universität wien

MASTERARBEIT

Titel der Masterarbeit

„Properties of Extragalactic IR Sources Obtained with Herschel Data“

Verfasser

Michael Mach, Bakk.rer.nat

angestrebter akademischer Grad

Master of Science (MSc.)

Wien, 2014

Studienkennzahl lt. Studienblatt:	A 066 861
Studienrichtung lt. Studienblatt:	Astronomie
Betreuerin / Betreuer:	ao. Univ.-Prof. Dr. Werner Zeilinger

Contents

Abstract

In this thesis, the existence and properties of luminous infrared objects in PACS (Photoconductor Array Camera and Spectrometer) images taken by the Herschel Space Observatory are investigated. Their coordinates are measured and aperture photometry is performed on them by using a pipeline called FPS-AP (FITS PREPARATOR for SEXTRACTOR and APERTURE PHOTOMETRY) that has been written for this thesis. A cross-correlation with the NASA/IPAC Extragalactic Database (NED) for each object, infrared colors for filter measurements at 70µm, 100µm and 160µm, and estimates for star formation rates (SFR), redshift and total infrared luminosity (L_{TIR}) for a sub-sample of these objects is provided. The majority of this sub-sample can be classified as luminous infrared galaxies (LIRGs), with a significantly high star formation rate (SFR), confirming the expectation of the nature of these sources.

Abstract

In dieser Arbeit wird die Existenz und Eigenschaften von leuchtkräftigen IR Objekten in PACS (Photoconductor Array Camera and Spectrometer) Aufnahmen des Herschel Space Observatory untersucht. Deren Koordinaten wurden vermessen und Apertur-Photometrie wurde mit Hilfe einer Pipeline namens FPS-AP (FITS PREPARATOR for SEXTRACTOR and APERTURE PHOTOMETRY) betrieben, welche für diese Arbeit geschrieben wurde. Weiters werden alle detektierten Quellen mit NED abgeglichen, IR Farben für Messungen in den Filtern bei 70µm, 100µm und 160µm bereitgestellt, und Abschätzungen für Sternentstehungsraten (Star Formation Rate - SFR), Rotverschiebung und totale Infrarot Leuchtkraft (L_{TIR}) werden für ein sub-sample präsentiert. Die Mehrheit des sub-samples kann als leuchtkräftige infrarot Galaxien (LIRGs) klassifiziert werden, mit einer signifikant hohen Sternentstehungsrate (SFR), wodurch die Vermutung über die Natur der Quellen bestätigt wird.

1 Introduction

1.1 Infrared Measurements in History and Today

Roughly two centuries ago, in the year 1800, infrared radiation was discovered by Frederick William Herschel (Herschel, 1800).
As the story goes, he was experimenting with sunlight in several filters, when one day, he left a thermometer just outside of the optical path of sunlight hitting a prism. By coincidence this happened next to red part of the spectrum, heating the instrument, leaving Herschel only with one conclusion, that there must be a part of solar radiation, that is not perceivable to the human eye.
Nowadays, the infrared regime of wavelength, ranging between the red part of the optical spectrum and sub-millimeter parts of light, is a crucial tool in astronomy if one wants to learn more about the cold universe.
Only small and few windows in the atmosphere exist that allow infrared radiation to reach us.
Telescopes have to be brought to places where the influence of water vapor is small to none, as this is one of main reasons why this part of radiation is blocked at sea level.
The second big factor of influence is thermal radiation by the instrument itself, hence cooling is of an important matter.
Observers are left with the problem of getting the telescope to higher elevation, preferably space, although this is the most expensive option. The other one would be placing the observatory on top of a mountain above ~2000 meters, where at least a lot of the earths atmosphere is located underneath it.
The Infrared Space Telescope (IRAS) (Neugebauer et al., 1984) has been the first space telescope for the mid and far infrared wavelength regime. Launched in 1983, it was operational for ~10 months, before its coolant consisting of liquid helium was depleted. It measured with a 60cm mirror roughly 300.000 IR sources across the whole sky at low resolution in the wavelengths 12µm, 25µm, 60µm and 100µm. This catalog is still of great importance to infrared astronomy today.
The next IR mission was called ISO (Infrared Space Observatory) (Kessler et al., 1996) and was operational from late 1995 to mid 1998. It measured at wavelengths between 2.4µm and 240µm at a spatial resolution between 1.5" to 90" with its 60 cm mirror and 4 instruments, consisting of a infrared camera, a photopolarimeter, and 2 spectrometers.
A third IR mission called Spitzer Space Telescope (SST) (Mainzer et al., 2003) was fully operational from 2003 to 2009. After it ran out of coolant in 2009, it was kept in operation at a higher temperature of ~31 Kelvin, and still keeps delivering scientific results until today. The coverage of wavelength ranges from 3µm to 180µm detected via its 3

detectors consisting of an infrared camera (IRAC), a spectrograph (IRS) and a multiband photometer (MIPS).
These missions delivered a lot of important scientific data to the astronomical community about the cool universe. Therefore large international agencies and organizations such as ESA still introduce so called corner stone missions such as the Herschel Space Observatory (Pilbratt et al., 2010) which was operational from 2009 to 2013 (see section 2 for detailed information).
By analyzing its data we are able to add to the knowledge base in this range of wavelength.
In this thesis, we present a method for the determination of star formation rates (SFR hereafter) from Herschel PACS images at 70µm (blue filter, B hereafter), 100µm (green filter, G hereafter) and 160µm (red filter, R hereafter). The images were obtained by the MESS-group (Mass-loss of Evolved StarS) (Groenewegen et al., 2011) and were originally meant for research on ~140 objects with circumstellar envelopes (CSE). A lot of yet unknown IR - luminous objects showed up on those frames, nurturing further interest and research into the nature of these findings. In order to amass more information for these sources, a semi-automatic pipeline-like approach, with heavy use of the capabilities of the programming language Python and the source-finding program SExtractor (Bertin & Arnouts, 1996) was chosen. The resulting Python script named fps-ap (FITS PREPARATOR for SEXTRACTOR and APERTURE PHOTOMETRY) consists of ~1500 lines of code, and has automatically found and cross-matched roughly 900 objects throughout all frames with the NASA/IPAC Extragalactic Database (NED) (Mazzarella & NED Team, 2007). Subsequently, the sample was processed manually for further analysis, resulting in SFR (Star Formation Rate) estimates according to Calzetti et al. (2010) from monochromatic measurements of model fitted ULIRGS (Ultra Luminous IR Galaxies), L_{TIR} estimates according to Galametz et al. (2013) and photometric redshifts for 22 objects. About 150 B-G colors and 170 B-R colors were calculated for the rest of the sample with at least two detections for the same object in their respective frames. A complete set of measurements for B,G, and R filters has been available for 52 sources from this sample.

1.2 Physical Background

Star forming regions are areas in galaxies where giant molecular clouds of gas collapse to form stars. According to McKee & Ostriker (1977) the interstellar medium (ISM) consists of a 3-phase medium, mainly driven by supernovae. This model describes the ISM of a galaxy by taking into account a dynamical evolution of star formation, cooling and stellar death, among other influencing factors.
By adding disturbances via explosions of massive stars, a cloud of cold gas can become Jeans-unstable (Jeans, 1902) and collapse under its own weight.

1.2.1 The Jeans Mass

Theoretically, the Jeans-Mass can be derived (see Dorfi, 2013) by taking the continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla(\rho u) = 0 \tag{1}$$

the Euler equation,

$$\rho\left(\frac{\partial u}{\partial t} + u(\nabla u)\right) = -\nabla P + \rho g \tag{2}$$

and the Laplace equation.

$$\frac{\partial^2 \psi}{\partial x^2} = 4\pi G\rho \tag{3}$$

By adding small perturbations to the variables $\rho = \rho_0 + \rho_1$, $P = P_0 + P_1$, and $u = u_0 + u_1 = u_1$, (assuming $u_0 = 0$) equations (1), (2) and (3) become

$$\frac{\partial(\rho_0 + \rho_1)}{\partial t} + \nabla((\rho_0 + \rho_1)u_1) = 0 \tag{4}$$

$$(\rho_0 + \rho_1)\left(\frac{\partial u_1}{\partial t} + u_1(\nabla u_1)\right) = -\nabla(P_0 + P_1) + (\rho_0 + \rho_1)g \tag{5}$$

$$\frac{\partial^2(\psi_0 + \psi_1)}{\partial x^2} = 4\pi G(\rho_0 + \rho_1) \tag{6}$$

After linearization, which means that after expanding these equations, we eliminate all terms of higher order, we are left with

$$\frac{\partial \rho_1}{\partial t} + \nabla(\rho_0 u_1) = 0 \tag{7}$$

$$\rho_0 \frac{\partial u_1}{\partial t} = -\nabla P_1 + \rho_0 g \tag{8}$$

$$\frac{\partial^2 \psi_1}{\partial x^2} = 4\pi G \rho_1 \tag{9}$$

It is worth mentioning, that here the so called "Jeans-Swindel" took place, when we assumed $\psi_0 = const.$ Furthermore, we can use $g = -\nabla\psi_1$. We introduce a simple equation of state $P_1 = \rho_1 c_s^2$, where c_s^2 is the isothermal speed of sound. By taking the divergence of the linearized Euler equation we can use the continuity equation to write the resulting form

$$\frac{\partial^2 \rho_1}{\partial t^2} - c_s^2 \nabla\rho_1 - 4\pi G \rho_0 \rho_1 = 0 \tag{10}$$

which is solvable with a wave Ansatz $P_1 = Ae^{i(kx-\omega t)}$, and leads to the following dispersion relation

$$-\frac{1}{c_s^2}\omega^2 + k^2 = \frac{4\pi G \rho_0}{c_s^2} \tag{11}$$

for solutions growing with time, we examine $\omega^2 \leq 0$ and get

$$k^2 = \frac{4\pi G \rho_0}{c_s^2} = \left(\frac{2\pi}{\lambda_J}\right)^2 \tag{12}$$

by taking a sphere with λ_J as its diameter we finally get the Jeans Mass

$$M_J = \frac{1}{6}\pi\rho_0 \left(\frac{\pi c_s^2}{G\rho_0}\right)^{\frac{3}{2}} \tag{13}$$

assuming an ideal gas we can use $P = \frac{\mathcal{R}}{\mu} kT$ and get

$$M_J = \frac{\pi^{\frac{5}{2}}}{6}\left(\frac{\mathcal{R}}{\mu G}\right)^{\frac{3}{2}} T_0^{\frac{3}{2}} \rho_0^{-\frac{1}{2}} \tag{14}$$

In the end, we are left with a rather simple correlation between mass, temperature, and density.

1.2.2 Initial Mass Function

Stars usually form in associations, following an Initial Mass Function (IMF). That means that within a given interval of mass, a certain number of stars are produced from a cloud of hydrogen. One of the best known IMFs has been described by Salpeter (1955).

$$\xi(M) \approx 0.03 \left(\frac{M}{M_\odot} \right)^{\alpha} \tag{15}$$

where $\alpha = -1.35$ in the Salpeter formulation. There are several other values for α for different parts of the IMF (see Kroupa, 2001; Chabrier, 2003). As this is just an informal part of an overview, we will settle for the fact of its existence, and the circumstance that it tells us, that massive stars get created to a certain amount in certain intervals of mass.
So called OB-Associations consist of stars from a single star formation process, with some massive members and lots of smaller ones. As these massive stars have a short lifespan of a few million years, they end quickly as supernovae, feeding the local ISM, thus keeping up the dynamical processes in the ISM.

1.2.3 Supernova Remnants

As said before in section 1.2, supernovae (SN) are the main sources of energy input into the ISM with typical energies of $E_{SN} = 10^{44}$ [J] and SN-rates of $\tau_{SN} = \frac{2-3}{100}$ $[yr^{-1}]$ in our galaxy. According to Dorfi (2014) there is a simple theoretical way of describing the evolution of a supernova remnant (SNR). Without going into further detail, the development of a SNR can be divided into three phases.
The first phase treats the movement of the shock front emitted from the star as free expansion, meaning that the ejected material does not 'care' about the ISM it runs over.
The second phase starts when the mass of the ISM swept up by the shock front becomes comparable to the mass ejected from the star. At this point the so called Sedov-Taylor phase (Sedov, 1959) will partake in the expansion by heating the inner part of the SNR through a reverse shock, reflected by the shock front. This is the pressure driven part of the expansion.
In the third phase radiative cooling takes place and the SNR expands due to inertia. The lifetime of a SNR is finite and short, as it will lose its identity in the ISM after typically about ~10^5 yrs.

1.2.4 Star Formation in IR Galaxies

IG (IR Galaxies) with $L_{TIR} \simeq 10^{11}\, L_{\odot}$ and ULIRGS (Ultra Luminous IR Galaxies) with $L_{TIR} \simeq 10^{12}\, L_{\odot}$ are some of the most luminous groups of objects in the universe, along with quasars. While the origin of the source of energy in these objects is being discussed controversially in lots of papers, we still can assume that it is reasonable to include the following processes. First of all, we saw that star formation needs a disturbance in the ISM, which can be provided by shock waves via a SNR. For a starburst to take place, which is characterized as a strong rise in star formation over a period of time in a galaxy, a larger scale of disturbance is needed. This can be mainly provided by interaction between galaxies called mergers and includes the formation of tidal arms or synonymous tidal tails. On the other hand, there is reason to believe that active galactic nuclei (AGN) can also be of great importance to providing energy in this process. Therefore the bright IR objects found and discussed in this thesis are likely to be part of these groups.

1.3 Overview

In short, the processes in section 1.2 explain the dynamical evolution of the ISM very superficially, and by far not completely. McKee & Ostriker (1977) use more processes like cooling among others in their paper. However, they still rely in their basic conception on these ideas sketched above. Apart from that, the important part is that star formation plays an important role during the lifetime of a galaxy.
The rate at which stars are formed can be measured in various ways, because of different physical processes that come into play during this development.
A star that forms, emits radiation that interacts with its environment. As this environment usually consists of dense clouds of gas it makes direct optical observation rather difficult. The hydrogen atom is one of the most contributing elements in the cooling process in the universe, by absorbing and re-emitting radiation and through ionization.
One tracer of star formation can be Hα at 656,2793 nm (Domínguez Sánchez et al., 2014) in the red part of optical wavelength regime. Another indicator is O/H (Moustakas et al., 2006). This is the ratio between abundances of oxygen and hydrogen. Its value mainly is a measure for metallicity and therefore traces the star formation process indirectly, because regions with a lot of star formation will get enriched with heavy elements more quickly than others. The theoretical idea behind this is that the most massive stars will die in the vicinity of their birthplace, expelling heavy elements locally into the ISM, which in turn, will be converted into new stars, triggered by the instability the shock waves of the SNe create. Therefore a rise in metallicity can be measured and correlated to star formation.

Unfortunately the O/H ratio can only be measured through spectroscopy, which is a time consuming and therefore expensive way of observation, and thus, not available for large amounts of objects. In the end, the most promising part of the spectrum for detecting star formation lies in the infrared (IR) bands. The advantage of these wavelengths is that they are able to pass through dust and dense clouds of gas more easily than the optical light. Calzetti et al. (2010) showed that there is a strong correlation between the star formation rate and light emitted at 70µm and 160µm, among other wavelengths in the IR.

2 The Herschel Space Observatory

The Herschel Space Observatory (Pilbratt et al., 2010) was a 3400 kg infrared telescope launched into space by ESA via an Ariane 5 rocket in May of 2009 and was operational until April of 2013, longer than its planned lifetime of 3 years. The total dimensions of the telescope were 7.5 x 4 x 4 meters. It was brought into an orbit around Lagrange point 2 (L2), about 1.5 million km away from earth, in order to minimize influencing radiation. The optical system consisted of a Cassegrain setup with a 3.5m main mirror, and a 0.3m secondary mirror, detecting wavelengths from 55µm to 672µm through 3 instruments that needed active cooling in the form of ~2300 liters of liquid helium at 2.15 Kelvin. The rest of the telescope had to be shielded against the solar radiation as, due to its size, cooling of the entire satellite was not an option.

Figure 1: *A computer generated view of the Herschel Space Observatory at L2, with adjusted position of shield to block off solar radiation. Picture by ESA*[1]

Some of Herschel´s main objectives were galactic evolution in the early universe, star formation and correlation to the ISM, and atmospheric composition of comets and other objects close to earth.

2.1 Instrumentation

Herschel, formerly known as FIRST (Far Infrared and Sub-millimeter Telescope), carried three instruments in its science payload. PACS and SPIRE being photometric cameras as well as spectrometers, and a high resolution spectrometer named HIFI. In case of further interest, the reader can refer to Swinyard et al. (2010) and Ivison et al. (2010) for detailed insight on SPIRE and HIFI.

[1] *Image courtesy of ESA: http://www.astro.cf.ac.uk/research/astro/instr/projects/?page=spire Visited on 28.09.2014*

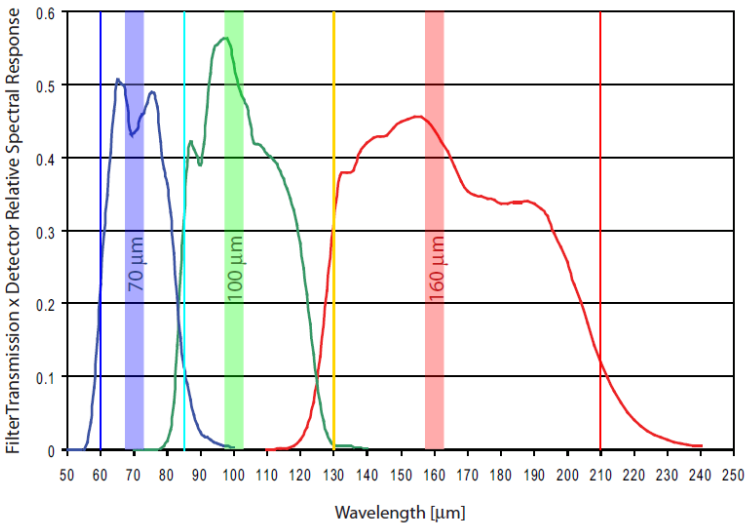


Figure 2: *Filter response curves for PACS taken from Poglitsch et al. (2010).*

2.1.1 PACS

The instrument PACS (Photoconductor Array Camera and Spectrometer) (Poglitsch et al., 2010) was built to be a general purpose instrument for wavelengths between 60µm to 210µm, consisting of a multi-color photometer, and an imaging spectrometer.

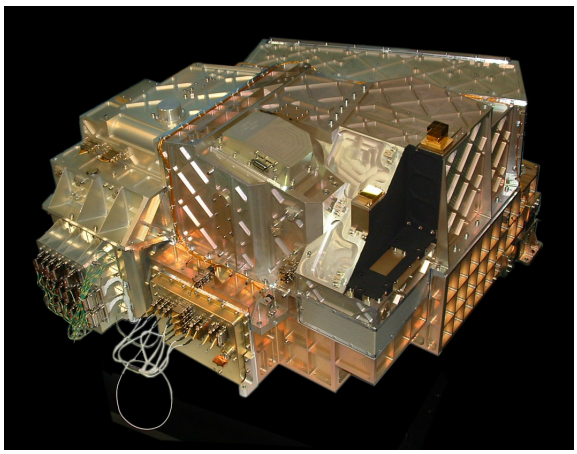


Figure 3: *The PACS instrument. Image taken from Poglitsch et al. (2010).*

When used in photometry mode, PACS used its two 16 x 25 pixel Ge:Ga photoconductor arrays at the same time, enabling it to take images in two of its three color bands simultaneously. In a field of view of ~1.75' x 3.5' imaging was possible in the bands 60 - 85µm, 85 - 125µm, and 125 - 210µm (see Figure 2).
In spectroscopy mode, a field of view over 47" x 47" in 5 x 5 pixels and a spectral resolution of ~175 $km\,s^{-1}$ was achieved, via its two silicon bolometer arrays with 16 x 32 and 32 x 64 pixels.

The most common observation technique for this instrument was called scan map, standard for larger areas in the sky. A scan map consisted of a series of images, taken as the telescope slowly moved its field of view along parallel lines over the area of the target. Three speeds were available for this technique in photometric mode ($10''\,s^{-1}$, $20''\,s^{-1}$, and $60''\,s^{-1}$), and two in parallel PACS/SPIRE mode ($20''\,s^{-1}$, $60''\,s^{-1}$) (see Figure 4).

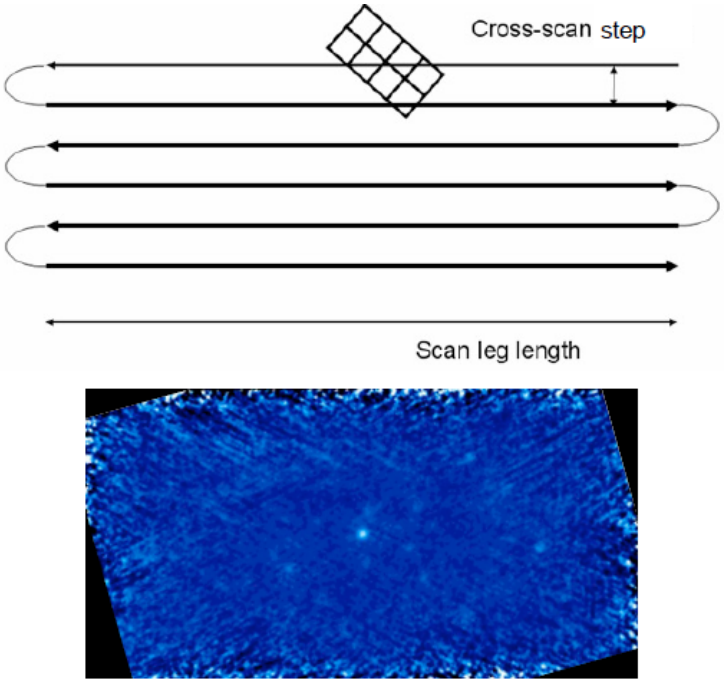


Figure 4: *Top: Schematic path of the telescope field of view over a large area to be observed. Bottom: Image of the scanned area of HD 159330 at 100μm. Image taken from Poglitsch et al. (2010).*

2.1.2 SPIRE and HIFI

The existence of these two instruments will be mentioned briefly here, as only SPIRE was involved in the collection of data for the MESS sample. The objects of interest to this thesis, however, were too faint for reasonable science with this additional spectroscopic data. In case of further interest, the reader can refer to SPIRE (Swinyard et al., 2010) and HIFI (Ivison et al., 2010) for detailed insight.

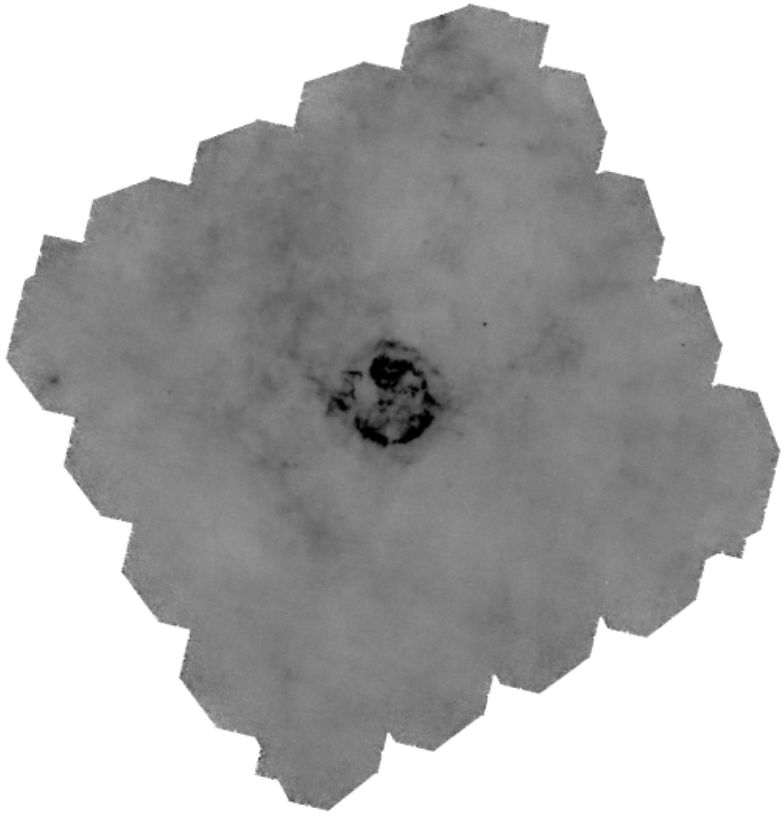


Figure 5: *A typical MESS image. Herschel PACS B-band image of CAS A at 70μm, with a nicely visible central object and several near point-like sources surrounding it. The edges of the frame are irregular due to the imaging mode "scan map", and very noisy due to that technique, as described above and in section 3. Image dimensions of 2030 x 2116 pixels, corresponding to 33.83' x 35.27'.*

3 Data - The MESS Sample

The data used for this thesis has been part of a Herschel key program with guaranteed observing time called MESS (Mass-loss of Evolved StarS) (Groenewegen et al., 2011). The aim of this program was research in the field of AGB (Asymptotic Giant Branch) stars, post-AGB stars, planetary nebulae (PNe), luminous blue variables, red supergiants, SNRs and Wolf-Rayet-Stars. For this undertaking, a sample of ~140 objects has been observed, with additional spectral data for 50 of those objects.

All images for photometry were taken in scan map mode, with the a scan speed of $20''s^{-1}$, and afterwards imaging of the same object orthogonal to the original scan direction was done. The theoretical aim was to produce images of homogeneous exposure time at the center of the target of about 31 seconds.

The flux calibration at 70μm amounted to ~10% and at 160μm to ~15% (see Poglitsch et al., 2010).
The raw image data has been processed by a pipeline based on scanamorphos (Poglitsch et al., 2010; Roussel, 2013). The scanamorphos pipeline is a fully automated algorithm to post-process scan observations from Herschel. Its main features are subtraction of thermal and non-thermal low frequency noise, masking cosmic rays, and projection of the data onto a map. The resulting data consists of a multi-layer FITS (Flexible Image Transport System) (Wells et al., 1981) file with the respective layers for a scan map, error map, total drifts and a weight map. Additionally, the J2000 world coordinate system (WCS) is applied to each frame. The scan map is the science ready map, where all corrections and calibrations are applied. The pipeline processes most of these operations without the need for a noise model or Fourier-space filtering, but solely with the redundant data resulting from observations with multiple instruments at the same time.
The data of the MESS team makes use of a modified version of scanamorphos at different levels adapted to the needs of the MESS dataset (see Groenewegen et al., 2011, Appendix B), due to the fact that Herschel performed observations in a different scan mode than described in section 2.1.1. The objects of interest to the MESS team needed the longest exposure time at the center of the image. Therefore a scan mode named cross-scan was chosen, which combines scan maps of the object taken in the usual scan mode, and then a run across the target area orthogonal to the original scan direction. The described operation continued for several times, on average 3-4 times for each filter. This method requires special care, as the high pass filter is likely to detect bright central sources that are not artifacts. They appear as such due to the comparatively long exposure time relative to the rest of the image. The data itself is not of uniform dimension. The pixel area of the final image varies slightly among the same object in the B, G, R filters, as well as (sometimes greatly) among all the objects observed.
Spectroscopy was also done to a certain extend for a subsample, however, as all objects of interest in this thesis were much too faint for spectroscopically conclusive analysis, further description of data quality in this regard is obsolete.
About ~100 objects had noticeably significant features of mainly point source like sources in the background of various frames of B, G and/or R filters. Further research for this sample has been carried out and is described in the following sections 5 & 6.

4 Tools

4.1 SExtractor

The program SExtractor (Source Extractor) is a widely used and accepted tool in the astronomical society around the world, written by Bertin & Arnouts (1996) that automatically detects sources in astronomical data, preferably in the form of FITS (Flexible Image Transport System) (Wells et al., 1981) files, and performs aperture photometry and various calculations on them. Some of these include aperture photometry, de-blending, background subtraction, star - galaxy classification, and association between objects in separate filters. The photometry SExtractor performs is sufficiently good for a quick first look at the data. However, if one wants to exercise more qualitative research on a few selected objects with known positions, a more accurate way (e.g. HIPE, IRAF etc.) would be advisable.
SExtractor is operated via command-line syntax, and has several modes of output to the screen. The mode used in the code is set to default, displaying a set of information for each individual frame, including the number of objects found in the respective image. If one does not want or need the display output, it can also be set to silent mode, which is more suitable for fully automated pipeline modes.
Over the past few years there were some major and minor updates to SExtractor made public. The version used in this thesis is SExtractor v2.19.5., however portions of the code will also be usable when run with the last stable release SExtractor 2.8.6, that is also implemented in the latest SCISOFT 7.7 package [2]

4.2 Python

Python is a high level scripting language, used throughout various fields of operation. The syntax of the code is characterized by indentation, providing readability on one hand, and eliminating the necessity for semicolons and other line terminators as in other programming languages, on the other hand.
Popular for its versatility, Python provides object oriented programming, as well as procedural and functional programming.
This and the shallow learning curve makes Python a very intuitive, usable and common tool in natural sciences, even though there are languages, that have much better performance regarding speed, like C or Fortran or any other low level programming language

[2] *SCISOFT package at ESO: http://www.eso.org/sci/software/scisoft/ visited on 28.9.2014. Picture provided and copyright by ESO.*

available.
If one wants to go through large FITS files, or large amounts of data in general, performing heavy calculations like fourier transformations, numerical calculations and so on, Python might not be the best fit for this job. If performance is not that much of an issue, or a quick look at a smaller subsample is sufficient, then Python is a nice and handy tool for science.
There are many libraries and plugins available for specific needs in astronomy for Python. A handful of these packages were used in the code, for example pyfits, a plugin for handling FITS files, along with others. The version of Python used in this thesis is 2.7.6. It is imperative not to use newer versions of Python, as some commands have changed with newer releases, and the code will not be executable for this matter.
Python 2.7.2 is also featured in the SCISOFT 7.7 package, which should suffice to run the code as well.

5 Data Analysis

5.1 Goals

In order to extract the information needed to do science from the given frames (see Figure 5), we first need to establish what it is we need to extract.
The goals of this research consist of detection of the positions of all tentatively confirmed real objects, perform photometry on them, if possible calculate color values for our new measurements, and acquire estimates for star formation rates and L_{TIR} where applicable. As all detected objects show significant brightness in the IR wavelength regime between 70µm - 160µm and appear as near point like source with average diameters of 20" to 60", the probability of the majority of these sources being IR galaxies, LIRGS and ULIRGS is a reasonable assumption.

5.2 Automated Steps

A starting point would be taking a look at the raw data, and determining what kind of workload is ahead of us. There are about ~320 frames for ~140 objects that were observed for the MESS sample, and in each frame there are roughly 2-5 interesting sources on average at the first glance. If done manually, this would mean that somewhere between 600 - 1500 positions need to be measured individually. This seems like a tedious task that is meant for automation. So we decide to let SExtractor do the work, and prepare all frames for automation.
In the second step, aperture photometry has to be done for each object, and we need to verify the nature of these sources, thus, we need to set boundaries for a correct detection. Thirdly, a crosscorrelation between the objects detected in their respective frames needs to take place, so we can measure the same object in more than one filter. This is crucial in two ways. For one, this also proves that the detected source is a real object and not some kind of artifact, and for the other and equally important, we obtain the ability to get colors for our sample. In the next step, we put our list of coordinates for these sources into various databases from surveys that are available on the internet (NED), and see if some of the detected objects have already been measured at other wavelengths.

On a side note, it is necessary to mention the still experimental, non functional piece of automated code documented in section 6.1.5, that is part of the program. Consisting of an approach to determine the surface brightness of each object, and calculating the surface luminosity from that, providing the invaluable convenience of being independent of

distance. This part is still actively calculated and printed in the output, although the data does not refer to any real result as of yet. However, this approach is worth pursuing, as it has major advantages, and will be finished after the completion of this thesis.

5.3 Non-Automated Steps

The manual part of analysis commences, as we continue from this point on, by examining only objects found in more than one filter from our sample.
This cuts down the work load dramatically, but needs a more detailed and diverse approach to collecting additional data.
We start by looking up all of the crosscorrelated sources that were found in NED (NASA/IPAC Extragalactic Database) (Mazzarella & NED Team, 2007) and gather as much complementary measurements across the whole energy spectrum as possible. This task uses various databases and catalogs (Sloan Digital Sky Survey (Ahn et al., 2012), WISE (Cutri & et al., 2014; Wright et al., 2010), and 2MASS (Skrutskie et al., 2006)), and is therefore rather complex to code, especially when we look at a sample of about ~20 - 30 objects. The benefit of coding it despite of that, would be the scalability of the sample put into this algorithm, yet the workload however, would exceed the time frame for completion of this thesis.
With the collected measurements, one is able to plot spectral energy distributions (SED) for each object in the sample.
Fortunately, some of the objects have already been measured in SDSS DR9 (Ahn et al., 2012) with information about the spectroscopic redshift available. We therefore can at least correct our measurements for distance, as these catalog redshifts are rather reliable, being extracted from spectroscopic data.
Now we can fit and normalize calculated models for this wavelength regime (Chary & Elbaz, 2001) to our plots, and discern between object types.
With that knowledge, we can correct the rest of our sample that fits the models for distance, as we know the values for fluxes that are corrected for distance and the values for the ones that are not.
At last, we can determine star formation rates from our distance corrected measurements according to Calzetti et al. (2010) at 70µm as well as 160µm and compare the quality of our results, and calculate estimates for L_{TIR} accoding to Galametz et al. (2013).

6 Data Analysis - In Depth

6.1 Automated Analysis in Detail

6.1.1 First Step - File Preparation

Looking at the files from the MESS sample, we find the following file structure:

```
./mess-scanam-21/
├── 3C 58 SN 1181-B-CLEAN/
├── 3C 58 SN 1181-G-CLEAN/
├── 3C 58 SN 1181-R-CLEAN/
├── AFGL 190-B-CLEAN/
├── AFGL 190-R-CLEAN/
├── AFGL 278-B-CLEAN/
└── AFGL 278-R-CLEAN/
```

We need to copy our Python script and the configuration files for SExtractor into the root directory. In this case the root directory is **/mess-scanam-21/**.
The filestructure is inconvenient, as we would like the filter frames for the corresponding central objects to be in the same directory. Thus a piece of code is written to do just that, and as a result we get a new directory with paired files in their respective folders:

```
./mess-scanam-21-paired/
├── 3C 58 SN 1181
├── AFGL 190
└── AFGL 278
```

Now we need to take a look at the FITS files inside these directories and find that SExtractor 2.19.5, as well as previous versions, have difficulties with the filenames with special characters in the following form:

3C 58 SN 1181-B-CLEAN_astro_1.0_pacs 70_0.fits

3C 58 SN 1181-G-CLEAN_astro_1.0_pacs100_0.fits

3C 58 SN 1181-R-CLEAN_astro_2.0_pacs160_0.fits

Therefore, we can rename those files as a quick fix like this:

```
./mess-scanam-21-paired/
├── 3C 58 SN 1181
│   ├── B.fits
│   ├── G.fits
│   └── R.fits
├── AFGL 190
│   ├── B.fits
│   └── R.fits
└── AFGL 278
    ├── B.fits
    └── R.fits
```

Since these are already located inside their correspondingly named folders, the ambiguity among the filenames does not pose a problem. The original files & file structure is left intact in order to provide a full set of untouched, original data. As backing up is always necessary, all files have to be copied. We are therefore in need of at least twice the discsize needed for the raw data.
Lastly, all SExtractor configuration files are copied to the new destination. SExtractor needs these files to be in each working directory it operates on.
Hereby the files have been prepared for further analysis with SExtractor, ending one of the two most time consuming parts of the code.

6.1.2 Second Step - Source Detection and Photometry

We can now start running SExtractor with some command line options to provide check-images for detection and apertures in each frame for each object. A convolution mask called MEXHAT (Cayón et al., 2001) has been chosen for detection of faint and extended sources, as this kind of mask seems to provide the best results.
Aperture photometry for galaxies is not as easy as for stars, because galaxies do not have uniform light profiles. This happens due to a distribution effect of stars in a given galaxy. Furthermore, galaxies do not have sharp edges. We therefore make use of apertures that adopt various approaches to provide some kind of uniformity in our measurements.
We chose two kinds of apertures. One being a Kron aperture (Kron, 1980), based on a luminosity weighted radius of the object.
The other one is a Petrosian aperture (Petrosian, 1976). This aperture is defined by the shape of the averaged light profile of an object, and determines its radius with a certain value as a threshold.

The reason for this dual system emerges from the idea to provide a comparison between aperture types when using the still experimental and not yet functional part of the code to perform surface photometry, with benefits listed in section 5.2. In the following analysis we chose to use the petrosian aperture type, for the obvious reason that it encloses 90% of the light of an object (see Petrosian, 1976). We can take a look at one of the check images, in order to confirm that these settings are valid.

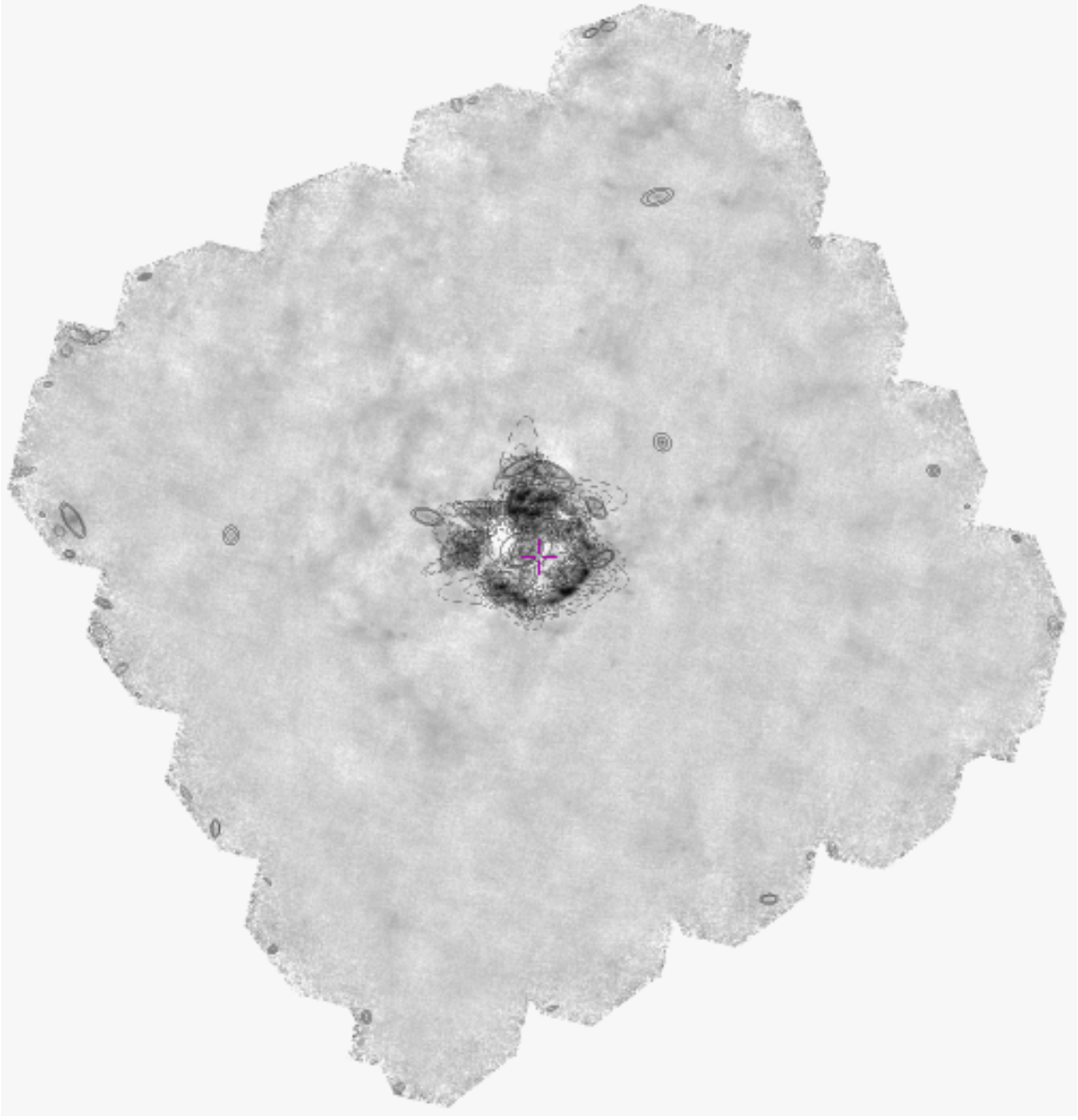


Figure 6: *A typical check image of CAS A from the MESS sample, with Kron (inner ring) and Petrosian (outer ring) apertures drawn for each detected object. Notice the amount of objects found at the edges and in the place of the central, resolved and extended source. Image dimensions of 2030 x 2116 pixels, corresponding to 33.83' x 35.27'.*

We can confirm that SExtractor has detected almost all objects of interest, yet there are a lot of artifacts and wrong detections as well. It is a difficult task to provide an algorithm to get rid of objects at the center because the central source does not cover the same pixel area in each filter. Even worse, the different objects do not cover the same area among each other.

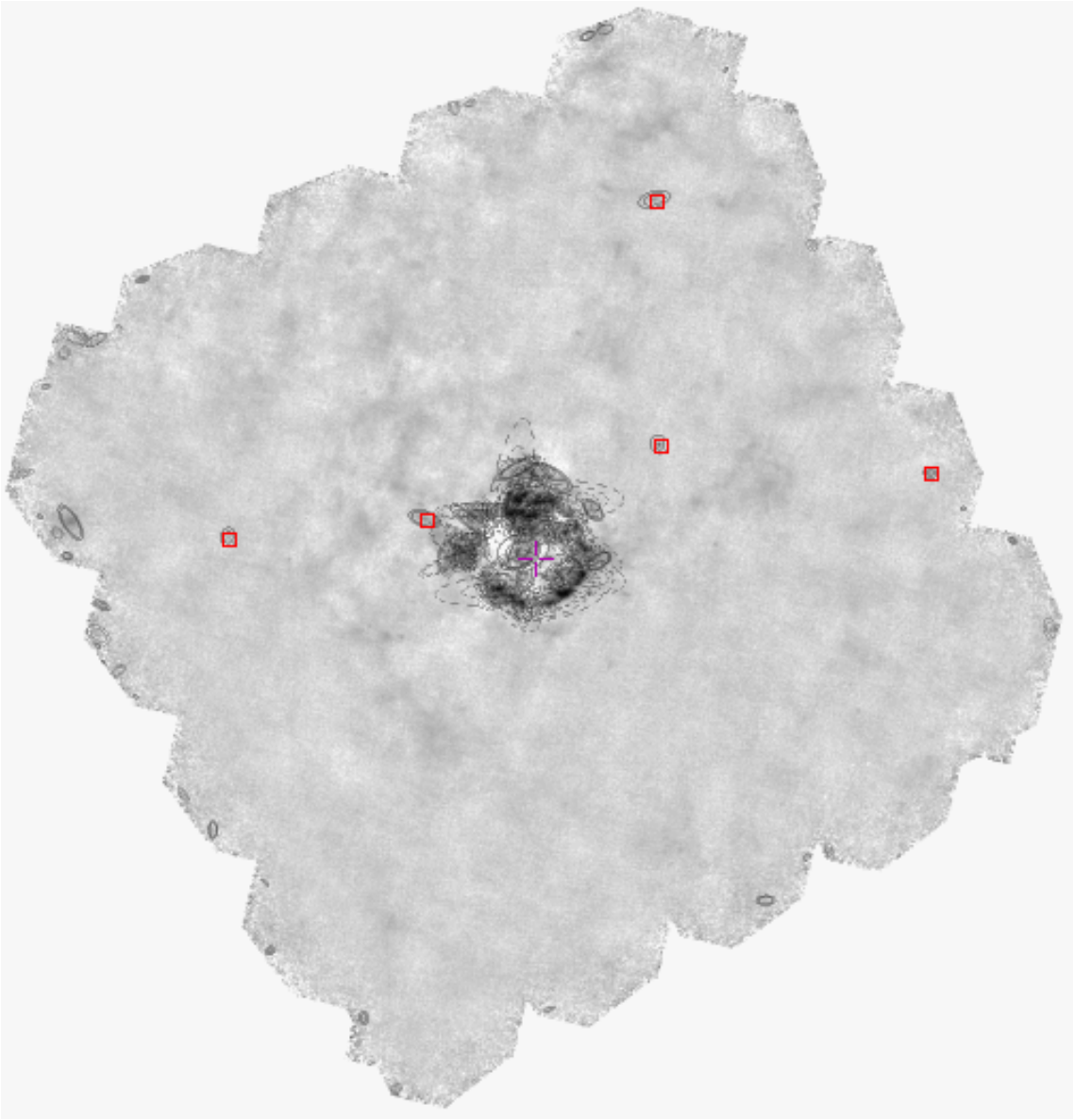


Figure 7: *Masked check image of CAS A, with original apertures drawn, and red rectangles as objects that were not excluded by the masking algorithm. Image dimensions of 2030 x 2116 pixels, corresponding to 33.83' x 35.27'.*

The idea is to relate the extension of the central object to the total dimension of the frame. We find that the central source roughly covers f = ~10 % of the total image size. A simple relation that we can code and put into the script looks like this:

$$r_{mask} = \frac{(x_{dim} + y_{dim})}{2} \cdot f \tag{16}$$

We can call this the radius of the central mask for the sake of simplicity, and define the radius of an object as:

$$r_{object} = \sqrt{(x_{object} - x_{center})^2 + (y_{object} - y_{center})^2)} \tag{17}$$

For every object the relation $r_{object} > r_{mask}$ needs to be fulfilled.
We can now turn our focus onto to the edges of the frame, which are very noisy as implicitly discussed in section 3.
This task seems to be a bit more challenging. The irregular shape of the edge makes it difficult to determine which source to keep, and which one to cut. We do not want to cut down the frame until we have a rectangular piece of the original image that is smooth, as this will exclude too much valuable information. A lightly advanced idea is to define a radius around each source, and throw out detections where the radius vector passes areas outside of the image. For performance reasons, we do not ask for the whole circle in the area, but only use directions orthogonal and in the direction of the position vector of the object itself.
We look at the position vector of the object relative to the center and calculate each component with a set factor (here the factor is set to g = 90) as an extension in order to test for adjacency to the border.

$$\begin{aligned} x_{outer-thresh} &= x_{object} + \frac{(x_{object} - x_{center})}{r_{object}} \cdot g \\ y_{outer-thresh} &= y_{object} + \frac{(y_{object} - y_{center})}{r_{object}} \cdot g \end{aligned} \tag{18}$$

We can also define the orthogonal direction of that position by

$$x_{outer-perpen} = x_{object} + \frac{(-y_{object} + y_{center})}{r_{object}} \cdot g$$
$$y_{outer-perpen} = y_{object} + \frac{(x_{object} - x_{center})}{r_{object}} \cdot g \tag{19}$$

And complementary the other orthogonal direction

$$x_{outer-perpen-other} = x_{object} + \frac{(y_{object} - y_{center})}{r_{object}} \cdot g$$
$$y_{outerperpen-other} = y_{object} + \frac{(-x_{object} + x_{center})}{r_{object}} \cdot g \tag{20}$$

If any of these positions deliver values of $-1.0\,x\,10^{30}$, which is the value that SExtractor sets in its check images for regions outside of the frame, the object is excluded from the sample.

The screen output of this part of code also features the number of objects found with masking in relation to total object number found for each frame by SExtractor. This offers a quick first look at the stability of detection. We expect 0 to ~5 objects of interest in each frame from a quick first visual inspection of the data. If we are confronted with values outside of that range, the code offers an easy way of making our inner masking radius larger or smaller by altering the last value (f) in equation 16. Likewise we can configure the radius for detection of objects at the edge by modifying the last value (g) in equations (18), (19) & (20).

We end up with a robust routine to automatically mask unwanted regions of a frame in the MESS sample (see Figure 7), and collect only interesting detections, that are now ready for further analysis.

6.1.3 Third Step - Matching Detected Sources

This optional part of the code uses the ASSOC mode of SExtractor. It is designed to measure sources in one filter frame, and then use those positions in other filter frames. Obviously this is of great use whenever information about the color of an object is required. As none of the resulting output is used in this code, this feature is strictly of informal nature to the user. It has been included as a tool for testing the manually established results of analysis, but is not included in these methods. It is deliberately put out of common operation in the code.
If the need for it ever arises, it is still available to the user.

6.1.4 Fourth Step - Crosscorrelation with NED

Our list of mask-filtered objects has now to be run through an online database for further information. Several astronomical databases feature automated queries via codes.
According to our assumption about the nature of the detected objects, we can safely say that the online database NED (NASA/IPAC Extragalactic Database) (Mazzarella & NED Team, 2007) is a reasonable starting point.
This portal offers some plugins as well as url based queries for automation. The later has been chosen for this code, and is used as follows with the only restriction that crosscorrelation will take place within a radius of 9" at most. This is chosen due to the fact, that the pointing of Herschel was not better than 4".
The output from the last step of the code consists of separate files with measurements for each detected object in each line. The code provides a list of all objects featured in all frames at 70µm , 100µm and 160µm individually.
As we encounter some limitations of the portal and the Python plugin named *urllib2*, we are obliged to work around that.
The first problem arises from the portal interface of NED itself, by limiting the query for correlating object positions for 500 objects at most in one run. This, however, is cut in half by said plugin. Our Python routine can handle 250 sources at most at a time. So we start off by splitting our big files for each filter into smaller ones, with the corresponding number of entries each.
Now we can feed those lists into NED and get the whole HTML source code as output. With basic knowledge of the structure of HTML we can quickly eliminate all unwanted information and leave only an ascii list of bar separated values for further inspection. We then format these results in the way we need it, and save them in a file.
By having collected the results, we are ready to put the split files from our own detections

back together into one big one for each frame, and start correlating the results from NED to our detections of object positions.
The correlation between original positions and NED detected ones is done by comparing each set of coordinates with the other. The NED output is formatted in a way that prints the input set of coordinates in the line with the closest separation in arcseconds (") into its output, and the rest of matches that have larger separations without the original coordinates beneath them. This enables us, to always get the closest match quite easily, by looking for the input coordinates in the respective output column of NED. After formatting our new correlated output file nicely into one big file for each frame we can now merge these files and collect each filter measurement for each position into one big master file in our root directory called *master_table_SFR_complete.cat*.
Finally we have everything we need for manual analysis.

It has to be mentioned that there is also a part of commented code by default, that enables the user to decide to delete temporary files, that are not necessary anymore. It is wise however, to keep those files, in order to check if all steps of detection and analysis are done correctly, and therefore to find out what went wrong at which step if problems arise.

6.1.5 (Fifth Step) - Experimental Surface Photometry

As an elegant approach to this topic, we initially chose to adopt surface photometry for our research. The advantage of this features the ability to not have to care about distance effects in the data. With the majority of the objects at an angular size of ~20" - 60" we are scratching the limit of reliability of this technique.
Unfortunately, there seems to be some fundamental problem in the formulation of the reduction method and therefore this part of the code has been tagged with comments and warns the user that the output of this routine is not finished yet.
The steps we take in the code are calculation of surface brightness of the detected objects for measurements in Kron and Petrosian (Kron, 1980; Petrosian, 1976) apertures, which seems to work nicely. We acquire a surface brightness by applying the following equation

$$\mu_{SB} = m_{obj} + 2.5 \cdot (Area) \tag{21}$$

The next step is the conversion of surface brightness to physical units $L_{surface}(\lambda)\,[L_{\odot}\,kpc^{-2}]$ as follows

$$L_{surface} = 10^{-0.4\,(\mu_{SB} - M_{\odot} - 21.572)} \cdot 1000^2 \tag{22}$$

The last term converts the units of the result from $[L_{\odot}\,pc^{-2}]$ to $[L_{\odot}\,kpc^{-2}]$. We also need to apply a value to $M_{\odot}$ which should be the bolometric absolute brightness of the Sun at $M_{\odot,\,Bol} = 4.74\,[mag]$, yet by doing so, we generate values of ~$10^{20-30}\,[erg\,s^{-1}\,kpc^{-2}]$ difference to the expected range of results of ~$10^{38-44}\,[erg\,s^{-1}\,kpc^{-2}]$. There must clearly be a mistake either in the conversion of units or in the formulation itself.
Thus the values calculated do not represent real results.
This additional part of the code runs in the background and parts of its calculations are also printed in the final output file mentioned above.

6.2 Non-Automated Analysis in Detail

6.2.1 First Step - Object Confirmation

In the first step of non-automated analysis, we take a look at our automatically generated master table and filter for entries with more than one measurement in the various filters. As a reference we use the B filter detections at 70µm, because this has been found to be the most reliable wavelength for detection of IR objects, as influencing radiation of other processes has the least effect at this wavelength. However, we only take into account object detections in at least two filters, as there can still be a lot of spurious artifacts of dust and molecular clouds that lead to wrong detections, especially at larger wavelengths i.e. the R filter at 160µm. We find 52 B-G-R-correlations, 151 B-G-correlations and 173 B-R-correlations (see Table 5, 6 & 7).

6.2.2 Second Step - Collecting Additional Data

As a starting point, we will now inspect the objects that are most likely to refer to real objects. In order to do that, we tighten the conditions to detection in all three filters. Out of the rather small sample of 52 objects we find that 7 positions were crosscorrelated to entries in NED. It is obvious now that this truncated list of objects really is possible to be handled manually.

By querying these objects via Vizier[3] a lot of existing data is presented, and homogeneous surveys for all objects have to be chosen. We take into account the SDSS survey (Ahn et al., 2012), for the availability of information in SDSS filters (ugriz) covering visual wavelengths between 374 nm and 890 nm, and, if fortunate, even reliable spectroscopic redshifts. The 2MASS catalog (Skrutskie et al., 2006) is used for wavelengths between 1.25µm and 2.15µm. Another survey suitable was WISE (Cutri & et al., 2014; Wright et al., 2010), delivering measurements in the near infrared wavelength regime between 3.4µm and 22µm. From there on, our own measurements continue the data into the far infrared between 70µm and 160µm.

After a first look at the SED of these objects, we can assume our approach delivers correct results and continue by querying the rest of our data that has been identified by NED via Vizier. In total we find data for 22 objects, and extend the number of objects with redshift information to 7 hits.

[3] Vizier online: http://vizier.u-strasbg.fr/viz-bin/VizieR visited on 28.9.2014

6.2.3 Third Step - Correcting for Distance

Now that we have a small sample of redshifts, that seem to belong to the same class of object, we can start correcting the rest of the sample for distance that fits the same pattern. By taking the relation for cosmological redshift

$$z = \frac{v}{c} \tag{23}$$

and using the equation

$$v = d \cdot H_0 \tag{24}$$

where d is the distance of the object in Mpc.
We adopt the hubble constant to be $H_0 = 100\,[km\,s^{-1}Mpc^{-1}]$ for the sake of simplicity. After some simple operations we find that the distance can be formulated like this

$$d = \frac{c \cdot z}{H_0} \tag{25}$$

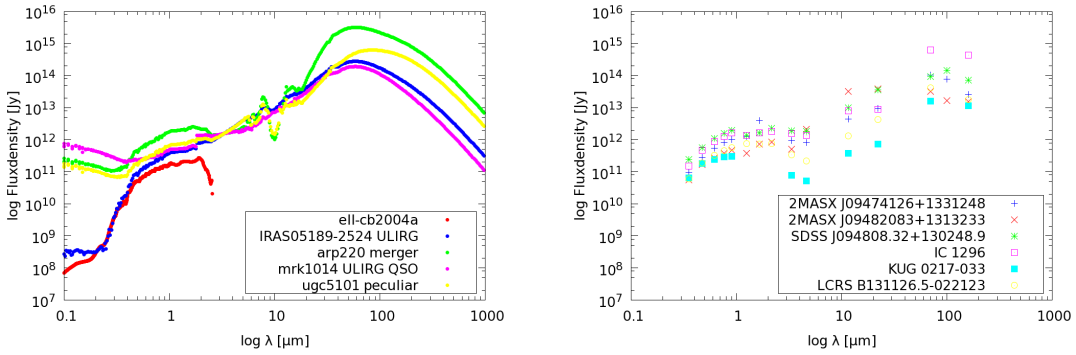
For each redshift the distance is calculated, and the associated data is corrected for cosmological extinction. We also put this method to a test by calculating distances for the already known sample with SDSS redshifts, and derive correct distances up until the 2nd decimal point. That value confirms quite obviously that the method is adapted correctly. This leaves us with the opportunity to calculate distances from the distance module for the rest of the sample, as we now know the absolute and apparent magnitude for the same type of object. The basic distance module without extinction is defined as follows

$$m - M = 5\,log(d) - 5 \tag{26}$$

Having calculated the distances for the rest of the sample, and applied distance corrections to them as well, we can continue to the next step.

6.2.4 Fourth Step - SED and Modeling

This makes plotting a distance corrected SED for each of these objects possible. We find a significant shape among the individual objects. Apart from the visual part of the spectrum, we can neglect effects of extinction by dust for our intentions. This is not the most reliable approach if one wants to obtain precisely measured, qualitative results, but is sufficient for a quick classification and overview, which is our goal for the moment. By following our initial notion about the nature of these objects, we try adopting calculated models for IR galaxies for this range of wavelengths between 374nm and 160µm according to Chary & Elbaz (2001).



(a) (b)

Figure 8: *(a) showing model data by Chary & Elbaz (2001). The models are normalized to a flux of IC 1296 at 3.35µm. Additionally we plot the model SED of an elliptical galaxy (red, short curve) by Coleman et al. (1980); Benítez et al. (2004). (b) showing a SED plot for the 6 identified objects that have spectroscopic redshifts from SDSS.*

The SED models (Chary & Elbaz, 2001) used in Figure 8a include a standard example of an ULIRG, a model of a merging galaxy indicated as merger in the plot, a Seyfert 1 galaxy, an ULIRG with a QSO and a peculiar IR galaxy.
All these models show a significant resemblance to the data we provide, although none of these models fit the measurements exactly. This can be due to the lack of the possibility to correct for extinction in the SDSS filters in the visible regime. However, as IR data is largely independent of extinction, the longer part of wavelengths should show a tighter fit to the model. A k-correction was applied to the data. It turned out however, that this correction accounts for 0.01 mag, which is well within the error of our measurements, and thus, neglected in further analysis.

Additionally the model of an elliptical galaxy (Coleman et al., 1980; Benítez et al., 2004) for a partial area of the IR wavelength range has been plotted to show that all our measured galaxies exhibit features of an elliptic galaxy as well, if no information about the longer wavelengths is present.
We normalize the model data to a suitable flux at a chosen wavelength of an object of our sample, in order to directly compare our results with the theoretical curves. This particular wavelength has been chosen for three reasons. One being the fact that it lies at the median of the number of measurement points, the second one derives from the little influence of extinction to IR data. We also find that this wavelength has the smallest spread between interpolation points of the model data, resulting in maximal reliability. We do so by following the established steps for calculating magnitudes from fluxes and vice verse. We take a look at the definition of the magnitude system

$$m_{obj} = -2.5 \cdot log\left(\frac{f_{\lambda,obj}}{f_{\lambda,Vega}}\right) \tag{27}$$

Where $f_{\lambda,obj}$ is the flux of an object at the wavelength λ, and $f_{\lambda,Vega}$ is the flux of a photometric reference point, in this case Vega, at the same wavelength.
We can see that a magnitude is defined by the ratio of two fluxes, and therefor independent of the system of units used, however we will adopt Jansky [Jy] as the standard unit system for our fluxes in our calculations.
There are more systems than just Vega for reference fluxes, but we will confine to this one, as this was also used for data reduction by the MESS team. A listing of values used is provided in Table 1.
By reorganizing equation (27) we get

$$f_{\lambda,obj} = f_{\lambda,Vega} \cdot 10^{-\frac{m_{obj}}{2.5}} \tag{28}$$

With this set of equations we can now convert all magnitudes calculated by SExtractor to fluxes and homogenize the rest of our additional measurements collected from online databases.
In order to use the values calculated from models, which are given in units of F_λ, we have to convert those with the following relation:

$$f_{Jy} = 3.33564095\,x\,10^4 \cdot F_\lambda[erg/cm^2/s/A] \cdot \lambda^2[A^2] \tag{29}$$

We can see the results of the application of these few easy steps in Figure 8a.
It is obvious that our suspicion has been confirmed. The rough shape of the model SED's follows the same pattern as our measurements. There is a clearly visible deviation from the model for most of the data, suggesting further research on that matter. In this thesis however, we will acknowledge this circumstance and move on with our last step of data analysis.

Table 1: The Vega-System reference points for flux calibration

Filter	$\lambda[\mu m]$	$f_{\lambda,Vega}$ [Jy]
u	0.354	3676
g	0.475	3640
r	0.622	3645
i	0.763	3641
z	0.905	3631
j	1.25	1577
h	1.65	1050
k	2.17	674.90
w1	3.35	309.54
w2	4.6	171.78
w3	11.6	31.67
w4	22.1	8.36
Herschel B	70	0.78
Herschel G	100	0.38
Herschel R	160	0.14

6.2.5 Fifth Step - SFR Estimation

We finally arrive at our last step of manual analysis with enough knowledge about the objects we have found to make profound predictions based on proven assumptions. The main goal of this thesis was to calculate star formation rates from IR data of photometric Herschel measurements, if the objects we detect actually turn out to be IR and ULIRG galaxies. Therefore we follow Calzetti et al. (2010), who showed that there is a relation between IR measurements of galaxies that show significant luminosity elevations, and their star formation rates. By collecting data across the whole energy spectrum Calzetti et al. was able determine the SFR for a large sample of well known objects in differ-

ent ways and therefore provide very accurate estimates of said value for this class of objects. She then correlated these estimates to luminosities and surface brightnesses of these objects measured in filter bands ranging from 24µm (Spitzer, (Mainzer et al., 2003)) to 160µm (Herschel, (Pilbratt et al., 2010)) and showed that a relation between these two quantities exist within a set of given boundary values.
The luminosity at a given frequency is defined by

$$L_\nu = 4\pi R^2 f_\nu \tag{30}$$

From (Calzetti et al., 2010) we adopt two equations:

$$SFR(70\mu m)\,[M_\odot\, yr^{-1}] = \frac{L(70\mu m)\,[erg\, s^{-1}]}{1.43\, x\, 10^{43}} \tag{31}$$

for $L(70\mu m) \geq 1.4\, x\, 10^{42}\, [erg\, s^{-1}]$. This estimate does inherent a dispersion of at least ~1 order of magnitude.
A second relation for measurements at 160µm is provided:

$$SFR(160\mu m)\,[M_\odot\, yr^{-1}] = \frac{L(160\mu m)\,[erg\, s^{-1}]}{7.0\, x\, 10^{42}} \tag{32}$$

for $L(160\mu m) > 1.0\, x\, 10^{42}\, [erg\, s^{-1}]$. The derivation of SFR from L(160µm) has almost twice the dispersion of that from L(70µm), and therefore is not as a good estimator for this value as L(70µm). This arises from the influence of heating of dust and by non-star-forming populations when using radiation at 160µm exclusively.
In the end, we provide calculated data for 22 identified objects in both wavelengths with estimates of their respective error.

6.2.6 Sixth Step - Total Infrared Luminosity Estimation

The total infrared luminosity (L_{TIR}) is an estimate of how much radiation a galaxy emits throughout the whole infrared spectrum of light. According to Galametz et al. (2013) we define it by

$$L_{TIR} = \int_{3\mu m}^{1100\mu m} L_\nu \, \mathrm{d}\nu \tag{33}$$

We can derive $L_{TIR}\,[ergs^{-1}]$ from our own measurements at 70µm (B) and 160µm (R), by using the following relations found by Galametz et al. (2013).

$$\log L_{TIR} = a_i \log\left(\nu\, L_\nu(i)\right) + b_i \tag{34}$$

and the following table of coefficients

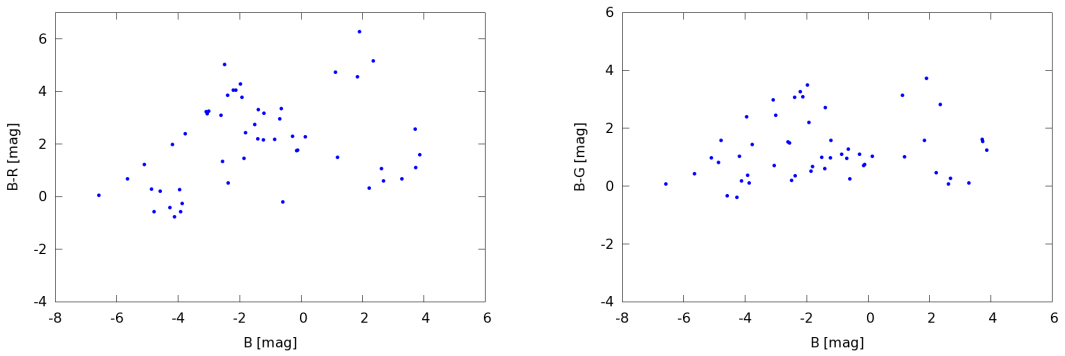
Table 2: Shortened table of coefficients for calculation of L_{TIR}. Taken from Galametz et al. (2013).

Wavelength	a_i	b_i
70 μm	0.973 ±(0.002)	0.567 ±(0.013)
160 μm	1.024 ±(0.003)	0.176 ±(0.018)

This approach is consistent up to z~2 according to Galametz et al. (2013), who derived this relation from near by resolved and known galaxies. We would have been able to provide further analysis if our sample had contained G filter measurements for all objects. With the current set we only would have been able to show that all detected objects in this thesis are indeed submilimeter galaxies (SMG).
The results are presented in table 4 in section 7.

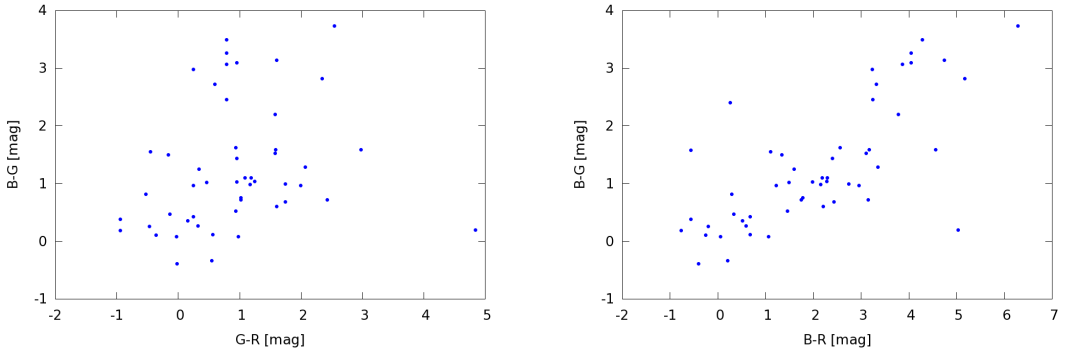
7 Results

At first we will have a look at a set of color magnitude diagrams created from the complete sample of detected objects with respective Herschel B(70µm), G(100µm), and/or R(160µm) measurements.
While the CMD (Color Magnitude Diagram) for B-R colors against B magnitude (see Figure 9a) does not deliver decisive information for sample selection, the next CMD for B-G colors against B (see Figure 9b) at least tells us that we are most likely dealing with the same class of object at different distances.
If we make our sample smaller and only take objects with all three filter bands into account, we can plot color-color diagrams for further inspection (see Figure 10a & 10b).



(a) (b)

Figure 9:
(a) showing a CMD diagram for all 173 objects with B and R measurements against apparent B magnitude. Unfortunately inconclusive for sample selection. (b) showing a CMD diagram for all 151 objects with B and G measurements against apparent B magnitude. Clearly visible correlation between B-G color and apparent B magnitude.



(a) (b)

Figure 10:
(a) showing a color color diagram for B-G and G-R, no correlation visible. (b) showing a color color diagram for B-G-B-R, a clear relation between these two colors is visible.

We can also plot all data for our sample of 6 identified objects with known distances. By trying to keep the following Figure 11 as simple and clear as possible, we have to make a compromise by not plotting the distance corrected data for all 26 objects. Instead, we will put the data for the 6 objects that had SDSS redshift information with the distance corrected fluxes into the plot, and leave the rest of the data un-plotted. The shape of the SED of the averaged objects clearly fits the model data at a first approximation, so there is no point in overloading the figure even more.

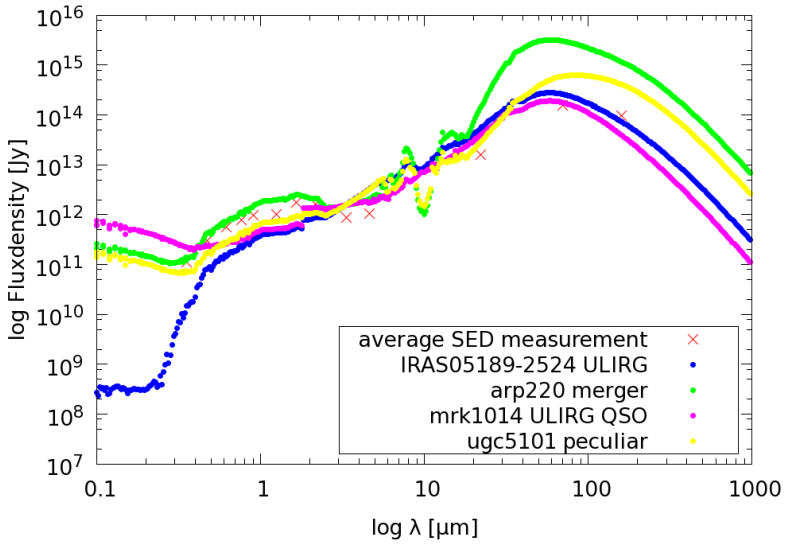


Figure 11: *Plots for 6 objects with distance correction and normalized model data to 3.35µm averaged to a single measurement for each filter.*

With our calculation for the derivation of SFRs for the selected sample we can plot the following Figure 12.

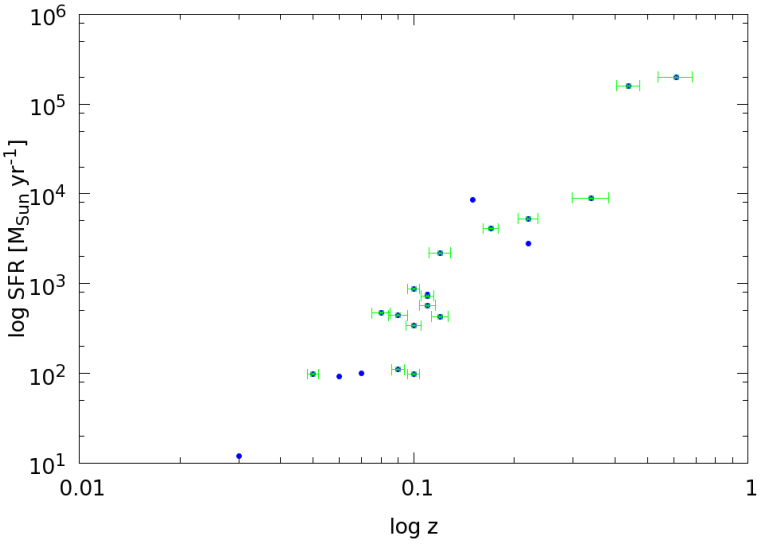


Figure 12: *SFR against redshift for the 26 identified objects in our sample with derived distances. Errorbars for z for all but the ones with SDSS redshifts.*

In Figure 12 we find a strong correlation between SFR and z. We expect galaxies at greater distances to start interacting with each other, causing phases of starbursts. This effect does get larger as we look farther away and, fortunately, this is also what we observe in this thesis.
We can now plot another CMD diagram for our sample with known & calculated distances and find a tighter correlation between these two measurements in Figure 13.

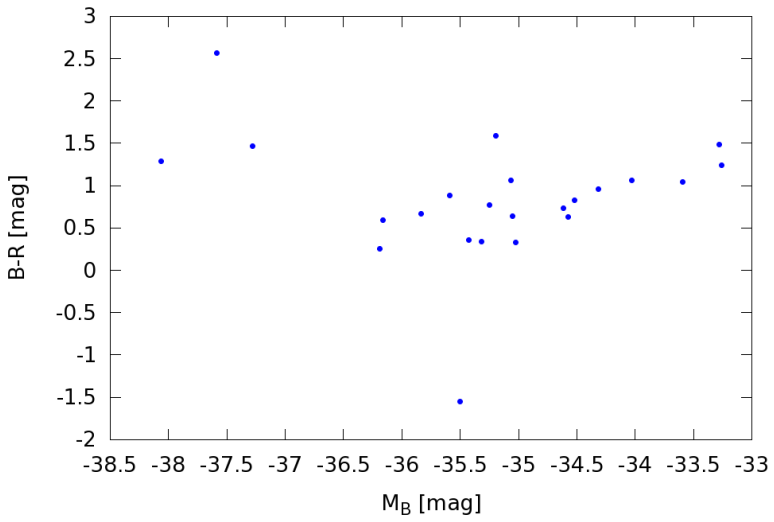


Figure 13: *CMD for color B-R against absolute B magnitude.*

A nice correlation is clearly visible in Figure 13 for all objects with distances $d \lesssim 700Mpc$ corresponding to $z \lesssim 0.13$.
Finally, we present the results (see table 3) with estimates for distance, redshift, L_ν and SFR. Additionally we provide estimates for TIR (Total Infrared Luminosity) for the same set of well known objects with the respective errors (see table 4). The overall object type has $L_{TIR} \simeq 10^{11}\, L_\odot$ allowing of classification of the objects to the class of LIRGs (Luminous Infrared Galaxy) with ULIRGs starting at $L_{TIR} \simeq 10^{12}\, L_\odot$. We find three objects to be in the range of ULIRGs with $L_{TIR} \geq 10^{12}\, L_\odot$, however we must keep in mind that these objects also are the ones with the greatest distances in the sample, and thus these results may not be as reliable as the rest of the of the sample.
This concludes this section, and leaves us with a lot of untouched topics. We remind the reader that this thesis is mainly meant to be about the methodology to accomplish analysis of large amounts of data, and simply shows the possibility of practical adaption to this kind of task.

Table 3: Results Table: Distance (d), Redshift (z), Luminosity in $[erg\,s^{-1}]$ at $70\mu m$ ($L_\nu(70)$), Luminosity in $[erg\,s^{-1}]$ at $160\mu m$ ($L_\nu(160)$), SFR in $[M_\odot\,yr^{-1}]$ derived at $70\mu m$ ($SFR(70)$), and SFR in $[M_\odot\,yr^{-1}]$ derived at $160\mu m$ ($SFR(160)$). Individual errors are indicated next to each column of measurment to right. The first 6 objects have redshift estimates from SDSS and therefore, no error estimate is available.

Identifier	RA(J2000)	DEC(J2000)	d [Mpc]	err	z	err	$L_\nu(70)$	err	$L_\nu(160)$	err	SFR(70)	err	SFR(160)	err
2MASX J09474126+1331248	146,9219827	13,523776	320	-	0,11	-	1,3E+46	2,2E+44	3,1E+45	1,0E+44	7,5E+02	1,3E+01	4,5E+02	1,4E+01
2MASX J09482083+1313233	147,0871447	13,2232555	213	-	0,07	-	1,7E+45	5,8E+43	8,3E+44	3,0E+43	1,0E+02	3,4E+00	1,2E+02	4,3E+00
SDSS J094808.32+130248.9	147,0344309	13,0473411	648	-	0,22	-	4,7E+46	3,4E+45	3,6E+46	1,9E+45	2,8E+03	2,0E+02	5,2E+03	2,7E+02
IC 1296	283,3301893	33,0660501	436	-	0,15	-	1,4E+47	2,1E+45	1,0E+47	1,1E+45	8,5E+03	1,2E+02	1,4E+04	1,6E+02
KUG 0217-033	34,9250722	-3,1193505	102	-	0,03	-	2,0E+44	5,7E+42	1,4E+44	2,5E+42	1,2E+01	3,4E-01	2,0E+01	3,5E-01
LCRS B131126.5-022123	198,5049134	-2,6215837	177	-	0,06	-	1,6E+45	3,3E+43	6,8E+44	1,8E+43	9,2E+01	1,9E+00	9,6E+01	2,5E+00
2XMM J020705.1+645642	31,7720999	64,9446652	1816	212	0,61	0,071	3,3E+48	2,2E+47	3,5E+49	2,0E+47	2,0E+05	1,3E+04	5,0E+06	2,0E+05
SDSSCGB 59955.02	147,1658069	13,4170536	667	45	0,22	0,015	8,9E+46	4,4E+45	1,7E+47	2,5E+45	5,3E+03	2,6E+02	2,4E+04	2,5E+03
2MASX J07223760-2550567	110,656553	-25,8495382	276	17	0,09	0,006	7,4E+45	3,2E+44	1,3E+46	1,4E+44	4,4E+02	1,9E+01	1,9E+03	1,4E+02
2MASX J09485546+1327144	147,2315527	13,4544467	374	28	0,12	0,009	3,8E+46	7,4E+44	6,6E+46	3,2E+44	2,2E+03	4,3E+01	9,4E+03	3,2E+02
2MASX J13483220-2833559	207,1337216	-28,5650403	350	21	0,12	0,007	7,4E+45	4,0E+44	1,6E+46	1,8E+44	4,3E+02	2,4E+01	2,3E+03	1,8E+02
2MASX J13492110-2833165	207,3375069	-28,5538624	319	19	0,11	0,006	9,7E+45	2,7E+44	1,3E+46	1,7E+44	5,7E+02	1,6E+01	1,9E+03	1,7E+02
GALEXASC J130300.09+050938.5	195,7503606	5,1602335	302	15	0,10	0,005	5,7E+45	3,0E+44	1,0E+46	1,4E+44	3,4E+02	1,8E+01	1,5E+03	1,4E+02
GALEXASC J134810.31-282957.5	207,0421381	-28,4987933	275	13	0,09	0,004	1,9E+45	1,4E+44	5,1E+45	7,1E+43	1,1E+02	8,1E+00	7,2E+02	7,1E+01
GALEXASC J134840.45-282056.5	207,1688204	-28,3487131	241	14	0,08	0,005	8,0E+45	9,1E+43	1,1E+46	5,9E+43	4,7E+02	5,3E+00	1,6E+03	5,9E+01
GALEXASC J191550.60-070036.8	288,9608368	-7,0109392	303	13	0,10	0,004	1,5E+46	3,4E+44	3,3E+46	1,9E+44	8,7E+02	2,0E+01	4,8E+03	1,9E+02
KUG 1422+260	216,0966503	25,7881984	159	7	0,05	0,002	1,7E+45	2,8E+43	3,3E+45	9,6E+42	9,8E+01	1,6E+00	4,7E+02	9,6E+00
LCRS B131127.6-022117	198,5093627	-2,6201783	1321	106	0,44	0,035	2,7E+48	9,0E+46	8,9E+48	4,9E+46	1,6E+05	5,3E+03	1,3E+06	4,9E+04
MCG +09-19-014	168,7426541	54,9608307	325	15	0,11	0,005	1,2E+46	3,9E+44	2,5E+46	1,6E+44	7,3E+02	2,3E+01	3,6E+03	1,6E+02
SBS 1111+553	168,6666879	55,102439	500	28	0,17	0,009	7,0E+46	1,8E+45	8,8E+46	8,9E+44	4,1E+03	1,0E+02	1,3E+04	8,9E+02
SDSS J094624.31+132042.9	146,6016421	13,3455446	1016	128	0,34	0,043	1,5E+47	1,7E+46	3,7E+46	4,6E+45	9,0E+03	9,7E+02	5,3E+03	4,6E+03
SDSS J124533.15+451903.6	191,3881286	45,3181193	298	12	0,10	0,004	1,7E+45	1,7E+44	5,2E+45	7,8E+43	9,8E+01	9,7E+00	7,4E+02	7,8E+01

Table 4: Total Infrared Luminosity (L_{TIR}) [$L_{\odot}$] calculated from $L_{\nu}(\lambda)$ according to Galametz et al. (2013) with error estimates.

Identifier	RA(J2000)	DEC(J2000)	$L_{\nu}(70)$	err	$L_{\nu}(160)$	err	log $L_{TIR}(70)$	err	log $L_{TIR}(160)$	err
2MASX J09474126+1331248	146,9219827	13,5237760	1,2822E+46	2,2476E+44	3,1333E+45	9,9674E+43	10,96	0,04	10,12	0,06
2MASX J09482083+1313233	147,0871447	13,2232555	1,7446E+45	5,7985E+43	8,3147E+44	3,0001E+43	10,11	0,05	9,53	0,06
SDSS J094808.32+130248.9	147,0344309	13,0473411	4,6938E+46	3,3863E+45	3,6485E+46	1,8628E+45	11,50	0,06	11,21	0,07
IC 1296	283,3301893	33,0660501	1,4468E+47	2,1164E+45	9,9878E+46	1,0887E+45	11,98	0,04	11,66	0,06
KUG 0217-033	34,9250722	-3,1193505	1,9845E+44	5,728E+42	1,3983E+44	2,4636E+42	9,20	0,04	8,73	0,05
LCRS B131126.5-022123	198,5049134	-2,6215837	1,5589E+45	3,2677E+43	6,755E+44	1,7561E+43	10,07	0,04	9,43	0,06
2XMM J020705.1+645642	31,7720999	64,9446652	3,3342E+48	2,1623E+47	3,5288E+49	1,964E+47	13,31	0,07	14,26	0,06
SDSSCGB 59955.02	147,1658069	13,4170536	8,9399E+46	4,4394E+45	1,6589E+47	2,4518E+45	11,78	0,06	11,88	0,06
2MASX J07223760-2550567	110,6565530	-25,8495382	7,4492E+45	3,2018E+44	1,3382E+46	1,3665E+44	10,73	0,05	10,76	0,05
2MASX J09485546+1327144	147,2315527	13,4544467	3,8175E+46	7,3816E+44	6,5835E+46	3,1991E+44	11,42	0,04	11,47	0,05
2MASX J13483220-2833559	207,1337216	-28,5650403	7,3743E+45	4,0096E+44	1,584E+46	1,7629E+44	10,72	0,06	10,84	0,05
2MASX J13492110-2833165	207,3375069	-28,5538624	9,7004E+45	2,7479E+44	1,3117E+46	1,6885E+44	10,84	0,05	10,75	0,05
GALEXASC J130300.09+050938.5	195,7503606	5,1602335	5,7495E+45	3,046E+44	1,0269E+46	1,4241E+44	10,62	0,06	10,64	0,05
GALEXASC J134810.31-282957.5	207,0421381	-28,4987933	1,937E+45	1,3709E+44	5,069E+45	7,0524E+43	10,16	0,06	10,33	0,05
GALEXASC J134840.45-282056.5	207,1688204	-28,3487131	8,0407E+45	9,0577E+43	1,1213E+46	5,9353E+43	10,76	0,04	10,68	0,05
GALEXASC J191550.60-070036.8	288,9608368	-7,0109392	1,477E+46	3,402E+44	3,3302E+46	1,9334E+44	11,02	0,04	11,17	0,05
KUG 1422+260	216,0966503	25,7881984	1,666E+45	2,7693E+43	3,2606E+45	9,5695E+42	10,09	0,04	10,13	0,05
LCRS B131127.6-022117	198,5093627	-2,6201783	2,7267E+48	9,0142E+46	8,9331E+48	4,9289E+46	13,22	0,05	13,65	0,06
MCG +09-19-014	168,7426541	54,9608307	1,2367E+46	3,9231E+44	2,5181E+46	1,6147E+44	10,94	0,05	11,04	0,05
SBS 1111+553	168,6666879	55,1024390	6,9928E+46	1,7678E+45	8,8237E+46	8,8731E+44	11,67	0,05	11,60	0,06
SDSS J094624.31+132042.9	146,6016421	13,3455446	1,5296E+47	1,6534E+46	3,6771E+46	4,6125E+45	12,00	0,08	11,21	0,10
SDSS J124533.15+451903.6	191,3881286	45,3181193	1,662E+45	1,6538E+44	5,2128E+45	7,8307E+43	10,09	0,07	10,34	0,05

8 Conclusions

In this thesis, we discussed some basic concepts of infrared radiation and its applications. A quick, abridged, yet comprehensive overview of the physics of star formation and dynamical behavior within a galaxy has been presented. We took a brief look into the Herschel Space Observatory and its main instrument that has been used for our research. The origin of the data provided by the MESS group has been outlined. We then give insight into our approach of analyzing this data with an epitomized look at the tools used. In the main part, we constituted a semi-automatic pipeline written in Python for handling medium sized amounts of photometric Herschel data. The basic steps were depicted and then formulated in detail. Finally, the manual part of analysis has been demonstrated and results were presented. We show that we observed properties of sources that confirmed our assumption about the class of object we presumed to be dealing with. An estimate for SFRs (Table 3) and TIR (Table 4) for these LIRGS has been established and we present calculated distances and redshifts for this sample of objects. Additionally, tables with B, G and R measurements and calculated colors for the whole sample (where possible) are included in the Appendix (see Table 5, 6 & 7)
This thesis has been based on the notion that there is a lot more science to do with data, apart from the initial purpose that it has been recorded for. We therefore emphasize the need creativity and creation of automated routines that can handle large amounts of data and extract new information, adding to new results for the astronomical community and a sophisticated scientific insight into the processes within our universe.

9 Outlook

In the future, we would like to extend the results for SFR estimates to the part of our sample, that has no yet been identified in other catalogs. In order to accomplish that, we can take another look at the color color plots from section 7. With these plots, we are theoretically able to distinguish between object classes. All we need is to determine where our desired objects are supposed to show up in the color-color plots. This has not been done for this thesis, as time was a factor. The basic approach, however, would be to look up already calculated model data for colors of this object type in the literature and correlate those results to ours. Another, more consistent approach, would be to show a dependency for the desired colors of LIRGs & ULIRGs within our Herschel filter data, by fitting a curve to a B-G against B-R plot for objects that qualify for these constrains, and then applying that result to the rest of our sample, where no further analysis has been carried out yet.
This will contribute additional data and insight into future research on star formation in IR galaxies, LIRGs and ULIRGS.

A Appendix

A.1 The Code

fps–ap_1.0.py

```
#!/bin/sh
# -*- coding: utf-8 -*-
# FPS-AP Python Script by Michael Mach
# ***************************************************
# FITS PREPARATOR for SEXTRACTOR and APERTURE PHOTOMETRY
# ***************************************************
#
# this script will take fits files from subfolders like these:
#
# |-mess-scanam-rep-21 (rootdir)
#  |---3C 58 SN 1181-B-CLEAN
#  |---3C 58 SN 1181-G-CLEAN
#  |---3C 58 SN 1181-R-CLEAN
#  |---AFGL 190-B-CLEAN
#  |---AFGL 190-R-CLEAN
#
# and save them in a new dir one level above rootdir like this:
#
# |-mess-scanam-rep-21-paired
#  |---3C 58 SN 1181
#  |---AFGL 190
#
# it then renames the files to sextractor readable files and
# executes sextractor on them
#
# fps-ap_1.*.py needs to be executed from rootdir.
# sextractor default*. (and other input-)files need to be in rootdir.
#
# a central/border mask is applied, and *_masked.cat files are run
# through ASSOC mode of sextractor.
#
# resulting sources are run through NED (NASA EXTRAGALACTIC DATABASE)
# and crosscorrelated.
#
# SFR and surface brightness values are still in an experimental
# phase, and do not resemble any real result as of yet.

import sys, os, shutil, glob, pyfits, math, urllib2, csv

# ------------------------------------------------------------------
# -----create parent subfloders for frames and put all corresponding---
# -----frames in them-----------------------------------------------
# ------------------------------------------------------------------

root = os.getcwd()

for path, subdirs, files in os.walk(root):
    for name in subdirs:
        subdir = os.path.join(path, name)
```

```
50          os.chdir(subdir)
51          for fits in glob.glob(subdir + '/*-[B,G,R]*.fits'):
52              foldername = os.path.basename(os.path.dirname(fits))[:-8]
53              destination = os.path.join(root, foldername)
54              filename = os.path.basename(fits)
55              old_foldername = os.path.dirname(fits)
56              new_destination = os.path.dirname(root) + \
57              '/mess-scanam-rep-21-paired/'
58
59  # ----------------------------------------------------------------------
60  # -----create new folderstructure in parent path of root---------------
61  # ----------------------------------------------------------------------
62
63              if not os.path.exists(os.path.join(new_destination, \
64                 foldername)):
65                 os.makedirs(os.path.join(new_destination, foldername))
66
67  # ----------------------------------------------------------------------
68  # -----copy fits files to new target dirs-----------------------------
69  #-----(switch off if not necessary anymore)--------------------------
70  # ----------------------------------------------------------------------
71
72              shutil.copy(filename, os.path.join(new_destination, \
73              foldername))
74              print 'moving ' + filename + ' ... done'
75
76  # ----------------------------------------------------------------------
77  # -----copy sextractor configuration files to target dirs--------------
78  # ----------------------------------------------------------------------
79
80              for sexconfig in glob.glob(root + '/default*.*'):
81                  shutil.copy(sexconfig, os.path.join(new_destination, \
82                  foldername))
83              for convolution in glob.glob(root + '/*.conv'):
84                  if not convolution == 'default.conv':
85                         shutil.copy(convolution, os.path.join \
86                         (new_destination, foldername))
87              for assoc in glob.glob(root + '/assoc.*'):
88                  if not assoc == 'default.param':
89                         shutil.copy(assoc, os.path.join \
90                         (new_destination, foldername))
91
92  print 'moving config files ... done'
93
94  # ----------------------------------------------------------------------
95  # -----rename files to 'sextractor readable' names--------------------
96  # -----B.fits, G.fits and R.fits---------------------------------------
97  # ----------------------------------------------------------------------
98
99  os.chdir(new_destination)
100
101 for sex_path, sex_subdirs, sex_files in os.walk(new_destination):
102     for dirs in sex_subdirs:
103         dirsname = os.path.join(new_destination, dirs)
104         for sexfile in glob.glob(dirsname + '/*-[B,G,R]-*'):
105             if '-B-' in sexfile:
```

```
106                 os.rename(sexfile, os.path.dirname(sexfile) + '/B.fits'
    )
107             elif '-G-' in sexfile:
108                 os.rename(sexfile, os.path.dirname(sexfile) + '/G.fits
    ')
109             elif '-R-' in sexfile:
110                 os.rename(sexfile, os.path.dirname(sexfile) + '/R.fits
    ')
111
112 print 'renaming fits files for sextractor ... done'
113
114 # ------------------------------------------------------------------
115 # -----run sextractor (default install 2.19.5, use 'sex' without path--
116 #------for 2.8.6 from scisoft package)-----------------------------
117 # ------------------------------------------------------------------
118
119 os.chdir(new_destination)
120
121 for sex_path, sex_subdirs, sex_files in os.walk(new_destination):
122     for fits in sex_files:
123         os.chdir(sex_path)
124         if fits == 'B.fits':
125             os.system('/usr/local/bin/sex B.fits -CATALOG_NAME B.cat -
    CHECKIMAGE_TYPE APERTURES -CHECKIMAGE_NAME B_apercheck.fits')
126         elif fits == 'G.fits':
127             os.system('/usr/local/bin/sex G.fits -CATALOG_NAME G.cat -
    CHECKIMAGE_TYPE APERTURES -CHECKIMAGE_NAME G_apercheck.fits')
128         elif fits == 'R.fits':
129             os.system('/usr/local/bin/sex R.fits -CATALOG_NAME R.cat -
    CHECKIMAGE_TYPE APERTURES -CHECKIMAGE_NAME R_apercheck.fits')
130
131 # ------------------------------------------------------------------
132 # -----open catalog *.cat and filter out central source(s) with simple-
133 # -----geometrics by calculation a 10% radius (of frame dimensions)----
134 # -----from the center and deleting entries withing a certain distance-
135 # -----from the border of the mosaic image.---------------------------
136 # ------------------------------------------------------------------
137
138 os.chdir(new_destination)
139
140 for sex_path, sex_subdirs, sex_files in os.walk(new_destination):
141     for cat in sex_files:
142         os.chdir(sex_path)
143         if cat == 'B.cat':
144             i, j = 0, 0
145             fits = pyfits.open('B_apercheck.fits')
146             pixvalue = pyfits.getdata('B_apercheck.fits')
147             fitsdata = fits[0].data
148             x_center = fits[0].header['CRPIX1']
149             y_center = fits[0].header['CRPIX2']
150             x_dim = fits[0].header['NAXIS1']
151             y_dim = fits[0].header['NAXIS2']
152             r_mask = (x_dim+y_dim)/2*0.10
153             B_cat = open('B.cat','r')
154             B_cat_masked = open('B_masked.cat','wb')
155             for line in B_cat:
```

```
156                 if line.startswith('#'):
157                     B_cat_masked.write(line)
158                 else:
159                     i = i+1
160                     elements = line.split()
161                     x_object = float(elements[7])
162                     y_object = float(elements[8])
163                     r_object = (((x_object-x_center)**2+(y_object-
    y_center)**2))**0.5
164                     x_outer_thresh = (x_object)+(x_object-x_center)/
    r_object*90
165                     y_outer_thresh = (y_object)+(y_object-y_center)/
    r_object*90
166                     x_outer_perpen = (x_object)+(-y_object+y_center)/
    r_object*90
167                     y_outer_perpen = (y_object)+(x_object-x_center)/
    r_object*90
168                     x_outer_perpen_other = (x_object)+(y_object-y_center)
    /r_object*90
169                     y_outer_perpen_other = (y_object)+(-x_object+x_center
    )/r_object*90
170                     if x_outer_thresh > 0 and x_outer_thresh < x_dim and
    y_outer_thresh > 0 and y_outer_thresh < y_dim:
171                         r_outer_thresh = pixvalue[y_outer_thresh,
    x_outer_thresh]
172                         r_outer_thresh_perpen = pixvalue[y_outer_perpen,
    x_outer_perpen]
173                         r_outer_thresh_perpen_other = pixvalue[
    y_outer_perpen_other, x_outer_perpen_other]
174                     else:
175                         r_outer_thresh = r_outer_thresh_perpen =
    r_outer_thresh_perpen_other = -1e30
176                     if r_object > r_mask and str(r_outer_thresh) != '-1e
    +30' and str(r_outer_thresh_perpen) != '-1e30' and str(
    r_outer_thresh_perpen_other) != '-1e30':
177                         j = j + 1
178                         B_cat_masked.write(line)
179             B_cat_masked.close()
180             fits.close()
181             print 'sources found outside of mask in B frame of ' + os.
    path.basename(sex_path) + ': ' + str(j) + '/' + str(i)
182 
183         if cat == 'G.cat':
184             i, j = 0, 0
185             fits = pyfits.open('G_apercheck.fits')
186             pixvalue = pyfits.getdata('G_apercheck.fits')
187             fitsdata = fits[0].data
188             x_center = fits[0].header['CRPIX1']
189             y_center = fits[0].header['CRPIX2']
190             x_dim = fits[0].header['NAXIS1']
191             y_dim = fits[0].header['NAXIS2']
192             r_mask = (x_dim+y_dim)/2*0.10
193             G_cat = open('G.cat','r')
194             G_cat_masked = open('G_masked.cat','wb')
195             for line in G_cat:
196                 if line.startswith('#'):
```

```
                G_cat_masked.write(line)
            else:
                i = i+1
                elements = line.split()
                x_object = float(elements[7])
                y_object = float(elements[8])
                r_object = (((x_object-x_center)**2+(y_object-
y_center)**2))**0.5
                x_outer_thresh = (x_object)+(x_object-x_center)/
r_object*90
                y_outer_thresh = (y_object)+(y_object-y_center)/
r_object*90
                x_outer_perpen = (x_object)+(-y_object+y_center)/
r_object*90
                y_outer_perpen = (y_object)+(x_object-x_center)/
r_object*90
                x_outer_perpen_other = (x_object)+(y_object-y_center)
/r_object*90
                y_outer_perpen_other = (y_object)+(-x_object+x_center
)/r_object*90
                if x_outer_thresh > 0 and x_outer_thresh < x_dim and
y_outer_thresh > 0 and y_outer_thresh < y_dim:
                    r_outer_thresh = pixvalue[y_outer_thresh,
x_outer_thresh]
                    r_outer_thresh_perpen = pixvalue[y_outer_perpen,
x_outer_perpen]
                    r_outer_thresh_perpen_other = pixvalue[
y_outer_perpen_other, x_outer_perpen_other]
                else:
                    r_outer_thresh = r_outer_thresh_perpen =
r_outer_thresh_perpen_other = -1e30
                if r_object > r_mask and str(r_outer_thresh) != '-1e
+30' and str(r_outer_thresh_perpen) != '-1e30' and str(
r_outer_thresh_perpen_other) != '-1e30':
                    j = j + 1
                    G_cat_masked.write(line)
        G_cat_masked.close()
        fits.close()
        print 'sources found outside of mask in G frame of ' + os.
path.basename(sex_path) + ': ' + str(j) + '/' + str(i)

    if cat == 'R.cat':
        i, j = 0, 0
        fits = pyfits.open('R_apercheck.fits')
        pixvalue = pyfits.getdata('R_apercheck.fits')
        fitsdata = fits[0].data
        x_center = fits[0].header['CRPIX1']
        y_center = fits[0].header['CRPIX2']
        x_dim = fits[0].header['NAXIS1']
        y_dim = fits[0].header['NAXIS2']
        r_mask = (x_dim+y_dim)/2*0.10
        R_cat = open('R.cat','r')
        R_cat_masked = open('R_masked.cat','wb')
        for line in R_cat:
            if line.startswith('#'):
                R_cat_masked.write(line)
```

```
238                 else:
239                     i = i+1
240                     elements = line.split()
241                     x_object = float(elements[7])
242                     y_object = float(elements[8])
243                     r_object = (((x_object-x_center)**2+(y_object-
    y_center)**2))**0.5
244                     x_outer_thresh = (x_object)+(x_object-x_center)/
    r_object*90
245                     y_outer_thresh = (y_object)+(y_object-y_center)/
    r_object*90
246                     x_outer_perpen = (x_object)+(-y_object+y_center)/
    r_object*90
247                     y_outer_perpen = (y_object)+(x_object-x_center)/
    r_object*90
248                     x_outer_perpen_other = (x_object)+(y_object-y_center)
    /r_object*90
249                     y_outer_perpen_other = (y_object)+(-x_object+x_center
    )/r_object*90
250                     if x_outer_thresh > 0 and x_outer_thresh < x_dim and
    y_outer_thresh > 0 and y_outer_thresh < y_dim:
251                         r_outer_thresh = pixvalue[y_outer_thresh,
    x_outer_thresh]
252                         r_outer_thresh_perpen = pixvalue[y_outer_perpen,
    x_outer_perpen]
253                         r_outer_thresh_perpen_other = pixvalue[
    y_outer_perpen_other, x_outer_perpen_other]
254                     else:
255                         r_outer_thresh = r_outer_thresh_perpen =
    r_outer_thresh_perpen_other = -1e30
256                     if r_object > r_mask and str(r_outer_thresh) != '-1e
    +30' and str(r_outer_thresh_perpen) != '-1e30' and str(
    r_outer_thresh_perpen_other) != '-1e30':
257                         j = j + 1
258                         R_cat_masked.write(line)
259             R_cat_masked.close()
260             fits.close()
261             print 'sources found outside of mask in R frame of ' + os.
    path.basename(sex_path) + ': ' + str(j) + '/' + str(i)
262
263 print 'creating *_masked.cat files ... done'
264
265 # ----------------------------------------------------------------------
266 # -----run sextractor in ASSOC-catalog mode----------------------------
267 # ----------------------------------------------------------------------
268
269 os.chdir(new_destination)
270
271 for sex_path, sex_subdirs, sex_files in os.walk(new_destination):
272     for fits in sex_files:
273         os.chdir(sex_path)
274         if fits == 'G.fits':
275             os.system('/usr/local/bin/sex G.fits -CATALOG_NAME
    G_masked_assoc.CAT -PARAMETERS_NAME assoc.param -CHECKIMAGE_NAME
    G_masked_assoc.fits -ASSOC_NAME B_masked.cat -ASSOCSELEC_TYPE
    MATCHED')
```

```
276             os.system('/usr/local/bin/sex B.fits -CATALOG_NAME B-
    G_masked_assoc.CAT -PARAMETERS_NAME assoc.param -CHECKIMAGE_NAME B-
    G_masked_assoc.fits -ASSOC_NAME G_masked.cat -ASSOCSELEC_TYPE ALL')
277         elif fits == 'R.fits':
278             os.system('/usr/local/bin/sex R.fits -CATALOG_NAME
    R_masked_assoc.CAT -PARAMETERS_NAME assoc.param -CHECKIMAGE_NAME
    R_masked_assoc.fits -ASSOC_NAME B_masked.cat -ASSOCSELEC_TYPE
    MATCHED')
279             os.system('/usr/local/bin/sex B.fits -CATALOG_NAME B-
    R_masked_assoc.CAT -PARAMETERS_NAME assoc.param -CHECKIMAGE_NAME B-
    R_masked_assoc.fits -ASSOC_NAME R_masked.cat -ASSOCSELEC_TYPE ALL')
280
281 print 'running SExtractor ASSOC mode ... done'
282
283 # ------------------------------------------------------------------
284 # -----run detected objects through NED (NASA EXTRAGALACTIC DATABASE):-
285 # -----create coordinate lists of found objects prepared for NED input-
286 # ----(max. size 500 objects)---------------------------------------
287 # ------------------------------------------------------------------
288
289 os.chdir(new_destination)
290
291 B_biglist1 = open('B_biglist1_temp.cat','wb')
292 B_biglist2 = open('B_biglist2_temp.cat','wb')
293 G_biglist1 = open('G_biglist1_temp.cat','wb')
294 G_biglist2 = open('G_biglist2_temp.cat','wb')
295 R_biglist1 = open('R_biglist1_temp.cat','wb')
296 R_biglist2 = open('R_biglist2_temp.cat','wb')
297
298 for sex_path, sex_subdirs, sex_files in os.walk(new_destination):
299     for cat in sex_files:
300         os.chdir(sex_path)
301         if cat == 'B_masked.cat':
302             i = 0
303             B_masked = open('B_masked.cat','r')
304             for line in B_masked:
305                 if not line.startswith('#'):
306                     i += 1
307                     elements = line.split()
308                     kron_mag = float(elements[1])
309                     kron_err = float(elements[2])
310                     petro_mag = float(elements[3])
311                     petro_err = float(elements[4])
312                     kron_rad = float(elements[5])
313                     petro_rad = float(elements[6])
314                     RA_object = float(elements[9])
315                     DEC_object = float(elements[10])
316                     A_world = float(elements[13])
317                     A_world_err = float(elements[17])
318                     B_world = float(elements[14])
319                     B_world_err = float(elements[18])
320                     flags = float(elements[21])
321                     class_star = float(elements[22])
322                     filename = os.path.basename(sex_path)
323                     if i < 501:
```

```
                    B_biglist1.write(str(RA_object) + ',' + str(
DEC_object) + ',' + str(kron_mag) + ',' + str(kron_err) + ',' + str
(petro_mag) + ',' + str(petro_err) + ',' + str(kron_rad) + ',' +
str(petro_rad) + ',' + str(A_world) + ',' + str(A_world_err) + ','
+ str(B_world) + ',' + str(B_world_err) + ',' + str(flags) + ',' +
str(class_star) + ',' + filename + '\n')
                elif i > 500 and i < 1001:
                    B_biglist2.write(str(RA_object) + ',' + str(
DEC_object) + ',' + str(kron_mag) + ',' + str(kron_err) + ',' + str
(petro_mag) + ',' + str(petro_err) + ',' + str(kron_rad) + ',' +
str(petro_rad) + ',' + str(A_world) + ',' + str(A_world_err) + ','
+ str(B_world) + ',' + str(B_world_err) + ',' + str(flags) + ',' +
str(class_star) + ',' + filename + '\n')
        B_masked.close()


    if cat == 'G_masked.cat':
        i = 0
        G_masked = open('G_masked.cat','r')
        for line in G_masked:
            if not line.startswith('#'):
                i += 1
                elements = line.split()
                kron_mag = float(elements[1])
                kron_err = float(elements[2])
                petro_mag = float(elements[3])
                petro_err = float(elements[4])
                kron_rad = float(elements[5])
                petro_rad = float(elements[6])
                RA_object = float(elements[9])
                DEC_object = float(elements[10])
                A_world = float(elements[13])
                A_world_err = float(elements[17])
                B_world = float(elements[14])
                B_world_err = float(elements[18])
                flags = float(elements[21])
                class_star = float(elements[22])
                filename = os.path.basename(sex_path)
                if i < 501:
                    G_biglist1.write(str(RA_object) + ',' + str(
DEC_object) + ',' + str(kron_mag) + ',' + str(kron_err) + ',' + str
(petro_mag) + ',' + str(petro_err) + ',' + str(kron_rad) + ',' +
str(petro_rad) + ',' + str(A_world) + ',' + str(A_world_err) + ','
+ str(B_world) + ',' + str(B_world_err) + ',' + str(flags) + ',' +
str(class_star) + ',' + filename + '\n')
                elif i > 500 and i < 1001:
                    G_biglist2.write(str(RA_object) + ',' + str(
DEC_object) + ',' + str(kron_mag) + ',' + str(kron_err) + ',' + str
(petro_mag) + ',' + str(petro_err) + ',' + str(kron_rad) + ',' +
str(petro_rad) + ',' + str(A_world) + ',' + str(A_world_err) + ','
+ str(B_world) + ',' + str(B_world_err) + ',' + str(flags) + ',' +
str(class_star) + ',' + filename + '\n')
        G_masked.close()


    if cat == 'R_masked.cat':
```

```
            i = 0
            R_masked = open('R_masked.cat','r')
            for line in R_masked:
                if not line.startswith('#'):
                    i += 1
                    elements = line.split()
                    kron_mag = float(elements[1])
                    kron_err = float(elements[2])
                    petro_mag = float(elements[3])
                    petro_err = float(elements[4])
                    kron_rad = float(elements[5])
                    petro_rad = float(elements[6])
                    RA_object = float(elements[9])
                    DEC_object = float(elements[10])
                    A_world = float(elements[13])
                    A_world_err = float(elements[17])
                    B_world = float(elements[14])
                    B_world_err = float(elements[18])
                    flags = float(elements[21])
                    class_star = float(elements[22])
                    filename = os.path.basename(sex_path)
                    if i < 501:
                      R_biglist1.write(str(RA_object) + ',' + str(
    DEC_object) + ',' + str(kron_mag) + ',' + str(kron_err) + ',' + str
    (petro_mag) + ',' + str(petro_err) + ',' + str(kron_rad) + ',' +
    str(petro_rad) + ',' + str(A_world) + ',' + str(A_world_err) + ','
    + str(B_world) + ',' + str(B_world_err) + ',' + str(flags) + ',' +
    str(class_star) + ',' + filename + '\n')
                    elif i > 500 and i < 1001:
                      R_biglist2.write(str(RA_object) + ',' + str(
    DEC_object) + ',' + str(kron_mag) + ',' + str(kron_err) + ',' + str
    (petro_mag) + ',' + str(petro_err) + ',' + str(kron_rad) + ',' +
    str(petro_rad) + ',' + str(A_world) + ',' + str(A_world_err) + ','
    + str(B_world) + ',' + str(B_world_err) + ',' + str(flags) + ',' +
    str(class_star) + ',' + filename + '\n')
            R_masked.close()

B_biglist1.close()
B_biglist2.close()
G_biglist1.close()
G_biglist2.close()
R_biglist1.close()
R_biglist2.close()

os.chdir(new_destination)

#---split biglist files into 250 entries each ---

B_biglist1 = open('B_biglist1_temp.cat','r')
B_biglist1_out = open('B_biglist1.cat','wb')
B_biglist2_out = open('B_biglist2.cat','wb')
B_biglist3_out = open('B_biglist3.cat','wb')
B_biglist4_out = open('B_biglist4.cat','wb')
i = 0
for line in B_biglist1:
  i += 1
```

```
406   if i < 251:
407     B_biglist1_out.write(line)
408   elif i > 250 and i < 501:
409     B_biglist2_out.write(line)
410   elif i > 500 and i < 751:
411     B_biglist3_out.write(line)
412   elif i > 750 and i < 1001:
413     B_biglist4_out.write(line)
414
415 B_biglist1_out.close()
416 B_biglist2_out.close()
417 B_biglist3_out.close()
418 B_biglist4_out.close()
419
420 G_biglist1 = open('G_biglist1_temp.cat','r')
421 G_biglist1_out = open('G_biglist1.cat','wb')
422 G_biglist2_out = open('G_biglist2.cat','wb')
423 G_biglist3_out = open('G_biglist3.cat','wb')
424 G_biglist4_out = open('G_biglist4.cat','wb')
425 i = 0
426 for line in G_biglist1:
427   i += 1
428   if i < 251:
429     G_biglist1_out.write(line)
430   elif i > 250 and i < 501:
431     G_biglist2_out.write(line)
432   elif i > 500 and i < 751:
433     G_biglist3_out.write(line)
434   elif i > 750 and i < 1001:
435     G_biglist4_out.write(line)
436
437 G_biglist1_out.close()
438 G_biglist2_out.close()
439 G_biglist3_out.close()
440 G_biglist4_out.close()
441
442 R_biglist1 = open('R_biglist1_temp.cat','r')
443 R_biglist1_out = open('R_biglist1.cat','wb')
444 R_biglist2_out = open('R_biglist2.cat','wb')
445 R_biglist3_out = open('R_biglist3.cat','wb')
446 R_biglist4_out = open('R_biglist4.cat','wb')
447 i = 0
448 for line in R_biglist1:
449   i += 1
450   if i < 251:
451     R_biglist1_out.write(line)
452   elif i > 250 and i < 501:
453     R_biglist2_out.write(line)
454   elif i > 500 and i < 751:
455     R_biglist3_out.write(line)
456   elif i > 750 and i < 1001:
457     R_biglist4_out.write(line)
458
459 R_biglist1_out.close()
460 R_biglist2_out.close()
461 R_biglist3_out.close()
```

```
462 R_biglist4_out.close()
463
464 # ------------------------------------------------------------------------
465 # -----put lists into NED and read out crossmatches to ned.cat---------
466 # ------------------------------------------------------------------------
467
468 os.chdir(new_destination)
469
470 # -----run B frame throug ned-----------------------------------------
471
472 biglist1 = open('B_biglist1.cat','r')
473 biglist2 = open('B_biglist2.cat','r')
474 biglist3 = open('B_biglist3.cat','r')
475 biglist4 = open('B_biglist4.cat','r')
476 ned = open('B_ned.cat','wb')
477
478 uplist_obj=[]
479 for line in biglist1:
480     elements = line.split(',')
481     RA_object = str(elements[0])
482     DEC_object = str(elements[1])
483     uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
484 ned_input = ''.join(uplist_obj)
485 response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
486 for line in response:
487     if line[0].isdigit():
488         ned.write(line)
489
490 uplist_obj=[]
491 for line in biglist2:
492     elements = line.split(',')
493     RA_object = str(elements[0])
494     DEC_object = str(elements[1])
495     uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
496 ned_input = ''.join(uplist_obj)
497 response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
```

```
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

uplist_obj=[]
for line in biglist3:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

uplist_obj=[]
for line in biglist4:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

biglist1.close()
biglist2.close()
biglist3.close()
biglist4.close()
```

```
ned.close()

# -----run G frame through ned----------------------------------------

biglist1 = open('G_biglist1.cat','r')
biglist2 = open('G_biglist2.cat','r')
biglist3 = open('G_biglist3.cat','r')
biglist4 = open('G_biglist4.cat','r')
ned = open('G_ned.cat','wb')

uplist_obj=[]
for line in biglist1:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

uplist_obj=[]
for line in biglist2:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

```

```
564 uplist_obj=[]
565 for line in biglist3:
566     elements = line.split(',')
567     RA_object = str(elements[0])
568     DEC_object = str(elements[1])
569     uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
570 ned_input = ''.join(uplist_obj)
571 response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
572 for line in response:
573     if line[0].isdigit():
574         ned.write(line)
575
576 uplist_obj=[]
577 for line in biglist4:
578     elements = line.split(',')
579     RA_object = str(elements[0])
580     DEC_object = str(elements[1])
581     uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
582 ned_input = ''.join(uplist_obj)
583 response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
584 for line in response:
585     if line[0].isdigit():
586         ned.write(line)
587
588 biglist1.close()
589 biglist2.close()
590 biglist3.close()
591 biglist4.close()
592 ned.close()
593
594 # -----run R frame through ned----------------------------------------
595
596 biglist1 = open('R_biglist1.cat','r')
597 biglist2 = open('R_biglist2.cat','r')
```

```
biglist3 = open('R_biglist3.cat','r')
biglist4 = open('R_biglist4.cat','r')
ned = open('R_ned.cat','wb')

uplist_obj=[]
for line in biglist1:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

uplist_obj=[]
for line in biglist2:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

uplist_obj=[]
for line in biglist3:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
```

```
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

uplist_obj=[]
for line in biglist4:
    elements = line.split(',')
    RA_object = str(elements[0])
    DEC_object = str(elements[1])
    uplist_obj.append(RA_object + '+' + DEC_object + '%0D%0A')
ned_input = ''.join(uplist_obj)
response = urllib2.urlopen('http://ned.ipac.caltech.edu/cgi-bin/nnd?
    uplist=' + ned_input + '&sr_arcsec=9.0&delimiter=bar&NO_LINKS=1&
    nondb=row_count&nondb=user_inp_msg&nondb=user_inplist&nondb=
    user_inp_sep&crosid=objname&position=ra%2Cdec&position=pretype&
    position=z&position=zflag&EXTswitch=on&SchEXT=Landolt+U&SchEXT=
    Landolt+B&SchEXT=Landolt+V&SchEXT=Landolt+R&SchEXT=Landolt+I&
    gphotoms_CON=1493&gphotoms_CON=1494&gphotoms_CON=1495&gphotoms_CON
    =1&gphotoms_CON=2&gphotoms_CON=3&gphotoms_CON=4&gphotoms_CON=88&
    gphotoms_CON=89&gphotoms_CON=90&gphotoms_CON=91&diamdat_CON=117&
    diamdat_CON=2&diamdat_CON=32&diamdat_CON=15&attdat=attned&gphotoms=
    q_value&gphotoms=q_unc&gphotoms=ned_value&gphotoms=ned_unc&diamdat=
    ned_maj_dia&distance=avg&distance=stddev_samp')
for line in response:
    if line[0].isdigit():
        ned.write(line)

biglist1.close()
biglist2.close()
biglist3.close()
biglist4.close()
ned.close()

print 'NED B,G,R catalogues created in ../' + os.path.basename(os.path.
    dirname(new_destination)) + '/*_ned.cat'

# ----------------------------------------------------------------------
# -----correlate found objects from ned.cat to individual user---------
# -----catalogues------------------------------------------------------
# ----------------------------------------------------------------------

os.chdir(new_destination)

```

```
# ------------------------------------------------------------------
# -----concatenate biglist*.cat files into one bigfile (hence the name)
# ------------------------------------------------------------------

# -----B frame conc.-------------------------------------------------

biglist1 = open('B_biglist1.cat','r')
biglist2 = open('B_biglist2.cat','r')
biglist3 = open('B_biglist3.cat','r')
biglist4 = open('B_biglist4.cat','r')
biglist_out = open('B_biglist.cat','wb')

biglist_out.write('RA,DEC,Kron_mag,Kron_err,Petro_mag,Petro_err,
    kron_rad,petro_rad,A_world,A_world_err,B_world,B_world_err,flags,
    class_star,filename\n')

for line in biglist1:
    biglist_out.write(line)
for line in biglist2:
    biglist_out.write(line)
for line in biglist3:
    biglist_out.write(line)
for line in biglist4:
    biglist_out.write(line)

biglist_out.close()
biglist1.close()
biglist2.close()
biglist3.close()
biglist4.close()

# -----G frame conc.-------------------------------------------------

biglist1 = open('G_biglist1.cat','r')
biglist2 = open('G_biglist2.cat','r')
biglist3 = open('G_biglist3.cat','r')
biglist4 = open('G_biglist4.cat','r')
biglist_out = open('G_biglist.cat','wb')

biglist_out.write('RA,DEC,Kron_mag,Kron_err,Petro_mag,Petro_err,
    kron_rad,petro_rad,A_world,A_world_err,B_world,B_world_err,flags,
    class_star,filename\n')

for line in biglist1:
    biglist_out.write(line)
for line in biglist2:
    biglist_out.write(line)
for line in biglist3:
    biglist_out.write(line)
for line in biglist4:
    biglist_out.write(line)

biglist_out.close()
biglist1.close()
biglist2.close()
biglist3.close()
```

```
biglist4.close()

# -----R frame conc.-----------------------------------------------------

biglist1 = open('R_biglist1.cat','r')
biglist2 = open('R_biglist2.cat','r')
biglist3 = open('R_biglist3.cat','r')
biglist4 = open('R_biglist4.cat','r')
biglist_out = open('R_biglist.cat','wb')

biglist_out.write('RA,DEC,Kron_mag,Kron_err,Petro_mag,Petro_err,
    kron_rad,petro_rad,A_world,A_world_err,B_world,B_world_err,flags,
    class_star,filename\n')

for line in biglist1:
    biglist_out.write(line)
for line in biglist2:
    biglist_out.write(line)
for line in biglist3:
    biglist_out.write(line)
for line in biglist4:
    biglist_out.write(line)

biglist_out.close()
biglist1.close()
biglist2.close()
biglist3.close()
biglist4.close()

print "concatenating biglist files ... done"

# ----------------------------------------------------------------------
# -----take result from NED query and format needed information nicely-
# ----------------------------------------------------------------------

# -----B frame format----------------------------------------------------

#count all lines in ned.cat
linenumber = sum(1 for line in open('B_ned.cat','r'))
ned = open('B_ned.cat','r')
master_list = open('B_masterlist.cat','wb')

i=0
for line in ned:
    elements = line.split('|')
    i += 1
    if i > 0 and i < linenumber - 1 and elements[2][0].isdigit():
        coordset = elements[2].split()
        RA_ned = coordset[0]
        DEC_ned = coordset[1]
        notes = elements[1].strip()
        obj_name = elements[4].strip()
        new_list = str(elements[0].strip()) + ',' + str(RA_ned) + ',' +
    str(DEC_ned) + ',' + str(obj_name) + '\n'
        master_list.write(new_list)
master_list.close()
```

```
770 ned.close()
771
772 # -----G frame format-------------------------------------------------
773
774 #count all lines in ned.cat
775 linenumber = sum(1 for line in open('G_ned.cat','r'))
776 ned = open('G_ned.cat','r')
777 master_list = open('G_masterlist.cat','wb')
778
779 i=0
780 for line in ned:
781     elements = line.split('|')
782     i += 1
783     if i > 4 and i < linenumber - 1 and elements[2][0].isdigit():
784         coordset = elements[2].split()
785         RA_ned = coordset[0]
786         DEC_ned = coordset[1]
787         notes = elements[1].strip()
788         obj_name = elements[4].strip()
789         new_list = str(elements[0].strip()) + ',' + str(RA_ned) + ',' +
    str(DEC_ned) + ',' + str(obj_name) + '\n'
790         master_list.write(new_list)
791 master_list.close()
792 ned.close()
793
794 # -----R frame format-------------------------------------------------
795
796 #count all lines in ned.cat
797 linenumber = sum(1 for line in open('R_ned.cat','r'))
798 ned = open('R_ned.cat','r')
799 master_list = open('R_masterlist.cat','wb')
800
801 i=0
802 for line in ned:
803     elements = line.split('|')
804     i += 1
805     if i > 4 and i < linenumber - 1 and elements[2][0].isdigit():
806         coordset = elements[2].split()
807         RA_ned = coordset[0]
808         DEC_ned = coordset[1]
809         notes = elements[1].strip()
810         obj_name = elements[4].strip()
811         new_list = str(elements[0].strip()) + ',' + str(RA_ned) + ',' +
    str(DEC_ned) + ',' + str(obj_name) + '\n'
812         master_list.write(new_list)
813 master_list.close()
814 ned.close()
815
816 print "formatting ned files ... done"
817
818 # -------------------------------------------------------------------
819 # -----use csv.reader to merge all information from both files---------
820 # ----(NED query and bigfile)-----------------------------------------
821 # -------------------------------------------------------------------
822
823 # -----B merge ned and bigfile---------------------------------------
```

```
824
825 master = file('B_masterlist.cat','r')
826 bigfile = file('B_biglist.cat','r')
827 out = file('B_master_out.cat','w')
828
829 master_reader = csv.reader(master)
830 bigfile_reader = csv.reader(bigfile)
831 master_out = csv.writer(out)
832
833 bigfile_list = list(bigfile_reader)
834
835 master_out.writerow(["#","RA","DEC","NED","Kron_mag","Kron_err","
    Petro_mag","Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    "])
836
837 for master_row in master_reader:
838     for big_row in bigfile_list:
839         results_row = master_row
840         RA_master = master_row[1]
841         DEC_master = master_row[2]
842         RA_big = big_row[0]
843         DEC_big = big_row[1]
844         kron_mag = big_row[2]
845         kron_err = big_row[3]
846         petro_mag = big_row[4]
847         petro_err = big_row[5]
848         kron_rad = big_row[6]
849         petro_rad = big_row[7]
850         A_world = big_row[8]
851         A_world_err = big_row[9]
852         B_world = big_row[10]
853         B_world_err = big_row[11]
854         flags = big_row[12]
855         class_star = big_row[13]
856         file_name = big_row[14]
857         if RA_master == RA_big and DEC_master == DEC_big:
858             results_row.extend([kron_mag,kron_err,petro_mag,petro_err,
    kron_rad,petro_rad,A_world,A_world_err,B_world,B_world_err,flags,
    class_star,file_name])
859             break
860     master_out.writerow(results_row)
861
862 master.close()
863 bigfile.close()
864 out.close()
865
866 # -----G merge ned and bigfile-------------------------------------
867
868 master = file('G_masterlist.cat','r')
869 bigfile = file('G_biglist.cat','r')
870 out = file('G_master_out.cat','w')
871
872 master_reader = csv.reader(master)
873 bigfile_reader = csv.reader(bigfile)
874 master_out = csv.writer(out)
```

```
875
876 bigfile_list = list(bigfile_reader)
877
878 master_out.writerow(["#","RA","DEC","NED","Kron_mag","Kron_err","
    Petro_mag","Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    "])
879
880 for master_row in master_reader:
881     for big_row in bigfile_list:
882         results_row = master_row
883         RA_master = master_row[1]
884         DEC_master = master_row[2]
885         RA_big = big_row[0]
886         DEC_big = big_row[1]
887         kron_mag = big_row[2]
888         kron_err = big_row[3]
889         petro_mag = big_row[4]
890         petro_err = big_row[5]
891         kron_rad = big_row[6]
892         petro_rad = big_row[7]
893         A_world = big_row[8]
894         A_world_err = big_row[9]
895         B_world = big_row[10]
896         B_world_err = big_row[11]
897         flags = big_row[12]
898         class_star = big_row[13]
899         file_name = big_row[14]
900         if RA_master == RA_big and DEC_master == DEC_big:
901             results_row.extend([kron_mag,kron_err,petro_mag,petro_err,
    kron_rad,petro_rad,A_world,A_world_err,B_world,B_world_err,flags,
    class_star,file_name])
902             break
903     master_out.writerow(results_row)
904
905 master.close()
906 bigfile.close()
907 out.close()
908
909 # -----R merge ned and bigfile----------------------------------------
910
911 master = file('R_masterlist.cat','r')
912 bigfile = file('R_biglist.cat','r')
913 out = file('R_master_out.cat','w')
914
915 master_reader = csv.reader(master)
916 bigfile_reader = csv.reader(bigfile)
917 master_out = csv.writer(out)
918
919 bigfile_list = list(bigfile_reader)
920
921 master_out.writerow(["#","RA","DEC","NED","Kron_mag","Kron_err","
    Petro_mag","Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    "])
922
```

```
for master_row in master_reader:
    for big_row in bigfile_list:
        results_row = master_row
        RA_master = master_row[1]
        DEC_master = master_row[2]
        RA_big = big_row[0]
        DEC_big = big_row[1]
        kron_mag = big_row[2]
        kron_err = big_row[3]
        petro_mag = big_row[4]
        petro_err = big_row[5]
        kron_rad = big_row[6]
        petro_rad = big_row[7]
        A_world = big_row[8]
        A_world_err = big_row[9]
        B_world = big_row[10]
        B_world_err = big_row[11]
        flags = big_row[12]
        class_star = big_row[13]
        file_name = big_row[14]
        if RA_master == RA_big and DEC_master == DEC_big:
            results_row.extend([kron_mag,kron_err,petro_mag,petro_err,
    kron_rad,petro_rad,A_world,A_world_err,B_world,B_world_err,flags,
    class_star,file_name])
            break
    master_out.writerow(results_row)

master.close()
bigfile.close()
out.close()

print "merging _ned.cat with biglist files ... done"

# ------------------------------------------------------------------
# -----remove obsolete temp files (activate individually for debugging)
# ------------------------------------------------------------------

#os.chdir(new_destination)

#os.remove('B_biglist.cat')
#os.remove('B_biglist1.cat')
#os.remove('B_biglist2.cat')
#os.remove('B_masterlist.cat')
#os.remove('B_ned.cat')

#os.remove('G_biglist.cat')
#os.remove('G_biglist1.cat')
#os.remove('G_biglist2.cat')
#os.remove('G_masterlist.cat')
#os.remove('G_ned.cat')

#os.remove('R_biglist.cat')
#os.remove('R_biglist1.cat')
#os.remove('R_biglist2.cat')
#os.remove('R_masterlist.cat')
#os.remove('R_ned.cat')
```

```
977
978 #print "deleting obsolete temp files ... done"
979
980 # ------------------------------------------------------------------
981 # -----calculate surface brightness and SFR (calzetti et al 2010)------
982 # ------------------------------------------------------------------
983
984 os.chdir(new_destination)
985
986 # B-file
987
988 B_master_in = file('B_master_out.cat','r')
989 out = file('B_table_SFR.cat','w')
990
991 master_reader = csv.reader(B_master_in)
992 master_out = csv.writer(out)
993
994 master_out.writerow(["#","RA","DEC","NED","Kron_mag","Kron_err","
    Petro_mag","Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    ","SFR_kron","SFR_kron_err","SFR_petro","SFR_petro_err"])
995
996 for master_row in master_reader:
997     results_row = master_row
998     if not master_row[0] is "#":
999         kron_mag = float(master_row[4])
1000        kron_err = float(master_row[5])
1001        petro_mag = float(master_row[6])
1002        petro_err = float(master_row[7])
1003        kron_rad = float(master_row[8])
1004        petro_rad = float(master_row[9])
1005        A_world = float(master_row[10])
1006        A_world_err = float(master_row[11])
1007        B_world = float(master_row[12])
1008        B_world_err = float(master_row[13])
1009        flags = float(master_row[14])
1010        class_star = float(master_row[15])
1011        if kron_mag < 99.0 and petro_mag < 99.0:
1012
1013 #--------kron-surface-brightness in [mag * arcsec**-2]-----------------
1014
1015            A_kron = (A_world * kron_rad) * 3600.0
1016            B_kron = (B_world * kron_rad) * 3600.0
1017            A_kron_err = (A_world_err * kron_rad) * 3600.0
1018            B_kron_err = (B_world_err * kron_rad) * 3600.0
1019
1020            area_kron = A_kron * B_kron * math.pi
1021            area_kron_err_max = (A_kron + A_kron_err) * (B_kron +
    B_kron_err) * math.pi
1022
1023            sb_kron = kron_mag + 2.5 * math.log10(area_kron)
1024            sb_kron_err_max = (kron_mag + kron_err) + 2.5 * math.log10(
    area_kron_err_max)
1025            sb_kron_err = sb_kron_err_max - sb_kron
1026
```

```
1027            sb_kron_handbook = kron_mag + 2.5 * math.log10(2827 + (A_kron
    *60) * (B_kron*60))
1028
1029 #--------petro-surface-brightness in [mag * arcsec**-2]----------------
1030
1031            A_petro = (A_world * petro_rad) * 3600.0
1032            B_petro = (B_world * petro_rad) * 3600.0
1033            A_petro_err = (A_world_err * petro_rad) * 3600.0
1034            B_petro_err = (B_world_err * petro_rad) * 3600.0
1035
1036            area_petro = A_petro * B_petro * math.pi
1037            area_petro_err_max = (A_petro + A_petro_err) * (B_petro +
    B_petro_err) * math.pi
1038
1039            sb_petro = petro_mag + 2.5 * math.log10(area_petro)
1040            sb_petro_err_max = (petro_mag + petro_err) + 2.5 * math.log10
    (area_petro_err_max)
1041            sb_petro_err = sb_petro_err_max - sb_petro
1042
1043            #print sb_kron, sb_kron_err, sb_petro, sb_petro_err
1044
1045 # -------convert surface-brightness to L_70 physical units ------------
1046 # ------[L_sun * kpc**-2]-------------------------------------------------
1047
1048            M_sun = 4.74 # need M_70 m_sun
1049
1050            # L in [L_sun * kpc**-2]
1051            L_kron = (10**(-0.4 * (sb_kron - M_sun - 21.572))) *
    (1000**2)
1052
1053            # L for max. error in [L_sun * kpc**-2]
1054            L_kron_err_max = (10**(-0.4 * (sb_kron_err_max - M_sun -
    21.572))) * (1000**2)
1055
1056            # L in [L_sun *  k p c  ]
1057            L_petro = (10**(-0.4 * (sb_petro - M_sun - 21.572))) *
    (1000**2)
1058
1059            # L for max. error in [L_sun * kpc**-2]
1060            L_petro_err_max = (10**(-0.4 * (sb_petro_err_max - M_sun -
    21.572))) * (1000**2)
1061
1062            L_kron_log = math.log10(L_kron)
1063
1064            # difference in Luminosity for L and L max. error
1065            L_kron_log_err = math.log10(L_kron) - math.log10(
    L_kron_err_max)
1066            L_petro_log = math.log10(L_petro)
1067
1068            # difference in Luminosity for L and L max. error
1069            L_petro_log_err = math.log10(L_petro) - math.log10(
    L_petro_err_max)
1070
1071            #print L_kron, L_kron_log, L_kron_log_err, L_petro_log,
    L_petro_log_err
1072
```

```
#--------convert L_70 to SFR (calzetti et al 2010)--------------------

#not working (theoretical mistake)
          L_sun = 3.826e33
          L_kron_direct = L_sun * 10**(-0.4*(kron_mag-M_sun))

          # direct conversion (without sb)
          SFR_kron = math.log10(L_kron_direct / 1.7e43)
          L_kron_err_max = L_sun * 10**(-0.4*(kron_mag + kron_err-M_sun
    ))
          SFR_kron_err_max = math.log10(L_kron_err_max / 1.7e43)

          # error kron
          SFR_kron_err = SFR_kron - SFR_kron_err_max

          L_petro_direct = L_sun * 10**(-0.4*(petro_mag-M_sun))

          # direct conversion (without sb)
          SFR_petro = math.log10(L_petro_direct / 1.7e43)
          L_petro_err_max = L_sun * 10**(-0.4*(petro_mag + petro_err-
    M_sun))
          SFR_petro_err_max = math.log10(L_petro_err_max / 1.7e43)

          # error petro
          SFR_petro_err = SFR_petro - SFR_petro_err_max

          #print L_kron_direct, L_petro_direct, SFR_kron, SFR_kron_err,
     SFR_petro, SFR_petro_err
#not working end

# not working

          # sb approach for low metallicity
          SFR_kron_sb = math.log10((L_kron / 3.3e44)**0.65)

          # max. error kron
          SFR_kron_sb_err_max = ((L_kron_err_max / 3.3e44)**0.65)

          # error kron
          SFR_kron_sb_err = SFR_kron_sb_err_max - SFR_kron_sb

          SFR_petro_sb = math.log10(((L_petro / 3.3e44)**0.65))

          # sb approach for low metallicity

          # max. error petro
          SFR_petro_sb_err_max = math.log10((L_petro_err_max / 3.3e44)
    **0.65)

          # error petro
          SFR_petro_sb_err = SFR_petro_sb - SFR_petro_sb_err_max

          #print SFR_kron_sb, SFR_kron_sb_err_max, SFR_petro_sb,
    SFR_petro_sb_err
# not working end

```

```
1124            results_row.extend([SFR_kron,SFR_kron_err,SFR_petro,
    SFR_petro_err])
1125            master_out.writerow(results_row)
1126
1127 B_master_in.close()
1128 out.close()
1129
1130
1131 # R-file
1132
1133 R_master_in = file('R_master_out.cat','r')
1134 out = file('R_table_SFR.cat','w')
1135
1136 master_reader = csv.reader(R_master_in)
1137 master_out = csv.writer(out)
1138
1139 master_out.writerow(["#","RA","DEC","NED","Kron_mag","Kron_err","
    Petro_mag","Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    ","SFR_kron","SFR_kron_err","SFR_petro","SFR_petro_err"])
1140
1141 for master_row in master_reader:
1142     results_row = master_row
1143     if not master_row[0] is "#":
1144        kron_mag = float(master_row[4])
1145        kron_err = float(master_row[5])
1146        petro_mag = float(master_row[6])
1147        petro_err = float(master_row[7])
1148        kron_rad = float(master_row[8])
1149        petro_rad = float(master_row[9])
1150        A_world = float(master_row[10])
1151        A_world_err = float(master_row[11])
1152        B_world = float(master_row[12])
1153        B_world_err = float(master_row[13])
1154        flags = float(master_row[14])
1155        class_star = float(master_row[15])
1156        if kron_mag < 99.0 and petro_mag < 99.0:
1157
1158 #--------kron-surface-brightness in [mag * arcsec**-2]-----------------
1159
1160            A_kron = (A_world * kron_rad) * 3600.0
1161            B_kron = (B_world * kron_rad) * 3600.0
1162            A_kron_err = (A_world_err * kron_rad) * 3600.0
1163            B_kron_err = (B_world_err * kron_rad) * 3600.0
1164
1165            area_kron = A_kron * B_kron * math.pi
1166            area_kron_err_max = (A_kron + A_kron_err) * (B_kron +
    B_kron_err) * math.pi
1167
1168            #print area_kron, master_row[0]
1169            sb_kron = kron_mag + 2.5 * math.log10(area_kron)
1170            sb_kron_err_max = (kron_mag + kron_err) + 2.5 * math.log10(
    area_kron_err_max)
1171            sb_kron_err = sb_kron_err_max - sb_kron
1172
```

```
1173            sb_kron_handbook = kron_mag + 2.5 * math.log10(2827 + (A_kron
    *60) * (B_kron*60))
1174
1175 #--------petro-surface-brightness in [mag * arcsec**-2]----------------
1176
1177            A_petro = (A_world * petro_rad) * 3600.0
1178            B_petro = (B_world * petro_rad) * 3600.0
1179            A_petro_err = (A_world_err * petro_rad) * 3600.0
1180            B_petro_err = (B_world_err * petro_rad) * 3600.0
1181
1182            area_petro = A_petro * B_petro * math.pi
1183            area_petro_err_max = (A_petro + A_petro_err) * (B_petro +
    B_petro_err) * math.pi
1184
1185            sb_petro = petro_mag + 2.5 * math.log10(area_petro)
1186            sb_petro_err_max = (petro_mag + petro_err) + 2.5 * math.log10
    (area_petro_err_max)
1187            sb_petro_err = sb_petro_err_max - sb_petro
1188
1189 #--------convert surface-brightness to L_70 physical units ------------
1190 # ------[L_sun * kpc**-2]-----------------------------------------------
1191
1192            M_sun = 4.74
1193
1194            # L in [L_sun * kpc**-2]
1195            L_kron = (10**(-0.4 * (sb_kron - M_sun - 21.572))) * 1000
1196
1197            # L for max. error in [L_sun * kpc**-2]
1198            L_kron_err_max = (10**(-0.4 * (sb_kron_err_max - M_sun -
    21.572))) * 1000
1199
1200            # L in [L_sun * kpc**-2]
1201            L_petro = (10**(-0.4 * (sb_petro - M_sun - 21.572))) * 1000
1202
1203            # L for max. error in [L_sun * kpc**-2]
1204            L_petro_err_max = (10**(-0.4 * (sb_petro_err_max - M_sun -
    21.572))) * 1000
1205
1206            L_kron_log = math.log10(L_kron)
1207
1208            # difference in Luminosity for L and L max. error
1209            L_kron_log_err = math.log10(L_kron) - math.log10(
    L_kron_err_max)
1210            L_petro_log = math.log10(L_petro)
1211
1212            # difference in Luminosity for L and L max. error
1213            L_petro_log_err = math.log10(L_petro) - math.log10(
    L_petro_err_max)
1214
1215            #print L_kron_log, L_kron_log_err, L_petro_log,
    L_petro_log_err
1216
1217 #--------convert L_70 to SFR (calzetti et al 2010)--------------------
1218
1219            L_sun = 3.826e33
1220            L_kron_direct = L_sun * 10**(-0.4*kron_mag)
```

```
1221
1222            # direct conversion (without sb)
1223            SFR_kron = math.log10(L_kron_direct / 7.0e42)
1224            L_kron_err_max = L_sun * 10**(-0.4*(kron_mag + kron_err))
1225            SFR_kron_err_max = math.log10(L_kron_err_max / 7.0e42)
1226
1227            # error kron
1228            SFR_kron_err = SFR_kron - SFR_kron_err_max
1229
1230            L_petro_direct = L_sun * 10**(-0.4*petro_mag)
1231
1232            # direct conversion (without sb)
1233            SFR_petro = math.log10(L_petro_direct / 7.0e42)
1234            L_petro_err_max = L_sun * 10**(-0.4*(petro_mag + petro_err))
1235            SFR_petro_err_max = math.log10(L_petro_err_max / 7.0e42)
1236
1237            # error petro
1238            SFR_petro_err = SFR_petro - SFR_petro_err_max
1239
1240            #print SFR_kron, SFR_kron_err, SFR_petro, SFR_petro_err
1241
1242            # not working
1243
1244            ## sb approach for low metallicity
1245            #SFR_kron_sb = (L_kron / 3.3e44)**0.65
1246
1247            ## max. error kron
1248            #SFR_kron_sb_err_max = ((L_kron_err / 3.3e44)**0.65)
1249
1250            ## error kron
1251            #SFR_kron_sb_err = SFR_kron_sb_err_max - SFR_kron_sb
1252
1253            ## sb approach for low metallicity
1254            #SFR_petro_sb = math.log10(((L_petro / 3.3e44)**0.65))
1255
1256            ## max. error petro
1257            #SFR_petro_sb_err_max = math.log10((L_petro_err_max / 3.3e44)
    **0.65)
1258
1259            ## error petro
1260            #SFR_petro_sb_err = SFR_petro_sb - SFR_petro_sb_err_max
1261
1262            #print SFR_kron_sb, SFR_kron_sb_err_max, SFR_petro_sb,
    SFR_petro_sb_err
1263            # not working end
1264
1265            results_row.extend([SFR_kron,SFR_kron_err,SFR_petro,
    SFR_petro_err])
1266            master_out.writerow(results_row)
1267
1268 R_master_in.close()
1269 out.close()
1270
1271 #-----------------------------------------------------------------------
1272 #---------merge SFR files-----------------------------------------------
1273 #-----------------------------------------------------------------------
```

```
1274
1275 # ---- merge B and G table ----
1276
1277 B_SFR = file('B_table_SFR.cat','r')
1278 G_SFR = file('G_master_out.cat','r')
1279 out = file('master_table_SFR_G.cat','w')
1280
1281 B_SFR_reader = csv.reader(B_SFR)
1282 G_SFR_reader = csv.reader(G_SFR)
1283 master_out = csv.writer(out)
1284
1285 G_SFR_list = list(G_SFR_reader)
1286
1287 master_out.writerow(["#","RA","DEC","NED","B_Kron_mag","B_Kron_err","
    B_Petro_mag","B_Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    ","B_SFR_kron","B_SFR_kron_err","B_SFR_petro","B_SFR_petro_err","
    G_petro_mag","G_petro_mag_err"])
1288
1289 for B_SFR_row in B_SFR_reader:
1290     for G_SFR_row in G_SFR_list:
1291         if not B_SFR_row[0] is "#" and G_SFR_row[0] != "#":
1292             results_row = B_SFR_row
1293             RA_B = B_SFR_row[1]
1294             DEC_B = B_SFR_row[2]
1295             RA_G = G_SFR_row[1]
1296             DEC_G = G_SFR_row[2]
1297             G_petro_mag = G_SFR_row[6]
1298             G_petro_mag_err = G_SFR_row[7]
1299             if format(float(RA_B),'.2f') == format(float(RA_G),'.2f')
    and format(float(DEC_B),'.2f') == format(float(DEC_G),'.2f'):
1300                 results_row.extend([G_petro_mag,G_petro_mag_err])
1301                 break
1302     else:
1303         results_row.extend(['-','-'])
1304
1305     master_out.writerow(results_row)
1306
1307 B_SFR.close()
1308 G_SFR.close()
1309 out.close()
1310
1311 #---- merge result from above with R table ----
1312
1313 B_SFR = file('master_table_SFR_G.cat','r')
1314 R_SFR = file('R_table_SFR.cat','r')
1315 out = file('master_table_SFR.cat','w')
1316
1317 B_SFR_reader = csv.reader(B_SFR)
1318 R_SFR_reader = csv.reader(R_SFR)
1319 master_out = csv.writer(out)
1320
1321 R_SFR_list = list(R_SFR_reader)
1322
1323 master_out.writerow(["#","RA","DEC","NED","B_Kron_mag","B_Kron_err","
    B_Petro_mag","B_Petro_err","Kron_rad","Petro_rad","A_world","
```

```
    A_world_err","B_world","B_world_err","flags","class_star","filename
    ","B_SFR_kron","B_SFR_kron_err","B_SFR_petro","B_SFR_petro_err","
    G_petro_mag","G_petro_mag_err","R_petro_mag","R_petro_mag_err","
    R_SFR_kron","R_SFR_kron_err","R_SFR_petro","R_SFR_petro_err"])

for B_SFR_row in B_SFR_reader:
    for R_SFR_row in R_SFR_list:
        if not B_SFR_row[0] is "#" and R_SFR_row[0] != "#":
            results_row = B_SFR_row
            RA_B = B_SFR_row[1]
            DEC_B = B_SFR_row[2]
            RA_R = R_SFR_row[1]
            DEC_R = R_SFR_row[2]
            R_petro_mag = R_SFR_row[6]
            R_petro_mag_err = R_SFR_row[7]
            R_SFR_kron = R_SFR_row[17]
            R_SFR_kron_err = R_SFR_row[18]
            R_SFR_petro = R_SFR_row[19]
            R_SFR_petro_err = R_SFR_row[20]
            if format(float(RA_B),'.2f') == format(float(RA_R),'.2f')
    and format(float(DEC_B),'.2f') == format(float(DEC_R),'.2f'):
                results_row.extend([R_petro_mag,R_petro_mag_err,
    R_SFR_kron,R_SFR_kron_err,R_SFR_petro,R_SFR_petro_err])
                break
    else:
        results_row.extend(['-','-','-','-','-','-',])

    master_out.writerow(results_row)

B_SFR.close()
R_SFR.close()
out.close()

#---- assign new identity to all entries ----

infile = file('master_table_SFR.cat','r')
outfile = file('master_table_SFR_complete.cat','w')

in_reader = csv.reader(infile)
out_reader = csv.writer(outfile)

out_reader.writerow(["#","RA","DEC","NED","B_Kron_mag","B_Kron_err","
    B_Petro_mag","B_Petro_err","Kron_rad","Petro_rad","A_world","
    A_world_err","B_world","B_world_err","flags","class_star","filename
    ","B_SFR_kron","B_SFR_kron_err","B_SFR_petro","B_SFR_petro_err","
    G_petro_mag","G_petro_mag_err","R_petro_mag","R_petro_mag_err","
    R_SFR_kron","R_SFR_kron_err","R_SFR_petro","R_SFR_petro_err","
    new_identity"])

i = 0

for row in in_reader:
    if not row[0] is "#":
        RA = row[1]
        DEC = row[2]
        i += 1
```

```
1368        new_identity = 'MESSGC ' + str(i) + ' '  + str(RA) + '+' + str(
    DEC)
1369        row.extend([new_identity])
1370        #break
1371        out_reader.writerow(row)
1372
1373 print "merging table files ... done"
1374 print "ALL OPERATIONS DONE. RESULTFILE: master_table_SFR_complete.cat"
```

A.2 Tables

Table 5: Table of B, G and B-G measurements, including respective errors.

Identity	B [mag]	err	G [mag]	err	B-G [mag]	err	NED
MESSGC 126 30.9697255+64.6875454	4,4917	0,1562	1,4914	0,0208	3,0003	0,1354	
MESSGC 130 31.9205225+64.6890115	3,9060	0,0836	7,3525	1,0653	-3,4465	0,9817	
MESSGC 133 31.1617533+64.7195425	2,6873	0,0373	2,4175	0,0357	0,2698	0,0016	
MESSGC 138 31.3699525+64.735742	4,2534	0,1315	3,6695	0,0716	0,5839	0,0599	
MESSGC 140 31.1018037+64.7554616	3,9195	0,0978	5,0267	0,2633	-1,1072	0,1655	
MESSGC 142 31.4232839+64.7656345	4,5909	0,1031	3,7107	0,0805	0,8802	0,0226	
MESSGC 149 31.7418071+64.8675394	3,0474	0,0453	2,9710	0,0450	0,0764	0,0003	
MESSGC 160 30.9207605+64.9603164	1,1170	0,0122	1,0939	0,0133	0,0231	0,0011	IRAS 02000+6443
MESSGC 162 30.978771+64.9364142	3,7267	0,0782	2,1782	0,0251	1,5485	0,0531	
MESSGC 164 31.7720999+64.9446652	3,7089	0,0728	2,0864	0,0328	1,6225	0,0400	2XMM J020705.1+645642
MESSGC 169 31.6861212+65.0364552	3,9301	0,1103	2,5575	0,0440	1,3726	0,0663	
MESSGC 267 351.1626644+58.8215071	2,9697	0,1009	2,4772	0,0770	0,4925	0,0239	
MESSGC 268 350.4545831+58.8550421	3,0012	0,0745	3,8217	0,1994	-0,8205	0,1249	
MESSGC 270 350.7280862+58.8704026	1,9290	0,0413	1,7331	0,0359	0,1959	0,0054	
MESSGC 273 83.8829985+22.0160891	2,7290	0,0715	1,0943	0,0193	1,6347	0,0522	2MASX J05353194+2200569
MESSGC 278 146.8738951+13.0189016	4,3154	0,1159	3,7673	0,1162	0,5481	0,0003	SDSS J094729.79+130107.1
MESSGC 280 147.0344309+13.0473411	3,8626	0,0813	2,6130	0,0548	1,2496	0,0265	SDSS J094808.32+130248.9
MESSGC 284 147.0871447+13.2232555	2,6132	0,0367	2,5305	0,0510	0,0827	0,0143	2MASX J09482083+1313233
MESSGC 288 146.7188401+13.3319888	2,7206	0,0271	2,7634	0,0609	-0,0428	0,0338	2MASSi J0946525+131953
MESSGC 291 147.1658069+13.4170536	3,2861	0,0553	3,1735	0,0679	0,1126	0,0126	SDSSCGB 59955.02
MESSGC 299 146.9219827+13.523776	2,2131	0,0192	1,7458	0,0261	0,4673	0,0069	2MASX J09474126+1331248
MESSGC 305 308.1207123+40.1772892	-8,2978	0,0002	-3,6762	0,0033	-4,6216	0,0031	
MESSGC 306 308.0715788+40.1174138	-3,0784	0,0054	-3,5116	0,0038	0,4332	0,0016	2MASX J20321702+4007080
MESSGC 307 308.0721604+40.1003438	1,2518	0,0650	0,1535	0,0444	1,0983	0,0206	
MESSGC 310 308.058871+40.1108631	-1,3228	0,0086	-1,7598	0,0089	0,4370	0,0003	
MESSGC 313 308.0832558+40.1166276	0,5336	0,0512	1,8058	0,1588	-1,2722	0,1076	
MESSGC 314 308.0867074+40.1368599	1,2159	0,1182	-4,6241	0,0023	5,8400	0,1159	
MESSGC 315 308.063091+40.1252014	-2,1689	0,0063	-2,7598	0,0043	0,5909	0,0020	
MESSGC 319 308.0852966+40.1411674	-2,0804	0,0056	-4,6241	0,0023	2,5437	0,0033	
MESSGC 320 308.0937182+40.1352259	-0,9778	0,0092	-4,6241	0,0023	3,6463	0,0069	
MESSGC 321 308.1346679+40.1626929	-3,4023	0,0023	-4,5585	0,0011	1,1562	0,0012	
MESSGC 324 308.1214017+40.1601456	-3,8012	0,0016	-4,8038	0,0010	1,0026	0,0006	
MESSGC 325 308.1160579+40.1481703	-3,2182	0,0019	-5,8075	0,0008	2,5893	0,0011	
MESSGC 326 308.1201603+40.1521588	-3,2263	0,0020	-5,8075	0,0008	2,5812	0,0012	
MESSGC 327 308.0056119+40.2503176	-4,7843	0,0025	-6,3587	0,0009	1,5744	0,0016	
MESSGC 328 308.0727823+40.1558125	-0,7900	0,0154	-0,6318	0,0195	-0,1582	0,0041	
MESSGC 329 308.1269053+40.1649474	-3,8743	0,0015	-4,5585	0,0011	0,6842	0,0004	
MESSGC 330 308.1069856+40.1567003	-2,4320	0,0030	-3,7657	0,0039	1,3337	0,0009	
MESSGC 332 308.123817+40.1707468	-2,7172	0,0024	-4,9048	0,0014	2,1876	0,0010	
MESSGC 334 308.0600446+40.3275143	-6,5885	0,0004	-6,6690	0,0004	0,0805	0,0000	
MESSGC 337 308.0887736+40.2672226	-6,5488	0,0004	1,2467	0,0473	-7,7955	0,0469	
MESSGC 338 308.0703875+40.1923003	-0,8515	0,0102	-2,0756	0,0062	1,2241	0,0040	
MESSGC 339 308.0695374+40.194678	-0,9998	0,0085	-2,0756	0,0062	1,0758	0,0023	
MESSGC 340 308.0720533+40.1950042	-1,0147	0,0093	-0,7701	0,0119	-0,2446	0,0026	
MESSGC 344 308.0687859+40.1986892	-1,0047	0,0102	-0,7701	0,0119	-0,2346	0,0017	
MESSGC 345 308.0805023+40.2028979	-1,0595	0,0107	-1,6484	0,0081	0,5889	0,0026	
MESSGC 349 308.0662456+40.2008784	-0,6943	0,0108	-0,7701	0,0119	0,0758	0,0011	
MESSGC 352 308.0740913+40.2079707	-0,8863	0,0105	-0,8999	0,0117	0,0136	0,0012	
MESSGC 353 308.0345324+40.2097995	-0,6982	0,0158	0,0503	0,0244	-0,7485	0,0086	
MESSGC 355 308.0013412+40.2085111	-0,3734	0,0137	0,9748	0,0698	-1,3482	0,0561	
MESSGC 356 308.0042015+40.2150732	-1,2303	0,0106	-1,2975	0,0105	0,0672	0,0001	
MESSGC 357 308.0041302+40.2096898	-0,3323	0,0141	0,9748	0,0698	-1,3071	0,0557	
MESSGC 358 308.0334721+40.2136691	0,5110	0,0339	0,0503	0,0244	0,4607	0,0095	
MESSGC 359 308.0423387+40.2342005	-0,7706	0,0135	-0,4436	0,0165	-0,3270	0,0030	
MESSGC 360 308.0179712+40.2173963	1,2295	0,0653	1,4448	0,0920	-0,2153	0,0267	
MESSGC 362 308.0863988+40.2145622	-1,2816	0,0074	-1,7497	0,0064	0,4681	0,0010	
MESSGC 364 307.7823962+40.2890732	-5,6530	0,0004	-6,0754	0,0005	0,4224	0,0001	
MESSGC 365 307.7590009+40.219545	3,8015	0,3424	3,1101	0,2256	0,6914	0,1168	
MESSGC 366 308.0066228+40.2232623	-0,5611	0,0111	-0,7705	0,0100	0,2094	0,0011	
MESSGC 367 307.7405604+40.2203276	2,3458	0,1385	-0,4736	0,0141	2,8194	0,1244	
MESSGC 368 308.0837549+40.2232236	0,7220	0,0377	2,0406	0,1284	-1,3186	0,0907	
MESSGC 369 308.0065551+40.2284421	-1,3032	0,0093	-1,6058	0,0068	0,3026	0,0025	
MESSGC 371 308.0240643+40.2513564	-4,2739	0,0021	-3,8815	0,0016	-0,3924	0,0005	
MESSGC 372 307.9029186+40.2298935	-0,1432	0,0102	-0,8652	0,0076	0,7220	0,0026	
MESSGC 374 308.0308122+40.2464692	-3,8842	0,0020	-3,9690	0,0020	0,0848	0,0000	
MESSGC 376 308.0077304+40.2386737	-1,8235	0,0050	-1,8482	0,0050	0,0247	0,0000	
MESSGC 377 308.0213353+40.2656353	-4,8419	0,0017	-3,2554	0,0025	-1,5865	0,0008	
MESSGC 378 307.9031985+40.2331192	-0,1127	0,0155	-0,8652	0,0076	0,7525	0,0079	
MESSGC 379 307.7609315+40.2976267	-4,8713	0,0014	-5,6852	0,0012	0,8139	0,0002	
MESSGC 380 308.0744753+40.2355886	-0,1650	0,0154	0,5703	0,0373	-0,7353	0,0219	
MESSGC 382 308.0420278+40.2483301	-3,0386	0,0027	-2,8788	0,0031	-0,1598	0,0004	
MESSGC 383 307.8052583+40.2447216	-0,6871	0,0118	-1,6493	0,0064	0,9622	0,0054	
MESSGC 384 307.9952843+40.2633635	-4,1273	0,0014	-4,3090	0,0015	0,1817	0,0001	
MESSGC 385 308.0287249+40.2590274	-3,8774	0,0020	-3,9827	0,0020	0,1053	0,0000	
MESSGC 386 307.7776423+40.29163	-5,1046	0,0012	-6,0754	0,0005	0,9708	0,0007	2MASX J20310684+4017313
MESSGC 387 308.0426872+40.2534965	-2,2757	0,0040	-2,8788	0,0031	0,6031	0,0009	
MESSGC 388 308.0109564+40.2547181	-3,9579	0,0014	-6,3587	0,0009	2,4008	0,0005	
MESSGC 389 308.0032239+40.2636901	-3,9249	0,0016	-4,3090	0,0015	0,3841	0,0001	
MESSGC 390 308.0490946+40.2569932	-0,6457	0,0128	0,3364	0,0278	-0,9821	0,0150	
MESSGC 391 307.954402+40.2575778	-0,4813	0,0163	0,0785	0,0249	-0,5598	0,0086	
MESSGC 392 308.1756167+40.2936489	-5,9436	0,0006	-4,1910	0,0013	-1,7526	0,0007	
MESSGC 393 307.9633524+40.2594775	-0,2171	0,0168	-0,1742	0,0215	-0,0429	0,0047	
MESSGC 394 307.5922555+40.2564545	0,0949	0,0179	-1,1546	0,0102	1,2495	0,0077	
MESSGC 395 307.9607899+40.2601052	-0,7703	0,0110	-0,1742	0,0215	-0,5961	0,0105	
MESSGC 397 308.0454941+40.2609101	-0,8839	0,0099	0,3364	0,0278	-1,2203	0,0179	
MESSGC 398 307.8005225+40.268763	-2,2027	0,0051	-3,2237	0,0030	1,0210	0,0021	
MESSGC 399 308.0363094+40.2689224	-0,7836	0,0172	-0,7798	0,0180	-0,0038	0,0008	

Identity	B [mag]	err	G [mag]	err	B-G [mag]	err	NED
MESSGC 400 307.8035747+40.2727934	-2,0403	0,0050	-3,2237	0,0030	1,1834	0,0020	
MESSGC 401 307.8055989+40.2662383	-1,5068	0,0065	-2,4983	0,0041	0,9915	0,0024	
MESSGC 402 307.9954028+40.2716658	-2,3970	0,0036	-2,3494	0,0042	-0,0476	0,0006	
MESSGC 403 307.8106895+40.2685197	-1,8127	0,0050	-2,4983	0,0041	0,6856	0,0009	
MESSGC 404 307.9853007+40.2683822	-1,7511	0,0044	-3,2758	0,0023	1,5247	0,0021	
MESSGC 405 307.9864507+40.2705089	-2,4086	0,0032	-3,2758	0,0023	0,8672	0,0009	
MESSGC 407 307.9860567+40.2728032	-2,5161	0,0030	-3,2758	0,0023	0,7597	0,0007	
MESSGC 408 307.9900668+40.2720318	-2,1497	0,0034	-3,2758	0,0023	1,1261	0,0011	
MESSGC 409 308.0013949+40.2726603	-1,4030	0,0055	-2,3494	0,0042	0,9464	0,0013	
MESSGC 410 307.9893478+40.2738512	-1,6366	0,0045	-3,2758	0,0023	1,6392	0,0022	
MESSGC 411 307.6207156+40.275236	1,1931	0,0509	0,1704	0,0267	1,0227	0,0242	
MESSGC 412 307.5712177+40.2787882	-0,9860	0,0096	-2,2090	0,0079	1,2230	0,0017	
MESSGC 414 307.5865739+40.2817215	-1,3236	0,0080	-2,1604	0,0056	0,8368	0,0024	
MESSGC 415 307.7847745+40.2872049	-3,0966	0,0011	-6,0754	0,0005	2,9788	0,0006	
MESSGC 416 308.0732929+40.3272613	-3,0617	0,0038	-3,7787	0,0016	0,7170	0,0022	
MESSGC 417 307.9793356+40.2898615	-0,2505	0,0185	0,8496	0,0567	-1,1001	0,0382	
MESSGC 418 307.564771+40.2874158	-0,6562	0,0115	-2,9721	0,0034	2,3159	0,0081	
MESSGC 420 307.7809756+40.2979774	-4,5940	0,0008	-4,2587	0,0009	-0,3353	0,0001	
MESSGC 423 308.0267761+40.3020964	-1,0092	0,0092	-3,0111	0,0034	2,0019	0,0058	
MESSGC 424 307.5721934+40.2912475	0,1109	0,0182	-2,0579	0,0053	2,1688	0,0129	
MESSGC 426 307.7519806+40.307563	-2,3669	0,0039	-2,7228	0,0037	0,3559	0,0002	
MESSGC 427 307.9230084+40.2982493	-0,2751	0,0094	-1,3774	0,0068	1,1023	0,0026	
MESSGC 428 308.0489638+40.3254661	-5,0111	0,0006	-5,0838	0,0007	0,0727	0,0001	
MESSGC 429 307.6373267+40.2984425	-0,0362	0,0165	-1,4788	0,0087	1,4426	0,0078	
MESSGC 430 308.1411183+40.3009711	-3,3665	0,0009	-4,0523	0,0007	0,6858	0,0002	
MESSGC 432 308.1309597+40.3007364	-4,3800	0,0006	-2,7912	0,0018	-1,5888	0,0012	
MESSGC 434 308.0569421+40.3181072	-3,0017	0,0009	-3,2285	0,0009	0,2268	0,0000	2MASX J20321363+4018570
MESSGC 436 307.6203135+40.309678	-0,5947	0,0144	-0,8547	0,0164	0,2600	0,0020	
MESSGC 437 308.0815769+40.3179036	-1,9299	0,0067	-4,1328	0,0026	2,2029	0,0041	
MESSGC 441 307.9924184+40.3101539	-1,4064	0,0034	-2,0132	0,0031	0,6068	0,0003	
MESSGC 442 307.9899371+40.3090182	1,1258	0,0283	-2,0132	0,0031	3,1390	0,0252	
MESSGC 443 308.0109319+40.3107884	-0,6520	0,0078	-1,9372	0,0044	1,2852	0,0034	
MESSGC 445 307.6342782+40.3145126	-0,0007	0,0182	-1,0863	0,0126	1,0856	0,0056	
MESSGC 446 308.081836+40.3229699	-2,6041	0,0028	-4,1328	0,0026	1,5287	0,0002	
MESSGC 447 307.6803825+40.3218673	-1,2316	0,0060	-2,2150	0,0032	0,9834	0,0028	
MESSGC 448 307.6745482+40.3223343	-0,8534	0,0081	-1,9516	0,0039	1,0982	0,0042	
MESSGC 449 308.0962599+40.3230307	-0,5657	0,0065	-1,9838	0,0038	1,4181	0,0027	
MESSGC 450 307.6010704+40.3249283	-0,1928	0,0147	-2,4297	0,0039	2,2369	0,0108	
MESSGC 451 308.091685+40.3273463	-3,0067	0,0013	-5,4608	0,0014	2,4541	0,0001	
MESSGC 452 308.0869045+40.3269316	-1,9704	0,0027	-5,4608	0,0014	3,4904	0,0013	
MESSGC 453 308.1194402+40.3282427	-2,4958	0,0018	-2,6875	0,0020	0,1917	0,0002	
MESSGC 454 308.0863582+40.3305947	-2,2021	0,0024	-5,4608	0,0014	3,2587	0,0010	
MESSGC 455 308.0923353+40.3381674	-4,1845	0,0006	-5,2110	0,0003	1,0265	0,0003	
MESSGC 456 308.0883854+40.333417	-2,3901	0,0019	-5,4608	0,0014	3,0707	0,0005	
MESSGC 457 307.5953008+40.3346577	-0,9270	0,0061	-0,9334	0,0130	0,0064	0,0069	2MASX J20302211+4020062
MESSGC 458 308.0899995+40.3374475	-3,7719	0,0006	-5,2110	0,0003	1,4391	0,0003	
MESSGC 459 308.0961253+40.3393156	-2,8977	0,0011	-3,7341	0,0007	0,8364	0,0004	
MESSGC 460 307.7369504+40.339464	-0,2525	0,0133	-1,2913	0,0069	1,0388	0,0064	
MESSGC 461 308.0883475+40.3406124	-2,1193	0,0021	-5,2110	0,0003	3,0917	0,0018	
MESSGC 463 307.6389463+40.3438828	-0,4955	0,0136	-0,8005	0,0130	0,3050	0,0006	
MESSGC 464 307.6346078+40.3429923	-0,4631	0,0136	-0,8820	0,0110	0,4189	0,0026	
MESSGC 466 308.0122814+40.347913	-1,4005	0,0056	-4,1231	0,0020	2,7226	0,0036	
MESSGC 469 308.0073595+40.3569167	-1,2079	0,0064	-2,7915	0,0040	1,5836	0,0024	
MESSGC 470 308.1820721+40.3561137	0,1542	0,0184	-0,1101	0,0175	0,2643	0,0009	
MESSGC 471 307.6883854+40.3640736	-1,8593	0,0037	-2,3795	0,0034	0,5202	0,0003	2MASX J20304522+4021501
MESSGC 473 308.2162804+40.3644841	2,5754	0,1117	1,8824	0,0864	0,6930	0,0253	
MESSGC 474 308.2310625+40.3645988	1,8210	0,0601	1,2676	0,0626	0,5534	0,0025	
MESSGC 475 308.013517+40.3711546	1,8988	0,0834	-1,8310	0,0080	3,7298	0,0754	
MESSGC 476 308.232644+40.3728518	-0,0357	0,0124	-0,8595	0,0078	0,8238	0,0046	
MESSGC 477 307.8322671+40.3746892	1,8411	0,0656	0,2539	0,0318	1,5872	0,0338	
MESSGC 479 307.9511297+40.4518966	0,1437	0,0200	-0,8973	0,0107	1,0410	0,0093	
MESSGC 480 308.1224032+40.424648	-2,5513	0,0030	-4,0522	0,0028	1,5009	0,0002	
MESSGC 483 308.131711+40.4271812	-4,0608	0,0017	-4,2346	0,0017	0,1738	0,0000	

Table 6: Table of B, R and B-R measurements, including respective errors.

Identity	B [mag]	err	R [mag]	err	B-R [mag]	err	NED
MESSGC 10 207.3577877+-28.547637	1,4017	0,0121	0,6545	0,0167	0,7472	0,0046	2MASX J13492603-2832535
MESSGC 107 191.3273751+45.5000278	3,9067	0,0721	2,5768	0,0828	1,3299	0,0107	
MESSGC 113 321.1003397+-69.7112037	-0,5231	0,0026	-0,6380	0,0052	0,1149	0,0026	SUMSS J212424-694238
MESSGC 12 207.0421381+-28.4987933	3,6057	0,0797	2,5612	0,0876	1,0445	0,0079	GALEXASC J134810.31-282957.5
MESSGC 133 31.1617533+64.7195425	2,6873	0,0373	2,0952	0,0595	0,5921	0,0222	
MESSGC 14 207.1252485+-28.4526398	3,5737	0,0837	1,7501	0,0541	1,8236	0,0296	
MESSGC 162 30.978771+64.9364142	3,7267	0,0782	2,6290	0,1086	1,0977	0,0304	
MESSGC 164 31.7720999+64.9446652	3,7089	0,0728	1,1473	0,0342	2,5616	0,0386	2XMM J020705.1+645642
MESSGC 17 207.3287873+-28.42517	2,3430	0,0228	1,8765	0,0449	0,4665	0,0221	
MESSGC 178 315.6089502+36.6955932	-7,1141	0,0001	-5,5438	0,0001	-1,5703	0,0000	
MESSGC 190 315.6126971+36.703051	-6,2982	0,0000	-5,5438	0,0001	-0,7544	0,0001	
MESSGC 194 315.5464823+36.6913802	-1,2176	0,0024	-1,0320	0,0061	-0,1856	0,0037	
MESSGC 198 315.6034031+36.6673031	-5,9430	0,0001	-4,7738	0,0002	-1,1692	0,0001	
MESSGC 199 315.5959548+36.6739446	-1,1491	0,0026	-4,7738	0,0002	3,6247	0,0024	
MESSGC 2 288.9608368+-7.0109392	0,9337	0,0357	0,9337	0,0357	0,0000	0,0000	GALEXASC J191550.60-070036.8
MESSGC 200 315.599393+36.6708252	-0,3452	0,0049	-4,7738	0,0002	4,4286	0,0047	
MESSGC 202 315.5702662+36.6652144	-0,2178	0,0050	-5,3776	0,0001	5,1598	0,0049	
MESSGC 213 29.754247+45.3894646	2,7896	0,0478	2,3607	0,0780	0,4289	0,0302	2MASXi J0159009+452322
MESSGC 23 207.1688204+-28.3487131	1,4859	0,0123	1,1248	0,0325	0,3611	0,0202	GALEXASC J134840.45-282056.5
MESSGC 231 88.8421463+7.1542983	2,9951	0,0622	3,1517	0,1495	-0,1566	0,0873	
MESSGC 233 89.0643486+7.3564067	3,7910	0,1039	2,2452	0,1013	1,5458	0,0026	
MESSGC 237 88.9670715+7.3955513	2,6293	0,0459	1,1406	0,0421	1,4887	0,0038	
MESSGC 253 88.660673+7.5076577	2,2913	0,0339	2,3008	0,0737	-0,0095	0,0398	
MESSGC 256 88.66853+7.516462	3,2525	0,0769	4,5950	0,6327	-1,3425	0,5558	
MESSGC 258 88.6655294+7.5422145	2,9679	0,0629	2,2859	0,0712	0,6820	0,0083	
MESSGC 280 147.0344309+13.0473411	3,8626	0,0813	2,2712	0,0569	1,5914	0,0244	SDSS J094808.32+130248.9
MESSGC 284 147.0871447+13.2232555	2,6132	0,0367	1,5529	0,0399	1,0603	0,0032	2MASX J09482083+1313233
MESSGC 289 146.6016421+13.3455446	4,5331	0,1242	6,0808	1,3031	-1,5477	1,1789	SDSS J094624.31+132042.9
MESSGC 290 147.1441421+13.3691042	3,5448	0,0675	1,8975	0,0457	1,6473	0,0218	
MESSGC 291 147.1658069+13.4170536	3,2861	0,0553	2,6149	0,0933	0,6712	0,0380	SDSSCGB 59955.02
MESSGC 299 146.9219827+13.523776	2,2131	0,0192	1,8781	0,0351	0,3350	0,0159	2MASX J09474126+1331248
MESSGC 300 147.2077519+13.4811862	2,8208	0,0421	2,0192	0,0514	0,8016	0,0093	SDSS J094850.31+132850.3
MESSGC 303 147.2315527+13.4544467	1,7036	0,0212	1,1119	0,0298	0,5917	0,0086	2MASX J09485546+1327144
MESSGC 32 206.9833328+-28.2538641	2,8813	0,0458	1,8442	0,0611	1,0371	0,0153	
MESSGC 327 308.0056119+40.2503176	-4,7843	0,0025	-4,2187	0,0048	-0,5656	0,0023	
MESSGC 334 308.0600446+40.3275143	-6,5885	0,0004	-6,6376	0,0013	0,0491	0,0009	
MESSGC 34 207.2248267+-28.3053562	0,5210	0,0099	0,1272	0,0153	0,3938	0,0054	
MESSGC 361 307.7732178+40.310055	-5,3391	0,0018	-5,4018	0,0030	0,0627	0,0012	
MESSGC 364 307.7823962+40.2890732	-5,6530	0,0004	-6,3240	0,0017	0,6710	0,0013	
MESSGC 367 307.7405604+40.2203276	2,3458	0,1385	-2,8188	0,0109	5,1646	0,1276	
MESSGC 37 207.2190531+-28.3057352	0,6282	0,0075	0,1272	0,0153	0,5010	0,0078	
MESSGC 371 308.0240643+40.2513564	-4,2739	0,0021	-3,8637	0,0049	-0,4102	0,0028	
MESSGC 372 307.9029186+40.2298935	-0,1432	0,0102	-1,8883	0,0242	1,7451	0,0140	
MESSGC 378 307.9031985+40.2331192	-0,1127	0,0155	-1,8883	0,0242	1,7756	0,0087	
MESSGC 379 307.7609315+40.2976267	-4,8713	0,0014	-5,1589	0,0038	0,2876	0,0024	
MESSGC 38 207.2216935+-28.3078387	0,8949	0,0102	0,1272	0,0153	0,7677	0,0051	
MESSGC 383 307.8052583+40.2447216	-0,6871	0,0118	-3,6407	0,0070	2,9536	0,0048	
MESSGC 384 307.9952843+40.2633635	-4,1273	0,0014	-3,3619	0,0096	-0,7654	0,0082	
MESSGC 385 308.0287249+40.2590274	-3,8774	0,0020	-3,6230	0,0059	-0,2544	0,0039	
MESSGC 386 307.7776423+40.29163	-5,1046	0,0012	-6,3240	0,0017	1,2194	0,0005	2MASX J20310684+4017313
MESSGC 388 308.0109564+40.2547181	-3,9579	0,0014	-4,2187	0,0048	0,2608	0,0034	
MESSGC 389 308.0032239+40.2636901	-3,9249	0,0016	-3,3619	0,0096	-0,5630	0,0080	
MESSGC 401 307.8055989+40.2662383	-1,5068	0,0065	-4,2465	0,0043	2,7397	0,0022	
MESSGC 403 307.8106895+40.2685197	-1,8127	0,0050	-4,2465	0,0043	2,4338	0,0007	
MESSGC 411 307.6207156+40.275236	1,1931	0,0509	-0,2943	0,0940	1,4874	0,0431	
MESSGC 415 307.7847745+40.2872049	-3,0966	0,0011	-6,3240	0,0017	3,2274	0,0006	
MESSGC 416 308.0732929+40.3272613	-3,0617	0,0038	-6,2088	0,0010	3,1471	0,0028	
MESSGC 420 307.7809756+40.2979774	-4,5940	0,0008	-4,8013	0,0023	0,2073	0,0015	
MESSGC 421 308.0876644+40.3191508	-1,0917	0,0169	-5,5495	0,0015	4,4578	0,0154	
MESSGC 426 307.7519806+40.307563	-2,3669	0,0039	-2,8794	0,0118	0,5125	0,0079	
MESSGC 427 307.9230084+40.2982493	-0,2751	0,0094	-2,5641	0,0140	2,2890	0,0046	
MESSGC 436 307.6203135+40.309678	-0,5947	0,0144	-0,3897	0,1182	-0,2050	0,1038	
MESSGC 437 308.0815769+40.3179036	-1,9299	0,0067	-5,7090	0,0014	3,7791	0,0053	
MESSGC 441 307.9924184+40.3101539	-1,4064	0,0034	-3,6145	0,0058	2,2081	0,0024	
MESSGC 442 307.9899371+40.3090182	1,1258	0,0283	-3,6145	0,0058	4,7403	0,0225	
MESSGC 443 308.0109319+40.3107884	-0,6520	0,0078	-4,0037	0,0046	3,3517	0,0032	
MESSGC 444 308.0857371+40.3217674	-2,7795	0,0024	-5,5495	0,0015	2,7700	0,0009	
MESSGC 446 308.081836+40.3229699	-2,6041	0,0028	-5,7090	0,0014	3,1049	0,0014	
MESSGC 447 307.6803825+40.3218673	-1,2316	0,0060	-3,3824	0,0074	2,1508	0,0014	
MESSGC 448 307.6745482+40.3223343	-0,8534	0,0081	-3,0408	0,0077	2,1874	0,0004	
MESSGC 45 76.3863619+1.131226	1,5731	0,0259	1,6945	0,0469	-0,1214	0,0210	NVSS J050532+010755
MESSGC 451 308.091685+40.3273463	-3,0067	0,0013	-6,2505	0,0008	3,2438	0,0005	
MESSGC 452 308.0869045+40.3269316	-1,9704	0,0027	-6,2505	0,0008	4,2801	0,0019	
MESSGC 453 308.1194402+40.3282427	-2,4958	0,0018	-7,5292	0,0013	5,0334	0,0005	
MESSGC 454 308.0863582+40.3305947	-2,2021	0,0024	-6,2505	0,0008	4,0484	0,0016	
MESSGC 455 308.0923353+40.3381674	-4,1845	0,0006	-6,1681	0,0008	1,9836	0,0002	
MESSGC 456 308.0883854+40.333417	-2,3901	0,0019	-6,2505	0,0008	3,8604	0,0011	
MESSGC 458 308.0899995+40.3374475	-3,7719	0,0006	-6,1681	0,0008	2,3962	0,0002	
MESSGC 46 85.7746484+-46.4292732	1,7488	0,0286	1,0883	0,0271	0,6605	0,0015	GALEXASC J054306.05-462548.3
MESSGC 461 308.0883475+40.3406124	-2,1193	0,0021	-6,1681	0,0008	4,0488	0,0013	
MESSGC 462 308.0305815+40.3492777	-2,6675	0,0032	-5,6601	0,0022	2,9926	0,0010	
MESSGC 466 308.0122814+40.347913	-1,4005	0,0056	-4,7157	0,0039	3,3152	0,0017	
MESSGC 469 308.0073595+40.3569167	-1,2079	0,0064	-4,3770	0,0048	3,1691	0,0016	
MESSGC 47 240.7187536+47.2256369	0,7835	0,0120	-0,2872	0,0132	1,0707	0,0012	UGC 10156 NED02
MESSGC 471 307.6883854+40.3640736	-1,8593	0,0037	-3,3132	0,0086	1,4539	0,0049	2MASX J20304522+4021501
MESSGC 472 308.147942+40.4242167	-4,8924	0,0022	-5,1513	0,0032	0,2589	0,0010	
MESSGC 475 308.013517+40.3711546	1,8988	0,0834	-4,3767	0,0050	6,2755	0,0784	
MESSGC 477 307.8322671+40.3746892	1,8411	0,0656	-2,7219	0,0132	4,5630	0,0524	
MESSGC 479 307.9511297+40.4518966	0,1437	0,0200	-2,1392	0,0203	2,2829	0,0003	
MESSGC 48 240.7123455+47.226161	0,9636	0,0104	-1,3810	0,0060	2,3446	0,0044	UGC 10156

Identity	B [mag]	err	R [mag]	err	B-R [mag]	err	NED
MESSGC 480 308.1224032+40.424648	-2,5513	0,0030	-3,8894	0,0066	1,3381	0,0036	
MESSGC 487 266.2200979+49.9537108	3,0054	0,0572	2,6290	0,0615	0,3764	0,0043	
MESSGC 49 240.7067078+47.225061	-0,3673	0,0048	-1,3810	0,0060	1,0137	0,0012	UGC 10156 NED01
MESSGC 494 337.2393209+54.8768783	2,1150	0,0184	-1,2014	0,0128	3,3164	0,0056	
MESSGC 521 294.071941+29.5777882	3,6727	0,0956	2,5772	0,0710	1,0955	0,0246	
MESSGC 532 291.8108344+11.484463	-0,4996	0,0027	0,1757	0,0226	-0,6753	0,0199	
MESSGC 547 325.9870749+58.5964241	-0,0486	0,0043	1,8945	0,0913	-1,9431	0,0870	
MESSGC 564 168.7426541+54.9608307	2,3126	0,0350	1,5406	0,0395	0,7720	0,0045	MCG +09-19-014
MESSGC 567 168.6666879+55.102439	2,3046	0,0278	2,0521	0,0626	0,2525	0,0348	SBS 1111+553
MESSGC 568 25.7105545+51.5960878	1,8690	0,0185	1,7335	0,0337	0,1355	0,0152	
MESSGC 571 25.4611068+51.6350312	3,4321	0,0551	2,8199	0,0825	0,6122	0,0274	
MESSGC 574 283.3301893+33.0660501	0,9170	0,0160	-0,5456	0,0119	1,4626	0,0041	IC 1296
MESSGC 576 299.8741591+22.7087839	-2,9533	0,0016	-2,9288	0,0035	-0,0245	0,0019	2MASX J19592944+2242257
MESSGC 577 299.9196107+22.7661853	2,4021	0,0409	-0,8302	0,0108	3,2323	0,0301	
MESSGC 585 299.920387+22.7427843	-0,1367	0,0050	-1,0987	0,0053	0,9620	0,0003	
MESSGC 586 299.9177457+22.7407426	0,1711	0,0053	-1,0987	0,0053	1,2698	0,0000	
MESSGC 588 299.9247545+22.7385965	0,6477	0,0060	-1,0987	0,0053	1,7464	0,0007	
MESSGC 589 299.9286158+22.7407826	-0,1120	0,0045	-1,2750	0,0058	1,1630	0,0013	
MESSGC 591 299.9217905+22.7390216	0,2210	0,0052	-1,0987	0,0053	1,3197	0,0001	
MESSGC 592 299.9197098+22.7366935	-0,3548	0,0041	-1,0987	0,0053	0,7439	0,0012	
MESSGC 593 299.9330838+22.7360659	-0,8413	0,0039	-1,2750	0,0058	0,4337	0,0019	
MESSGC 595 299.9257074+22.732282	-0,3356	0,0041	-1,2020	0,0049	0,8664	0,0008	
MESSGC 597 299.9288424+22.7291496	0,8889	0,0089	-1,2020	0,0049	2,0909	0,0040	
MESSGC 599 299.8783869+22.7000153	-0,7784	0,0027	-1,4269	0,0040	0,6485	0,0013	
MESSGC 6 207.1337216+-28.5650403	3,2002	0,0607	2,3701	0,0695	0,8301	0,0088	2MASX J13483220-2833559
MESSGC 604 299.8848932+22.70324	-0,5356	0,0030	-1,4269	0,0040	0,8913	0,0010	
MESSGC 605 299.8828256+22.7014905	-0,6088	0,0028	-1,4269	0,0040	0,8181	0,0012	
MESSGC 606 299.8834388+22.6988308	-1,7085	0,0019	-1,4269	0,0040	-0,2816	0,0021	
MESSGC 607 299.8811399+22.6953514	-1,6067	0,0021	-1,4269	0,0040	-0,1798	0,0019	
MESSGC 608 299.9376673+22.6973684	3,1921	0,0622	2,4380	0,0827	0,7541	0,0205	
MESSGC 610 299.8773544+22.6932341	-0,3463	0,0041	-0,5356	0,0102	0,1893	0,0061	
MESSGC 612 299.8819939+22.6870217	-0,1317	0,0047	-0,5356	0,0102	0,4039	0,0055	
MESSGC 615 299.8871856+22.6849173	-0,1276	0,0047	-0,7860	0,0088	0,6584	0,0041	
MESSGC 616 299.891873+22.6834957	0,1860	0,0061	-0,7860	0,0088	0,9720	0,0027	
MESSGC 617 299.8895735+22.6828229	-0,0422	0,0054	-0,7860	0,0088	0,7438	0,0034	
MESSGC 620 311.5430669+39.9655452	-0,6644	0,0052	-2,9167	0,0031	2,2523	0,0021	
MESSGC 621 311.5837328+39.9782178	0,2287	0,0109	-2,4617	0,0066	2,6904	0,0043	
MESSGC 622 311.6110227+39.989576	-0,6306	0,0069	-4,3694	0,0020	3,7388	0,0049	
MESSGC 623 311.6082749+39.999878	-1,7468	0,0034	-4,6787	0,0021	2,9319	0,0013	
MESSGC 624 311.6170907+40.0101968	-2,3656	0,0035	-4,2689	0,0026	1,9033	0,0009	
MESSGC 627 311.6933418+40.0209046	0,9804	0,0173	-0,5793	0,0282	1,5597	0,0109	
MESSGC 629 311.5993782+40.0439928	-1,1468	0,0051	-3,4097	0,0048	2,2629	0,0003	
MESSGC 634 58.3246466+11.5714531	1,6802	0,0292	1,1021	0,0458	0,5781	0,0166	
MESSGC 638 279.3945664+-5.4391618	-1,4370	0,0061	-3,6950	0,0106	2,2580	0,0045	
MESSGC 639 279.392306+-5.4393236	-1,1256	0,0076	-3,6950	0,0106	2,5694	0,0030	
MESSGC 640 279.3984098+-5.4441642	-0,7860	0,0089	-2,8300	0,0206	2,0440	0,0117	
MESSGC 642 34.9250722+-3.1193505	1,7537	0,0318	0,2689	0,0193	1,4848	0,0125	KUG 0217-033
MESSGC 643 34.7166532+-3.0411896	0,7097	0,0118	0,2198	0,0188	0,4899	0,0070	MRK 1031
MESSGC 648 359.6455341+51.4397195	2,6928	0,0603	2,6834	0,1279	0,0094	0,0676	
MESSGC 649 359.6083161+51.3201186	2,7939	0,0685	1,0406	0,0377	1,7533	0,0308	
MESSGC 655 165.0811802+-18.3401651	2,9743	0,0459	1,8497	0,0663	1,1246	0,0204	GALEXASC J110019.48-182023.5
MESSGC 659 165.1518869+-18.2715615	2,2899	0,0405	2,1338	0,0883	0,1561	0,0478	GALEXASC J110036.46-181618.3
MESSGC 661 69.2904315+-61.9264707	3,2308	0,0856	2,8663	0,1252	0,3645	0,0396	GALEXASC J043709.31-615536.9
MESSGC 667 146.8433199+11.3053757	0,9870	0,0117	1,1227	0,0300	-0,1357	0,0183	SDSS J094722.40+111817.9
MESSGC 670 74.8685687+-14.8354139	2,3456	0,0468	1,4819	0,0464	0,8637	0,0004	
MESSGC 671 74.8666239+-14.8363186	2,3760	0,0465	1,4819	0,0464	0,8941	0,0001	
MESSGC 68 228.834759+-69.936419	3,1886	0,0598	2,2697	0,0855	0,9189	0,0257	
MESSGC 685 195.7503606+5.1602335	2,8251	0,0591	2,1953	0,0873	0,6298	0,0282	GALEXASC J130300.09+050938.5
MESSGC 686 195.6870799+5.326408	1,8527	0,0272	1,4198	0,0435	0,4329	0,0163	SDSS J130244.94+051934.7
MESSGC 691 216.0966503+25.7881984	1,3949	0,0182	0,6658	0,0179	0,7291	0,0003	KUG 1422+260
MESSGC 705 282.6654496+-7.8929864	2,2671	0,0415	2,4011	0,1510	-0,1340	0,1095	
MESSGC 712 198.4863879+-2.9743204	2,6961	0,0447	1,0518	0,0346	1,6443	0,0101	LCRS B131122.0-024233
MESSGC 724 198.6766156+-2.6947668	2,7544	0,0399	1,8883	0,0649	0,8661	0,0250	SDSS J131442.43-024138.1
MESSGC 726 198.5093627+-2.6201783	2,5443	0,0365	1,2559	0,0339	1,2884	0,0026	LCRS B131127.6-022117
MESSGC 729 198.5049134+-2.6215837	1,9241	0,0230	0,9672	0,0286	0,9569	0,0056	LCRS B131126.5-022123
MESSGC 743 356.6322989+3.4816672	2,3292	0,0353	1,4918	0,0335	0,8374	0,0018	
MESSGC 747 158.8469206+-39.649482	3,8023	0,0896	2,8860	0,1148	0,9163	0,0252	
MESSGC 756 158.8403419+-39.4452196	3,1052	0,0473	1,3022	0,0298	1,8030	0,0175	
MESSGC 760 159.4317306+-13.466394	3,3699	0,0803	2,0644	0,0644	1,3055	0,0159	
MESSGC 77 228.6177954+-70.129925	2,7455	0,0427	1,9425	0,0622	0,8030	0,0195	
MESSGC 801 272.0435448+-22.0868516	0,8597	0,0111	0,3209	0,0928	0,5388	0,0817	
MESSGC 802 272.0723432+-22.0890223	-3,7529	0,0002	-4,9877	0,0018	1,2348	0,0016	
MESSGC 804 272.0697697+-22.0910718	-2,9140	0,0005	-4,9877	0,0018	2,0737	0,0013	
MESSGC 810 272.1023231+-22.1417223	0,6827	0,0085	-3,4984	0,0055	4,1811	0,0030	
MESSGC 822 110.921101+-25.8826421	2,7607	0,0653	-0,6951	0,0140	3,4558	0,0513	
MESSGC 823 110.656553+-25.8495382	2,1464	0,0477	1,5104	0,0636	0,6360	0,0159	2MASX J07223760-2550567
MESSGC 831 111.0104122+-25.746945	-0,0524	0,0085	-3,2667	0,0025	3,2143	0,0060	
MESSGC 832 111.0110493+-25.7497282	0,1923	0,0091	-3,2667	0,0025	3,4590	0,0066	
MESSGC 84 229.0311894+-70.002398	1,5389	0,0119	0,9754	0,0279	0,5635	0,0160	2MASX J15160761-7000067
MESSGC 840 288.9608566+-7.0108936	1,8164	0,0253	0,9337	0,0357	0,8827	0,0104	GALEXASC J191550.60-070036.8
MESSGC 9 207.3375069+-28.5538624	2,4951	0,0312	2,1675	0,0808	0,3276	0,0496	2MASX J13492110-2833165
MESSGC 93 191.3881286+45.3181193	4,1094	0,1138	2,8683	0,0949	1,2411	0,0189	SDSS J124533.15+451903.6
MESSGC 96 191.3204518+45.3684724	3,9227	0,0789	1,6361	0,0418	2,2866	0,0371	GALEXASC J124517.26+452206.5

Table 7: Table of B, G, R and B-G, B-R, G-R measurements, including respective errors.

Identity	B [mag]	err	G [mag]	err	R [mag]	err	B-G [mag]	err	B-R [mag]	err	G-R [mag]	err	NED
MESSGC 133 31.1617533+64.7195425	2,6873	0,0373	2,4175	0,0357	2,0952	0,0595	0,2698	0,0016	0,5921	0,0222	0,3223	0,0238	
MESSGC 162 30.978771+64.9364142	3,7267	0,0782	2,1782	0,0251	2,6290	0,1086	1,5485	0,0531	1,0977	0,0304	-0,4508	0,0835	
MESSGC 164 31.7720999+64.9446652	3,7089	0,0728	2,0864	0,0328	1,1473	0,0342	1,6225	0,0400	2,5616	0,0386	0,9391	0,0014	2XMM J020705.1+645642
MESSGC 280 147.0344309+13.0473411	3,8626	0,0813	2,6130	0,0548	2,2712	0,0569	1,2496	0,0265	1,5914	0,0244	0,3418	0,0021	SDSS J094808.32+130248.9
MESSGC 284 147.0871447+13.2232555	2,6132	0,0367	2,5305	0,0510	1,5529	0,0399	0,0827	0,0143	1,0603	0,0032	0,9776	0,0111	2MASX J09482083+1313233
MESSGC 291 147.1658069+13.4170536	3,2861	0,0553	3,1735	0,0679	2,6149	0,0933	0,1126	0,0126	0,6712	0,0380	0,5586	0,0254	SDSSCGB 59955.02
MESSGC 299 146.9219827+13.523776	2,2131	0,0192	1,7458	0,0261	1,8781	0,0351	0,4673	0,0069	0,3350	0,0159	-0,1323	0,0090	2MASX J09474126+1331248
MESSGC 327 308.0056119+40.2503176	-4,7843	0,0025	-6,3587	0,0009	-4,2187	0,0048	1,5744	0,0016	-0,5656	0,0023	-2,1400	0,0039	
MESSGC 334 308.0600446+40.3275143	-6,5885	0,0004	-6,6690	0,0004	-6,6376	0,0013	0,0805	0,0000	0,0491	0,0009	-0,0314	0,0009	
MESSGC 364 307.7823962+40.2890732	-5,6530	0,0004	-6,0754	0,0005	-6,3240	0,0017	0,4224	0,0001	0,6710	0,0013	0,2486	0,0012	
MESSGC 367 307.7405604+40.2203276	2,3458	0,1385	-0,4736	0,0141	-2,8188	0,0109	2,8194	0,1244	5,1646	0,1276	2,3452	0,0032	
MESSGC 371 308.0240643+40.2513564	-4,2739	0,0021	-3,8815	0,0016	-3,8637	0,0049	-0,3924	0,0005	-0,4102	0,0028	-0,0178	0,0033	
MESSGC 372 307.9029186+40.2298935	-0,1432	0,0102	-0,8652	0,0076	-1,8883	0,0242	0,7220	0,0026	1,7451	0,0140	1,0231	0,0166	
MESSGC 378 307.9031985+40.2331192	-0,1127	0,0155	-0,8652	0,0076	-1,8883	0,0242	0,7525	0,0079	1,7756	0,0087	1,0231	0,0166	
MESSGC 379 307.7609315+40.2976267	-4,8713	0,0014	-5,6852	0,0012	-5,1589	0,0038	0,8139	0,0002	0,2876	0,0024	-0,5263	0,0026	
MESSGC 383 307.8052583+40.2447216	-0,6871	0,0118	-1,6493	0,0064	-3,6407	0,0070	0,9622	0,0054	2,9536	0,0048	1,9914	0,0006	
MESSGC 384 307.9952843+40.2633635	-4,1273	0,0014	-4,3090	0,0015	-3,3619	0,0096	0,1817	0,0001	-0,7654	0,0082	-0,9471	0,0081	
MESSGC 385 308.0287249+40.2590274	-3,8774	0,0020	-3,9827	0,0020	-3,6230	0,0059	0,1053	0,0000	-0,2544	0,0039	-0,3597	0,0039	
MESSGC 386 307.7776423+40.29163	-5,1046	0,0012	-6,0754	0,0005	-6,3240	0,0017	0,9708	0,0007	1,2194	0,0005	0,2486	0,0012	2MASX J20310684+4017313
MESSGC 388 308.0109564+40.2547181	-3,9579	0,0014	-6,3587	0,0009	-4,2187	0,0048	2,4008	0,0005	0,2608	0,0034	-2,1400	0,0039	
MESSGC 389 308.0032239+40.2636901	-3,9249	0,0016	-4,3090	0,0015	-3,3619	0,0096	0,3841	0,0001	-0,5630	0,0080	-0,9471	0,0081	
MESSGC 401 307.8055989+40.2662383	-1,5068	0,0065	-2,4983	0,0041	-4,2465	0,0043	0,9915	0,0024	2,7397	0,0022	1,7482	0,0002	
MESSGC 403 307.8106895+40.2685197	-1,8127	0,0050	-2,4983	0,0041	-4,2465	0,0043	0,6856	0,0009	2,4338	0,0007	1,7482	0,0002	
MESSGC 411 307.6207156+40.275236	1,1931	0,0509	0,1704	0,0267	-0,2943	0,0940	1,0227	0,0242	1,4874	0,0431	0,4647	0,0673	
MESSGC 415 307.7847745+40.2872049	-3,0966	0,0011	-6,0754	0,0005	-6,3240	0,0017	2,9788	0,0006	3,2274	0,0006	0,2486	0,0012	
MESSGC 416 308.0732929+40.3272613	-3,0617	0,0038	-3,7787	0,0016	-6,2088	0,0010	0,7170	0,0022	3,1471	0,0028	2,4301	0,0006	
MESSGC 420 307.7809756+40.2979774	-4,5940	0,0008	-4,2587	0,0009	-4,8013	0,0023	-0,3353	0,0001	0,2073	0,0015	0,5426	0,0014	
MESSGC 426 307.7519806+40.307563	-2,3669	0,0039	-2,7228	0,0037	-2,8794	0,0118	0,3559	0,0002	0,5125	0,0079	0,1566	0,0081	
MESSGC 427 307.9230084+40.2982493	-0,2751	0,0094	-1,3774	0,0068	-2,5641	0,0140	1,1023	0,0026	2,2890	0,0046	1,1867	0,0072	
MESSGC 436 307.6203135+40.309678	-0,5947	0,0144	-0,8547	0,0164	-0,3897	0,1182	0,2600	0,0020	-0,2050	0,1038	-0,4650	0,1018	
MESSGC 437 308.0815769+40.3179036	-1,9299	0,0067	-4,1328	0,0026	-5,7090	0,0014	2,2029	0,0041	3,7791	0,0053	1,5762	0,0012	
MESSGC 441 307.9924184+40.3101539	-1,4064	0,0034	-2,0132	0,0031	-3,6145	0,0058	0,6068	0,0003	2,2081	0,0024	1,6013	0,0027	
MESSGC 442 307.9899371+40.3090182	1,1258	0,0283	-2,0132	0,0031	-3,6145	0,0058	3,1390	0,0252	4,7403	0,0225	1,6013	0,0027	
MESSGC 443 308.0109319+40.3107884	-0,6520	0,0078	-1,9372	0,0044	-4,0037	0,0046	1,2852	0,0034	3,3517	0,0032	2,0665	0,0002	
MESSGC 446 308.081836+40.3229699	-2,6041	0,0028	-4,1328	0,0026	-5,7090	0,0014	1,5287	0,0002	3,1049	0,0014	1,5762	0,0012	
MESSGC 447 307.6803825+40.3218673	-1,2316	0,0060	-2,2150	0,0032	-3,3824	0,0074	0,9834	0,0028	2,1508	0,0014	1,1674	0,0042	
MESSGC 448 307.6745482+40.3223343	-0,8534	0,0081	-1,9516	0,0039	-3,0408	0,0077	1,0982	0,0042	2,1874	0,0004	1,0892	0,0038	
MESSGC 451 308.091685+40.3273463	-3,0067	0,0013	-5,4608	0,0014	-6,2505	0,0008	2,4541	0,0001	3,2438	0,0005	0,7897	0,0006	
MESSGC 452 308.0869045+40.3269316	-1,9704	0,0027	-5,4608	0,0014	-6,2505	0,0008	3,4904	0,0013	4,2801	0,0019	0,7897	0,0006	
MESSGC 453 308.1194402+40.3282427	-2,4958	0,0018	-2,6875	0,0020	-7,5292	0,0013	0,1917	0,0002	5,0334	0,0005	4,8417	0,0007	
MESSGC 454 308.0863582+40.3305947	-2,2021	0,0024	-5,4608	0,0014	-6,2505	0,0008	3,2587	0,0010	4,0484	0,0016	0,7897	0,0006	
MESSGC 455 308.0923353+40.3381674	-4,1845	0,0006	-5,2110	0,0003	-6,1681	0,0008	1,0265	0,0003	1,9836	0,0002	0,9571	0,0005	
MESSGC 456 308.0883854+40.333417	-2,3901	0,0019	-5,4608	0,0014	-6,2505	0,0008	3,0707	0,0005	3,8604	0,0011	0,7897	0,0006	
MESSGC 458 308.0899995+40.3374475	-3,7719	0,0006	-5,2110	0,0003	-6,1681	0,0008	1,4391	0,0003	2,3962	0,0002	0,9571	0,0005	
MESSGC 461 308.0883475+40.3406124	-2,1193	0,0021	-5,2110	0,0003	-6,1681	0,0008	3,0917	0,0018	4,0488	0,0013	0,9571	0,0005	
MESSGC 466 308.0122814+40.347913	-1,4005	0,0056	-4,1231	0,0020	-4,7157	0,0039	2,7226	0,0036	3,3152	0,0017	0,5926	0,0019	
MESSGC 469 308.0073595+40.3569167	-1,2079	0,0064	-2,7915	0,0040	-4,3770	0,0048	1,5836	0,0024	3,1691	0,0016	1,5855	0,0008	
MESSGC 471 307.6883854+40.3640736	-1,8593	0,0037	-2,3795	0,0034	-3,3132	0,0086	0,5202	0,0003	1,4539	0,0049	0,9337	0,0052	2MASX J20304522+4021501
MESSGC 475 308.013517+40.3711546	1,8988	0,0834	-1,8310	0,0080	-4,3767	0,0050	3,7298	0,0754	6,2755	0,0784	2,5457	0,0030	
MESSGC 477 307.8322671+40.3746892	1,8411	0,0656	0,2539	0,0318	-2,7219	0,0132	1,5872	0,0338	4,5630	0,0524	2,9758	0,0186	
MESSGC 479 307.9511297+40.4518966	0,1437	0,0200	-0,8973	0,0107	-2,1392	0,0203	1,0410	0,0093	2,2829	0,0003	1,2419	0,0096	
MESSGC 480 308.1224032+40.424648	-2,5513	0,0030	-4,0522	0,0028	-3,8894	0,0066	1,5009	0,0002	1,3381	0,0036	-0,1628	0,0038	

Acknowledgements

I would like to convey sincere gratitude to Prof. W. W. Zeilinger for constant support, constructive feedback and careful reading of this thesis, my dear colleagues at the Institute for Astrophysics at the University of Vienna, my family and friends, and all people who supported me on my journey to becoming a scientist.

This research has made use of the NASA/IPAC Extragalactic Database (NED), which is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

This publication makes use of data products from the Wide-field Infrared Survey Explorer, which is a joint project of the University of California, Los Angeles, and the Jet Propulsion Laboratory/California Institute of Technology, funded by the National Aeronautics and Space Administration.

References

Ahn, C. P., Alexandroff, R., Allende Prieto, C., et al. 2012, ApJS, 203, 21

Benítez, N., Ford, H., Bouwens, R., et al. 2004, ApJS, 150, 1

Bertin, E. & Arnouts, S. 1996, A&AS, 117, 393

Calzetti, D., Wu, S.-Y., Hong, S., et al. 2010, ApJ, 714, 1256

Cayón, L., Sanz, J. L., Martínez-González, E., et al. 2001, MNRAS, 326, 1243

Chabrier, G. 2003, PASP, 115, 763

Chary, R. & Elbaz, D. 2001, ApJ, 556, 562

Coleman, G. D., Wu, C.-C., & Weedman, D. W. 1980, ApJS, 43, 393

Cutri, R. M. & et al. 2014, VizieR Online Data Catalog, 2328, 0

Domínguez Sánchez, H., Bongiovanni, A., Lara-López, M. A., et al. 2014, MNRAS, 441, 2

Dorfi, E. 2013, Astrophysik 1 Skript, University Lecture

Dorfi, E. 2014, Astrophysik 2 Skript, University Lecture

Galametz, M., Kennicutt, R. C., Calzetti, D., et al. 2013, MNRAS, 431, 1956

Groenewegen, M. A. T., Waelkens, C., Barlow, M. J., et al. 2011, A&A, 526, A162

Herschel, W. 1800, Philosophical Transactions of the Royal Society of London, 90, 284

Ivison, R. J., Magnelli, B., Ibar, E., et al. 2010, A&A, 31, 1

Jeans, J. H. 1902, Royal Society of London Philosophical Transactions Series A, 199, 1

Kessler, M. F., Steinz, J. A., Anderegg, M. E., et al. 1996, A&A, 315, L27

Kron, R. G. 1980, ApJS, 43, 305

Kroupa, P. 2001, MNRAS, 322, 231

Mainzer, A. K., Young, E. T., Huff, L. W., & Swanson, D. 2003, Proc. SPIE, 4850, 122

Mazzarella, J. M. & NED Team. 2007, Astronomical Society of the Pacific Conference Series, 376, 153

McKee, C. F. & Ostriker, J. P. 1977, ApJ, 218, 148

Moustakas, J., Kennicutt, Jr., R. C., & Tremonti, C. A. 2006, ApJ, 642, 775

Neugebauer, G., Habing, H. J., van Duinen, R., et al. 1984, ApJL, 278, L1

Petrosian, V. 1976, ApJ, 209, L1

Pilbratt, G. L., Riedinger, J. R., Passvogel, T., et al. 2010, A&A, 518, L1

Poglitsch, A., Waelkens, C., Geis, N., et al. 2010, A&A, 518, L2

Roussel, H. 2013, PASP, 125, 1126

Salpeter, E. E. 1955, ApJ, 121, 161

Sedov, L. I. 1959, Similarity and Dimensional Methods in Mechanics, New York: Academic Press, 1959

Skrutskie, M. F., Cutri, R. M., Stiening, R., et al. 2006, AJ, 131, 1163

Swinyard, B. M., Ade, P., Baluteau, J., et al. 2010, A&A, 4, 2

Wells, D. C., Greisen, E. W., & Harten, R. H. 1981, A&APS, 44, 363

Wright, E. L., Eisenhardt, P. R. M., Mainzer, A. K., et al. 2010, AJ, 140, 1868

CURRICULUM VITAE

Personal Data

Name:	Michael
Last Name:	Mach
Date of Birth:	01.01.1983
Place of Birth:	Vienna
Citizenship :	AT

Education

1989 – 1993	Elementary School Alterlaa, 1230 Vienna
1993 – 2002	Secondary School Bundesgymnasium und Bundesrealgymnasium 23, 1230 Vienna Anton Baumgartnerstr. 123 (Matura)
2013	Bakk.rer.nat. of Astronomy at University of Vienna *Thesis Title: Observation of the Local Bubble* *Supervisor: Prof. Breitschwerdt / Prof. Zeilinger*
11/2014	projected acquisition of MSc. Of Astronomy at University of Vienna *Thesis Title: Properties of Extragalactic Objects with Herschel Data* *Supervisor: Prof. Zeilinger / Prof. Kerschbaum*

Languages

German (mothertongue)
English (fluent)
Spanish (working knowledge)
Italian (working knowledge)

Soft Skills

Advanced Knowledge of:
Python, Fortran, SExtractor, Aladin, Skycat, ds9,
Unix, Linux, IRAF, R, gnuplot, Scisoft-Package, Html,
Javascript, Office/OpenOffice , LaTeX

Working Knowledge in:
C, C++, HIPE, Topcat, Css

Other Education

B- Drivers License

Scientific Employment

WS 2013	Tutor for Mathematics 1A at the Department for Earth Sciences at the University of Vienna (Dorfi / Reimers)
SS 2014	Tutor for Mathematics 1B + own Tutorium at the Department for Earth Sciences at the University of Vienna (Dorfi / Reimers)
	Tutor for Mathematics II at the Department for Earth Sciences at the University of Vienna (Dorfi / Reimers)
	Tutor for "beobachtungsorientiertes Praktikum" at the department for Astrophysics at the University of Vienna (Zeilinger/Alves/Forbich/Hacar Gonzales)