Spectroscopic Analysis of Chemical Abundances in Nearby Galaxies

Stephen Morton

Spring 2010

Contents

List of Figures

List of Tables

1 Introduction

Understanding the nature and evolution of galaxies, especially spiral galaxies, is an important task in astronomy. The most commonly used system of classification for spiral galaxies, the Hubble classification, is based on three criteria relating to the morphological type of galaxies: the tightness of the spiral arms, the resolution of the spiral arms, and the ratio of central bulge to disk. The Hubble classification of galaxies forms a sequence running from elliptical galaxies, type E, through late type spiral galaxies, types Sc and Sd, which have loose spiral arms and little in the way of central bulge. At the intermediate stages there are lenticular galaxies, type S0, which have a bulge and a disk, but no spiral structure, and early type spiral galaxies, types Sa and Sb, which are spiral galaxies with large central bulges which are more prominent than the spiral arms. Intermediate stages can be designated by combining the letters of the two closest types, for example type Sab. Other important physical properties of galaxies, which generally correlate with Hubble type, are their total mass, their gas content, metallicity, their bar structure, the environment that a galaxy is contained in, populations of existing stars, and the rates and locations of active star formation within a galaxy [13].

Star formation begins with gravitationally bound clouds of gas. Gas molecules in a cloud frequently interact and radiate energy away. Over time, this causes the cloud to collapse. As the cloud becomes increasingly dense, random variations will cause individual clumps of gas to form. Individual clumps of gas will grow by accretion. The temperature and pressure at the center of this clump of gas will increase along with the mass of the clump. When the temperature at the core increases sufficiently, fusion will begin and the clump of gas will become a star.

Initially, this is hydrogen fusion, via the proton-proton process. This has the lowest temperature requirements of any fusion process, and occurs at temperatures greater than 10^7 K [3]. A star begins composed mostly of hydrogen. As the star evolves, hydrogen supplies in the core are eventually depleted. Helium fusion begins in the core if the mass of the star is high enough to support it, while hydrogen fusion continues in a shell around the core. This process repeats itself, leading to increasingly higher mass fusion products accumulating in the core of the star. Iron is the end product of this sequence: Iron fusion is an endothermic process. The energies required to fuse Iron can not be sustained. At this stage of stellar evolution, the iron core of the star will continue to grow until either fusion stops entirely or, if the star is massive enough, a large enough core of iron accumulates so that the degeneracy pressure supporting it is overcome and it begins to collapse. This triggers a complex series of events which leads to a type II supernova. One of the critical effects of such an event is to release the heavy elements that have formed within the star. This is the source for most elements.

1.1 Metallicity

The only elements not formed exclusively within a star are hydrogen, helium, and, to a negligible extent, lithium. Everything with an atomic number higher than helium is referred to as a metal in astronomy for this reason; these elements represent the accumulated product of previous generations of stellar activity. Tracing the metallicity of a galaxy is therefore a matter of tracing the chemical composition and history, which determines the history and evolution of the galaxies themselves. The abundances can be affected by gas flows (infall, outflow, or internal flows). Infall of gas dilutes the metals already present in a galaxy, but allows for increased star formation. Outflow of gas strips the galaxy of already formed metals. The effects of internal flows of gas on metallicity are complex and depend heavily on the details of the flow.

Two major regimes of metallicity are commonly used: metallicity of stars and metallicity of nebulae and the interstellar medium. The metallicity of nebulae is usually represented by the oxygen abundance relative to hydrogen, O/H (usually quoted as $12 + \log \mathrm{O/H}$), while the metallicity of stars is usually represented by Iron abundance [8]. Investigations of the metallicity of individual stars is limited mostly to studies within the Milky Way, while investigations into the metallicity of nebula can take place over a wide array of distances. In particular, regions of space characterized by ionized hydrogen, H II regions, are a logical target given their high surface brightness and luminosity.

Within the visible spectrum of H II regions it is possible to easily observe emission lines from oxygen, nitrogen, neon, silicon, and argon, although all of these except for oxygen have important ionization states that emit only in the ultraviolet or infrared [8]. Measuring the strengths of these emission lines allows for a look into the abundances of these elements, but it is not a simple picture. Within the ISM, many elements can be strongly depleted onto grains, but the degree to which this is true is poorly known [8]. This means that an unknown proportion of the total abundance of an element may be non-participating in the sort of processes which lead to emission lines. For this reason, $12 + \log \mathrm{O/H}$ represents the proportion of gas phase oxygen relative to gas phase hydrogen. If oxygen is present not in the gas phase, emission line spectra will not account for it.

1.1.1 Measuring Metallicity

Direct abundance measurements are possible when the electron temperature T_e is known, but the emission lines needed to calculate this are faint and difficult to measure. Another problem with the direct method is that ionized nebulae are not actually isothermal, but rather the temperature at any point depends on the local abundance of O^{++} because [O III] is usually the most efficient coolant [8]. Additionally, there is a disagreement between abundances derived from this direct method calibrated with forbidden lines and the direct method calibrated with recombination lines. It is unknown whether the recombination lines or the forbidden lines provide the more reliable abundances [8].

When T_e cannot be measured, the next best method of abundance determination is based on the strong forbidden lines and empirically calibrated to O/H. The most widely used of these strong line calibrations, R_{23} was initially proposed by Pagel, et al 1979 [23]. This is defined as

$$R_{23} = \frac{F([OII]\lambda 3727) + F([OIII]\lambda\lambda 4959, 5007)}{F(H\beta)} \tag{1}$$

This has been well calibrated to O/H using results found using the direct method over the range $8 \leq 12 + \log \mathrm{O/H} \leq 8.5$. For higher metallicities, there are few available measurements using the direct method, so this calibration is impossible [8]. Instead, a calibration based on photoionization models is used. These models are subject to certain additional uncertainties, primarily due to the physical geometry of the nebulae and effects related to dust. Traditionally, H II regions are modeled as uniform spheres of ionized gas surrounding a point source of ionizing photons. This is rarely, if ever, actually the case, based on samples of local, well resolved H II regions. H II regions are more often found to be complicated shapes resembling bubbles with extended distributions of sources of ionization [8]. This has additional effects on the assumption made by direct methods that H II regions are isothermal, given the density dependent nature of collisional cooling.

The distribution of dust is also generally unknown, but can have large effects on the dynamics of ionization in an H II region. Dust can absorb continuum radiation and affect cooling and heating processes within the nebulae. Additionally, obscuration by dust is generally patchy, making the degree to which these effects happen unpredictable and causing greater extinction in some areas than in others. Overall dust has the effect of systematically biasing the results in favor of an interpretation of a higher ages of the stellar population [8].

This dependence on photoionization models is common to all metallicities derived using forbidden line calibrations. R_{23} has also some problems associated specifically with it. Primarily, at $12 + \log \mathrm{O/H} \sim 8.2$ the relation between O/H and R_{23} reverses. On the high metallicity branch, $12 + \log \mathrm{O/H} > 8.2$, R_{23} decreases with increasing metallicity. On the low metallicity branch, R_{23} increases with increasing metallicity. There are much larger uncertainties in $12 + \log \mathrm{O/H}$ in the turn around region and this leads to an ambiguity in the calibration [8]. The ambiguity can be resolved in a number of different ways (see e.g. Kewley & Dopita 2002 [14], Pilyugin, et al. 2004 [24]). I use the empirical disambiguation given by Nagao, et al. 2006 [18], where an R_{23} value is on the upper branch if $F([OIII]\lambda5007)/F([OII]\lambda3727) \leq 2$, and on the lower branch otherwise. Some alternative tracers of oxygen abundance have been proposed to get around these difficulties in R_{23} entirely, but these all have the down sides of either larger scatter or uncertain calibration due to lack of study [8].

1.1.2 The Chemical Composition of Spiral Galaxies

Two quantities are relevant when discussing the chemical composition of a galaxy: the characteristic abundance and the abundance gradient. The choice of characteristic abundance is somewhat arbitrary: I will follow Zaritsky, Kennicutt, and Huchra (1994) (ZKH) in using the metallicity at 0.4 times the isophotal radius, R_{25}, which is the radius of the semi-major axis at the B_{25} mag arcsec^{-2} isophote. The abundance gradient is sensitive to whether galactocentric radius in terms of kiloparsecs, fractional isophotal radius, or fractional scale length is used [32].

Characteristic abundances differ between galaxies, with more scatter found between galaxies than within them. This indicates that abundance is primarily the result of properties global to a galaxy rather than primarily the result of local effects unrelated to the host galaxy. Either abundance is driven directly by macroscopic properties of the host galaxy, or it is driven by local properties that depend on macroscopic ones [32]. Hubble type drives the gas fraction in a galaxy while galactic mass drives the luminosity of the galaxy. It is unclear which of these is more correlated with abundance. If Hubble type is the better correlation, then chemical evolution is likely more determined by environment and initial structure of the galaxy, possibly by way of star formation history and initial mass function. If galactic mass is more closely correlated then processes such as galactic mass loss are likely more important for abundance [32].

The steepness of the spatial abundance gradient has been found to depend on the presence of a bar. In unbarred or weakly barred galaxies, metallicity as a function of galactocentric radius usually exhibits a strong radial gradient with an exponential profile. In strongly barred galaxies, the observed gradient is more shallow [8]. Certain other correlations such as luminosity are present in a abundance gradient in dex/kpc but not in other measures [8].

1.2 Star Formation Rates

1.2.1 Measuring Rates of Star Formation

Given the feedback mechanisms between star formation and metallicity, a discussion of star formation rates (SFR) is relevant. The earliest measurements of star formation rates in spiral galaxies were based on integrated color measurements [13]. This method is based on the fact that an aging stellar population produces a spectral signature that is much redder than that produced by a young stellar population. Measuring rates of star formation on the basis of the integrated colors of a galaxy, then, is an attempt to estimate the ratio of young, blue, recently formed stars to old redder stars.

Except in extreme cases dominated by young stars, a simple linear scaling cannot be assumed and a more complex weighting must be developed by using a evolutionary synthesis model [13]. To generate a synthesis model, first a set of equations for the effective temperatures or bolometric luminosities of various stellar masses as a function of time is obtained [13]. This takes into account the various ways in which a star can evolve over time as a function of its mass. These equations are converted into broadband luminosities or spectra using stellar atmospheric modeling or spectral libraries. Next, these individual stellar templates are added together and weighted using an initial mass function (IMF). This results in equations for the luminosities (or colors, or spectra) of single age stellar populations as a function of age. These can be added together to construct the spectrum for a galaxy with arbitrary star formation history. This process generates a model which has at least four free parameters: the star formation history of the galaxy, the age of the galaxy, the metallicity of the galaxy, and the initial mass function used [13].

Fortunately, the colors of normal galaxies are well modeled with one free parameter: the star formation history of a galaxy with fixed age, metallicity, and initial mass function [13]. This type of model allows for a conversion of color into star formation rates which are usually normalized as star formation per unit mass or star formation per unit luminosity in order to allow for comparisons between galaxies of different sizes. Evolutionary synthesis models are used in the calibration of all method of detecting star formation, not just that of integrated color analysis. In particular, this particular method is prone to systematic error from reddening, incorrect IMF, and imprecision in the age or metallicity of the galaxy, but it can be useful as a means of comparing average star formation rates of large samples where there is not likely to be systematic changes in dust content, abundances, or IMFs [13].

Other techniques for measuring rates of star formation attempt to avoid the limitations of integrated color analysis by using a more direct tracer of star formation; measuring some property more directly linked to populations of young stars. These more direct methods have largely supplanted integrated color measurements [13]. One such method is measuring the ultraviolet continuum, optimally between 1250 and 1500 Å [13]. This range is dominated by young, recently formed star populations, while also avoiding the Lyα forest, a set of absorption lines unrelated to star formation resulting from neutral hydrogen in the intergalactic medium [3]. Failure to avoid this would result in other dependencies being introduced. This range has the drawback of being inaccessible from the ground for $z < 0.5$, but it can be observed for $z \sim 1$–5 from the ground, and can be observed from space based telescopes for nearby galaxies [13].

The UV luminosity can be converted to an SFR using a calibration derived from a synthesis model. Several published calibrations are available, which differ mostly in their choice of stellar libraries and assumptions about the time scales involved. In many situations, the continuous star formation approximation can be applied, in which case it is assumed that the SFR has remained constant over time scales that are long relative to the lifetimes of the UV emitting population ($< 10^8$ years). For younger populations, such as starburst galaxies, this approximation does not hold and a different calibration is needed [13], illustrating the importance of selecting an appropriate calibration for the galaxies in question.

The advantages of using the ultraviolet continuum as a tracer of star formation are that it is directly tied to the photospheric emissions of young stellar populations and it can be applied over a range of redshifts, providing a useful probe of the evolution of SFR [13]. Some of the disadvantages of this technique are its sensitivity to the form of the IMF and to extinction. Extinction in particular is a problem, due to the patchy spatial distribution, which leads to the emissions being dominated by areas of relatively low obscuration [13]. This makes the UV continuum hard to calibrate accurately, and it is best to use additional modeling which takes the clumping into account, and which takes advantage of reddening information derived from the Balmer decrement or from infrared recombination lines.

The dependance on the form of the IMF is a limitation shared by all of the direct tracers of star formation [13]. In the case of UV tracers, the integrated spectrum in the 1500–2500 Å range is dominated by stars with masses greater than $\sim 5\ M_\odot$. As such, there is a large amount of extrapolation to lower masses. Fortunately, there is little evidence for systematic variations in IMF, except for IR starbursts, where UV tracers are unhelpful anyway [13].

Clear trends in nebular emission line strength make it natural to use these recombination lines as a tracer of star formation. These lines re-emit the integrated stellar luminosity shortwards of the Lyman limit, and as such act as a direct probe of the young massive stellar population. The most commonly used recombination line is Hα, but Hβ, Pα, Pβ, Brα, and Brγ can also be used [13]. As with other methods, an evolutionary synthesis model is used to find a calibration. In the case of recombination lines, only stars with masses greater than 10 $M_\odot$ and ages shorter than 20 Myr are significant contributors to the ionizing flux which generates the lines, and as such this method provides a nearly instantaneous measure of SFR independent of star formation history [13]. There is a significant variation among published calibrations, which is largely due to differences in stellar evolution or atmosphere models used.

The advantages of using nebular emission lines are its high sensitivity and the direct coupling between nebular emission and massive SFR. Additionally, the emission distributions of nearby galaxies can be mapped even with small telescopes, and the Hα line can be detected in the redshifted spectra of starburst galaxies out to $z \gg 2$ [13]. Drawbacks of this method are sensitivity to uncertainties in the form of the IMF and in extinction. In addition, this method makes the assumption that all massive star formation is traced by emissions. Studies of the escape fraction of ionizing radiation from H II have been carried out in both directly and through analysis of diffuse Hα emission. Both methods found that the escape fraction is between 15 and 50%, emphasizing the need to include diffuse Hα emission when measuring the SFR of a galaxy [13].

In contrast to the escape fraction from individual H II regions, the escape fraction from the galaxy as a whole should be fairly low. Leitherer et al. (1995a) found the upper limit for escape fraction in starburst galaxies to be 3%. This result needs to be tested against a more diverse sample before it can be taken as fully representative, however [13].

The most important source of systematic error when using nebular emission lines is extinction. It can be measured by comparing Hα fluxes to infrared recombination lines or to the thermal ratio continuum. Using radio flux studies to gauge extinction of Hα, Kennicutt (1983a) and Niklas et al (1997) found a mean extinction of $A(H\alpha) = 0.8$–1.1 mag [13]. Studying samples of individual H II regions in nearby galaxies yields similar results, with a mean extinction of $A(H\alpha) = 0.5$–1.8 mag [13].

Localized regions can exhibit much higher levels of extinction, especially in the dense H II regions in circumnuclear starbursts. In such cases, it is possible to use the near-infrared Paschen or Brackett lines instead. These lines have the drawback of being one to two orders of magnitude weaker than Hα, which effectively limits their use to high surface brightness nuclear H II regions. Advances in techniques and equipment are gradually making it possible to use these recombination lines to map entire galaxies, however [13].

Nebular emission derived SFRs are especially sensitive to the form of the IMF used, due to the fact that the ionizing flux that produces recombination lines is produced almost exclusively by stars with masses larger than 10 $M_\odot$ [13]. As an example, the Scalo IMF gives rates of star formation that are three times higher than those derived using the Salpeter IMF. Fortunately, however, Hα equivalent widths and broadband colors are both very sensitive to the slope of the initial mass function over the rage 1–30 $M_\odot$, which can be used to constrain the slope of the IMF. Using this information, normal galactic disks are best fitted by using a Salpeter IMF, or a modified Scalo IMF with Salpeter slope above 1 $M_\odot$ [13]. When using published SFRs, it is important to take into account the initial mass function that was used.

Because Hα is out of visible range beyond $z \sim 0.5$, there is an interest in calibrating bluer lines. Hβ and the higher order Balmer recombination lines are in the right range, but are too weak to be useful tracers, in addition to being rarely seen in galaxies earlier than Sc [13]. The strongest emission feature in blue is the [OII]λ3727 forbidden line doublet. Forbidden line luminosity is not directly coupled to ionizing luminosity, and is sensitive to abundance and the ionization state of the gas. Fortunately, the excitation of [OII] is well behaved enough to calibrate empirically using Hα. Because [OII] is in the visible range for redshifts up to $z \sim 1.6$, it is very useful for lookback studies, even with the level of indirection involved [13].

SFRs derived from [OII] calibrations are less precise than those derived from Hα due to this indirection, and they may be prone to systematic errors from extinction and variations the diffuse gas fraction. For example, the excitation rates of [OII] is particularly high in diffuse ionized gas found in starburst galaxies and the $L_{[OII]}$/SFR ratio in these cases is more than double that in normal galaxies. Metallicity, by contrast, has a fairly small effect on [OII] over the range of interest [13].

Overall, forbidden lines are useful as an estimate of the systematic patterns in star formation across distant galaxies, or as a check of consistency for SFRs derived using other methods. When more precise data are needed, the level of indirection and uncertainty involved in using forbidden lines to derive SFRs make it an unsuitable method.

1.2.2 What We Know About Star Formation

Global rates of star formation range greatly: elliptical, S0, and dwarf galaxies often exhibit no star formation activity while gas rich spirals average ~ 20 M$_\odot$/year. Optical starbursts can range up to ~ 100 M$_\odot$/year and IR luminous starbursts can have rates as high as 1000 M$_\odot$/year. The highest SFRs result almost exclusively from strong tidal interactions and mergers [13]. This large range is in part due to a large range in galaxy mass. Normalizing the SFR by mass or luminosity yields a range that spans two orders of magnitude, and exhibits a strong dependence on Hubble type, increasing by a factor of 20 between Sa and Sc type galaxies [13]. Within a spiral galaxy, star formation takes place in two different environments: the extended disk and the central bulge. Both of these are significant contributors to total star formation, but they exhibit different trends across the Hubble sequence and must be discussed separately.

Rates of star formation in the disk of a galaxy increases with the Hubble sequence, due to increases in both the number of star forming regions per unit mass or area and in the masses of individual regions. Because of this, the typical OB star in an Sa galaxy is in a cluster containing only a few large stars while the typical OB star in a large Sc galaxy is in a giant association with hundreds or thousands of OB stars [13]. This difference may have an effect on the structure and dynamics of the interstellar medium along the Hubble sequence.

There is a dispersion of a factor of ten in disk star formation rates among galaxies with the same type. This is larger than expected from extinction or other errors and likely reflects real variation. This may be due to underlying variation in gas content or nuclear emission, interactions that a particular galaxy is undergoing, or possibly short term variations in SFR within an individual galaxy.

It is also useful to parametrize SFR per unit disk area in order to avoid bulge contamination of total luminosities. This parametrization exhibits the same general trends, but the magnitude of change is weaker than in the parametrization by unit luminosity case, and the dispersion among the same Hubble type is large [13]. In general, SFR per mass is a good indicator of the evolutionary properties of disks while SFR per unit area is a good indicator of the dependance of SFR on gas density.

SFRs derived from FIR continuum measurements deviate somewhat from the picture given by ultraviolet and visually derived SFRs. These exhibit weaker trends, due to the fact that some early type spirals have strong FIR emission produced by luminous, dusty regions. If these regions are being heated primarily by young stars this reveals an important bias in visual and UV based studies. Conversely, it may be that much of this heating is a result of other populations, in which case FIR emissions are misleading as a tracer of SFR [13].

Important elements which have an effect on rates of star formation in the disks of galaxies include gas content, bar structure, spiral arm structure, and interactions and extragalactic environment. Gas content is strongly correlated with star formation, independent of Hubble type. Bar structure disrupts gas flow in the disk and triggers star formation in the nuclear region, while a strong grand-spiral structure localizes star formation in the arms of a galaxy. Neither of these effects changes total SFR [13].

The central bulge of many spiral galaxies harbor bright emission line nuclei, the most extreme of which are labeled starbursts. This characteristic constitutes a large portion of what sets this region off from the disc, in terms of star formation. Visual and UV derived SFRs for this region follow trends similar to those of the disk region. However, mid- and far-infrared observations reveal extremely luminous regions in many galaxies, which correspond to SFRs equivalent to the entire star formation of normal galaxies at the lower end of the range. This range extends upward to approximately 20 times higher than the highest rates of star formation found in normal galaxies [13].

These regions are strongly associated with the presence of unusually large amounts of molecular gas. This partly accounts for the high SFRs, but the star formation per unit gas mass is also much higher than normal, with efficiencies up to a factor of 20 higher than those found in normal galaxies [13]. These efficiencies could be even larger due to possible systematic error in the calculation of H_2 masses from CO masses.

In characterizing the evolutionary properties of galaxies, a useful measure is the ratio of current SFR to past SFR, averaged over the age of the disk, denoted b. Finding this value involves the use of evolutionary synthesis models to provide relationships between b and the broadband colors or Hα equivalent widths. The typical late type spiral forms stars constantly, giving $b \sim 1$, while early type spirals exhibit declining SFRs ($b \sim 0.01$–0.1) and elliptical and S0 galaxies have ceased star formation ($b \sim 0$). These values are based on integrated measurements however, which are subject to bulge contamination. Mean values of b, for the disk alone, are < 0.07 for Sa, 0.3 for Sb, and 1.0 for Sc [13]. The remaining trend is much stronger than the change in bulge mass fraction over these types, which implies that the variation is an inherent quality of the history of the disks, although the bulge structure may still have an influence [13].

There is, however, a large dispersion of b among galaxies of the same type, implying some real, long term variations in star formation history. In particular, there is a similar range of dispersion of gas contents, the variations in which correlate roughly with variations in both SFR and b. Some of the dispersion may also be explained by short term variations in SFR. Starbursts, in particular, are thought to be very important in the histories of many galaxies, especially low mass ones [13].

Circumnuclear regions, in contrast, show little dependence on either gas contents or the bulge to disk ratio, but instead are very correlated with dynamical influences such as gas transportation or gravitational perturbations (which induce gas flow to the center). Most optically selected spiral nuclei exhibit some Balmer emission, with an average Hα equivalent width comparable to the average value in late type spiral disks [13]. This is in the range expected for constant star formation, and the corresponding rates of star formation are consistent with steady state or declining histories. At least some of these nuclei, however, are likely to be in the interim phase between major starbursts. In infrared selected samples of galaxies, however, starbursts are the dominant mode of star formation. Typical gas consumption times are in the 10^8–10^9 years range, and as such the corresponding SFRs can only be sustained for a small percentage of Hubble time [13].

There remain many unknowns in this picture. Most of these results are based on crude, integrated averages which do not take into account intragalactic variations, as well as extrapolations to past histories from current SFRs. Nearly all models used make integral use of an initial mass function, a parameter which remains poorly understood. Moving forward, this will be improved through a two-pronged approach. On the one hand, more detailed and extensive lookback studies will result in better understanding of the histories of star formation in different types of galaxies, and provide a better means to test theories of galaxy formation and evolution. On the other hand, more detailed and extensive studies of nearby galaxies provide will provide a means to better resolve intragalactic trends in star formation, as well as the feedback mechanisms between star formation and other properties within the parent galaxy such as gas content and metallicity. A better understanding of these relationships will enable the creation of more accurate models of galaxy evolution along the Hubble sequence.

1.3 This Project

It is the latter of the two approaches which this study takes. The basic data used here consist of spectroscopic imaging of a small number of nearby, early type spiral galaxies. I examined spectra for a large number of H II (star forming) regions from within two different galaxies: NGC 3169 and NGC4725. Some key points to be addressed in this study include:

- Early type spirals are underrepresented in the existing literature. Focusing on early type spirals in this study helps to address this deficit.

- The number of individual spectra from each galaxy is sufficient to perform an analysis of chemical variations on scale of less than one kiloparsec.

From the basic spectra, the strengths of important emission lines were measured: [O II] $\lambda 3727$, [O III] $\lambda\lambda 4959, 5007$, [N II] $\lambda\lambda 6548, 6583$, H$\alpha$ $\lambda 6563$, and Hβ $\lambda 4861$. Using these lines, the R_{23} parameter can be constructed, an important measure of chemical composition, which can be used to determine the O/H ratio. The [O II]/[O III] can be used to disambiguate the O/H ratio in cases where R_{23} is double valued with respect to this ratio [18]. The multiple HI lines are used to determine the dust content and extinction levels.

2 Methodology

The process of data reduction for this project has two primary goals. The first goal is to minimize the effects of both noise and systematic bias on the data. Noise is random variations encountered while the data were being taken. Because of its random nature, noise is dealt with mostly through taking several observations and averaging them, which increases the signal to noise ratio. Bias is systematic variations introduced in predictable ways by either instrumental or sky based sources, and is dealt with by understanding the nature of the variation and taking steps to counteract that. The second goal of the data reduction process is the extraction of the data I am actually interested in from the raw spectra, and the processing of this data into useful metrics of metallicity.

The data for this project was gathered by Charles Hoopes, David Thilker, and Salman Hameed using the 4-Meter R.C/Cryocam Spectrograph CCD at Kitt Peak National Observatory. The observations were taken over eight nights, April 18, 2004 through April 26, 2004. For every night of observation, the data set contains many zero images. These are exposures of zero time with no light hitting the detector. This represents the zero point of the detector. In the ideal situation, this would return a uniform value across the entire array, which could then be subtracted from every pixel in subsequent images. In the real world, the value returned may vary across the array, with individual pixels having different values, so it is necessary to read out the whole detector array and subtract the resulting image from each observation. The purpose of taking many such zero images is twofold. First, it is necessary to verify that any patterns detected are stationary over the relevant time periods. Secondly, the array possesses a certain level of readout noise. This is a level of random variation that is inherent in the process of reading out the values from the detector. Zero images are particularly affected by readout noise, because the variation of read out noise is large compared to the values present in a zero image.

This accounts for any static variations present, but does not take into account short term variations, which may vary from exposure to exposure. In order to deal with this, each of the images in this data set includes an overscan region, an area 32 pixels wide which does not correspond to any real part of the array. Rather, the detector is instructed to continue reading out values past the end of the physical portion, in order to gauge the count value inherent in reading a value from the array, independent of any physical part of it. This can vary from one image to another, so the average value of the overscan for a given image must be subtracted from the values for each pixel in that image before two different images can be compared [9]. After this, the image is cropped such that the overscan region is discarded.

In the ideal case, reading out a zero image would be instantaneous. It is intended to represent the zero integration time electronics background of the CCD [9]. Exposures of longer lengths are affected by variations in the efficiency of individual pixels, which is accounted for in a different process. In the real world, however, this process takes a small amount of time to complete. Because of this, there is a small chance that the array was hit during the time it took to read out the values by a cosmic ray, high energy particles with non-terrestrial origins. These result in very high values being read by the individual pixel that was struck. When combining the zero images, pixels that appear to have been contaminated in this manner can be rejected using an averaged sigma clipping algorithm. This algorithm works by comparing the standard deviation for a given pixel across several images with the square root of the average (mean or median) of the values, times a proportionality constant, as described by the equation: $\sigma = \sqrt{k \cdot \overline{x}}$. The average values and proportionality constant are both calculated, and then pixels found to have a standard deviation larger than the calculated value for σ times a multiple are rejected. The average value is then calculated again, and the process repeated until either no more rejections happen or the pool of values drops below the minimum needed to run the algorithm. [20]

The `combine` task is a very general and powerful program for the combination of two dimensional CCD images, so I have used a simplified version called `zerocombine`, which eliminates some of the options that do not apply to the case of zero images, and changes the presets on some other options to be more appropriate. Because the zero image represents a value that the detector reads on top of any exposure that it might have, this value is later subtracted from each of the images of observed objects using the IRAF task `ccdproc`.

Also included in the data set are three flat field images for every object observed, taken at close to the time that the object was imaged. These are images of a uniform light field, taken in order to expose any variations in the efficiency with which different parts of the CCD array converts photons to counts. Because the efficiency of the CCD can depend on the wavelength of light being observed, choosing a field of illumination with similar light distribution to the object observation is necessary. This is balancd against logistical concerns: the most accurate form of flat field, in this regard, is obtained by taking an exposure of the night sky in a region devoid of stars but this method requires a lengthy exposure time and cuts into time that could be spent making astronomical observations. I have dome flats, meaning that the inside of the dome of the observatory was illuminated with a bright lamp, and an exposure made of an area of this. Multiple images were taken in order to allow for averaging with cosmic ray removal as in the case of zero images. Fewer images are needed due to the lower importance of read noise relative to the brightness of the flat itself, but a minimum of three images is required in order to perform cosmic ray removal. When using ccdproc, this single flat-field image is scaled so that its average value is unity. The object image is then divided by this, producing the image as it would be if the CCD array had a uniform response efficiency.

In order to perform spectroscopy, light from astronomical sources must be dispersed out into a spectrum. For this study, this was performed by viewing the sources through narrow slits in a metal disk known as an entrance mask. A given mask has approximately 20 to 30 slits which correspond to the positions of objects of interest and which will line up with these when placed over the entrance to the telescope. The layout of each mask was determined by a program called MSLIT which accepts a list of target coordinates, weighted by importance, and attempts to maximize the cumulative weight of the mask, given the constraints necessitated by the need to prevent spectra on the final image from overlapping and contaminating each other. In addition to the objects being observed, a mask will also include one or more slits positioned so that no astronomical object lies behind it. This allows for the observation of the baseline spectrum of light contributed by atmosphere, called a sky spectrum, which is present in all observations. Also present on the mask are narrow slits cut for purposes of ensuring that the telescope is properly aligned. These are used during the process of taking the observations, and are not important afterwards; they are not object spectra but neither is there a guarantee that they are pure sky spectra [6].

For each observation taken with a particular mask, six images were taken. Three of these are observations of the actual object and three of them are observations of a helium-neon-argon (HeNeAr) comparison lamp taken with identical setup. This comparison lamp produces a known spectra which can be used to convert the pixel numbering of the raw images into wavelength numbering. Two aspects need to be taken into consideration here. First is the simple pixel to wavelength conversion, which varies across the dispersion axis of the image. Additionally, the slits on the mask have different locations in the dispersion axis, which results in the resulting spectra becoming offset relative to each other by up to 15% [6]; this offset is matched as well. The details of this process will be discussed slightly later in this section.

All six images are processed with the IRAF task `ccdproc` in order to correct for zero level bias, overscan bias, and flat field the image as described already. After this, the three object images are combined into a single object image and the three comparison images into a single comparison image, using the `combine` IRAF task. As before, this served to improve the signal to noise ratio and to remove any cosmic rays that might be present. This resulting image formed the base from which I could begin to extract spectra. An example of what the data looks like at this point of the reduction process is seen in figure 1.

The data in this image are in the form of roughly horizontal strips running across the length of the image. Each of these strips represents the spectrum of either an object of interest or of the night sky. As a result of the fact that the slits on the mask were not perfectly square with respect to the dimensions of the detector, these strips are at some tilt relative to the axes of the image. Additionally, at the higher wavelengths the image appears to lose focus and a broadening of the individual strips is observed. This indicates also that the angle of individuals strips may vary, and it is necessary to calculate the angle of each strip individually.

To deal with these effects, I used the output file from MSLIT that was used to cut the masks. This file contains the physical location and extent of every slit on the mask. I assumed that for a given column of the image, there was a constant conversion factor between the location of the strips on the image in pixels and the physical location of the slits on the mask, given in millimeters. In order to calculate this conversion factor, I first determined the total vertical extent of all strips, in pixel coordinates, at two different locations on the image. Using the ratio of pixel extent to physical extent, I converted all of the physical positions to pixel positions, valid for that column. The actual spectra on the image were slightly broader than this, due to light bleeding over from the edges of the slits on the mask, so the size of all slices was increased by a small amount to account for this effect.

From the location of a given strip at two different columns, I was able to calculate the angle which that particular strip made with regard to the axes of the image. For every spectrum, I rotated the image by this angle so that that strip was now horizontal. I then cropped this rotated image around the strip. For the coordinates of this, I used the averages

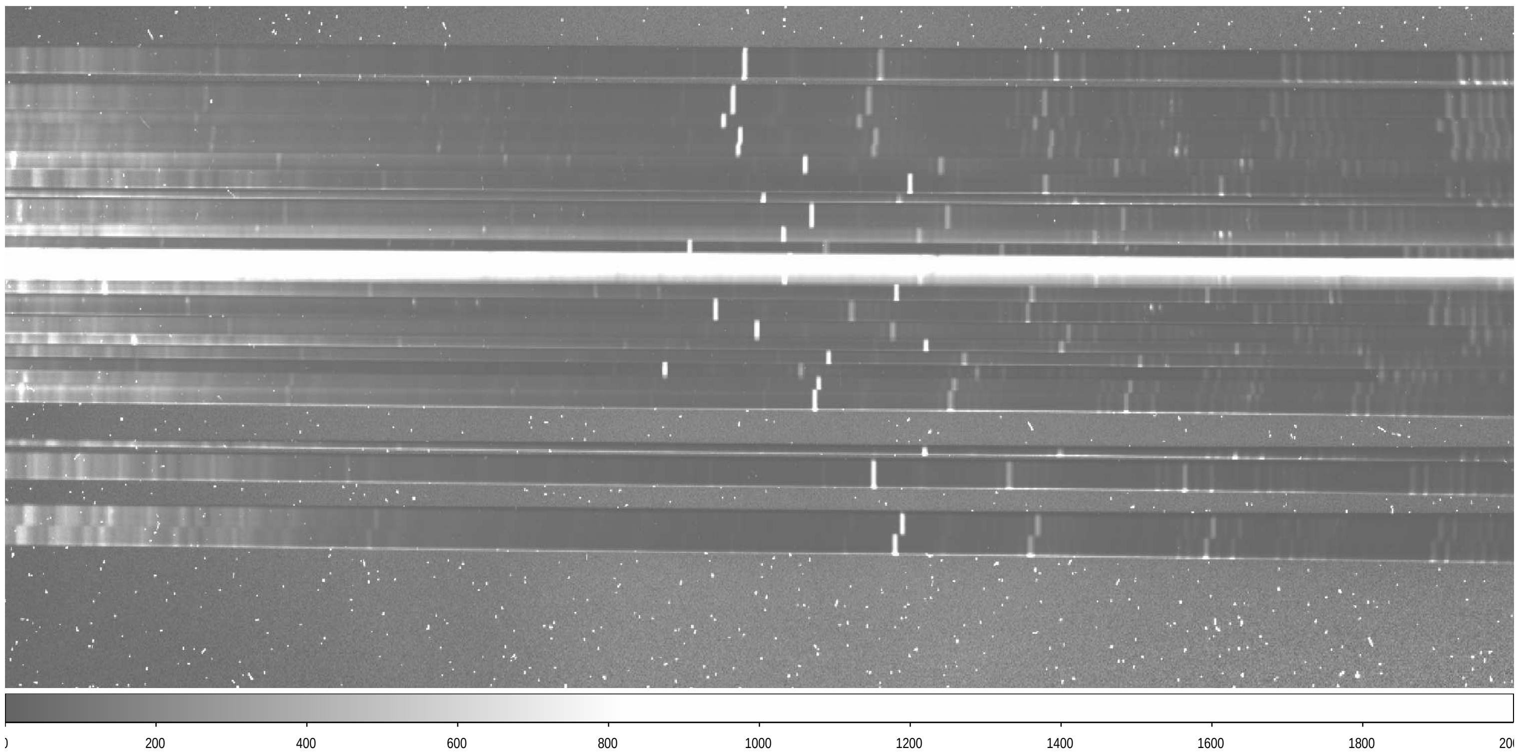

Figure 1: This figure shows an example of what the CCD image looks like before full data reduction has been performed. This image has already been corrected for zero bias, flat-fielded, and combined, but it has not yet been been split out into individual spectra.

of the locations and made no effort to save areas where the strip had broadened beyond this size in the unfocused region. I elected to do this and not a more complicated transformation because the broadening present in that region made it unsuitable for the extraction of reliable data, and there were no emission lines in that region that I was interested in measuring anyway. After this I performed a visual inspection of every strip, to determine whether it contained the full amount of light from its spectra. When I found some that were cropped too closely, I modified the values for the total vertical extent of all strips which served as the input for this process, and I regenerated the cropped strips for these new values. I iterated this process until I found values for which no spectra had missing light. This process was a trade off between the desire to know that all of the light measured in a given spectrum is actually from the object being observed and desire that all of the light emitted by an object is accounted for in the resulting spectrum. By choosing to optimize for completion, some spectra include light from neighboring spectra, but I do not suffer from decreased line strengths.

At this stage, for every spectrum that was on the image I have two rotated and cropped strips, the object and the comparison. From these two dimensional strips, I produced one dimensional spectra using the IRAF task `apsum`. This produced the simple sum of each spatial column. Then I used the IRAF task `identify`. This program allowed me to label individual peaks on the spectrum produced by a HeNeAr comparison lamp with the wavelength that they are known to be located at. Using the spectral atlas maintained by NOAO [22], I identified a large number of the emission lines present in one of my comparison spectra. The `identify` task then matched the remaining lines against a line list included as a library in IRAF, and calculated a function to convert pixel coordinates to wavelength ones along the axis of dispersion. This function is known as a dispersion solution. Some of the line identifications would be inaccurate and not fit this function very well. I was then able to interactively remove those with high residuals, as well as alter the type of equation being fit to the data. In this process, I tried to keep the number of line identifications in the sample high, in order to produce an accurate dispersion solution. This was balanced against the need keep the root mean squared of the sample low, in order to produce a precise dispersion solution. Additionally, the order of the fitting function could not be too high, or else it could become non-monotonic and thus unusable. I settled on a solution which had around 45-50 line identifications, a root means squared of less than 0.25 Å, and a cubic spline fitting function with an order of 3. When I was satisfied with this, I saved it to the database that IRAF maintains.

Using this existing solution, I used the IRAF task `reidentify` to attempt to automatically find a dispersion solution for all remaining spectra. `reidentify` is able to automatically find any coordinate shift present, in addition to finding and identifying the spectral lines. If the found solution had a high root means squared value or if it was unable to match an appropriate number of points, I altered the solution until it was acceptable. I then applied the dispersion solution I had found for every comparison lamp spectra to their corresponding object spectra, resulting in dispersion corrected and wavelength calibrated object spectra.

These spectra can be grouped into three classes of spectra, as discussed earlier: night sky spectra, object spectra, and spectra which are the results of slits cut for the purpose of verifying the alignment of the telescope. This third class of spectra were now disregarded, as they were no longer useful for anything. The IRAF task `sarith` is a simple program for performing arithmetic of one dimensional spectra. Using this, I divided each night sky spectra by the width of the strip that had produced it. I did this based on the assumption that the light from the sky had been uniformly distributed along the length each slit, so that the brightness of each spectra was uniformly distributed. Once these spectra had a common scaling, I combined them using the IRAF task `scombine`, a process similar to the two dimensional combinations discussed earlier.

Now I wanted to subtract this combined sky spectrum from each object spectra, in order to reveal the object spectra without any light from the atmosphere. The `sarith` task makes this subtraction easy, but I needed to determine how much the sky spectrum should be scaled up to match the object spectrum. In order to do this, I devised a metric by which the quality of a given scaling could be measured. The primary measure I was concerned with was the removal of emission lines present in the atmosphere. In the ideal case, these emission lines would match perfectly between the sky spectra and the object spectra, and after subtraction there would be a flat continuum where there had been an emission line previously. In the real world, variations in the profile of an emission line are present, as well as slight variation in the dispersion solution which can result in emission lines being slightly misaligned. After subtraction, some portions of the emission line remain above the continuum, while some portions are below the continuum level. A good subtraction should balance these. In order to quantify this, I took several important atmospheric emission lines from Massey (1990)[16]: Na I $\lambda5890$, O I $\lambda5577$, O I $\lambda6300$, and O I $\lambda6364$. For a given sky subtraction, I calculated the standard deviation for the area around each of those lines and then the average of these values. Given this metric, I used SciPy's fmin optimization to find a minimum. See the attached source code for the full details of this process. I then inspected the results of this, and made adjustments where I judged that the optimization had produced poor results. This usually took the form of oversubtraction of the sky spectra. In this case the sky lines would be visible, now completely below the continuum level of the spectrum and I would need to decrease the subtraction factor until these lines were balanced between above and below the continuum level.

My next step was to flux calibrate the sky subtracted spectra. This process converts the values of the spectra from counts into fluxes. The sensitivity of the detector can vary with respect to wavelength, so it is necessary to fit the response profile of the detector across all relevant wavelengths. For each mask, an image of a reference star was taken through that mask. For NGC 3169 this was the reference star Feige 34 and for NGC 4725 this was the reference star PG 1708+602. Both of these are hot white dwarfs, characterized by a smooth emission profile with few emission lines. [17] Their emission profiles are well studied, and detailed information about both are included in the standard IRAF distribution. I extracted the spectra for these reference stars following identical procedure to that used to extract the object spectra already described. Once I had the sky subtracted, flux calibrated spectrum for a given reference star, I used the IRAF task `standard`. This task integrates over various bandpasses in the spectrum of the reference star, which have known values. [21] This is written out to a calibration file which serves as the input for the `sensfunc` IRAF task. This task allows for the interactive fitting of a sensitivity function to these calibration data. Once I had it, I applied this sensitivity function to the relevant object spectra. A fully reduced and calibrated spectrum is shown in figure 2.

With all of my spectra flux calibrated, I was then able to begin measuring line strengths. I did this using the deblend capability of the IRAF task `splot`, a program for plotting and working with one dimensional spectra. This allowed me to specify continuum points on either side of the line I am interested in measuring, and to mark any number of lines within that region to be de-blended. Each of these lines is then fitted with profile of a certain type; I used gaussian profiles. The relevant documentation [19] goes into detail as to how this fitting is achieved. Because there is a certain amount of variation in this process, mostly related to the selection of continuum points, I measured every line multiple times and averaged together the values that I found.

The lines that I measured are [O II] λ3727, [O III] $\lambda\lambda$4959, 5007, [N II] $\lambda\lambda$6548, 6583, Hα, and Hβ. The hydrogen lines allow for extinction correction and the oxygen and Hβ lines allow for the construction of the R_{23} parameter to determine the O/H ratio [11]. R_{23} is defined as

$$R_{23} = \frac{F([OII]\lambda 3727) + F([OIII]\lambda 4959) + F([OIII]\lambda 5007)}{F(H\beta 4861)} \tag{2}$$

R_{23} can be calibrated to 12 + log(O/H), the primary measure of oxygen abundance, using the following equation [18], ($x \equiv 12 + \log \mathrm{O/H}$):

$$\log(R_{23}) = 1.2299 - 4.1926x + 1.0246x^2 - (6.3169 \cdot 10^{-2})x^3 \tag{3}$$

As discussed in the introduction, R_{23} has the problem of having two solutions for any given value; a low metallicity solution and a high metallicity solution. In order to disambiguate these branches, I calculated the $F([OIII]\lambda 5007)/F([OII]\lambda 3727)$ ratio. If this ratio is less than 2 then region is on the high metallicity branch [18]. Figure 3, adapted from Nagao et al. (2006) [18], shows this equation plotted over top of the data they used to derive the equation. The double valued nature of the conversion is readily apparent in this figure.

I used the hydrogen series lines to calculate extinction. I calculated the E(B-V) metric from the Hα/Hβ ratio as calibrated by Reynolds, et al (1997) [25].

$$E(B-V) = 2.21 \log(\frac{F(H\alpha)/F(H\beta)}{2.76}) \tag{4}$$

Once I have the E(B-V) value, I can use the method given by Calzetti (1997) [2] to correct all of my measured fluxes for extinction effects. This is done with the following equation

$$F_{obs}(\lambda) = F_0(\lambda) 10^{-0.4E(B-V)\cdot k(\lambda)} \tag{5}$$

where k(λ), the selective attenuation of the stellar continuum, is defined as

$$k(\lambda) = 1.73 - 0.1\lambda^{-1} + 1.86\lambda^{-2} - 0.48\lambda^{-3} \tag{6}$$

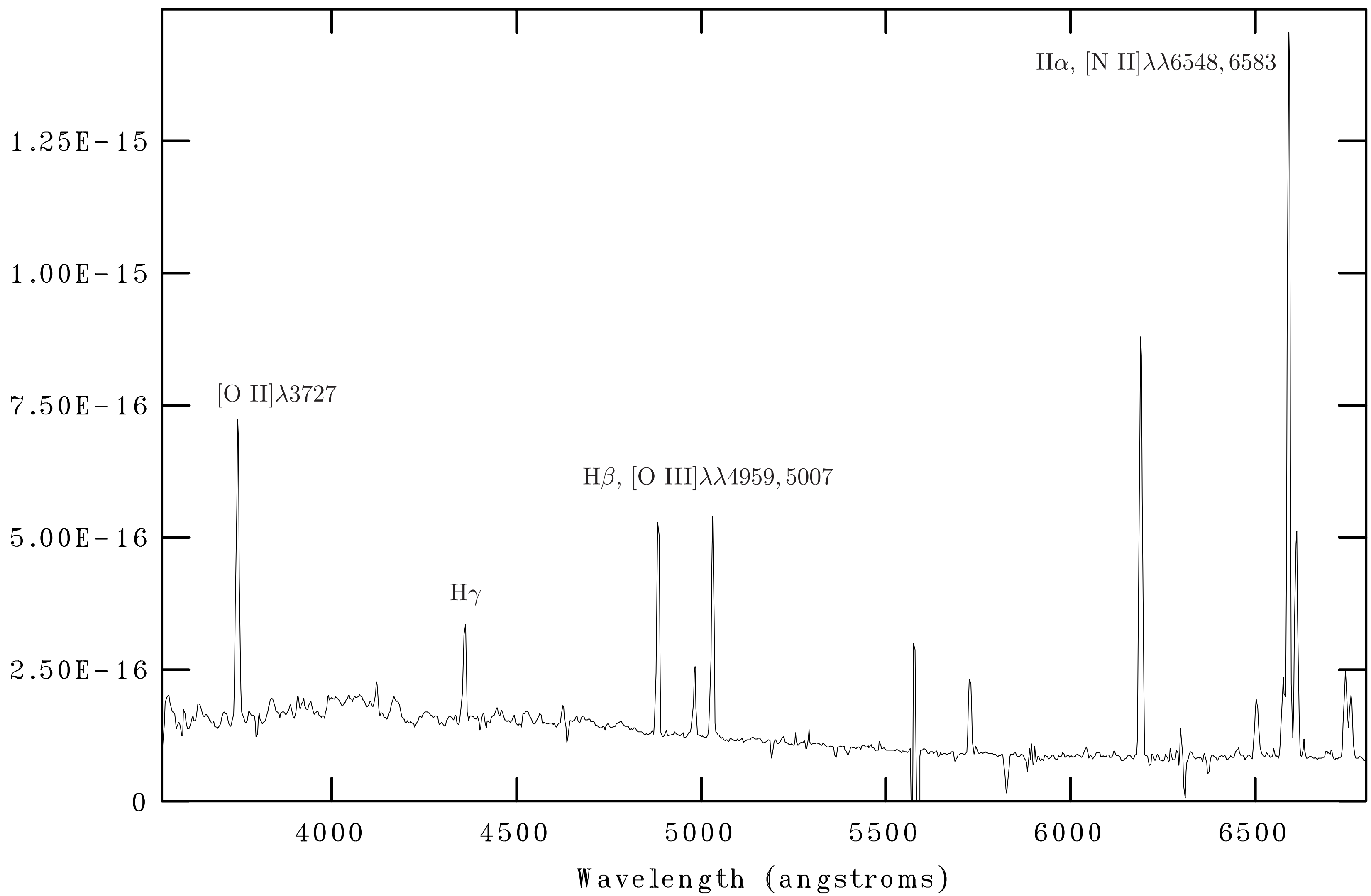

Figure 2: This figure shows a representative sample of a spectrum from an H II region following data reduction. Important emission lines are labeled. Values on the vertical axis are in units of ergs $\cdot$ cm^{-2} $\cdot$ s^{-1}

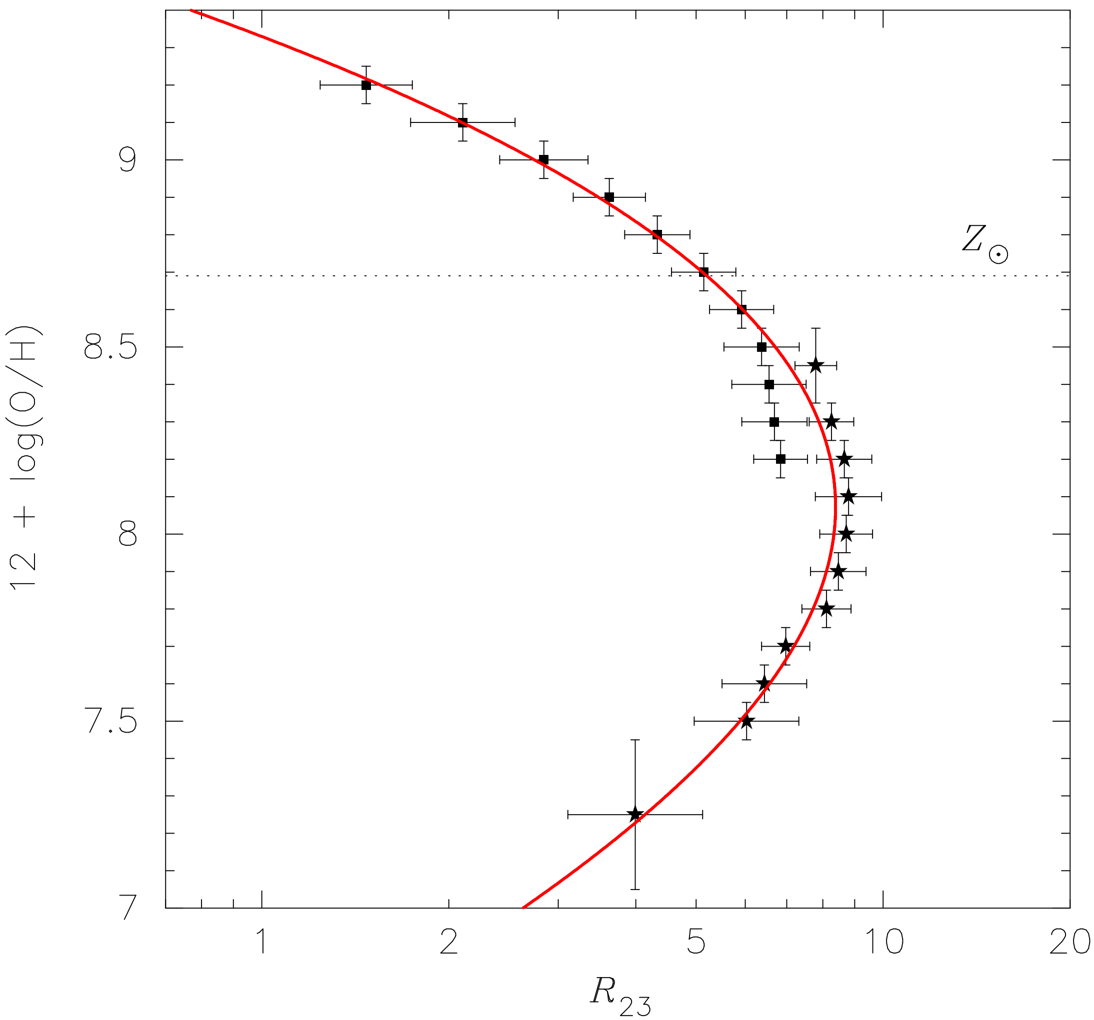

Figure 3: This figure, adapted from Nagao et al. (2006) [18], shows the R_{23} to $12 + \log O/H$ relation plotted over top of the data they used to derive the equation. Each point represents the binned average of galaxies for a given oxygen abundance. Two data sets are present. The double valued nature of the relation can be seen in the way a given R_{23} value corresponds to two possible values of oxygen abundance.

for 0.63 μm $\leq \lambda \leq 1.0\mu$m and

$$k(\lambda) = 4.88 + 2.656(-2.156 + 1.509\lambda^{-1} - 0.198\lambda^{-2} + 0.01\lambda^{-3}) \tag{7}$$

for 0.12 μm $\leq \lambda \leq 0.63\mu$m. Using these equations, I converted all of my measured line strengths (F_{obs}) to intrinsic emission strengths (F_0). Both are in units of ergs $\cdot$ cm^{-2} $\cdot$ s^{-1}.

I also used my Hα fluxes to calculate star formation rates, using the equation given by Kennicutt (1998) [13]:

$$\mathrm{SFR}(M_\odot\ \mathrm{year}^{-1}) = 7.9 \times 10^{-42} \times 4\pi \times F(\mathrm{H}\alpha) \times D^2 \tag{8}$$

Where D is the distance to the galaxy in centimeters and $F(\mathrm{H}\alpha)$ is the flux of Hα in ergs per second per square centimeter.

3 Results and Analysis

3.1 The Data

At the end of my data reduction process I have extinction corrected fluxes for 47 different H II regions, 18 regions from NGC 3169 and 29 regions from NGC 4725. For most of these regions I have fluxes for seven emission lines: [N II]λ6548, [N II]λ6583, [O II]λ3727, [O III]λ4959, [O III]λ5007, Hα, and Hβ. In a few cases I was unable to measure one these due to the weakness of the line. For the three nuclear regions I was unable to measure Hβ. This is likely due to hydrogen in these regions being nearly entirely ionized. For three faint regions in NGC 4725 I was unable to measure the strength of one of the oxygen lines. I have used these emission lines to calculate the rate of star formation and the gas phase relative oxygen abundance in each H II region.

For each of the two galaxies, I have analyzed three relations: oxygen abundance as a function of galactocentric radius, star formation rate (SFR) as a function of radius, and oxygen abundance as a function of star formation rate. For both the metallicity and the star formation rates, I have analyzed the data in terms of the radius normalized to the isophotal radius of the disk, R_{25}. To derive this value, an oval is fitted to the surface brightness of the galaxy so that there is a roughly constant magnitude density (isophotal) along the entirety of the oval. R_{25} is defined as the semi-major axis of the oval that fits the 25th magnitude per square arcsecond isophote in the B band. By normalizing all radii by this value, direct comparison between galaxies of differing sizes can take place. To further facilitate direct comparisons, I have plotted all radius and metallicity values over the same range. The differing ranges involved in star formation rates between NGC 3169 and NGC 4725 rendered identical plotting for that variable impractical however, and I have allowed that range to vary appropriately.

For oxygen abundance as a function of radius, a negative correlation is expected, with high metallicity regions being found towards the center of the galaxy and low metallicity regions being found in the outer regions of the galaxy. This is due to the nature of metallicity as the residue of previous generations of star formation and the density profile of spiral galaxies. Galaxies are more dense in their centers which results in higher concentrations of stars and corresponding higher amounts of metal rich material. This is, however, dependent on several other factors, chief among them patterns of gas transport within the galaxy. If a galaxy has a well-mixed interstellar medium, this will result in a flattened metallicity curve [8]. For a quantitative analysis of the metallicity, I fit a linear function to the plot of $12+\log \mathrm{O/H}$ versus isophotal radius fraction, and calculated the gradient of this function and a characteristic oxygen abundance. The gradient is in terms of dex/R_{25} and the characteristic oxygen abundance is the value of the fitted function at 0.4 R_{25}. The choice of performing these analyses in terms of R_{25} is made in order to minimize systematic biases related to disc scale or bulge/disk ratio [32]. This works because R_{25} scales along with these factors.

To get a better sense of how metallicity relates to star formation, I plotted rates of star formation as a function of radius and oxygen abundance as a function of star formation rate. Rates of star formation with radius are generally expected to exhibit similar trends to that of metallicity, but an analysis of metallicity as a function of star formation rate, or vice versa, is not readily apparent in the literature.

Additionally, I have taken information about the emission line strengths for 149 H II regions from twelve different galaxies from Zaritsky, Kennicutt, and Huchra (1994) [32] (ZKH). Using their data, I calculated the gas phase relative oxygen abundance for each of these regions to serve as a basis for comparison. I have discarded the data from two of their galaxies for this purpose. One galaxy consisting of two H II regions was confusingly labeled, owing to a probably typo in the text, but given the small number of regions it was easiest to simply discard it. Their data also included 9 H II regions from NGC 4725 and I have not used these data to avoid plotting NGC 4725 twice. For these data, separate values for [O III]λ4959 and [O III]λ5007 were not available, so I was unable to calculate whether a region was on the upper or lower branch of the $12 + \log \mathrm{O/H}$ — R_{23} relation using the [O III]λ5007 / [O II]λ3727 ratio. I have instead assumed that all H II regions are located on the upper branch, and calculated their oxygen abundance accordingly.

3.2 NGC 3169

3.2.1 Galaxy Information

NGC 3169 is an early type (type Sa), unbarred spiral galaxy and the distance to it is approximately 20.5 Mpc [31]. It has a low-ionization nuclear emission-line region (LINER) in its center [10], a form of active galactic nuclei characterized by strong emission lines from weakly ionized particles and weak emission lines from strongly ionized particles. LINERs are very common but poorly understood: up to one-third of all galaxies have nuclear spectra typical of LINERs but the source of energy driving them is unknown in general. Possible explanations include photoionization from a black hole (the conventional AGN energy source), fast shocks, photoionization from a population of hot stars, or photoionization from a population of old, metal-rich stars [15].

It is part of a group including NGC 3166 and NGC 3156, both of which are lenticular galaxies. In a study of these three galaxies, Sil'chenko and Afanasiev (2006) found that their centers are dominated by a fairly young stellar population, no older than 1-2 Gyr and concluded that this was the result of a synchronous star formation burst. Due to the fact that lenticular galaxies have very low stocks of gas in their interstellar medium, they concluded that gas inflow from the H I cloud that all three galaxies are embedded in must have occurred, and that the natural way to account for this effect would be interactions between each galaxy and the embedding H I cloud, rather than pairwise interactions. Additionally, they found evidence of non-circular gas motion in the central region of NGC 3169, possibly indicating the presence of a compact bar [27]. Gonzalez-Delgado et al (1997) studied the galaxy as part of a survey of nearby galaxies with active galactic nuclei and found that the nucleus features an extension approximately 1 kpc long almost perpendicular to the plane of the galaxy, cospatial with a radio emission jet. This alignment between radio continuum emission and the ionized gas of a LINER is an arrangement that has been found in other LINERs [10]. The jet in these situations is a possible source of the energy powering the LINER, either via

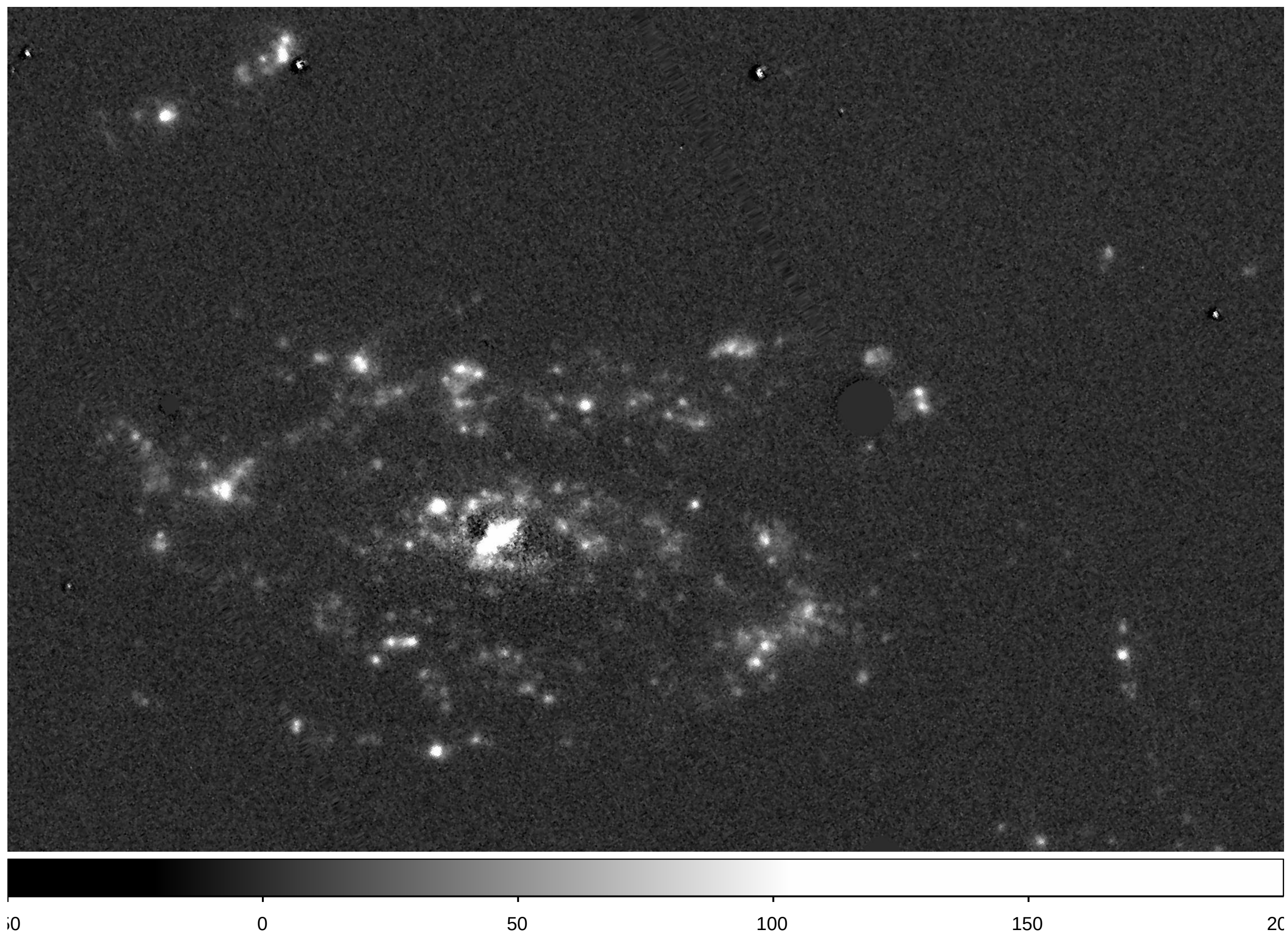

Figure 4: NGC 3169 imaged in Hα.

Number	Radial Distance (kpc)	$12 + \log \mathrm{O/H}$	SFR($M_\odot$ / year)
1	1.5×10^{1}	8.6	1.2×10^{-3}
2	6.6	8.5	2.6×10^{-2}
3	6.0	8.5	5.7×10^{-2}
4	5.2	8.4	2.4×10^{-2}
5	9.4	8.7	6.9×10^{-3}
6	3.8	8.3	3.0×10^{-3}
7	5.1	8.8	1.0×10^{-2}
8	6.8	8.7	1.6×10^{-2}
9[a]	7.4×10^{-1}	. . .	2.7×10^{-3}
10[a]	5.9×10^{-2}	. . .	6.2×10^{-4}
11	6.3	8.8	6.7×10^{-3}
12	1.5	8.9	1.7×10^{-1}
13	2.6	8.4	4.7×10^{-2}
14	4.9	8.7	2.3×10^{-2}
15	6.3	8.8	1.3×10^{-1}
16	6.9	8.6	2.9×10^{-2}
17	8.0	8.6	9.0×10^{-3}
18	8.5	8.3	4.7×10^{-2}

Table 1: This table gives the gas phase relative oxygen abundance (quoted as $12+\log \mathrm{O/H}$), star formation rates (in $M_\odot$ per year), and galactocentric radius for 18 H II regions in NGC 3169. [a]The star formation rate for H II region numbers 9 and 10 are not corrected for extinction because I could not obtain a measurement of Hβ for these regions.

shocks or a local ionizing continuum [12].

3.2.2 Data for NGC 3169

Figure 5 shows the gas phase relative oxygen abundance, quoted as $12 + \log \mathrm{O/H}$, for H II regions in NGC 3169 as a function of fractional isophotal radius. Solar oxygen abundance is shown as a dotted line and a linear fitted function as a solid line.

The expected trend for this is for metallicity to decrease as radius increases, as discussed earlier. It is difficult, however, to discern any particular trend from this graph. The majority of H II regions fall within approximately 0.8 R_{25}, and seem split into a line of sub-solar metallicity H II regions that increase in oxygen abundance with radius and a line of super-solar metallicty regions that decrease in oxygen abundance with radius. These trends seem unlikely to reflect any real processes, but are probably just the particular configuration of scatter for the set of H II regions available in the data I am using.

I calculate a gradient for this relation of 0.0030, which is nearly flat, in good agreement with what seems to be the case from visual inspection of the plot. This result is consistent with the findings of ZKH, who found that while the majority of galaxies exhibit negative exponential gradients in their metallicity, some galaxies are characterized by flat or even positive gradients in metallicity [32]. As discussed earlier, this indicates that NGC 3169 has fairly homogenous oxygen abundance throughout the galaxy, implying good level of gas

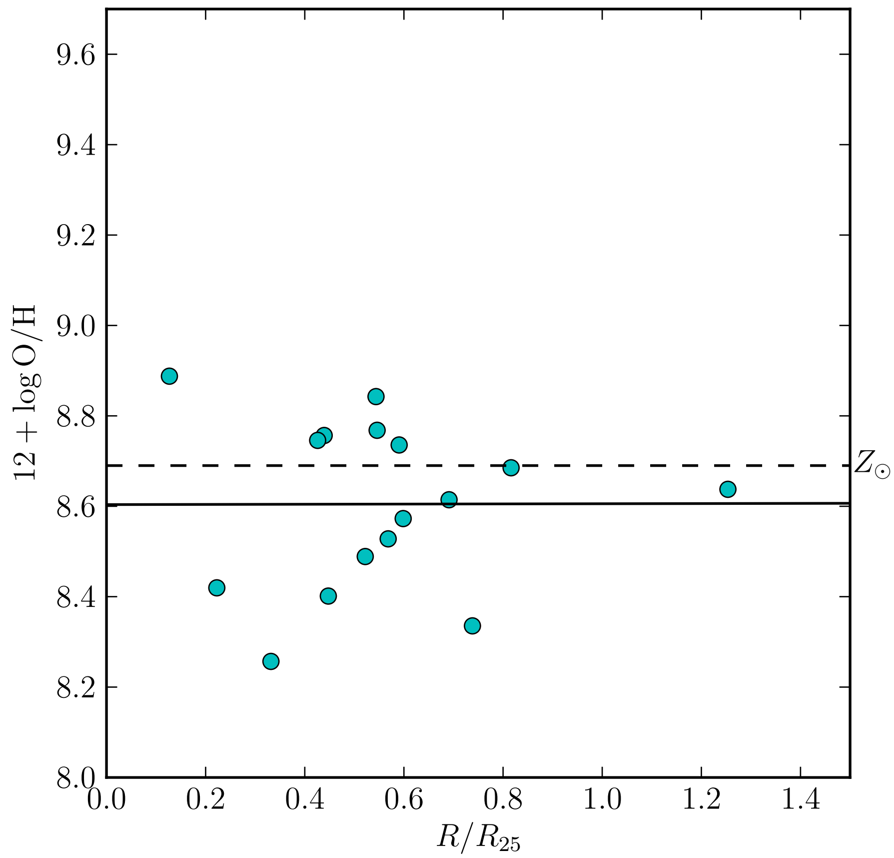

Figure 5: Gas phase relative oxygen abundance for H II regions in NGC 3169 as a function of galactocentric radius is shown here. Each plotted point of data represents a single H II region. Solar metallicity ($Z_\odot$) is marked with a dashed line and a fitted linear function is plotted as a solid line.

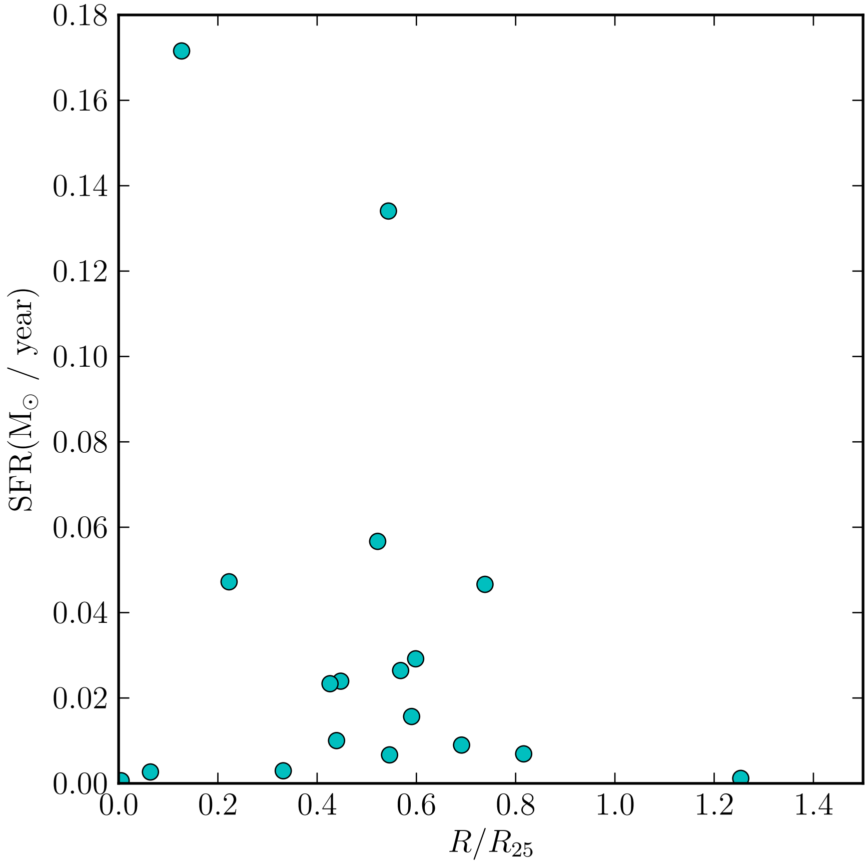

Figure 6: Rates of star formation in solar masses per year for H II regions in NGC 3169 as a function of galactocentric radius is shown here. Each plotted point of data represents a single H II region.

transportation and mixing. I calculate a characteristic oxygen abundance at 0.4 R_{25} as $12 + \log O/H = 8.604$.

Figure 6 shows the star formation rates in solar masses per year for H II regions in NGC 3169 as a function of fractional isophotal radius. Most of the H II regions seem to be clustered around low rates of star formation (< 0.4 $M_\odot$/year) at a radius between 0.3 R_{25} and 0.8 R_{25}. Some outliers are seen, ranging up to almost 0.18 $M_\odot$/year at a little under 0.2 R_{25} and as far out as > 1.2 R_{25} with close to no star formation, forming an outer envelope which seems to conform to the expectation that star formation rates will decrease with radius. However, the number of points within that envelope which do not show this pattern, combined with the limited number of data points at the edge of the envelope make this analysis uncertain. The galaxy is characterized by moderate levels of star formation, with a few regions of high rates of star formation, with a maximum at around 0.17 $M_\odot$/year.

Figure 7 shows the gas phase relative oxygen abundance, quoted as $12+\log\mathrm{O/H}$, for H II regions in NGC 3169 as a function of star formation rate in solar masses per year. Multiple interpretations are possible for this plot. The two points with the highest star formation rate seem to be very much outliers from the general cluster of H II regions, which have star formation rates less than half that of the most active H II regions. If these two high star formation regions are ignored, then the data seem to form a rough triangular envelope centered around $12+\log\mathrm{O/H}\approx 8.5$. In this interpretation, a broad range of metallicities possible for low rates of star formation is seen which narrows with increasing SFR. However, one of the data points at low SFR is significantly (≈ 0.2 dex) lower in oxygen abundance than the remainder of the low SFR H II regions. If this point is discarded, then a strong negative correlation between oxygen abundance and star formation rate is seen.

If the envelope interpretation is more representative of the actual processes involved, this seems to imply a situation where relatively inactive (in terms of star formation) H II regions can have any metallicity, but more active H II regions are somehow preferentially at a specific level of oxygen abundance. Because active star formation requires sufficient gas supply to function, it is possible that the level of oxygen abundance present in the more active H II regions is related to the oxygen abundance of the interstellar medium of the galaxy. To begin with, an H II region might be any metallicity. As gas flows into the H II region from the general interstellar medium, it changes the metallicity of the H II region to something more like the metallicity of interstellar medium. In this interpretation, we can infer that the interstellar medium has a gas phase relative oxygen abundance of somewhere around $12+\log\mathrm{O/H}<8.5$. This is reasonable given the fact that the interstellar medium is expected to be relatively unenriched [32] and I calculated the characteristic metallicity of H II regions as $12+\log\mathrm{O/H}=8.604$. As the star formation activity in a region slows down, it can either keep its newly formed metals and increase in metallicity or lose its metals back to the interstellar medium, creating the triangle of the envelope.

Alternatively, a triangular envelope could be formed by H II regions which start lower metallicities, gain metals as star formation begins, which both increase for a period, followed by a period of decreasing star formation activity while metallicity continues to increase. In this case the large number of H II regions higher than the metallicity of the peak star formation rate relative to the number of H II regions with lower metallicities implies that the galaxy had more star formation in the past and star formation activity is now slowing down overall.

In the interpretation of an inverse correlation between oxygen abundance and star formation rate, the trend can be interpreted physically as also drawing on the fact that gas inflow from the interstellar medium both causes star formation and lowers the metallicity of the region. In this case, the H II regions primarily keep their metals and return to a high metallicity as star formation activity slows down.

Neither of these models account for the two H II regions with both high oxygen abundance and a high rate of star formation. In interpreting these regions, it seems relevant that the trends I have just described take place over a fairly small range of star formation rates.

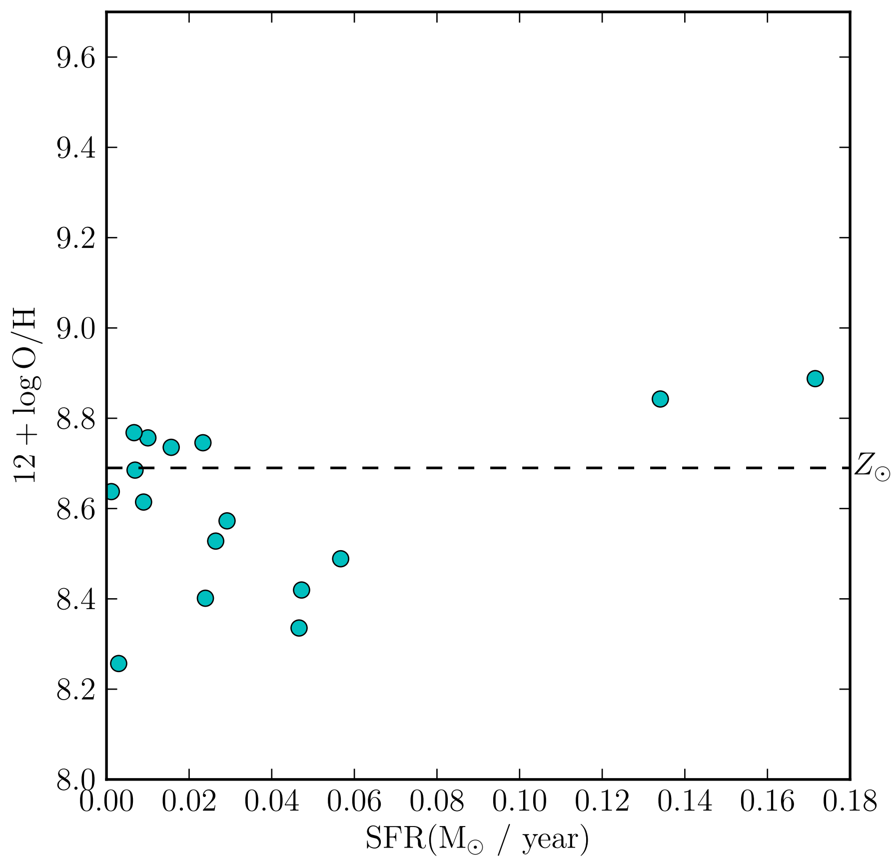

Figure 7: Gas phase relative oxygen abundance, quoted as $12 + \log \mathrm{O/H}$, as a function of rates of star formation in solar masses per year for H II regions in NGC 3169 is shown here. Each plotted point of data represents a single H II region. Solar oxygen abundance is plotted as a dashed line.

The high oxygen abundance, high star formation case can be interpreted as the result of a substantial inflow of gas from the interstellar medium, capable of causing sustained star formation and correspondingly significant levels of enrichment. A more complete survey of H II regions would be useful in determining the accuracy of these possibilities. Selection biases may be currently playing a large role in the appearance of these possible trends.

For two H II regions, I was unable to measure an Hβ line strength. Both of these regions are within 0.1 kpc of the galactic center, and exhibit very low Balmer series emission relative to other spectral lines. This is consistent with the classification of NGC 3169 as hosting a LINER in its central region by Gonzalez-Delgado et al. (1997) [10]. Storchi-Bergmann (1991) [28], found that Seyfert 2 and LINERs are characterized by an NII/Hα ratio of greater than 1.5, with a maximum value at approximately 4.75. The average value of this ratio for the two nuclear regions in my data for NGC 3169 is 16.9, while the average value for this ratio in non-nuclear regions is 0.66. This is quite a bit higher than the values found by Storchi-Bergman, and indicates that my data is likely in error about the magnitude of this effect in NGC 3169. This is, however, consistent with Storchi-Bergmann's finding that this effect is highly localized to the nuclear region of galaxies.

The average metallicity of NGC 3169 in $12 + \log \mathrm{O/H}$ is 8.60. It is characterized by moderate rates of star formation, with the most active regions having a star formation rate on the order of 10^{-1} $\mathrm{M}_{\odot}$/year.

3.3 NGC 4725

3.3.1 Galaxy Information

NGC 4725 is an early type (type Sab) galaxy with an intermediate bar structure and a ring. The distance to it is approximately 13.5 Mpc [26]. Wevers, Appleton and Davies (1984) found that NGC 4725 and NGC 4747 are tidally interacting, based on an examination of the kinematics and distribution of H I clouds [30]. Notably, they found that the velocity distribution of H I clouds in NGC 4725 is strongly asymmetric, due to two separate anomalies. The first is that the outer arm of the galaxy contains H I regions with very low velocities, coinciding with a pitch angle change, which they interpreted to represent the arm being bent out of the plane of the galaxy, away from the observer. The second is that the ring of the galaxy is poorly modeled by circular rotation, but is instead rotating and expanding at the same time, a result confirmed by Buta (1988). Buta examined the possible causes of this unusual behavior, but could not make any strong conclusions as to the origin of the ring or its expansion. She suggests several possibilities which all have problems associated with them. Uniform expansion could explain the large size of the ring, but the morphology of the ring suggests that the ring functions in orbital resonance with some co-rotating body. This is, however, difficult to reconcile with pure expansion. An explosive origin would require an implausibly high energy. An origin related to tidal interactions with NGC 4747 is currently difficult to argue either way, due to lack of enough information about the interaction history of NGC 4725 and NGC 4747. Buta concludes that this option should be further explored [1].

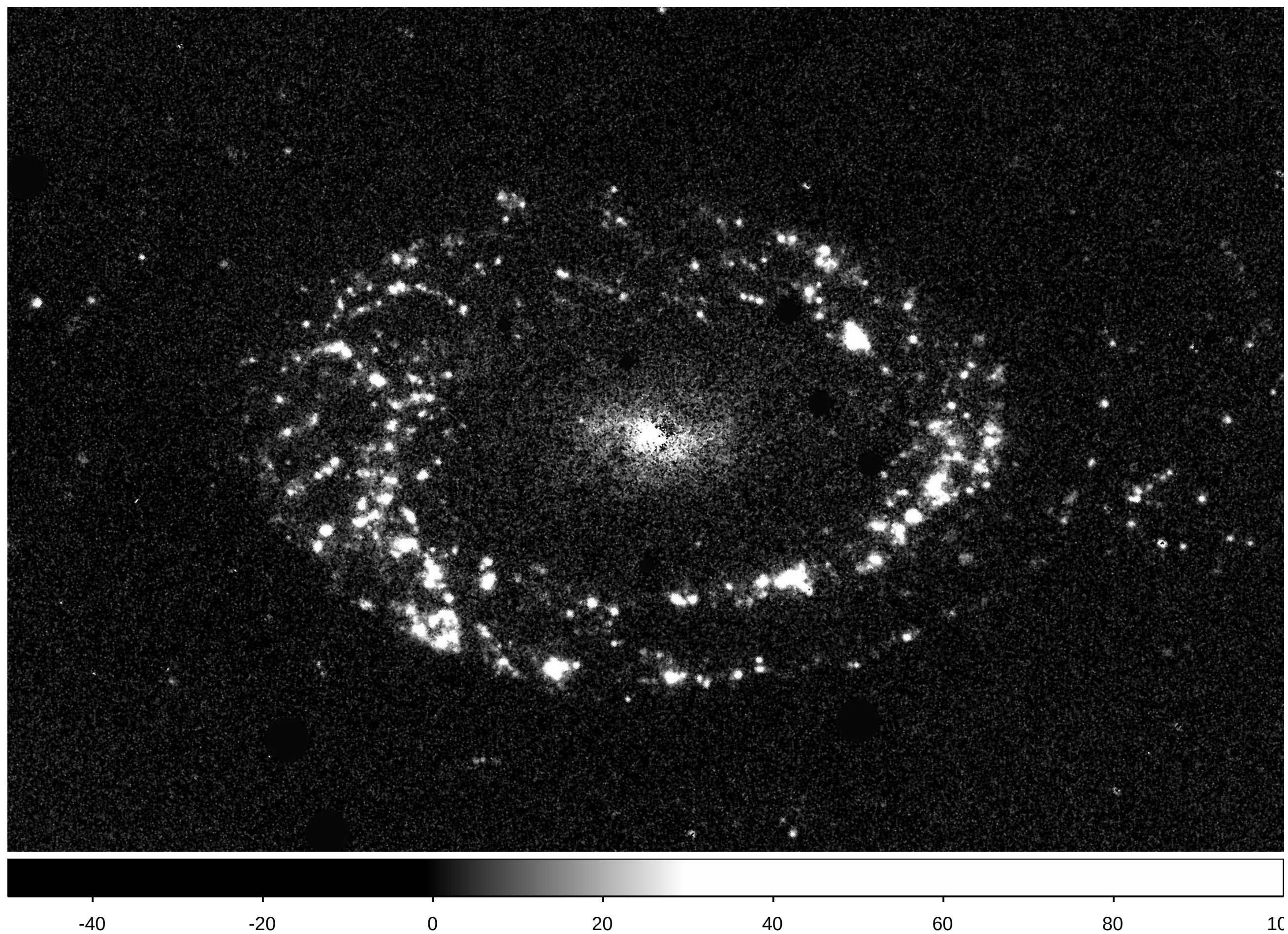

Figure 8: NGC 4725 imaged in Hα.

3.3.2 Data for NGC 4725

Figure 9 shows the gas phase relative oxygen abundance, quoted as $12 + \log \mathrm{O/H}$, for H II regions in NGC 3169 as a function of fractional isophotal radius. Solar oxygen abundance is shown as a dotted line and a linear fitted function as a solid line.

As already discussed, the expected relation for this is a negative correlation between oxygen abundance and radius. When plotted, this seems to be the case although the exact slope of the relation is unclear for two reasons. First, all available data for this galaxy fall within a fairly small range of radii, between 0.2 R_{25} and 0.5 R_{25}. This narrow range introduces a fair amount of uncertainty in the fitted line, as a great deal of extrapolation is needed for other radii. Secondly, even within this range there is a large number of regions at very similar radius around 0.2 R_{25}. If these were removed, the gradient could be considerably higher than it is here. That said, I calculate a gradient of -0.533 dex/R_{25} and a characteristic oxygen abundance of $12 + \log \mathrm{O/H} = 9.026$ at $R = 0.4$ R_{25}. These values are similar to those found by ZKH, who found a gradient of -0.43 ± 1.22 dex/R_{25} and a characteristic oxygen abundance of 9.14 ± 0.73 for NGC 4725 [32]. Both of the values that I found are within their listed error ranges.

Figure 10 shows the star formation rates in solar masses per year for H II regions in NGC 4725 as a function of fractional isophotal radius. Most of the H II regions exhibit very low rates of star formation (< 0.01 $M_\odot$/year). There is a single H II region with more than twice this amount of star formation at around 0.4 R_{25}, but this is still a quite low level of star formation activity. The galaxy is characterized overall by lower levels of star formation than NGC 3169, with a maximum at around 0.02 $M_\odot$/year.

Figure 11 shows the gas phase relative oxygen abundance, quoted as $12+\log \mathrm{O/H}$, for H II regions in NGC 4725 as a function of star formation rate in solar masses per year. Ignoring the one H II region with a star formation rate of ≈ 0.02 $M_\odot$/year, all of the H II regions in the galaxy form a fairly tight triangular envelope with a peak at approximately $12 + \log \mathrm{O/H}$ $= 9.2$ and SFR $= 0.01$ $M_\odot$/year. The discussion earlier of what an envelope interpretation would mean for figure 7, the same plot for NGC 3169, applies here. The strength of the envelope feature on this plot implies different things depending on interpretation. In the case of monotonically increasing metallicity over the lifetime of an H II region as a site of active star formation, this plot seems to imply that the H II regions in NGC 4725 are currently evenly distributed across this H II region life cycle.

In the other interpretation of the envelope of allowed values of metallicity with regard to star formation rates, the metallicity peak star formation rate is mostly determined by the metallicity of the interstellar medium. Away from this peak, the metallicity of an individual region can either increase or decrease depending on whether the gas enriched by star formation is mostly returned to the interstellar medium or mostly retained. The slope of the two edges supports this interpretation: the steeper slope along the bottom edge of the envelope is a result of the metallicity of the H II regions being increased by two factors: higher metallicity gas flowing in from the interstellar medium as well as gas enrichment through the star formation processes triggered by this gas. Along the upper edge of the envelope, gas inflow results in a decrease in metallicity, acting against the effects of gas enrichment through star

Number	Radial Distance (kpc)	$12 + \log \mathrm{O/H}$	SFR($M_\odot$ / year)
1	6.0	9.2	5.1×10^{-3}
2	5.6	9.2	9.1×10^{-3}
3	5.2	9.0	1.3×10^{-3}
4	5.0	9.2	3.3×10^{-3}
5	7.8	9.0	7.0×10^{-4}
6	5.2	9.1	8.2×10^{-3}
7	7.4	9.1	6.4×10^{-3}
8^a	3.3×10^{-1}	. . .	2.2×10^{-4}
9	5.5	9.2	1.5×10^{-3}
10	5.5	9.1	3.8×10^{-3}
11	7.5	9.0	2.1×10^{-3}
12	7.8	9.1	5.4×10^{-3}
13	5.6	9.1	4.1×10^{-3}
14	8.0	8.9	4.7×10^{-4}
15	6.8	9.1	3.9×10^{-3}
16	7.4	. . .	2.9×10^{-4}
17	9.1	9.0	4.8×10^{-3}
18	7.6	. . .	4.7×10^{-3}
19	7.2	9.2	5.2×10^{-3}
20	8.1	9.1	7.5×10^{-3}
21	8.5	9.1	8.3×10^{-3}
22	9.4	9.1	2.9×10^{-3}
23	1.0×10^{1}	9.3	1.2×10^{-3}
24	1.0×10^{1}	8.9	2.1×10^{-2}
25	1.0×10^{1}	9.1	6.5×10^{-4}
26	1.1×10^{1}	. . .	2.8×10^{-3}
27	1.1×10^{1}	8.9	2.4×10^{-3}
28	1.2×10^{1}	9.2	4.9×10^{-4}
29	1.2×10^{1}	8.8	1.1×10^{-3}

Table 2: This table gives the gas phase relative oxygen abundance (quoted as $12 + \log \mathrm{O/H}$), star formation rates (in $M_\odot$ per year), and galactocentric radius for 29 H II regions in NGC 4725. [a]The star formation rate for H II region number 8 is not corrected for extinction because I could not obtain a measurement of Hβ for this region.

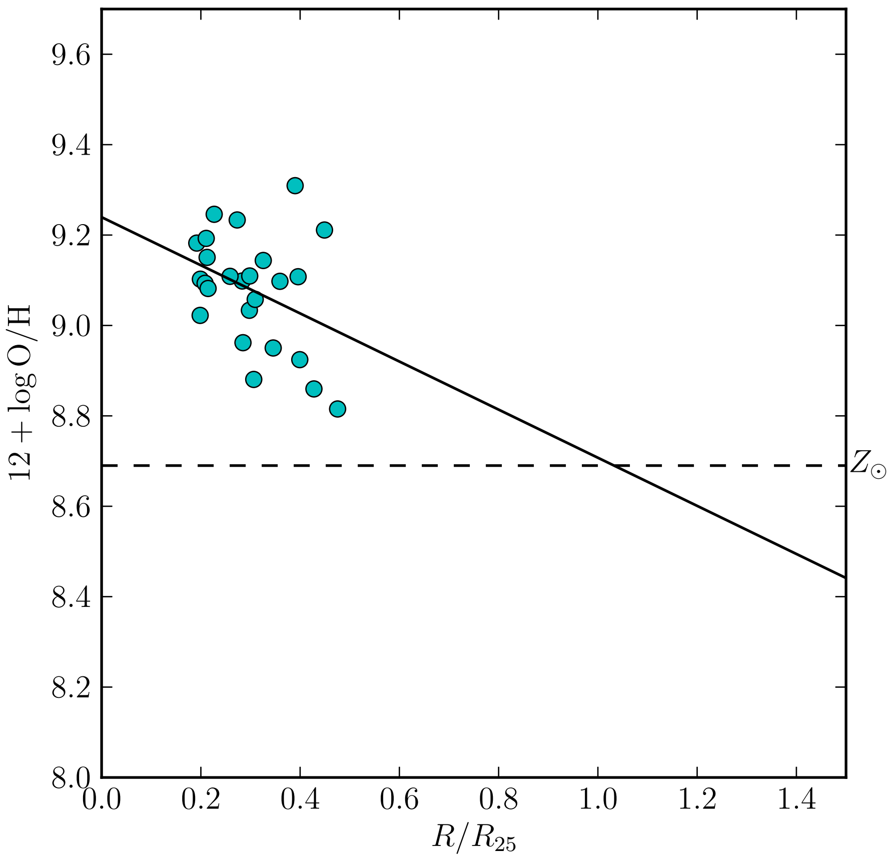

Figure 9: Gas phase relative oxygen abundance for H II regions in NGC 4725 as a function of galactocentric radius is shown here. Each plotted point of data represents a single H II region. Solar metallicity ($Z_\odot$) is marked with a dashed line and a fitted linear function is plotted as a solid line.

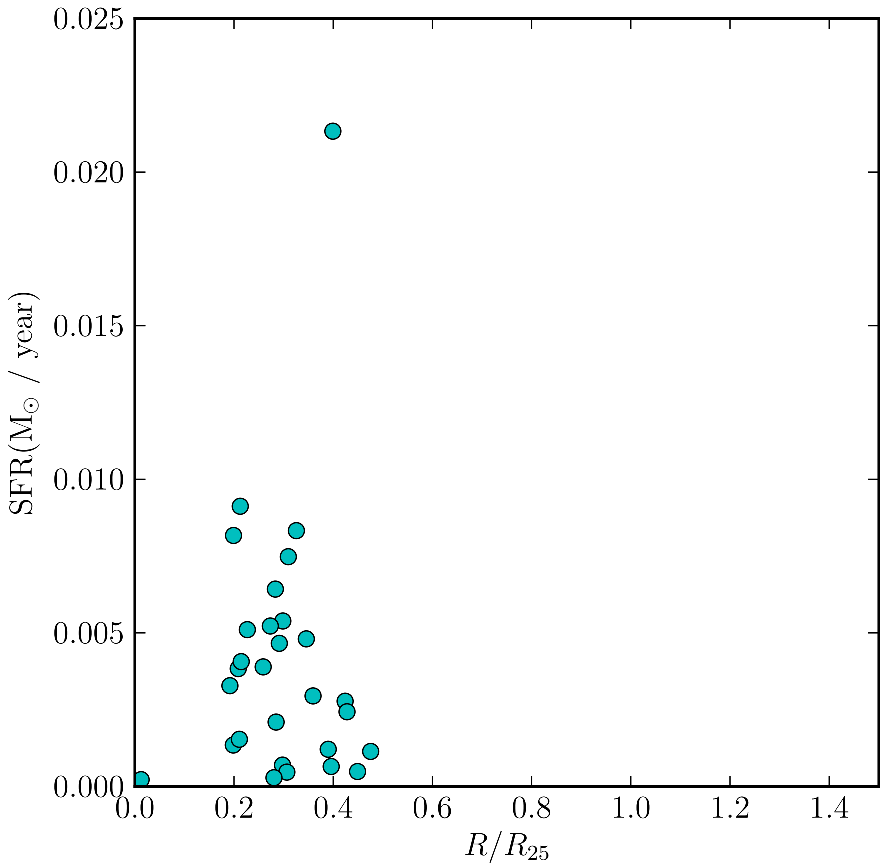

Figure 10: Rates of star formation in solar masses per year for H II regions in NGC 4725 as a function of galactocentric radius is shown here. Each plotted point of data represents a single H II region.

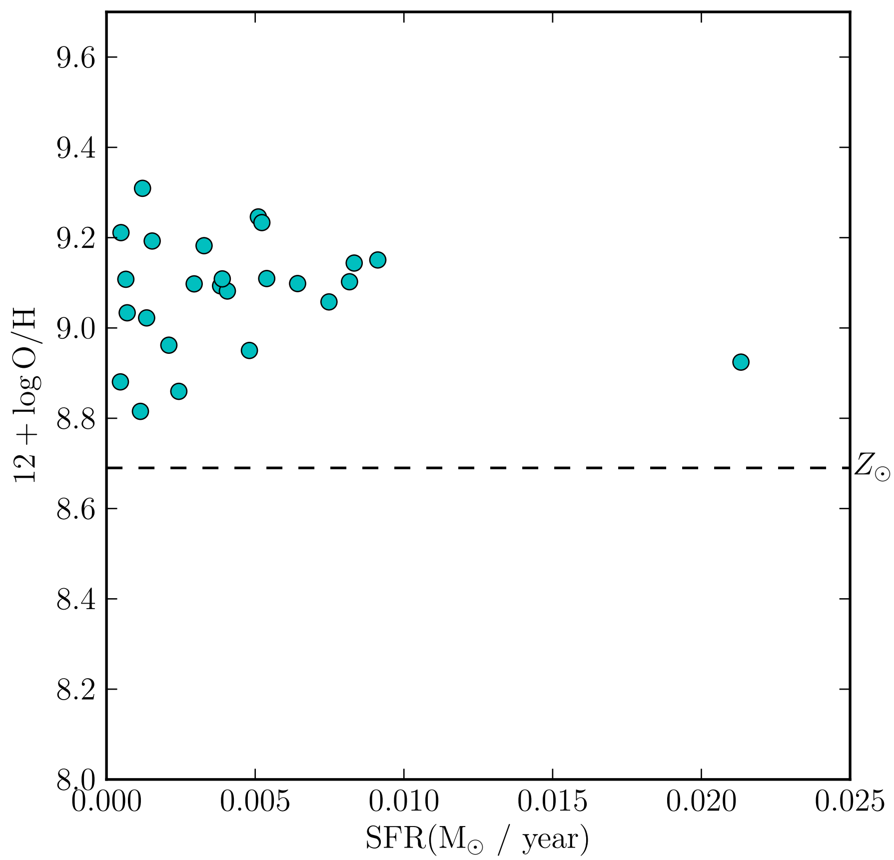

Figure 11: Gas phase relative oxygen abundance, quoted as $12 + \log \mathrm{O/H}$, as a function of rates of star formation in solar masses per year for H II regions in NGC 4725 is shown here. Each plotted point of data represents a single H II region. Solar oxygen abundance is plotted as a dashed line.

formation, and the result is a less severe slope.

From this interpretation, some conclusions about NGC 4725 can be made. Primarily, NGC 4725 is a galaxy with an interstellar medium that is quite metallic; the gas that is driving star formation in this sample appears to have a relative oxygen abundance of $12 + \log \mathrm{O/H} \approx 9.1$, well above solar metallicity. Care should be taken with this interpretation, however, because of the narrow range of radii that this sample represents. These H II regions are all either directly on or near the ring of NGC 4725, where the total Hα emission from the galaxy is strongly concentrated, a phenomenon common to all galactic rings [1]. Because of this, the ring will have a much richer history of star formation than the surrounding galaxy. If the gas in the ring that is enriched by this is not contributed back to the host galaxy efficiently, then the metallicity of the ring will be higher then the metallicity of the rest of the galaxy. As a result this high metallicity may not be characteristic of the galaxy outside of the ring.

3.4 Comparisons

Table 3 shows numerical information for NGC 3169 and NGC 4725 along with 12 other galaxies taken from ZKH [32]. Included in the table are the gradient of the relative oxygen abundance profile in dex per R_{25} and a characteristic abundance, defined as the value of $12 + \log \mathrm{O/H}$ at 0.4 R_{25}, calculated from an exponential profile fit to the H II regions of that galaxy. Neither NGC 3169 and NGC 4725 are particularly exceptional when compared to the other galaxies on this table. NGC 3169 has a nearly flat gradient, which is unusual. However, NGC 2541 has moderately strong positive gradient, making a flat gradient well within the range of the other galaxies. The gradient of NGC 4725 is moderate when compared to the other galaxies. In terms of characteristic abundance, NGC 3169 is low and NGC 4725 is high, but neither is an extreme value. In the 12 galaxies in this data set, two galaxies have higher characteristic abundances than NGC 4725 and three galaxies have lower characteristic abundances than NGC 3169.

Figure 12 shows the gas phase relative oxygen abundance for the H II regions in both NGC 3169 and NGC 4725 plotted together, along with a large number of H II regions taken from ZKH. H II regions from NGC 3169 are plotted as red triangles and H II regions from NGC 4725 are plotted as cyan circles. The remainder of the H II regions are plotted as open diamonds. On the basis of this plot, high abundance is only possible near the center of a galaxy, but relatively low metallicity is also possible at these radii. The very highest levels of oxygen abundance are not found all the way in the center of the galaxy, but rather at about 0.1 R_{25}. Few H II regions within this radius are available, however, so more data may paint a different picture.

This aside, as radius increases, maximum abundance decreases, along with minimum abundance, which decreases more rapidly with radius than maximum abundance does. This broadens the range of possible abundances for a given radius. Starting at around 0.2 R_{25}, the minimum $12 + \log \mathrm{O/H}$ in this data set levels off at $12 + \log \mathrm{O/H} \approx 8.2$. The maximum abundance for a given radius decreases continuously to a radius of approximately 0.9 R_{25}. Beyond this point, this data set contains only three H II regions and conclusions are difficult

Name	Gradient (dex/R_{25})	Metalicity at $0.4R_{25}$	Hubble Type	Bar	Ring	Environment	Number of Regions
NGC 3169	1.90×10^{-3}	8.60	Sa	A	s	pair	18
NGC 41725	-5.32×10^{-1}	9.03	Sab	AB	r	pair	29
NGC 2903	-1.10	9.11	Sbc	AB	rs	group	20
NGC 4258	-4.04×10^{-1}	8.90	Sbc	AB	s	pair	9
NGC 2541	2.07×10^{-1}	8.48	Scd	A	s	group	19
NGC 3621	-7.62×10^{-1}	8.84	Sd	A	s	unknown	7
NGC 4559	-5.53×10^{-1}	8.65	Scd	AB	rs	pair	20
NGC 3319	-7.61×10^{-1}	8.49	Scd	B	rs	isolated	13
NGC 5033	-9.06×10^{-1}	8.84	Sc	A	s	isolated	8
NGC 3344	-1.42	8.83	Sbc	AB	r	isolated	9
NGC 3184	-1.38	9.19	Scd	AB	rs	isolated	10
NGC 3521	-1.61	8.97	Sbc	AB	rs	isolated	10
NGC 925	-2.37×10^{-1}	8.55	Sd	AB	s	group	9
NGC 3198	-9.26×10^{-1}	8.76	Sc	B	rs	isolated	15

Table 3: This table contains the gradient of the abundance profile as well as the characteristic abundance for both NGC 3169 and NGC 4725 as well as 12 additional galaxies taken from ZKH. Type and environment information taken from various sources ([32], [4], [5], [29], [7]).

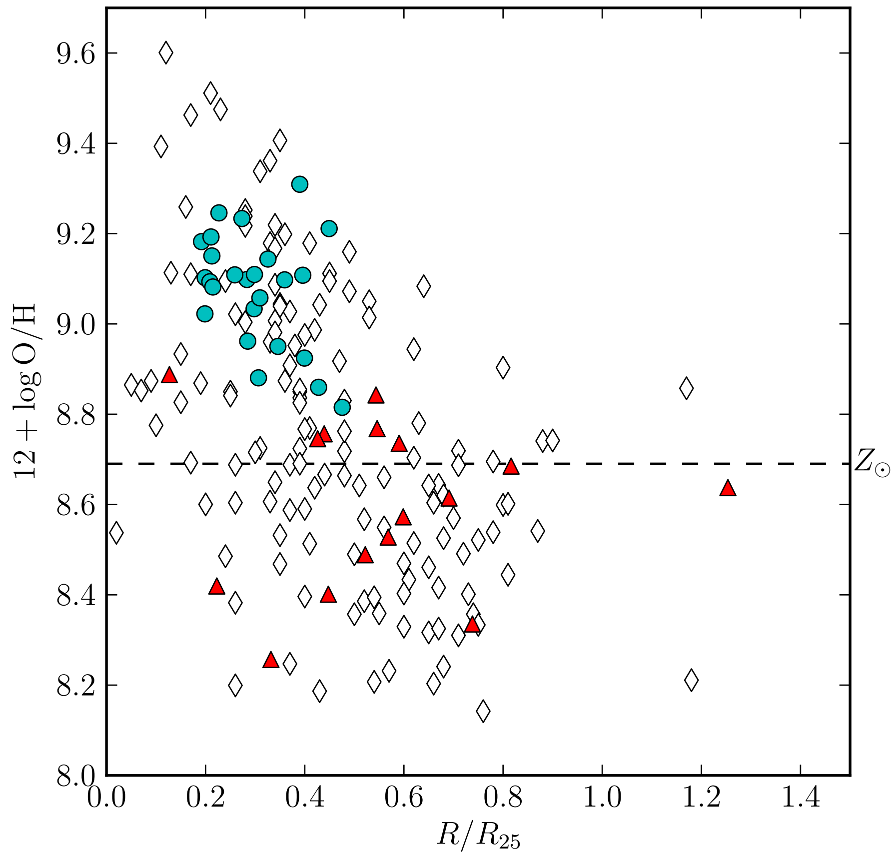

Figure 12: This plot shows the gas phase relative oxygen abundance, quoted as $12+\log \mathrm{O/H}$, for H II regions from many different galaxies. H II regions taken from NGC 3169 are plotted as red triangles; H II regions taken from NGC 4725 are plotted as cyan circles. A large body of H II regions from different galaxies is plotted as hollow diamonds underneath these two galaxies. Data for these H II regions is taken from ZKH [32].

to draw. From the data available, it appears that the maximum abundance levels off at $12 + \log \mathrm{O/H} \approx 8.9$ and H II regions at radii greater than R_{25} can have a reasonably wide range of abundances.

The H II regions from NGC 3169 and NGC 4725 fit well within this context. The H II regions from NGC 3169 occupy the lower portion of this plot. The majority of these H II regions fall well within the envelope sketched out by the ZKH H II regions. A set of three H II regions trace the lower boundary of low oxygen abudance and low radius and one other H II region is at very high radius, with slightly subsolar oxygen abundance levels. The H II regions from NGC 4725 on the other hand, occupy the upper portion of this plot. They cover roughly the whole range of available positions with relatively high abundance and relatively small radii. Two H II regions lie on the upper boundary of high oxygen abundance. On the low abundance side, the H II regions from NGC 4725 are roughly continuous with the high abundance H II regions from NGC 3169. There is very little overlap between these two populations.

Plotted alongside of NGC 3169, it becomes very easy to see how restricted the range of radii that the H II regions in NGC 4725 have. As discussed earlier, this is due to the presence of a strong ring structure in NGC 4725. Buta (1988) observed that Hα emission in ringed galaxies has its greatest intensity and concentration within the rings [1]. If gas transport out of the ring is poor, this is a possible cause of the high oxygen abundance seen in NGC 4725.

Given the distinctness of the two populations of NGC 3169 and NGC 4725, I wanted to investigate whether there were any corresponding trend across the whole of the data set. Table 4 shows data for the galaxies as broken up into various subgroups. The average value and the standard deviation for both the abundance gradient and the characteristic abundance are shown. Four types of classification are shown: Hubble type, bar status, ring status, and the closest form of galactic association that the host galaxy is a part of.

Figure 13 shows the same basic plot as figure 12, with the H II regions coded by the Hubble type of their parent galaxy. H II regions from galaxies with types Sa and Sab are plotted as yellow triangles, types Sb and Sbc are plotted as red narrow diamonds, types Sc and Scd are plotted as magenta pentagons and type Sd are plotted as blue square diamonds.

No overall trend is apparent on this graph. NGC 3169 and NGC 4725 are the only galaxies with types Sa or Sab, and they occupy approximately the whole range of possible values. In the next grouping along the Hubble sequence, there are four type Sbc galaxies in the data, plotted as red narrow diamonds. Only four H II regions from these galaxies have subsolar metallicity. Additionally, H II regions with the highest metallicities in the data are from these galaxies. The data in table 4 also reflect this: both the gradient and characteristic abundance for Sb and Sbc galaxies are the highest out of the four groupings, with a comparable standard deviation. Later types (Sc, Scd, and Sd) occupy roughly the whole range of possible values, similar to the H II regions from early type galaxies.

ZKH, identified the same trend in their data, and hypothesized that this was due to several effects. First, that early type galaxies have shallow gradients because they have exhausted most of their supplies of gas. Second, that late type galaxies have shallow gradients because their size is close to a mixing length. As a result, only intermediate type galaxies

Hubble Type	Gradient (dex/R_{25})	Standard Deviation	Metalicity at $0.4R_{25}$	Standard Deviation
Sc and Scd	-7.2×10^{-1}	4.8×10^{-1}	8.7	2.4×10^{-1}
Sa and Sab	-2.7×10^{-1}	2.7×10^{-1}	8.8	2.1×10^{-1}
Sb and Sbc	-1.1	4.6×10^{-1}	9.0	1.0×10^{-1}
Sd	-5.0×10^{-1}	2.6×10^{-1}	8.7	1.5×10^{-1}
Ring	**Gradient (dex/R_{25})**	**Standard Deviation**	**Metalicity at $0.4R_{25}$**	**Standard Deviation**
Intermediate Type	-1.1	3.6×10^{-1}	8.9	2.5×10^{-1}
S-Shaped	-3.5×10^{-1}	3.9×10^{-1}	8.7	1.6×10^{-1}
Ringed	-9.8×10^{-1}	4.5×10^{-1}	8.9	9.8×10^{-2}
Bar	**Gradient (dex/R_{25})**	**Standard Deviation**	**Metalicity at $0.4R_{25}$**	**Standard Deviation**
No Bar	-3.6×10^{-1}	4.8×10^{-1}	8.7	1.6×10^{-1}
Strongly Barred	-8.4×10^{-1}	8.2×10^{-2}	8.6	1.3×10^{-1}
Weakly Barred	-9.0×10^{-1}	5.0×10^{-1}	8.9	2.1×10^{-1}
Environment	**Gradient (dex/R_{25})**	**Standard Deviation**	**Metalicity at $0.4R_{25}$**	**Standard Deviation**
Pair	-3.7×10^{-1}	2.2×10^{-1}	8.8	1.8×10^{-1}
Group	-3.8×10^{-1}	5.4×10^{-1}	8.7	2.8×10^{-1}
Isolated	-1.2	3.1×10^{-1}	8.8	2.1×10^{-1}

Table 4: This table shows averages and standard deviations for the oxygen abundance gradient and the characteristic abundance for galaxies by various properties of the galaxy. Four types of classification are shown: Hubble type, bar status, ring status, and the closest form of galactic association that the host galaxy is a part of. All abundances are the gas phase relative oxygen abundance, quoted as $12 + \log \mathrm{O/H}$. Characteristic abundance is the abundance at forty percent of the isophotal radius, R_{25}.

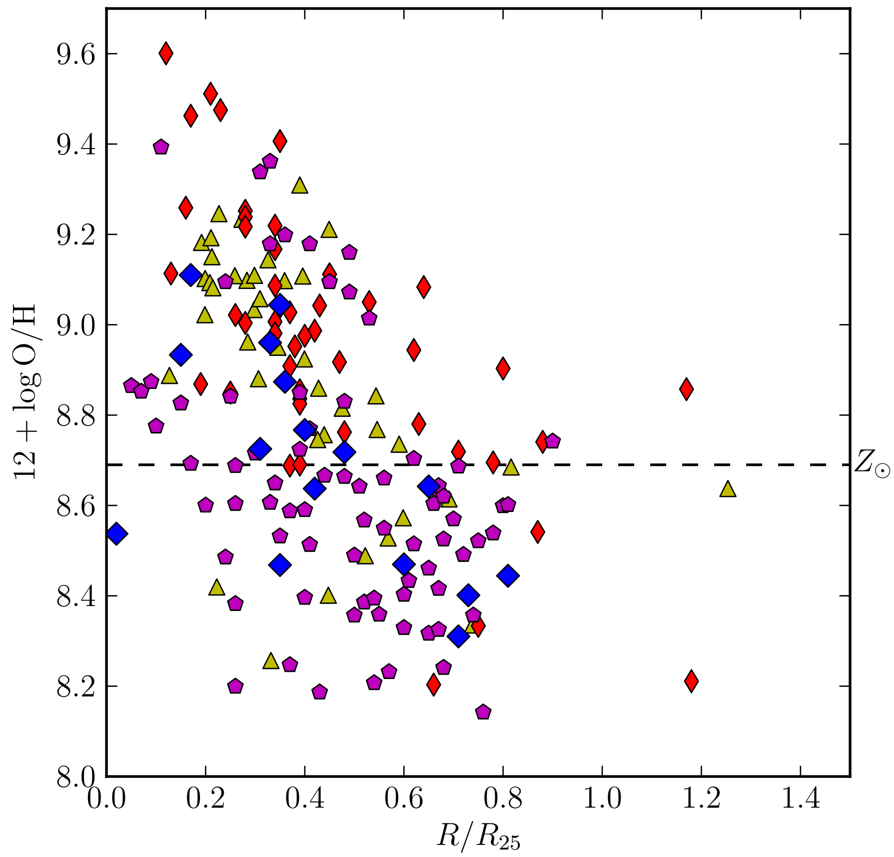

Figure 13: This plot shows the gas phase relative oxygen abundance, quoted as $12+\log \mathrm{O/H}$, for H II regions from many different galaxies, broken down by Hubble type. H II regions from galaxies with types Sa and Sab are plotted as yellow triangles. Types Sb and Sbc are plotted as red narrow diamonds. Types Sc and Scd are plotted as magenta pentagons. Type Sd are plotted as blue square diamonds.

can have steep gradients [32].

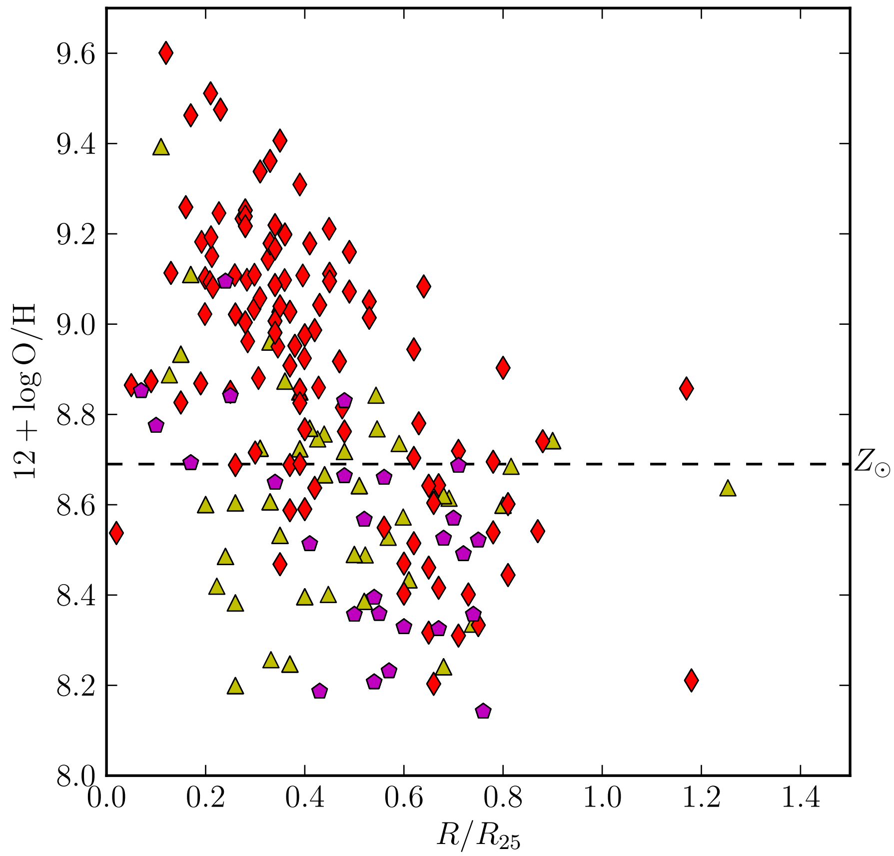

Figure 14: This plot shows the gas phase relative oxygen abundance, quoted as $12+\log \mathrm{O/H}$, for H II regions from many different galaxies, categorized by the presence of bar in the host galaxy. H II regions from a galaxy without a bar are plotted as yellow triangles. Those from a galaxy with a strong bar are plotted as magenta pentagons. Those from a galaxy with a weak or intermediate bar are plotted as red diamonds.

Figure 14 shows the same plot of H II regions from fourteen galaxies, this time categorized by the presence of a bar in the host galaxy. H II regions from a galaxy without a bar, type SA, are plotted as yellow triangles. Those from a galaxy with a strong bar, type SB, are plotted as magenta pentagons. Those from a galaxy with a weak or intermediate bar, type SAB, are plotted as red diamonds. Again, no clear trends are apparent, aside from the fact that the majority of the data are from galaxies with intermediate type bars. The H II regions from these galaxies comprise almost all of the H II regions with $12 + \log \mathrm{O/H} < 9.1$, although they also occupy most of the range of possible values as well. I can see no clear distinction between the H II regions that come from strongly barred galaxies and the H II regions that come from galaxies without bars. Table 4 shows that weakly barred galaxies have the steepest gradient and highest abundances. Strongly barred galaxies have the lowest abundances, while galaxies without a bar have the shallowest gradients.

This is at odds with the results of ZKH, who found that barred galaxies have generally flatter gradients than unbarred galaxies [32]. If any trend is present in these data, I find the opposite, that steep gradients are associated with barred galaxies.

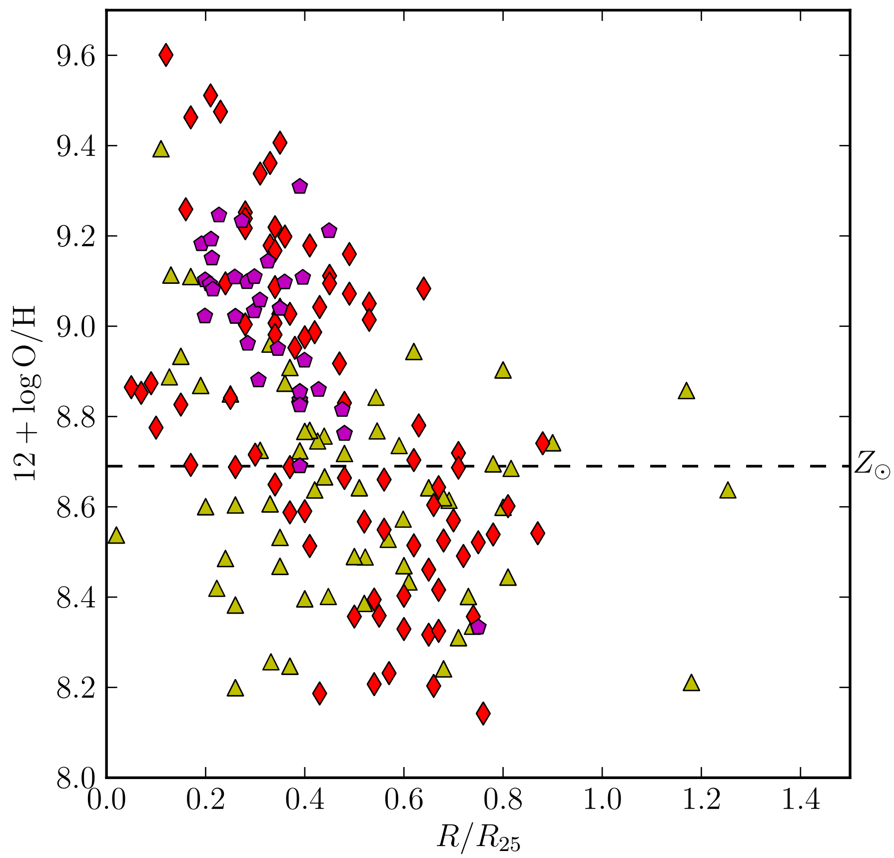

Figure 15: This plot shows the gas phase relative oxygen abundance, quoted as $12+\log \mathrm{O/H}$, for H II regions from many different galaxies, categorized by the presence of ring in the host galaxy. H II regions from an S-shpaed galaxy are plotted as yellow triangles. Those from a galaxy with a strong ring are plotted as magenta pentagons. Those from a galaxy with an intermediate ring are plotted as red diamonds.

Figure 15 shows the same plot of H II regions from fourteen galaxies, this time categorized by the presence of a ring in the host galaxy. H II regions from an S-shaped galaxy, type

S(s), are plotted as yellow triangles. Those from a galaxy with an inner ring, type S(r), are plotted as magenta pentagons. Those from a galaxy with an intermediate ring, type S(rs), are plotted as red diamonds.

This shows more of a possible trend. H II regions from S-shaped galaxies almost all have $12 + \log \mathrm{O/H} < 9.0$, and also form the extreme for low oxygen abundance/low radius on this plot. By contrast, H II regions from galaxies with a strong ring almost all have supersolar oxygen abundance. Intermediate type galaxies span a wide range, account for the highest levels of oxygen abundance, and do not drop off as quickly with radius as S-shaped galaxies do. This suggests that the presence of a ring results in high metallicities. Table 4 shows that ringed and intermediate type galaxies have very similar gradients and characteristic abundances. S-shaped galaxies have a much shallower gradient, and a somewhat lower characteristic abundance.

The sample of strongly ringed galaxies is limited to two galaxies, which would give low confidence in this picture. However, both the gradient and the characteristic abundance, as well as their standard deviations, are nearly the same for both strongly ringed and intermediate type galaxies. This suggests that the effects of a ring are independent of the strength of the ring, as long as a ring is present. This effect could account for a large portion of the difference seen between NGC 3169 and NGC 4725.

Figure 16 shows the same plot of H II regions from fourteen galaxies, this time categorized by the closest form of galactic association of the host galaxy. H II regions from an isolated galaxy are plotted as yellow triangles. Those from a galaxy in a group environment are plotted as red diamonds. Those from a galaxy that is part of a galaxy pair are plotted as magenta pentagons. All three types occupy the full range of values available, and no trend can be seen. The only notable feature is that only galaxies that are part of a pair show high radius H II regions, although the sample size is small. This makes some sense: galaxies undergoing interaction have extended arms formed as material is drawn out from the disk during interaction. Star formation is more likely to happen in a region like this than in an ordinary high radius region.

Table 4 reveals that while characteristic abundance is the same for all three categories, isolated galaxies have a much steeper gradient.

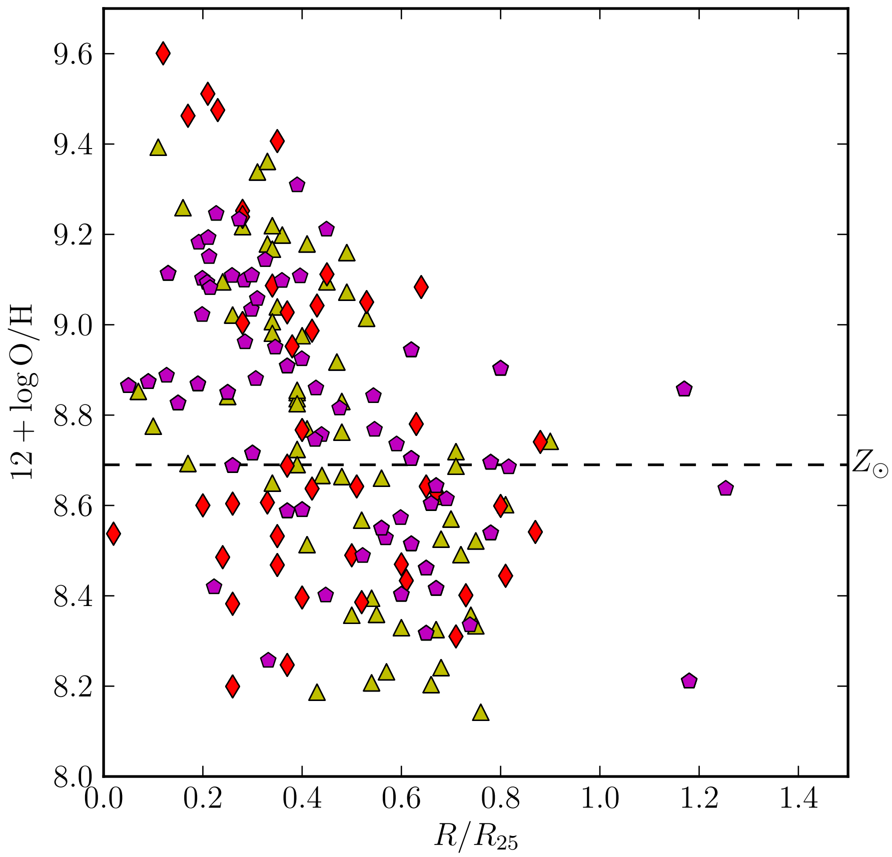

Figure 16: This plot shows the gas phase relative oxygen abundance, quoted as $12+\log \mathrm{O/H}$, for H II regions from many different galaxies, categorized by the closest form of galactic association of the host galaxy. H II regions from an isolated galaxy are plotted as yellow triangles. Those from a galaxy in a group environment are plotted as red diamonds. Those from a galaxy that is part of a galaxy pair are plotted as magenta pentagons.

4 Conclusions

In trying to understand the long term history and evolution of galaxies there are many galactic properties which must be considered. Among the most important properties of galaxies are the Hubble type, the total mass of the galaxy, the galactic environment, gas content of the galaxy, locations and rates of star formation, and the metallicity of the galaxy. Of these, the last three are the most mutable, and form an important dynamic feedback loop which begins with inflow of gas content into a region causing star formation to occur. Star formation results in the production of metals, which enrich the remaining gas content. The increased metallicity of the gas content in the galaxy then affects the details of star formation in the next iteration of the process.

Understanding the details of this feedback cycle has important implications for total behavior of material within a galaxy over time. A key link in this chain is the behavior and distribution of metals within a galaxy. In order to get an understanding of this, I have analyzed the gas phase relative oxygen abundance for a large number of H II regions within two galaxies, NGC 3169 and NGC 4725. I found that NGC 3169 is characterized by a relatively low level of oxygen abundance and a flat abundance profile. I interpret this as indicating a galaxy with a well mixed interstellar medium. The low abundance of NGC 3169 could be the result of a number of factors including low levels of metal formation or depletion of metals through solar winds, resulting in enrichment of the intergalactic medium.

I found that NGC 4725 is characterized by high levels of oxygen abundance and a medium abundance gradient. Possible selection biases are present here, however, given the fact that all of the H II regions available for NGC 4725 on or near the ring structure that is present NGC 4725. My results for both of these galaxies are in good agreement with or well within the range of values derived by earlier work such as Zaritsky, Kennicutt and Huchra (1994) (ZKH) [32].

Lacking from this sample is a clear view of the abundance of at high radii: the large majority of the H II regions in the data I have analyzed here have a radius of under 1.0 R_{25}. A data set with significant numbers of H II regions with $R > R_{25}$ is needed to needed to address this uncertainty.

I also analyzed these two galaxies for a relationship between oxygen abundance and rates of star formation, in an effort to characterize better the interplay between these two galactic properties. In both galaxies I found that the majority of H II regions has a low rate of star formation and that, taken as a whole, they sketched out a triangular envelope on the chart of SFR versus $12 + \log \mathrm{O/H}$. The apparent indication is that within a certain range of low star formation activity, the regions with higher rates of star formation have more tightly constrained oxygen abundances. The physical processes that produce this relation are unclear, but it has something to do with the metallicity of the interstellar medium and gas transport. One possible situation is that an H II region with no active star formation can be a wide rage of metallicities. As interstellar medium flows into a region, star formation can star and the metallicity of the region is brought towards the metallicity of the interstellar medium of the galaxy.

As the rate of star formation in the region falls off again, either the metals formed by this star formation are kept within the H II region and the metallicity will increase or gas transport will result in these metals leaving the H II region and entering the interstellar medium, in which case the metallicity of the H II region will fall off again. Whether or not this is the physical process responsible for the triangle shaped envelope of allowed values is difficult to tell from the data used here.

Additionally, this is an effect that seems to show up only at low levels of star formation. Higher levels of star formation appear to be related to metallicity according to a different regime. I was only able to analyze a very small number of such regions in this data set, so drawing any conclusions about them is difficult. Due to the low levels of star formation corresponding to this effect, it could easily be the case that the star formation which occurs under this regime is negligible with regard to the global dynamics of the galaxy. Data which includes significantly higher levels of star formation is needed before any conclusions about the importance of this effect on a broader scale can be made.

Data for H II regions with higher rates of star formation as well as H II regions with larger galactocentric radii would significantly broaden the scope of what can be concluded from the data I have analyzed. These relations are critical to understanding the process of feedback between star formation and the interstellar medium. That this feedback cycle is poorly understood remains one of the major stumbling blocks in in the development of accurate models of galaxy evolution [13]. More detailed and broader ranging studies in this area are necessary to better understand the dynamics involved.

I have also combined my data for NGC 3169 and NGC 4725 with the data that ZKH published, and performed analysis of trends in abundance over four different properties of the host galaxy: Hubble type, bar status, ring status, and the closest form of galactic association that the host galaxy is part of. I found that intermediate Hubble types have the steepest abundance gradients, while both early and late type galaxies have shallow gradients, in agreement with ZKH. Additionally, I found that intermediate type galaxies also have much higher characteristic abundances, defined as the value of $12 + \log \mathrm{O/H}$ at 0.4 times the isophotal radius, R_{25}.

I found that the presence of a bar in the galaxy is linked to steeper gradients, which is at odds with the results of ZKH. This is particularly strange, given that the data I used is in large part a subset of the data they used. As a result, this seems unlikely to be a real phenomenon, but rather a result of selection effects and small sample size.

I found that isolated galaxies have much steeper gradients than galaxies in group environments or those engaged in pair interactions, but no difference in characteristic abundance among these groups. This suggests that interacting galaxies are better mixed or more depleted of their gas content: either of these is possible.

Finally, I found that the presence of a ring structure in the galaxy results in steeper abundance gradients, regardless of the strength of the ring. I found that the characteristic abundance is higher for galaxies with rings, but the difference is not as large. This is particularly relevant to the study of NGC 3169 and NGC 4725: NGC 4725 has a ring and NGC 3169 does not. Additionally, the difference in abundance gradient and characteristic abundance between the two galaxies is similar to the difference between ringed and nonringed galaxies. This suggests that the differences I see between these galaxies is in large part due to the presence or absence of a ring.

The fact that it doesn't seem to matter whether the ring is strong or intermediate is very interesting. Future work could attempt a more detailed classification of ring structure in order to see how strong the ring must be in order to affect abundances in this way. This would give insight into the physical nature of the processes associated with the ring that affect the abundance gradient.

With a large number of both galaxies and H II regions per galaxy, we can address these issues of correlations between the other properties of spiral galaxies and abundance. This allows for insight into the basic nature of the life cycle of galaxies: their formation and their evolution.

References

[1] R. Buta. The structure and dynamics of ringed galaxies. V - The kinematics of NGC 1512, NGC 3351, NGC 4725, and NGC 4736. *Astrophysical Journal Supplement Series*, 66:233–259, March 1988.

[2] D. Calzetti. UV opacity in nearby galaxies and application to distant galaxies. In W. H. Waller, editor, *American Institute of Physics Conference Series*, volume 408 of *American Institute of Physics Conference Series*, pages 403–412, May 1997.

[3] Bradley W. Carroll and Dale A. Ostlie. *An Introduction to Modern Astrophysics*. Pearson, second edition, 2007.

[4] G. de Vaucouleurs, A. de Vaucouleurs, and H. G. Corwin, Jr. *Second reference catalogue of bright galaxies. Containing information on 4,364 galaxies with references to papers published between 1964 and 1975.* University of Texas Press, 1976.

[5] G. de Vaucouleurs, A. de Vaucouleurs, H. G. Corwin, Jr., R. J. Buta, G. Paturel, and P. Fouque. *Third Reference Catalogue of Bright Galaxies*. Springer-Verlag, 1991.

[6] J. De Veny, P. Massey, and V. Sarajedini. *Multi-slits at Kitt Peak: A Manual for Designing and Using Entrance Masks for Low/Moderate-Resolution Spectroscopy*. Kitt Peak National Observatory, March 1996.

[7] R.-J. Dettmar and B. A. Skiff. Declining rotation curves in interacting galaxies. In *Evolution of Galaxies and their Environment*, pages 251–252, January 1993.

[8] D. R. Garnett. Element abundances in nearby galaxies. In C. Esteban, R. García López, A. Herrero, & F. Sánchez , editor, *Cosmochemistry. The melting pot of the elements*, pages 171–216, 2004.

[9] R. L. Gilliland. Details of Noise Sources and Reduction Processes. In S. B. Howell, editor, *Astronomical CCD Observing and Reduction Techniques*, volume 23 of *Astronomical Society of the Pacific Conference Series*, pages 68–+, 1992.

[10] R. M. Gonzalez-Delgado, E. Perez, C. Tadhunter, J. M. Vilchez, and J. M. Rodriguez-Espinosa. H II Region Population in a Sample of Nearby Galaxies with Nuclear Activity. I. Data and General Results. *Astrophysical Journal Supplement Series*, 108:155–+, January 1997.

[11] Charles Hoopes, David Thilker, and Salman Hameed. Revealing the fossil record of chemical evolution in nearby galaxies. NOAO Observing Proposal, September 2003.

[12] W. C. Keel and E. Hummel. Jets and loops from the nucleus of NGC 2655. *Astronomy and Astrophysics*, 194:90–98, April 1988.

[13] R. C. Kennicutt, Jr. Star Formation in Galaxies Along the Hubble Sequence. *Annual Review of Astronomy and Astrophysics*, 36:189–232, 1998.

[14] L. J. Kewley and M. A. Dopita. Using Strong Lines to Estimate Abundances in Extragalactic H II Regions and Starburst Galaxies. *Astrophysical Journal Supplement Series*, 142:35–52, September 2002.

[15] L. J. Kewley, B. Groves, G. Kauffmann, and T. Heckman. The host galaxies and classification of active galactic nuclei. *Monthly Notices of the Royal Astronomical Society*, 372:961–976, November 2006.

[16] P. Massey, C. Gronwall, and C. A. Pilachowski. The spectrum of the Kitt Peak night sky. *Publications of the Astronomical Society of the Pacific*, 102:1046–1051, August 1990.

[17] P. Massey, K. Strobel, J. V. Barnes, and E. Anderson. Spectrophotometric standards. *The Astrophysical Journal*, 328:315–333, May 1988.

[18] T. Nagao, R. Maiolino, and A. Marconi. Gas metallicity diagnostics in star-forming galaxies. *Astronomy and Astrophysics*, 459:85–101, November 2006.

[19] National Optical Astronomy Observatories. *IRAF `splot` Help Page*, 2.11 edition, July 1995. `http://iraf.noao.edu/scripts/irafhelp?splot`.

[20] National Optical Astronomy Observatories. *IRAF `combine` Help Page*, 2.11 edition, August 1996. `http://iraf.noao.edu/scripts/irafhelp?combine`.

[21] National Optical Astronomy Observatories. *IRAF `standard` Help Page*, 2.10.4 edition, January 2000. `http://iraf.noao.edu/scripts/irafhelp?standard`.

[22] National Optical Astronomy Observatories. *Helium - Neon - Argon Spectral Atlas*. `http://www.noao.edu/kpno/specatlas/henear/henear.html`.

[23] B. E. J. Pagel, M. G. Edmunds, D. E. Blackwell, M. S. Chun, and G. Smith. On the composition of H II regions in southern galaxies. I - NGC 300 and 1365. *Monthly Notices of the Royal Astronomical Society*, 189:95–113, October 1979.

[24] L. S. Pilyugin, J. M. Vílchez, and T. Contini. Oxygen and nitrogen abundances in nearby galaxies. Correlations between oxygen abundance and macroscopic properties. *Astronomy and Astrophysics*, 425:849–869, October 2004.

[25] C. S. Reynolds, M. J. Ward, A. C. Fabian, and A. Celotti. A multiwavelength study of the Seyfert 1 galaxy MCG-6-30-15. *Monthly Notices of the Royal Astronomical Society*, 291:403–+, November 1997.

[26] A. Saha, F. Thim, G. A. Tammann, B. Reindl, and A. Sandage. Cepheid Distances to SNe Ia Host Galaxies Based on a Revised Photometric Zero Point of the HST WFPC2 and New PL Relations and Metallicity Corrections. *Astrophysical Journal Supplement Series*, 165:108–137, July 2006.

[27] O. K. Sil'chenko and V. L. Afanasiev. Central regions of the early-type galaxies in the NGC 3169 group. *Astronomy Letters*, 32:534–544, August 2006.

[28] T. Storchi-Bergmann. On the ratio (N II)/H-alpha in the nucleus of Seyfert 2 and LINER galaxies. *Monthly Notices of the Royal Astronomical Society*, 249:404–408, April 1991.

[29] L. Verdes-Montenegro, A. Bosma, and E. Athanassoula. A detailed study of the ringed galaxy NGC 3344. *Astronomy and Astrophysics*, 356:827–839, April 2000.

[30] B. M. H. R. Wevers, P. N. Appleton, R. D. Davies, and L. Hart. Neutral hydrogen observations of the interacting galaxies NGC 4725 and NGC 4747. *Astronomy and Astrophysics*, 140:125–140, November 1984.

[31] W. M. Wood-Vasey, A. S. Friedman, J. S. Bloom, M. Hicken, M. Modjaz, R. P. Kirshner, D. L. Starr, C. H. Blake, E. E. Falco, A. H. Szentgyorgyi, P. Challis, S. Blondin, K. S. Mandel, and A. Rest. Type Ia Supernovae Are Good Standard Candles in the Near Infrared: Evidence from PAIRITEL. *The Astrophysical Journal*, 689:377–390, December 2008.

[32] D. Zaritsky, R. C. Kennicutt, Jr., and J. P. Huchra. H II regions and the abundance properties of spiral galaxies. *The Astrophysical Journal*, 420:87–109, January 1994.

A Source Code

The code I have written is split between a library (`mslit`) and two scripts (`reduce.py` and `modify_sky.py`). The primary script is `reduce.py`, which allows for efficient reduction of the multi-slit spectroscopic data that I worked on, while `modify_sky.py` is a basic wrapper around a function in `mslit`.

In designing this code I wanted everything to be as easily repeatable as possible. I've put as much configuration data as I can in the filesystem, where it can be used repeatedly without needing to re-enter it. Also, each step of data reduction creates new subdirectories for it's output. If at any time a step needs to be redone, simply deleting the folder associated with that step will reset the data. The original data is never modified. For the data files the user must create, I chose YAML formatting for it's human-readability.

As a note, the set of images for a galaxy observation and the image for a calibration star observation are taken through identical processes in the data reduction. Unless I say otherwise, I will use "galaxy" as shorthand for both the galaxy observations and the calibration star observations.

At the beginning of data reduction, there is a simple directory with many fits images in it. Throughout this guide, I will refer this directory as the base directory, '`./`'. Additionally, if I do not specify otherwise, all functions referenced are IRAF functions.

A.1 Using the `reduce.py`

The main script, `reduce.py` provides a number of different commands: `zeroflat`, `init`, `extract`, `disp`, `sky`, `calibrate`, and `analyze`. In basic use, it takes two arguments: a command and the path to the data. All of the commands except for `zeroflat` and `analyze` can take an optional extra parameter, the `--name` or `-n` parameter. This is for use in the case where you are only working on one galaxy from a set. Specify the name of the galaxy you want to run this step on with `-n`, and only that galaxy will be taken through the step. If no name is given, then all galaxies in the set will be acted on.

A.1.1 Setup

To begin, prior to running any of the code, you need to setup a few files that will help `reduce.py` know what to do. First, create a subdirectory called lists. Within `./lists/` there needs to be various lists of the type that IRAF will accept in IRAF's @list notation: plain text lists of files, one file name per line. At a minimum, there should be a list for each of the following: the set of flats, the set of zero images, the calibration star observation, the comparison lamp observation, and the galaxy observations. These lists are necessary even if there is only one item in the list. All lists can be arbitrarily named. I named them as the objects that they represented observations of, for clarity. There also needs to be a bad pixel mask, which can also be named anything. The bad pixel mask I used is `./Mask.pl`.

You also need to create a file that tells `reduce.py` what galaxies take what flats, zeros, etc. This file is `./input/groups.yaml`. This is a YAML formatted file with a list of dictionaries, one for each of the galaxy observations in the set. Each dictionary need to have the following keys: galaxy, zero, flat, star, mask, and lamp. The values for each of these are the names of the corresponding list in `./lists/`, except for mask. Galaxy, zero, and flat are straightforward. Star is the calibration star, and lamp is the HeNeAr comparison lamp. The value of mask is just the name of the bad pixel mask, as it doesn't have a corresponding list. An example of this file is:

```
- galaxy: ngc3169
  zero: Zero
  flat: Flat1
  star: feige34
  mask: Mask.pl
  lamp: henear1

- galaxy: ngc4725
  zero: Zero
  flat: Flat2
  star: pg1708+602
  mask: Mask.pl
  lamp: henear2
```

A.1.2 First Step: `reduce.py zeroflat`

Now run the `zeroflat` command. This command will first collect all the zero images and flats, combine them appropriately using `zerocombine` and `flatcombine`, and then apply the bad pixel mask to the combined images using `fixpix`.

A.1.3 First Step: `reduce.py init`

The `init` will create a subdirectory for each galaxy and a copy of each of the galaxy's images will be placed in it using `imcopy`. The bad pixel mask will be applied to these using `fixpix`. Each image is run through `ccdproc`, applying zero correction and flat fielding. Finally, all the images are combined using `combine` and the resulting image is saved as `base.fits` (for example, `./ngc3169/base.fits`).

A.1.4 Second Step: `reduce.py extract`

Before the `extract` step can be taken, two more files need to be created for each galaxy. First you need to go to the output of the MSLIT run that was used to generate the slits in the first place, saved as .out files in the data set I used. Within the output there are multiple plate definitions, and it is necessary to identify which one is the plate that your image uses.

I did this by using the image headers which record the angular position of the plate at the time of the observation. Match this up to the position angle of one of the plate definitions in the data set. From this, create a new file that begins with the field headers. The first couple lines of the file should look like this:

```
 OBJ  NAME          RA  (2000)  DEC         XLO     XHI (MM) Y

  70 HIIREGION  10:14:22.55 +03:29:40.4  -16.774 -15.792  -4.954
  68 HIIREGION  10:14:22.68 +03:29:32.5  -15.391 -14.258  -5.658
   4 NIGHTSKY   10:14:20.90 +03:29:14.9  -11.830 -10.320  -2.878
   0 SETUP      10:14:20.99 +03:28:12.7   -9.584  -9.206  -7.827
```

One caveat, however, is that any objects labeled in the original output 'CHECK STAR' must have the space removed so that they become 'CHECKSTAR'. This is because the parser I wrote is very basic and simply splits the line on any whitespace. If there are any other fields with spaces in them, the space must also be removed from these. Save this new file as `./input/name.out`, replacing name with the name you are using for the galaxy. For example, `./input/ngc3169.out`.

The second file is trickier. This is a file that tells the code where on the image the light that has come through the slits is. It does this by specifying, in pixel coordinates, two columns and where the light begins on the lowest strip of light and ends on the highest strip of light. It should be saved as `./input/name-pixel.yaml`, where name is replaced with the name of the galaxy, as in the .out file. An example of this file is:

```
- column: 1500
  start: 106
  end: 454

- column: 450
  start: 116
  end: 458
```

In this example, the two columns being specified are column 450 and column 1500. Column 1500 begins at the pixel (1500, 106) and ends at the pixel (1500,454), while column 450 runs from pixel (450, 116) to pixel (450, 458). These values are very tricky to get right, and this next step must be run iteratively until a satisfactory version is found. Neither of these files need to be created for the calibration stars, because their plates are the same as the galaxy's plates, so these files can be reused.

Once these files are created, `reduce.py extract` can be run. First this will use the information from the two new files to calculate the locations of every strip on the image. The .out file lists the physical size of the slits that are placed over the CCD array. The highest physical value in the .out file is matched to the highest pixel value in the .yaml file. A linear scaling across the physical dimension of the image is assumed, whatever offset is present is accounted for, and the physical dimensions given in the .out file are converted into pixel coordinates. At this point, there are four pixel coordinate values for every slit on the plate/strip on the image.

With this information, the angle of rotation for each strip is calculated. For each of the two columns, the midpoint of each strip is calculated. The calculated angle is then the rotation that will move the two midpoints to the same location in the physical direction.

After this, the section of the rotated image to be cropped is calculated. This is the calculation of two values: the upper bound and the lower bound. For the upper bound, the average of the upper bounds for each column is taken. In order to account for light that ends up outside of the boundary of the physical slit, 1.5 is added to this value. Then, the value is rounded off to the nearest integer. This is fudged slightly in the favor of a wider, more inclusive strip. The same procedure is used to derive the lower bound, with 1.5 subtracted from the average (again, to be more inclusive). This is done by the `get_sections` function in `mslit.data` and changes there can fine tune this process.

All calculated values are saved to `./input/name-value.yaml`, where name is the name of the galaxy and value is one of angles, positions, sections, sizes, or types. The image is rotated and then cropped for each srtip using `rotate` and `imcopy`, respectively. Rotated images are saved in `./name/rot` and cropped imaged are saved in `./name/slice`. Images derived from the galaxy have names like `004.fits` and images derived from the comparison lamp have names like `004c.fits`.

Now you need to check how well the rotation and cropping matches up to the actual image. I found it useful to take the calculated section for cropping for each strip and then display the matching rotated image in SAOImage DS9. I would look at whether or not the strip is level across it's whole length and whether or not the strip is contained within the section to be cropped. I found it easiest to adjust the values in the .yaml file until all the strips were level first. This can be done by raising or lowering one column's values (or one corner's values) relative to the other. Next I would raise or lower the all four values uniformly to get the offset right. This process takes many iterations to get right, and is one of the slower stages of data reduction using this codebase.

A.1.5 Third Step: `reduce.py disp`

Before the dispersion correction can be preformed, you need to find the dispersion functions using `identify` and `reidentify`. Do this in IRAF as you normally would, except for making sure to run these functions from the base directory. This is to ensure that the identifications are saved in `./database` and not, e.g., `./ngc3169/database`.

Once you are done with `identify` and `reidentify`, run `reduce.py disp`. This will apply the dispersion functions you found, using `dispcor`, saving the results in `./name/disp`.

A.1.6 Fourth Step: `reduce.py sky`

For sky subtraction, the information in the .out files will be used to determine which spectra are sky spectra. For observation of the calibration stars, however, this will not work. You need to tell `reduce.py` which spectrum contains the information from the calibration star. Do this by adding a 'star_num' field to each group in goups.yaml. For example:

```
- galaxy: ngc3169
  zero: Zero
  flat: Flat1
  star: feige34
  star_num: 10
  mask: Mask.pl
  lamp: henear1

- galaxy: ngc4725
  zero: Zero
  flat: Flat2
  star: pg1708+602
  star_num: 20
  mask: Mask.pl
  lamp: henear2
```

Now run `reduce.py sky`. This will first create a combined sky spectrum for each galaxy; scaling each sky spectrum by the physical width of the slit on the plate with `sarith`, and combining them with `scombine`. Intermediate stages are saved in `./name/sky` and the end result is saved as `./name/sky.1d.fits`. Scaling levels for sky subtraction are kept in a file which may or may not already exist: `./input/name-sky.yaml`. If this file exists already, the levels in there will be used, otherwise a guess at the appropriate level of sky subtraction is made and applied using `sarith`. See the `data.sky` file for details of how the guess is made. The end result is that for each spectrum of an object, the combined sky spectrum is scaled by a certain factor, and then this scaled sky spectrum is subtracted from the object spectrum. Resulting spectra are saved in `./name/sub`.

In order to fine tune the sky subtraction, use the `modify_sky.py` script. Arguments are the path the base directory, the name of the galaxy, the number of the spectrum in question, the operation to preformed, and the amount to change the scaling by. For example `/modify_sky.py ./n3 ngc3169 15 + 0.5` means to increase the scaling factor for the sky subtracted from spectrum 15 of ngc3169 from night three by 0.5. Available operations are limited to + and -. Make modifications to the sky subtraction of each spectrum in this manner until satisfied.

This step will also take care of running `setairmass`. You do not need to worry about it, unless you want something else than the `setairmass` function from the `kpnoslit` package.

A.1.7 Final Step: `reduce.py calibrate`

In order to preform flux calibration for each spectra, you need to generate sensitivity spectra using `standard` and `sensfunc`. Do this in the normal manner, making sure that the sensitivity spectrum for each calibration star is saved as `./name/sens.fits`. Running `reduce.py calibrate` now will use `calibrate` to produce flux calibrated spectra for all H II region spectra, saving the results to `./name/cal`. This is as far as `reduce.py` can take the data, and line measurements can now be made.

A.1.8 Extra Step: `reduce.py analyze`

`Reduce.py` has one more command available, `analyze`. This command produces LaTeX formatted tables and some graphs of data. Data about your galaxies are taken from `splot` logs of line measurements, which must be saved under `./name/measurements`. Data to about H II regions in other galaxies, for comparison purposes, must be stored under `./other_data`. The other data needs to be a file describing the galaxies themselves, `./other_data/key.txt`, and any number of files containing the actual measurements, which can be any filename under `./other_data` starting with 'table'.

The key file consists of multiple lines of tab separated values, which should be the NGC designation of the galaxy (number only), the distance to the galaxy in megaparsecs, the isophotal radius in arcminutes, the Hubble type, bar classification, ring classification, and environmental status. The first line of the file is always discarded, use this for a header line. For example:

```
ngc   D (mpc) r_0  type bar ring env
925   9.4     5.48 Sd   AB  s    group
2541  10.6    3.31 Scd  A   s    group
```

In the table files, whitespace only lines are skipped. A block of data begins with a line that consists of an asterisk followed by the NGC number of the galaxy that the measurements are from. Subsequent lines should be tab separated data: the radius of the H II region as the real radius divided by the isophotal radius, the Hβ flux in units of 10^{-16} ergs cm^{-2} s^{-1}, the OII flux divided by the Hβ flux, and the OIII flux divided by the Hβ flux. Each line describes a new H II region from the galaxy. A new galaxy begins when another line beginning with an asterisk is found. For example:

```
*925
0.65 247.3  2.75  2.85
0.6  191.2  3.6   3.32

*2541
0.52  5.7   2.34  6.78
0.26  60.3  3.64  3.83
```

The ordering and units in both of these files are due to the source of data I used: Zaritsky, Kennicutt, and Huchra (1994) [32].

Once these data are set up, run `reduce.py analyze`. Output is saved under `tables` and consists of encapsulated postscript graphs, and LaTeX formatted tables.

A.2 Source for reduce.py

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

"""
Reduce multi-slit spectroscopic data.

Commands:

zeroflat: combine any zero and flat images for a night
init: initialize a galaxy or star
extract: extract one dimensional spectra from a galaxy or star
disp: apply dispersion correction to a galaxy or star
sky: perform sky subtraction for a galaxy or star
calibrate: flux calibrate a galaxy
analyze: produce graphs and tables of measured data
"""


import argparse
import os
from mslit import analyze, calibrate_galaxy, dispcor_galaxy, get_groups
from mslit import init_galaxy, slice_galaxy, skies, zero_flats


def main(command, path, name):
    """Execute commands from the command line."""
    commands = {'init': init_galaxy, 'extract': slice_galaxy,
                'disp': dispcor_galaxy,
                'sky': skies, 'calibrate': calibrate_galaxy}
    os.chdir(path)
    if command == 'zeroflat':
        zero_flats()
    elif command == 'analyze':
        analyze()
    elif name == 'all':
        groups = get_groups()
```

```
        for group in groups:
            commands[command](group['galaxy'])
            if command != 'calibrate':
                commands[command](group['star'])
    else:
        names = name.split(',')
        for name in names:
            commands[command](name)


def parse_args():
    """Parse the arguments from the command line."""
    parser = argparse.ArgumentParser(
             description='Reduce multi-slit spectroscopic data.',
             formatter_class=argparse.RawDescriptionHelpFormatter,
             epilog="""\
Commands:

zeroflat: combine any zero and flat images for a night
init: initialize a galaxy or star
extract: extract one dimensional spectra from a galaxy or star
disp: apply dispersion correction to a galaxy or star
sky: perform sky subtraction for a galaxy or star
calibrate: flux calibrate a galaxy
analyze: produce graphs and tables of measured data""")
    parser.add_argument('command', help="command to run",
                        choices=['zeroflat', 'init', 'extract', 'disp', 'sky',
                                 'calibrate', 'analyze'])
    parser.add_argument('path', help="path to the set of files")
    parser.add_argument('-n', '--name', default="all",
                        help="name of the galaxy or star to act on (default: "
                             "%(default)s)")
    args = vars(parser.parse_args())
    return (args['command'], args['path'], args['name'])

if __name__ == '__main__':
    main(*parse_args())
```

A.3 Source for modify_sky.py

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

```python
"""
Change sky subtraction levels for a region by an increment.
"""

from argparse import ArgumentParser
from mslit import modify_sky


def parse_args():
    """Parse arguments form the command line."""
    parser = ArgumentParser(description='Change the sky subtraction level for '
                            'a region by an increment.')
    parser.add_argument('path', help="path to the set of files")
    parser.add_argument('name', help='name of the galaxy')
    parser.add_argument('number', help='number of the region to act on',
                        type='int')
    parser.add_argument('op', help='operation to preform', choices=['+', '-'])
    parser.add_argument('value', help='increment', type='float')
    args = vars(parser.parse_args())
    return (args['path'], args['name'], args['number'], args['op'],
            args['value'])

if __name__ == '__main__':
    modify_sky(*parse_args())
```

A.4 Source for mslit.__init__.py

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

"""
Library for reduction and analysis of multi-slit spectroscopic data.
"""

from .analyze import analyze
from .data import get_groups
from .iraf_high import calibrate_galaxy, dispcor_galaxy, init_galaxy
from .iraf_high import slice_galaxy, zero_flats
from .sky import skies, modify_sky


__all__ = ['analyze', 'calibrate_galaxy', 'dispcor_galaxy', 'get_groups',
           'init_galaxy', 'modify_sky', 'skies', 'slice_galaxy', 'zero_flats']
```

A.5 Source for mslit.analyze.py

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

"""
Functions for preforming analysis of measured data.

Primary entry point is the analyze function.

Classes: GalaxyClass, RegionClass

Functions:
Parsing splot logs: get_measurements, get_num, is_labels, is_region_head,
                    parse_line, parse_log
Processing data: collate_lines, id_lines
Parsing data from other_data: process_galaxies, parse_keyfile, get_other
calculating extinction: correct_extinction, extinction_k
calculating metallicity: calculate_OH, calculate_r23, fit_OH
other calculations: calculate_radial_distance, calculate_sfr

"""

from __future__ import with_statement
import math
import os
import os.path
import coords
import numpy
import scipy.optimize
from .const import GROUPS, LINES, LOG_FORMAT
from .data import get, get_groups
from .graphs import compare, compare_basic
from .graphs import graph_metalicity, graph_sfr, graph_sfr_metals
from .misc import avg, cubic_solve, remove_nan
from .tables import make_comparison_table, make_data_table, make_flux_table
from .tables import make_group_comparison_table


def analyze():
    """Run the complete set of data analyzations and output tables and
       graphs."""
    if not os.path.isdir('tables'):
```

```python
        os.mkdir('tables')
    groups = get_groups()
    galaxies = []
    for group in groups:
        data = get(group['galaxy'], 'key')
        galaxies.append(GalaxyClass(group['galaxy'], data))
    for galaxy in galaxies:
        galaxy.run()
        galaxy.fit_OH()
        galaxy.output()
    other_data = get_other()
    for galaxy in other_data:
        galaxy.fit_OH()
    compare_basic(galaxies, other_data)
    make_comparison_table(galaxies, other_data)
    make_group_comparison_table(galaxies, other_data)
    for key, group in GROUPS.items():
        compare(galaxies, other_data, group, key)


## Useful classes ##


class GalaxyClass:
    """A class for information related to a galaxy of multiple regions."""

    def __init__(self, name, data):
        self.name = name      # basic name, for id purposes
        self.regions = []     # list of all the regions, empty for now
        self.fit = None      # least square fitting solution for metallicity
        self.grad = None      # fitted O/H metallicity gradient
        self.metal = None       # standard metallicity of galaxy
        self.region_number = None       # number of regions in galaxy
        self.print_name = data['name']       # printable name for galaxy
        self.distance = data['distance'] # distance to galaxy in kpc
        self.r25 = data['r25']  # r_25 scale measure, in kpc
        self.type = data['type']     # Sa to Sd hubble type
        self.bar = data['bar']       # bar type
        self.ring = data['ring']     # ring type
        self.env = data['env']       # environment: group, pair, isolated
        if 'redshift' in data:
            self.redshift = data['redshift']  # redshift factor for galaxy
```

```
    if 'center' in data:
        self.center = data['center']    # Galactic center as RA, DEC string

def fit_OH(self):
    """Find a linear fit of O/H metallicity in galaxy using least
       squares fitting."""
    self.fit = fit_OH(self.regions, self.r25)
    self.grad = self.fit[0][0]
    # standard metallicity is the metallicity at r = 0.4
    self.metal = self.fit[0][1] + self.grad * 0.4

def output(self):
    """Produce table and graph output."""
    graph_metalicity(self)
    graph_sfr(self)
    graph_sfr_metals(self)
    # remove regions with no data
    # go backwards so that indexing is kept as items are deleted
    for region in self.regions[::-1]:
        if numpy.isnan(region.fluxes['halpha']):
            del self.regions[region.number]
    for count, region in enumerate(self.regions):
        region.printnumber = count + 1
    make_flux_table(self)
    make_data_table(self)

def run(self):
    """Setup a galaxy from splot measurements."""
    measurements = get_measurements(self.name)
    self.region_number = len([r for r in measurements if r != []])
    lines = LINES.copy()
    for key, value in lines.items():
        lines.update({key: (value * (self.redshift + 1))})
    for group in measurements:
        id_lines(group, lines)
    groups = [collate_lines(region) for region in measurements]
    data = get(self.name, 'positions')
    for i, (fluxes, centers) in enumerate(groups):
        region = RegionClass(i, fluxes, centers)
        region.distance = self.distance
        region.position = '%s %s' % (data[i]['ra'], data[i]['dec'])
        region.center = self.center
```

```python
            region.calculate()
            self.regions.append(region)


class RegionClass:
    """A class for data related to individual regions."""

    def __init__(self, number, fluxes, centers=None):
        self.number = number     # number of the region
        self.fluxes = fluxes     # list of averaged flux measurements
        self.centers = centers   # list of averaged wavelength centers
        self.corrected = False  # has extinction correction been applied?
        self.distance = None    # distance to galaxy
        self.center = None      # RA, Dec of galactic center
        self.OH = None      # O/H metallicity
        self.SFR = None        # Star formation rate
        self.rdistance = None   # galactocentric radius
        self.position = None        # RA, Dec of the region
        self.r23 = None     # r23 metallicity

    def calculate(self):
        """Perform astrophysical calculations related to the region."""
        self.correct_extinction()
        self.rdistance = calculate_radial_distance(self.position, self.center,
                                                   self.distance)
        self.r23 = calculate_r23(self.fluxes)
        self.calculate_OH()
        self.SFR = calculate_sfr(self.distance, self.fluxes['halpha'])

    def calculate_OH(self, disambig=True):
        """Calculate O/H metallicity for a region, optionally checking which
           branch of the solution it is on."""
        if disambig:
            branch = self.fluxes['OIII2'] / self.fluxes['OII']
            self.OH = calculate_OH(self.r23, branch)
        self.OH = calculate_OH(self.r23)

    def correct_extinction(self):
        """If possible, correct for all the regions flux measurements for
           extinction."""
        R_obv = self.fluxes['halpha'] / self.fluxes['hbeta']
        if not numpy.isnan(R_obv):
```

```
            self.fluxes = correct_extinction(R_obv, self.fluxes, self.centers)
            self.corrected = True


## Log parsing functions ##


def get_measurements(name):
    """For a given galaxy, return a list of all of the measurements taken."""
    fns = os.listdir('%s/measurements/' % name)
    measurements = []
    for fn in fns:
        if fn[-4:] == '.log':
            with open('%s/measurements/%s' % (name, fn)) as f:
                measurements.append(parse_log(f.readlines()))
    collated = measurements[0][:]
    for log in measurements[1:]:
        for i, region in enumerate(log):
            collated[i].extend(region)
    return collated


def get_num(line):
    """Return the number of the region, given it's header line."""
    start = line.find('[') + 1
    return int(line[start:start + 3])


def is_labels(line):
    """Return true if the line from the log is the field labeling line."""
    labelstr = ("    center      cont       flux        eqw       core      gfwhm"
                "     lfwhm\n")
    if line == labelstr:
        return True
    return False


def is_region_head(line):
    """Return true if a line is a region header line. Test this by testing
       if it begins with the abbreviation for a month."""
    if line[:3] in ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
                    'Sep', 'Oct', 'Nov', 'Dec'):
```

```
            return True
    return False


def parse_line(line):
    """Return a dictionary of the values from a line of the log."""
    values = [float(x) if x != 'INDEF' else float('nan') for x in line.split()]
    return dict(zip(LOG_FORMAT, values))


def parse_log(raw):
    """Parse raw lines from a splot log file and return a list of all the
       measurements found."""
    length = max([get_num(line) for line in raw if is_region_head(line)])
    current = None
    measurements = [[] for i in range(length + 1)]
    for line in raw:
        if is_region_head(line):
            current = get_num(line)
        elif line.strip() != '' and not is_labels(line):
            measurements[current].append(parse_line(line))
    return measurements


## Processing ##


def collate_lines(region):
    """Given all the measurements for a region, return the averaged fluxes of
       each spectral line that we are interested in. Also return the average
       wavelength that the line was found at."""
    # This could be easily modified if you wanted to get the averages of other
    # values that splot provides, or their standard deviations, etc.
    fluxes = {}
    centers = {}
    for name in LINES:
        sources = [measurement for measurement in region
                   if measurement['name'] == name]
        line = {}
        for item in LOG_FORMAT:
            line.update({item: avg(*[s[item] for s in sources])})
        fluxes.update({name: line['flux']})
```

```python
        centers.update({name: line['center']})
    return fluxes, centers


def id_lines(region, lines):
    """For all the measurements in a region, determine which spectral line
       they are closest to in wavelength."""
    for measurement in region:
        badness = dict([(abs(measurement['center'] - lines[name]), name)
                        for name in lines])
        measurement.update({'name': badness[min(badness.keys())]})


## Functions for reading in tables of data ##


def process_galaxies(fn, galaxydict):
    """Read data from a file in other_data, and add that data to the
       appropriate galaxy in galaxydict."""
    with open('other_data/%s' % fn) as f:
        raw = [line.strip() for line in f.readlines()]
    current = None
    number = None
    for line in raw:
        if line == '':
            pass
        elif line[0] == '*':
            current = galaxydict[line[1:]]
            number = 0
        else:
            (r, hbeta, r2, r3) = [float(item) for item in line.split('\t')]
            data = {'hbeta': hbeta, 'OII': r2 * hbeta, 'OIII1': r3 * hbeta}
            region = RegionClass(number, data)
            region.r23 = r2 + r3
            region.rdistance = current.r25 * r
            region.distance = current.distance
            region.calculate_OH(disambig=False)
            current.regions.append(region)
            number += 1


def parse_keyfile():
```

```
    """Return a dictionary of the galaxies described in other_data/key.txt."""
    with open('other_data/key.txt') as f:
        raw = f.readlines()
    del raw[0]
    galaxydict = {}
    for line in raw:
        line = line.strip()
        (ngc, distance, r_0, htype, bar, ring, env) = line.split('\t')
        # convert distance to kpc from Mpc for consistency
        distance = float(distance) * 1000
        # convert r_0 from arcminutes to kpc
        r_0 = distance * math.tan(math.radians(float(r_0) * 60))
        data = {'distance': distance, 'r25': r_0, 'type': htype,
                'bar': bar, 'ring': ring, 'env': env,
                'name': 'NGC %s' % ngc}
        galaxydict.update({ngc: GalaxyClass(ngc, data)})
    return galaxydict


def get_other():
    """Return a list of galaxy objects, one for each galaxy described in
       the other_data directory."""
    files = os.listdir('other_data/')
    files = [fn for fn in files if fn.startswith('table')]
    galaxydict = parse_keyfile()
    for fn in files:
        process_galaxies(fn, galaxydict)
    for galaxy in galaxydict.values():
        galaxy.region_number = len(galaxy.regions)
    return galaxydict.values()


## Calculating extinction ##


def correct_extinction(R_obv, fluxes, centers):
    """Given an halpha/hbeta ratio, and a list of fluxes and their wavelength
       locations, return extinction corrected fluxes. Uses the Calzetti
       method."""
    # using the method described here:
# <http://www.astro.umd.edu/~chris/publications/html_papers/aat/node13.html>
    R_intr = 2.76
```

```
    a = 2.21
    extinction = a * math.log10(R_obv / R_intr)
    # Now using the Calzetti method:
    values = {}
    for name, flux in fluxes.items():
        flux = flux / (10 ** (-0.4 * extinction *
                              extinction_k(centers[name])))
        values.update({name: flux})
    return values


def extinction_k(l):
    """Inner 'k' function for use in the Calzetti method of extinction
       correction."""
    # for use in the calzetti method
    # convert to micrometers from angstrom
    l = l / 10000.
    if 0.63 <= l <= 1.0:
        return ((1.86 / l ** 2) - (0.48 / l ** 3) -
            (0.1 / l) + 1.73)
    elif 0.12 <= l < 0.63:
        return (2.656 * (-2.156 + (1.509 / l) -
            (0.198 / l ** 2) + (0.011 / l ** 3)) + 4.88)
    else:
        return float('nan')


## Calculating metallicity ##


def calculate_OH(r23, branch=None):
    """Convert r_23 to O/H metallicity, using conversion given by Nagao
       2006."""
    b0 = 1.2299 - math.log10(r23)
    b1 = -4.1926
    b2 = 1.0246
    b3 = -6.3169 * 10 ** -2
    # solving the equation
    solutions = cubic_solve(b0, b1, b2, b3)
    solutions = [x.real if x.imag == 0.0 else float('nan') for x in solutions]
    if branch is not None:
        # if given, branch should be the ratio OIII2 / OII
```

```python
        if branch < 2:
            return solutions[2]
        return solutions[1]
    return solutions[2]


def calculate_r23(fluxes):
    """Return the calucated R_23 metallicity, given a set of fluxes."""
    r2 = fluxes['OII'] / fluxes['hbeta']
    r3 = (fluxes['OIII1'] + fluxes['OIII2']) / fluxes['hbeta']
    r23 = r2 + r3
    return r23


def fit_OH(regions, r25):
    """Use least squares method to find a linear fit for O/H metallicity over
       a set of regions."""
    # inital guess: flat and solar metallicity
    slope = 0
    intercept = 8.6
    x = [s.rdistance for s in regions]
    y = [s.OH for s in regions]
    remove_nan(x, y)
    x = numpy.array(x)
    y = numpy.array(y)
    x = x / r25

    def f((slope, intercept)):
        """Function to be fit by scipy.optimize.leastsq."""
        return y - (intercept + slope * x)

    return scipy.optimize.leastsq(f, (slope, intercept))


## Other calculations ##

def calculate_radial_distance(position1, position2, distance):
    """Calculate the distance between two sky locations at the same distance
       from earth."""
    position = coords.Position(position1)
    theta = coords.Position(position2).angsep(position)
    # radial distance returned in whatever units distance is in
```

```python
    return distance * math.tan(math.radians(theta.degrees()))


def calculate_sfr(distance, halpha_flux):
    """Calculate star formation rate from H_alpha flux, using the calibration
       given by Kennicutt 1998."""
    d = distance * 3.0857 * (10 ** 21)
    luminosity = halpha_flux * 4 * math.pi * (d ** 2)
    return luminosity * 7.9 * (10 ** -42)
```

A.6 Source for mslit.const.py

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

"""
Some useful values.

This file contains some useful values, all used during analyzing data.
"""

LOG_FORMAT = ('center', 'cont', 'flux', 'eqw', 'core', 'gfwhm', 'lfwhm')

LINES = {'OII': 3727, 'hgamma': 4341, 'hbeta': 4861, 'OIII1': 4959,
         'OIII2': 5007, 'NII1': 6548, 'halpha': 6563, 'NII2': 6583,
         'SII1': 6717, 'SII2': 6731, 'OIII3': 4363}

LOOKUP = {'OII': '[O II]$\lambda3727$', 'hgamma': 'H$\gamma$',
          'hbeta': 'H$\\beta$', 'OIII1': '[O III]$\lambda4959$',
          'OIII2': '[O III]$\lambda5007$', 'NII1': '[N II]$\lambda6548$',
          'halpha': 'H$\\alpha$', 'NII2': '[N II]$\lambda6583$',
          'SII1': '[S II]$\lambda6717$', 'SII2': '[S II]$\lambda6731$',
          'OIII3': '[O III]$\lambda4363$',
          'OH': '$12 + \log{\\textnormal{O/H}}$',
          'SFR': 'SFR(M$_\odot$ / year)', 'rdistance': 'Radial Distance (kpc)',
          'extinction': 'E(B - V)', 'r23': '$R_{23}$'}

GROUPS = {'env': [{'name': 'Isolated', 'members': ('isolated',)},
                  {'name': 'Group', 'members': ('group',)},
                  {'name': 'Pair', 'members': ('pair',)}],
          'ring': [{'name': 'S-Shaped', 'members': ('s',)},
                   {'name': 'Intermediate Type', 'members': ('rs',)},
                   {'name': 'Ringed', 'members': ('r',)}],
```

```
        'bar': [{'name': 'No Bar', 'members': ('A',)},
                {'name': 'Weakly Barred', 'members': ('AB',)},
                {'name': 'Strongly Barred', 'members': ('B',)}],
        'type': [{'name': 'Sa and Sab', 'members': ('Sa', 'Sab')},
                 {'name': 'Sb and Sbc', 'members': ('Sb', 'Sbc')},
                 {'name': 'Sc and Scd', 'members': ('Sc', 'Scd')},
                 {'name': 'Sd', 'members': ('Sd', 'Irr')}]}
```

A.7 Source for mslit.data.py

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

"""
Functions for working with metadata about the observations.

low level: get, get_groups, get_mslit_data, write
manipulation functions: get_group, get_object_spectra, get_sky_spectra
                         init_data
calculation functions: calculate_angles, calculate_sections,
                        calculate_pixel_coordinates
"""
from __future__ import with_statement
import math
import os.path
import yaml
from .misc import avg, threshold_round

## Functions for low level reading and writing ##


def get(name, suffix):
    """Get the contents of a previously saved metadata file."""
    fn = 'input/%s-%s.yaml' % (name, suffix)
    with open(fn) as f:
        return yaml.load(f)


def get_groups():
    """Get the contents of the groups file."""
    fn = 'input/groups.yaml'
    with open(fn) as f:
        return yaml.load(f.read())
```

```python
def get_mslit_data(name):
    """Get the important data from MSLIT's output."""
    fn = 'input/%s.out' % name
    with open(fn) as f:
        raw = f.readlines()
    # 0 is headers, 1 is a blank line, 2+ is data
    # if a header field begins with a (, it's not really a header field
    headers = [h for h in raw[0].split() if h[0] != '(']
    data = []
    for line in raw[2:]:
        pairs = dict(zip(headers, line.split()))
        data.append({'type': pairs['NAME'], 'xlo': pairs['XLO'],
                     'xhi': pairs['XHI'],
                     'pos': {'ra': pairs['RA'], 'dec': pairs['DEC']}})
    return data


def write(name, suffix, data):
    """Write some metadata to disk."""
    fn = 'input/%s-%s.yaml' % (name, suffix)
    with open(fn, 'w') as f:
        f.write(yaml.dump(data))


## Functions for basic manipulation ##

def get_group(name):
    """Return the group data for a given galaxy or star."""
    groups = get_groups()
    for group in groups:
        if name in group.values():
            return group


def get_object_spectra(name):
    """Return the indexes of the spectra that contain objects."""
    group = get_group(name)
    if name == group['star']:
        return [group['star_num']]
    else:
```

```python
        items = get(name, 'types')
        return [i for i, x in enumerate(items) if x == 'HIIREGION']


def get_sky_spectra(name):
    """Return the indexes of the spectra that contain sky."""
    group = get_group(name)
    sky_types = ['NIGHTSKY']
    items = get(name, 'types')
    # we can assume that slits marked HIIREGION for calibration star
    # observations also contain sky, but make sure not to include the star.
    if name == group['star']:
        items.pop(group['star_num'])
        sky_types.append('HIIREGION')
    sky_list = [i for i, x in enumerate(items) if x in sky_types]
    return sky_list


def init_data(name):
    """Generate extra data files from name.out and name-pixel.yaml."""
    group = get_group(name)
    use = group['galaxy']
    data = get_mslit_data(use)
    pixel_data = get(use, 'pixel')
    real_sizes = [(float(i['xlo']), float(i['xhi'])) for i in data]
    types = [item['type'] for item in data]
    coord = calculate_pixel_coordinates(pixel_data, real_sizes)
    angles = calculate_angles(coord)
    sections, sizes = calculate_sections(coord)
    write(name, 'angles', angles)
    write(name, 'sections', sections)
    write(name, 'sizes', sizes)
    write(name, 'types', types)
    write(name, 'positions', [item['pos'] for item in data])
    if not os.path.isfile('input/%s-sky.yaml' % name):
        write(name, 'sky', [None] * len(types))


## Functions for calculations ##


def calculate_angles(data):
```

```python
    """Calculate the angles described by a set of pixel coordinates."""
    (column1, column2) = data.keys()
    run = column1 - column2
    angles = []
    for item1, item2 in zip(*data.values()):
        mid1 = avg(item1['start'], item1['end'])
        mid2 = avg(item2['start'], item2['end'])
        rise = mid1 - mid2
        slope = rise / run
        angles.append(math.degrees(math.atan(slope)))
    return angles


def calculate_sections(data):
    """Calculate slicing sections for a set of pixel coordinates."""
    sections = []
    size = []
    fudge_factor = 1.5
    rounding_threshold = 0.70
    for (item1, item2) in zip(*data.values()):
        start = avg(item1['start'], item2['start']) - fudge_factor
        end = avg(item1['end'], item2['end']) + fudge_factor
        start = threshold_round(start, rounding_threshold)
        end = threshold_round(end, 1 - rounding_threshold)
        size.append(end - start)
        sections.append('[1:2048,%s:%s]' % (start, end))
    return sections, size


def calculate_pixel_coordinates(pixel_data, real_sizes):
    """Covert physical coordinates from MSLIT .out files into pixel
       coordinates."""
    real_start = real_sizes[0][0]
    real_end = real_sizes[-1][1]
    real_size = real_end - real_start
    coord = {}
    for pixel in pixel_data:
        pixel_size = pixel['end'] - pixel['start']
        ratio = pixel_size / real_size
        values = []
        for item in real_sizes:
            start = ratio * (item[0] - real_start) + pixel['start']
```

```
            end = ratio * (item[1] - real_start) + pixel['start']
            values.append({'start': start, 'end': end})
        coord.update({pixel['column']: values})
    return coord
```

A.8 Source for mslit.graphs.py

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

"""
Functions for oupting graphs.

Generic plotting functions: add_solar_metallicity, plot
Single galaxy graphs: graph_metallicity, graph_sfr, graph_sfr_metals
Mutiple galaxy graphs: compare, compare_basic
"""

import matplotlib
from matplotlib.backends.backend_ps import FigureCanvasPS as FigureCanvas
import numpy
from .misc import remove_nan


matplotlib.rc('text', usetex=True)
matplotlib.rc('font', family='serif', serif='Computer Modern Roman')


## Generic plotting functions ##


def add_solar_metallicity(axes):
    """Overplot solar metallicity on a given set of axes."""
    xbound = axes.get_xbound()
    t = numpy.arange(0, xbound[1] * (4 / 3.), xbound[1] * 0.05)
    solardata = 8.69 + t * 0
    axes.plot(t, solardata, 'k--')
    axes.text(xbound[1] * (301 / 300.), 8.665, r'$Z_\odot$',
              transform=axes.transData)


def plot(galaxy_sets, axes, colors, xkey, ykey, only_corrected=False):
    """Make a plot.
```

```
        galaxy_sets: iterable containing sets of galaxies
        axes: axes to plot on
        colors: matplotlib color codes, matched to the sets in galaxy_sets
        xkey: x values will be region.xkey for every region in a galaxy
        ykey: y values will be region.ykey for every region in a galaxy
        only_corrected: If set to true, only plot regions with extinction
                        correction applied. Defaults to false."""
    for group, color in zip(galaxy_sets, colors):
        for galaxy in group:
            regions = galaxy.regions
            if only_corrected:
                regions = [s for s in regions if s.corrected]
            x = [r.__dict__[xkey] for r in regions]
            y = [r.__dict__[ykey] for r in regions]
            remove_nan(x, y)
            if xkey is 'rdistance':
                x = numpy.array(x)
                x = x / galaxy.r25
            axes.plot(x, y, color)


## Single Galaxy Plots ##


def graph_metalicity(galaxy):
    """Graph the O/H metallicity of a galaxy versus R/R_25 distance. Include
        a linear fitted function."""
    # setup the graph
    fig = matplotlib.figure.Figure(figsize=(5, 5))
    canvas = FigureCanvas(fig)
    axes = fig.add_axes((.125, .1, .775, .8))
    axes.set_xlabel(r'$R/R_{25}$')
    axes.set_ylabel(r'$12 + \log{\textnormal{O/H}}$')
    axes.set_autoscale_on(False)
    axes.set_xbound(lower=0, upper=1.5)
    axes.set_ybound(lower=8.0, upper=9.7)
    # plot the data
    plot(((galaxy,),), axes, ('co',), 'rdistance', 'OH')
    # overplot the fitted function
    t = numpy.arange(0, 2, .1)
    fit = galaxy.fit[0]
```

```
    fitdata = fit[1] + t * fit[0]
    axes.plot(t, fitdata, 'k-')
    #overplot solar metalicity
    add_solar_metallicity(axes)
    canvas.print_eps('tables/%s_metals.eps' % galaxy.name)


def graph_sfr(galaxy):
    """Graph the star formation rate within a galaxy versus the R/R_25
       distance."""
    # set up the graph
    fig = matplotlib.figure.Figure(figsize=(5, 5))
    canvas = FigureCanvas(fig)
    axes = fig.add_axes((.125, .1, .775, .8))
    axes.set_xlabel(r'$R/R_{25}$')
    axes.set_ylabel(r'SFR(M$_\odot$ / year)')
    axes.set_autoscalex_on(False)
    axes.set_xbound(lower=0, upper=1.5)
    # plot the data
    plot(((galaxy,),), axes, ('co',), 'rdistance', 'SFR')
    canvas.print_eps('tables/%s_sfr.eps' % galaxy.name)


def graph_sfr_metals(galaxy):
    """Graph metallicity versus star formation rate, excluding regions which
       couldn't be extinction corrected."""
    fig = matplotlib.figure.Figure(figsize=(5, 5))
    canvas = FigureCanvas(fig)
    axes = fig.add_axes((.125, .1, .775, .8))
    axes.set_xlabel(r'SFR(M$_\odot$ / year)')
    axes.set_ylabel(r'$12 + \log{\textnormal{O/H}}$')
    plot(((galaxy,),), axes, ('co',), 'SFR', 'OH', only_corrected=True)
    axes.set_xbound(lower=0)
    axes.set_ybound(lower=8.0, upper=9.7)
    axes.set_autoscale_on(False)
    add_solar_metallicity(axes)
    canvas.print_eps('tables/%s_sfr-metal.eps' % galaxy.name)


## Mutliple Galaxy Plots ##
```

```python
def compare(galaxies, other, groups, key):
    """Make a plot comparing many galaxies, color coding by groups.

       galaxies: list my galaxies
       other: list of other galaxies
       groups: as from mslit.const.GROUPS
       key: function will check galaxy.key for all galaxies"""
    colors = ['y^', 'rd', 'mp', 'bD']
    fig = matplotlib.figure.Figure(figsize=(5, 5))
    canvas = FigureCanvas(fig)
    axes = fig.add_axes((.125, .1, .775, .8))
    axes.set_xlabel(r'$R/R_{25}$')
    axes.set_ylabel(r'$12 + \log{\textnormal{O/H}}$')
    #overplot solar metalicity
    t = numpy.arange(0, 2, .1)
    solardata = 8.69 + t * 0
    axes.plot(t, solardata, 'k--')
    axes.text(1.505, 8.665, r'$Z_\odot$')
    # plot the data
    galaxies += other
    galaxy_sets = [[galaxy for galaxy in galaxies
                    if galaxy.__dict__[key] in group['members']]
                   for group in groups]
    plot(galaxy_sets, axes, colors, 'rdistance', 'OH')
    axes.set_xbound(lower=0, upper=1.5)
    axes.set_ybound(lower=8.0, upper=9.7)
    canvas.print_eps('tables/%s_comparison.eps' % key)


def compare_basic(galaxies, other):
    """Print metallicity versus galactocentric radius for many galaxies."""
    fig = matplotlib.figure.Figure(figsize=(5, 5))
    canvas = FigureCanvas(fig)
    axes = fig.add_axes((.125, .1, .775, .8))
    axes.set_xlabel(r'$R/R_{25}$')
    axes.set_ylabel(r'$12 + \log{\textnormal{O/H}}$')
    #overplot solar metalicity
    t = numpy.arange(0, 2, .1)
    solardata = 8.69 + t * 0
    axes.plot(t, solardata, 'k--')
    axes.text(1.505, 8.665, r'$Z_\odot$', transform=axes.transData)
    # mine in color, the other data as black diamonds
```

```
    plot((other, (galaxies[0],), (galaxies[1],)), axes, ('wd', 'r^', 'co'),
         'rdistance', 'OH')
    axes.set_xbound(lower=0, upper=1.5)
    axes.set_ybound(lower=8.0, upper=9.7)
    canvas.print_eps('tables/basic_comparison.eps')
```

A.9 Source for mslit.iraf_high.py

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

"""
High level wrappers around IRAF functions.

wrappers: apsum_galaxy, calibrate_galaxy, dispcor_galaxy, fix_galaxy
          imcopy_galaxy, init_galaxy, rotate_galaxy, setairmass_galaxy
          slice_galaxy, zero_flats
"""

import os
import os.path
from .data import get, get_group, get_groups, get_object_spectra
from .data import get_sky_spectra, init_data
from .iraf_low import apsum, calibrate, dispcor, hedit, imcopy, fixpix, ccdproc
from .iraf_low import rotate, combine, zerocombine, flatcombine
from .misc import list_convert, namefix, zerocount


## Higher level IRAF wrappers ##

def apsum_galaxy(name):
    """Create one dimensional spectra for a galaxy."""
    sections = get(name, 'sections')
    if not os.path.isdir('%s/sum' % name):
        os.mkdir('%s/sum' % name)
    for i, section in enumerate(sections):
        num = zerocount(i)
        apsum('%s/slice/%s' % (name, num),
              '%s/sum/%s.1d' % (name, num), section)
        apsum('%s/slice/%sc' % (name, num),
              '%s/sum/%sc.1d' % (name, num), section)
        namefix('%s/sum/%s.1d' % (name, num))
        namefix('%s/sum/%sc.1d' % (name, num))
```

```python
def calibrate_galaxy(name):
    """Flux calibrate all object spectra in a galaxy."""
    group = get_group(name)
    if not os.path.isdir('%s/cal' % name):
        os.mkdir('%s/cal' % name)
    sens = '%s/sens' % group['star']
    spectra = get_object_spectra(name)
    for spectrum in spectra:
        num = zerocount(spectrum)
        calibrate('%s/sub/%s.1d' % (name, num), sens,
            '%s/cal/%s.1d' % (name, num))


def dispcor_galaxy(name):
    """Apply dispersion correction to all spectra in a galaxy."""
    group = get_group(name)
    use = group['galaxy']
    if not os.path.isdir('%s/disp' % name):
        os.mkdir('%s/disp' % name)
    spectra = set(get_object_spectra(name) + get_sky_spectra(name))
    for spectrum in spectra:
        num = zerocount(spectrum)
        hedit('%s/sum/%s.1d' % (name, num), 'REFSPEC1',
            '%s/sum/%sc.1d' % (use, num))
        dispcor('%s/sum/%s.1d' % (name, num),
            '%s/disp/%s.1d' % (name, num))


def fix_galaxy(name):
    """Apply a bad pixel mask to all images in a galaxy."""
    group = get_group(name)
    imcopy('@lists/%s' % name, '%s/' % name)
    with open('lists/%s' % name) as f:
        items = ['%s/%s' % (name, item.strip()) for item in f.readlines()]
    strlist = list_convert(items)
    hedit(strlist, 'BPM', group['mask'])
    fixpix(strlist, 'BPM')


def imcopy_galaxy(name):
```

```
    """Create cropped images for all sections in a galaxy."""
    if not os.path.isdir('%s/slice' % name):
        os.mkdir('%s/slice' % name)
    sections = get(name, 'sections')
    for i, section in enumerate(sections):
        num = zerocount(i)
        imcopy('%s/rot/%s%s' % (name, num, section),
            '%s/slice/%s' % (name, num))
        imcopy('%s/rot/%sc%s' % (name, num, section),
            '%s/slice/%sc' % (name, num))


def init_galaxy(name):
    """Apply a bad pixel mask, then run ccdproc and combine."""
    group = get_group(name)
    if not os.path.isdir(name):
        os.mkdir(name)
    fix_galaxy(name)
    with open('lists/%s' % name) as f:
        items = ['%s/%s' % (name, item.strip()) for item in f.readlines()]
    ccdproc(list_convert(items), zero=group['zero'], flat=group['flat'])
    combine(list_convert(items), '%s/base' % name)


def rotate_galaxy(name):
    """Create a rotated image for every spectra in a galaxy."""
    group = get_group(name)
    if not os.path.isdir('%s/rot' % name):
        os.mkdir('%s/rot' % name)
    angles = get(name, 'angles')
    for i, angle in enumerate(angles):
        num = zerocount(i)
        rotate('%s/base' % name, '%s/rot/%s' % (name, num), angle)
        rotate('@lists/%s' % group['lamp'], '%s/rot/%sc' % (name, num), angle)


def slice_galaxy(name):
    """Create one dimensional spectra for a galaxy."""
    init_data(name)
    rotate_galaxy(name)
    imcopy_galaxy(name)
    apsum_galaxy(name)
```

```python
    # needed for next step
    try:
        os.makedirs('database/id%s/sum' % name)
    except OSError:
        pass


def zero_flats():
    """Combine the zeros and flats for a night, then apply a bad pixel mask."""
    groups = get_groups()
    done = []
    for group in groups:
        if group['zero'] not in done:
            done.append(group['zero'])
            zerocombine('@lists/%s' % group['zero'], output=group['zero'])
            hedit(group['zero'], 'BPM', group['mask'])
            fixpix(group['zero'], 'BPM')
        if group['flat'] not in done:
            done.append(group['flat'])
            flatcombine('@lists/%s' % group['flat'], output=group['flat'])
            hedit(group['flat'], 'BPM', group['mask'])
            fixpix(group['flat'], 'BPM')
```

A.10 Source for mslit.iraf_low.py

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

"""
Low level wrappers around IRAF functions.

loaders: load_apextract, load_ccdred, load_imgeom, load_kpnoslit,
         load_onedspec
wrappers: apsum, calibrate, ccdproc, combine, dispcor, flatcombine, fixpix,
          hedit, imcopy, rotate, sarith, scombine, setairmass, zerocombine
misc: set_aperture

All function wrappers can be passed arbitrary key values which will be
passed on to the corresponding IRAF function.
"""

from __future__ import with_statement
import os
```

```python
import os.path
import pyraf.iraf


## Wrappers for loading IRAF packages ##


def load_apextract():
    """Load the apextract package."""
    pyraf.iraf.noao(_doprint=0)
    pyraf.iraf.twodspec(_doprint=0)
    pyraf.iraf.apextract(_doprint=0)


def load_ccdred():
    """Load the ccdred package."""
    pyraf.iraf.noao(_doprint=0)
    pyraf.iraf.imred(_doprint=0)
    pyraf.iraf.ccdred(_doprint=0)


def load_imgeom():
    """Load the imgeom package."""
    pyraf.iraf.images(_doprint=0)
    pyraf.iraf.imgeom(_doprint=0)


def load_kpnoslit():
    """Load the kpnoslit package."""
    pyraf.iraf.imred(_doprint=0)
    pyraf.iraf.kpnoslit(_doprint=0)


def load_onedspec():
    """Load the onedspec package."""
    pyraf.iraf.noao(_doprint=0)
    pyraf.iraf.onedspec(_doprint=0)


## Wrappers around IRAF functions ##
```

```python
def apsum(infiles, outfiles, section, **kwargs):
    """Call the apsum function from the apextract package, creating the
       apeture automatically and setting some defaults appropriately."""
    load_apextract()
    set_aperture(infiles, section)
    kwargs.setdefault('format', 'onedspec')
    kwargs.setdefault('interactive', 'no')
    kwargs.setdefault('find', 'no')
    kwargs.setdefault('trace', 'no')
    kwargs.setdefault('fittrace', 'no')
    pyraf.iraf.apsum.unlearn()
    pyraf.iraf.apsum(input=infiles, output=outfiles, **kwargs)


def calibrate(infiles, sens, outfiles, **kwargs):
    """Call the calibrate function from the kpnoslit package."""
    load_kpnoslit()
    pyraf.iraf.calibrate.unlearn()
    pyraf.iraf.calibrate(input=infiles, output=outfiles, sens=sens, **kwargs)


def ccdproc(images, **kwargs):
    """Call the ccdproc function from the ccdred package,
       with some defaults set appropriately."""
    load_ccdred()
    kwargs.setdefault('darkcor', 'no')
    kwargs.setdefault('fixpix', 'no')
    kwargs.setdefault('biassec', '[2049:2080,1:501]')
    kwargs.setdefault('trimsec', '[1:2048,1:501]')
    pyraf.iraf.ccdproc.unlearn()
    pyraf.iraf.ccdproc(images=images, **kwargs)


def combine(infiles, outfiles, **kwargs):
    """Call the combine function from the ccdred package."""
    load_ccdred()
    pyraf.iraf.combine.unlearn()
    pyraf.iraf.combine(input=infiles, output=outfiles, **kwargs)


def dispcor(infiles, outfiles, **kwargs):
    """Call the dispcor function from the onedspec package."""
```

```python
    load_onedspec()
    pyraf.iraf.dispcor.unlearn()
    pyraf.iraf.dispcor(input=infiles, output=outfiles, **kwargs)


def flatcombine(infiles, **kwargs):
    """Call the flatcombine function from the ccdred package."""
    load_ccdred()
    kwargs.setdefault('process', 'no')
    pyraf.iraf.flatcombine.unlearn()
    pyraf.iraf.flatcombine(input=infiles, **kwargs)


def fixpix(image, mask, **kwargs):
    """Call the fixpix function from the core IRAF package."""
    pyraf.iraf.fixpix.unlearn()
    pyraf.iraf.fixpix(images=image, masks=mask, **kwargs)


def hedit(images, fields, value, **kwargs):
    """Add a field to a file's header using the hedit function."""
    kwargs.setdefault('add', 'yes')
    kwargs.setdefault('verify', 'no')
    pyraf.iraf.hedit.unlearn()
    pyraf.iraf.hedit(images=images, fields=fields, value=value, **kwargs)


def imcopy(infiles, outfiles, **kwargs):
    """Call the imcopy function from the core IRAF package."""
    pyraf.iraf.imcopy.unlearn()
    pyraf.iraf.imcopy(input=infiles, output=outfiles, **kwargs)


def rotate(infiles, outfiles, angle, **kwargs):
    """Call the rotate function from the imgeom package."""
    load_imgeom()
    pyraf.iraf.rotate.unlearn()
    pyraf.iraf.rotate(input=infiles, output=outfiles, rotation=-angle,
                      **kwargs)


def sarith(infile1, op, infile2, outfile, **kwargs):
```

```python
    """Call the sarith function from the onedspec package."""
    load_onedspec()
    pyraf.iraf.sarith.unlearn()
    pyraf.iraf.sarith(input1=infile1, op=op, input2=infile2, output=outfile,
        **kwargs)


def scombine(infiles, outfiles, **kwargs):
    """Call the scombine function from the onedspec package."""
    load_onedspec()
    pyraf.iraf.scombine.unlearn()
    pyraf.iraf.scombine(input=infiles, output=outfiles, **kwargs)


def setairmass(images, **kwargs):
    """Call the setairmass function from the kpnoslit package."""
    load_kpnoslit()
    pyraf.iraf.setairmass.unlearn()
    pyraf.iraf.setairmass(images=images, **kwargs)


def zerocombine(infiles, **kwargs):
    """Call the zerocombine function from the ccdred package."""
    load_ccdred()
    pyraf.iraf.zerocombine.unlearn()
    pyraf.iraf.zerocombine(input=infiles, **kwargs)


## Misc ##

def set_aperture(infile, section):
    """Create an aperture definition file for apsum to use."""
    # section is [left:right,down:up]
    column = section[1:-1].split(',')[1]
    (down, up) = column.split(':')
    center = (float(up) - float(down) + 1) / 2.
    rup = center
    rdown = -center
    tmp = []
    # details here obtained through reverse engineering of aperture files
    # generated by IRAF
    tmp.append('begin\taperture %s 1 1024. %s\n' % (infile, center))
```

```
    tmp.append('\timage\t%s\n' % infile)
    tmp.append('\taperture\t1\n')
    tmp.append('\tbeam\t1\n')
    tmp.append('\tcenter\t1024. %s\n' % center)
    tmp.append('\tlow\t-1023. %s\n' % rdown)
    tmp.append('\thigh\t1024. %s\n' % rup)
    tmp.append('\tbackground\n')
    tmp.append('\t\txmin -10.\n')
    tmp.append('\t\txmax 10.\n')
    tmp.append('\t\tfunction chebyshev\n')
    tmp.append('\t\torder 1\n')
    tmp.append('\t\tsample -10:-6,6:10\n')
    tmp.append('\t\tnaverage -3\n')
    tmp.append('\t\tniterate 0\n')
    tmp.append('\t\tlow_reject 3.\n')
    tmp.append('\t\thigh_reject 3.\n')
    tmp.append('\t\tgrow 0.\n')
    tmp.append('\taxis\t2\n')
    tmp.append('\tcurve\t5\n')
    tmp.append('\t\t2.\n')
    tmp.append('\t\t1.\n')
    tmp.append('\t\t1.\n')
    tmp.append('\t\t2048.\n')
    tmp.append('\t\t0.\n')
    tmp.append('\n')
    if not os.path.isdir('./database'):
        os.mkdir('./database')
    with open('./database/ap%s' % infile.replace('/', '_'), 'w') as f:
        f.writelines(tmp)
```

A.11 Source for mslit.misc.py

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

'''
Some basic math functions and some convienience functions.

math functions: avg, rms, threshold_round, std
convienience functions: base, list_convert, remove_nan, zerocount
'''

import cmath
```

```python
import math
import os
import numpy

## Some Math ##


def avg(*args):
    """Return the average of a list of values."""
    float_nums = [float(x) for x in args]
    remove_nan(float_nums)
    if len(float_nums) == 0:
        return float('NaN')
    return sum(float_nums) / len(float_nums)


def cubic_solutions(a, alpha, beta):
    """Calculate the solutions to a cubic function with given simplified
       parameters."""
    w1 = -.5 + .5 * math.sqrt(3) * 1j
    w2 = -.5 - .5 * math.sqrt(3) * 1j
    solution1 = -(1.0 / 3) * (a + alpha + beta)
    solution2 = -(1.0 / 3) * (a + w2 * alpha + w1 * beta)
    solution3 = -(1.0 / 3) * (a + w1 * alpha + w2 * beta)
    return [solution1, solution2, solution3]


def cubic_solve(b0, b1, b2, b3):
    """Calculate the solutions to a cubic function with given parameters."""
    a = b2 / b3
    b = b1 / b3
    c = (b0) / b3
    m = 2 * (a ** 3) - 9 * a * b + 27 * c
    k = (a ** 2) - 3 * b
    n = (m ** 2) - 4 * (k ** 3)
    alpha = (.5 * (m + cmath.sqrt(n))) ** (1.0 / 3)
    beta = (.5 * (m - cmath.sqrt(n))) ** (1.0 / 3)
    return cubic_solutions(a, alpha, beta)



def rms(*args):
```

```python
    """Return the root mean square of a list of values."""
    squares = [(float(x) ** 2) for x in args]
    return math.sqrt(avg(*squares))


def threshold_round(number, threshold):
    """Round a number using a configurable threshold value."""
    if math.modf(number)[0] < threshold:
        return int(math.floor(number))
    else:
        return int(math.ceil(number))


def std(*args):
    """Return the standard deviation of a list of values."""
    mean = avg(*args)
    deviations = [(float(x) - mean) for x in args]
    return rms(*deviations)


## Convenience functions ##


def base(location, values, step):
    """Increase location by step until values[location] stops decreasing.
       Return this value."""
    while True:
        if values[location] >= values[location + step]:
            location += step
        else:
            break
    return step


def list_convert(pylist):
    """Convert python lists to the strings that IRAF accepts as lists."""
    stringlist = pylist[0]
    for item in pylist[1:]:
        stringlist += ', %s' % item
    return stringlist
```

```python
def namefix(name):
    """Rename files to get rid of the silly naming scheme that apsum uses."""
    os.rename('%s.0001.fits' % name, '%s.fits' % name)


def remove_nan(*lists):
    """Remove NaNs from one or more lists. If more than one list is given,
       keep the shape of all lists the same."""
    for l in lists:
        count = 0
        for i, item in enumerate(l[:]):
            if numpy.isnan(item):
                for item in lists:
                    del item[i - count]
                count += 1


def zerocount(number):
    """Return the three digit representation of a number."""
    if number < 10:
        return '00%s' % number
    elif number < 100:
        return '0%s' % number
    else:
        return '%s' % number
```

A.12 Source for mslit.sky.py

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

"""
Functions related to sky subtraction.

high level iraf wrappers: combine_sky_spectra, setairmass_galaxy, skies
                          sky_subtract_galaxy
low level FITS functions: find_line_peak, find_lines, get_continuum,
                          get_peak_cont, get_wavelength_location
functions for solving: get_std_sky, guess_scaling, try_sky
high level functions: generate_sky, modify_sky, sky_subtract
"""

import os
```

```python
import subprocess
import pyfits
import scipy.optimize
from .data import get, get_object_spectra, get_sky_spectra, write
from .iraf_low import sarith, scombine, setairmass
from .misc import avg, base, list_convert, rms, std, zerocount


# define some atmospheric spectral lines
LINES = [5893, 5578, 6301, 6365]


## High level IRAF wrappers ##

def combine_sky_spectra(name):
    """Convert all sky spectra to the same scaling, then combine them."""
    sky_list = get_sky_spectra(name)
    sizes = get(name, 'sizes')
    scaled = []
    for spectra in sky_list:
        scale = sizes[spectra] # scale by the number of pixels arcoss
        num = zerocount(spectra)
        sarith('%s/disp/%s.1d' % (name, num), '/', scale,
            '%s/sky/%s.scaled' % (name, num))
        scaled.append('%s/sky/%s.scaled' % (name, num))
    if os.path.isfile('%s/sky.1d' % name):
        os.remove('%s/sky.1d' % name)
    scombine(list_convert(scaled), '%s/sky.1d' % name)


def setairmass_galaxy(name):
    """Set effective air mass for each object spectra in a galaxy."""
    spectra = get_object_spectra(name)
    for spectrum in spectra:
        num = zerocount(spectrum)
        setairmass('%s/sub/%s.1d' % (name, num))

def skies(name):
    """Create a combined sky spectrum, perform sky subtraction, and set
       airmass metadata """
    if not os.path.isdir('%s/sky' % name):
        os.mkdir('%s/sky' % name)
```

```python
    combine_sky_spectra(name)
    if not os.path.isdir('%s/sub' % name):
        os.mkdir('%s/sub' % name)
    sky_subtract_galaxy(name)
    setairmass_galaxy(name)


def sky_subtract_galaxy(name):
    """Remove sky lines from each spectra in a galaxy, making a guess at an
       appropriate scaling level if none is stored already."""
    spectra = get_object_spectra(name)
    sky_levels = get(name, 'sky')
    for spectrum in spectra:
        sky_level = sky_levels[spectrum]
        if not sky_level:
            sky_level = sky_subtract(name, spectrum)
        generate_sky(name, spectrum, sky_level)
    write(name, 'sky', sky_levels)


## Functions for manipulating the fits data at a low level ##


def find_line_peak(data, location, search):
    """Find the local maximum near a given location. The third option control
       how far on either side of the expected wavelength location to
       consider."""
    search = range(int(location - search), int(location + search))
    values = [data[i] for i in search]
    peak_num = search[values.index(max(values))]
    return peak_num


def find_lines(name, num):
    """Find the locations of a number of sky lines in a FITS file."""
    fn = '%s/disp/%s.1d.fits' % (name, num)
    hdulist = pyfits.open(fn)
    data = hdulist[0].data
    header = hdulist[0].header
    locations = []
    for line in LINES:
        line_loc = get_wavelength_location(header, line)
```

```python
        locations.append(find_line_peak(data, line_loc, 5))
    return locations


def get_continuum(location, data):
    """Return the root means square of the continuum values around a
       location."""
    upcont_num = base(location, data, 1)
    downcont_num = base(location, data, -1)
    data = data.tolist()
    values = data[upcont_num:(upcont_num + 3)]
    values.extend(data[(downcont_num - 3):downcont_num])
    return rms(*values)


def get_peak_cont(hdulist, wavelength, search):
    """Return the maximum value near a given wavelength; also the local
       continuum level."""
    data = hdulist[0].data
    header = hdulist[0].header
    wavelength_location = get_wavelength_location(header, wavelength)
    peak_location = find_line_peak(data, wavelength_location, search)
    peak = data[peak_location]
    cont = get_continuum(peak_location, data)
    return peak, cont


def get_wavelength_location(headers, wavelength):
    """Find the location of a given wavelength withing a FITS file."""
    start = headers['CRVAL1']
    step = headers['CDELT1']
    distance = wavelength - start
    number = round(distance / step)
    return number


## Functions for solving for the proper level of sky subtraction ##


def get_std_sky(scale, name, num):
    """Attempt a sky subtraction at a given scaling, and return a metric of
       how good that scaling is.
```

```python
        A proper sky subraction should result in a basically smooth continuum
        left, so this function looks at the standard deviaton of the spectrum
        around spectral lines known to be atmospheric. These values are
        averaged and return, lower numbers are better."""
    scale = float(scale)
    try_sky(scale, name, num)
    locations = find_lines(name, num)
    fn = '%s/tmp/%s/%s.1d.fits' % (name, num, scale)
    hdulist_out = pyfits.open(fn)
    deviations = []
    for item in locations:
        values = hdulist_out[0].data[(item - 50):(item + 50)]
        deviations.append(std(*values))
    return avg(*deviations)


def guess_scaling(name, spectrum):
    """Make a guess at an appropriate scaling factor for sky subtraction.

        For each atmospheric spectral line given, find the difference
        between the peak and the continuum levels in both the sky spectrum
        and the object spectrum. The ratio of these is the scaling factor.
        Average the ratios from each line and return this value."""
    spectra = '%s/disp/%s.1d.fits' % (name, zerocount(spectrum))
    skyname = '%s/sky.1d.fits' % name
    spectrafits = pyfits.open(spectra)
    skyfits = pyfits.open(skyname)
    scalings = []
    for line in LINES:
        spec_peak, spec_cont = get_peak_cont(spectrafits, line, 5)
        sky_peak, sky_cont = get_peak_cont(skyfits, line, 5)
        scale = ((spec_peak - spec_cont) / (sky_peak - sky_cont))
        scalings.append(scale)
    return avg(*scalings)


def try_sky(scale, name, num):
    """Preform a sky subtraction at a given scaling, saving the result to a
        temporary location."""
    sky = '%s/sky.1d' % name
    scaled_sky = '%s/tmp/%s/%s.sky.1d' % (name, num, scale)
```

```
    in_fn = '%s/disp/%s.1d' % (name, num)
    out_fn = '%s/tmp/%s/%s.1d' % (name, num, scale)
    if not (os.path.isfile('%s.fits' % scaled_sky) or
            os.path.isfile('%s.fits' % out_fn)):
        sarith(sky, '*', scale, scaled_sky)
        sarith(in_fn, '-', scaled_sky, out_fn)


## Functions wrapping the solvers and providing output ##


def generate_sky(name, spectrum, sky_level):
    """Use sarith to perform sky subtraction at a given scaling level."""
    num = zerocount(spectrum)
    in_fn = '%s/disp/%s.1d' % (name, num)
    in_sky = '%s/sky.1d' % name
    out_fn = '%s/sub/%s.1d' % (name, num)
    out_sky = '%s/sky/%s.sky.1d' % (name, num)
    subprocess.call(['rm', '-f', '%s.fits' % out_fn])
    subprocess.call(['rm', '-f', '%s.fits' % out_sky])
    sarith(in_sky, '*', sky_level, out_sky)
    sarith(in_fn, '-', out_sky, out_fn)


def modify_sky(path, name, number, op, value):
    """Change the level of sky subtraction for a region by an increment."""
    os.chdir(path)
    sky_levels = get(name, 'sky')
    sky_level = sky_levels[number]
    if op == '+':
        new_sky_level = sky_level + value
    elif op == '-':
        new_sky_level = sky_level - value
    sky_levels[number] = new_sky_level
    write(name, 'sky', sky_levels)
    generate_sky(name, number, new_sky_level)

def sky_subtract(name, spectrum):
    """Optimize the get_std_sky function to determine the best level of sky
       subtraction. Return the value found."""
    num = zerocount(spectrum)
    guess = guess_scaling(name, spectrum)
```

```python
    os.mkdir('%s/tmp' % name)
    os.mkdir('%s/tmp/%s' % (name, num))
    xopt = scipy.optimize.fmin(get_std_sky, guess,
        args=(name, num), xtol=0.001)
    subprocess.call(['rm', '-rf', '%s/tmp' % name])
    return float(xopt)
```

A.13 Source for mslit.tables.py

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

"""
Functions for outputting LaTeX tables of data.

Functions:
Formatting: sigfigs_format
Basic tables; make_data_table, make_flux_table
Comparison tables: make_comparison_table, make_group_comparison_table
Table construction: make_table, make_multitable
Table details: arrange_galaxies, arrange_group, arrange_regions
"""

import numpy
from .const import GROUPS, LINES, LOOKUP
from .misc import avg, remove_nan, std


## Formatting ##


def sigfigs_format(x, n):
    """Format a number to a certain amount of significant figures."""
    if numpy.isnan(x):
        return '. . .'
    if n < 1:
        raise ValueError("number of significant digits must be >= 1")
    string = '%.*e' % (n-1, x)
    value, exponent = string.split('e')
    exponent = int(exponent)
    if exponent == 0:
        return value
    else:
```

```python
        return '$%s \\times 10^{%s}$' % (value, exponent)


## Basic tables ##


def make_data_table(galaxy):
    """Create a table of basic data for a galaxy."""
    keys = ['rdistance', 'OH', 'SFR']
    values = [LOOKUP[key] for key in keys]
    string = make_table((galaxy.regions,), keys, values, 'Number',
                        arrange_regions)
    with open('tables/%s.tex' % galaxy.name, 'w') as f:
        f.write(string)


def make_flux_table(galaxy):
    """Create a table of measured fluxes for all regions in a galaxy."""
    order = ['NII1', 'NII2', 'OII', 'OIII1', 'OIII2', 'OIII3', 'SII1', 'SII2',
             'halpha', 'hbeta', 'hgamma']
    lines = LINES.keys()
    lines.sort()
    for item in lines[:]:
        unused = [numpy.isnan(spec.fluxes[item]) for spec in galaxy.regions]
        if False not in unused:
            lines.remove(item)
    keys = [item for item in order if item in lines]
    values = [LOOKUP[item] for item in lines]
    string = make_table((galaxy.regions,), keys, values, 'Number',
                        arrange_regions)
    with open('tables/%sflux.tex' % galaxy.name, 'w') as f:
        f.write(string)


## Comparison tables ##


def make_comparison_table(galaxies, other):
    """Make a table showing data of different galaxies."""
    keys = ['grad', 'metal', 'type', 'bar', 'ring', 'env', 'region_number']
    values = ['Gradient (dex/R$_{25}$)', 'Metalicity at 0.4R$_{25}$',
              'Hubble Type', 'Bar', 'Ring', 'Environment', 'Number of Regions']
```

```python
    string = make_table((galaxies, other), keys, values, 'Name',
                        arrange_galaxies)
    with open('tables/comparison.tex', 'w') as f:
        f.write(''.join(string))


def make_group_comparison_table(galaxies, other):
    """Make a table showing data for groups of different kinds of galaxies."""
    galaxies += other
    keys = ['grad', 'grad_std', 'metal', 'metal_std']
    values = ['Gradient (dex/R$_{25}$)', 'Standard Deviation',
              'Metalicity at 0.4R$_{25}$', 'Standard Deviation']
    titles = {'env': 'Environment', 'ring': 'Ring', 'bar': 'Bar',
              'type': 'Hubble Type'}
    string = make_multitable(galaxies, keys, values, titles, arrange_group)
    with open('tables/comparison2.tex', 'w') as f:
        f.write(''.join(string))


## Actual tables ##


def make_table(groups, keys, values, name, command, multi=False):
    """Return a string representing a LaTeX table.

       groups: iterable containing different groups of galaxies
       keys: iterable containing the key values to be put on the table
       values: iterable containing the printable names corresponding to the
               keys in keys
       name: name of the first column
       command: every item in groups will be run through this command
       multi: optional parameter, if true, table will be used inside a
              multitable"""
    if multi:
        string = []
    else:
        string = ['\\begin{tabular}{ *{%s}{c}}\n' % (len(keys) + 1)]
    string += ['\\toprule\n %s ' % name]
    string += ['& %s ' % item for item in values]
    string.append('\\\\\n')
    for group in groups:
        string += command(group, keys)
```

```
    string.append('\\bottomrule\n')
    if not multi:
        string.append('\\end{tabular}\n')
    return ''.join(string)


def make_multitable(galaxies, keys, values, titles, command):
    """Return a string representing a LaTeX table containing multiple
       sub-tables.

       galaxies: list of galaxies
       keys: iterable containing the key values to be put on the table
       titles: dictionary of short name - print name pairs, one pair for each
               subtable
       command: this is passed through to the make_table function."""
    string = ['\\begin{tabular}{ *{%s}{c}}\n' % (len(keys) + 1)]
    for title in titles:
        v = dict([(group['name'], {}) for group in GROUPS[title]])
        for group in GROUPS[title]:
            for item in ('grad', 'metal'):
                items = [g.__dict__[item] for g in galaxies
                         if g.__dict__[title] in group['members']]
                v[group['name']].update({item: items})
        string += make_table((v,), keys, values, titles[title], command, True)
    string.append('\\end{tabular}\n')
    return ''.join(string)


## Details within tables ##


def arrange_galaxies(items, keys):
    """Return details of a table for comparing multiple galaxies."""
    string = ['\\midrule\n']
    for item in items:
        string.append(' %s ' % item.print_name)
        for key in keys:
            value = item.__dict__[key]
            if type(value) == str or key == 'region_number':
                value = str(value)
            else:
                value = sigfigs_format(value, 3)
```

```python
            string.append('& %s ' % value)
        string.append('\\\\\n')
    return string


def arrange_group(data, keys):
    """Return details of a table comparing groups of different kinds of
       galaxies."""
    string = ['\\midrule\n']
    for header, group in data.items():
        string.append(' %s ' % header)
        for key in keys:
            if key[-4:] != '_std':
                values = group[key]
                remove_nan(values)
                a = sigfigs_format(avg(*values), 2)
                s = sigfigs_format(std(*values), 2)
                string.append('& %s & %s ' % (a, s))
        string.append('\\\\\n')
    return string


def arrange_regions(regions, keys):
    """Return details of a table comapring regions within a galaxy."""
    string = ['\\midrule\n']
    for region in regions:
        string.append(' %s' % region.printnumber)
        if region.corrected != True:
            string.append('$^a$')
        for item in keys:
            if item in region.fluxes:
                value = region.fluxes[item]
            else:
                value = region.__dict__[item]
            string.append(' & %s' % sigfigs_format(value, 2))
        string.append(' \\\\\n')
    return string
```

B Measured Emission Line Fluxes

Number	[N II]λ6548	[N II]λ6583	[O II]λ3727	[O III]λ4959	[O III]λ5007	Hα	Hβ
1	2.7×10^{-15}	1.7×10^{-15}	4.9×10^{-15}	5.0×10^{-16}	9.1×10^{-16}	3.1×10^{-15}	1.1×10^{-15}
2	1.0×10^{-14}	3.2×10^{-14}	1.4×10^{-13}	1.1×10^{-14}	1.9×10^{-14}	6.9×10^{-14}	2.6×10^{-14}
3	8.1×10^{-14}	6.1×10^{-14}	3.2×10^{-13}	2.5×10^{-14}	4.7×10^{-14}	1.5×10^{-13}	5.8×10^{-14}
4	1.7×10^{-14}	3.9×10^{-14}	1.7×10^{-13}	9.6×10^{-15}	5.0×10^{-15}	6.3×10^{-14}	2.5×10^{-14}
5	2.5×10^{-15}	8.4×10^{-15}	2.8×10^{-14}	1.9×10^{-15}	5.7×10^{-15}	1.8×10^{-14}	6.7×10^{-15}
6	2.0×10^{-15}	4.8×10^{-15}	2.0×10^{-14}	1.4×10^{-15}	1.5×10^{-15}	7.7×10^{-15}	2.9×10^{-15}
7	3.5×10^{-15}	9.3×10^{-15}	3.5×10^{-14}	3.0×10^{-15}	6.9×10^{-15}	2.6×10^{-14}	9.7×10^{-15}
8	6.1×10^{-15}	1.6×10^{-14}	5.3×10^{-14}	6.4×10^{-15}	1.5×10^{-14}	4.1×10^{-14}	1.5×10^{-14}
9[a]	4.8×10^{-15}	5.9×10^{-15}	1.0×10^{-14}	2.3×10^{-15}	3.5×10^{-15}	7.0×10^{-15}	. . .
10[a]	1.4×10^{-14}	4.3×10^{-14}	1.9×10^{-14}	1.2×10^{-14}	9.0×10^{-15}	1.6×10^{-15}	. . .
11	3.6×10^{-15}	6.0×10^{-15}	2.1×10^{-14}	2.9×10^{-15}	5.7×10^{-15}	1.7×10^{-14}	6.5×10^{-15}
12	5.2×10^{-14}	2.0×10^{-13}	5.4×10^{-13}	4.2×10^{-14}	4.6×10^{-14}	4.5×10^{-13}	1.7×10^{-13}
13	1.5×10^{-14}	5.2×10^{-14}	3.0×10^{-13}	1.3×10^{-14}	3.3×10^{-14}	1.2×10^{-13}	4.8×10^{-14}
14	6.8×10^{-15}	2.5×10^{-14}	8.6×10^{-14}	6.0×10^{-15}	1.6×10^{-14}	6.1×10^{-14}	2.3×10^{-14}
15	3.5×10^{-14}	1.3×10^{-13}	4.3×10^{-13}	3.1×10^{-14}	7.8×10^{-14}	3.5×10^{-13}	1.4×10^{-13}
16	1.2×10^{-14}	2.9×10^{-14}	1.4×10^{-13}	1.2×10^{-14}	2.8×10^{-14}	7.6×10^{-14}	2.9×10^{-14}
17	3.0×10^{-15}	1.0×10^{-14}	4.2×10^{-14}	2.7×10^{-15}	6.9×10^{-15}	2.4×10^{-14}	8.8×10^{-15}
18	1.4×10^{-14}	4.9×10^{-14}	3.0×10^{-13}	2.3×10^{-14}	3.9×10^{-14}	1.2×10^{-13}	4.7×10^{-14}

Table 5: This table shows the extinction corrected fluxes of emission lines from 18 H II regions in NGC 3169. All values are quoted in units of ergs $\cdot$ cm$^{-2}\cdot$ s^{-1} $\cdot$ 10^{-16}. [a]The fluxes for all emission lines corresponding to H II regions nine and ten are not corrected for extinction, because I was unable to measure Hβ for these regions.

Number	[N II]λ6548	[N II]λ6583	[O II]λ3727	[O III]λ4959	[O III]λ5007	Hα	Hβ
1	8.6×10^{-15}	1.6×10^{-14}	1.1×10^{-14}	1.1×10^{-15}	2.1×10^{-15}	2.8×10^{-14}	1.0×10^{-14}
2	5.0×10^{-15}	2.0×10^{-14}	3.0×10^{-14}	6.6×10^{-16}	2.7×10^{-15}	5.0×10^{-14}	1.9×10^{-14}
3	7.4×10^{-16}	3.1×10^{-15}	5.6×10^{-15}	4.6×10^{-16}	9.7×10^{-16}	7.4×10^{-15}	2.7×10^{-15}
4	2.4×10^{-15}	8.6×10^{-15}	8.6×10^{-15}	8.4×10^{-16}	1.5×10^{-15}	1.8×10^{-14}	6.7×10^{-15}
5	4.8×10^{-16}	1.6×10^{-15}	2.9×10^{-15}	2.0×10^{-16}	4.4×10^{-16}	3.8×10^{-15}	1.4×10^{-15}
6	9.5×10^{-15}	1.8×10^{-14}	2.8×10^{-14}	1.9×10^{-15}	5.2×10^{-15}	4.5×10^{-14}	1.7×10^{-14}
7	3.5×10^{-15}	1.3×10^{-14}	2.4×10^{-14}	8.6×10^{-16}	2.8×10^{-15}	3.5×10^{-14}	1.3×10^{-14}
8[a]	3.7×10^{-15}	6.4×10^{-15}	4.6×10^{-15}	3.4×10^{-15}	4.4×10^{-15}	1.2×10^{-15}	...
9	8.6×10^{-16}	5.6×10^{-15}	4.0×10^{-15}	4.6×10^{-16}	4.4×10^{-16}	8.4×10^{-15}	3.1×10^{-15}
10	2.2×10^{-15}	7.9×10^{-15}	1.2×10^{-14}	1.4×10^{-15}	3.5×10^{-15}	2.1×10^{-14}	7.7×10^{-15}
11	1.7×10^{-15}	4.5×10^{-15}	1.1×10^{-14}	6.1×10^{-16}	1.4×10^{-15}	1.1×10^{-14}	4.2×10^{-15}
12	3.8×10^{-15}	1.1×10^{-14}	1.9×10^{-14}	7.7×10^{-16}	2.7×10^{-15}	2.9×10^{-14}	1.1×10^{-14}
13	3.5×10^{-15}	1.0×10^{-14}	1.3×10^{-14}	1.2×10^{-15}	4.5×10^{-15}	2.2×10^{-14}	8.3×10^{-15}
14	4.0×10^{-16}	1.1×10^{-15}	2.0×10^{-15}	5.0×10^{-16}	9.3×10^{-16}	2.6×10^{-15}	9.3×10^{-16}
15	2.5×10^{-15}	7.6×10^{-15}	1.3×10^{-14}	1.4×10^{-15}	2.1×10^{-15}	2.1×10^{-14}	7.9×10^{-15}
16	4.8×10^{-16}	7.9×10^{-16}	6.3×10^{-16}	...	1.2×10^{-16}	1.6×10^{-15}	5.7×10^{-16}
17	3.8×10^{-15}	8.3×10^{-15}	1.9×10^{-14}	2.4×10^{-15}	8.2×10^{-15}	2.6×10^{-14}	9.7×10^{-15}
18	4.6×10^{-15}	1.4×10^{-14}	...	2.1×10^{-15}	1.5×10^{-15}	2.5×10^{-14}	9.8×10^{-15}
19	3.5×10^{-15}	1.0×10^{-14}	1.2×10^{-14}	8.8×10^{-16}	2.0×10^{-15}	2.9×10^{-14}	1.1×10^{-14}
20	8.4×10^{-15}	2.0×10^{-14}	2.3×10^{-14}	5.2×10^{-15}	8.5×10^{-15}	4.1×10^{-14}	1.6×10^{-14}
21	5.3×10^{-15}	1.7×10^{-14}	2.8×10^{-14}	5.6×10^{-16}	2.1×10^{-15}	4.5×10^{-14}	1.7×10^{-14}
22	3.2×10^{-15}	6.8×10^{-15}	1.0×10^{-14}	6.8×10^{-16}	1.5×10^{-15}	1.6×10^{-14}	6.0×10^{-15}
23	2.3×10^{-15}	2.4×10^{-15}	1.2×10^{-15}	7.9×10^{-16}	6.5×10^{-16}	6.6×10^{-15}	2.5×10^{-15}
24	1.4×10^{-14}	4.5×10^{-14}	1.0×10^{-13}	9.8×10^{-15}	3.0×10^{-14}	1.2×10^{-13}	4.4×10^{-14}
25	1.0×10^{-15}	1.6×10^{-15}	2.4×10^{-15}	4.9×10^{-17}	2.2×10^{-16}	3.6×10^{-15}	1.3×10^{-15}
26	2.1×10^{-15}	5.6×10^{-15}	...	3.1×10^{-16}	1.5×10^{-15}	1.5×10^{-14}	5.7×10^{-15}
27	3.6×10^{-15}	5.8×10^{-15}	1.5×10^{-14}	6.6×10^{-16}	3.1×10^{-15}	1.3×10^{-14}	5.0×10^{-15}
28	8.2×10^{-16}	1.5×10^{-15}	9.2×10^{-16}	2.1×10^{-16}	3.5×10^{-16}	2.7×10^{-15}	9.8×10^{-16}
29	6.7×10^{-16}	2.9×10^{-15}	7.0×10^{-15}	5.8×10^{-16}	2.1×10^{-15}	6.2×10^{-15}	2.3×10^{-15}

Table 6: This table shows the extinction corrected fluxes of emission lines from 29 H II regions in NGC 4725. All values are quoted in units of ergs $\cdot$ cm$^{-2}\cdot$ s^{-1}. [a]The fluxes for all emission lines corresponding to H II region eight are not corrected for extinction, because I was unable to measure Hβ for this region.