

OPTICAL-FAINT COMPACT RADIO SOURCES

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Physics

by
Jacob Brown
Dr. Haojing Yan, Thesis Supervisor
MAY 2017

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

OPTICAL-FAINT COMPACT
RADIO SOURCES

presented by Jacob Brown,
a candidate for the degree of Doctor of Philosophy in Physics and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Haojing Yan

Dr. Angela Speck

Dr. Aigen Li

Dr. Sergei Kopeikin

Dr. Adam Helfer

ACKNOWLEDGMENTS

I would like to thank Dr. Haojing Yan for his guidance and support throughout this project. I want to thank Zhiyuan Ma, James Runge, Marat Musin, and Chengxiaoji Ling for their assistance in the OSMOS observations as well as their helpful insight and suggestions for this project. I would also like to thank Bradley Mills for his insight and suggestions into this project. Finally, I would like to thank the members of my committee for their input into insightful paths for this research.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
LIST OF TABLES	viii
LIST OF FIGURES	x
ABSTRACT	xiv
CHAPTER	
1 Introduction	1
2 Data	4
2.1 Radio Fundamental Catalog	4
2.2 Sloan Digital Sky Survey	5
2.3 OFCORS Selection	6
2.4 Wide-Field Infrared Survey Explorer	14
2.5 UKIRT Deep Sky Survey	25
2.6 Visible and Infrared Survey Telescope for Astronomy	26
2.7 Canadian France Hawaii Telescope	27
2.8 Unified Radio Catalog	28
2.9 Pan-STARRS	29
3 Observations	31
3.1 Is SDSS deep enough?	31
3.2 OSMOS description	32

3.3	OSMOS Observations	34
4	Data Reduction	36
4.1	OSMOS Data Reduction	36
4.1.1	Bias and Overscan	36
4.1.2	Flat Fielding	37
4.1.3	Illumination Correction	41
4.1.4	Fringe Correction	42
4.1.5	Scamp	44
4.1.6	SWarp	46
5	Photometry	48
5.1	Modifying The Weight Image	48
5.2	Initial Photometry	49
5.3	Zeropoint Calculation	50
5.4	Photometry	52
5.5	Source Identification	53
5.6	Upper Limit	53
6	SED fitting	56
6.1	Creating the catalogs	56
6.1.1	OSMOS+UKIDSS+WISE	56
6.1.2	CFHT+UKIDSS+WISE	57
6.1.3	Pan-STARRS+UKIDSS+WISE	57
6.2	LePhare	58

6.3	Output	58
6.3.1	OSMOS+UKIDSS+WISE	58
6.3.2	CFHT+UKIDSS+WISE	63
6.3.3	Pan-STARRS+UKIDSS+WISE	64
6.4	Plots	64
7	Infrared relations and power laws	69
7.1	K-z relation	69
7.2	W1-z relation	74
7.3	IR Power Law	78
7.3.1	Motivation	78
7.3.2	IR power law calculation	78
7.3.3	IR power law results	81
8	Radio power law	85
8.1	Motivation	85
8.2	Building master catalog	85
8.3	Calculating the Radio Power Law	86
8.4	Radio Power Law of the RFC	87
8.5	Comparision of the OFCORS the entire RFC	87
9	Conclusions	97
APPENDIX		
A	IRAF parameter files	99
A.1	Flatcombine parameter file	99

A.2	ccdproc parameter file for flat-field correction	100
A.3	Illumination stack with combine	101
A.4	boxcar	102
A.5	ccdproc for illumination correction	102
A.6	precess	103
B	Weight Script	105
B.1	jobgetweights_master.cl	105
B.2	jobcreator_getmskregions.sh	106
B.3	jobcreator_makepl.sh	106
B.4	jobcreator_getpl2fits.sh	106
B.5	jobcreator_imcalcstep1.sh	106
B.6	jobcreator_imcalcstep2.sh	107
B.7	jobcreator_makesegmentation.sh	107
B.8	jobcreator_imcalcsegsetto1.sh	107
B.9	jobcreator_imcalccombine.sh	108
C	Scamp and Swarp	109
C.1	Sextractor for Scamp	109
C.1.1	Sextractor parameter file	112
C.2	Scamp configuration file	113
C.3	Swarp	116
D	Photometry	120
D.1	Weight image modification	120

D.1.1	weightfix.dat	120
D.2	Source photometry	120
D.2.1	SExtractor on stacked image	121
D.2.2	SExtractor parameter file	124
D.3	Zero Point Calculation	124
D.3.1	Zero point C++ code	125
D.3.2	Matrix.h	133
D.4	Upperlimit calculation	136
D.4.1	xyrandomnumber.cpp	136
D.4.2	phot	138
D.5	Upperlimitclipping algorithm c++	140
E	Lephare	146
F	IR Power Law C++ code	152
G	Radio Power Law C++ code	157
	BIBLIOGRAPHY	166
	VITA	169

LIST OF TABLES

Table	Page
2.1 RFC bands in catalog	5
2.2 SDSS bands and central wavelengths	6
2.3 Coordinates of the OFCORS sources	14
2.4 WISE Bands and Wavelengths	15
2.5 Data for the OFCORS in WISE	24
2.6 UKIDSS Bands and Wavelengths	25
2.7 UKIDSS data for the OFCORS	26
2.8 VISTA Bands and Wavelengths	27
2.9 VISTA data for the OFCORS	27
2.10 CFHT data for the OFCORS	28
2.11 PanSTARRS Filters and Wavelengths	30
2.12 PanSTARRS data for the OFCORS	30
3.1 Observation dates related to run number.	34
3.2 columns are as follows: a) the source name b) the RA coordinate in J2000, c) the DEC coordinate in J2000, d) the observation run number, e) the filters used, f) the exposure time in seconds in order of filters used	35

4.1	Flat-fields and which ones applied to which night. Column 1 is the observation date, Column 2 is the filter used, and Column 3 is which nights flat-fields were used to build the twilight flat	39
5.1	Sources and filters that OFCORS were detected in OSMOS	53
5.2	Upperlimit for the OFCORS observed by OSMOS	55
7.1	value of α for each source	83
8.1	Spectral index for the OFCORS sources	96

LIST OF FIGURES

Figure	Page	
2.1	Examples of OFCORS selection from visible inspection. a) Fell outside of SDSS footprint; b) Inside the SDSS footprint but no data; c) RFC source visible in SDSS; d) RFC source not visible in SDSS	7
2.2	Fits images of J0037+0808 in all five bands of SDSS. From left to right, the filters are urigz. Green circle represents where the radio source should be located based off the coordinates from RFC.	7
3.1	OSMOS instrument	33
4.1	Example of bias correction. a) is the raw image while b) is after the bias correction has been applied	37
4.2	Example of a flat-field correction	40
4.3	Example of flat correction. a) is the bias corrected image while b) is after the flat field correction has been applied	41
4.4	Example of the illumination correction	43
4.5	Example of illum correction. a) the flat-field corrected image b) after the illumination correction has been applied	44

4.6	Example of the output from SWarp. a) the stacked science image. b) the stacked weight image.	47
5.1	magnitude difference between OSMOS and SDSS vs OSMOS magnitude for J2117+5431 r band. a) all of the sources matched between SDSS and OSMOS. b) sources left after cuts	50
5.2	Final zeropoint calculation for J2114++5431 r band. The points are what was left over after the zeropoint calculation. The line represents the final zero point value.	52
6.1	SED for J0210-0737 using OSMOS for optical	59
6.2	SED for J0253+0024 using OSMOS for optical	59
6.3	SED for J0305+0523 using OSMOS for optical	60
6.4	SED for J0418+5457 using OSMOS for optical	60
6.5	SED for J0805+2106 using OSMOS for optical	60
6.6	SED for J0904+2729 using OSMOS for optical	61
6.7	SED for J1110+6028 using OSMOS for optical	61
6.8	SED for J1434+4236 using OSMOS for optical	61
6.9	SED for J1439+2114 using OSMOS for optical	62
6.10	SED for J2210+2013 using OSMOS for optical	62
6.11	SED for J0210-0737 using CFHT for optical	63
6.12	SED for J0253+0024 using CFHT for optical	63
6.13	SED for J2210+2013 using CHFT for optical	64
6.14	SED for J0305+0523 using Pan-STARRS for optical	65
6.15	SED for J1152+3327 using Pan-STARRS for optical	65

6.16	SED for J1735+5049 using Pan-STARRS for optical	65
6.17	SED for J1746+6421 using Pan-STARRS for optical	66
6.18	SED for J2210+2013 using Pan-STARRS for optical	66
6.19	SED for J2219+2141 using Pan-STARRS for optical	66
6.20	Redshift distribution from LePhare	67
6.21	Stellar mass distribution from LePhare	68
7.1	K-z relation determined by Willott 2003.	70
7.2	K-z relation comparing Willott 2003 to RFC sources matched with 2MASS. Red points are the Willott 2003 data and the blue dots represent the RFC matched with 2MASS sources	72
7.3	K-z relation for the RFC sources matched with 2MASS seperated by the SDSS class parameter. Green dots represent when class is equal to galaxy and the purple dots represent when the class parameter is equal to QSO.	73
7.4	K-z relation comparing Willott 2003 to RFC sources matched with UKIRT. Red points are the Willott 2003 data and the blue dots represent the RFC matched with 2MASS sources	75
7.5	W1-z relation by cross matching Willott03 with WISE	76
7.6	W1-z relation comparing Willott03 to RFC sources matched with W1. Red points are the Willott03 data and the blue dots represent the RFC matched with W1 sources. The dashed line represents the range of W1 detections of the OFCORS sources.	77
7.7	IR power law for J0158-1413 at redshift of 0.05	82
7.8	Histogram of α values from the IR calculation	84

8.1	Example of the radio power law fit for source J0001+6051	88
8.2	Distribution of the spectral index for the entire RFC	89
8.3	Comparison of the RFC and the OFCORS subset values of the spectral index. The RFC is in red and the OFCORS is in blue.	90

Optical-Faint Compact Radio Sources

Jacob Brown

Dr. Haojing Yan, Dissertation Supervisor

ABSTRACT

This work investigates a population of bright, compact radio sources that are very faint in optical, which has never been studied before in a comprehensive manner. The sample consists of 135 radio sources selected from the Radio Fundamental Catalog (RFC). These sources have accurate VLBI positions, have ultra-compact radio morphologies (<10 mas), are strong in radio (median flux density ~ 100 mJy), and yet are completely invisible in the SDSS DR10 data. While it is natural to guess that the OFCORS are powerful AGN, their exact nature is unclear. This multiwavelength study was performed by obtaining deeper optical imaging with OSMOS, CFHT, and PanSTARRS, perform SED fitting to determine the sources properties, explore the K-z and W1-z relations, and determine the power law in the infrared and radio regime.

Chapter 1

Introduction

Celestial objects emit energy in all wavelengths of the electromagnetic spectrum. Up until the early 20th century, astronomers could only observe the night sky in optical wavelengths. Then, in 1933, Karl Jansky opened up a new spectral window by detecting the center of the Milky Way in radio wavelengths. [1] After WWII, radio astronomy became the main stream. Surveys at the radio wavelengths have been carried out routinely since 1950s, and currently radio observation is an essential part of the multi-wavelength investigation of our universe.

In most cases, however, radio observation *alone* does not allow us to determine the nature of a detected radio source. Historically, the nature of any radio populations were revealed only after their counterparts at other wavelengths were identified. Optical counterpart identification is often particularly important because optical spectroscopy is usually the most useful means to understand the physics of an object in question. From spectroscopy, one can calculate the redshift, and then using Hubble's Law determine the distance. [2] Distances become important for determining the absolute radio luminosity from the radio flux density, and how large the source is from the angular size. Optical observations may

serve as a key component as the cause of radio emission in these sources.

In this sense, it is bothersome to realize that a large fraction of the radio sources that have been detected lack optical counterpart identification. For example, the Faint Images in the Radio Sky at Twenty-cm Survey (FIRST) [3] is a Radio survey of the sky with a coverage of 10000 square degrees. This survey is capable of detecting radio sources down to a threshold of 1 mJy. As of May 1st 2012, FIRST had a total of 945464 sources cataloged. Comparing this to the Sloan Digital Sky Survey (SDSS) [4], 889572 of the FIRST sources fall inside of the SDSS footprint. However, when matching the FIRST catalog to the SDSS catalog with a matching radius of 3", 563565 radio sources or 63.4% do not have an optical counterpart. By looking at the brightest radio sources, with $S_{1.4GHz} > 100mJy$, there are 8140 FIRST selected SDSS-invisible sources. These may only be $\sim 1.4\%$ of the total FIRST selected SDSS-invisible sources, but their impact could be significant if these sources are star-forming or AGNs. If they turn out to be star-forming, then we miss out on a significant portion of extreme star-forming galaxies. Likewise, if they are powered by AGN, then we could miss a significant amount of the most luminous quasars. If we use quasar 3C 273 as an example and redshift it to $z \sim 1.4$, then it would be roughly a 100 mJy source in radio, but will have an apparent magnitude in SDSS r-band of 19.04. The nominal $5-\sigma$ sensitivity limit for SDSS r band is 22.31 [5], so 3C 273 would be visible at a $z \sim 1.4$. Therefore, it is important to determine what these Radio-loud, optically faint sources are and why they may be optically faint.

To determine why these radio bright optically faint sources appear optically faint, first one needs to be able to identify the optical counterpart to each radio source. An issue arises when finding the optical counterparts for radio sources: spatial resolution. For optical telescopes, astronomers can reach a spatial resolution of $< 1''$ however, radio surveys

have had a spatial resolution from 1'' to 5'' [4]. This makes it difficult to pinpoint which optical source is the counterpart of radio source. We can get around this issue however, by using interferometry, which gives us a large baseline, reducing the spatial resolution. For example, the VLBI can achieve an accuracy in the milliarcseconds range. This leads us to the Radio Fundamental Catalog, which will be talked about in Chapter 2.

In this thesis, I will discuss the process employed to understand the nature of Optical-Faint Compact Radio Sources (OFCORS). Chapter 2 will discuss the various surveys that were used. Chapter 3 will discuss the independent observations of our sources using OSMOS. Chapter 4 will discuss the data reduction of the OSMOS images. Chapter 5 will discuss the photometry process used on the OSMOS images. Chapter 6 will discuss fitting an SED to the data. Chapter 7 will discuss various IR properties. Chapter 8 will discuss the radio power law. Finally Chapter 9 will be the overall conclusions of this project.

Chapter 2

Data

2.1 Radio Fundamental Catalog

The Radio Fundamental Catalog (RFC) is a catalog constructed from radio sources detected using Very Long Baseline Interferometry (VLBI). The observations were performed under many different absolute astrometry and geodesy programs over the past 20 plus years. [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] Due to this, the catalog is continuously updated quarterly. The current catalog, version 2016d,¹ contains 12651 sources consisting of detections in at least one of its five bands, listed in Table 2.1. The RFC is unique because all of the sources are compact, with sizes <10 arcseconds and their positions have been measured accurately to the milli-arcsecond level, in which 97% have an astrometric accuracy better than 12 milli-arcseconds. The high accuracy is ideal for overcoming the spatial resolution issue that affects optical counterpart identification. 65% of these sources are very bright and have a flux density greater than 100 mJy. Many of them are so bright, radio astronomers use them as calibrators for VLBI observations.

¹<http://astrogeo.org/rfc/>

Band	Frequency
S	2.3 GHz
C	6 GHz
X	8.4 GHz
K	22 GHz
U	50 GHz

Table 2.1: RFC bands in catalog

2.2 Sloan Digital Sky Survey

An optical survey is needed to search for optically faint radio sources. For this purpose, the Sloan Digital Sky Survey (SDSS) was chosen because it is one of the largest optical surveys. The SDSS [5] is a multiwavelength imaging and spectroscopic survey using a dedicated 2.5 m wide-field telescope located at Apache Point Observatory in New Mexico. The telescope performs observations by drift scanning, tracking along the sky in a great circle and continuously collects data from strips of the sky. The CCD then detects stars in the focal plane of the telescope as they drift along the CCD, while the charge is electronically shifted along the detector at the same rate. This is different from other observation methods, in which telescopes follow a point in the sky so stars will stay in the same location on the CCD. The drift scanning method allows for higher astrometry accuracy over a wide field and minimizes the overhead for observations.

In imaging mode, images of the sky are taken in 5 filters, listed in Table 2.2 covering 14555 square degrees, or 25%, of the sky. The 5 sigma detection limit ranges from 20.5 in the z band to 22.2 in the r band. The observations have an angular resolution of $\sim 1.2''$ and an astrometric accuracy of $\sim 0.1''$.

SDSS selects sources for spectroscopic follow up from the photometric data. SDSS performed spectroscopy by observing several sources at once using multiple fibers, in which

Band	Wavelength
u	354 nm
g	475 nm
r	622 nm
i	763 nm
z	905 nm

Table 2.2: SDSS bands and central wavelengths

each fiber was used for one source. This allows for SDSS to use the spectrograph to observe up to 1000 sources at once.

The SDSS catalog data release 10 (DR10) [22] was used to search for optical faint compact radio sources. This data release contains the photometric information for about 500 million sources and spectra for over 3 million sources.

2.3 OFCORS Selection

The Radio Fundamental Catalog (RFC) version 2012b was used to select the OFCORS by cross-matching with SDSS DR10 using a matching radius of 1" to grab the jpeg files of the area of the source. Then, to determine if the source was an OFCORS, the SDSS jpeg images were visually inspected and classified into four categories: fell outside of the SDSS footprint, fell inside the SDSS footprint but had no photometric data, fell in the SDSS footprint and had an optical counterpart, and fell in the SDSS footprint and had no obvious optical counterpart (Figure 2.1). The sources that fell in the SDSS footprint but had no obvious optical counterpart were then inspected using the SDSS fits images in all five bands (2.2) to confirm that there was no optical counterpart.

After visual inspection, 135 sources were classified as OFCORS. The coordinates for the OFCORS are listed in Table 2.3. Their median radio-to-optical flux density (S_r/S_o) can

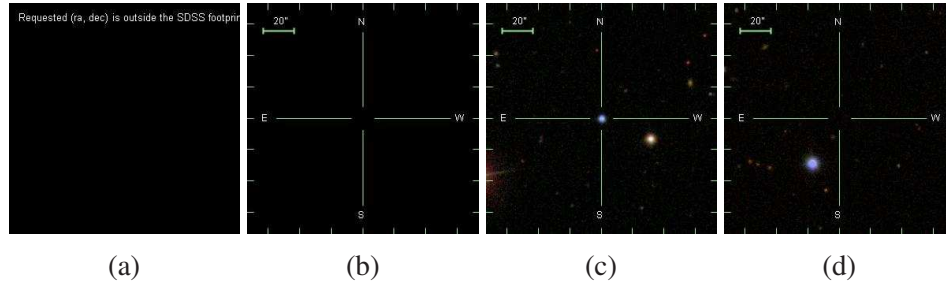


Figure 2.1: Examples of OFCORS selection from visible inspection. a) Fell outside of SDSS footprint; b) Inside the SDSS footprint but no data; c) RFC source visible in SDSS; d) RFC source not visible in SDSS

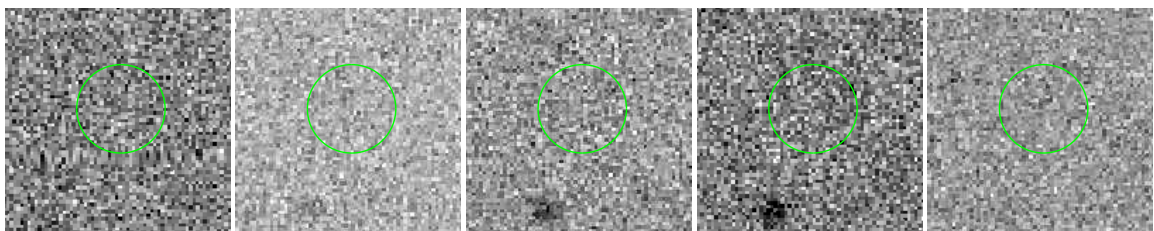


Figure 2.2: Fits images of J0037+0808 in all five bands of SDSS. From left to right, the filters are urigz. Green circle represents where the radio source should be located based off the coordinates from RFC.

be estimated to be $\geq 4.4 \times 10^4$. In comparison, radio-loud quasars have a median S_r/S_o that is a magnitude lower. The difference in the ratios hints that the OFCORS have distinct, intrinsically different properties than other radio sources.

Table 2.3

Name	RA	DEC
J0037+0808	00:37:32.197178	+08:08:13.05782
J0155-1443	01:55:33.242617	-14:43:17.43883
J0158-1413	01:58:43.719623	-14:13:07.11866
J0200-1542	02:00:51.148836	-15:42:36.86440
J0210-0737	02:10:16.528895	-07:37:20.69396
J0253+0024	02:53:33.663262	+00:24:31.55365
J0254-0225	02:54:19.086736	-02:25:23.56125
J0305+0523	03:05:48.191585	+05:23:31.52706
J0311+0508	03:11:47.993846	+05:08:02.72176
J0330-0551	03:30:34.508387	-05:51:22.66244
J0418+5457	04:18:19.340226	+54:57:15.33481
J0422+0827	04:22:07.078609	+08:27:07.09868
J0425+0833	04:25:38.649316	+08:33:32.46800
J0425+1755	04:25:24.475441	+17:55:24.98018
J0501+1356	05:01:45.270830	+13:56:07.22031
J0543+3023	05:43:54.237445	+30:23:39.67698
J0559+2353	05:59:32.033135	+23:53:53.92668
J0636+0801	06:36:39.072894	+08:01:27.90540
J0720-1308	07:20:22.945402	-13:08:29.21554

Continuation of Table 2.3

Name	RA	DEC
J0724-1505	07:24:03.056551	-15:05:17.26750
J0726+3912	07:26:04.737235	+39:12:23.32129
J0754+5324	07:54:15.217424	+53:24:56.44291
J0805+2106	08:05:38.534233	+21:06:51.59251
J0811+4308	08:11:37.363989	+43:08:29.43094
J0819+3823	08:19:00.956171	+38:23:59.81015
J0857+2805	08:57:35.904780	+28:05:38.57086
J0904+2729	09:04:53.760810	+27:29:53.93055
J0911+3033	09:11:14.738625	+30:33:51.52746
J0917+6530	09:17:55.568105	+65:30:15.12741
J0923+3849	09:23:14.452935	+38:49:39.91002
J0945+2729	09:45:15.626641	+27:29:11.34058
J0947+5907	09:47:04.864933	+59:07:41.47532
J0952+4941	09:52:00.665686	+49:41:25.95749
J1002+5701	10:02:41.666385	+57:01:11.48337
J1006+3050	10:06:46.537493	+30:50:28.38148
J1006+4836	10:06:39.577954	+48:36:31.20176
J1009+0622	10:09:49.808081	+06:22:00.98577
J1013+3043	10:13:07.038726	+30:43:33.82278
J1025+8144	10:25:57.122525	+81:44:51.85447
J1041+4900	10:41:29.908670	+49:00:01.64745
J1042+2949	10:42:36.511766	+29:49:45.14793

Continuation of Table 2.3

Name	RA	DEC
J1048+0141	10:48:22.867957	+01:41:48.11540
J1058+4248	10:58:00.753089	+42:48:44.15548
J1102+7905	11:02:08.953236	+79:05:51.49583
J1103+2203	11:03:23.067640	+22:03:37.72109
J1104+5752	11:04:12.957968	+57:52:12.87331
J1109+3744	11:09:28.853835	+37:44:31.10495
J1110+6028	11:10:13.088075	+60:28:42.56613
J1120+3257	11:20:35.229207	+32:57:45.83584
J1124+3214	11:24:03.013892	+32:14:14.07724
J1131+3013	11:31:26.829618	+30:13:34.38948
J1135+4258	11:35:55.989025	+42:58:44.65673
J1137+4741	11:37:34.981484	+47:41:09.75463
J1143+1834	11:43:26.069654	+18:34:38.35996
J1152+3327	11:52:03.268962	+33:27:04.82946
J1204+3023	12:04:58.017129	+30:23:25.46089
J1208+5413	12:08:27.499673	+54:13:19.51766
J1220+6327	12:20:46.846919	+63:27:29.01125
J1237+2034	12:37:56.590988	+20:34:18.85045
J1239+3705	12:39:36.366706	+37:05:06.23754
J1241+5458	12:41:27.703714	+54:58:19.05134
J1251+2102	12:51:27.702740	+21:02:53.69053
J1254+1843	12:54:34.020459	+18:43:32.30371

Continuation of Table 2.3

Name	RA	DEC
J1256+2220	12:56:42.841057	+22:20:24.51493
J1304+3523	13:04:34.495017	+35:23:33.53091
J1304+5056	13:04:48.014367	+50:56:23.21609
J1311+1658	13:11:23.819959	+16:58:44.19483
J1314+5306	13:14:43.830703	+53:06:27.72996
J1320+5036	13:20:42.208740	+50:36:07.79584
J1322+4303	13:22:21.202947	+43:03:41.22909
J1326+4434	13:26:36.905964	+44:34:59.14282
J1340-0335	13:40:13.304445	-03:35:20.80178
J1344+1409	13:44:23.744558	+14:09:14.88911
J1350+6428	13:50:55.699941	+64:28:56.78275
J1412+1334	14:12:36.372664	+13:34:38.15309
J1412+3200	14:12:24.981893	+32:00:54.32143
J1416+3444	14:16:04.186254	+34:44:36.42736
J1434+4236	14:34:27.863776	+42:36:20.07968
J1439+2114	14:39:08.902366	+21:14:50.82056
J1442+3042	14:42:41.533343	+30:42:32.92125
J1448+5326	14:48:59.174127	+53:26:09.28711
J1457+0938	14:57:52.531605	+09:38:16.55006
J1457+2435	14:57:43.425371	+24:35:07.71933
J1500+0839	15:00:34.003909	+08:39:41.80496
J1504+3249	15:04:07.525397	+32:49:21.18192

Continuation of Table 2.3

Name	RA	DEC
J1514+3046	15:14:34.151142	+30:46:03.92911
J1532+2919	15:32:20.882832	+29:19:31.60518
J1538+0019	15:38:15.953094	+00:19:05.32330
J1556+1825	15:56:54.816381	+18:25:13.57691
J1559+1624	15:59:25.067823	+16:24:40.88004
J1602+3326	16:02:07.263461	+33:26:53.07221
J1603+2126	16:03:32.450584	+21:26:51.82191
J1604+4746	16:04:27.865330	+47:46:34.94647
J1605+2214	16:05:36.438615	+22:14:12.48927
J1605+5931	16:05:50.205383	+59:31:43.34517
J1624+2748	16:24:35.717288	+27:48:57.69967
J1630+2131	16:30:11.230580	+21:31:34.30818
J1632+3547	16:32:31.256768	+35:47:37.74223
J1648+4233	16:48:31.507739	+42:33:22.39380
J1700+3830	17:00:19.967237	+38:30:34.13945
J1718+2948	17:18:27.177877	+29:48:00.28820
J1722+6144	17:22:40.742693	+61:44:53.80434
J1735+5049	17:35:49.005172	+50:49:11.56583
J1741+4751	17:41:34.821981	+47:51:32.53644
J1746+6421	17:46:06.678150	+64:21:49.66055
J1846+2036	18:46:04.349500	+20:36:10.35010
J1909+7813	19:09:18.955813	+78:13:29.88190

Continuation of Table 2.3

Name	RA	DEC
J1922+1504	19:22:33.272307	+15:04:47.53382
J1922+1530	19:22:34.699382	+15:30:10.03171
J1924+1558	19:24:02.999165	+15:58:55.65633
J1949+7654	19:49:58.567977	+76:54:13.72357
J1955+3233	19:55:56.350196	+32:33:04.51296
J1957+3427	19:57:34.454854	+34:27:54.63069
J1958+3424	19:58:00.888263	+34:24:57.15190
J2001+3323	20:01:42.209147	+33:23:44.76688
J2035+5821	20:35:23.752366	+58:21:18.74492
J2042+0356	20:42:14.501776	+03:56:13.93861
J2051-0553	20:51:12.420847	-05:53:04.76987
J2105+0033	21:05:07.715375	+00:33:25.00833
J2105+5356	21:05:12.766363	+53:56:09.55415
J2108+5238	21:08:10.979713	+52:38:09.21789
J2117+5431	21:17:56.484554	+54:31:32.50118
J2118+5339	21:18:25.374120	+53:39:04.51437
J2130+0502	21:30:32.877414	+05:02:17.47505
J2131+5214	21:31:58.114861	+52:14:16.79792
J2210+2013	22:10:51.652450	+20:13:24.05463
J2219+2141	22:19:38.521601	+21:41:12.56039
J2229+0114	22:29:51.801901	+01:14:56.72416
J2245+6537	22:45:42.584983	+65:37:14.96734

Continuation of Table 2.3

Name	RA	DEC
J2253+6005	22:53:49.855993	+60:05:00.45667
J2253+6043	22:53:40.703743	+60:43:36.00985
J2254+6133	22:54:52.745207	+61:33:13.56031
J2258+6325	22:58:59.512568	+63:25:29.20646
J2302+6246	23:02:03.750247	+62:46:08.35119
J2314+5813	23:14:19.085658	+58:13:47.63800

Table 2.3: Coordinates of the OFCORS sources

2.4 Wide-Field Infrared Survey Explorer

Launched in December 2009, the Wide-Field Infrared Survey Explorer (WISE) is an infrared space observatory with a 40 cm diameter aperture and a field of view of 47'. [23] Observations were performed in four bands, listed in Table 2.4, with a spatial resolution of 6.1", 6.4", 6.5", and 12" respectively. The WISE mission can be separated into two parts. The first 10 month mission focused on an all sky survey in the four bands [23], covering 99% of the sky, and ended as a result of running out of its reservoir of liquid hydrogen required to keep the instrument at its operating temperature. Since W1 and W2 can operate at a higher temperature, WISE continued to observe using these two bands to search for near Earth objects and brown dwarfs under the Near-Earth Object WISE survey, or NEOWISE, which lasted for four months. [24] In 2013, WISE resumed observations of near-earth objects using these bands.

The WISE All-Sky catalog [23] was used in this project. This catalog contains infrared

Band	Wavelength
W1	3.4 μm
W2	4.6 μm
W3	12 μm
W4	22 μm

Table 2.4: WISE Bands and Wavelengths

data for 564 million objects that have a signal-to-noise ratio greater than 5 in at least one band. Using a 3'' cross-matching radius the WISE counterparts of each OFCORS were identified, as seen in Figure 2.5, 87 of the OFCORS were detected in at least W1. The magnitude for the WISE catalogs are in the Vega magnitude system, which are converted to the AB magnitude system by using the offset between AB and Vega magnitudes for each band from Martini's useful astronomical data page.² Some of the bands only have the magnitude upper limits. For these sources the magnitude error is set to -1.0. For the bands that have no detections or upper limits, -99.0 is used in the catalog as a placeholder for magnitude and magnitude error.

²<http://www.astronomy.ohio-state.edu/martini/usefuldata.html>

Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J0037+0808	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0155-1443	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0158-1413	20.038	0.141	20.113	0.31	17.129	0.272	15.08	0.269
J0200-1542	18.652	0.052	17.501	0.042	15.711	0.079	15.636	0.459
J0210-0737	18.654	0.045	18.621	0.084	17.386	-1.0	15.806	-1.0
J0253+0024	19.089	0.06	19.104	0.124	17.469	-1.0	15.631	-1.0
J0254-0225	19.811	0.105	19.582	0.178	17.722	-1.0	15.937	-1.0
J0305+0523	18.838	0.054	18.089	0.059	16.513	0.177	15.264	-1.0
J0311+0508	19.706	0.098	19.519	0.166	16.968	-1.0	15.207	0.405
J0330-0551	17.897	0.039	18.042	0.06	17.005	-1.0	15.555	-1.0
J0418+5457	17.755	0.037	17.393	0.043	16.71	0.177	14.865	-1.0
J0422+0827	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0425+0833	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0425+1755	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0501+1356	18.9	0.07	19.057	0.146	17.224	0.339	15.489	0.517
J0543+3023	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J0559+2353	17.8	0.049	17.111	0.04	15.892	0.126	15.018	0.378
J0636+0801	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0720-1308	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0724-1505	19.411	0.103	18.655	0.1	16.502	0.166	15.186	0.283
J0726+3912	19.381	0.09	18.868	0.108	17.073	-1.0	15.274	-1.0
J0754+5324	19.301	0.077	18.695	0.097	16.649	0.192	15.238	-1.0
J0805+2106	18.352	0.047	17.923	0.056	16.011	0.129	15.263	0.379
J0811+4308	19.406	0.09	20.054	0.32	17.598	-1.0	15.614	-1.0
J0819+3823	18.817	0.059	18.754	0.099	17.08	0.311	15.074	-1.0
J0857+2805	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0904+2729	19.012	0.072	18.712	0.113	17.473	0.428	15.531	-1.0
J0911+3033	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J0917+6530	18.99	0.063	18.814	0.105	16.635	-1.0	14.828	-1.0
J0923+3849	18.356	0.046	17.417	0.043	15.887	0.115	14.884	0.228
J0945+2729	19.348	0.085	19.228	0.151	17.783	-1.0	15.344	-1.0
J0947+5907	20.472	0.161	20.592	-1.0	17.534	-1.0	15.171	-1.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J0952+4941	19.588	0.093	19.678	0.194	17.108	-1.0	15.751	-1.0
J1002+5701	21.215	0.347	19.871	0.218	16.467	0.133	15.173	0.305
J1006+3050	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1006+4836	18.897	0.057	19.177	0.124	17.194	-1.0	15.536	0.445
J1009+0622	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1013+3043	19.54	0.103	19.712	0.241	17.182	-1.0	15.563	-1.0
J1025+8144	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1041+4900	20.314	0.17	19.849	0.228	17.226	-1.0	15.511	-1.0
J1042+2949	19.699	0.116	19.169	0.139	17.596	0.439	15.129	-1.0
J1048+0141	19.016	0.077	18.318	0.08	17.06	0.345	15.173	0.451
J1058+4248	20.819	0.287	19.24	0.147	17.917	-1.0	15.388	-1.0
J1102+7905	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1103+2203	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1104+5752	20.576	0.21	20.329	-1.0	17.983	-1.0	14.895	-1.0
J1109+3744	19.299	0.078	18.955	0.121	16.829	0.213	15.443	0.39
J1110+6028	19.951	0.114	19.427	0.144	17.515	0.359	15.264	-1.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J1120+3257	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1124+3214	18.526	0.05	18.144	0.062	17.558	0.398	15.091	-1.0
J1131+3013	18.738	0.055	18.459	0.08	17.348	-1.0	15.104	-1.0
J1135+4258	19.018	0.054	19.054	0.094	17.782	0.395	15.623	-1.0
J1137+4741	19.124	0.064	18.992	0.109	17.275	-1.0	15.445	-1.0
J1143+1834	19.12	0.075	19.03	0.159	17.477	0.412	15.575	0.541
J1152+3327	18.511	0.048	18.646	0.091	17.77	-1.0	15.695	0.534
J1204+3023	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1208+5413	19.661	0.09	18.821	0.093	17.179	0.281	15.345	-1.0
J1220+6327	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1237+2034	18.959	0.065	18.176	0.064	16.165	0.114	15.078	0.315
J1239+3705	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1241+5458	20.065	0.124	20.112	0.265	17.354	-1.0	15.596	-1.0
J1251+2102	18.819	0.054	19.098	0.114	17.723	0.497	15.96	-1.0
J1254+1843	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1256+2220	19.755	0.1	20.256	0.323	17.857	0.471	15.624	-1.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J1304+3523	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1304+5056	19.912	0.107	19.812	0.192	17.021	0.232	15.888	-1.0
J1311+1658	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1314+5306	18.979	0.057	18.224	0.058	16.487	0.124	14.996	0.243
J1320+5036	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1322+4303	20.126	0.138	19.984	0.239	17.839	-1.0	15.173	-1.0
J1326+4434	19.939	0.121	19.253	0.125	17.568	-1.0	15.982	-1.0
J1340-0335	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1344+1409	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1350+6428	17.7	0.027	17.268	0.029	15.598	0.047	15.003	0.156
J1412+1334	20.011	0.127	20.019	0.251	17.362	-1.0	15.58	0.401
J1412+3200	19.141	0.063	19.893	0.221	17.853	-1.0	16.029	-1.0
J1416+3444	19.981	0.116	19.354	0.136	17.578	0.295	15.722	0.497
J1434+4236	19.028	0.055	18.298	0.057	16.959	0.172	15.187	-1.0
J1439+2114	18.889	0.055	17.786	0.045	15.51	0.057	14.376	0.141
J1442+3042	19.224	0.064	18.997	0.099	17.391	0.242	16.077	0.529

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J1448+5326	19.551	0.065	19.239	0.093	18.032	-1.0	16.158	-1.0
J1457+0938	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1457+2435	20.648	0.204	20.03	0.259	17.652	-1.0	15.928	-1.0
J1500+0839	19.43	0.076	19.346	0.145	18.185	-1.0	15.86	-1.0
J1504+3249	20.612	0.166	19.757	0.149	17.967	0.356	15.884	-1.0
J1514+3046	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1532+2919	19.616	0.077	19.652	0.152	18.281	-1.0	16.169	-1.0
J1538+0019	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1556+1825	18.904	0.058	17.974	0.057	16.064	0.12	15.537	0.41
J1559+1624	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1602+3326	18.417	0.042	17.765	0.042	16.103	0.071	15.478	0.263
J1603+2126	18.065	0.038	17.855	0.05	16.396	0.152	15.072	0.27
J1604+4746	19.779	0.071	19.565	0.112	18.026	0.428	15.734	-1.0
J1605+2214	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1605+5931	19.069	0.043	18.785	0.058	18.283	0.334	16.029	-1.0
J1624+2748	19.501	0.073	19.748	0.167	17.329	-1.0	15.467	-1.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J1630+2131	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1632+3547	18.237	0.037	18.195	0.053	17.891	0.537	15.614	0.472
J1648+4233	18.891	0.044	18.236	0.047	17.316	0.294	15.253	-1.0
J1700+3830	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1718+2948	18.877	0.052	17.502	0.035	15.703	0.074	15.517	0.468
J1722+6144	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1735+5049	18.478	0.035	17.882	0.035	16.826	0.123	15.331	0.229
J1741+4751	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1746+6421	18.401	0.03	18.244	0.038	17.337	0.139	16.209	0.384
J1846+2036	18.477	0.058	18.154	0.069	16.926	0.212	15.271	0.307
J1909+7813	19.663	0.066	19.547	0.098	17.695	-1.0	16.213	-1.0
J1922+1504	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1922+1530	15.56	0.054	14.923	0.04	14.063	0.087	13.06	0.071
J1924+1558	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1949+7654	19.455	0.054	18.867	0.06	18.298	0.464	15.745	-1.0
J1955+3233	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J1957+3427	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J1958+3424	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2001+3323	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2035+5821	20.019	0.125	18.515	0.066	16.846	0.142	15.394	-1.0
J2042+0356	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2051-0553	19.398	0.092	19.056	0.151	17.12	-1.0	15.04	-1.0
J2105+0033	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2105+5356	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2108+5238	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2117+5431	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2118+5339	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2130+0502	18.895	0.056	18.847	0.134	17.33	-1.0	15.604	-1.0
J2131+5214	18.852	0.077	19.07	0.104	16.979	0.233	15.92	-1.0
J2210+2013	17.807	0.036	17.313	0.039	16.531	0.215	14.945	0.306
J2219+2141	18.028	0.036	17.471	0.04	16.188	0.121	15.127	0.328
J2229+0114	19.719	0.122	18.787	0.114	17.704	0.529	15.391	-1.0

Continuation of Table 2.5

Name	W1	W1_error	W2	W2_error	W3	W3_error	W4	W4_error
J2245+6537	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2253+6005	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2253+6043	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2254+6133	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0
J2258+6325	17.797	0.031	17.743	0.036	15.505	0.05	14.312	0.099
J2302+6246	16.891	0.035	17.18	0.042	16.402	0.276	15.975	-1.0
J2314+5813	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0

Table 2.5: Data for the OFCORS in WISE

Band	Wavelength
Y	1.02 μm
J	1.25 μm
H	1.65 μm
K	2.2 μm

Table 2.6: UKIDSS Bands and Wavelengths

2.5 UKIRT Deep Sky Survey

The UKIRT Deep Sky Survey [25], or UKIDSS, is a set of five astronomical surveys performed by the United Kingdom Infrared Telescope, located on Mauna Kea in Hawaii, using the WFCAM wide field camera. [26] The UKIDSS mission goals can be separated into four parts: looking at the coolest and nearest brown dwarfs, high redshift dusty starburst galaxies, elliptical galaxies and galaxy clusters at $1 < z < 2$, and the highest redshift quasars located at $z=7$. It covered 7500 square degrees of the Northern Sky to varying depths using YJHK bands, whose wavelength information is listed in Table 2.6.

Each of the UKIDSS five surveys are designed to achieve these goals. Two of these were directed toward galactic targets while the other three targeted sources outside of our galaxy. The Large Area Survey (LAS) covers an area of 4000 square degrees in YHJK to a depth of $K=18.4$. The LAS has also been covered by the SDSS, which makes it ideal for the use of multi-wavelength studies for sources that fall in this field. The Galactic Plane Survey (GPS) covers an area of 1800 square degrees in JHK to a depth of $K=19.0$, and 300 square degrees have also been covered using a H_2 filter. The goal of GPS is to observe the Galactic Plane in the infrared, since infrared wavelengths do not suffer from dust absorption like optical wavelengths. The Galactic Clusters Survey (GCS) covers an area of 1400 square degrees in JHK to a depth of $K=18.7$. This survey focuses on ten open star clusters with the goal of measuring the mass function in different galactic environments. The Deep Extragalactic

Name	Y	Y Err	J	J Err	H	H Err	K	K Err
J0253+0024	20.179	0.176	-99.0	-99.0	-99.0	-99.0	17.364	0.119
J0305+0523	20.362	0.252	20.143	0.359	18.510	0.136	17.469	0.096
J0805+2106	-99.0	-99.0	-99.0	-99.0	18.782	0.150	17.067	0.062
J0904+2729	-99.0	-99.0	20.145	0.226	-99.0	-99.0	17.886	0.156
J1048+0141	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	18.105	0.166
J1532+2919	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	18.215	0.176

Table 2.7: UKIDSS data for the OFCORS

Survey (DXS) covers an area of 35 square degrees in JK to a depth of $K=21.0$ with 5 square degrees also imaged in H. It targets fields at high galactic latitudes with low extinction and overlaps other deep observations made at other wavelengths. The Ultra Deep Survey (UDS) covers 0.77 square degrees of the XMM-LSS field in JHK to a depth of $K=23.0$. The goal of this survey was to perform the deepest near-infrared survey over an area of sky with the aim of studying the galaxy evolution in the early universe.

A 3" search radius was used to find the OFCORS infrared counterparts in UKIDSS. Six of the OFCORS are detected using UKIDSS, whose magnitudes and magnitude errors are listed in Table 2.7. None of these OFCORS are observed with OSMOS because the UKIDSS data was released after the OSMOS observations.

2.6 Visible and Infrared Survey Telescope for Astronomy

The Visible and Infrared Survey Telescope for Astronomy, or VISTA, is a set of six infrared surveys observing the Southern sky. [27] The observations were performed by the VLT Survey Telescope, a 4.1 m telescope, located at the Paranal Observatory in Chile, using the VISTA InfraRed CAMera (VIRCAM) that uses 6 filters listed in Table 2.8. The VISTA survey consisted of the following goals: searching for brown dwarfs, variable stars, and

Band	Wavelength
Z	0.88 μm
Y	1.02 μm
J	1.25 μm
H	1.65 μm
Ks	2.15 μm
NB	1.18 μm

Table 2.8: VISTA Bands and Wavelengths

Name	Y	Y Err	J	J Err	H	H Err	K	K Err
J0210-0737	-99.0	-99.0	20.910	0.115	-99.0	-99.0	20.098	0.173
J0253+0024	20.812	0.175	20.749	0.162	-99.0	-99.0	19.969	0.176

Table 2.9: VISTA data for the OFCORS

quasars, mapping the structure of our galaxy, studying the Magellanic Clouds, and studying galaxy evolution.

Using a cross-matching radius of 3'', only two of the OFCORS were detected in OS-MOS. One of these sources, J0253+0024, was also detected using UKIDSS. The magnitude and magnitude error for the OFCORS detected by UKIDSS are listed in Table 2.9. For non-detections, magnitude is recorded as -99.

2.7 Canadian France Hawaii Telescope

The Canadian France Hawaii Telescope, or CFHT, is a 3.6 m optical/infrared telescope located at Mauna Kea. This telescope is operated in an agreement by the University of Hawaii, the National Research Council (NRC) in Canada, and the Centre National de la Recherche Scientifique (CNRS) in France. CFHT currently operates four instruments: MegaCam, a one square degree field optical imager imaging in the SDSS ugriz bands, WIRCam, an infrared instrument optimized for J, H, and K spectral bands, ESPaDOnS

name	u	u err	g	g err	r	r err	i	i err	z	z err
J0210-0737	24.272	0.082	24.361	0.049	23.421	0.050	22.439	0.038	21.817	0.053
J0253+0024	-99.0	-99.0	-99.0	-99.0	-99.0	-99.0	22.247	0.040	-99.0	-99.0
J2210+2013	-99.0	-99.0	-99.0	-99.0	20.696	0.004	-99.0	-99.0	-99.0	-99.0

Table 2.10: CFHT data for the OFCORS

an echelle spectrograph/spectropolarimeter, and SITELLE, a wide-field Fourier transform spectrograph.

The CFHT has a searchable archive that includes both science images and photometric catalogs of all the sources detected in the science image. Using an 1" search radius, 3 of the OFCORS were detected in at least one band and listed in Table 2.10.

2.8 Unified Radio Catalog

The Unified Radio Catalog (URC) is a combined catalog of five large area radio sky surveys and SDSS. [28] The five radio surveys used to create this catalog are FIRST (Faint Images of the Radio Sky at Twenty Centimeters), [3] NVSS (NRAO-VLA Sky Survey, 20cm), [29] GB6 (Green Bank Telescope 6 cm Sky Survey), [30] WENSS (Westerbork Northern Sky Survey, 92cm), [31] and VLSSr (VLA Low-Frequency Sky Survey Redux). [32] To build the URC, the FIRST sources were matched to the NVSS survey and then NVSS sources were matched to the FIRST survey. For each FIRST and NVSS source, the closest three neighbors from the other catalog are listed within 30". The combined catalogs of FIRST and NVSS are then crossmatched with the GB6, WENSS, and VLSSr surveys using a cross matching radius of 120". Each position of FIRST and NVSS sources were also matched to SDSS using a 30" radius. The current version of the URC is version 2.0.

The complete catalog contains 2,866,856 entries in the region of the sky north of -40 declination covered by the NVSS (which completely contains the part of sky observed by

FIRST). 3269 deg^2 is observed by all five radio surveys and by the SDSS. The overlapping region contains about 178,000 FIRST-NVSS matches, 110,000 isolated FIRST sources, and 23,000 isolated NVSS sources. The catalog includes data parameters for the closest positional matches.

2.9 Pan-STARRS

The Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) is an ongoing survey continuously observing the sky across 5 filters, as seen in Table 2.11. [33] The overall goal is to search for changes in the sky, for example high proper motion objects or variable stars. Its primary mission is to detect near-Earth objects that threaten impact events, and it is expected to create a database of all objects visible from Hawaii (three-quarters of the entire sky) down to an apparent magnitude of 24. Consisting of two 1.8m telescopes, Pan-STARRS is capable of covering all of the sky visible from Hawaii, or 75% of the sky, four times every month. By the end of its initial three-year mission in April 2014, PS1 had imaged the sky 12 times in each of the 5 filters. In December 2016, the first data release occurred, which has a $5\text{-}\sigma$ magnitude limit that is an order of magnitude deeper than SDSS. Since Pan-STARRS covers the entire SDSS field, making it ideal for searching for OFCORS, which may be invisible in SDSS due to having a higher magnitude limit.

The Pan-STARRS data for the OFCORS is grabbed from the Pan-STARRS archive using a matching radius of $3''$. Since the Pan-STARRS catalog contains both single image photometry and stacked photometry, only sources that were detected in the stack and by visual inspection were considered detections. Using this method, seven of the OFCORS were detected in Pan-STARRS and their magnitude and magnitude error are listed in Table 2.12.

Filter	Wavelength
g	486.6 nm
r	621.5 nm
i	754.5 nm
z	867.9 nm
y	963.3 nm

Table 2.11: PanSTARRS Filters and Wavelengths

name	g	g err	r	r err	i	i err	z	z err	y	y err
J0305+0523	99.99	99.99	99.99	99.99	21.872	0.189	99.99	99.99	20.76	0.19
J1152+3327	99.99	99.99	99.99	99.99	21.761	0.127	20.553	0.151	99.99	99.99
J1630+2131	99.99	99.99	21.345	0.162	19.501	0.571	18.336	0.032	17.952	0.048
J1735+5049	99.99	99.99	99.99	99.99	21.99	0.151	99.99	99.99	99.99	99.99
J1746+6421	99.99	99.99	23.192	0.34	99.99	99.99	99.99	99.99	99.99	99.99
J2210+2013	20.953	0.09	20.313	0.045	20.191	0.069	19.667	0.11	19.588	0.046
J2219+2141	19.021	0.02	18.895	0.016	18.767	0.011	18.83	0.012	18.822	0.06

Table 2.12: PanSTARRS data for the OFCORS

Chapter 3

Observations

3.1 Is SDSS deep enough?

The OFCORS are defined as all compact radio sources in the SDSS footprint that lack an optical counterpart. When it comes to optical/infrared imaging however, an object's invisibility is subjective. It is dependent on how many photons can be collected by the telescope. [34] The number of photons collected can be calculated by

$$\begin{aligned} \text{number of photons} = & (\text{photon flux})(\text{collecting area of telescope}) \times \\ & (\text{CCD quantum efficiency})(\text{exposure time}). \end{aligned} \quad (3.1)$$

An increase in the photon flux, collecting area of the telescope, CCD quantum efficiency, or exposure time increases the number of photons collected. This increase allows for fainter sources to be detected. Photon flux is calculated by

$$\text{photon flux} = (\text{number of photonsemitted})(1 - f)/(4\pi * d^2) \quad (3.2)$$

where f is the fraction reflected, scattered, absorbed by the interstellar medium or the earth's atmosphere and d is the distance from the telescope to the light source. The collection area of the telescope is approximately

$$\text{collection area of the telescope} = \pi(D/2)^2 \quad (3.3)$$

where D is the diameter of the telescope. The quantum efficiency of the CCD represents the ratio of the number of electrons produced to the number of incoming photons and is a function of the wavelength of light being detected. Finally, exposure time is how long the telescope observes the sky. By increasing the diameter of the telescope, the quantum efficiency of the CCD, or exposure time will allow for deeper observations and fainter sources to be detected. The quantum efficiency of the CCD and the diameter of the telescope cannot be changed without either using a different instrument or a different telescope. However, the exposure time can be changed and is often the easiest way to increase the number of photons detected.

Since SDSS has a magnitude limit of ~ 22.2 in r band, the OFCORS requires deeper observations for detection. Detection of the OFCORS at optical wavelengths allows for the exploration of their properties. To obtain optical observations deeper than SDSS, we performed new observations using OSMOS, and an archive search using CFHT and Pan-STARRS.

3.2 OSMOS description

In order to perform deeper optical imaging, we used the Ohio-State Multi-Object Spectrograph (OSMOS) located on the 2.4m Hiltner Telescope, the instrument can be seen in



Figure 3.1: OSMOS instrument

run	date
1	04/21/12-04/26/12
2	12/04/12-04/10/12
3	03/13/13-03/18/13
4	11/26/29-11/29/13
5	03/24/14-03/26/14

Table 3.1: Observation dates related to run number.

Figure 3.1. [35] OSMOS uses an all-refractive, zero-deviation design that projects a 20' diameter field of view onto a 4064x4064 MDM4K CCD with a plate scale of 0.273"/pixel. Even though spectrograph is in the name, this instrument is capable of performing both imaging and spectroscopy, and our observations only used the imaging mode.

3.3 OSMOS Observations

We performed a total of five observation runs using OSMOS. Table 3.1 lists the run number and observation dates. 17 sources were chosen for observation and selected from those OFCORS that were the brightest in radio, subsequently giving the most extreme values of $F_{optical}/F_{radio}$. Table 3.2 lists the sources observed and their coordinates. During the first run we focused on the i and z bands, and had an exposure time of 96 minutes. After this run, a concern arose that the OFCORS may not have a steep rising SED. Therefore, it was decided to stop detections in z band and instead focus on g, r, and i bands. A decision was also made to increase the total exposure time to 3 hours to obtain deeper images, however, this was not always possible due to weather and observation constraints. Table 3.2 lists the run, the filters used, and the total exposure time for each source observed by OSMOS.

name	RA	DEC	run	filters	exptime (s)
J0037+0808	00:37:32.19	+08:08:13.05	2	r	10080
J0210-0737	02:10:16.52	-07:37:20.69	2	r	6480
J0418+5457	04:18:19.34	+54:57:15.33	4	r,i	10080,10800
J0559+2353	05:59:32.03	+23:53:53.92	2,4	g,r,i	9360, 10080,10800
J0947+5907	09:47:04.86	+59:07:41.47	2,4	g,r,i	7200, 5760 ,10800
J1110+6028	11:10:13.08	+60:28:42.56	5	r	5760
J1204+3023	12:04:58.01	+30:23:25.46	3	r,i	10800,7200
J1220+6327	12:20:46.84	+63:27:29.01	1,3	r,i,z	10800,10800,2160
J1254+1843	12:54:34.02	+18:43:32.30	5	r	10800
J1320+5036	13:20:42.20	+50:36:07.79	3	r,i	10800,10800
J1416+3444	14:16:04.18	+34:44:36.42	5	r	5760
J1434+4236	14:34:27.86	+42:36:20.07	1	i,z	5760,5760
J1439+2114	14:39:08.90	+21:14:50.82	1	z	5760
J1602+3326	16:02:07.26	+33:26:53.07	1	z	1440
J2117+5431	21:17:56.48	+54:31:32.50	2	g,r	7920,9360
J2210+2013	22:10:51.65	+20:13:24.05	2	g,r	6480,2880
J2314+5813	23:14:19.08	+58:13:47.63	4	r,i	10800,10800

Table 3.2: columns are as follows: a) the source name b) the RA coordinate in J2000, c) the DEC coordinate in J2000, d) the observation run number, e) the filters used, f) the exposure time in seconds in order of filters used

Chapter 4

Data Reduction

4.1 OSMOS Data Reduction

Raw images have many imperfections that can be corrected by a series of data reduction steps to obtain a usable science image. Then, to increase the signal-to-noise ratio, the science images are stacked, resulting in a stacked image in which photometry will be performed.

4.1.1 Bias and Overscan

The first data reduction step is to correct for the bias, which is done by using the overscan region of each CCD. To produce an overscan region, the clocks on the CCD read several pixels after the last physical column, producing extra columns in the raw image containing unexposed pixels. The median pixel values in the overscan region are chosen as the bias level, which are used in a bias correction by the equation

$$Z_i = Z + (\text{median}P(\omega_i - \Omega_Z)) \quad (4.1)$$

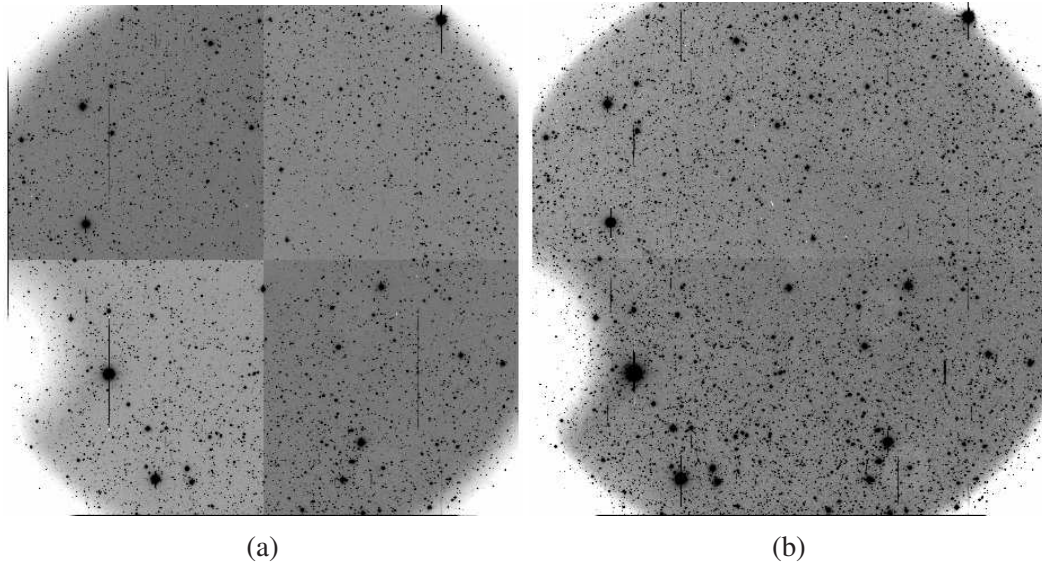


Figure 4.1: Example of bias correction. a) is the raw image while b) is after the bias correction has been applied

where Z is the full two dimensional image, ω_i is the overscan portion of the image i , and Ω_Z is the overscan portion of the combined Z , and Z_i is for image i . [34]

For OSMOS images, the bias is corrected by a python script `bias4k.cl` located in the OSMOS handbook.¹ This script was applied to all raw and flat-field images to get bias corrected images and bias corrected flat-field images. Figure 4.1a shows the raw image taken by OSMOS while Figure 4.1b shows the bias corrected image.

4.1.2 Flat Fielding

Identical signals do not produce identical responses in every pixel of a detector array. Therefore, the signal is not smooth across the entire image. This defect is caused by structural quantum-efficiency differences intrinsic to the array, vignetting, or other imperfections in the optical system. This is corrected by stacking the individual flat-field images to build

¹<http://www.astronomy.ohio-state.edu/martini/osmos/>

a stacked flat-field image. Then, the bias corrected image is divided by the stacked flat-field image to get the flat-field corrected image. [34]

For the OSMOS data, the bias-corrected flat-field images for each band from each night were used to build the flat-field images. If there were not at least 3 flat-field images in a band on a single night, then the flat-field images from multiple nights during the same observation run were used. Table 4.1 lists the nights were used to build each flat-field correction.

To build each flat-field correction, the IRAF task flatcombine was used. An example parameter file for flatcombine is in Appendix A.1. Only 3 of the parameters were changed from the the default parameters for flatcombine. First, the reject parameter was set to cddclip which is used when the gain, readout noise, and sensitivity noise parameters for the CCD are known. Then, using the values from the OSMOS handbook, the gain parameter is set to 2.2 and the rdnoise parameter is set to 5. The input parameter contains the file that lists all the flat field images used in creating the flat field correction, and the output parameter is the name of the flat-field correction. Figure 4.2 shows an example of a flat-field correction created by flatcombine.

Then, using the IRAF task ccdproc, the flat-field correction is applied to the bias-corrected images for the applicable nights, to get the flat-field corrected images. An example of the ccdproc parameter file is in Appendix A.2, where starting with the default parameters for ccdproc, all of the correction parameters are set to no except for flatcor. An example of a "before" bias corrected image is in Figure 4.3a, and Figure 4.3b is an example of "after" flat field correction.

Date of observation	filter	Flats used for flat-fielding
20120421	i	20120426
20120421	z	20120422 first half
20120422 first half	z	20120422 first half
20120422 second half	z	20120422 second half
20120423	z	20120423
20120424	z	20120424
20120426	i	20120426
20120426	z	20120424
20121204	g	20121205
20121205	g	20121205
20121206	g	20121206
20121206	r	20121206
20121207	r	20121207, 20121208
20121208	r	20121207, 20121208
20121209	r	20121209
20121210	g	20121210
20121210	r	20121209
20130313	r	20130313
20130314	i	20130314, 20130317
20130315	r	20130315
20130316	i	20130314, 20130317
20130317	i	20130314, 20130317
20130318	r	20130318
20130318	i	20130318
20131126	r	20131127
20131127	r	20131127
20131128	i	20131128
20131129	r	20131127
20131129	i	20131129
20140324	r	20140324, 20140326
20140325	r	20140324, 20140326
20140325	i	20140325
20140326	r	20140324, 20140326

Table 4.1: Flat-fields and which ones applied to which night. Column 1 is the observation date, Column 2 is the filter used, and Column 3 is which nights flat-fields were used to build the twilight flat

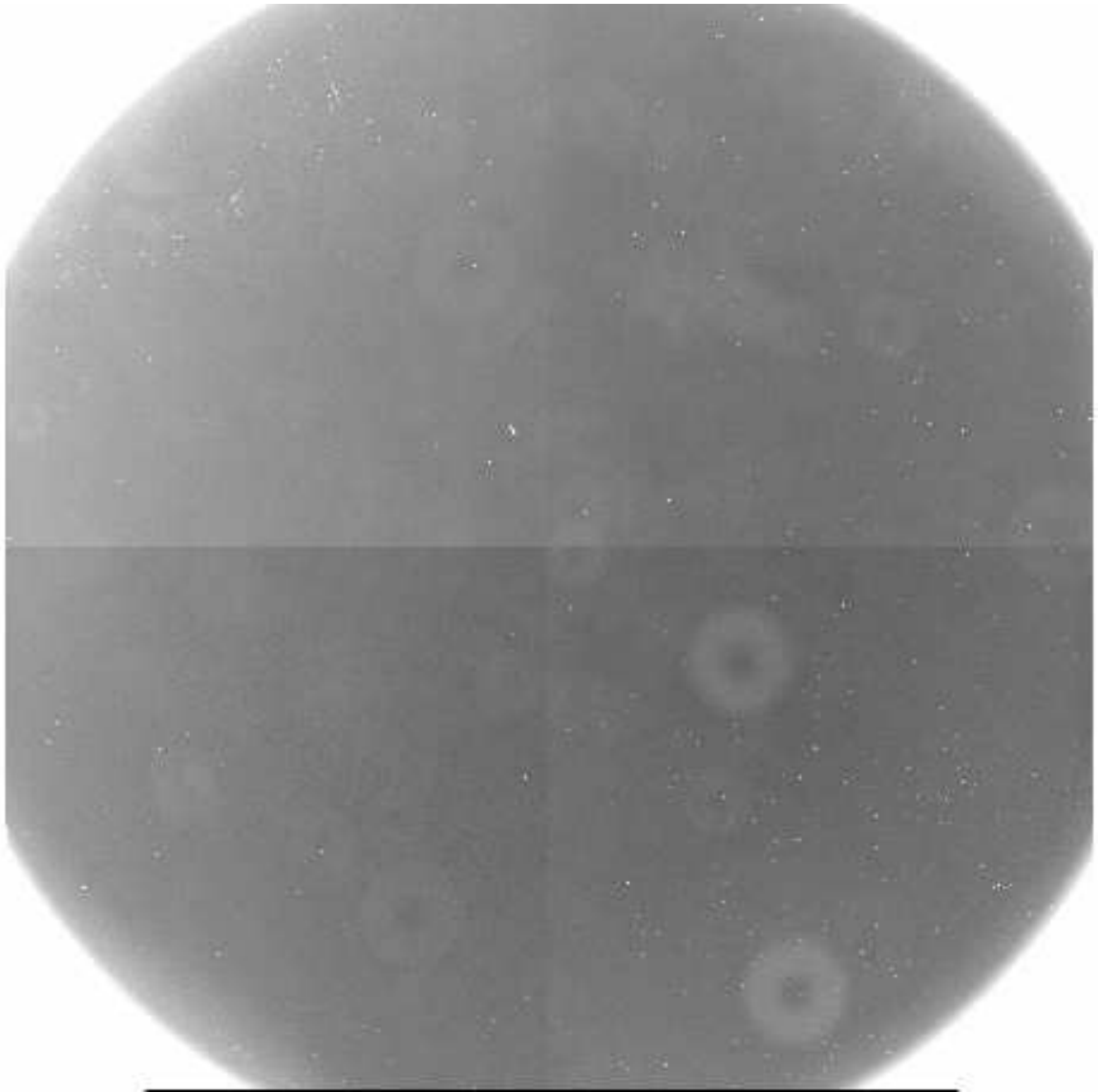


Figure 4.2: Example of a flat-field correction

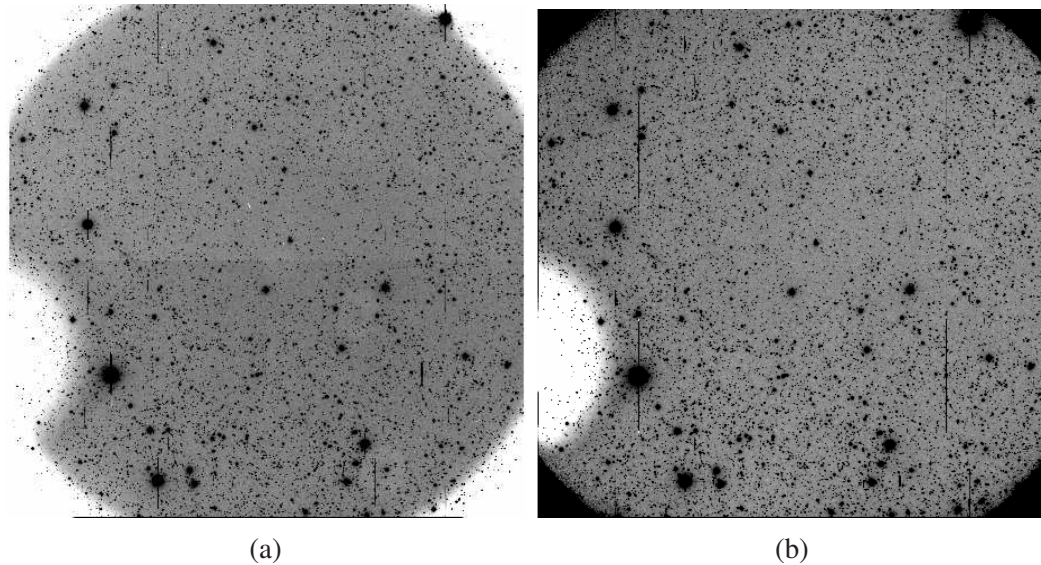


Figure 4.3: Example of flat correction. a) is the bias corrected image while b) is after the flat field correction has been applied

4.1.3 Illumination Correction

After flat-field correction, the images still do not have an even background, requiring an illumination correction. To start building the illumination corrections, first a weight image needs to be built which contains information on which pixels to ignore when building the illumination correction. To build a weight image, each flat-field corrected image was displayed in Ds9 and regions were created to define the areas in the image that do not have data. Then the weight image was built with a homebrew script of steps using the shell and Pyraf tasks to create the weight image, as seen in Appendix B.

Next, the flat-field corrected images are stacked to remove stars, leaving only the background. This stack is then used to create the illumination correction by using the IRAF task `combine`. An example set of parameters is seen in Appendix A.3, where starting with the default parameters and changing the `reject` parameter to `ccdclip`, the `scale` parameter to `mode`, the `zero` parameter to `none`, the `rdnoise` to 5, and the `gain` parameter to 2.2. The

input parameter was set to the file containing the list of images to combine and the weight parameter is set to the file containing the list of weight images corresponding to the images to combine.

Then, the IRAF task `imedit` is used to remove any bad pixels by user inspection. After removing the bad pixels, the IRAF task `boxcar` is used to smooth the stacked image to create the illumination correction. The smoothing by boxcar will be done by either 200x200, an example of the parameter input can be seen in Appendix A.4. An example of the illumination correction can be seen in Figure 4.4. Finally, the illumination correction will be applied to each of the flat-field corrected images using the IRAF task `ccdproc`. An example of which can be seen in Appendix A.5, where the default settings are used, but this time around all of the correction parameters are turned off except the `illumcor` parameter. The output from `ccdproc` will be the science images. An example of before the illumination correction is Figure 4.5a and after the correction is Figure 4.5b.

4.1.4 Fringe Correction

For z band images, a fringe correction is needed. These fringes result from the brightness patterns in a CCD image due to reflection and interference within the thin layers of the device, usually caused by narrow night-sky emission lines. They tend to occur in either very narrow band images or in the far red where the night-sky upper-atmospheric OH emission is bright.

To remove the fringes in the z band images, first the weight image needs to be built for each z band illumination corrected image by displaying the z band illumination corrected image in Ds9, and creating regions defining all areas that do not have data. Then, a home-brew script, Appendix B, was used to create the weight images for each flat-field corrected

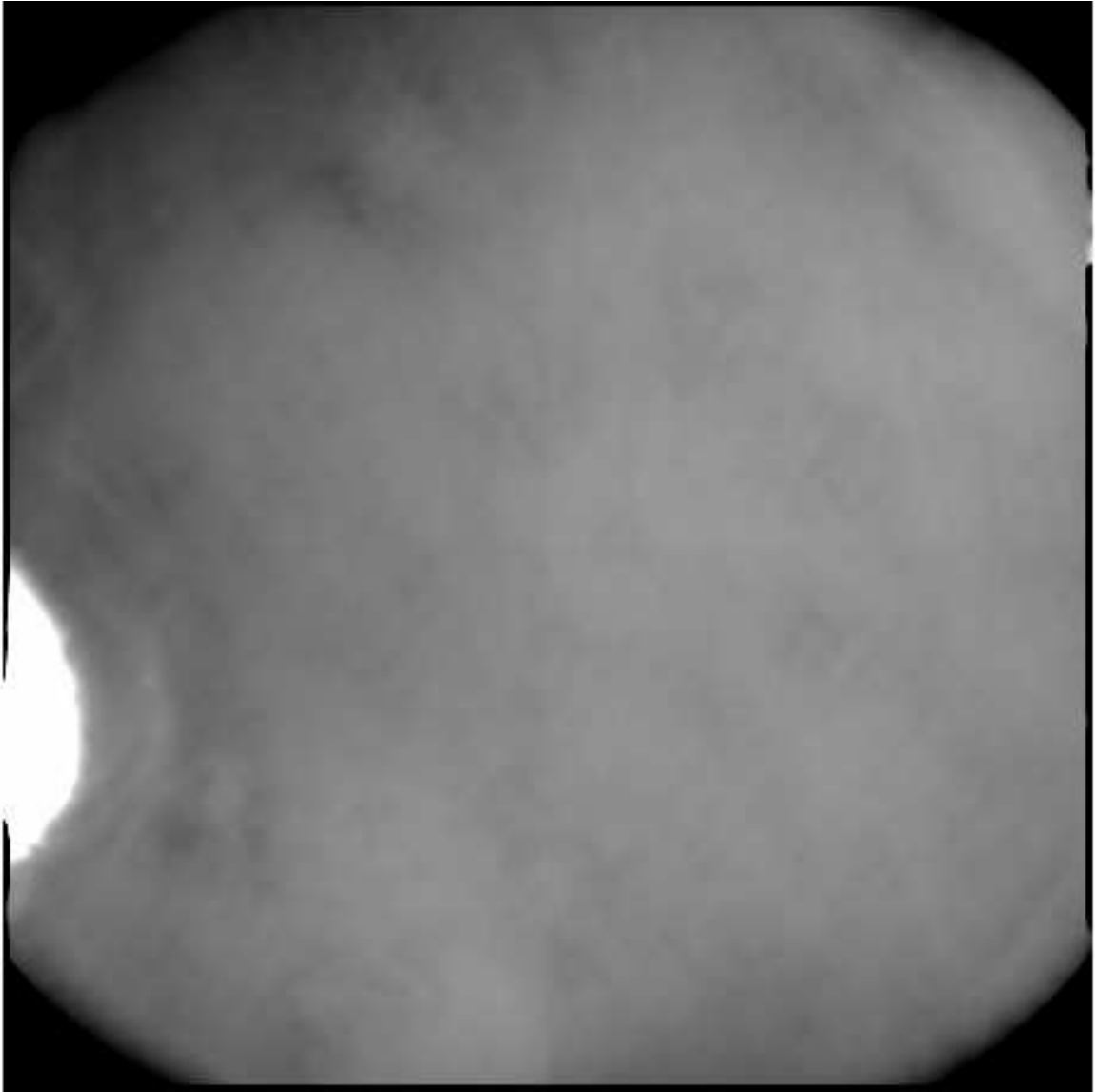


Figure 4.4: Example of the illumination correction

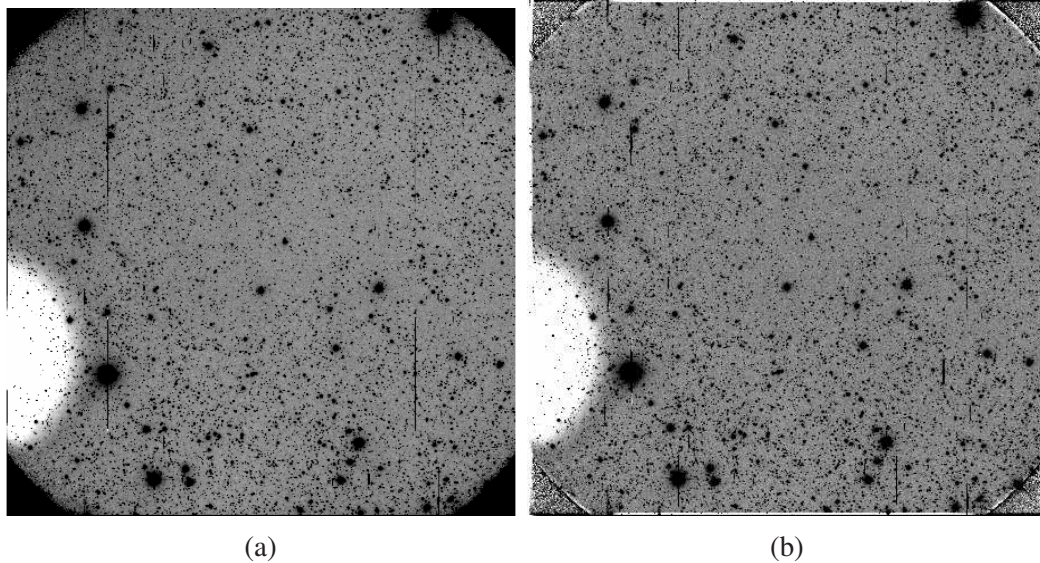


Figure 4.5: Example of illum correction. a) the flat-field corrected image b) after the illumination correction has been applied

image.

Next, the illumination corrected images are stacked using the IRAF task `combine`. The default parameters were used except the following: the `reject` parameter is set to `ccdclip`, the `scale` parameter is set to `mode`, the `zero` parameter is set to `none`, the `rdnoise` is set to 5, and the `gain` parameter is set to 2.2.

After stacking, the fringe correction is created using the IRAF task `mkfringecor` with a 200x200 smoothing parameter. Then the fringe correction is applied to the illumination corrected images by the IRAF task `rmfringe` to get the z band science images.

4.1.5 Scamp

The next data reduction step is to define a coordinate system for each science image. This allows a stacking algorithm to know which pixels to match between multiple images. The coordinate system is defined using the World-Coordinate System or WCS. To get the

WCS for each science image, the Software for Calibrating AstroMetry and Photometry, or SCAMP, was employed using the coordinate catalog created by SExtractor of all the sources in the science image, and matching it with a reference catalog. Then by comparing the geometry of both catalogs, SCAMP calculates the WCS information, and puts it into a header file.

To use SCAMP, a weight image is first built for the science images by displaying each science image with Ds9 and creating regions covering any areas of the image not usable for science, like the corners and the guider region. Then a homebrew script, Appendix B, can be used to build the weight images for each science image.

Next, an approximation of the WCS is needed to compare the output from SExtractor to the reference catalog, which is obtained by the RA and DEC keywords in the fits header. The RA and DEC keywords come from the location at which the telescope is pointing at the time of the observation. The keywords are sent through a C++ script which filled in the WCS information for each science image. On runs 3, 4, and 5, there was an offset between what was recorded in the fits header and the reference catalog due to precession. This is due to the RA and DEC keywords recorded in the fits header for the epoch of the run, while the reference catalog is converted to J2000 coordinates. Since sources will move over time due to the precession of the Earth, and SCAMP reads the WCS coordinates as J2000 for both catalogs, the separation distance between the two epochs became large enough for SCAMP to fail. To overcome this problem, on runs 3, 4, and 5 the IRAF task `precess`, an example in Appendix A.6 was used, which allows the WCS coordinates in the header to be converted from the observing epoch to J2000.

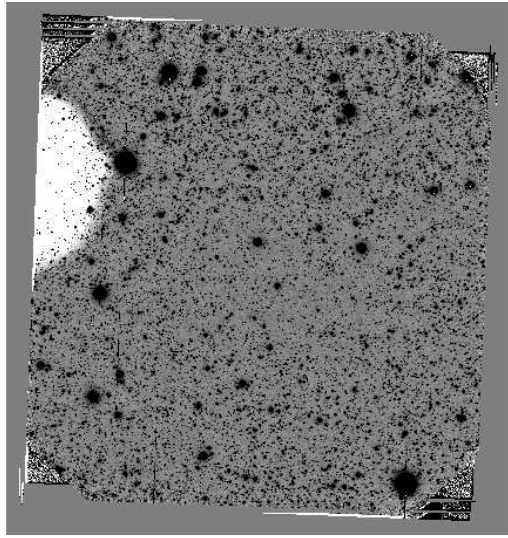
Then, SExtractor is used to calculate the coordinates of sources on each image by running a configuration file, as seen in Appendix C.1. Since Scamp requires the input cata-

log to be in LDAC format, the `CATALOG`TYPE` parameter is set to `FITS`LDAC`. Also, SCAMP will fail if there are many mis-matched sources between the two catalogs. To prevent this, only sources with high SNR should be detected by SExtractor, which was done by setting `DETECT`THRESH` and `ANALYSIS`THRESH` to 20.0. `FILTER`NAME` is set to the correct SExtractor convolution file based off what the FWHM of the image is and using a Gaussian, or for crowded fields Mexhat. `WEIGHT`IMAGE` was set to the name of the weight image corresponding to the image SExtractor will be run on. Scamp also needs SExtractor to output values for certain parameters, which can be found in Appendix C.1.1.

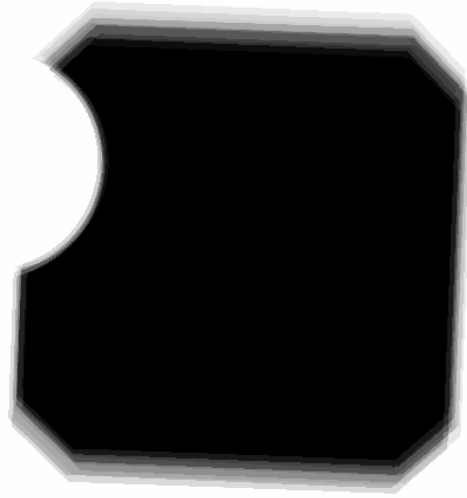
After running SExtractor, SCAMP ran by using a SCAMP configuration file, an example is found in Appendix C.2, and the SExtractor output catalog. In the configuration file, the reference catalog was chosen by setting the `ASTREF`CATALOG` to `SDSS-R7`, which is the SDSS DR7 data. `MOSAIC`TYPE` is set to `FIX`FOCALPLANE`. Since the coordinates from SExtractor are x and y coordinates, there is a possibility of the reference catalog being at a different orientation than the SExtractor output, which can be overcome by setting the `PIXSCALE`MAXERR` parameter to 2.0, `POSANGLE`MAXERR` to 180.0 and `POSITION`MAXERR` to 3.0. After running SCAMP, the output gives the new header with the corrected WCS information for each image.

4.1.6 SWarp

After calculating the corrected WCS by SCAMP, the stacking of science images was performed by SWarp. SWarp resamples and co-adds fits images using any arbitrary astrometric projection defined in the WCS standard. SWarp was performed on each set of science images using the default SWarp configuration file, an example is found in Appendix C.3. The list of the science images was used as input as well as their respective weight image and the



(a)



(b)

Figure 4.6: Example of the output from SWarp. a) the stacked science image. b) the stacked weight image.

header information calculated from SCAMP. The output from SWarp is the stacked image, an example is in Figure 4.6a, and the stacked weight image, an example is in Figure 4.6b.

Chapter 5

Photometry

5.1 Modifying The Weight Image

When SWarp builds the weight image, the value of each pixel is based on how many images were used in the stack at that particular pixel. Since each image in the stack contains an OFCORS, the OFCORS will fall in the area of the science image that corresponds to the weight image with the highest pixel values. This uneven weight image causes a couple of issues when calculating the zeropoint during photometry. First, SExtractor will make false detections in regions of the science image that correspond to the regions with the non-highest pixel values in the weight image, which can cause a mismatch with a reference catalog, which in turn could cause the wrong zeropoint value to be calculated. Second, if the source was real, then the photometry in the regions with the non-highest pixel values in the weight image will not be as accurately determined by SExtractor as the rest of the image, which can cause a higher error in the zeropoint calculation.

Therefore, the weight image needs to be modified so SExtractor will only be run on the sources that fall in region with the highest pixel values of the weight image. This was done in two steps by a script, as seen in Appendix D.1. First, the maximum value of the weight

image needed to be determined using Ds9 to visually inspect the weight image. Second, we recorded the maximum value of the weight image and put it into the `weightfix.dat`, as seen in Appendix D.1.1. Then using the IRAF task `imcalc`, a new weight image is created by setting all pixels equal to the maximum value of the weight image to 1 and everything else to 0.

5.2 Initial Photometry

Next, the initial photometry was performed on each science image using the script in Appendix D.2, which modifies and runs SExtractor and outputs a region file to check the output. For SExtractor, a configuration file is needed, an example can be found in Appendix D.2.1 and the parameter file in Appendix D.2.2. Starting with a default configuration file several of the parameters were modified for the OSMOS images. Since the OFCORS are invisible in SDSS, a low threshold is needed to make sure that if they do show up, it will not be missed so the `DETECT_THRESH` and `ANALYSIS_THRESH` parameters are set to 0.5. The convolution file was chosen by looking at the FWHM of the sources by using the IRAF task `imexamine` and determining it to be about 5. Therefore, the `FILTER_NAME` parameter was set to `gauss_5.0_9x9.conv`, which is a precreated convolution file by SExtractor. For crowded fields, a gaussian convolution fails, so instead a mexhat convolution was used, so the `FILTER_NAME` parameter would be set to `mexhat_5.0_11x11.conv`. The `SATUR_LEVEL` parameter is set to 65000. The `GAIN` parameter is obtained from the header of the stacked image. The `SEEING_FWHM` parameter was calculated by taking the `FWHM+1` to get 6 and multiplying by the plate scale of OSMOS of $0.273''/\text{pixel}$ to get 1.65. Then SExtractor uses this configuration file to perform the photometry.

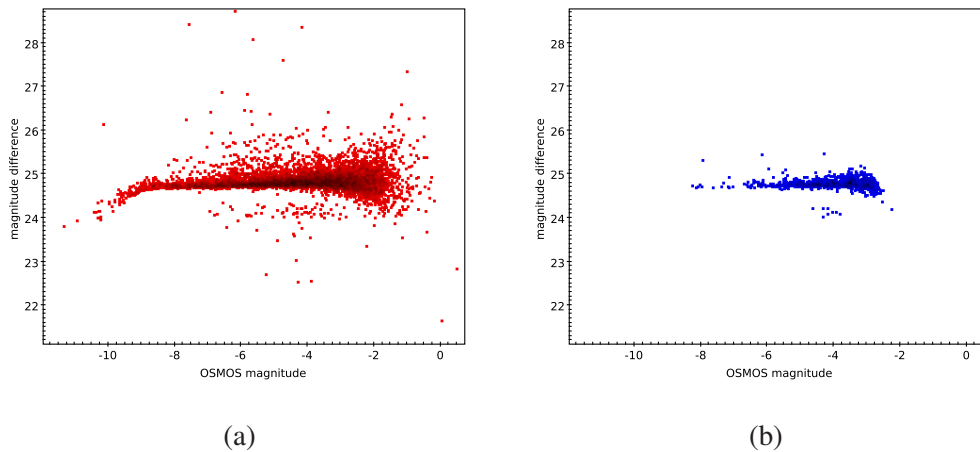


Figure 5.1: magnitude difference between OSMOS and SDSS vs OSMOS magnitude for J2117+5431 r band. a) all of the sources matched between SDSS and OSMOS. b) sources left after cuts

5.3 Zeropoint Calculation

In the previous section, SExtractor was used to calculate the initial photometry, but to get the correct photometry, the zeropoint is needed. The zeropoint can be calculated by comparing photometry for a science image with a reference catalog, in this case SDSS DR10. The SDSS DR10 catalog was downloaded from the SDSS archive and contains the magnitude and magnitude error for sources in all five bands within 15' radius of each OFCORS. Then, the zeropoint was calculated by the script Appendix D.3.

First, a cross match is performed between the two catalogs using TOPCAT, a software that is designed for astronomers to perform analysis and manipulation of source catalogs and other tables, using a matching radius of 3" to create a matched catalog. Next, the magnitude difference between the two catalogs was calculated, making sure to use the same SDSS band as OSMOS data. The matched catalog with the calculated magnitude difference is saved to be used as input in a zeropoint calculation code.

The zeropoint calculation code, as seen in Appendix D.3.1, calculates the average of the magnitude difference, which will be the zeropoint. However, there is a large spread in the magnitude difference at the lower and higher magnitudes, which can be seen by plotting the magnitude difference vs OSMOS magnitude in Figure 5.1a. So, a few cuts are needed to make the zeropoint calculation more accurate. First, only stars should be used, however the combined catalog is made up of both stars and galaxies. To only choose stars, sources with the parameter `type=6` in the the SDSS catalog were used. Second, saturated sources in either SDSS or OSMOS caused the photometry not to be calculated correctly. To account for this, the flag parameter from SExtractor was used, where `flag=0` represents no saturation. The third contamination comes from the fainter sources in the SDSS having a low signal-to-noise (SNR) ratio, which causes the photometry not to be as accurate as sources with higher SNR ratio. This will cause the magnitude difference between the two catalogs to be skewed. Therefore, only SDSS sources with a SNR greater than 10 were used in the zeropoint calculation by determining if the magnitude error of each source was smaller than a SNR of 10. Figure 5.1b shows what sources are left over after the cuts.

After the cuts, the average magnitude difference and dispersion were calculated using a C++ code in Appendix D.3.1. In the code, the aforementioned cuts are made, then the average magnitude difference and dispersion are calculated. Then, there is a possibility of some outliers still existing so a clipping algorithm is needed. This clipping algorithm ran for 30 iterations and each iteration threw out any sources that fell outside of the range determined by the average magnitude difference ± 2 times the dispersion, and recalculates the average magnitude difference and dispersion. The final average magnitude difference is the zeropoint for the image. Figure 5.2 shows an example of the zeropoint calculation for J2114++5431 in r band for what sources are kept after the zeropoint calculation, and

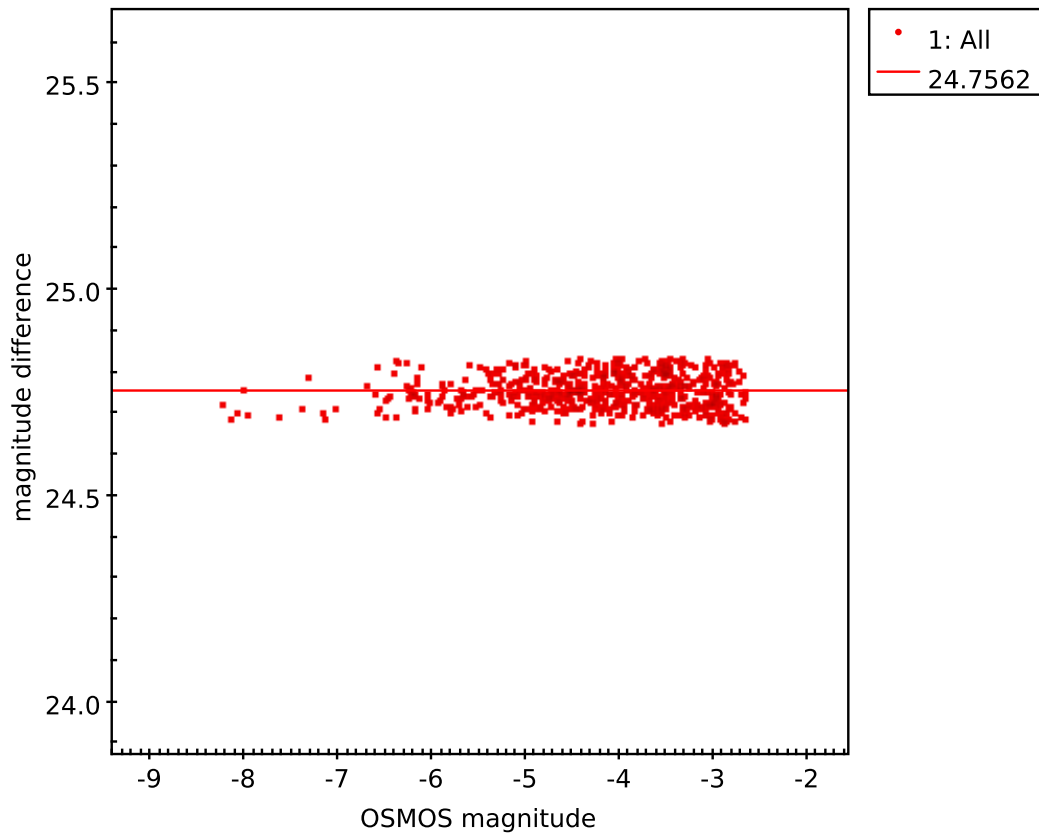


Figure 5.2: Final zeropoint calculation for J2114++5431 r band. The points are what was left over after the zeropoint calculation. The line represents the final zero point value.

the line represents the final zeropoint used.

5.4 Photometry

Once the zeropoint was calculated, the `MAG_ZEROPOINT` parameter in the `SExtractor` configuration file for each stacked image is changed to the zeropoint, see Appendix D.2.1. Then, `SExtractor` was rerun using the new configuration file to get the final photometry.

Name	Filter	Magnitude	Magerr
J0210-0737	r	23.1813	0.1172
J0418+5457	r	23.4706	0.056
J0418+5457	i	22.3222	0.0398
J1110+6028	r	24.0994	0.2303
J1204+3023	r	23.9677	0.2057
J1254+1843	r	23.9814	0.1465
J1254+1843	i	23.0576	0.0843
J1320+5036	r	24.2438	0.1236
J1320+5036	i	23.9919	0.138
J1434+4236	z	23.916	0.2306
J1439+2114	z	22.5624	0.147
J2210+2013	g	21.2177	0.0131
J2210+2013	r	20.7411	0.0135

Table 5.1: Sources and filters that OFCORS were detected in OSMOS

5.5 Source Identification

Next, to determine if the OFCORS source showed up, and if so what is its magnitude, the output catalog from SExtractor is crossmatched by the OFCORS coordinates using TOPCAT and a 3'' crossmatching radius. To make sure there was no false or missed detections, each image was visually inspected. Table 5.1 lists the magnitude and magnitude error for each source detected by filter.

5.6 Upper Limit

Even though deeper images than SDSS were obtained using OSMOS, not all of the OFCORS were detected. In these images, the upper limit was calculated to determine the faintest source that can be detected in the image. To calculate the upper limit, the photometry needs to be performed in areas with no sources, and looking at the fluxations in the background by using the script found in Appendix D.4.

The first step is to determine random places on the image to perform the photometry by using a random number generator using a C++ code, found in Appendix D.4.1. This random number generator works by the user inputting the minimum and maximum x and y pixels' coordinates to determine the area that the upper limit determination will be done on the image. Then the random number generator creates a catalog of 1000 sets of x and y coordinates.

Next, using the coordinates from the random number generator, IRAF task phot was used, which performs aperture photometry using predefined coordinates. In the phot parameter file, as seen in Appendix D.4.2, the image and the coords parameters are set to stacked science image and the catalog created by the random number generator, with the output set to the output catalog. Phot also has several sub-parameter files. In datapars, Appendix D.4.2, the scale parameter is set to 1, the FWHM parameter is set to the FWHM of the sources in the science image, readnoise was set to 5 which is the same as the OSMOS single image noise, the GAIN parameter is set to the value in the header, and itime is set to 1. In centerpars, Appendix D.4.2, the default parameters were used. In fitskypars, Appendix D.4.2, annulus parameter was changed to 15 and the salgorithm parameter was set to node. In photpars, Appendix D.4.2, the aperture parameter was set to $1.5 * FWHM$ and the zmag parameter was set to the zero point calculated during the zero point calculation step. After running phot, the IRAF task pdump was used on the output catalog to only grab the x and y coordinates, and the total number of counts in the aperture from the phot output catalog, and saves it into another file.

To calculate the magnitude upperlimit, the average of the total counts in the apertures and the dispersion can be calculated using a C++ code, Appendix D.5, which reads the output from pdump. To eliminate any outliers due to the random number generator choosing

Name	Filter	Upperlimit
J0037+0808	r	25.7054
J0559+2353	g	25.5030
J0559+2353	r	25.1638
J0559+2353	i	22.4511
J0947+5907	g	25.5698
J0947+5907	r	24.3347
J0947+5907	i	24.6775
J1204+3023	i	24.8069
J1220+6327	r	25.3930
J1220+6327	i	24.7422
J1220+6327	z	23.0735
J1434+4236	i	24.9145
J2117+5431	g	25.0496
J2117+5431	r	24.8610
J2314+5813	r	23.8529
J2314+5813	i	23.7183

Table 5.2: Upperlimit for the OFCORS observed by OSMOS

locations on the image with sources, a clipping algorithm was used where anything outside the range of twice the dispersion was thrown out and the average total number of counts and dispersion were recalculated. The magnitude upperlimit is then calculated by using the dispersion in the equation

$$upperlimit = (-2.5) \log_{10}(2 * dispersion) + zeropoint \quad (5.1)$$

where 2 times the dispersion was used due to the clipping algorithm. Table 5.2 lists the upper limits calculated and which filter.

Chapter 6

SED fitting

By looking at the spectral energy distributions (SED), the nature of different sources can be explored. This chapter will discuss how SED fitting was performed on the data using the SED fitting software, LePhare.

6.1 Creating the catalogs

To build the SED's for the OFCORS, a master catalog needs to be built containing all data points from optical to the mid-IR. The WISE catalog was chosen for the mid-IR, while UKIDSS and VISTA catalogs were used for the near-IR. For the optical wavelengths, OSMOS, CFHT and Pan-STARRS data was used; however, since each instrument's filters have a different response function, a separate master catalog for each of the optical instrument must be built. The following subsections describe how each catalog was built.

6.1.1 OSMOS+UKIDSS+WISE

For the OSMOS based catalog, data from OSMOS was used in the optical range, UKIDSS and VISTA data was used in the near-IR range, and WISE data was used for the mid-IR.

The final catalog contains the magnitude and magnitude error for OSMOS g,r,i,z bands, UKIDSS and VISTA's Y,J,H, and K bands, and WISE's W1, W2, W3 and W4 bands, as well as an ID number that represents each OFCORS. Although not all of the OFCORS have OSMOS data, all of the OFCORS were run through LePhare using this input catalog. This allows us to explore how only sources with no optical detections are affected by SED fitting.

6.1.2 CFHT+UKIDSS+WISE

For the CFHT based catalog, CFHT data was used in the optical range, UKIDSS and VISTA data was used in the near-IR range, and the WISE catalog was used in the mid-IR range. The final catalog spans 13 bands which includes CFHT's u,g,r,i, and z bands, UKIDSS and VISTA's Y,J,H and K bands, and WISE's W1, W2, W3 and W4 bands, as well as the ID number that represent each OFCORS. For the CFHT based catalog, only sources with CFHT detections were run through LePhare. This is due to anything without CFHT detections are expected to give the same results as the OSMOS based catalog.

6.1.3 Pan-STARRS+UKIDSS+WISE

For the Pan-STARRS based catalog, Pan-STARRS data was used in the optical range, UKIDSS and VISTA data was used in the near-IR range, and the WISE catalog was used in the mid-IR range. The final input catalog spans 13 bands, which includes Pan-STARRS g, r, i, z, and y bands, UKIDSS and VISTA's Y, J, H, and K bands, and WISE's W1, W2, W3, and W4 bands, as well as the ID number that corresponds to each OFCORS.

6.2 LePhare

LePhare is a set of Fortran programs which uses the SED fitting technique to compute photometric redshifts. It allows for the user to select the SED models used, select a set of filters, and to compute the template magnitudes. Then LePhare will derive the best fit photometric redshift and stellar properties by matching the observed magnitudes to the theoretical ones using the χ^2 method.

A parameter file, a set of templates, and set of filters responses are needed to run LePhare. The parameter file governs what parameters are needed to make LePhare run and can be seen in Appendix E. The galaxy templates used are the BC03 templates at solar metallicity, where solar metallicity was chosen to cut down on the number of degrees of freedom. The FILTER_LIST parameter, which tells LePhare what filters and instruments are used, is set to a list of the filters in the input catalog. The EXTINC_LAW parameter, which tells LePhare what kind of extinction law to use, is set to a Calzetti extinction law. The EB_V parameter ranges from 0 to 2 in steps of 0.1. The output of LePhare is best fit to each OFCORS source from the input catalog. A java program, Paps.jar, is provided by LePhare to visualize the SED fitting result including the fitted properties like redshift and stellar mass.

6.3 Output

6.3.1 OSMOS+UKIDSS+WISE

After building the SED's with LePhare, a common theme became noticeable with the fits. Many of them are unreliable due to a high degree of freedom from lack of data points. Many

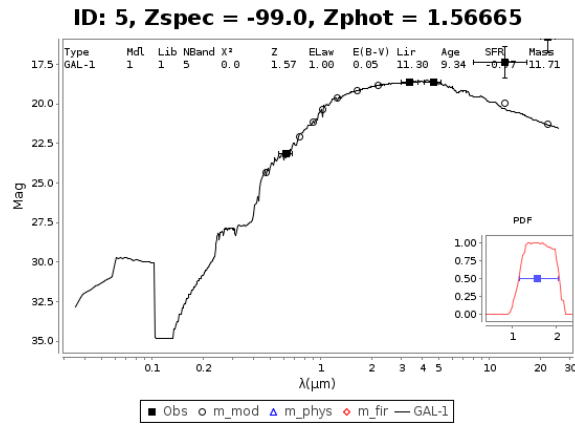


Figure 6.1: SED for J0210-0737 using OSMOS for optical

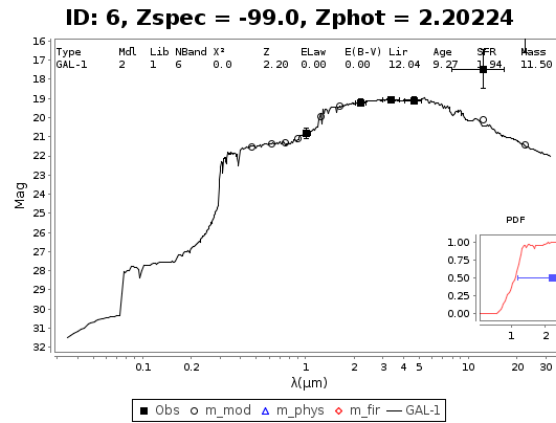


Figure 6.2: SED for J0253+0024 using OSMOS for optical

of the sources only have W1 and W2 detections, which is not enough data to constrain the SEDs. Therefore, a restriction was placed on what bands had to have data in the input catalog to have a good fit: the source must have W1 and W2 detection to constrain the mid-IR end, and at least 1 detection in the OSMOS bands to constrain the optical end, or 2 detections in the middle to constrain the near-IR regime which should help to constrain the optical end. This left us with 10 sources whose LePhare SED outputs are shown in Figures 6.1 through 6.10.

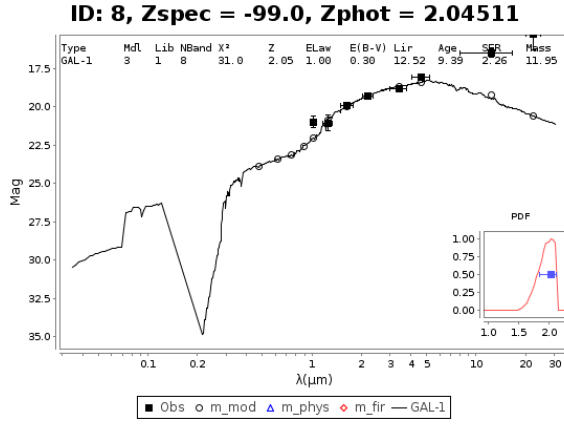


Figure 6.3: SED for J0305+0523 using OSMOS for optical

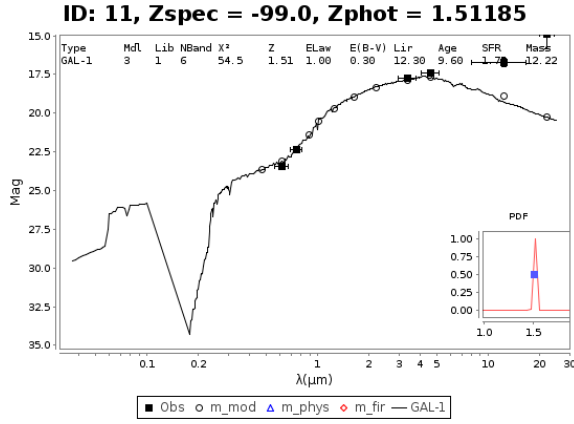


Figure 6.4: SED for J0418+5457 using OSMOS for optical

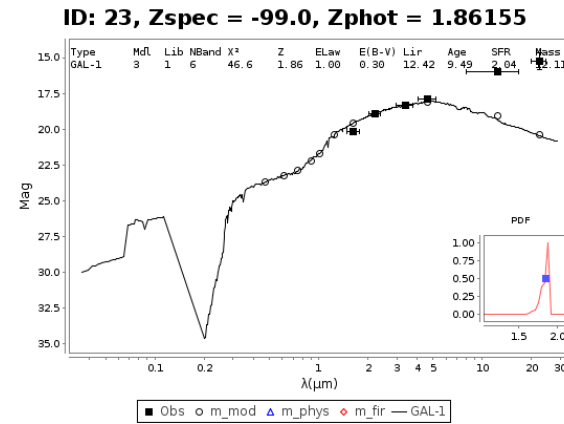


Figure 6.5: SED for J0805+2106 using OSMOS for optical

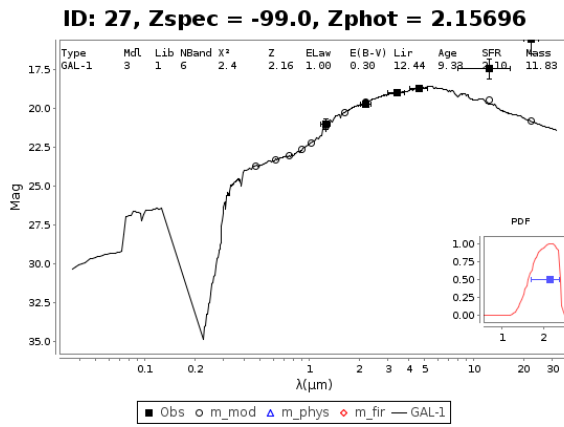


Figure 6.6: SED for J0904+2729 using OSMOS for optical

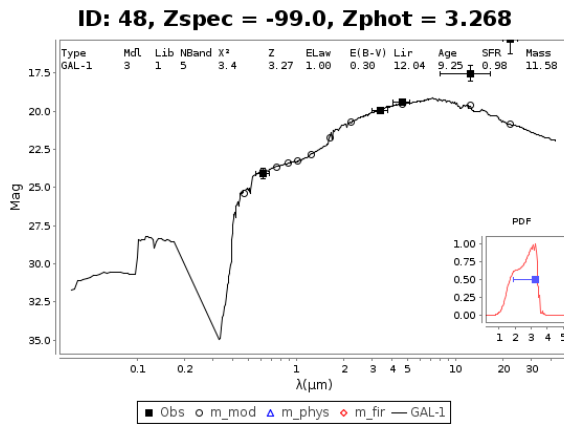


Figure 6.7: SED for J1110+6028 using OSMOS for optical

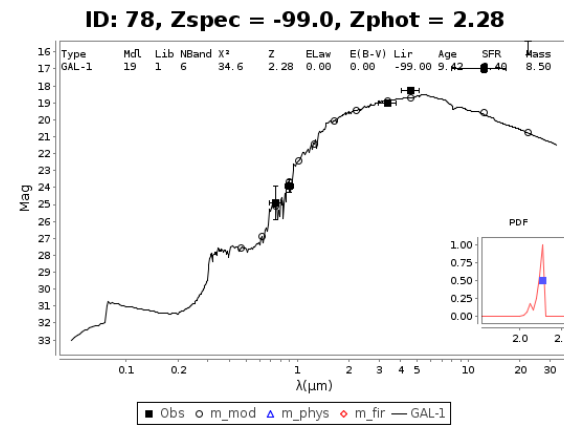


Figure 6.8: SED for J1434+4236 using OSMOS for optical

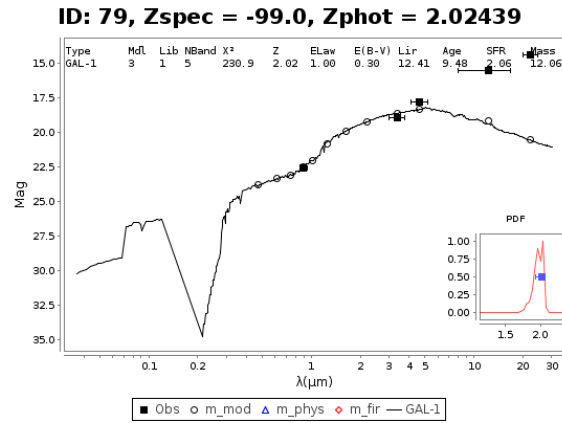


Figure 6.9: SED for J1439+2114 using OSMOS for optical

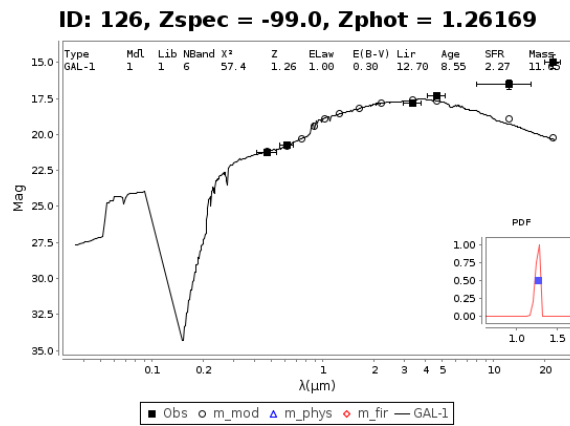


Figure 6.10: SED for J2210+2013 using OSMOS for optical

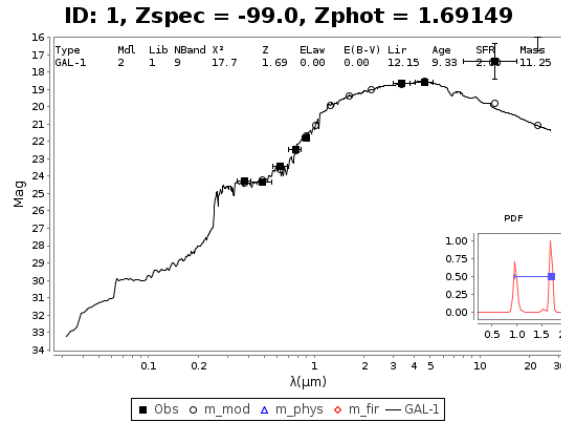


Figure 6.11: SED for J0210-0737 using CFHT for optical

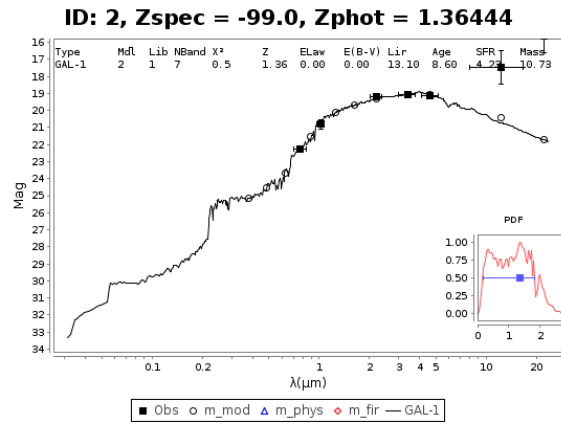


Figure 6.12: SED for J0253+0024 using CFHT for optical

6.3.2 CFHT+UKIDSS+WISE

After running LePhare on the 3 CFHT based sources, the output is showing in the Figures 6.11 to 6.13. Comparing these three OSMOS based SEDs with the CFHT based SEDs, it can be seen that the redshift and stellar mass values are widely different for the same source. This difference is a result on the degree of freedom that is not as well constrained due to the lack of data points in the fit.

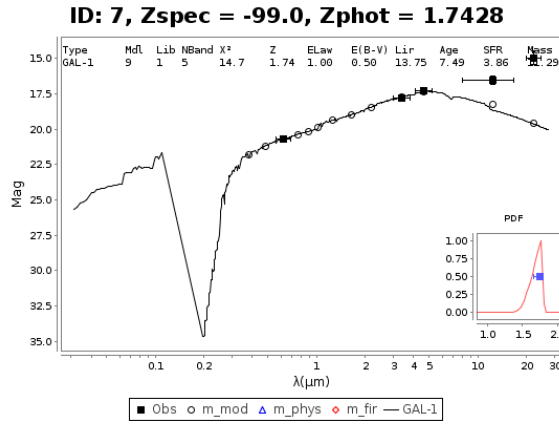


Figure 6.13: SED for J2210+2013 using CHFT for optical

6.3.3 Pan-STARRS+UKIDSS+WISE

Only 6 of the 7 Pan-STARRS detected OFCORS were run through LePhare. The remaining source did not have a detection in the UKIDSS or WISE catalog. Two of the sources were detected in both Pan-STARRS and OSMOS or CFHT. Figures 6.14 through 6.19 is the output from LePhare. Comparing the two sources that were detected in both Pan-STARRS and OSMOS or CFHT, it can be seen that redshift and stellar mass values are widely different for the same source. This difference is a result on the degree of freedom that is not as well constrained due to the lack of data points in the fit.

6.4 Plots

LePhare determined the redshift and stellar mass of each OFCOR during the fitting. Figure 6.20 shows the redshift distribution, while Figure 6.21 shows the stellar mass distribution of the OFCORS. If a source was run through LePhare in more than one input catalog, only those with the most photometry data points was used, tightening the degrees of freedom. From the plots, LePhare determined the OFCORS have a redshift between 1 and

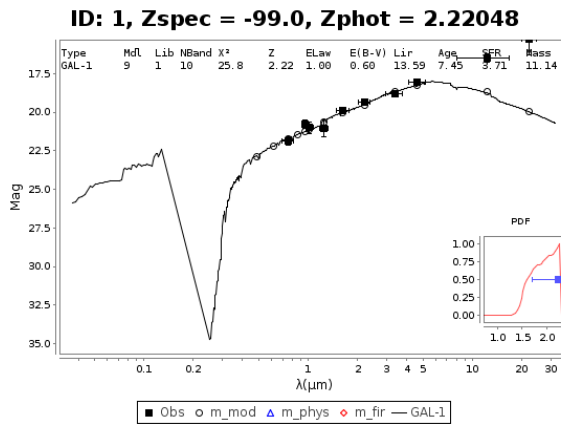


Figure 6.14: SED for J0305+0523 using Pan-STARRS for optical

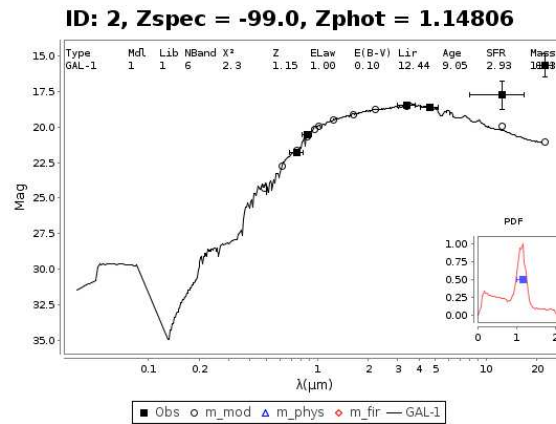


Figure 6.15: SED for J1152+3327 using Pan-STARRS for optical

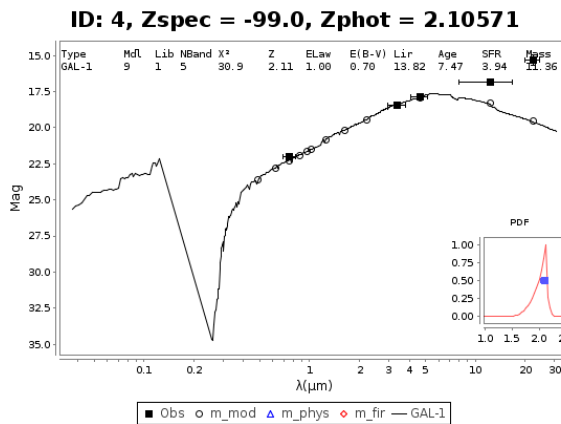


Figure 6.16: SED for J1735+5049 using Pan-STARRS for optical

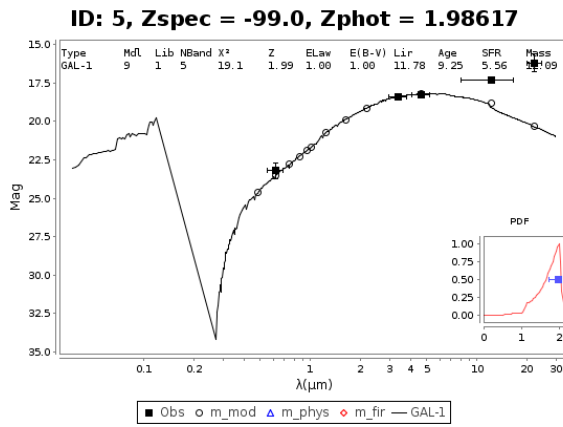


Figure 6.17: SED for J1746+6421 using Pan-STARRS for optical

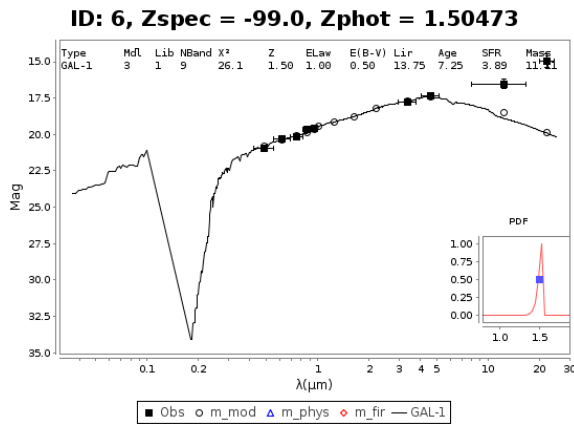


Figure 6.18: SED for J2210+2013 using Pan-STARRS for optical

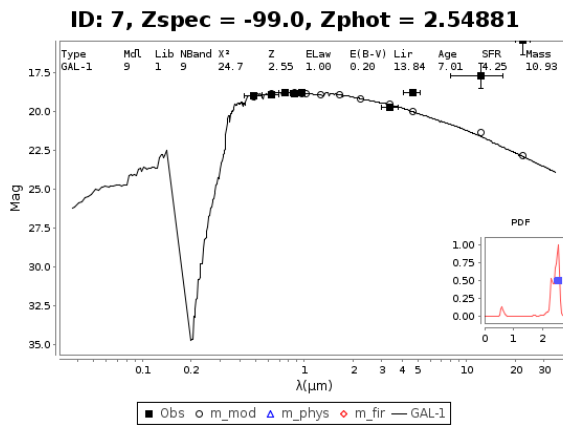


Figure 6.19: SED for J2219+2141 using Pan-STARRS for optical

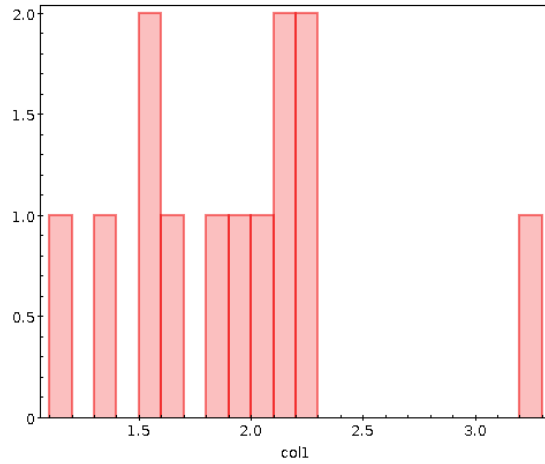


Figure 6.20: Redshift distribution from LePhare

2.5, and stellar masses of $10^{11-12}M_{\odot}$. The outliers in both plots are for different sources. Based off the stellar masses, the OFCORS would be some of the most massive galaxies in the universe. However, the stellar masses greater than $10^{12}M_{\odot}$ may not be reliable because the most massive galaxies that have been detected have been around the $10^{11}M_{\odot}$ range. These high values for stellar masses are most likely because there is not enough data points to fully constrain the SED fit and the degrees of freedom.

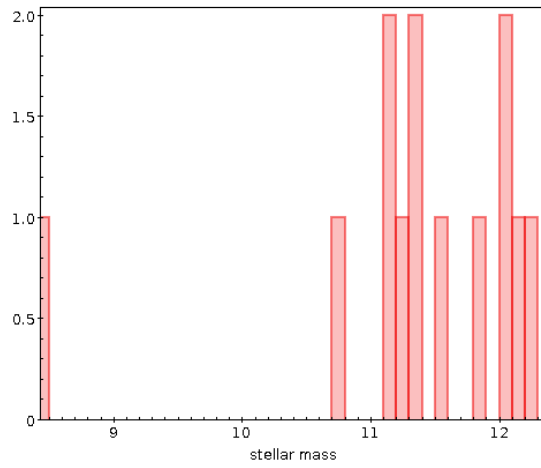


Figure 6.21: Stellar mass distribution from LePhare

Chapter 7

Infrared relations and power laws

Stellar mass and redshift of the OFCORS can be approximately determined from the SED fitting. However, due to the lack of data points for many of the OFCORS, only a small number of SED's could be fit. This causes the interpretation of the stellar mass and redshifts for these sources to suffer from small scale statistics. To remedy this, different routes can be used to try to infer the properties the sources. In this chapter, properties of the sources in the infrared regime are explored while in the next chapter the radio regime will be explored.

7.1 K-z relation

For radio galaxies, it has been known for many years that there exists a relationship between K band magnitude at $2.2 \mu m$ and redshift. An example can be seen in Figure 7.1 created by plotting the data from Willott's (2003) [36], Willott03 here after, where they observed multiple radio sources in the K band as well as calculated the redshifts from spectra. Therefore if radio galaxies follow the K-z relationship, then knowing the K band magnitude of the OFCORS, the redshift can be inferred.

A check to see if the entire RFC catalog followed the K-z relationship was first per-

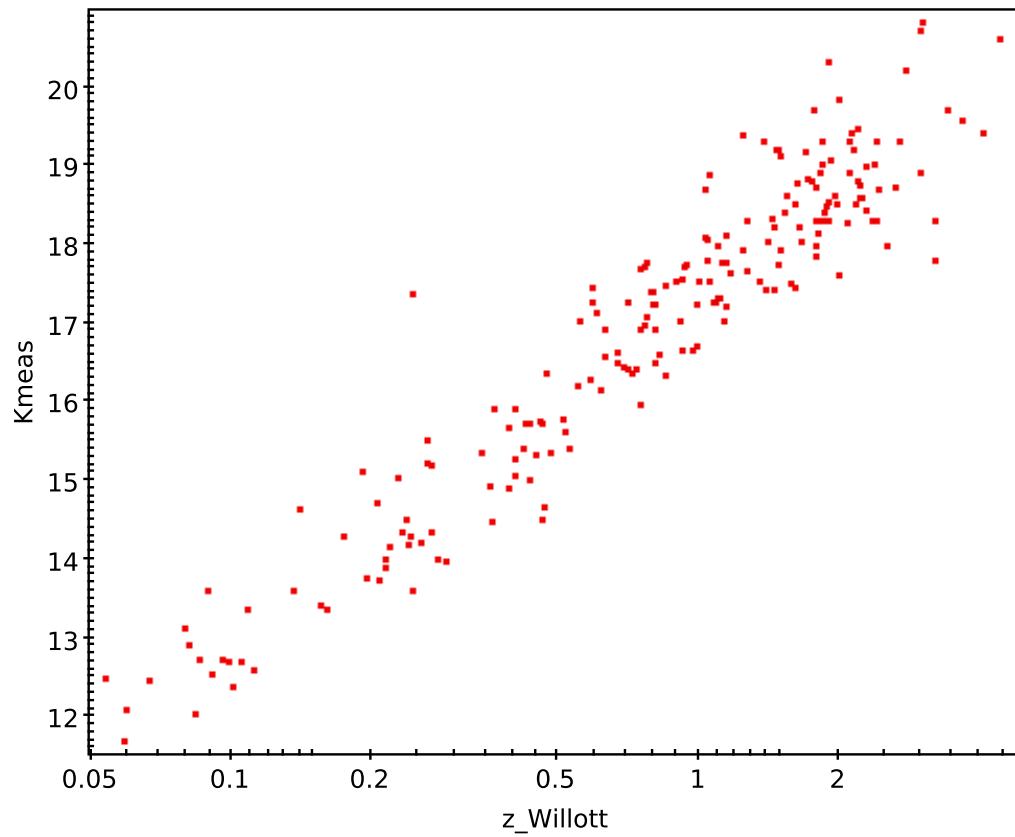


Figure 7.1: K-z relation determined by Willott 2003.

formed by using the RFC catalog version 2016b which contained 11000 sources. This version of the catalog was chosen over version 2012c, which was used to define the OFCORS, to check the infrared and radio properties because it had 4000 more sources, which helps to decrease the effect of small scale statistics. For this chapter and the next, version 2016b will be referenced when referring to the RFC catalog unless otherwise specified. The K band magnitudes of the sources in the RFC catalog were determined by crossmatching with the 2MASS catalog using a 3" matching radius. The redshifts for the sources in the RFC catalog were determined by crossmatching with SDSS DR10 using a 1" matching radius. Roughly 900 RFC sources had both K band data from 2MASS and redshift from SDSS. Looking at Figure 7.2, the red points represent the Willott03 data and the blue points represent the RFC data cross matched with 2MASS and SDSS. At low redshifts the RFC data follows the K-z relationship shown by the the Willott03 data; however at $z \sim 0.5$, the RFC deviates. This deviation of the K-z relation begs the question: if the RFC are radio galaxies, then why does the RFC not follow the K-z relation?

A hint of what could be causing the deviation can be observed by looking at the templates used by SDSS when they did the spectra analysis to determine the redshift. When spectra analysis was performed on the SDSS data, they used either an galaxy or QSO template, which can be found in the SDSS catalog as the class parameter. In Figure 7.3, separating the RFC sources by the class parameter reveals that at low redshifts the class parameter was set to galaxy, and at high redshifts the class parameter was set to QSO. This separation by redshift shows a sign of 2 different populations in the RFC and a possible cause behind the deviation in the K-z relationship for the RFC sources.

Another issue that could be causing this deviation is 2MASS is not that deep. Therefore at fainter magnitudes, the uncertainty is higher. Also none of the OFCORS were detected

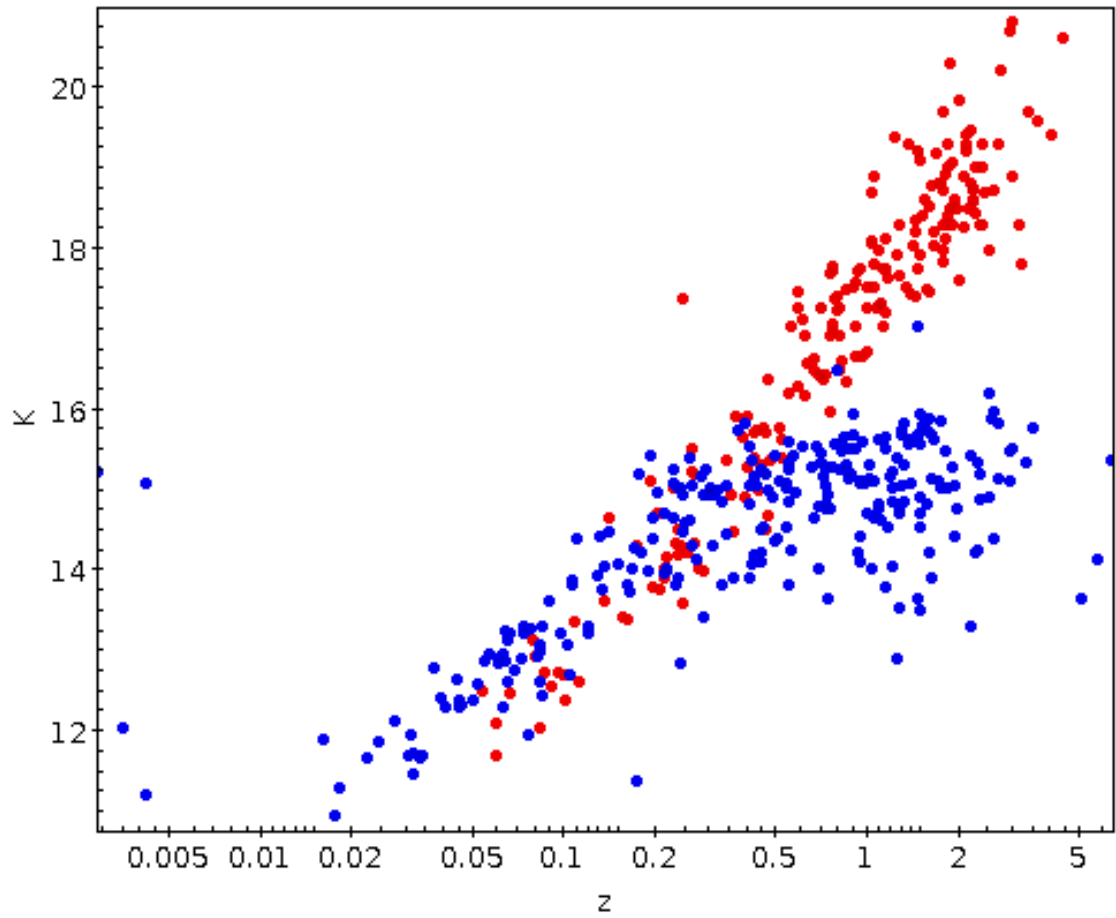


Figure 7.2: K-z relation comparing Willott 2003 to RFC sources matched with 2MASS. Red points are the Willott 2003 data and the blue dots represent the RFC matched with 2MASS sources

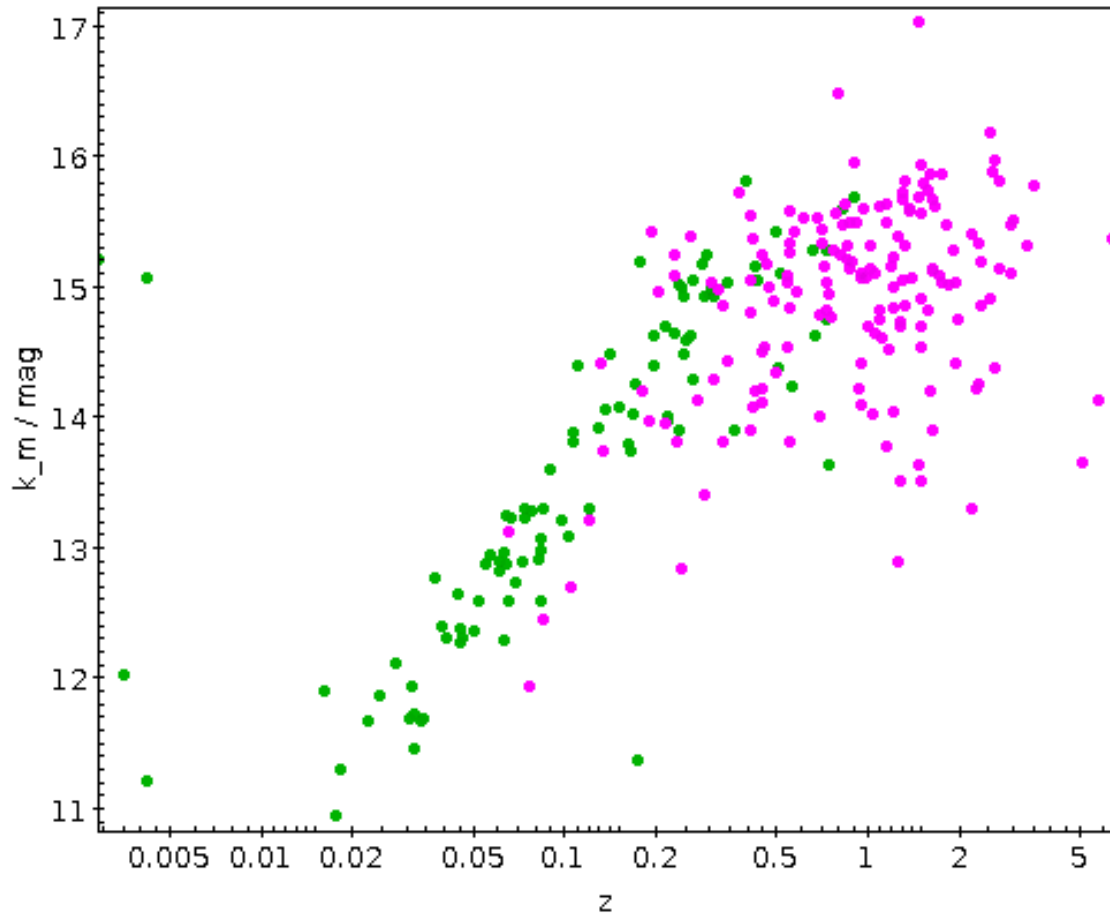


Figure 7.3: K-z relation for the RFC sources matched with 2MASS separated by the SDSS class parameter. Green dots represent when class is equal to galaxy and the purple dots represent when the class parameter is equal to QSO.

in 2MASS, which does not make it ideal to use with the K-z relation. To check any issues due to fainter magnitudes, a deeper survey in K band is needed. UKIDSS is ideal for this because even though it is not an all-sky survey, it is deeper than 2MASS and has a higher limiting magnitude. A cross-match was performed of the RFC with UKIDSS using a 3'' matching radius to get K band magnitude and with SDSS DR10 to get the redshift. In Figure 7.4, the spread at higher redshifts for the RFC is still visible. Furthermore, comparing the 2MASS and the UKIDSS data, the gap seen between the two data sets in Figure 7.2 at high redshift has been filled in. Therefore the deepness of the K band data does not seem to be the cause of the deviation from the K-z relation.

7.2 W1-z relation

Only 6 of the OFCORS were detected in K band by UKIDSS. Therefore, the issue of small scale statistics still arises. Then the question to ask here is by looking at different wavelengths would more OFCORS be detected, and a similar relationship with redshift be still arise. A good starting point would be with W1 band since 87 of the OFCORS were detected in this band. Since the wavelength of W1 ($3.4 \mu m$) is close to K ($2.2 \mu m$), there is a possibility that a similar relation between W1 and redshift as the K-z relation could be seen. Looking at the sources from Willott03, a W1-z relation can be built by performing cross-match of these sources with WISE using a 3'' cross-matching radius to get W1 magnitude. Figure 7.5 plots a W1 vs z for the sources from Willott03, and a tight correlation is visible. Therefore we can argue that there is a W1-z relation.

Next to check the W1-z relation for the RFC sources, a 3'' cross matching radius between the RFC and WISE catalogs was used to get the W1 magnitudes. A 1'' cross match radius between RFC and SDSS DR10 was used to get the redshifts. Figure 7.6, the RFC

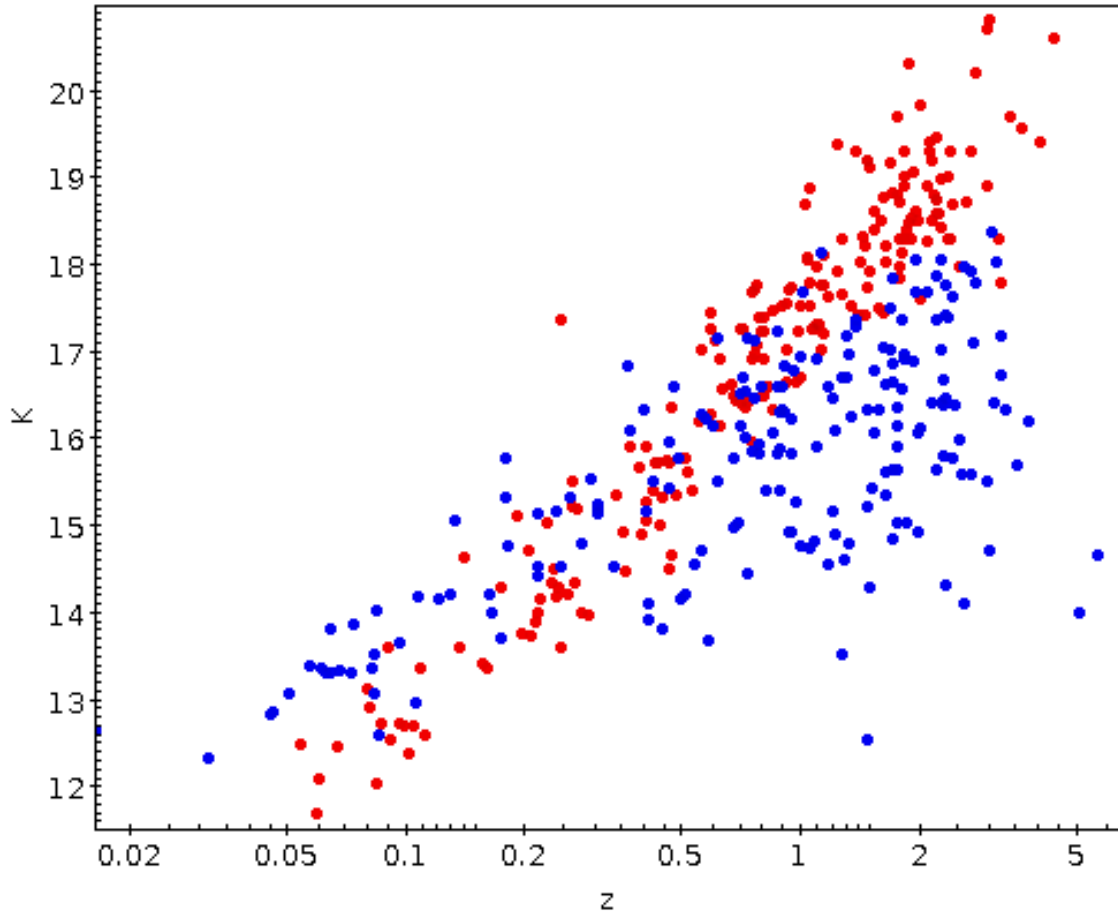


Figure 7.4: K-z relation comparing Willott 2003 to RFC sources matched with UKIRTs. Red points are the Willott 2003 data and the blue dots represent the RFC matched with 2MASS sources

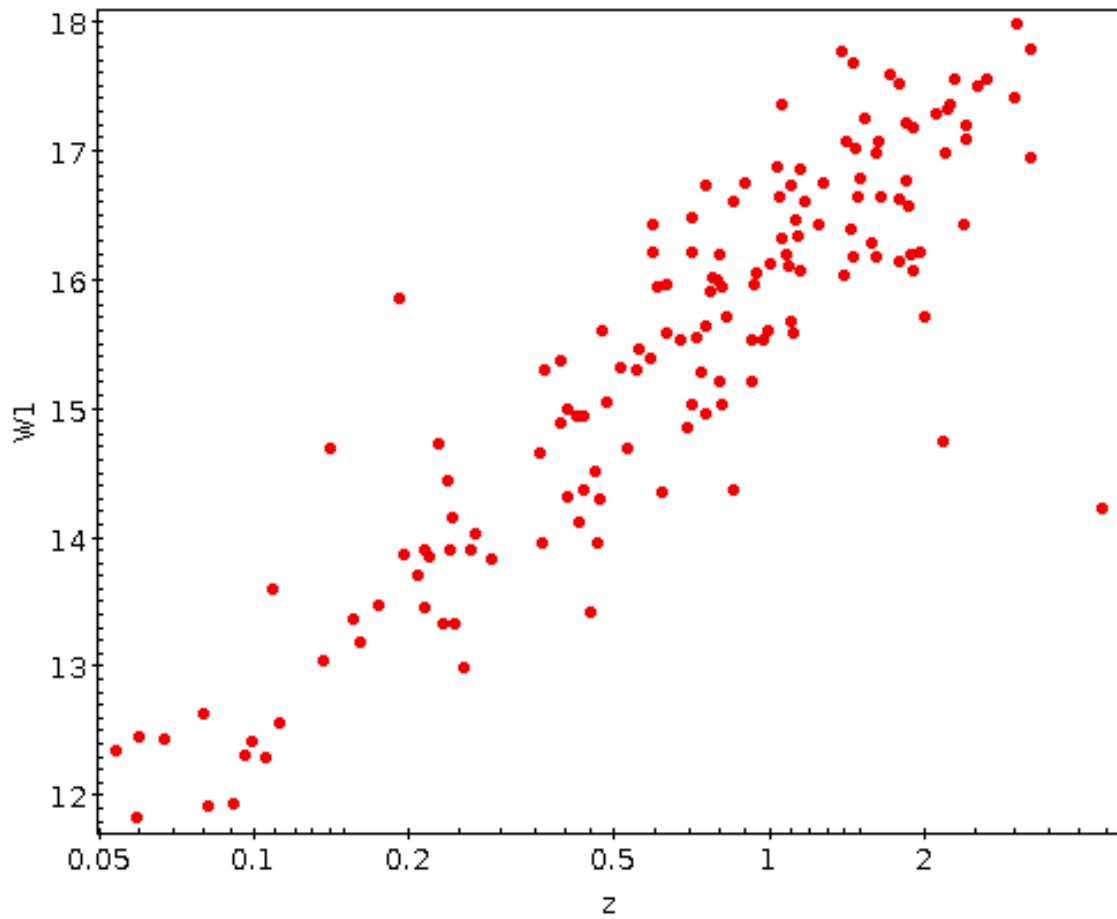


Figure 7.5: W1- z relation by cross matching Willott03 with WISE

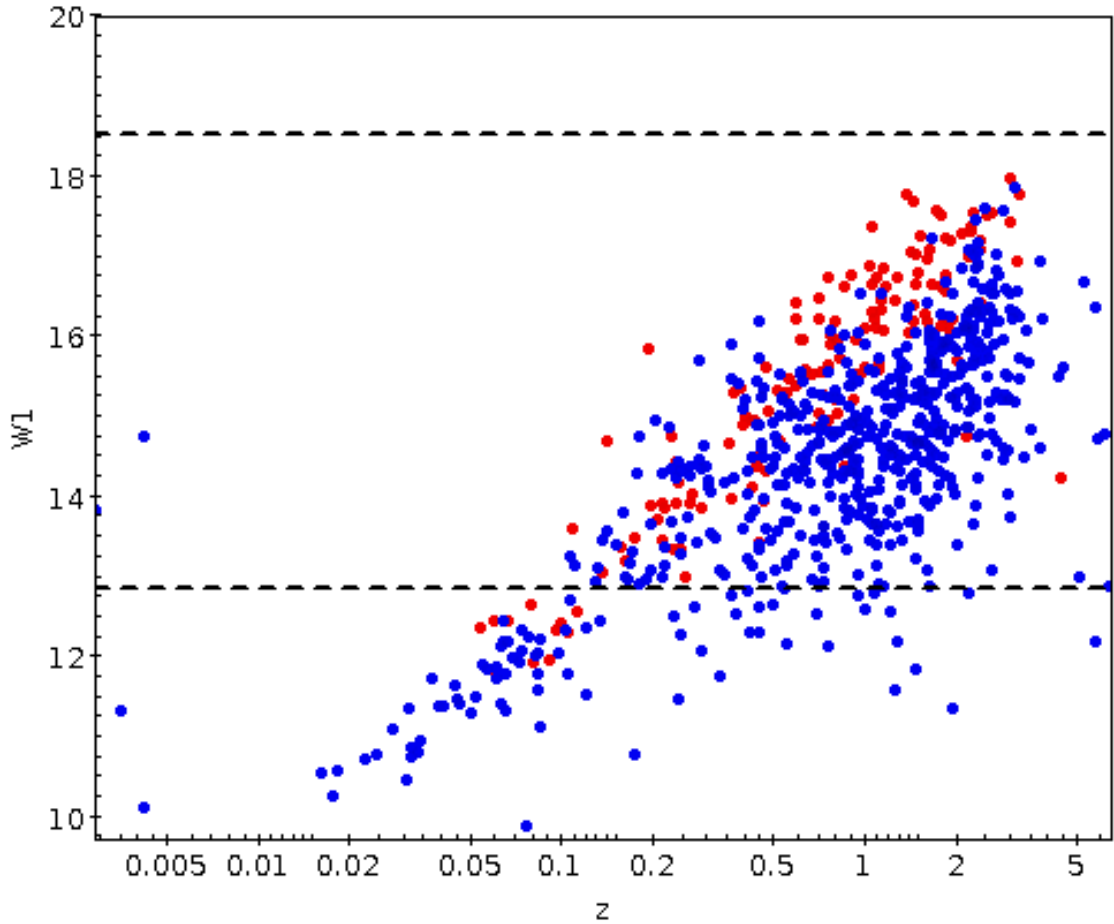


Figure 7.6: W1-z relation comparing Willott03 to RFC sources matched with W1. Red points are the Willott03 data and the blue dots represent the RFC matched with W1 sources. The dashed line represents the range of W1 detections of the OFCORS sources.

W1 vs. z data is plotted in blue and the W1 vs. z data from Willott03. Just like the K-z plot, the W1-z has a large spread at higher redshifts. Looking at the range of W1 for the OFCORS, represented by the black dashed lines, and comparing it to the RFC data points, it can be inferred that the redshift of these OFCORS $\sim 1 - 2$. Their bright radio flux densities imply they have a bright radio flux luminosity $\geq 10^{42-43} \text{ erg/s}$. For sources that are not detected, they most likely fall above this range. By comparing the trend of the RFC data, this implies that these sources could be at a higher redshift of $z > 4$.

7.3 IR Power Law

7.3.1 Motivation

For AGNs, the SED will be shaped close to a power law relating flux and frequency in the mid-IR. This power-law has the form

$$F_{\nu} = \nu^{-\alpha} \quad (7.1)$$

where F_{ν} is the flux at a particular wavelength, ν is the frequency, and α is the power-law index. The steepness of the power-law index does not mean anything physically for our purposes, what matters is whether a power law can be fit. To calculate the power law, the four WISE bands were used. To make sure there were enough points to fit a line, the power law was only calculated for the subset of OFCORS that were detected in all four WISE bands. Only 28 of the OFCORS fell into this subset.

7.3.2 IR power law calculation

To calculate the power law, the least square method was used. The input of this will be the wavelength of the WISE bands in μm , AB magnitude, and magnitude error. Then each of these are converted to observed frequency, ν_{obs} , observed flux, $F_{\nu,obs}$, and flux error by the following

$$\nu_{obs} = 3 * 10^8 / \lambda \quad (7.2)$$

$$F_{\nu,obs} = 10^{\frac{ABmag+48.6}{2.5}} \quad (7.3)$$

$$fluxerror = \frac{magerror}{2.5} \quad (7.4)$$

Next, the power law needs to be converted into a linear equation to use the least square fit method. To do this, both sides of the power law were multiplied by \log_{10} . This puts the power law into the form

$$\log_{10}(F_{v,obs}) = \log_{10}(constants) - \alpha * \log_{10}(v_{obs}) \quad (7.5)$$

where $F_{v,obs}$ represents the flux observed and v_{obs} is the frequency observed. Due to redshift, the flux that is observed is dependent upon the frequency observed. Moving the same source to a different redshift, will result in the flux at a particular frequency to change based off its SED. This means that the power law could have a redshift dependence that has to be accounted for when calculating α . Therefore, the power law was calculated for a range of redshifts from 0.05 to 6 in steps of 0.05. In each step the rest flux and rest frequency needs to be calculated. This is done by

$$v_{rest} = v_{obs} * (1 + z) \quad (7.6)$$

$$F_{v,rest} = F_{v,obs} / (1 + z). \quad (7.7)$$

Then for each redshift, the α and constant terms are calculated by the least square fit method with the form

$$y = a * x - b \quad (7.8)$$

where y is $Flux_{v,rest}$, a is α , x is v_{rest} , and b is *constants*. The a term can be calculated by

$$a = \frac{(n * xysum - ysum * xsum)}{(n * x2sum - xsum * xsum)} \quad (7.9)$$

while b is calculated by

$$b = \frac{(ysum * x2sum - xsum * xysum)}{(n * x2sum - xsum * xsum)} \quad (7.10)$$

where n is the number of data points. The $xysum$, $ysum$, $xsum$, $x2sum$, $y2sum$ are calculated by the following equations:

$$xsum = \sum v_{rest,i} \quad (7.11)$$

$$ysum = \sum F_{v,rest,i} \quad (7.12)$$

$$x2sum = \sum v_{rest,i}^2 \quad (7.13)$$

$$y2sum = \sum F_{v,rest,i}^2 \quad (7.14)$$

$$xysum = \sum v_{rest,i} * F_{v,rest,i} \quad (7.15)$$

To check the goodness of the fit, the χ^2 can be calculated by

$$\chi^2 = \sum \frac{(\log_{10}(F_{v,rest,i}) - (b + a * \log_{10}(v_{rest,i})))^2}{fluxerror} \quad (7.16)$$

The above calculations were done by using a C++ script as shown in Appendix F. This script calculates the values of a and b , and outputs the values used in the calculation at each redshift, the equation for the power law, and the script to plot the data and equation using gnuplot.

7.3.3 IR power law results

After running the IR power law code, it is found that all 28 OFCORS in the subset followed a power law, which means they are AGN. An example of this is found in Figure 7.7. It was also found calculating the value of α was not affected by redshift, instead redshift only affected the y intercept term. For comparison purposes, only plots for redshift of 0.05 was looked at. Table 7.1 list the values of α for each source and Figure 7.8 shows the distribution of α . From Table 7.1 and Figure 7.8, the value of α mainly ranges from 1 to 2.2 with some outliers. However the value of α in the IR regime does not tell us anything physically about the source so currently we are not sure what the significance behind this range is.

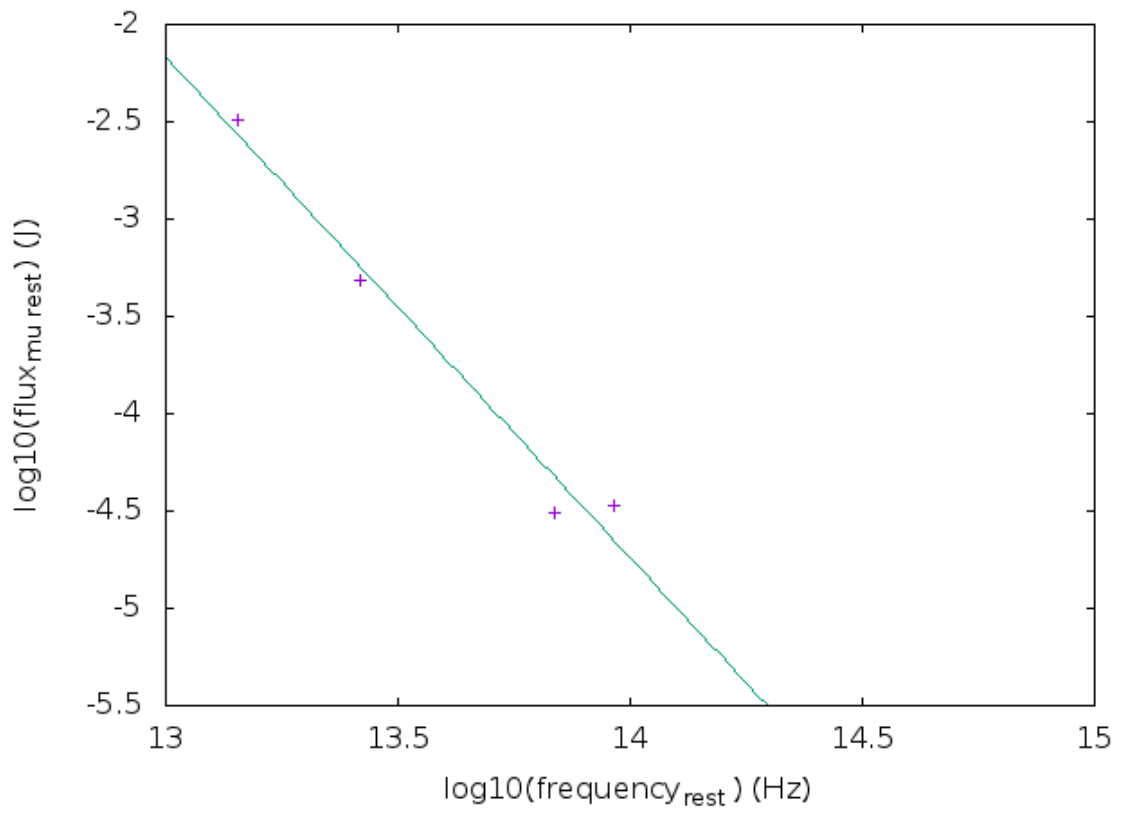


Figure 7.7: IR power law for J0158-1413 at redshift of 0.05

name	α
J0158-1413	2.573833
J0200-1542	1.486264
J0501+1356	1.740656
J0559+2353	1.322245
J0724-1505	2.076699
J0805+2106	1.582506
J0923+3849	1.646664
J1002+5701	3.007544
J1048+0141	1.771379
J1109+3744	1.943955
J1143+1834	1.734363
J1237+2034	1.907606
J1314+5306	1.906234
J1350+6428	1.378179
J1416+3444	2.036078
J1439+2114	2.195720
J1442+3042	1.568351
J1556+1825	1.666804
J1602+3326	1.464100
J1603+2126	1.480554
J1632+3547	1.143759
J1718+2948	1.617378
J1735+5049	1.453345
J1746+6421	1.057155
J1846+2036	1.523862
J1922+1530	1.147792
J2210+2013	1.294972
J2258+6325	1.825909

Table 7.1: value of α for each source

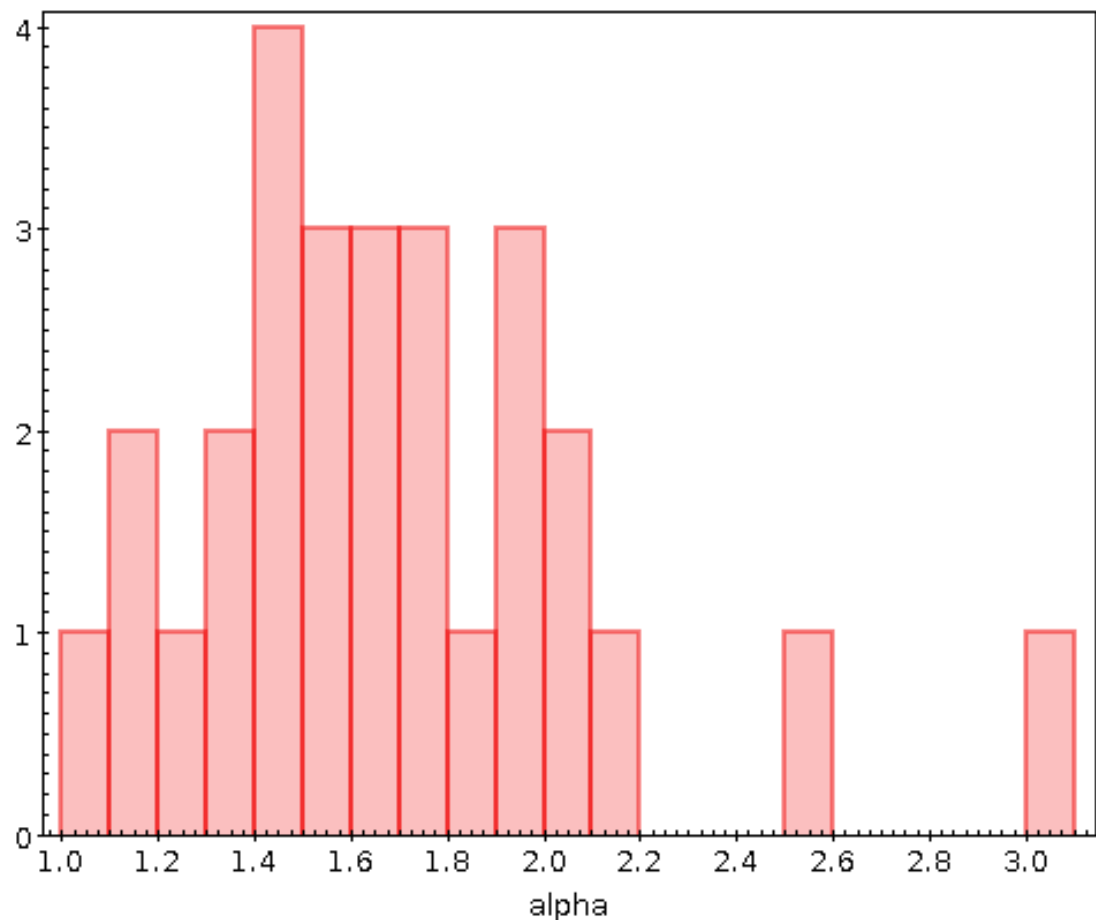


Figure 7.8: Histogram of α values from the IR calculation

Chapter 8

Radio power law

8.1 Motivation

For the radio regime, the SED is also shaped like a power law with the same form as the infrared power law. The α , for radio galaxies is called the spectral index, and gives a hint at what is powering the emission of the source. If the spectral index, is 0 to 2, then the source is powered by thermal emission. If the spectral index is less than 0, then the source is powered by synchrotron emission. The goal of this chapter is to determine what powers the OFCORS and if there is anything unique about what is powering this population compared to the overall RFC.

8.2 Building master catalog

To perform the power law calculation in the radio regime, the starting point was the RFC, which contains data for 5 bands. However for many of the sources, the sources have only been observed in one or two bands. Therefore more data in the radio regime is needed. This can be done by using the data from the URC which adds 4 more bands. Since the

coordinates of the URC are based off the FIRST and NVSS surveys, which have a spatial resolution $> 1''$, a cross matching radius of $3''$ was used between the two catalogs. From the match catalog, a subset catalog was made which only contained the OFCORS sources for comparison with the overall RFC. The final catalogs contained the source name, as well as the flux and flux error for each band. Any source that had no data for a band was given a value of -99.0.

8.3 Calculating the Radio Power Law

To calculate the radio power law, the least square method that was discussed in the previous chapter was used and was performed by using a C++ code, as seen in Appendix G, for calculating the radio power law using the combine URC and RFC catalogs. To make sure there was enough data points in the fit, only sources with at least 3 data points were used. Frequency for all bands were calculated in GHz. Since FIRST and NVSS are observed at the same frequency, only the band in which there were detections was used. If both bands contained a detection, the FIRST data was automatically chosen due to having a higher spatial resolution than NVSS. Since we found that changing the redshift did not affect the power law during the infrared power law determination, there was no redshift dependency added to the calculation of the spectral index for radio sources. The final output of the code was the source name, number of sources used in the calculation, and the value of the spectral index. The code also produced a script to create a plot using gnuplot as well as the equation for the best fit line.

8.4 Radio Power Law of the RFC

First, the radio power law code was run on all of the members of the RFC to calculate the value of the spectral index for each source. Overall 7464 out of the of 11000 sources in RFC catalog had enough data to be able calculate the spectral index. An example fit can be seen in Figure 8.1. In Figure 8.1, the normalized distribution of the spectral index for the RFC, in red, are plotted. From the histogram, it can be seen that the distribution for the spectral index for the RFC sources is roughly a Gaussian centered near -0.2. Based off the values of the spectral index, it can be argued that that the RFC sources are not entirely powered by synchrotron radiation or thermal emission exclusively, but more of a mixture of the two.

8.5 Comparision of the OFCORS the entire RFC

Next, only the spectral index of OFCORS subset was looked at. Only 124 of the 135 OFCORS sources had enough data points to perform the power law fit. The spectral index for the OFCORS sources is in Table 8.1. There is a possibility that the OFCORS are powered by more thermal or synchrotron radiation than the RFC sources, which can be determined by comparing their distributions. This in itself would be significant to see why the OFCORS are special and even may give a clue why they are faint in the optical. In Figure 8.3, the normalized distributions of the spectral indexes for the RFC, seen in red, and the OFCORS, seen in blue, are plotted. Comparing the distributions, the OFCORS appears to be a Gaussian distribution as well centered roughly around -0.2 with a very similar shape to the RFC. This similarity between the two distributions, leads us to believe that there is nothing special about what is powering the OFCORS compared to the entire

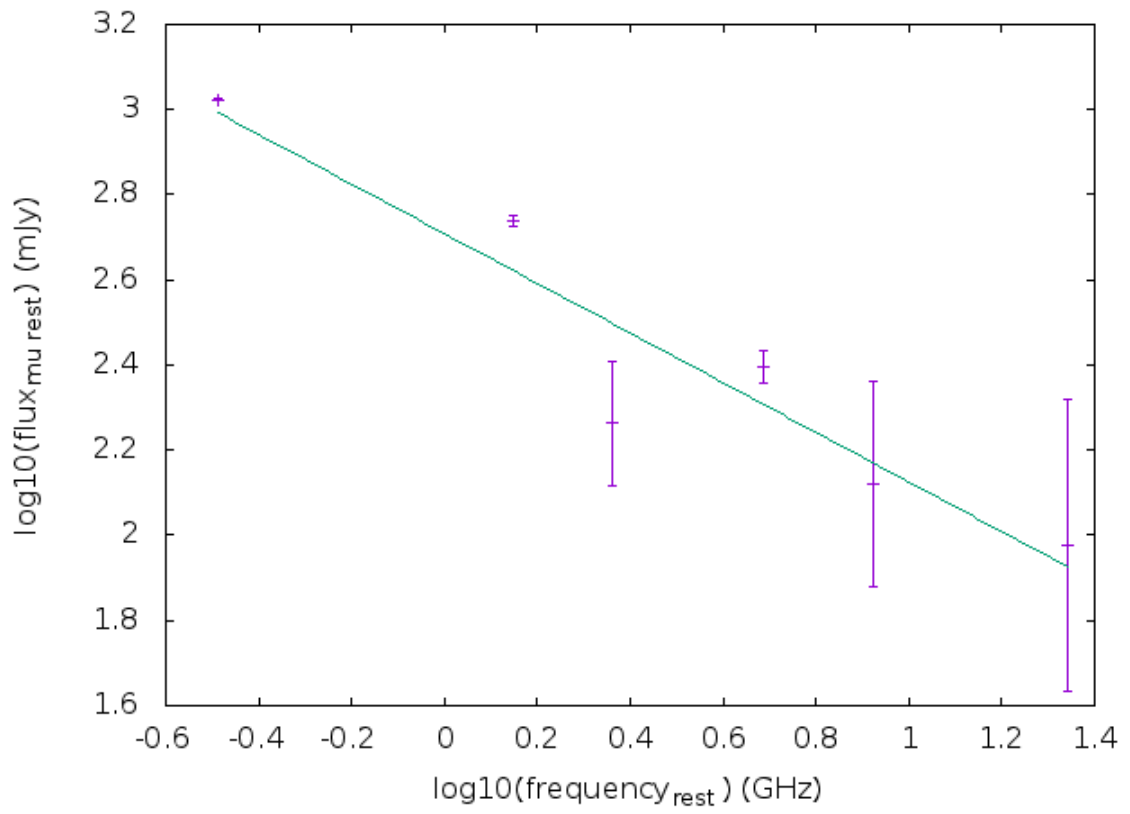


Figure 8.1: Example of the radio power law fit for source J0001+6051

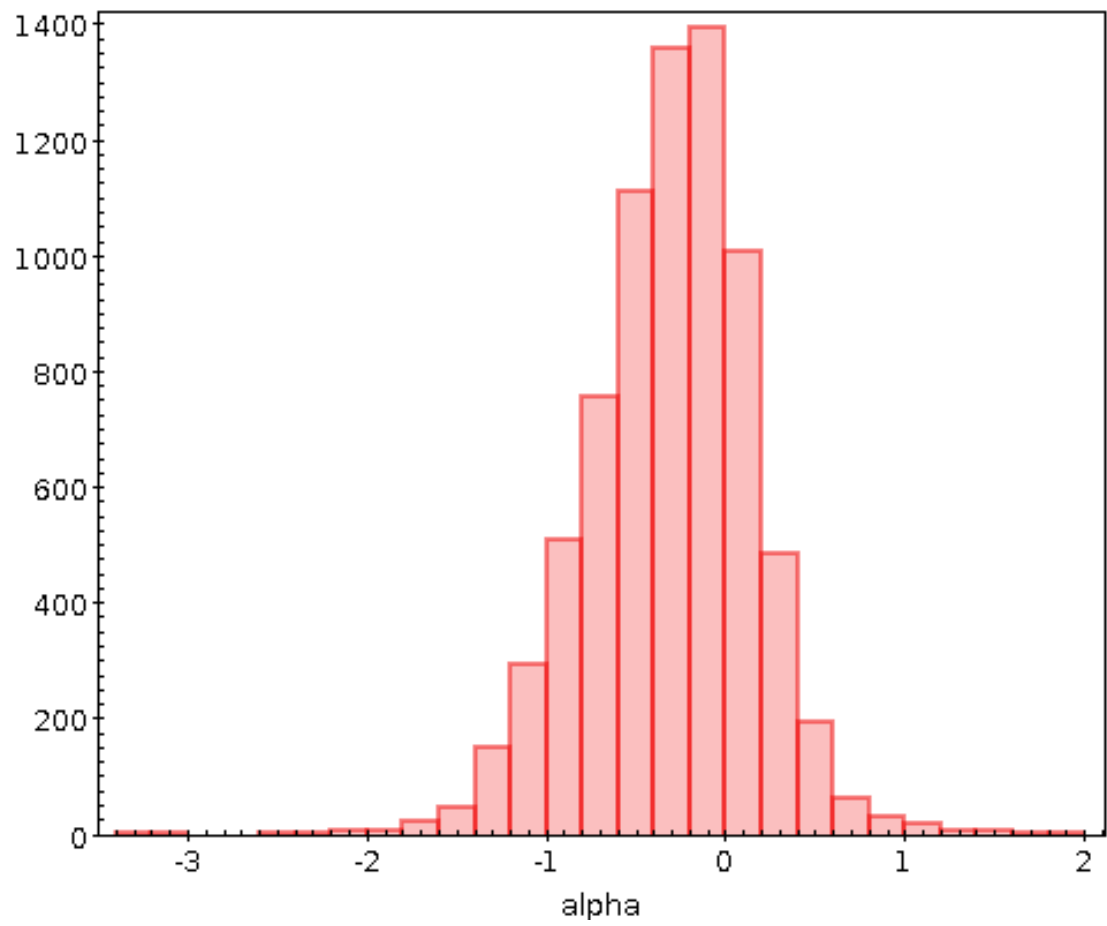


Figure 8.2: Distribution of the spectral index for the entire RFC

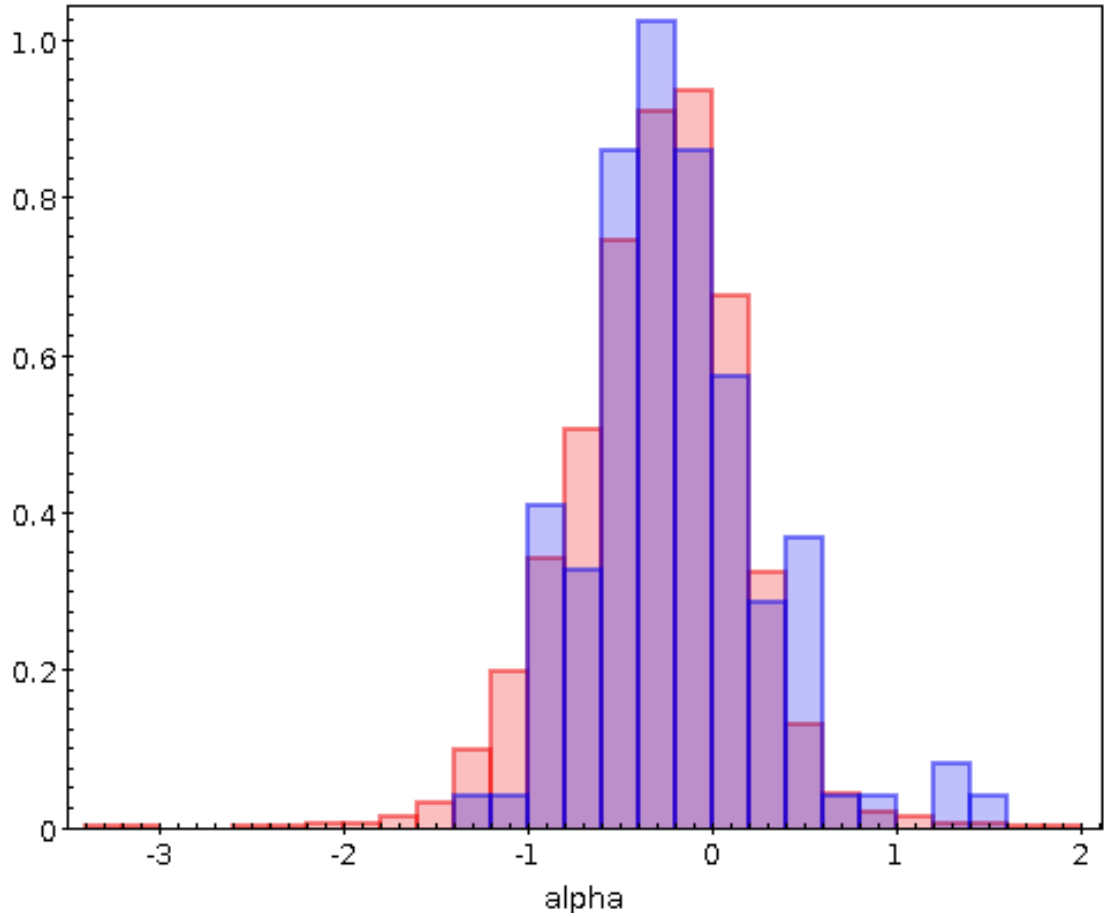


Figure 8.3: Comparison of the RFC and the OFCORS subset values of the spectral index. The RFC is in red and the OFCORS is in blue.

RFC.

Table 8.1

Name	# of bands	α
J0037+0808	4	0.542131
J0155-1443	3	-0.362828
J0158-1413	4	-0.618455
J0200-1542	3	0.521534

Continuation of Table 8.1

Name	# of bands	α
J0210-0737	3	-0.0819667
J0305+0523	4	0.336329
J0311+0508	4	-1.17597
J0330-0551	4	-0.933571
J0418+5457	6	-0.258521
J0425+0833	4	-0.771076
J0425+1755	4	-0.755378
J0501+1356	4	0.120688
J0543+3023	5	-0.184711
J0559+2353	6	-0.0601623
J0636+0801	3	-0.108744
J0726+3912	4	-0.314358
J0754+5324	7	-0.11093
J0805+2106	5	-0.489042
J0811+4308	4	0.212249
J0819+3823	3	1.20651
J0857+2805	3	-0.364374
J0904+2729	3	0.170461
J0911+3033	3	-0.00965964
J0917+6530	6	-0.557937
J0923+3849	7	-0.139479
J0945+2729	3	-0.378045

Continuation of Table 8.1

Name	# of bands	<i>alpha</i>
J0947+5907	6	0.573033
J0952+4941	4	-0.0946654
J1002+5701	4	0.112711
J1006+3050	4	0.181808
J1006+4836	4	0.362956
J1009+0622	4	-0.205063
J1013+3043	4	-0.19069
J1025+8144	4	-0.721709
J1041+4900	5	-0.517058
J1042+2949	4	-0.349945
J1048+0141	5	-0.405948
J1058+4248	4	-0.436734
J1102+7905	4	-1.29363
J1103+2203	5	-0.246167
J1104+5752	4	-0.0592271
J1109+3744	8	-0.437197
J1110+6028	6	0.035108
J1120+3257	4	-0.269173
J1124+3214	4	-0.235441
J1131+3013	4	-0.716753
J1135+4258	4	-0.152554
J1137+4741	3	0.104203

Continuation of Table 8.1

Name	# of bands	α
J1143+1834	6	-0.376858
J1152+3327	4	0.0127152
J1204+3023	4	0.384036
J1208+5413	8	-0.253182
J1220+6327	4	0.468364
J1237+2034	5	0.159348
J1239+3705	3	1.42358
J1241+5458	6	0.297251
J1251+2102	3	-0.230815
J1254+1843	4	-0.128452
J1256+2220	3	-0.586068
J1304+3523	4	-0.870562
J1304+5056	4	-0.442889
J1311+1658	6	-0.547749
J1314+5306	6	0.923327
J1320+5036	8	-0.513596
J1322+4303	4	-0.498429
J1326+4434	5	-0.54148
J1344+1409	4	-0.96184
J1350+6428	5	-0.834001
J1412+1334	6	-0.279613
J1412+3200	4	-0.524803

Continuation of Table 8.1

Name	# of bands	<i>alpha</i>
J1416+3444	6	-0.378314
J1434+4236	5	-0.414725
J1439+2114	5	0.56391
J1442+3042	4	-0.366821
J1448+5326	6	0.475354
J1457+0938	4	-0.190685
J1457+2435	3	-0.358407
J1500+0839	4	-0.848946
J1504+3249	6	-0.158804
J1514+3046	3	0.0107109
J1532+2919	4	-0.291604
J1538+0019	4	-0.386598
J1556+1825	5	0.248463
J1559+1624	3	-0.435184
J1602+3326	8	-0.174653
J1603+2126	3	-0.723517
J1604+4746	5	-0.376923
J1605+2214	3	-0.497244
J1605+5931	4	0.116869
J1624+2748	3	1.32186
J1630+2131	6	-0.655305
J1632+3547	4	-0.136196

Continuation of Table 8.1

Name	# of bands	<i>alpha</i>
J1648+4233	4	-0.460051
J1700+3830	4	-0.484724
J1718+2948	4	0.236547
J1722+6144	3	0.595869
J1735+5049	7	0.591899
J1741+4751	6	0.401874
J1746+6421	7	-0.854933
J1846+2036	3	0.0696388
J1909+7813	4	-0.963231
J1955+3233	4	-0.458373
J1957+3427	4	0.095803
J1958+3424	4	-0.0797863
J2001+3323	4	-0.224665
J2035+5821	5	-0.0224027
J2042+0356	5	-0.507125
J2105+0033	4	-0.28974
J2105+5356	4	-0.243971
J2117+5431	6	0.0599207
J2118+5339	4	-0.0841736
J2130+0502	5	-0.663836
J2131+5214	5	-0.812177
J2210+2013	5	0.766677

Continuation of Table 8.1

Name	# of bands	α
J2219+2141	4	-0.452703
J2229+0114	4	-0.0806387
J2245+6537	4	0.0539172
J2253+6005	4	-0.197695
J2253+6043	5	-0.840879
J2254+6133	4	-0.326203
J2258+6325	4	-0.899337
J2314+5813	3	-0.21499

Table 8.1: Spectral index for the OFCORS sources

Chapter 9

Conclusions

In conclusion, we have constructed a catalog of unique sources that are invisible in the SDSS, yet they are very bright in the radio regime. What makes them even more unique is even after taking deeper optical imaging using OSMOS, not all of the sources were detected down to a magnitude limit of ~ 25 . This could either be because of dust extinction blocking the optical emission or due to the source being located at a high redshift. By looking at the W1-z plot, and the range of sources detected, and assuming the OFCORS should fall in that plot, then it can be inferred that the OFCORS that are detected in W1 should have $z \sim 1-2$. Since W1 is not effected by dust extinction, then the OFCORS that are not detected in W1 could have a redshift greater than 4. Therefore taking this into account, we can not say if the OFCORS are faint in the optical due to their redshift or dust extinction, but a mixture of the two. Looking at the radio power law, there is no difference in the spectral index population between the OFCORS and RFC which we conclude that there must be nothing unique that is powering the OFCORS.

There are several things that still can be done to explore the nature of the OFCORS by looking at various wavelengths. First, looking at these sources in the sub-mm and far-infrared would allow for the exploration of the FIR-radio relation for the OFCORS, explore

the the gas properties of these sources in the sub-mm, and fill in the gap between mid-IR and radio ranges allowing the construction of a more complete SED. Also more optical and IR data is coming out in the next few years through various surveys. Since Pan-STARRS is taking images of the sky repeatedly to look for transients, future data releases of Pan-STARRS will have deeper and deeper optical imaging, which should allow for fainter objects to be detected. Since Pan-STARRS is already deeper and has a wider coverage than SDSS, it would be ideal to use as the optical basis for redefining the OFCORS. Another optical survey that is coming online is the Large Synoptic Survey Telescope (LSST) which is a wide-field survey telescope with an 8.4m mirror that will observe the night sky every few nights. In the IR regime, the James Webb Space Telescope (JWST), scheduled for launch in 2018, will allow for some of the deepest observations of the OFCORS. Finally when this project started the RFC only had ~ 7000 sources in it. However observations of new radio sources are being continuously done with the goal of finding new VLBI calibrators and adding to the RFC, where today the RFC now has ~ 12000 sources, and updating their positions to be more accurate. Further iterations of the RFC will allow for more possible OFCORS to be found in the radio regime.

Appendix A

IRAF parameter files

This appendix contains examples of the IRAF parameter files used.

A.1 Flatcombine parameter file

Example of a parameter file for the IRAF task flatcombine used during the flat-fielding step to create the flat field image.

```
input = "@flatlist.txt" List of flat field images to combine
(output = "twilightflat.20140325.i.fits") Output flat field root name
(combine = "average") Type of combine operation
(reject = "ccdclip") Type of rejection
(ccdtype = "") CCD image type to combine
(process = no) Process images before combining?
(subsets = yes) Combine images by subset parameter?
(delete = no) Delete input images after combining?
(clobber = no) Clobber existing output image?
(scale = "mode") Image scaling
(statsec = "") Image section for computing statistics
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.0) Lower sigma clipping factor
(hsigma = 3.0) Upper sigma clipping factor
(rdnoise = "5") ccdclip: CCD readout noise (electrons)
(gain = "2.2") ccdclip: CCD gain (electrons/DN)
```

(snoise = "0.") ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 1.0) Value if there are no pixels
(mode = "al")

A.2 ccdproc parameter file for flat-field correction

This is an example parameter file for ccdproc for applying the flat field correction to each raw image.

images = "@objectlist.txt" List of CCD images to correct
(output = "@pf_objectlist.txt") List of output CCD images
(ccdtype = "") CCD image type to correct
(max_cache = 0) Maximum image caching memory (in Mbytes)
(noprocs = no) List processing steps only?

(fixpix = no) Fix bad CCD lines and columns?
(overscan = no) Apply overscan strip correction?
(trim = no) Trim the image?
(zerocor = no) Apply zero level correction?
(darkcor = no) Apply dark count correction?
(flatcor = yes) Apply flat field correction?
(illumcor = no) Apply illumination correction?
(fringe = no) Apply fringe correction?
(readcor = no) Convert zero level image to readout correction?
(scancor = no) Convert flat field image to scan correction?

(readaxis = "line") Read out axis (column—line)
(fixfile = "") File describing the bad lines and columns
(biassec = "") Overscan strip image section
(trimsec = "") Trim data section
(zero = "") Zero level calibration image
(dark = "") Dark count calibration image
(flat = "twilightflat.20130314_17.i.fits") Flat field images
(illum = "smooth_200x200_20130316.i.fits") Illumination correction images
(fringe = "") Fringe correction images
(minreplace = 1.0) Minimum flat field value
(scantype = "shortscan") Scan type (shortscan—longscan)

(nscan = 1) Number of short scan lines

(interactive = no) Fit overscan interactively?
(function = "legendre") Fitting function
(order = 4) Number of polynomial terms or spline pieces
(sample = "*"*) Sample points to fit
(naverage = 1) Number of sample points to combine
(niterate = 1) Number of rejection iterations
(low_reject = 3.0) Low sigma rejection factor
(high_reject = 3.0) High sigma rejection factor
(grow = 0.0) Rejection growing radius
(mode = "al")

A.3 Illumination stack with combine

input = "@pfc_objectlist.txt" List of images to combine
output = "illum_cor_input.fits" List of output images
(plfile = "") List of output pixel list files (optional)
(sigma = "") List of sigma images (optional)

(ccdtype = "") CCD image type to combine (optional)
(subsets = no) Combine images by subset parameter?
(delete = no) Delete input images after combining?
(clobber = no) Clobber existing output image?

(combine = "average") Type of combine operation
(reject = "ccdclip") Type of rejection
(project = no) Project highest dimension of input images?
(outtype = "real") Output image pixel datatype
(offsets = "none") Input image offsets
(masktype = "none") Mask type
(maskvalue = 0.0) Mask value
(blank = 0.0) Value if there are no pixels

(scale = "mode") Image scaling
(zero = "none") Image zero point offset
(weight = "@pfc_objectweight.txt") Image weights
(statsec = "") Image section for computing statistics

(lthreshold = INDEF) Lower threshold
(hthreshold = INDEF) Upper threshold
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.0) Lower sigma clipping factor
(hsigma = 3.0) Upper sigma clipping factor
(rdnoise = "5") ccdclip: CCD readout noise (electrons)
(gain = "2.2") ccdclip: CCD gain (electrons/DN)
(snoise = "0.") ccdclip: Sensitivity noise (fraction)
(sigscale = 0.1) Tolerance for sigma clipping scaling corrections
(pclip = -0.5) pclip: Percentile clipping parameter
(grow = 0) Radius (pixels) for 1D neighbor rejection
(mode = "al")

A.4 boxcar

input = "test23.fits" Input images to be fit
output = "smooth_200x200_20130316.i.fits" Output images
xwindow = 200 X dimension of box
ywindow = 200 Y dimension of box
(boundary = "nearest") Boundary (constant,nearest,reflect,wrap)
(constant = 0.0) Constant for boundary extension
(mode = "ql")

A.5 ccdproc for illumination correction

images = "@pf_objectlist.txt" List of CCD images to correct
(output = "@pi_objectlist.txt") List of output CCD images
(ccdtype = "") CCD image type to correct
(max_cache = 0) Maximum image caching memory (in Mbytes)
(noproc = no) List processing steps only?

(fixpix = no) Fix bad CCD lines and columns?
(overscan = no) Apply overscan strip correction?

(trim = no) Trim the image?
(zeroeor = no) Apply zero level correction?
(darkcor = no) Apply dark count correction?
(flatcor = no) Apply flat field correction?
(illumcor = yes) Apply illumination correction?
(fringecor = no) Apply fringe correction?
(readcor = no) Convert zero level image to readout correction?
(scancor = no) Convert flat field image to scan correction?

(readaxis = "line") Read out axis (column—line)
(fixfile = "") File describing the bad lines and columns
(biassec = "") Overscan strip image section
(trimsec = "") Trim data section
(zero = "") Zero level calibration image
(dark = "") Dark count calibration image
(flat = "twilightflat.20130314_17.i.fits") Flat field images
(illum = "smooth_200x200_20130316.i.fits") Illumination correction images
(fringe = "") Fringe correction images
(minreplace = 1.0) Minimum flat field value
(scantype = "shortscan") Scan type (shortscan—longscan)
(nscan = 1) Number of short scan lines

(interactive = no) Fit overscan interactively?
(function = "legendre") Fitting function
(order = 4) Number of polynomial terms or spline pieces
(sample = "") Sample points to fit
(naverage = 1) Number of sample points to combine
(niterate = 1) Number of rejection iterations
(low_reject = 3.0) Low sigma rejection factor
(high_reject = 3.0) High sigma rejection factor
(grow = 0.0) Rejection growing radius
(mode = "al")

A.6 precess

input = "STDIN" input files
startyear = 2014.2 precess from year
endyear = 2000.0 precess to year

(stdepoth = 0.0) standard epoch for second set of output coords (0 to skip)
(mode = "al")

Appendix B

Weight Script

This Appendix is devoted to my weight script for the illumination correction step. The master code is given by `jobgetweights_master.cl` and is to be run in Pyraf. This code runs through a series of bash scripts which in turn creates the scripts that will be inputted into pyraf tasks.

B.1 `jobgetweights_master.cl`

```
delete 0*.pl
delete 0*.fits
!bash jobcreator_getmskregions.sh
cl < jobgetmskreg.cl
!bash jobcreator_makepl.sh
cl < jobgetmakepl.cl
!bash jobcreator_getpl2fits.sh
cl < jobgetpl2fits.cl
stsdas
!bash jobcreator_imcalcstep1.sh
cl < jobimcalcstep1.cl
!bash jobcreator_imcalcstep2.sh
cl < jobimcalcstep2.cl
delete 0*interstepweights.fits
!bash jobcreator_makesegmentation.sh
!bash jobmakesegmentation.sh
!bash jobcreator_imcalcsegsetto1.sh
cl < jobimcalcsegsetto1.cl
!bash jobcreator_imcalccombine.sh
cl < jobimcalccombine.cl
```


B.2 jobcreator_getmskregions.sh

```
ls 0*.reg > tmp1.txt
awk '{print $1}' tmp1.txt | sed -e 's/.reg/.mskreg/' > tmp2.txt
paste tmp1.txt tmp2.txt > tmp3.txt
awk '{print $1, $2}' tmp3.txt | sed -e 's// > /' -e 's/^\//grep -e circle -e polygon /'
    > jobgetmskreg.cl rm tmp1.txt
rm tmp2.txt
rm tmp3.txt
```

B.3 jobcreator_makepl.sh

```
ls 0*.mskreg > tmp1.txt
ls 0*.mskreg | sed -e 's/.mskreg/.pl/' > tmp2.txt
paste tmp1.txt tmp2.txt | sed -e 's/^\//mskregions /' -e 's/$/ """""" /' > jobgetmakepl.cl
rm tmp1.txt
rm tmp2.txt
```

B.4 jobcreator_getpl2fits.sh

```
ls 0*.pl > tmp1.txt
ls 0*.pl | sed -e 's/.pl/.fits/' > tmp2.txt
paste tmp1.txt tmp2.txt | sed -e 's/^\//imcopy /' > jobgetpl2fits.cl
rm tmp1.txt
rm tmp2.txt
```

B.5 jobcreator_imcalcstep1.sh

```
ls 0*.fits > tmp1.txt
ls 0*.fits | sed -e 's/.fits/interstepweights.fits/' > tmp2.txt
paste tmp1.txt tmp2.txt | sed -e 's/^\//imcalc /' -e 's/$/ ""im1-1"" /' > jobimcalcstep1.cl
rm tmp1.txt
```

```
rm tmp2.txt
```

B.6 jobcreator_imcalcstep2.sh

```
ls 0*inter*.fits > tmp1.txt
ls 0*inter*.fits | sed -e 's/interstepweights.fits/.weight.fits/' > tmp2.txt
paste tmp1.txt tmp2.txt | sed -e 's/^\s/imcalc /' -e 's/$/ @exp.dat/' > jobimcalcstep2.cl
rm tmp1.txt
rm tmp2.txt
```

B.7 jobcreator_makesegmentation.sh

```
rm 0*segmentation.fits
ls pf_*.fits > tmp1.txt
ls 0*weight.fits > tmp2.txt
ls 0*weight.fits | sed -e 's/weight.fits/segmentation.fits/' > tmp3.txt
awk 'print $1' tmp1.txt | sed -e 's/^\s/sex /' -e 's/$/ -c crap.conf/' > tmp4.txt
awk 'print $1' tmp2.txt | sed -e 's/^\s/-WEIGHT_IMAGE /' > tmp5.txt
awk 'print $1' tmp3.txt | sed -e 's/^\s/-CHECKIMAGE_NAME /' > tmp6.txt
paste tmp4.txt tmp5.txt tmp6.txt > jobmakesegmentation.sh
rm tmp1.txt
rm tmp2.txt
rm tmp3.txt
rm tmp4.txt
rm tmp5.txt
rm tmp6.txt
```

B.8 jobcreator_imcalcsegsetto1.sh

```
ls 0*segmentation.fits > tmp1.txt
ls 0*segmentation.fits | sed -e 's/.fits/.1.fits/' > tmp2.txt
paste tmp1.txt tmp2.txt | sed -e 's/^\s/imcalc /' -e 's/$/ @seg.dat/' > jobimcalcsegsetto1.cl
rm tmp1.txt
rm tmp2.txt
```

B.9 jobcreator_imcalccombine.sh

```
ls 0*weight.fits > tmp1.txt
ls 0*segmentation.1.fits > tmp2.txt
ls 0*weight.fits | sed -e 's/.weight.fits/.ws_combine.fits/' > tmp3.txt
paste tmp1.txt tmp2.txt > tmp4.txt
awk 'print $1,$2' tmp4.txt | sed -e 's//,/ ' > tmp5.txt
paste tmp5.txt tmp3.txt | sed -e 's/^/imcalc /' -e 's/$/ @combine.dat/'
  > jobimcalccombine.cl
rm tmp1.txt
rm tmp2.txt
rm tmp3.txt
rm tmp4.txt
rm tmp5.txt
```

Appendix C

Scamp and Swarp

This Appendix includes all configuration and parameter files that went into calculating the .head files by using Scamp as well as stacking by Swarp.

C.1 SExtractor for Scamp

```
# Default configuration file for SExtractor 2.8.6
# EB 2012-03-15
#

#----- Catalog -----

CATALOG_NAME RADIO_A.cat # name of the output catalog
CATALOG_TYPE FITS_LDAC # NONE,ASCII,ASCII_HEAD, ASCII_SKYCAT,
# ASCII_VOTABLE, FITS_1.0 or FITS_LDAC
PARAMETERS_NAME sex.param # name of the file containing catalog contents

#----- Extraction -----

DETECT_TYPE CCD # CCD (linear) or PHOTO (with gamma correction)
DETECT_MINAREA 10 # minimum number of pixels above threshold
THRESH_TYPE RELATIVE # threshold type: RELATIVE (in sigmas)
# or ABSOLUTE (in ADUs)
DETECT_THRESH 20.0 # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
ANALYSIS_THRESH 20.0 # <sigmas> or <threshold>,<ZP> in mag.arcsec-2

FILTER Y # apply filter for detection (Y or N)?
FILTER_NAME gauss_5.0_9x9.conv # name of the file containing the filter
```

FILTER_THRESH # Threshold[s] for retina filtering

 DEBLEND_NTHRESH 32 # Number of deblending sub-thresholds
 DEBLEND_MINCONT 0.005 # Minimum contrast parameter for deblending

 CLEAN Y # Clean spurious detections? (Y or N)?
 CLEAN_PARAM 1.0 # Cleaning efficiency

 MASK_TYPE CORRECT # type of detection MASKing: can be one of
 # NONE, BLANK or CORRECT

 #————— WEIGHTing —————

 WEIGHT_TYPE MAP_WEIGHT # type of WEIGHTing: NONE, BACKGROUND,
 # MAP_RMS, MAP_VAR or MAP_WEIGHT
 WEIGHT_IMAGE J1434+4236_A.20120421.0075.z.weight.fits # weight-map filename
 WEIGHT_GAIN Y # modulate gain (E/ADU) with weights? (Y/N)
 WEIGHT_THRESH # weight threshold[s] for bad pixels

 #————— FLAGging —————

 FLAG_IMAGE flag.fits # filename for an input FLAG-image
 FLAG_TYPE OR # flag pixel combination: OR, AND, MIN, MAX
 # or MOST

 #————— Photometry —————

 PHOT_APERTURES 5 # MAG_APER aperture diameter(s) in pixels
 PHOT_AUTOPARAMS 2.5, 3.5 # MAG_AUTO parameters: <Kron_fact>,<min_radius>
 PHOT_PETROPARAMS 2.0, 3.5 # MAG_PETRO parameters: <Petrosian_fact>,
 # <min_radius>
 PHOT_AUTOAPERS 0.0,0.0 # <estimation>,<measurement> minimum apertures
 # for MAG_AUTO and MAG_PETRO
 PHOT_FLUXFRAC 0.5 # flux fraction[s] used for FLUX_RADIUS

 SATUR_LEVEL 65000.0 # level (in ADUs) at which arises saturation
 SATUR_KEY SATURATE # keyword for saturation level (in ADUs)

 MAG_ZEROPOINT 0.0 # magnitude zero-point
 MAG_GAMMA 4.0 # gamma of emulsion (for photographic scans)

GAIN 1584.22 # detector gain in e-/ADU
 GAIN_KEY GAIN # keyword for detector gain in e-/ADU
 PIXEL_SCALE 0.0 # size of pixel in arcsec (0=use FITS WCS info)

#----- Star/Galaxy Separation -----

SEEING_FWHM 1.2 # stellar FWHM in arcsec
 STARNNW_NAME default.nnw # Neural-Network_Weight table filename

#----- Background -----

BACK_TYPE AUTO # AUTO or MANUAL
 BACK_VALUE 0.0 # Default background value in MANUAL mode
 BACK_SIZE 64 # Background mesh: <size> or <width>,<height>
 BACK_FILTERSIZE 3 # Background filter: <size> or <width>,<height>

BACKPHOTO_TYPE LOCAL # can be GLOBAL or LOCAL
 BACKPHOTO_THICK 24 # thickness of the background LOCAL annulus
 BACK_FILTTHRESH 0.0 # Threshold above which the background-
 # map filter operates

#----- Check Image -----

CHECKIMAGE_TYPE NONE # can be NONE, BACKGROUND, BACKGROUND_RMS,
 # MINIBACKGROUND, MINIBACK_RMS, -BACKGROUND,
 # FILTERED, OBJECTS, -OBJECTS, SEGMENTATION,
 # or APERTURES
 CHECKIMAGE_NAME check.fits # Filename for the check-image

#----- Memory (change with caution!) -----

MEMORY_OBJSTACK 3000 # number of objects in stack
 MEMORY_PIXSTACK 300000 # number of pixels in stack
 MEMORY_BUFSIZE 1024 # number of lines in buffer

#----- ASSOCIation -----

ASSOC_NAME sky.list # name of the ASCII file to ASSOCIate
 ASSOC_DATA 2,3,4 # columns of the data to replicate (0=all)
 ASSOC_PARAMS 2,3,4 # columns of xpos,ypos[,mag]

ASSOC_RADIUS 2.0 # cross-matching radius (pixels)
ASSOC_TYPE NEAREST # ASSOCIation method: FIRST, NEAREST, MEAN,
MAG_MEAN, SUM, MAG_SUM, MIN or MAX
ASSOCSELEC_TYPE MATCHED # ASSOC selection type: ALL, MATCHED or -MATCHED

#————— Miscellaneous —————

VERBOSE_TYPE NORMAL # can be QUIET, NORMAL or FULL
WRITE_XML N # Write XML file (Y/N)?
XML_NAME sex.xml # Filename for XML output
XSL_URL file:///scisoft/share/sextractor/sextractor.xsl
Filename for XSL style-sheet
NTHREADS 0 # Number of simultaneous threads for
the SMP version of SExtractor
0 = automatic

FITS_UNSIGNED N # Treat FITS integer values as unsigned (Y/N)?
INTERP_MAXXLG 16 # Max. lag along X for 0-weight interpolation
INTERP_MAXYLG 16 # Max. lag along Y for 0-weight interpolation
INTERP_TYPE ALL # Interpolation type: NONE, VAR_ONLY or ALL

#————— Experimental Stuff —————

PSF_NAME default.psf # File containing the PSF model
PSF_NMAX 9 # Max.number of PSFs fitted simultaneously
PSFDISPLAY_TYPE SPLIT # Catalog type for PSF-fitting: SPLIT or VECTOR
PATTERN_TYPE RINGS-HARMONIC # can RINGS-QUADPOLE, RINGS-OCTOPOLE,
RINGS-HARMONICS or GAUSS-LAGUERRE
SOM_NAME default.som # File containing Self-Organizing Map weights

C.1.1 SExtractor parameter file

XWIN_IMAGE
YWIN_IMAGE
ERRAWIN_IMAGE
ERRBWIN_IMAGE
ERRTHETAWIN_IMAGE
FLUX_AUTO
FLUXERR_AUTO

FLUX_RADIUS

C.2 Scamp configuration file

```
# Default configuration file for SCAMP 2.0.4
# EB 2015-05-29
#

#----- Field grouping -----

FGROUP_RADIUS 1.0 # Max dist (deg) between field groups

#----- Reference catalogs -----

REF_SERVER cocat1.u-strasbg.fr # Internet addresses of catalog servers
REF_PORT 80 # Ports to connect to catalog servers
CDSCLIENT_EXEC aclient.cgi # CDSclient executable
ASTREF_CATALOG SDSS-R7 # NONE, FILE, USNO-A1,USNO-A2,USNO-B1,
# GSC-1.3,GSC-2.2,GSC-2.3,
# TYCHO-2, UCAC-1,UCAC-2,UCAC-3,UCAC-4,
# NOMAD-1, PPMX, CMC-14, 2MASS, DENIS-3,
# SDSS-R3,SDSS-R5,SDSS-R6,SDSS-R7,
# SDSS-R8, SDSS-R9
ASTREF_BAND DEFAULT # Photom. band for astr.ref.magnitudes
# or DEFAULT, BLUEST, or REDDEST
ASTREFCAT_NAME SDSS-R7_0559+2354_r15_cuts.cat # Local astrometric reference cat-
alogs
ASTREFCENT_KEYS X_WORLD,Y_WORLD # Local ref.cat. centroid parameters
ASTREFERR_KEYS ERRA_WORLD, ERRB_WORLD, ERRTHETA_WORLD
# Local ref.cat. err. ellipse params
ASTREFMAG_KEY MAG # Local ref.cat. magnitude parameter
ASTREFMAGERR_KEY MAGERR # Local ref.cat. mag. error parameter
ASTREFOBSDATE_KEY OBSDATE # Local ref.cat. obs. date parameter
ASTREFMAG_LIMITS -99.0,99.0 # Select magnitude range in ASTREF_BAND
SAVE_REFCATALOG Y # Save ref catalogs in FITS-LDAC format?
REFOUT_CATPATH . # Save path for reference catalogs

#----- Merged output catalogs -----
```


MERGEDOUTCAT_TYPE NONE # NONE, ASCII_HEAD, ASCII, FITS_LDAC
MERGEDOUTCAT_NAME merged.cat # Merged output catalog filename

#----- Full output catalogs -----

FULLOUTCAT_TYPE NONE # NONE, ASCII_HEAD, ASCII, FITS_LDAC
FULLOUTCAT_NAME full.cat # Full output catalog filename

#----- Pattern matching -----

MATCH Y # Do pattern-matching (Y/N) ?
MATCH_NMAX 0 # Max.number of detections for MATCHing
(0=auto)
PIXSCALE_MAXERR 2.0 # Max scale-factor uncertainty
POSANGLE_MAXERR 180.0 # Max position-angle uncertainty (deg)
POSITION_MAXERR 3.0 # Max positional uncertainty (arcmin)
MATCH_RESOL 0 # Matching resolution (arcsec); 0=auto
MATCH_FLIPPED N # Allow matching with flipped axes?
MOSAIC_TYPE FIX_FOCALPLANE # UNCHANGED, SAME_CRVAL, SHARE_PROJAXIS,
FIX_FOCALPLANE or LOOSE
FIXFOCALPLANE_NMIN 1 # Min number of dets for FIX_FOCALPLANE

#----- Cross-identification -----

CROSSID_RADIUS 4.0 # Cross-id initial radius (arcsec)

#----- Astrometric solution -----

SOLVE_ASTROM Y # Compute astrometric solution (Y/N) ?
PROJECTION_TYPE SAME # SAME, TPV or TAN
ASTRINSTRU_KEY FILTER,QRUNID # FITS keyword(s) defining the astrom
STABILITY_TYPE INSTRUMENT # EXPOSURE, PRE-DISTORTED or INSTRUMENT
CENTROID_KEYS XWIN_IMAGE,YWIN_IMAGE # Cat. parameters for centroiding
CENTROIDERR_KEYS ERRAWIN_IMAGE,ERRBWIN_IMAGE,ERRTHETAWIN_IMAGE
Cat. params for centroid err ellipse
DISTORT_KEYS XWIN_IMAGE,YWIN_IMAGE # Cat. parameters or FITS keywords
DISTORT_GROUPS 1,1 # Polynom group for each context key
DISTORT_DEGREES 3 # Polynom degree for each group
FOCDISTORT_DEGREE 1 # Polynom degree for focal plane coords

ASTREF_WEIGHT 1.0 # Relative weight of ref.astrom.cat.
 ASTRACCURACY_TYPE SIGMA-PIXEL # SIGMA-PIXEL, SIGMA-ARCSEC,
 # or TURBULENCE-ARCSEC
 ASTRACCURACY_KEY ASTRACCU # FITS keyword for ASTR_ACCURACY param.
 ASTR_ACCURACY 0.01 # Astrom. uncertainty floor parameter
 ASTRCLIP_NSIGMA 3.0 # Astrom. clipping threshold in sigmas
 COMPUTE_PARALLAXES N # Compute trigonom. parallaxes (Y/N)?
 COMPUTE_PROPERMOTIONS N # Compute proper motions (Y/N)?
 CORRECT_COLOURSHIFTS N # Correct for colour shifts (Y/N)?
 INCLUDE_ASTREFCATALOG Y # Include ref.cat in prop.motions (Y/N)?
 ASTR_FLAGSMASK 0x00fc # Astrometry rejection mask on SEx FLAGS
 ASTR_IMAFLAGSMASK 0x0 # Astrometry rejection mask on IMAFLAGS

#----- Photometric solution -----

SOLVE_PHOTOM Y # Compute photometric solution (Y/N) ?
 MAGZERO_OUT 0.0 # Magnitude zero-point(s) in output
 MAGZERO_INTERR 0.01 # Internal mag.zero-point accuracy
 MAGZERO_REFERR 0.03 # Photom.field mag.zero-point accuracy
 PHOTINSTRU_KEY FILTER # FITS keyword(s) defining the photom.
 MAGZERO_KEY PHOT_C # FITS keyword for the mag zero-point
 EXPOTIME_KEY EXPTIME # FITS keyword for the exposure time (s)
 AIRMASS_KEY AIRMASS # FITS keyword for the airmass
 EXTINGT_KEY PHOT_K # FITS keyword for the extinction coeff
 PHOTOMFLAG_KEY PHOTFLAG # FITS keyword for the photometry flag
 PHOTFLUX_KEY FLUX_AUTO # Catalog param. for the flux measurement
 PHOTFLUXERR_KEY FLUXERR_AUTO # Catalog parameter for the flux error
 PHOTCLIP_NSIGMA 3.0 # Photom.clipping threshold in sigmas
 PHOT_ACCURACY 1e-3 # Photometric uncertainty floor (frac.)
 PHOT_FLAGSMASK 0x00fc # Photometry rejection mask on SEx FLAGS
 PHOT_IMAFLAGSMASK 0x0 # Photometry rejection mask on IMAFLAGS

#----- Check-plots -----

CHECKPLOT_CKEY SCAMPCOL # FITS keyword for PLPLOT field colour
 CHECKPLOT_DEV PNG # NULL, XWIN, TK, PS, PSC, XFIG, PNG,
 # JPEG, AQT, PDF or SVG
 CHECKPLOT_RES 0 # Check-plot resolution (0 = default)
 CHECKPLOT_ANTIALIAS Y # Anti-aliasing using convert (Y/N) ?
 CHECKPLOT_TYPE FGROUPTS, DISTORTION, ASTR_INTERROR2D, ASTR_INTERROR1D,

ASTR_REFERROR2D, ASTR_REFERROR1D, ASTR_CHI2, PHOT_ERROR
CHECKPLOT_NAME fgroups, distort, astr_interror2d, astr_interror1d,
astr_referror2d, astr_referror1d, astr_chi2, psphot_error # Check-plot filename(s)

#----- Check-images -----

CHECKIMAGE_TYPE NONE # NONE, AS_PAIR, AS_REFPAIR, or AS_XCORR
CHECKIMAGE_NAME check.fits # Check-image filename(s)

#----- Miscellaneous -----

SN_THRESHOLDS 10.0,100.0 # S/N thresholds (in sigmas) for all and
high-SN sample
FWHM_THRESHOLDS 0.0,100.0 # FWHM thresholds (in pixels) for sources
ELLIPTICITY_MAX 0.5 # Max. source ellipticity
FLAGS_MASK 0x00f0 # Global rejection mask on SEX FLAGS
WEIGHTFLAGS_MASK 0x00ff # Global rejec. mask on SEX FLAGS_WEIGHT
IMAFLAGS_MASK 0x0 # Global rejec. mask on SEX IMAFLAGS_ISO
AHEADER_GLOBAL scamp.ahead # Filename of the global INPUT header
AHEADER_SUFFIX .ahead # Filename extension for additional
INPUT headers
HEADER_SUFFIX .head # Filename extension for OUTPUT headers
HEADER_TYPE NORMAL # NORMAL or FOCAL_PLANE
VERBOSE_TYPE NORMAL # QUIET, NORMAL, LOG or FULL
WRITE_XML Y # Write XML file (Y/N)?
XML_NAME scamp.xml # Filename for XML output
XSL_URL file:///usr/local/share/scamp/scamp.xsl
Filename for XSL style-sheet
NTHREADS 0 # Number of simultaneous threads for
the SMP version of SCAMP
0 = automatic

C.3 Swarp

Default configuration file for SWarp 2.38.0
EB 2015-05-28

#----- Output -----

```

IMAGEOUT_NAME coadd.fits # Output filename
WEIGHTOUT_NAME coadd.weight.fits # Output weight-map filename

HEADER_ONLY N # Only a header as an output file (Y/N)?
HEADER_SUFFIX .head # Filename extension for additional headers

#----- Input Weights -----

WEIGHT_TYPE MAP_WEIGHT # BACKGROUND,MAP_RMS,MAP_VARIANCE
# or MAP_WEIGHT
RESCALE_WEIGHTS Y # Rescale input weights/variances (Y/N)?
WEIGHT_SUFFIX .weight.fits # Suffix to use for weight-maps
WEIGHT_IMAGE # Weightmap filename if suffix not used
# (all or for each weight-map)
WEIGHT_THRESH # Bad pixel weight-threshold

#----- Co-addition -----

COMBINE Y # Combine resampled images (Y/N)?
COMBINE_TYPE MEDIAN # MEDIAN,AVERAGE,MIN,MAX,WEIGHTED,CLIPPED
# CHI-OLD,CHI-MODE,CHI-MEAN,SUM,
# WEIGHTED_WEIGHT,MEDIAN_WEIGHT,
# AND,NAND,OR or NOR
CLIP_AMPFRAC 0.3 # Fraction of flux variation allowed
# with clipping
CLIP_SIGMA 4.0 # RMS error multiple variation allowed
# with clipping
CLIP_WRITELOG N # Write output file with coordinates of
# clipped pixels (Y/N)
CLIP_LOGNAME clipped.log # Name of output file with coordinates
# of clipped pixels
BLANK_BADPIXELS N # Set to 0 pixels having a weight of 0

#----- Astrometry -----

CELESTIAL_TYPE NATIVE # NATIVE, PIXEL, EQUATORIAL,
# GALACTIC,ECLIPTIC, or SUPERGALACTIC
PROJECTION_TYPE TAN # Any WCS projection code or NONE
PROJECTION_ERR 0.001 # Maximum projection error (in output
# pixels), or 0 for no approximation

```

CENTER_TYPE ALL # MANUAL, ALL or MOST
 CENTER 00:00:00.0, +00:00:00.0 # Coordinates of the image center
 PIXELSCALE_TYPE MEDIAN # MANUAL, FIT, MIN, MAX or MEDIAN
 PIXEL_SCALE 0.0 # Pixel scale
 IMAGE_SIZE 0 # Image size (0 = AUTOMATIC)

#————— Resampling —————

RESAMPLE Y # Resample input images (Y/N)?
 RESAMPLE_DIR . # Directory path for resampled images
 RESAMPLE_SUFFIX .resamp.fits # filename extension for resampled images

RESAMPLING_TYPE LANCZOS3 # NEAREST, BILINEAR, LANCZOS2, LANCZOS3
 # LANCZOS4 (1 per axis) or FLAGS
 OVERSAMPLING 0 # Oversampling in each dimension
 # (0 = automatic)
 INTERPOLATE N # Interpolate bad input pixels (Y/N)?
 # (all or for each image)

FSCALASTRO_TYPE FIXED # NONE, FIXED, or VARIABLE
 FSCALE_KEYWORD FLXSCALE # FITS keyword for the multiplicative
 # factor applied to each input image
 FSCALE_DEFAULT 1.0 # Default FSCALE value if not in header

GAIN_KEYWORD GAIN # FITS keyword for effect. gain (e-/ADU)
 GAIN_DEFAULT 0.0 # Default gain if no FITS keyword found
 # 0 = infinity (all or for each image)
 SATLEV_KEYWORD SATURATE # FITS keyword for saturation level (ADU)
 SATLEV_DEFAULT 50000.0 # Default saturation if no FITS keyword

#————— Background subtraction —————

SUBTRACT_BACK Y # Subtraction sky background (Y/N)?
 # (all or for each image)

BACK_TYPE AUTO # AUTO or MANUAL
 # (all or for each image)
 BACK_DEFAULT 0.0 # Default background value in MANUAL
 # (all or for each image)
 BACK_SIZE 128 # Background mesh size (pixels)

```

# (all or for each image)
BACK_FILTERSIZE 3 # Background map filter range (meshes)
# (all or for each image)
BACK_FILTTHRESH 0.0 # Threshold above which the background-
# map filter operates

#----- Memory management -----

VMEM_DIR . # Directory path for swap files
VMEM_MAX 2047 # Maximum amount of virtual memory (MB)
MEM_MAX 256 # Maximum amount of usable RAM (MB)
COMBINE_BUFSIZE 256 # RAM dedicated to co-addition(MB)

#----- Miscellaneous -----

DELETE_TMPFILES Y # Delete temporary resampled FITS files
# (Y/N)?
COPY_KEYWORDS OBJECT # List of FITS keywords to propagate
# from the input to the output headers
WRITE_FILEINFO N # Write information about each input
# file in the output image header?
WRITE_XML Y # Write XML file (Y/N)?
XML_NAME swarp.xml # Filename for XML output
XSL_URL file:///usr/local/share/swarp/swarp.xsl
# Filename for XSL style-sheet
VERBOSE_TYPE NORMAL # QUIET,LOG,NORMAL, or FULL
NNODES 1 # Number of nodes (for clusters)
NODE_INDEX 0 # Node index (for clusters)

NTHREADS 0 # Number of simultaneous threads for
# the SMP version of SWarp
# 0 = automatic
NOPENFILES_MAX 512 # Maximum number of files opened by SWarp

```

Appendix D

Photometry

This appendix is devoted to the scripts and steps used perform the photometry.

D.1 Weight image modification

```
delete coadd.weight.reduced.fits
cp /run/media/jacob/astro2/research/osmos_datared_stacking_testdir/gettingthedatasteps/weightfix.dat
.
ds9 -zscale -invert coadd.weight.fits
emacs weightfix.dat
stdas
imcalc coadd.weight.fits coadd.weight.reduced.fits @weightfix.dat
```

D.1.1 weightfix.dat

```
if im1 .gt. 2640 then 1 else 0
```

D.2 Source photometry

```
hselect coadd.fits gain yes
emacs sex_coadd.conf
!sex coadd.fits -c sex_coadd.conf -WEIGHT_IMAGE coadd.weight.reduced.fits
!awk 'NR>11print 1,2' coadd.cat | sed -e 's/ /,/' -e 's/^\(circle\)/ -e 's/$/,7.5)/' -e '1s/^\(image\
n/' > coadd.reg
```

D.2.1 SExtractor on stacked image

```
# Default configuration file for SExtractor 2.8.6
# EB 2012-03-15
#

#----- Catalog -----

CATALOG_NAME coadd.cat # name of the output catalog
CATALOG_TYPE ASCII_HEAD # NONE,ASCII,ASCII_HEAD, ASCII_SKYCAT,
# ASCII_VOTABLE, FITS_1.0 or FITS_LDAC
PARAMETERS_NAME sex_coadd.param # name of the file containing catalog contents

#----- Extraction -----

DETECT_TYPE CCD # CCD (linear) or PHOTO (with gamma correction)
DETECT_MINAREA 5 # minimum number of pixels above threshold
THRESH_TYPE RELATIVE # threshold type: RELATIVE (in sigmas)
# or ABSOLUTE (in ADUs)
DETECT_THRESH 0.5 # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
ANALYSIS_THRESH 0.5 # <sigmas> or <threshold>,<ZP> in mag.arcsec-2

FILTER Y # apply filter for detection (Y or N)?
FILTER_NAME gauss_5.0_9x9.conv # name of the file containing the filter
FILTER_THRESH # Threshold[s] for retina filtering

DEBLEND_NTHRESH 32 # Number of deblending sub-thresholds
DEBLEND_MINCONT 0.005 # Minimum contrast parameter for deblending

CLEAN Y # Clean spurious detections? (Y or N)?
CLEAN_PARAM 1.0 # Cleaning efficiency

MASK_TYPE CORRECT # type of detection MASKing: can be one of
# NONE, BLANK or CORRECT

#----- WEIGHTing -----

WEIGHT_TYPE MAP_WEIGHT # type of WEIGHTing: NONE, BACKGROUND,
# MAP_RMS, MAP_VAR or MAP_WEIGHT
WEIGHT_IMAGE coadd.weight.fits # weight-map filename
```


WEIGHT_GAIN Y # modulate gain (E/ADU) with weights? (Y/N)
WEIGHT_THRESH # weight threshold[s] for bad pixels

#----- FLAGging -----

FLAG_IMAGE flag.fits # filename for an input FLAG-image
FLAG_TYPE OR # flag pixel combination: OR, AND, MIN, MAX
or MOST

#----- Photometry -----

PHOT_APERTURES 15 # MAG_APER aperture diameter(s) in pixels
PHOT_AUTOPARAMS 2.5, 3.5 # MAG_AUTO parameters: <Kron_fact>,<min_radius>
PHOT_PETROPARAMS 2.0, 3.5 # MAG_PETRO parameters: <Petrosian_fact>,
<min_radius>
PHOT_AUTOAPERS 0.0,0.0 # <estimation>,<measurement> minimum apertures
for MAG_AUTO and MAG_PETRO
PHOT_FLUXFRAC 0.5 # flux fraction[s] used for FLUX_RADIUS

SATUR_LEVEL 65000.0 # level (in ADUs) at which arises saturation
SATUR_KEY SATURATE # keyword for saturation level (in ADUs)

MAG_ZEROPOINT 24.7562 # magnitude zero-point
MAG_GAMMA 4.0 # gamma of emulsion (for photographic scans)
GAIN 80271.4867 # detector gain in e-/ADU
GAIN_KEY GAIN # keyword for detector gain in e-/ADU
PIXEL_SCALE 0.0 # size of pixel in arcsec (0=use FITS WCS info)

#----- Star/Galaxy Separation -----

SEEING_FWHM 1.65 # stellar FWHM in arcsec
STARNNW_NAME default.nnw # Neural-Network_Weight table filename

#----- Background -----

BACK_TYPE AUTO # AUTO or MANUAL
BACK_VALUE 0.0 # Default background value in MANUAL mode
BACK_SIZE 64 # Background mesh: <size> or <width>,<height>
BACK_FILTERSIZE 3 # Background filter: <size> or <width>,<height>

```

BACKPHOTO_TYPE LOCAL # can be GLOBAL or LOCAL
BACKPHOTO_THICK 24 # thickness of the background LOCAL annulus
BACK_FILTTHRESH 0.0 # Threshold above which the background-
# map filter operates

#----- Check Image -----

CHECKIMAGE_TYPE SEGMENTATION # can be NONE, BACKGROUND, BACK-
GROUND_RMS,
# MINIBACKGROUND, MINIBACK_RMS, -BACKGROUND,
# FILTERED, OBJECTS, -OBJECTS, SEGMENTATION,
# or APERTURES
CHECKIMAGE_NAME check.fits # Filename for the check-image

#----- Memory (change with caution!) -----

MEMORY_OBJSTACK 3000 # number of objects in stack
MEMORY_PIXSTACK 300000 # number of pixels in stack
MEMORY_BUFSIZE 1024 # number of lines in buffer

#----- ASSOCiation -----

ASSOC_NAME sky.list # name of the ASCII file to ASSOCiate
ASSOC_DATA 2,3,4 # columns of the data to replicate (0=all)
ASSOC_PARAMS 2,3,4 # columns of xpos,ypos[,mag]
ASSOC_RADIUS 2.0 # cross-matching radius (pixels)
ASSOC_TYPE NEAREST # ASSOCiation method: FIRST, NEAREST, MEAN,
# MAG_MEAN, SUM, MAG_SUM, MIN or MAX
ASSOCSELEC_TYPE MATCHED # ASSOC selection type: ALL, MATCHED or -MATCHED

#----- Miscellaneous -----

VERBOSE_TYPE NORMAL # can be QUIET, NORMAL or FULL
WRITE_XML N # Write XML file (Y/N)?
XML_NAME sex.xml # Filename for XML output
XSL_URL file:///scisoft/share/sextractor/sextractor.xsl
# Filename for XSL style-sheet
NTHREADS 0 # Number of simultaneous threads for
# the SMP version of SExtractor
# 0 = automatic

```

FITS_UNSIGNED N # Treat FITS integer values as unsigned (Y/N)?
INTERP_MAXXLAG 16 # Max. lag along X for 0-weight interpolation
INTERP_MAXYLAG 16 # Max. lag along Y for 0-weight interpolation
INTERP_TYPE ALL # Interpolation type: NONE, VAR_ONLY or ALL

#————— Experimental Stuff —————

PSF_NAME default.psf # File containing the PSF model
PSF_NMAX 9 # Max.number of PSFs fitted simultaneously
PSFDISPLAY_TYPE SPLIT # Catalog type for PSF-fitting: SPLIT or VECTOR
PATTERN_TYPE RINGS-HARMONIC # can RINGS-QUADPOLE, RINGS-OCTOPOLE,
RINGS-HARMONICS or GAUSS-LAGUERRE
SOM_NAME default.som # File containing Self-Organizing Map weights

D.2.2 Sextractor parameter file

X_IMAGE
Y_IMAGE
ALPHA_J2000
DELTA_J2000
MAG_AUTO
MAGERR_AUTO
FWHM_IMAGE
CLASS_STAR
FLAGS
ELONGATION
ELLIPTICITY

D.3 Zero Point Calculation

```
!topcat  
emacs zeropointinput.txt  
! /cppcodes/zeropointcalculation  
emacs sex_coadd.conf  
!sex coadd.fits -c sex_coadd.conf -WEIGHT_IMAGE coadd.weight.reduced.fits  
!awk 'NR>11print $1,$2' coadd.cat | sed -e 's/ /,/' -e 's/^circle(/' -e 's/$/,7.5)/' -e '1s//image\n/'
```

```
> coadd.reg
```

D.3.1 Zero point C++ code

```
//this program is going to be used to calculate zeropoint for z band only  
//using my combined sdss and my calculation.
```

```
//what you need to run it. currently will need the output from topcat after  
//matching sdss and coadd sextractor sources and have the magzero column
```

```
//should have 32 columns in total
```

```
//stuff that will need to be added over time of this script:  
//1) clipping program to kill it if there is less than x number of sources  
//2) ability to do different bands  
//3) ability to do Y band data - may need a seperate script  
//4) ability to do all of the different clippings, not just the z band ones.  
// -may have different lower limits, worth exploring.
```

```
#include <iostream>  
#include <fstream>  
#include <assert.h>  
#include <math.h>  
#include "Matrix.h"  
#include <iomanip>  
#include <string>
```

```
using namespace std;
```

```
int main(){
```

```
//constants  
double pi=3.14159, SNRcut10=1.0857/10;  
//*variables  
// dummy variables for loops  
int i, j, k, m, n, inputcol;  
//input file names  
string inputfile;  
//diminsion parameters
```

```

int dim, dimcut, savenewdim, newdim;
//outputfiles
ofstream myfile;
//parameters for zeropoint and standard deviation calculations
double magdiffsum, magdiff, magdispdiffsum, magdisp;
//outlier variables
int killme;
double magoutlier, absmagdiff;
//parameters to keep track of what band I am using
string band;
int magcol, magerrcol;
int stopwhileloop;

//correct output precision
cout.precision(16);
myfile.precision(16);

//*Input the coordinate file
cout << "What is the coordinate file name" << endl;
cin >> inputfile;
cout << endl;

//determine dimension
ifstream vector_b(inputfile.c_str());
dim=0;
string line;
while(getline(vector_b , line)){
dim ++;
}

//input the coordinate file into a matrix format.
inputcol=32;
Matrix Input(dim,inputcol);
vector_b.clear();
vector_b.seekg(0);
for(i=1; i<=dim; i++){
for(j=1; j<=inputcol; j++){
vector_b >> Input(i,j);
}
}
}

```

```

/**flagcut and SNRCuts and star cuts need to be done
//err_z<=1.0857/10 and flag=0 and type=6

/**first the star cut. this is column 18
//need input matrix.
Matrix StarOnly(dim,inputcol);

//do star cut with column 18 = 6.
for(i=1; i<=dim; i++){
if(Input(i,18) == 6){
for(j=1; j<=inputcol; j++){
StarOnly(i,j) = Input(i,j);
}
}
}

//put data in file to be checked later. This section will be used as a check
//only.
myfile.open("zeropoint_starcut.cat");
//fill file
for(i=1; i<=dim; i++){
if(StarOnly(i,1) != 0){
for(j=1; j<=inputcol; j++){
myfile << StarOnly(i,j) << " ";
}
myfile << endl;
}
}
myfile.close(); //close file

//apply the flag cut, this is column 9
//matrix with both flag and star cuts
Matrix FlagandStarcuts(dim,inputcol);

//do flag cut with column 9 == 0
for(i=1; i<=dim; i++){
if(StarOnly(i,9) == 0){
for(j=1; j<=inputcol; j++){
FlagandStarcuts(i,j) = StarOnly(i,j);
}
}
}

```

```

}
}
}

//put data in file to be checked later. This section will be used as a check
//only
myfile.open("zeropoint_flagandstarcut.cat");
//fill file
for(i=1; i<=dim; i++){
if(FlagandStarcuts(i,1) !=0){
for(j=1; j<=inputcol; j++){
myfile << FlagandStarcuts(i,j) << " ";
}
myfile << endl;
}
}
myfile.close(); //close file

//Tell it what band I am in so I know how to procede
//following table is where each band and magerr should be
//u - 21
//g - 22
//r - 23
//i - 24
//z - 25
//err_u - 26
//err_g - 27
//err_r - 28
//err_i - 29
//err_z - 30

stopwhileloop=1;
while(stopwhileloop!=2){
cout << "What band are you calculating the zeropoint for" << endl;
cin >> band;
cout << endl;
//fill in the correct column for correct band
if(band == "u"){
magcol=21;
magerrcol=26;

```

```

stopwhileloop=2;
}
else if(band == "g"){
magcol=22;
magerrcol=27;
stopwhileloop=2;
}
else if(band == "r"){
magcol=23;
magerrcol=28;
stopwhileloop=2;
}
else if(band == "i"){
magcol=24;
magerrcol=29;
stopwhileloop=2;
}
else if(band == "z"){
magcol=25;
magerrcol=30;
stopwhileloop=2;
}
else{
cout << "This is not one of the bands please reenter" << endl;
}
}

/*SNRcut10. will need to add in a step to determine which band to use
//for the time being programing this for z band only. err_z is column 30

Matrix SNRcut(dim,inputcol);
for(i=1; i<=dim; i++){
if(FlagandStarcuts(i,magerrcol) <= SNRcut10){
for(j=1; j<=inputcol; j++){
SNRcut(i,j) = FlagandStarcuts(i,j);
}
}
}

myfile.open("zeropoint_SNRcut.cat");

```



```

//fill file for a check
for(i=1; i<=dim; i++){
if(SNRcut(i,1) != 0){
for(j=1; j<=inputcol; j++){
myfile << SNRcut(i,j) << " ";
}
myfile << endl;
}
}
myfile.close();

```

```

**** calculate the original mean and dispersion
**do a count of how many sources I have going into my clipping algorithm
dimcut=0; //star the count of the dim.
for(i=1; i<=dim; i++){
if(SNRcut(i,1) != 0){
dimcut ++;
}
}

```

```

**add up the mag diff.
**will probably redo this section later depending on if i make the magdiff
**column or not
magdiffsum = 0;
for(i=1; i<=dim; i++){
magdiffsum = SNRcut(i,32) + magdiffsum;
}
**calculate the magdiff
magdiff = magdiffsum/dimcut;

```

```

Matrix magdispdiff(dim);
for(i=1; i<=dim; i++){
if(SNRcut(i,1) == 0.0){
magdispdiff(i) = 0.0;
}
else{
magdispdiff(i) = (SNRcut(i,32)-magdiff)*(SNRcut(i,32)-magdiff);
}
}

```

```

//add up the differences
magdispdiffsum = 0.0;
for(i=1; i<=dim; i++){
magdispdiffsum = magdispdiffsum + magdispdiff(i);
}

//calculate the dispersion
magdisp=sqrt(magdispdiffsum/dimcut);

cout << "Finding the zeropoint now" << endl;
cout << dimcut << " " << magdiff << " " << magdisp << endl;
/**OUTLIER STEP

//checkfor outliers. If there is any then they needed.
killme=0; //killme allows to kill the while loop when killme=10
Matrix SaveData(dim,inputcol); //matrix to save data

while(killme != 30){
for(i=1; i<=dim; i++){ //fill the save data matrix
for(j=1; j<=inputcol; j++){
SaveData(i,j) = SNRcut(i,j);
}
}
if(killme == 0.0){
savenewdim = dimcut;
}
else{
savenewdim = newdim; //save the old value of newdim
}
newdim = 0; //will be my new dim value
//check to throw out bad sources
for(i=1; i<=dim; i++){
magoutlier = fabs(SNRcut(i,32));
absmagdiff = fabs(magdiff);
//cout << absmagdiff << " " << magoutlier << " " << absmagdiff+2*magdisp << " "
<< absmagdiff-2*magdisp << endl;
if(magoutlier>=(absmagdiff+2*magdisp) || magoutlier<=(absmagdiff-2*magdisp)){
for(j=1; j<=inputcol; j++){
SNRcut(i,j) = 0;
}
}
}
}

```

```

}
}
else {
newdim = newdim + 1;
}
}
//stop it if newdim gets to small
if(newdim <= 10){
for(i=1; i<=dim; i++){
for(j=1; j<=inputcol; j++){
SNRcut(i,j) = SaveData(i,j);
}
}
killme=29; //needs to be set to 29 so it will be smart and go to 30
newdim=savenewdim;
cout << "WARNING: I threw out too many sources so I had to stop" << endl;
}

//recalculate the offset and dispersion using the newdim values
//note because of how the matrix works go ahead and leave it as dim in the
//for loops. use newdim in the calculations since the bad values are now 0.

//calculate the average
//first get the sum
magdiffsum = 0; //reset the magdiffsum parameter
for(i=1; i<=dim; i++){
magdiffsum = SNRcut(i,32) + magdiffsum;
}
//next get the average
magdiff = magdiffsum/newdim;

//need to calculate the new dispersion value
//reset matrix magdispdiff to 0
for(i=1; i<=dim; i++){
if(SNRcut(i,1)==0.0){
magdispdiff(i)=0.0;
}
else{
magdispdiff(i) = (SNRcut(i,32)-magdiff)*(SNRcut(i,32)-magdiff);
}
}

```

```

}

//add up the differences
magdispdiffsum = 0.0;
for(i=1; i<=dim; i++){
magdispdiffsum = magdispdiffsum + magdispdiff(i);
}
//calculate the dispersion
magdisp=sqrt(magdispdiffsum/newdim);

//kill the while loop step
killme=killme+1;
cout << newdim << " " << magdiff << " " << magdisp << endl;
}

//need to keep record of sources used
myfile.open("zeropoint_sources_used.cat");
//file file
for(i=1; i<=dim; i++){
if(SNRcut(i,1) != 0){
for(j=1; j<=inputcol; j++){
myfile << SNRcut(i,j) << " ";
}
myfile << endl;
}
}
myfile.close(); //close file

} //end of program

```

D.3.2 Matrix.h

This file is needed to run many of the C++ codes in this thesis. texttt #include <assert.h>
// Defines the assert function.

```

class Matrix {

public:

```

```

// Default Constructor. Creates a 1 by 1 matrix; sets value to zero.
Matrix () {
nRow_ = 1; nCol_ = 1;
data_ = new double [1]; // Allocate memory
set(0.0); // Set value of data_[0] to 0.0
}

// Regular Constructor. Creates an nR by nC matrix; sets values to zero.
// If number of columns is not specified, it is set to 1.
Matrix(int nR, int nC = 1) {
assert(nR > 0 && nC > 0); // Check that nC and nR both > 0.
nRow_ = nR; nCol_ = nC;
data_ = new double [nR*nC]; // Allocate memory
assert(data_ != 0); // Check that memory was allocated
set(0.0); // Set values of data_[] to 0.0
}

// Copy Constructor.
// Used when a copy of an object is produced
// (e.g., passing to a function by value)
Matrix(const Matrix& mat) {
this->copy(mat); // Call private copy function.
}

// Destructor. Called when a Matrix object goes out of scope.
Matrix() {
delete [] data_; // Release allocated memory
}

// Assignment operator function.
// Overloads the equal sign operator to work with
// Matrix objects.
Matrix& operator=(const Matrix& mat) {
if( this == &mat ) return *this; // If two sides equal, do nothing.
delete [] data_; // Delete data on left hand side
this->copy(mat); // Copy right hand side to l.h.s.
return *this;
}

// Simple "get" functions. Return number of rows or columns.

```

```

int nRow() const { return nRow_; }
int nCol() const { return nCol_; }

// Parenthesis operator function.
// Allows access to values of Matrix via (i,j) pair.
// Example: a(1,1) = 2*b(2,3);
// If column is unspecified, take as 1.
double& operator() (int i, int j = 1) {
assert(i > 0 && i <= nRow_); // Bounds checking for rows
assert(j > 0 && j <= nCol_); // Bounds checking for columns
return data_[ nCol_*(i-1) + (j-1) ]; // Access appropriate value
}

// Parenthesis operator function (const version).
const double& operator() (int i, int j = 1) const{
assert(i > 0 && i <= nRow_); // Bounds checking for rows
assert(j > 0 && j <= nCol_); // Bounds checking for columns
return data_[ nCol_*(i-1) + (j-1) ]; // Access appropriate value
}

// Set function. Sets all elements of a matrix to a given value.
void set(double value) {
int i, iData = nRow_*nCol_;
for( i=0; i<iData; i++ )
data_[i] = value;
}

/*****
private:

// Matrix data.
int nRow_, nCol_; // Number of rows, columns
double* data_; // Pointer used to allocate memory for data.

// Private copy function.
// Copies values from one Matrix object to another.
void copy(const Matrix& mat) {
nRow_ = mat.nRow_;
nCol_ = mat.nCol_;
int i, iData = nRow_*nCol_;

```

```

data_ = new double [iData];
for(i = 0; i<iData; i++)
data_[i] = mat.data_[i];
}

}; // Class Matrix

```

D.4 Upperlimit calculation

```

!ds9 -zscale -invert coadd.fits &
! /cppcodes/xyrandomnumber
!awk '{print $1,$2}' upperlimcoords.txt | sed -e 's/ /,/ ' -e 's/^/circle(/' -e 's/$/,7.5)/' -e
'1s/^/image\n/' > upperlimcoords.reg
!ds9 -zscale -invert coadd.fits -region upperlimcoords.reg
hselect coadd.fits $I,gain yes
epar phot
pdump upperlimcoords.mag xcenter,ycenter,sum yes > upperliminput.txt
ls
! /cppcodes/upperlimitclipping
!awk '{print $1,$2}' upperlimitclippingcheck.txt | sed -e 's/ /,/ ' -e 's/^/circle(/' -e 's/$/,7.5)/'
-e '1s/^/image\n/' > upperlimitclippingcheck.reg
!ds9 -zscale -invert coadd.fits -regions upperlimitclippingcheck.reg
cp ../gettingthedatasteps/sex_coadd2.param .
!sex coadd.fits -c sex_coadd.conf -CATALOG_NAME coadd2.cat -PARAMETERS_NAME
sex_coadd2.param -WEIGHT_IMAGE coadd.weight.reduced.fits

```

D.4.1 xyrandomnumber.cpp

```

// rand example: guess the number
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>
#include <fstream>

```

```

using namespace std;
int main(){
int x, y, i;
int x1, x2, y1, y2;
int xdiff, ydiff;
ofstream coordinatelist;
coordinatelist.open("upperlimcoords.txt");

srand (time(NULL)); //randomizing based off time

//input range from user

cout << "Please input the smallest x value" << endl;
cin >> x1;
cout << endl;

cout << "Please input the largest x value" << endl;
cin >> x2;
cout << endl;

cout << "Please input the smallest y value" << endl;
cin >> y1;
cout << endl;

cout << "Please input the largest y value" << endl;
cin >> y2;
cout << endl;

//calculate the difference
xdiff=x2-x1;
ydiff=y2-y1;

//position
for(i=1; i<=1000; i++){
x=rand() y=rand()
coordinatelist << x << " " << y << endl;
}
coordinatelist.close();

```


}

D.4.2 phot

image = "coadd.fits" The input image(s)
skyfile = "" The input sky file(s)
(coords = "upperlimcoords.txt") The input coordinate files(s) (default: image.coo.?)
(output = "coadd.mag") The output photometry file(s) (default: image.mag.?)
(plotfile = "") The output plots metacode file
(datapars = "") Data dependent parameters
(centerpars = "") Centering parameters
(fitskypars = "") Sky fitting parameters
(photpars = "") Photometry parameters
(interactive = no) Interactive mode ?
(radplots = no) Plot the radial profiles in interactive mode ?
(icommands = "") Image cursor: [x y wcs] key [cmd]
(gcommands = "") Graphics cursor: [x y wcs] key [cmd]
(wcsin = "_wcsin") The input coordinate system (logical,tv,physical,world)
(wcsout = "_wcsout") The output coordinate system (logical,tv,physical)
(cache = "_cache") Cache the input image pixels in memory ?
(verify = no) Verify critical parameters in non-interactive mode ?
(update = "_update") Update critical parameters in non-interactive mode ?
(verbose = "_verbose") Print messages in non-interactive mode ?
(graphics = "_graphics") Graphics device
(display = "_display") Display device
(mode = "ql")

datapars

(scale = 1.0) Image scale in units per pixel
(fwhmpsf = 5.0) FWHM of the PSF in scale units
(emission = yes) Features are positive ?
(sigma = INDEF) Standard deviation of background in counts
(datamin = INDEF) Minimum good data value
(datamax = INDEF) Maximum good data value
(noise = "poisson") Noise model
(ccdread = "") CCD readout noise image header keyword

(gain = "") CCD gain image header keyword
(readnoise = 5.0) CCD readout noise in electrons
(epadu = 34888.267) Gain in electrons per count
(exposure = "") Exposure time image header keyword
(airmass = "") Airmass image header keyword
(filter = "") Filter image header keyword
(obstime = "") Time of observation image header keyword
(itime = 1.0) Exposure time
(xairmass = INDEF) Airmass
(ifilter = "INDEF") Filter
(otime = "INDEF") Time of observation
(mode = "ql")

centerpars

(calgorithm = "none") Centering algorithm
(cbox = 5.0) Centering box width in scale units
(cthreshold = 0.0) Centering threshold in sigma above background
(minsratio = 1.0) Minimum signal-to-noise ratio for centering algorithm
(cmaxiter = 10) Maximum number of iterations for centering algorithm
(maxshift = 1.0) Maximum center shift in scale units
(clean = no) Symmetry clean before centering ?
(rclean = 1.0) Cleaning radius in scale units
(rclip = 2.0) Clipping radius in scale units
(kclean = 3.0) Rejection limit in sigma
(mkcenter = no) Mark the computed center on display ?
(mode = "ql")

fitskypars

(salgorithm = "mode") Sky fitting algorithm
(annulus = 15.0) Inner radius of sky annulus in scale units
(dannulus = 10.0) Width of sky annulus in scale units
(skyvalue = 0.0) User sky value
(smaxiter = 10) Maximum number of sky fitting iterations
(sloclip = 0.0) Lower clipping factor in percent
(shiclip = 0.0) Upper clipping factor in percent
(snreject = 50) Maximum number of sky fitting rejection iterations

(sloreject = 3.0) Lower K-sigma rejection limit in sky sigma
(shireject = 3.0) Upper K-sigma rejection limit in sky sigma
(khist = 3.0) Half width of histogram in sky sigma
(binsize = 0.1) Binsize of histogram in sky sigma
(smooth = no) Boxcar smooth the histogram
(rgrow = 0.0) Region growing radius in scale units
(mksky = no) Mark sky annuli on the display
(mode = "al")

photpars

(weighting = "constant") Photometric weighting scheme for wphot
(apertures = "7.5") List of aperture radii in scale units
(zmag = 23.7956) Zero point of magnitude scale
(mkapert = no) Draw apertures on the display
(mode = "ql")

D.5 Upperlimitclipping algorithm c++

//this program is used to try and determine the upperlimit sum and standard
//deviation the input will come from phot output sum column.

```
#include <iostream>
#include <fstream>
#include <assert.h>
#include <math.h>
#include "Matrix.h"
#include <iomanip>
#include <string>

using namespace std;

int main(){

//constants
double pi=3.14159;
//variables
// dummy variables for loops
```

```

int i, j, k, m, n, inputcol;
//input file names
string inputfile;
//diminsion parameters
int dim, dimcut, savenewdim, newdim;
//outputfiles
ofstream myfile;
//parameters for sum and standard deviation calculations
double sum, average, dispdiffsum, disp;
//outlier variables
int killme;
double outlier, absdiff;
double sigma;

//correct output precision
cout.precision(16);
myfile.precision(16);

//Input the coordinate file
cout << "What is the coordinate file name" << endl;
cin >> inputfile;
cout << endl;

//Input the sigma for clipping
cout << "What value for clipping do you want me to use?" << endl;
cin >> sigma;
cout << endl;

//determine dimension
ifstream vector_b(inputfile.c_str());
dim=0;
string line;
while(getline(vector_b , line)){
dim++;
}

//input the coordinate files into a matrix format
Matrix Input(dim,3);
vector_b.clear();
vector_b.seekg(0);

```

```

for(i=1; i<=dim; i++){
for(j=1; j<=3; j++){
vector_b >> Input(i,j);
}
}

// calculate the original mean and dispersion
// do a count of how many sources I have going into my clipping algorithm
dimcut=0; //how many detections are not equal to 0.
for(i=1; i<=dim; i++){
if(Input(i,3) != 0){
dimcut ++;
}
}

//calculating the mean
sum = 0;
for(i=1; i<=dim; i++){
sum = Input(i,3) + sum;
}
average=sum/dimcut; //will give the mean

//calculate the standard deviation
Matrix dispdiff(dim);
for(i=1; i<=dim; i++){
if(Input(i,3) == 0.0){
dispdiff(i) = 0.0;
}
else{
dispdiff(i) = (Input(i,3)-average)*(Input(i,3)-average);
}
}

//add up the differences
dispdiffsum = 0.0;
for(i=1; i<=dim; i++){
dispdiffsum = dispdiffsum + dispdiff(i);
}

//calculate the dispersion

```

```

disp=sqrt(dispdiffsum/dimcut);

cout << dimcut << " " << average << " " << disp << endl;

//check for outliers.
killme=0; //killme allows to kill the while loop when killme=30
Matrix SaveData(dim,3); //matrix to save data

while(killme != 50){ //try and do clipping 30 times
for(i=1; i<=dim; i++){
for(j=1; j<=3; j++){
SaveData(i,j) = Input(i,j);
}
}
//dim book keeping
if(killme==0.0){
savenewdim = dimcut;
}
else{
savenewdim = newdim;
}

newdim=0.0; //will be my new dim value
//check to throw out bad sources
for(i=1; i<=dim; i++){
outlier = Input(i,3);
absdiff = average;
if(outlier>=(absdiff+sigma*disp) || outlier<=(absdiff-sigma*disp) || Input(i,3)==0.0){
Input(i,3) = 0.0;
}
else{
newdim=newdim+1;
}
}

//stop if newdim gets too small
if(newdim<=10){
for(i=1; i<=dim; i++){
Input(i,1)=SaveData(i,1);
}
}

```

```

}
killme=49; //needs to be set to 29 so it will be smart and go to 30
newdim=savenewdim;
cout << "WARNING: I threw out too many sources so I had to stop" << endl;
}

//recalculate the offset and dispersion using the newdim values
//note because of how the matrix works go ahead and leave it as dim in the
//for loops. use newdim in the calculations since the bad values are now 0.

//calculate the average
sum = 0; //reset the difffsum parameter
for(i=1; i<=dim; i++){
sum = Input(i,3) + sum;
}
//next to get the average
average=sum/newdim;

//need to calculate the new dispersion value
//reset matrix dispdiff to 0
for(i=1; i<=dim; i++){
if(Input(i,3)==0.0){
dispdiff(i)=0.0;
}
else{
dispdiff(i)=(Input(i,3)-average)*(Input(i,3)-average);
}
}

//add up the differences
dispdiffsum = 0.0;
for(i=1; i<=dim; i++){
dispdiffsum = dispdiffsum+dispdiff(i);
}
//calculate the dispersion
disp=sqrt(dispdiffsum/newdim);

//kill the while loop setp
killme=killme+1;

```

```
cout << newdim << "" << average << "" << disp << endl;

} //end while loop

//check of coordinates
myfile.open("upperlimitclippingcheck.txt");

for(i=1; i<=dim; i++){
if(Input(i,3) != 0){
for(j=1; j<=3; j++){
myfile << Input(i,j) << " ";
}
myfile << endl;
}
}

myfile.close();

} //end file
```


Appendix E

Lephare

```
#####  
# CREATION OF LIBRARIES FROM SEDs List #  
# $LEPHAREDIR/source/sedtolib -t (S/Q/G) -c $LEPHAREDIR/config/zphot.para #  
# help : $LEPHAREDIR/source/sedtolib -h (or -help) #  
#####  
#  
#—— STELLAR LIBRARY (ASCII SEDs)  
STAR_SED $LEPHAREDIR/sed/STAR/STAR_MOD.list # STAR list (full path)  
STAR_FSCALE 3.432E-09 # Arbitrary Flux Scale  
STAR_LIB LIB_STAR # Bin. STAR LIBRARY ->  
# $LEPHAREWORK/lib_bin  
#  
#—— QSO LIBRARY (ASCII SEDs)  
QSO_SED $LEPHAREDIR/sed/QSO/QSO_MOD.list # QSO list (full path)  
QSO_FSCALE 1 # Arbitrary Flux Scale  
QSO_LIB LIB_QSO # Bin. QSO LIBRARY ->  
# $LEPHAREWORK/lib_bin  
#  
#—— GALAXY LIBRARY (ASCII or BINARY SEDs)  
GAL_SED $LEPHAREDIR/sed/GAL/BC03_CHAB/BC03_MOD.list # GAL list (full path)  
GAL_FSCALE 1. # Arbitrary Flux Scale  
GAL_LIB LIB_BC03 # Bin. GAL LIBRARY ->  
# $LEPHAREWORK/lib_bin  
#SEL_AGE $LEPHAREDIR/sed/GAL/BC03_CHAB/BC03_AGE.list # Age list(full path)  
# (def=NONE)  
AGE_RANGE 0.,13.e9 # Age Min-Max in yr  
#
```

```

#####
# FILTERS #
# $LEPHAREDIR/source/filter -c $LEPHAREDIR/config/zphot.para #
# help: $LEPHAREDIR/source/filter -h (or -help) #
#####
# Filter number and context
# f300 f450 f606 f814 J H K
# 1 2 3 4 5 6 7
# 1 2 4 8 16 32 64 = 127
#
FILTER_LIST sdss/gp.pb,sdss/rp.pb,sdss/ip.pb,sdss/zp.pb,ukidss/Y.pb,ukidss/J.pb,ukidss/H.pb,
ukidss/K.pb,wise/RSR-W1.txt,wise/RSR-W2.txt,wise/RSR-W3.txt,wise/RSR-W4.txt
# (in $LEPHAREDIR/filt/*)
TRANS_TYPE 0 # TRANSMISSION TYPE
# 0[-def]: Energy, 1: Nb of photons
FILTER_CALIB 0 # 0[-def]: fnu=ctt
# 1 : nu.fnu=ctt
# 2 : fnu=nu
# 3 : fnu=Black Body @ T=10000K
# 4 : for MIPS (leff with nu fnu=ctt and flux with BB @ 10000K
FILTER_FILE HDF.filt # output name of filter's file ->
# $LEPHAREWORK/filt/
#
#####
# THEORETICAL MAGNITUDES #
# $LEPHAREDIR/source/mag_star -c $LEPHAREDIR/config/zphot.para (star only)#
# help: $LEPHAREDIR/source/mag_star -h (or -help) #
# $LEPHAREDIR/source/mag_gal -t (Q or G) -c $LEPHAREDIR/config/zphot.para #
# (for gal. & QSO) #
# help: $LEPHAREDIR/source/mag_gal -h (or -help) #
#####
#
#----- From STELLAR LIBRARY
STAR_LIB_IN LIB_STAR # Input STELLAR LIBRARY in $LEPHAREWORK/lib_bin/
STAR_LIB_OUT STAR_HDF # Output STELLAR MAGN -> $LEPHAREWORK/lib_mag/
#
#----- From QSO LIBRARY
QSO_LIB_IN LIB_QSO # Input QSO LIBRARY in $LEPHAREWORK/lib_bin/
QSO_LIB_OUT QSO_HDF # Output QSO MAGN -> $LEPHAREWORK/lib_mag/
#

```

```

#----- From GALAXY LIBRARY
GAL_LIB_IN LIB_BC03 # Input GAL LIBRARY in $LEPHAREWORK/lib_bin/
GAL_LIB_OUT BC03_HDF # Output GAL LIBRARY -> $LEPHAREWORK/lib_mag/
#
#----- MAG + Z_STEP + EXTINCTION + COSMOLOGY
MAGTYPE AB # Magnitude type (AB or VEGA)
Z_STEP 0.04,6.,0.1 # dz, zmax, dzsup(if zmax>6)
COSMOLOGY 70,0.3,0.7 # H0,om0,lbd0 (if lb0>0->om0+lbd0=1)
MOD_EXTINC 0,20 # model range for extinction
EXTINC_LAW calzetti.dat # ext. law (in $LEPHAREDIR/ext/*)
EB_V 0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9, 1.0,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0# E(B-
V) (<50 values)
EM_LINES NO
# Z_FORM 8,7,6,5,4,3 # Zformation for each SED in GAL_LIB_IN
#
#----- ASCII OUTPUT FILES OPTION
LIB_ASCII NO # Writes output in ASCII
# in working directory
#
#####
# PHOTOMETRIC REDSHIFTS #
# $LEPHAREDIR/source/zphot -c $LEPHAREDIR/config/zphot.para #
# help: $LEPHAREDIR/source/zphot -h (or -help) #
#####
#
#
#----- Input Catalog Informations
CAT_IN /run/media/jacob/astro2/research/osmos_datared_stacking_testdir/lephare/SEDlist_v6.1.txt
# Input catalog (full path)
INP_TYPE M # Input type (F:Flux or M:MAG)
CAT_MAG AB # Input Magnitude (AB or VEGA)
CAT_FMT MEME # MEME: (Mag,Err)i
# MMEE: (Mag)i,(Err)i
CAT_LINES 1,100000 # MIN and MAX RANGE of ROWS used in input cat [def:-99,-99]
CAT_TYPE SHORT # Input Format (LONG,SHORT-def)
CAT_OUT /run/media/jacob/astro2/research/osmos_datared_stacking_testdir/lephare/zphot.out
# Output catalog (full path)
PARA_OUT $LEPHAREDIR/config/zphot_output.para # Ouput parameter (full path)
#

```

```

BD_SCALE 0 # Bands used for scaling
# (Sum 2^n; n=0->nbd-1, 0[-def]:all bands)
GLB_CONTEXT -1 # Overwrite Context (Sum 2^n; n=0->nbd-1,
# 0 : all bands used,
# -1[-def]: used context per object
# FORB_CONTEXT -1 # context for forbidden bands
# ERR_SCALE 0.03,0.02,0.02,0.02,0.04,0.04,0.04 # errors per band added in quadrature
ERR_FACTOR 1.5 # error scaling factor 1.0 [-def]
#
#----- Theoretical libraries
ZPHOTLIB BC03_HDF,STAR_HDF,QSO_HDF # Library used for Chi2 (max:3)
ADD_EMLINES NO
#
##### PHOTOMETRIC REDSHIFTS OPTIONS #####
# FIR LIBRARY
FIR_LIB NONE
FIR_LMIN 7.0 # Lambda Min (micron) for FIR analysis
FIR_CONT -1
FIR_SCALE -1
FIR_FREESCALE YES # ALLOW FOR FREE SCALING
FIR_SUBSTELLAR NO
# PHYSICAL LIBRARY with Stochastic models from BC07
PHYS_LIB NONE
PHYS_CONT -1
PHYS_SCALE -1
PHYS_NMAX 100000
#
#----- Priors
# MASS_SCALE 6.,16. # Lg(Scaling) min,max [0,0-def]
MAG_ABS -10.,-26. # Mabs_min , Mabs_max [0,0-def]
MAG_REF 4 # Reference number for band used by Mag_abs
# ZFORM_MIN 5,5,5,5,5,5,3,1 # Min. Zformation per SED -> Age constraint
Z_RANGE 0.,99.99 # Z min-max used for the Galaxy library
EBV_RANGE 0,9 # E(B-V) MIN-MAX RANGE of E(B-V) used
# NZ_PRIOR 4,2,4 # I Band for prior on N(z)
#
#----- Fixed Z (need format LONG for input Cat)
ZFIX NO # fixed z and search best model [YES,NO-def]
#
#----- Parabolic interpolation for Zbest

```

```

Z_INTERP YES # redshift interpolation [YES,NO-def]
#
#----- Analysis of normalized ML( $\exp(-0.5*\text{Chi}^2)$ ) curve
#----- Secondary peak analysis
DZ_WIN 0.5 # Window search for 2nd peaks [0->5;0.25-def]
MIN_THRES 0.1 # Lower threshold for 2nd peaks[0->1; 0.1-def]
#
#----- Probability (in # PROB_INTZ 0,0.5,0.5,1.,1.,1.5 # even number
#
##### ABSOLUTE MAGNITUDES COMPUTATION #####
#
MABS_METHOD 1 # 0[-def] : obs->Ref
# 1 : best obs->Ref
# 2 : fixed obs->Ref
# 3 : mag from best SED
# 4 : Zbin
MABS_CONTEXT -1 # CONTEXT for Band used for MABS

MABS_REF 4 # 0[-def]: filter obs chosen for Mabs :
# ONLY USED IF MABS_METHOD=2
MABS_FILT 1,2,3,4 # Chosen filters per redshift bin (MABS_ZBIN)
# ONLY USED IF MABS_METHOD=4
MABS_ZBIN 0,0.5,1,1.5,2,3,3.5,4 # Redshift bins (even number)
# ONLY USED IF MABS_METHOD=4
##### OUTPUT SPECTRA #####
#
SPEC_OUT YES # spectrum for each object? [YES,NO-def]
CHI2_OUT NO # output file with all values : z,mod,chi2,E(B-V),...
# BE CAREFUL can take a lot of space !!
##### OUTPUT PDZ ANALYSIS
PDZ_OUT NONE # pdz output file name [def-NONE]
# add automatically PDZ_OUT[.pdz/.mabsx/.mod/.zph]
PDZ_MABS_FILT 2,10,14 # MABS for REF FILTERS to be extracted
#
##### FAST MODE : color-space reduction #####
#
FAST_MODE NO # Fast computation [NO-def]
COL_NUM 3 # Number of colors used [3-def]
COL_SIGMA 3 # Enlarge of the obs. color-errors[3-def]
COL_SEL AND # Combination between used colors [AND/OR-def]

```

```

#
##### MAGNITUDE SHIFTS applied to libraries #####
#
# APPLY_SYSSHIFT 0. # Apply systematic shifts in each band
# used only if number of shifts matches
# with number of filters in the library
#
##### ADAPTIVE METHOD using Z spectro sample ###
#
AUTO_ADAPT NO # Adapting method with spectro [NO-def]
ADAPT_BAND 4,2,4 # Reference band, band1, band2 for color
ADAPT_LIM 18,22.0 # Mag limits for spectro in Ref band [18,21.5-def]
ADAPT_POLY 1 # Number of coef in polynom (max=4) [1-def]
ADAPT_METH 1 # Fit as a function of
# 1 : Color Model [1-def]
# 2 : Redshift
# 3 : Models
ADAPT_CONTEXT -1 # Context for bands used for training
# -1[-def] used context per object
ADAPT_ZBIN 0.01,6 # Redshift's interval used for training
# [0.001,6-Def]
ADAPT_MODBIN 1,1000 # Model's interval used for training
# [1,1000-Def]
ERROR_ADAPT NO # [YES,NO-def]
# Add error in quadrature according to
# the difference between observed
# and predicted apparent magnitudes
#

```

Appendix F

IR Power Law C++ code

```
//Linear Fit for a set of wavelenghts and fluxes
//input is wavelength and AB mag

//final output will give me something in the form of
//log10(Fobs_mu) = log(a bunch of constants) - alpha*log10(mu_obs)
//slope will end up giving alpha
//intercept will give log(a bunch of constrants)

//output file for data will give me f_mu obs

#include<iostream>
#include<iomanip>
#include<cmath>
#include<fstream>
#include <string>
#include "Matrix.h"
#include<stdio.h>
#include<math.h>
using namespace std;
int main()
{
//variables
int dim; //tells me the dimension of input matrix
int i, j, k, l, m; //for loop counters
ofstream myfile, myfile2, myfile3; //output files
double z; // redshift parameter
string line, inputfile;
```

```

double xsum, ysum, x2sum, xysum, y2sum;
double a, b;
double yy, chisqr, delta;

//how many z steps I want
double zstep=0.05;

//calculate how many times this needs to run from z of 0 to z of 6
double zrun=6/0.05;

//correct output precision
cout.precision(16);

//Input the coordinate file
cout << "What is the coordinate file name" << endl;
cin >> inputfile;
cout << endl;

//deterime dimension
ifstream vector_b(inputfile.c_str());
dim=0;
while(getline(vector_b, line)){
dim++;
}

//define matrixes
Matrix obsfreq(dim,1);
Matrix restfreq(dim,1);
Matrix obsflux(dim,1);
Matrix restflux(dim,1);
Matrix logobsfreq(dim,1);
Matrix logobsflux(dim,1);
Matrix logrestfreq(dim,1);
Matrix logrestflux(dim,1);

//input the data into a matrix format
Matrix Input(dim,3);
vector_b.clear();
vector_b.seekg(0);

```



```

for(i=1; i<=dim; i++){
for(j=1; j<=3; j++){
vector_b >> Input(i,j);
}
}

//break up the input matrix into something more useable
Matrix wavelength(dim,1);
Matrix ABmag(dim,1);
Matrix error(dim,1);
for(i=1; i<=dim; i++){
wavelength(i,1)=Input(i,1)*pow(10,-6); //gives wavelength in meter
ABmag(i,1)=Input(i,2);
error(i,1)=Input(i,3)/2.5;
}

//convert wavelength to obs frequency
for(i=1; i<=dim; i++){
obsfreq(i,1)=3*pow(10,8)/wavelength(i,1);
}

//convert ABmag to flux
for(i=1; i<=dim; i++){
obsflux(i,1)=pow(10,23-(ABmag(i,1)+48.6)/2.5);
}

//loop to run through all the different redshifts from 0 to zrun
for(m=1; m<=zrun; m++){

//calculate the restframe frequency
z=m*zstep;
cout << z << endl;
for(i=1; i<=dim; i++){
restfreq(i,1)=obsfreq(i,1)*(1+z);
}

//calculate the rest fram flux
for(i=1; i<=dim; i++){
restflux(i,1)=obsflux(i,1)/(1+z);
}

```

```

//calculate the log10 of the flux and frequency
for(i=1; i<=dim; i++){
logrestfreq(i,1)=log10(restfreq(i,1));
logrestflux(i,1)=log10(restflux(i,1));
logobsfreq(i,1)=log10(obsfreq(i,1));
logobsflux(i,1)=log10(obsflux(i,1));
}

//set each of the sum terms to 0;
xsum=0.0;
ysum=0.0;
x2sum=0.0;
xysum=0.0;
y2sum=0.0;

//sum up each sum
for(i=1; i<=dim; i++){
xsum=xsum+logrestfreq(i,1);
ysum=ysum+logrestflux(i,1);
x2sum=x2sum+pow(logrestfreq(i,1),2);
xysum=xysum+logrestfreq(i,1)*logrestflux(i,1);
y2sum=y2sum+pow(logrestflux(i,1),2);
}

//calculate a and b terms
a=(dim*xysum-ysum*xsum)/(dim*x2sum-xsum*xsum); //slope
b=(ysum*x2sum-xsum*xysum)/(dim*x2sum-xsum*xsum); //y-intercept

cout << "slope = " << a << endl;
cout << "y-intercept = " << b << endl;

//calculate chisqr
chisqr = 0.0;
for(i=1; i<=dim; i++){
yy=b+a*logrestfreq(i,1);
delta = logrestflux(i,1)-yy;
chisqr = chisqr + delta*delta/error(i,1);
}
cout << "chisqr = " << chisqr << endl;

```

```

cout << endl;

//output the data used to plot
char title[28];
sprintf(title, "logflux_vs_logfreqmyfile.open(title);
for(i=1; i<=dim; i++){
myfile << logobsfreq(i,1) << " " << logobsflux(i,1) << " ";
myfile << logrestfreq(i,1) << " " << logrestflux(i,1) << " ";
myfile << error(i,1) << endl;
}
myfile.close();

//output the equations
char title2[28];
sprintf(title2, "equationmyfile2.open(title2);
myfile2 << "f(x) = " << a << "*x+" << b << endl;
myfile2.close();

char title3[28];
char output[28];
sprintf(title3, "gnuplot_sprintf(output, "plotmyfile3.open(title3);
myfile3 << "set xrange [13:15]" << endl;
myfile3 << "set yrange [-5.5:-2]" << endl;
myfile3 << "set xlabel log10(frequency_{rest}) (Hz)" << endl;
myfile3 << "set ylabel log10(flux_{mu rest}) (J)" << endl;
myfile3 << "unset key" << endl;
myfile3 << "set term png" << endl;
myfile3 << "set output " << output << "" << endl;
myfile3 << "f(x) = " << a << "*x+" << b << endl;
myfile3 << "plot " << title << "using 3:4, f(x)" << endl;
myfile3.close();
}

} //end file

```

Appendix G

Radio Power Law C++ code

```
//code to calculate the power law from the combine ufc and rfc
//frequencys are in GHz and fluxes are in mJy
//used form  $\log_{10}(F_{obs\_mu}) = \log(\text{a bunch of constants}) - \alpha * \log_{10}(\mu_{obs})$ 
#include <iostream>
#include <fstream>
#include <string>
#include <assert.h>
#include <math.h>
#include <iomanip>
#include "Matrix.h"
#include <cmath>
#include <stdio.h>

using namespace std;

int main()
{

//variables
int i,j; //for loop counters
string part1; //name of the file
string line;
ofstream myfile, myfile2, myfile3; //output files
//variable names from input
double S_flux, S_unres, C_flux, C_unres, X_flux, X_unres, U_flux, U_unres;
double K_flux, K_unres, FIRST_FLUX, FIRST_RMS, NVSS_FLUX, NVSS_RMS;
double GB6_Flux, GB6_RMS, WENSS_Flux, WENSS_RMS, VLSS_FLUX, VLSS_RMS;
```

```

string title1, title2, title3, title4;
string dummy="_logflux_vs_logfreq.txt";
string dummy2="_equation.txt";
string dummy3="_gnuplot.txt";
string dummy4="_plot.png";
string filename;

//least square fit info
double xsum, ysum, x2sum, xysum, y2sum;
double a, b;
double yy, chisqr, delta;
int fewpoints;
int badfilename;

//frequency values in GHz
double first_freq=1.4; //1.4 GHz
double nvss_freq=1.4; //1.4 GHz
double gb6_freq=4.85; //4.85 GHz
double wenss_freq=325.0/1000.0; //325 MHz
double vlss_freq=74.0/1000.0; //74 MHz
double s_freq=2.3; //2.15-2.35 GHz
double c_freq=6; //3.9-7.9 GHz
double u_freq=50; //40-60GHz
double k_freq=22; //21.7-24.1 GHz
double x_freq=8.4; //8-8.8 GHz

badfilename=0;
//Matrix to put stuff in
Matrix funtimes(10,3);
ifstream file;

while(badfilename!=1){
//input the file
cout << "What is the file name?" << endl;
cin >> filename;

file.open(filename.c_str());

//file isnt there
if( !file ){

```

```

cerr << "No file by that name. Try again" << endl;
}
else{
badfilename=1;
}

}

ofstream myfile4;
myfile4.open("whats_my_alpha.txt");

//loop to do calculations in
while( file >> part1 >> S_flux >> S_unres >> C_flux >> C_unres >> X_flux >>
X_unres >> U_flux >> U_unres >> K_flux >> K_unres >> FIRST_FLUX >> FIRST_RMS
>> NVSS_FLUX >> NVSS_RMS >> GB6_Flux >> GB6_RMS >> WENSS_Flux >>
WENSS_RMS >> VLSS_FLUX >> VLSS_RMS ){

//cout << part1 << " " << S_flux << " " << S_unres << " " << C_flux << " "
// << C_unres << " " << X_flux << " " << X_unres << " " << U_flux
// << " " << U_unres << " " << K_flux << " " << K_unres << " "
// << FIRST_FLUX << " " << FIRST_RMS << " " << NVSS_FLUX << " "
// << NVSS_RMS << " " << GB6_Flux << " " << GB6_RMS << " "
// << WENSS_Flux << " " << WENSS_RMS << " " << VLSS_FLUX << " "
// << VLSS_RMS
// << endl;

/***/This section fills in the data into a matrix funtimes. Also ingores
//data points that have no info. left error equal some value so divide
//by 0 doesnt happen

//first data
if(FIRST_FLUX== -99){ //flux == -99 means no data
funtimes(1,1)=0.0; //frequency
funtimes(1,2)=0.0; //flux
funtimes(1,3)=FIRST_RMS; //error
}
else{
funtimes(1,1)=log10(first_freq); //log10 of frequency
funtimes(1,2)=log10(FIRST_FLUX); //log 10 of flux
funtimes(1,3)=0.434*FIRST_RMS/FIRST_FLUX; //error
}
}

```

```

}

//NVSS data
if(NVSS_FLUX==-99){
funtimes(2,1)=0.0;
funtimes(2,2)=0.0;
funtimes(2,3)=NVSS_RMS;
}
else{
funtimes(2,1)=log10(nvss_freq); //frequency of NVSS
funtimes(2,2)=log10(NVSS_FLUX); //flux of NVSS
funtimes(2,3)=0.434*NVSS_RMS/NVSS_FLUX; //RMS of NVSS
}

//GB6_data
if(GB6_Flux==-99){
funtimes(3,1)=0.0; //frequency of GB6
funtimes(3,2)=0.0; //flux of GB6
funtimes(3,3)=GB6_RMS; //RMS of GB6
}
else{
funtimes(3,1)=log10(gb6_freq); //frequency of GB6
funtimes(3,2)=log10(GB6_Flux); //flux of GB6
funtimes(3,3)=0.434*GB6_RMS/GB6_Flux; //RMS of GB6
}

//WENSS_data
if(WENSS_Flux==-99){
funtimes(4,1)=0.0; //frequency of WENSS
funtimes(4,2)=0.0; //flux of WENSS
funtimes(4,3)=WENSS_RMS; //RMS of WENSS
}
else{
funtimes(4,1)=log10(wenss_freq); //frequency of WENSS
funtimes(4,2)=log10(WENSS_Flux); //flux of WENSS
funtimes(4,3)=0.434*WENSS_RMS/WENSS_Flux; //RMS of WENSS
}

//VLSS_Data
if(VLSS_FLUX==-99){

```

```

funtimes(5,1)=0.0; //frequency of VLSS
funtimes(5,2)=0.0; //flux of VLSS
funtimes(5,3)=VLSS_RMS; //RMS of VLSS
}
else{
funtimes(5,1)=log10(vlss_freq); //frequency of VLSS
funtimes(5,2)=log10(VLSS_FLUX); //flux of VLSS
funtimes(5,3)=0.434*VLSS_RMS/VLSS_FLUX; //RMS of VLSS
}

//S band data
if(S_flux==-1.0){
funtimes(6,1)=0.0; //frequency of S band
funtimes(6,2)=0.0; //flux of S band
funtimes(6,3)=S_unres; //RMS of S band
}
else{
funtimes(6,1)=log10(s_freq); //frequency of S band
funtimes(6,2)=log10(S_flux*1000.0); //flux of S band
funtimes(6,3)=0.434*S_unres/S_flux; //RMS of S band
}

//C band data
if(C_flux==-1.0){
funtimes(7,1)=0.0; //frequency of C band
funtimes(7,2)=0.0; //flux of C band
funtimes(7,3)=C_unres; //RMS of C band
}
else{
funtimes(7,1)=log10(c_freq); //frequency of C band
funtimes(7,2)=log10(C_flux*1000.0); //flux of C band
funtimes(7,3)=0.434*C_unres/C_flux; //RMS of C band
}

//U band data
if(U_flux==-1.0){
funtimes(8,1)=0.0; //frequency of U band
funtimes(8,2)=0.0; //flux of U band
funtimes(8,3)=U_unres; //RMS of U band
}

```



```

else{
funtimes(8,1)=log10(u_freq); //frequency of U band
funtimes(8,2)=log10(U_flux*1000.0); //flux of U band
funtimes(8,3)=0.434*U_unres/U_flux; //RMS of U band
}

//K band data
if(K_flux==-1.0){
funtimes(9,1)=0.0; //frequency of K band
funtimes(9,2)=0.0; //flux of K band
funtimes(9,3)=K_unres; //RMS of K band
}
else{
funtimes(9,1)=log10(k_freq); //frequency of K band
funtimes(9,2)=log10(K_flux*1000.0); //flux of K band
funtimes(9,3)=0.434*K_unres/K_flux; //RMS of K band
}

//X band data
if(X_flux==-1.0){
funtimes(10,1)=0.0; //frequency of X band
funtimes(10,2)=0.0; //flux of X band
funtimes(10,3)=X_unres; //RMS of X band
}
else{
funtimes(10,1)=log10(x_freq); //frequency of X band
funtimes(10,2)=log10(X_flux*1000.0); //flux of X band
funtimes(10,3)=0.434*X_unres/X_flux; //RMS of X band
}

**** Since both NVSS and FIRST have same freqdata need to keep only FIRST
if(funtimes(1,1)!=0.0 && funtimes(2,1)!=0.0){
funtimes(2,1)=0.0;
funtimes(2,2)=0.0;
funtimes(2,3)=NVSS_RMS;
} //end if

****check how many useable points we have
fewpoints=0.0; //reset counting mechanism
for(i=1; i<=10; i++){

```

```

if(funtimes(i,1)!=0.0){
fewpoints=fewpoints+1;
}
}

/*time to do the calculation to get alpha
xsum=0.0;
ysum=0.0;
x2sum=0.0;
xysum=0.0;
y2sum=0.0;

//sum up each sum
for(i=1; i<=10; i++){
xsum=xsum+funtimes(i,1);
ysum=ysum+funtimes(i,2);
x2sum=x2sum+pow(funtimes(i,1),2);
xysum=xysum+funtimes(i,1)*funtimes(i,2);
y2sum=y2sum+pow(funtimes(i,2),2);
}

//calculate a and b terms
a=(fewpoints*xysum-ysum*xsum)/(fewpoints*x2sum-xsum*xsum); //slope
b=(ysum*x2sum-xsum*xysum)/(fewpoints*x2sum-xsum*xsum); //y-intercept

cout << "slope = " << a << endl;
cout << "y-intercept = " << b << endl;

//calculate chisqr – not sure if this is good.
chisqr = 0.0;
for(i=1; i<=10; i++){
if(funtimes(i,1)!=0.0){
yy=b+a*funtimes(i,1);
delta = funtimes(i,2)-yy;
chisqr = chisqr + delta*delta/funtimes(i,3);
}
}
cout << "chisqr = " << chisqr << endl;
cout << endl;

```

```

title1 = part1 + dummy; //combines name and subtext
title2 = part1 + dummy2; //title for equation file
title3 = part1 + dummy3;
title4 = part1 + dummy4;

if(fewpoints>=3){
myfile.open(title1.c_str()); //somehow .c_str converts it properly
for(i=1; i<=10; i++){
if(funtimes(i,1)!=0.0){
for(j=1; j<=3; j++){
myfile << funtimes(i,j) << " ";
}
myfile << endl;
}
}
}

myfile.close();

//file for equation
myfile2.open(title2.c_str());
myfile2 << "f(x) = " << a << "*x+" << b << endl;
myfile2.close();

myfile3.open(title3.c_str());
myfile3 << "set xlabel log10(frequency_{rest}) (GHz)" << endl;
myfile3 << "set ylabel log10(flux_{mu rest}) (mJy)" << endl;
myfile3 << "unset key" << endl;
myfile3 << "set term png" << endl;
myfile3 << "set output " << title4 << ".png" << endl;
myfile3 << "f(x) = " << a << "*x+" << b << endl;
myfile3 << "plot " << title1 << "using 1:2, f(x)" << endl;
myfile3.close();

//my output for what my output will be.
//file name, number of points used, alpha
myfile4 << part1 << ".png" << fewpoints << ".png" << a << endl;

} //end if loop

```

```
}//end while loop
```

```
myfile4.close();  
file.close(); //close file  
return 0;  
}//end file
```

Bibliography

- [1] Karl G. Jansky. *Nature*, 132(66), 1933.
- [2] Hubble E. *Proceedings of the National Academy of Sciences*, 15(168), 1929.
- [3] R. H. Becker et al. The VLA's FIRST Survey. *ASPC*, 61(165), 2008.
- [4] D.J. Eisenstien. *AJ*, 142(72E), 2011.
- [5] Donald G. York et al. The Sloan Digital Sky Survey: Technical Summary. *Astron. J.*, 120(3):1579–1587.
- [6] Gordon D. Peck A.B. Petrov L. MacMillan D.S. Fomalont E.B. Ma C. Beasley, A.J. The VLBA Calibrator Survey - VCS1. *Astrophys. J. Sup.*, 141:13–21, 2002.
- [7] Petrov L. McMillan D.S. Gordon D. Ma C. Fomalont, E. The Second VLBA Calibrator Survey: VCS2. *Astron. J.*, 126:2562, 2003.
- [8] Kovalev Yu.Y. Fomalont E. Petrov, L. and D. Gordon. The Third VLBA Calibrator Survey – VCS3. *Astron. J.*, 129:1163, 2005.
- [9] Kovalev Yu.Y. Fomalont E. Petrov, L. and D. Gordon. The Fourth VLBA Calibrator Survey – VCS4. *Astron. J.*, 131:1872, 2006.
- [10] Petrov L. Fomalont E. Kovalev, Yu.Y. and D. Gordon. The Fifth VLBA Calibrator Survey – VCS5. *Astron. J.*, 133:1236, 2007.
- [11] Kovalev Yu.Y. Fomalont E. Petrov, L. and D. Gordon. The Sixth VLBA Calibrator Survey – VCS6. *Astron. J.*, 136:580, 2008.
- [12] Phillips C. Bertarini A. Murphy T. Sadler E.M Petrov, L. The LBA Calibrator Survey of southern compact extragalactic radio sources — LCS1. *Mon. Not. of Roy. Astron. Soc.*, 414(3):2528–2539, 2011.
- [13] Kovalev Y.Y. Fomalont E.B. Gordon D. Petrov, L. The VLBA Galactic Plane Survey — VGaPS. *Astron. J.*, 142:35, 2011.

- [14] Petrov L. Gordon D. Fomalont E.B. Kovalev, Y.Y. The Northern Polar Cap VLBI survey. *to be submitted to Astron. J.*, 2011.
- [15] Brunthaler A. Reid M. J. Bartkiewicz A. Choi Y. K. Menten K. M. Moscadelli L. Sanna A. Wu Y. W. Xu Y. Zhang B. Zheng X. W. Immer, K. The VLBA Calibrator Search for the BeSSeL Survey. *ApJS*, 194:25, 2011.
- [16] Taylor G. Petrov, L. Precise absolute astrometry from the VLBA imaging and polarimetry survey at 5 GHz. *Astron. J.*, 142:89, 2011.
- [17] L. Petrov. The Catalogue of Positions of Optically Bright Extragalactic Radio Sources OBRS-1. *Astron. J.*, 142:105, 2011.
- [18] L. Petrov. The Catalogue of Positions of Optically Bright Extragalactic Radio Sources OBRS-2. *Astron. J.*, 146(5), 2013.
- [19] L. Petrov. The EVN Galactic Plane Survey — EGaPS. *MNRAS*, 416:1097–1105, 2012.
- [20] A. Beasley A. Peck R. Gaume P. Charlot A. Fey C. Ma O. Titov D. Boboltz Gordon D., C. Jacobs. Second Epoch VLBA Calibrator Survey Observations: VCS-II. *Astron. J.*, 151(6), 2016.
- [21] Jiang W. Xia B. Chen Zh. Hao L. Liu X. Takefuji K. McCallum J. Lovell J. Yi S.-O. Shu F., Petrov L. and Li J. The VLBI Ecliptic Plane Survey VEPS-1. arxiv.org/abs/1701.07287, 2017.
- [22] C.P. Ahn et al. THE TENTH DATA RELEASE OF THE SLOAN DIGITAL SKY SURVEY: FIRST SPECTROSCOPIC DATA FROM THE SDSS-III APACHE POINT OBSERVATORY GALACTIC EVOLUTION EXPERIMENT. *ApJS*, 211(2), 2014.
- [23] Edward L. Wright et al. The Wide-field Infrared Survey Explorer (WISE): Mission Description and Initial On-orbit Performance. *AJ*, 140(6), 2010.
- [24] A. Mainzer et al. Preliminary Results from NEOWISE: An Enhancement to the Wide-field Infrared Survey Explorer for Solar System Science. *AJ*, 1(53), 2011.
- [25] A. Lawrence et al. The UKIRT Infrared Deep Sky Survey (UKIDSS). *MNRAS*, 379(4):1599–1617, 2007.
- [26] M. Casali et al. The UKIRT Wide-Field Camera. *Astronomy and Astrophysics*, 467(2):777–784, 2007.

- [27] Will Sutherland et al. The Visible and Infrared Survey Telescope for Astronomy (VISTA): Design, Technical overview, and Performance. *Astronomy and Astrophysics*, 575(A2), 2015.
- [28] Amy E. Kimball and eljko Ivezi. A UNIFIED CATALOG OF RADIO OBJECTS DETECTED BY NVSS, FIRST, WENSS, GB6, AND SDSS. *AJ*, 136(2), 2008.
- [29] J.J Cordon et al. *AJ*, 115(1693), 1998.
- [30] Douglas K. Condon J.J. Gregory P.C., Scott W.K. The GB6 Catalog of Radio Sources.
- [31] Rengelink et al. *AA Supplement Ser.*, 124, 1997.
- [32] van Velzen S. Clarke T.E. Kassim N.E. Helmboldt J.F. Lazio T.J.W. Cohen A.S. Lane W.M., Cotton W.D. The Very Large Array Low-frequency Sky Survey Redux (VLSSr).
- [33] K. C. Chambers et al. The Pan-STARRS1 Surveys.
- [34] Frederick Chromey. *To Measure the Sky: An Introduction to Observational Astronomy*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 8RU, UK, first edition, 2010.
- [35] Martini et al. *PASP*, 123(187), 2011.
- [36] Jarvis M.J. Blundell K.M. Willott C.J., Rawlings S. Near-infrared imaging and the K-z relation for radio galaxies in the 7C Redshift Survey.

VITA

Jacob Brown was born in Watseka, Illinois on August 24th 1986. He attended Western Illinois University where he graduated Magna Cum Laude with his Bachelors' Degrees in Physics and Mathematics in May 2011. In May 2013, he graduated with his Master's in Physics from Western Illinois University. While at Western Illinois University, he worked with Dr. Kishor Kapale on orbital angular momentum of light in the radio range as well as Dr. Araya Esteban on radio masers. He graduated with his Ph.D. in Physics with a Minor in College Teaching from the University of Missouri - Columbia in August 2017. At the University of Missouri, he has worked with Dr. Haojing Yan on galaxy evolution.