Díaz, M. C., Beroiz, M., Peñuela, T., et al. 2016, *ApJL*., 828: L16.

Diaz, M. C., Garcia Lambas, D., Macri, L. M. et al. 2017, GCN, 21619.

Díaz, M. C. et al. 2017a, 2017b, 2017c, ApJL., 848: L29.

Einstein, A.” Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften.” Berlin, (1916): Seite 688-696.

Einstein, Albert and Born, Max. The Born-Einstein Letters. Friendship, Politics and Physics in Uncertain Times. Macmillan: London, UK (2005): 121.

Hanna, C., Mandel, I., and Vousden, W. 2014, ApJ, 784, 8.

Hubble News. “Hubble Observes Source of Gravitational Waves for the First Time.” Hubble Space Telescope, 16 October 2017. Web. July 2018.

Larson, Shane L. “My Brain Is Meltin-GW150914 (Part 2).” WordPress, 23 February 2016. Web. July 2018.

LIGO Caltech. “LIGO and Virgo announce the detection of a black hole binary merger from June 8, 2017.” LIGO, 15 November 2017. Web. July 2018.

LIGO-M1300550, VIR-0540A-15. 6 December 2015.

Magalhães, Nadja S. “Challenges in signal analysis of resonant-mass gravitational wave detectors.” *Brazilian Journal of Physics* 35.4b (2005): 1196.

Manreza Paret, D., Pérez Martínez, A. “Ondas del Espacio-Tiempo: El Nobel de Física.” *Revista Cubana de Física* 34.2 (2017):172-77.

Metzger, B. D. et al. 2017, arXiv:1001.5029 [astro-ph.HE].

Michelson, A. A., and Morley, E. W. “On the Telative Motion of The Earth and The Luminiferous Ether.” American Journal of Science 34 (1887): 333.

Pirani, F. A. E. “On the Significance of the Rienmann Tensor.” *Acta Phys. Polon.* 15 (1956):389-405.

Saulson, Peter R. *Fundamentals of Interferometric Gravitational Wave Detectors*. Singapore: World Scientific Publishing Co. Pte. Ltd, 1994. 2-5, 29-42.

Singer, L. P. and Price, L. R., Farr, B., et al. 2014, *ApJ*, 795, 105.

Singer, L. P. and Price, L. R. 2016, Phys. Rev. D, 93, 024013.

Sosa, Marina. *Breve Introducción al Uso de IRAF*. Diss. Universidad Nacional de la Plata, 2015.

Van Rossum, Guido. “El Tutorial de Python.” Python Software Foundation, 22 June 2001. Web. July 2018.

Weber, J. “Evidence for Discovery of Gravitational Radiation.” *Physical Review Letter* 22.24 (1969): 1321.

Wheeler, John, A., and Ford, Kenneth W. *Geons, Black Holes, and Quantum Foam; A Life in Physics.* New York: Norton. 2000. 235.

Villicaña-Pedraza, I. *Specific results for 3C 454.3. Multifrequency Study of Extragalactic Sources of Gamma Rays*. Diss. New Mexico State University, 2010.

APPENDIX

Imageredux Python Script:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
imageredux - Script to reduce images for LIGO counterpart searches.
Copyright (c) 2017-2018, Martin Beroiz, Richard Camuccio, Juan Garcia,
Pamela Lara, Moises Castillo, Adam Zadrożny
All rights reserved.
"""
__version__ = "1.0a1"
from astropy import units as u
import numpy as np
import ccdproc
import glob
import os
from pathlib import Path
from astropy.table import Table, Column
import logging
# create logger
logger = logging.getLogger(__name__)
_IN_DIR = None
"The path to the images directory."
```

```
_OUT_DIR = None
"The path to the directory where new files will be saved."
def do_dark_combine(dark_list, master_frame_dir):
    """
    Create master dark by median-combining a list of dark images.
    Args:
        dark_list: a list of CCDData objects containing the individual dark frames
    Returns:
        a CCDData object containing the master dark
    """
    out_filename = os.path.join(master_frame_dir, "master-dark.fit")
    if not os.path.isfile(out_filename):
        logger.info("Combining darks")
        # Median combine darks
        master_dark = ccdproc.combine(dark_list, method="median", unit="adu", clobber=True)
        logger.info("Writing master dark to disk")
        # Write master dark to disk
        ccdproc.fits_ccddata_writer(master_dark, out_filename)
    else:
        logger.warning("Skipping dark combine: assigning existing file 'master-dark.fit'")
        # Read master dark from disk and assign to variable
        master_dark = ccdproc.fits_ccddata_reader(out_filename)
    return master_dark, out_filename
```

```
def do_flat_combine(flat_list, master_dark, master_frame_dir):
    """
    Create master flat.
    Args:
        flat_list: a list of CCDData objects containing the individual flat frames
        master_dark: a CCDData object containing the master dark
        master_frame_dir: a string identifying the output path for writing to disk
    Returns:
        a CCDData object containing the master flat.
    """
    out_filename = os.path.join(master_frame_dir, "master-flat.fit")
    if not os.path.isfile(out_filename):
        logger.info("Combining flats")
        # Median combine flats
        combined_flat = ccdproc.combine(flat_list, method="median", unit="adu")
        logger.info("Subtracting dark from flat")
        # Subtract master dark from combined flat
        master_flat = ccdproc.subtract_dark(
            combined_flat, master_dark,
            data_exposure=combined_flat.header["exposure"] * u.second,
            dark_exposure=master_dark.header["exposure"] * u.second,
            scale=True)
        logger.info("Writing master flat to disk")
```

```
        # Write master flat to disk
        ccdproc.fits_ccddata_writer(master_flat, out_filename)
    else:
        logger.warning("Skipping flat combine: assigning existing file 'master-flat.fit'")
        # Read master flat from disk and assign to variable
        master_flat = ccdproc.fits_ccddata_reader(out_filename)
    return master_flat, out_filename
def do_calibrate(object_list, master_flat, master_dark, object_name, cal_frame_dir,
return_fits_objs=False):
    """
    Calibrate a list of images.
    Args:
        object_list: a list of CCDData objects containing the light frames to be calibrated
        master_flat: a CCDData object containing the master flat
        master_dark: a CCDData object containing the master dark
        cal_frame_dir: a string identifying the output path for writing to disk
        return_fits_obj=False: determine, if function should return list fits obj; set to False for
memory saving
    Returns:
        A list of the calibrated CCDData objects (if return_fits_obj=True) and a list of the file paths
where they were saved.
    """
    cal_dir = "cal_{}".format(object_name)
```

```
    check_path = os.path.join(cal_frame_dir, cal_dir)
    if not os.path.exists(check_path):
        os.makedirs(check_path)
    processed_frames, processed_fnames = [], []
    # Calibration begins if directory cal_object exists and is empty
    if not os.listdir(check_path):
        for item in object_list:
            frame = os.path.basename(item)
            logger.info("Reading object {}".format(frame))
            # Read CCDData object
            object_frame = ccdproc.fits_ccddata_reader(item, unit="adu")
            # Check if object frame is same size as master frames
            if not object_frame.shape == master_dark.shape:
                logger.warning("Skipping object calibration... Object frame is not same shape as
Master frames")
                break
            else:
                logger.info("Subtracting dark from {}".format(frame))
                # Subtract dark from object
                object_min_dark = ccdproc.subtract_dark(
                    object_frame, master_dark,
                    data_exposure=object_frame.header["exposure"] * u.second,
                    dark_exposure=master_dark.header["exposure"] * u.second,
```

```
                scale=True,
                )
            logger.info("Dividing {} by flat".format(frame))
            # Divide object by flat
            cal_object_frame = ccdproc.flat_correct(object_min_dark, master_flat)
            logger.info("Writing object {} to disk".format(frame))
            # Write calibrated object to disk
            out_filename = os.path.join(check_path, "cal-{}".format(frame))
            ccdproc.fits_ccddata_writer(cal_object_frame, out_filename)
            if return_fits_objs:
                processed_frames.append(cal_object_frame)
            processed_fnames.append(out_filename)
    # processed_frames is an empty list, if return_fits_obj=False
    # the true value should be used for the tests
    return processed_frames, processed_fnames
def do_file_list():
    """
    Make array of file paths in this directory and all subdirectories, recursively and filter by suffix.
      Assumes path:
       root_path/ObservationDate/objectFolder/objectFrames
    or
       root_path/ObservationDate/randomName/objectFolder/objectFrames
    """
```

```
    file_array = np.asarray(sorted(Path(_IN_DIR).glob('**/*.*')))
    file_array_len = len(file_array)
    # Filters array for specified file suffix
    suffix_search = '*.fit' # Examples: '.txt' '.py'
    # Filter list by suffix
    filtered_list = [file_array[x] for x in range(file_array_len) if
file_array[x].match(suffix_search)]
    file_array_len = len(filtered_list)
    # Parse path to populate table
    object_name = [filtered_list[x].parent.name for x in range(file_array_len)]
    file_name = [filtered_list[x].name for x in range(file_array_len)]
    observation_date = [filtered_list[x].parent.parent.name if
filtered_list[x].parent.parent.name.isdigit() else filtered_list[x].parent.parent.parent.name for x in
range(file_array_len)]
    # Make list into Columns
    object_name = Column(object_name)
    file_name = Column(file_name)
    observation_date = Column(observation_date)
    # Populate Table
    file_table = Table([observation_date, object_name, file_name, filtered_list],
names=('OBS_Date','Object','Frame','Path'))
    # Create groups by similar observation date
    obs_by_date = file_table.group_by('OBS_Date')
```

```
    return obs_by_date
def main():
    # Fancy header
    logger.info("Starting image redux")
    nights_dirs = [os.path.join(_IN_DIR, anight)
                   for anight in os.listdir(_IN_DIR)
                   if os.path.isdir(os.path.join(_IN_DIR, anight))]
    for anight in nights_dirs:
        # Create output paths
        anight_base = os.path.basename(os.path.normpath(anight))
        # Create lists
        bias_list = glob.glob(os.path.join(anight, "bias", "*[bB]ias*.fit*"))
        dark_list = glob.glob(os.path.join(anight, "dark", "*[dD]ark*.fit*"))
        flat_list = glob.glob(os.path.join(anight, "flat", "*[fF]lat*.fit*"))
        logger.debug("bias_list = {}".format(bias_list))
        logger.debug("dark_list = {}".format(dark_list))
        logger.debug("flat_list = {}".format(flat_list))
        if dark_list and flat_list:
            # Create directory to save masters
            master_frame_dir = os.path.join(_OUT_DIR, anight_base, "master_frames")
            logger.info("master_frame_dir = {}".format(master_frame_dir))
            if not os.path.exists(master_frame_dir):
```

```
            logger.info("not os.path.exists(master_frame_dir) = {}".format(not
os.path.exists(master_frame_dir)))
            os.makedirs(master_frame_dir)
        # Create master calibration frames
        master_dark, __ = do_dark_combine(dark_list, master_frame_dir)
        master_flat, __ = do_flat_combine(flat_list, master_dark, master_frame_dir)
        # Create list of object directories
        obj_dirs = [f for f in os.listdir(anight)
                    if os.path.isdir(os.path.join(anight, f)) and
                    f not in ['bias', 'dark', 'flat']]
        logger.debug("obj_dirs = {}".format(obj_dirs))
        logger.debug("(Bool) dir 'master_frame_dir' exists >
{}".format(os.path.exists("master_frame_dir")))
        # Create directory to save calibrated objects
        cal_frame_dir = os.path.join(_OUT_DIR, anight_base, "cal_frames")
        logger.info("cal_frame_dir = {}".format(cal_frame_dir))
        if not os.path.exists(cal_frame_dir):
            logger.debug("not os.path.exists(cal_frame_dir) = {}".format(not
os.path.exists(cal_frame_dir)))
            os.makedirs(cal_frame_dir)

        # Calibrate object frames
        for obj in obj_dirs:
```

```
            logger.info("obj = {}".format(obj))
            object_list = glob.glob(os.path.join(anight, obj, "*.fit*"))
            logger.debug("object_list = {}".format(object_list))
            do_calibrate(object_list, master_flat, master_dark, obj, cal_frame_dir)
        else:
            logger.info("Skipping directory: {}".format(anight))
if __name__ == '__main__':
    import argparse
    parser = argparse.ArgumentParser(description='Arguments for imageredux')
    parser.add_argument(
        '-i', default='.',
        help='Path to root directory where FITS files are. Default is current dir.',
        metavar='DIR',
        dest='input_path',
        )
    parser.add_argument(
        '-o', default='.',
        help='Path where intermediate and final files will be saved. Default is current dir.',
        metavar='DIR',
        dest='output_path',
        )
    args = parser.parse_args()
    _OUT_DIR = args.output_path
```

```
_IN_DIR = args.input_path
logger.setLevel(logging.DEBUG)
# create a file handler and set level to debug
fh = logging.FileHandler(os.path.join(_OUT_DIR, "redux.log"), mode='w')
fh.setLevel(logging.DEBUG)
# create formatter and add it to the logger
formatter = logging.Formatter('%(asctime)s %(levelname)s - %(message)s')
fh.setFormatter(formatter)
logger.addHandler(fh)
    # create a console handler and set level
ch = logging.StreamHandler()
ch.setLevel(logging.INFO)
logger.addHandler(ch)
logger.debug("args = {}".format(args))
logger.debug("_OUT_DIR = {}".format(_OUT_DIR))
logger.debug("_IN_DIR =ls {}".format(_IN_DIR))
main()
logger.info("Image redux complete")
```

VITA

Pamela Lara was born in Chile. She moved to United States of America to stay with her family in 1998. Her Bachelor degree in Physics was completed at Utah Valley University (UVU), in Orem, Utah, where she lived for several years. There her research was done at Brigham Young University under the supervision of Dr. Mike Joner and Dr. Denise Stephens. The area of Ms. Lara's research was in variable stars and planetary science. As a result of her research seven uncatalogued variable stars were found resulting in one publication titled "A New Contact Eclipsing Binary in the Field of KOI 1152" published at Commissions 27 and 42 of the IAU Information Bulletin on Variable Stars Number 6147. After graduating from UVU she worked in a local High school as a Physics, Astronomy, and Spanish teacher.

In 2016 she moved to El Paso, Texas, to continue her graduate studies at the University of Texas at El Paso. She was admitted to the Master in Physics program the fall semester of 2016. On the summer of 2017 Pamela Lara was invited by Dr. Mario Diaz to the University of Texas Rio Grande Valley in Brownsville. There she was incorporated to the Center for Gravitational Astronomy TOROS group to do research in gravitational waves, especially, in the creation of a python script to reduce the images captured by TOROS in their pursuit of the electromagnetic counterparts to gravitational waves.

Contact Information: pilara@miners.utep.edu

This thesis/dissertation was typed by Pamela I. Lara