Markov Chain Monte Carlo Constraints on the Evolution of the Far-Infrared Galaxy Luminosity Function

An Honors Thesis for the Department of Physics and Astronomy

Noah Alexander Kurinsky

Tufts University, 2014

Abstract

In this thesis, I describe a program designed to constrain the redshift evolution of the far-infrared luminosity function, in an attempt to provide a general package useful for studying the Cosmic Infrared Background (CIB). I adopt a model which is phenomenological in nature, modeled after local luminous and ultra-luminous infrared galaxies, and include color evolution to account for the higher luminosity of such galaxies at large redshift. The program employs a Markov Chain Monte Carlo approach to constrain the evolution of the luminosity function where the locally determined parameters are held constant.

This program has been designed to be highly general, and is not specific to any electromagnetic regime, instrument, or survey. I have implemented robust methods for taking into account a wide range of different observational scenarios, and have produced a program which is modular and easily adaptable to a wide variety of analyses. A large goal in construction of this program was to provide a general tool useful for characterizing any type of large survey conducted in a regime for which even basic models are available.

As an example of an application of this code, I present the results from the analysis of a Herschel SPIRE survey of the Spitzer First Look Survey field, and discuss the results found by my program compared to earlier studies from the literature. Due to differences between the way the parameters are fit and the way in which others, namely Caputi et al. (2007) and Marsden et al. (2011) fit theirs, one does see some marked differences, which I will explain in the context of our fitting methodology and models. Lastly, I will discuss future improvements which might improve the fitting ability of the program.

Contents

1 Introduction

In the late 1920's, Edwin Hubble employed a large optical telescope and photographic plates to produce detailed spectra of tens of galaxies. He observed that the frequency of the light emitted by different galaxies was shifted in proportion to their independently measured distances, and assumed that this shift, which we term redshift, could be attributed to the Doppler effect. By fitting a linear trend to his plot of velocity versus distance, he provided the first evidence suggesting that the local universe is expanding (Hubble, 1929). The slope of this line, shown in figure 1, is the local value of the Hubble constant, H_0, and was the first true cosmological parameter ever measured.

Contrast Hubble's measurement of H_0 to the most recent cosmological measurements made by the BICEP2 team, who employed 3 years of cumulative observations of the cosmic microwave background (CMB[1]) to measure the "tensor-to-scalar-ratio", r, with high enough precision to conclude it is non-zero (BICEP2 Collaboration et al., 2014). To do this, they developed a cutting edge microwave telescope with brand new detector technology to measure polarization of the CMB signal, and accumulated spectra of a small patch of the southern sky every day for 3 years. Combining this data with that from the *Planck* (Planck Collaboration et al., 2011) and *WMAP* (Komatsu et al., 2011) CMB space telescopes and two other ground based CMB observatories, ACT and SPT (see references in BICEP2 Collaboration et al., 2014), they employed a fitting package called COSMOMC (Lewis & Bridle, 2002) to fit not only r but the 20 other parameters in the ΛCDM model, the cosmological model thought to best describe the evolution of the universe. The likelihood space they obtain for r using this package, both with and without their data, can also be seen in figure 1, on the right.

While these examples come from cosmology, they illustrate how the astrophysical sciences have changed as a result of both technological and scientific advances. Both discoveries have immense implications for the field given the scientific climate they were made in, but the modern effort was much more time consuming and complex, requiring a whole host of expertise from instrument design to computational modeling. The complexity of modern astrophysical experiments make end to end analyses by individuals or small teams a much rarer event. The role of standard "data reduction pipelines" has become much more important, and a survey of literature detailing various stages of catalogs from such telescopes reveals the importance of a few key packages[2].

The study of galactic evolution is a field which faces similar challenges. Hubble's work demonstrated that redshift is a proxy for distance, and due to the fixed speed of light, we observe more distant galaxies as they were when the universe was younger. As the distant galaxies we want to observe are immensely faint, observing them has required the development of new generations of powerful observatories. Due to the more recent and disparate nature of the large observational efforts to observe high redshift galaxies, fewer of these standard pipelines exist. As new observatories capable of imaging more distant galaxies have come on-line, we have seen an influx of large datasets which require more robust analyses than we have previously seen. For those observatories studying relatively local objects with traditional technologies (e.g. CCDs and optical spectrographs), spectra can be sampled with high enough precision to study each galaxy in its own right. Large surveys of high redshift galaxies cannot typically produce highly resolved, continuous, high confidence spectra, due both to their intrinsic faintness and the limitations imposed by the observational regime in which we must observe them due to their high degree of redshift. These galaxies typically have redshifts greater than a redshift of one, corresponding to observed emissions at half the frequency they were originally emitted or less.

[1]The CMB is a radiative surface created shortly after the big bang, when matter began to dominate the universe. It is often called an "echo" of the big bang.

[2]The *Planck* and *Herschel* (Pilbratt et al., 2010) space telescope point source catalogs are an excellent example

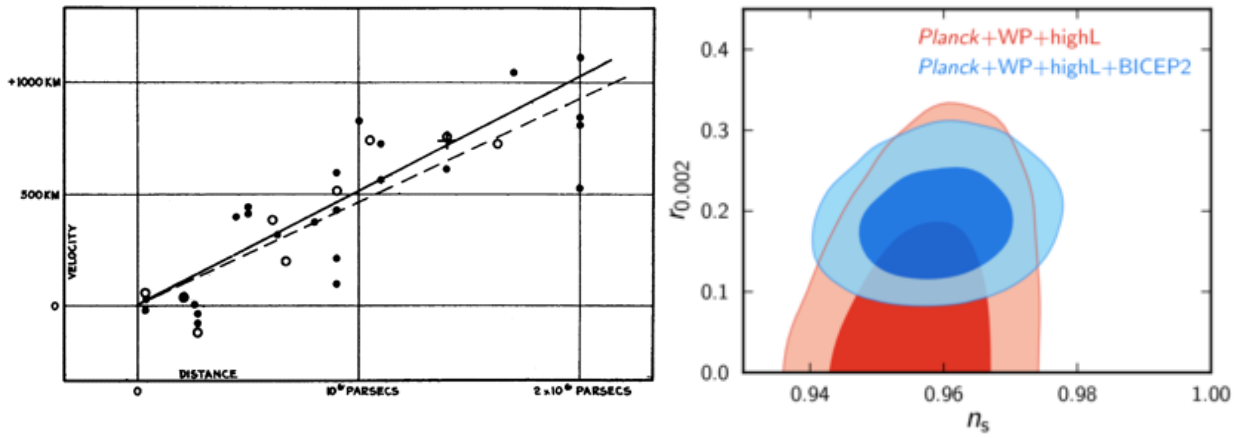


Figure 1: Left: Famous result from Hubble (1929) demonstrating the expansion of the universe, plotting recession velocity measured by redshift versus distance."The black discs and full line represent the solution for solar motion using the nebulae individually; the circles and broken line represent the solution combining the nebulae into groups; the cross represents the mean velocity corresponding to the mean distance of 22 nebulae whose distances could not be estimated individually" (Hubble, 1929) Right: Results from BICEP2 Collaboration et al. (2014) showing the effect on the likelihood space of r when the BICEP2 data was added. The contrast between these two plots, in combination with the discussion in the text, is meant to illustrate the new challenges facing the study of evolutionary processes in the universe, comparing a relatively simple measurement with an immensely complicated one.

What these surveys produce instead are catalogs of compact sources[3] with flux[4] measurements at a few different frequencies, which are then typically analyzed as a population to characterize the trends seen across age and intrinsic brightness of galaxies. If independent distance measures are available, this is relatively straightforward; given a distance and observed brightness, the age and intrinsic luminosity of a galaxy are trivial to compute. The majority of the most distant sources are brand new, however, and due to the sparse nature of the spectra it is impossible to determine an accurate distance measure for these galaxies. Alternatively, one can reconstruct the same basic plots made when distance measures were available by employing simulation techniques similar to those used by the CMB efforts described earlier. If we have spectral models of galaxies at various ages and brightnesses, a prescription for how they evolve, and a way to model the effects of observing them with a particular instrument, we can determine both the likelihood of different evolutionary scenarios as well as the validity of our spectral models.

It is this approach which I attempt to build upon in this thesis, and thus the thesis will contain a section that describes evolutionary models, as well a a section devoted to simulation of observations and statistics of model fitting. In this section, I will first give a brief introduction to the study and characterization of galaxies, and how we study their evolution, and then I will summarize my intended simulation approach in the context of the state of similar approaches within the field.

1.1 Characterizing and Observing Individual Galaxies

Galaxies are traditionally studied, on an individual basis, by obtaining detailed "spectral energy distributions" (SEDs), or measurements of observed brightness as a function of the frequency of radiation observed. These are typically compiled either by diffraction of light (to separate it by wavelength) onto a detector, to

[3]Observed objects which can not be resolved by the telescope due to limited resolution, believed to be galaxies

[4]Here and throughout this paper, consider flux to be synonymous with "observed intensity" if the term is unfamiliar to you.

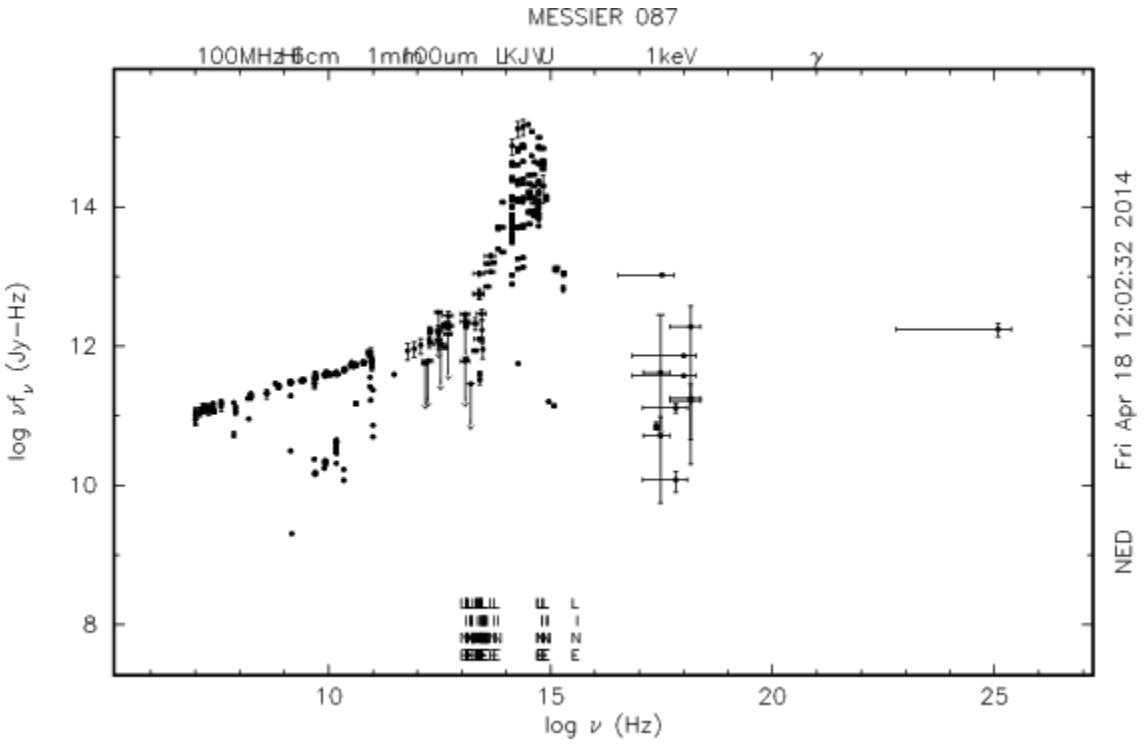


Figure 2: SED from NASA NED (`ned.ipac.caltech.edu`) of M87, a massive local elliptical galaxy ($z \sim 0.004$). Note that the optical regime is around 10^{15} Hz, where M87 is brightest; if we were to place this galaxy at redshift 2 or 3, it would shift to the left by around an order of magnitude, and would move down the intensity scale by a few orders of magnitude as well, making it virtually undetectable in the optical and incredibly faint in the infrared.

produce a continuous frequency versus intensity curve, or by making closely spaced measurements at many frequencies. For 'local' galaxies (those with very low redshifts), we have been conducting such observations for decades, and have a wealth of historical data and analyses collecting these spectra and explaining their physical origins within the local galaxies. In a few cases, we have spectra covering all electromagnetic regimes from radio waves, through the infrared, optical, ultraviolet, x-ray and gamma rays. M87, a massive elliptical galaxy, is one such example. It hosts billions of intermediate and low mass stars and a very active radio core/jet. It is close enough to be imaged easily and with a high degree of precision by both ground and space based observatories, and very intrinsically bright, making it easy to study even with relatively insensitive observatories. The spectrum obtained from the NASA NED database can be seen in figure 2. Note in this figure that the maximum brightness occurs in the visual band; if we were to redshift this to redshift 2, it would be invisible in the optical regime, as it would be shifted left (red), and it would be much fainter due to the increased distance.

If detailed spectra can be obtained for an individual galaxy, many important physical processes which occur within the galaxy can be studied, to determine the dominant processes occurring within a galaxy of that type and age. Performing a meta-analysis of such studies of individual galaxies or small galactic population, we can produce detailed models of galaxy spectral energy distributions (SEDs) as a function of intrinsic luminosity and, to a certain extent, elemental and stellar composition. These are well established for galaxies in the local universe across many electromagnetic regimes, most notably in the ultraviolet (15 - 350 nm), optical (350 - 1000 nm), and near infrared (1 - 300 μm). There are exhaustive databases of spectra from galaxies across the electromagnetic spectrum, and very detailed analyses of classes of spectra

in specific regimes as in works such as Elvis et al. (1994) (QSO[5] SEDs) and Rieke et al. (2009) (local infrared galaxies).

Such libraries are much scarcer for older, more distant galaxies, which we expect to be systematically different as a consequence of evolutionary trends. There exist extensive empirical and phenomenological models for local galaxies, however depending again on regime, non-local templates vary greatly in detail to which studies have been conducted. There is a mounting body of evidence that the spectra of high redshift galaxies differ substantially from local galaxies of similar stellar mass and structure. Sanders & Mirabel (1996) were the first to observe local infrared galaxies with unusually large intrinsic infrared luminosity. They named these galaxies Luminous and Ultra-luminous Infrared Galaxies (LIRGS and ULIRGS), and due to their intrinsic brightness and propensity to become brighter at higher redshift, they are one of only a few classes of high redshift galaxies that we have been able to systematically study, though not in the extensive detail that we have studied local galaxies. The major limitation to the study of these galaxies is the inherent challenges to conducting infrared astronomy, including high atmospheric opacity across much of the infrared (restricting sensitive studies to space and balloon-borne missions) and technological challenges, which have only recently been surmounted[6]. It was not until the space-based Spitzer mission and similar efforts in the 1990s that these galaxies were first studied on a statistical level (see Sanders & Mirabel, 1996, and references therein), and not until the *Herschel* mission, launched in 2009, that large numbers could be individually resolved. Recent efforts to create IR SED template libraries focus mainly on the way that distant galaxies differ from local galaxies (Farrah et al., 2008; Sajina et al., 2012; Kirkpatrick et al., 2013). Such templates are integral not just to the study of galaxies as static objects, but more importantly, to the study of galactic evolution.

1.2 Studying Galactic Evolution

Moving from individual galaxies to galaxy populations, the study of galactic evolution entails characterizing the ways in which average properties of galaxies change over cosmic time. Evolutionary quantities discussed in the context of galaxy populations often include spatial density and intrinsic luminosity, as well as more derivative properties such as star formation rates, merger rates, and trends in galaxy morphology[7]. These properties are contained within or related to the galaxy luminosity function (LF), which is an analytical function describing the number density of galaxies as a function of intrinsic luminosity, time, and occasionally galaxy morphology. The most basic form of this function, the Schechter function (Schechter, 1976), is rooted in the physics of galaxy formation, and thus by constraining the shape of this function we can often make conclusions about formative galactic processes. The constraint of this function is central to most if not all evolutionary studies, and is the main product of most intense studies of new large, deep surveys (Binggeli et al., 1988).

The central problem in constraint of the LF is how best to parse the information in the survey into metrics related to the LF. The easiest and most ideal means by which to do this is to try to obtain distance measures for each source, and use one of the statistical methods in common practice (see those in Johnston, 2011) to fit a function at various redshifts. As discussed earlier, however, few of our sources will have measured distances. Without measured distances, this becomes a much more complicated and highly degenerate problem.

[5] A QSO, or quasar, is a high luminosity, high redshift Active Galactic Nucleus (AGN) with large emission in both the optical and radio regimes.

[6] The far infrared in particular is a difficult regime to study, as it is difficult to create a radiometer small enough to be sensitive to micron and mm wavelengths, and traditional radiation detectors for optical, near-infrared, and UV rely on semiconductor band-gaps which can't be as small as the energy of an infrared photon.

[7] In other words, the structural class to which a galaxy belongs, roughly separated into ellipticals (roughly spheroidal galaxies) and spirals, which have extended arm-like structures.

Such characterization of surveys is often accomplished by casting multiple flux measurements into spectral color plots. These metrics compare the ratios of fluxes in neighboring band-passes, in an effort to separate different types of galaxies based on their tendency to fall within given regions of the metric (for instance, as in Kurinsky et al., 2013, and references therein). Depending on the regime of operation and frequencies chosen, this method can either be highly illuminating or relatively meaningless; careful selection of observation bands can create color spaces which segregate even slight differences between galaxies. The mid and far infrared regimes display very strong behavior along these lines, as evidenced by Kirkpatrick et al. (2013) and others; color spaces have become the default means by which to characterize a large number of sources in a single metric, however the nature of infrared galaxies makes for particularly illuminating plots.

We can thus empirically model galaxy populations and their evolution, given SED templates, via the constraint of the LF and its dependence on redshift through comparison to such color-based metrics. We typically model evolution of the LF through redshift dependence of the overall spatial density (Φ^*) and characteristic luminosity (L^*), and we can adopt parameterized forms for these values in terms of their local value and the redshift under consideration, constraining the free parameters of these functions. Saunders et al. (1990) proposed the parameterization of early infrared forms as power laws in terms of the factor (1+z). A function parameterized in this way is able to describe the number density of galaxies of a given intrinsic luminosity at a given redshift in an analytical form. More recent forms extend this parameterization to higher degrees of freedom, allowing variation of the high-luminosity end of the luminosity function independently of the low luminosity end (Negrello et al., 2013), and including more explicit redshift parameterization in number density, characteristic luminosity, and evolution of model galaxy spectra (Caputi et al., 2007; Marsden et al., 2011).

1.3 Constraining Model Parameters

The challenge is thus to combine a luminosity function, SED models, and a survey metric in order to constrain all evolutionary parameters of the luminosity function, as well as any parameters in the SED models. It is imperative, given the high level of degeneracy[8] between many astronomical quantities, to employ a fitting method designed to illuminate such degeneracies within our chosen parameterization. In other words, we need to be sure to explore our entire parameter space, accurately reconstructing the likelihood space of each parameter, to find the global best fit set of parameters and their uncertainties.

Markov Chain Monte Carlo (MCMC) fitting has recently become a popular method by which LF parameter spaces can be explored and constrained through comparison to extragalactic surveys (such as in e.g. Marsden et al. (2011) Acquaviva et al. (2011), Johnson et al. (2013)), employing an approach similar to that described above. MCMC has been a standard tool in the cosmology community for constraining the cosmological parameters of the ΛCDM model from the CMB observations for over a decade (Lewis & Bridle, 2002; Spergel et al., 2003; Dunkley et al., 2009; Planck Collaboration et al., 2013; BICEP2 Collaboration et al., 2014, and references therein). The COSMOMC package discussed earlier employs the MCMC fitting technique, and has been the pioneering example of the application of MCMC to astrophysical datasets. Marsden et al. (2011) in particular used many components of COSMOMC in their fitting routines, demonstrating the applicability of the technique to LF evolutionary constraint. I will discuss MCMC in more detail in this paper, as it is one of the central features of the software package this thesis describes.

1.4 Thesis Overview

In this thesis, I extend the ideas presented in Marsden et al. (2011) from one spectral color to two. Whereas they attempt to fit their model using the single 60-100 μm color, I instead employ a two color model,

[8] In this paper, degeneracy refers to the tendency of two values to be correlated. When fitting parameters, degenerate parameters are those which, through their covariance, can both be varied to maintain the quality of a fit.

using three spectral bands, to create a two-dimensional diagnostic spectral color density plot of an observed population. I then employ a library of LIRG and ULIRG SED models to try to reproduce the observed trend, exploring the parameter space of my parameterized LF to constrain the its redshift evolution in the far-infrared. Using the luminosity function as a general prescription for number density and luminosity of galaxies at a given redshift, one can use the constrained parameters to simulate a high-redshift survey, given characteristics of the instrument performing the survey.

The majority of efforts to study galaxies on evolutionary time-scales develop an analytic pipeline specific to their needs, explicitly using assumed forms of the above elements and producing code which is hard to extend to efforts employing different observations or models. The complex nature of isolating evolutionary trends amid mixed galactic populations, and the requirement that either new methods be engineered, or past efforts reconstructed, makes the process of arriving at a publishable result immensely time consuming and tedious.

The availability of a general tool for the constraint of LF evolution would greatly reduce the prerequisite knowledge required to conduct this type of analysis, and allow for more efficient and reproducible studies to be carried out on a wider variety of datasets previously deemed unworthy of the large time commitment required if conducting this analysis from scratch. I have begun the development of such a tool in this thesis, developing a general MCMC algorithm, and a software package capable of simulating a wide range of surveys, taking into account various LF forms, SED models, and instrumental properties, including noise levels and intensity limits. My program allows me to characterize a survey as well as highlight the successes and failures of the SED template and the luminosity function form used by the fitting procedure, in order to speak to the nature of the data as well as the strength and weakness of the chosen models.

I have put a large emphasis on ensuring that the fitting program is highly generalizable and easily accessible to those working with standard computational tools for astronomy, including IDL and python. All models are stored externally to the main program in FITS format[9], and the luminosity function and various other procedures are highly modularized in the C++ code, which forms the main analytical engine for the package. Due to these features, the SED templates desired by the user can be specified at runtime, and the luminosity function can be easily modified. The program is not specific to any instrument or observational regime, and the user can specify the exact survey characteristics (such as flux limits and statistical noise levels) to achieve the best fit possible. Such considerations allow us to address the need for reusable code, with the intent being that the analysis of data sets of this nature can be done much more efficiently, and result in a larger, richer scientific yield. In addition, I see some dependence of survey outcome on instrumental properties, and thus ability to tweak such characteristics has a large impact on the success or failure of fitting attempts.

In the remainder of this thesis, I describe the statistical methods and theoretical considerations inherent in my program, present an example use case, and discuss the results of this case. In section 2, I discuss the assumptions made by the current form of the fitting procedure, such as luminosity functional form as well as the form of the redshift and model distributions and the various cosmological considerations. In section 3, I present the mathematical methods employed to randomize the simulation, perform the fitting, and generate the fitting statistic, and discuss the program's implementation. In section 4, I present the results of testing the program with a far-infrared survey of the the Spitzer FLS field obtained by the HerMES survey (Smith et al., 2012), using Herschel's SPIRE instrument (Pilbratt et al., 2010; Griffin et al., 2010). In section 5, I discuss the results of this effort, including successes and failures of the simulation methodology, as well as places where improvement should be made within various model facets. Finally, I summarize my results and conclusions from this thesis in section 6.

[9]The FITS format is the file structure used to store the vast majority of astronomical data and images, and any astronomer should be able to view these files with any number of free programs readily available on the internet

2 Modeling Galactic Populations

To produce a simulated galaxy population suitable for comparison to an observed population, we must adopt models for individual galaxies which cover the entire range of galaxy types we expect to observe, a prescription for how different classes of galaxies combine to create the distribution of intrinsic luminosities described by the luminosity function, and a model describing the cosmology of the space between the simulated galaxy and the observer. The most robust way to meet the first requirement is to adopt a library of galaxy SED templates based in part on real spectra from diverse galaxy types, as I describe in section 2.1. Additionally, we must select a luminosity function which is well parametrized and amenable to frequent parameter changes, and can, through such changes, produce universes with widely ranging characteristics. I discuss the luminosity function employed for this effort in section 2.2. To describe the universe at various redshifts, and therefore various times, we must be able to model distance and volume in an ever changing universe, and thus need to adopt a cosmological model, which I will discuss in section 2.3. Finally, we must take into account observational effects, biases, and errors, which I will discuss briefly in section 2.3.3. Combining all of these, as described in section 2.4, finally allows me to reach an accurate, robust simulation of an observation over a given area adequate for comparison to observation.

Even though the following sections at times describe specific implementations of the various aspects of the fitting program, it is important to note that many of these specific forms are only those adopted for my far-infrared fitting and modeling methods; these include mainly the SEDs and luminosity function methods. The program has been designed and implemented to work with arbitrary SEDs models, and thus the specific SED templates are not integrated into the compiled code itself, but are read in from files and are fully changeable. In order to handle generic SEDs, the functions which extract flux densities and compute observed fluxes based on a galaxy SED are general, and in the interchangeability of the SEDs lies much of the power of this approach. The luminosity function may, in the future, operate in a similar way; currently it is hard-coded, although coded such that a user with basic knowledge of C++ could easily modify it and recompile the fitting routine. The overall design of the program is to strive for as few specific model assumptions as possible and focus on making the methods common to all regimes as robust as possible.

2.1 Infrared SED Libraries

For the analysis of the infrared sample discussed in section 4, I adopt a library of SED templates from Rieke et al. (2009), who construct a set of templates from averaged fits of theoretically based models to local galaxies, binned in terms of luminosity. These templates, seen in figure 3, have associated intrinsic luminosities, due to their adaptation from real local galaxies, and range in far-infrared luminosity from $10^9 - 10^{13} L_{sun}$. A closer view of our specific region of interest, with the brightest and faintest SEDs at redshifts between $z \sim 0 - 5$, can be seen in figure 4.

These templates are mainly based on spectra obtained from local IRAS galaxies selected for being highly star forming. It is uncertain to what degree they remain valid models at high redshift, though we know there must be a limit to their viability due to overall galaxy evolution (see e.g. Sajina et al., 2012). Rieke et al. (2009) discuss model validity up to redshifts around $z \sim 2.5$, and conclude that their models can be considered fairly accurate for mid-infrared analyses, although they suggest that some boosting due to AGN should be added. I attempt to compensate for the local nature of these templates by introducing color evolution into our models, discussed in short below.

The design of this program allows for evaluation of morphological completeness, and thus it would be ideal to eventually to employ an SED library which has built in redshift evolution, and multiple morphological classes. As noted in Wiklind (2003) (who employed modified blackbodies to constrain the redshift of far-infrared galaxies), the introduction of such an SED treatment leads to a large amount of parameter degeneracy, and for this iteration of the program my main concern was to implement the Monte Carlo fit-

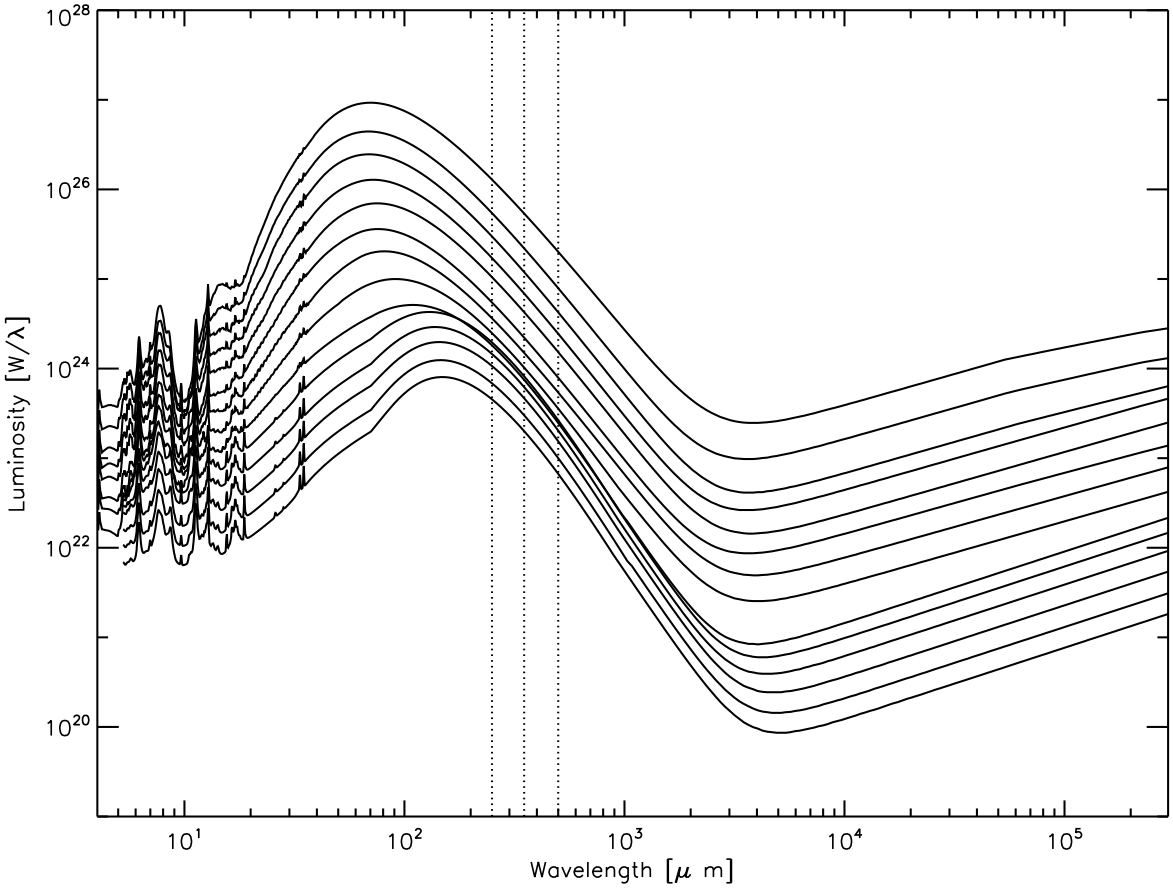


Figure 3: SED Library consisting of LIRG and ULIRG templates ranging in far-infrared luminosity from 10^9 to 10^{13} L_{sun}, taken from Rieke et al. (2009). These templates are based on a mix of data from local galaxies and models of galaxies as modified blackbodies in the far-infrared regime, mixed with radio-loud tails seen on the far right of the models in the above plot. The transition between LIRG and ULIRG can be seen where the models seem to group through the FIR region. Currently, these models are used with their associated luminosities over the entire redshift range, however with color evolution we would see intrinsic luminosities increase with redshift without a corresponding temperature rise (these models will rise along the Luminosity axis with redshift).

ting routine and perform a basic sample fit, thus I chose a simpler model template in order to focus on the creation of this fitting routine.

2.1.1 SED Color Evolution

One of the most substantial findings to come out of the first infrared observations with IRAS was first the discovery of luminous infrared galaxies (LIRGs) (Sanders & Mirabel, 1996), which represent a class of the brightest objects in the universe, most numerous during the peak epoch of star formation around and above a redshift of ~ 1 (Devlin et al., 2009). LIRGs appear similar to more local galaxies, except with luminosities far exceeding those we normally encounter at lower redshift. In order to reproduce such galaxies, I introduce

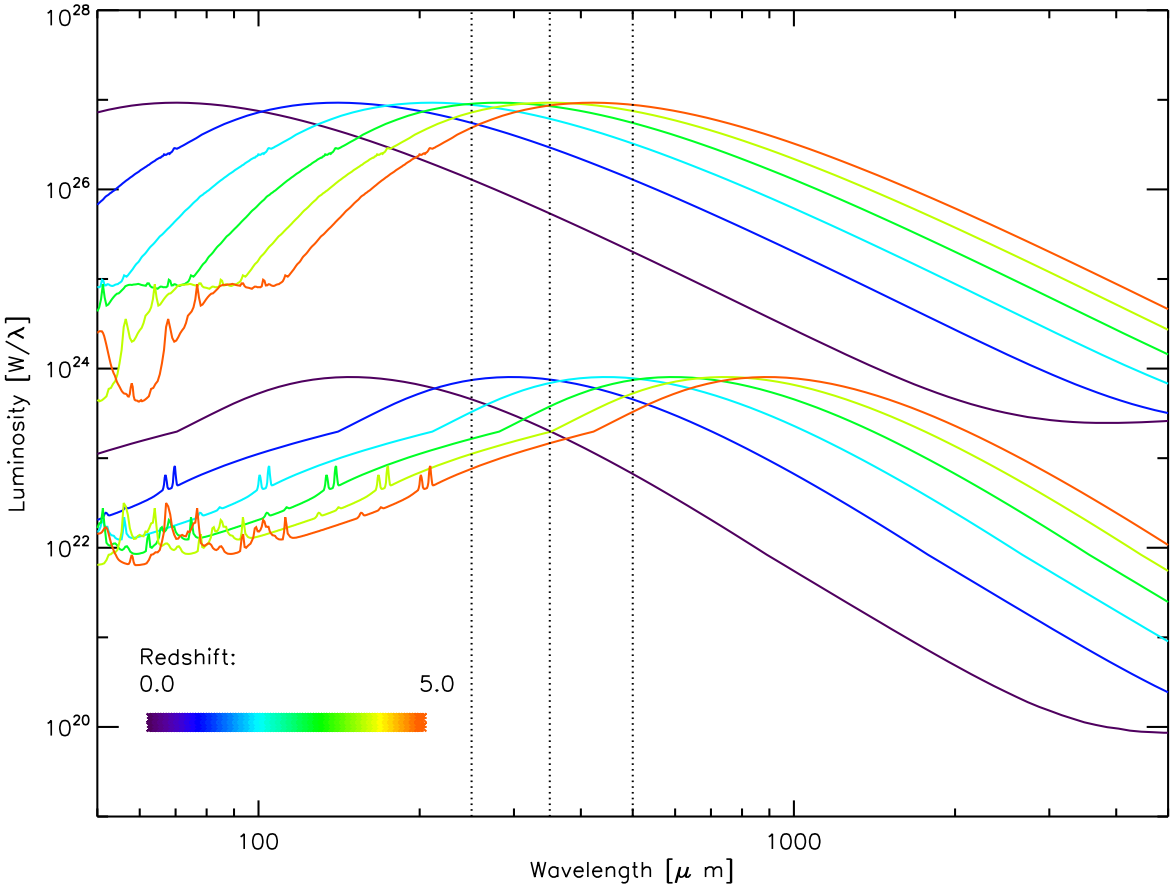


Figure 4: An illustration of how the intensity and slope of the SED varies as a function of redshift, assuming the intrinsic luminosity of the template is the same at all redshifts. Here we see the brightest and faintest SED models from the Rieke et al. (2009) template library, redshifted between $z = 0$ and $z = 5$ in steps of $z = 1$. We can see that intrinsic luminosity increases from low redshift, and slope across the three marked bands (250, 350, and 500 μm) rises from negative to positive with redshift. This helps to explain the overall trend we see when projecting the models into our color-color space.

the concept of color evolution into our models, using the parameterization of Marsden et al. (2011):

$$L(z) = L_0(1 + z)^a$$

Here $L(z)$ is the luminosity given to a template with luminosity L_0 at $z = 0$, and a is a parameter which takes on values greater than or equal to 0. We term this "color evolution" due to the fact that if we select the same intrinsic luminosity at different redshifts, we end up observing different models, and hence see different colors. I do not evolve the color within a given template, but by evolving the luminosity in this way allow the properties of the library to evolve as well. I will discuss the effects this has on the overall simulations when discussing results of fitting, and will demonstrate the impact of adding color evolution to the SED models, removing color evolution, or adding evolution as a fitted parameter.

2.2 The Luminosity Function

The luminosity function Φ, described qualitatively in the introduction, is formally defined as the number density of galaxies at a given redshift with a given luminosity:

$$\Phi(L, z) = \frac{dN(L, z)}{dL(z)dV(z)}$$

where N is the number of galaxies in a given co-moving volume V as a function of redshift and luminosity, and $L(z)$ is the rest-frame luminosity of a galaxy (Johnston, 2011). The most common and most general functional form of the LF comes from Schechter (1976) and is referred to as the Schechter function:

$$\Phi(L)dL = \phi_0^* \left(\frac{L}{L_0^*}\right)^{\alpha} \exp\left(\frac{-L}{L_0^*}\right) \frac{dL}{L_0^*}$$

where Φ_0^* is the characteristic density of galaxies with characteristic luminosity L_0^*, and α is usually ~ -1.

The shape of this function, with a "knee" at L_0^*, is brought about through the effects of the initial mass function (IMF) and the efficiency of star formation as a function of galaxy mass. The IMF describes the distribution of pre-galaxy mass clusters which give rise to galaxies. If the IMF looks like a power law (a straight line in log-log space), and there exists an intermediate mass at which star formation reaches a peak efficiency, we will see a larger number of galaxies at this luminosity than higher and lower luminosities than we would expect if we assumed the LF was directly proportional to the IMF. This argument was that used by Schechter (1976) to initially derive this function, and the fact that the natural distributions we observe tend to follow this trend allows us to use the luminosity function as a probe of star formation and the IMF at various stages in the universe. The basic shape of the Schechter function can be seen in the leftmost panel of figure 5.

By far the most important component of the program, the luminosity function determines the number and luminosity of generated sources for a given redshift, which I modify by altering the manner in which the luminosity function evolves with redshift. In this sense, it is the prescriptive aspect of the model. To generate a galaxy population in a given luminosity and redshift bin, I simply integrate the luminosity function over these ranges to determine the number of galaxies intended, and draw this number of galaxies from our SED template library. Centering our simulation on such a prescription allows for easy comparison to surveys in which the LF can be constructed directly from data due to availability of spectroscopic redshifts.

The common way in which the LF is parameterized according to redshift is by adopting (Johnston, 2011):

$$\Phi^*(z) = \Phi_0^*(1+z)^p \quad L^*(z) = L_0^*(1+z)^q$$

This essentially explores the space of density and luminosity evolution, where p controls the degree to which co-moving galaxy density increases (for positive p) or decreases (for negative p) with cosmic time, and q describes the same change but for characteristic luminosity of galaxies. The effect of each type of evolution on the schechter function can be seen in the middle and right panels of figure 5. The co-moving aspect of this analysis means all changes are independent of universal scale factor changes; in other words, the results I find will be density increase or decrease on top of the background of universal expansion. It should be noted that these evolutionary trends can be used to make sweeping implications about formation processes within galaxies, as the relative evolution of these terms dictates whether a top-down or bottom-up theory of evolution is appropriate, and tells us, through characteristic luminosity, what the optimal luminosity of a galaxy is for formation during a given epoch.

The construction of luminosity functions, and their local measurements, is a very intense field of research, and I advise the reader to refer to Binggeli et al. (1988); Johnston (2011) for a more rigorous treatment. Furthermore, it should be noted that the specific form, due to the connection of the LF to the radiative

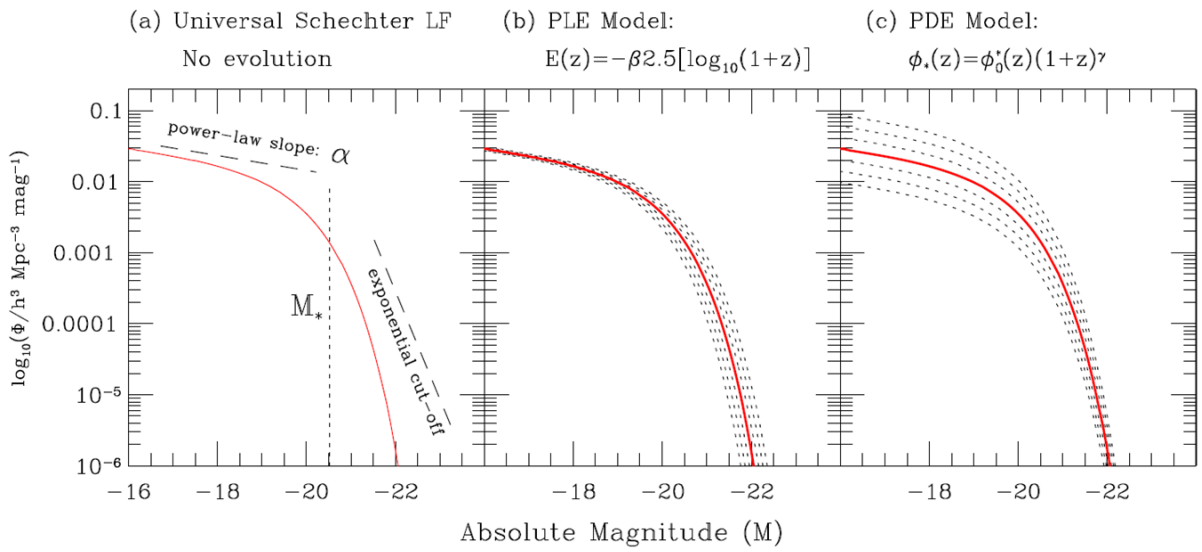


Figure 5: An illustration of the effect on the two types of evolution on a schechter function, compared to the no evolution model (figure taken from Johnston, 2011). Note here that the x axis is in units of magnitude, which is related to luminosity as $M = -2.5Log_{10}(L) + C$, where C is a constant dependent on the measured luminosity and magnitude of the sun and the units of luminosity employed.

mechanisms in the galaxy and their dependence on the IMF, is highly dependent on the regime under study. For more information on the high and low energy regimes, I further refer the reader to studies of the gamma ray (Ajello et al., 2012) and radio (Sommer et al., 2011) LFs and their evolution with redshift.

2.2.1 The Far-Infrared Luminosity Function

For the test sample described in section 4, I employ an IRAS-type double power-law luminosity function from Negrello et al. (2013):

$$\frac{dN}{dVdlogL}(L|L^*,\Phi^*,\alpha,\beta,z) = \Phi^*(z) * \left[\left(\frac{L}{L^*(z)}\right)^{\alpha} + \left(\frac{L}{L^*(z)}\right)^{\beta}\right]^{-1}$$

where Φ^* and L^* are evolved according to the parameterization in section 2.2; this parameterization has been explored in the infrared regime by Caputi et al. (2007). This parameterization is strongly supported by non-parameterized simulations performed by Marsden et al. (2011) in the far-infrared, where the general evolutionary trends are well fit by power laws.

The default values for α, β, Φ^*_0, and L^*_0 for 350 μm were taken from Negrello et al. (2013), and for this analysis are considered fixed values, as the luminosity function is well constrained at low redshift. The default values for p and q were taken from Caputi et al. (2007) for redshift evolution out to $z \approx 2$; the values I choose are those consistent with the high-redshift evolution from that paper. These values only serve as defaults, and can be altered via the interactive interface, as will be discussed below. For the current form of the program, I assume to start no evolution after redshift of ~ 2, as it is generally accepted due to analyses such as that presented in Marsden et al. (2011) that evolution ceases after some redshift cutoff. In section 4, I will show examples where this is fixed and where it is left as a free parameter. I will also discuss how this translates to simulated source numbers in section 2.4.

2.3 Cosmological Considerations

As I am simulating galactic populations spanning a large portion of the observable universe, correct calculation of distance and volume as a function of redshift, as well as correct compensation for redshift effects, is vital to the accuracy of our effort. Specifically, I need to calculate luminosity distance, co-moving volume, and correctly apply k-corrections when calculating observed flux values. I describe the formulae I apply in this section, and discuss various approximations and special considerations. I adopt the cosmology from WMAP ($\Omega_\Lambda = 0.728, \Omega_M = 0.272, H_0 = 70.4$) for my default values (Komatsu et al., 2011). Aside from the normal assumption of a flat universe ($\Omega_k = 0$), all cosmological parameters are easily modifiable, so that future users can adjust them as more precise measurements are made.

2.3.1 Luminosity Distance

In order to convert luminosity and redshift into a flux, as well as determine the detectability of a source given the magnitude limit of a sample, the program needs to be able to compute luminosity distance. Luminosity distance is computed by the standard formula:

$$d_L(z) = \frac{(1+z)c}{H_0} \int_0^z \frac{dz'}{\sqrt{(1+z')^3\Omega_M + \Omega_\Lambda}}$$

I compute this using numerical integration in redshift steps of $z = 0.001$ to maximize precision while allowing for real-time luminosity distance calculations.

2.3.2 Comoving Volume

In order to convert the luminosity function into a source number, I also need to compute the co-moving volume for a given redshift bin. For the case of a flat universe, the co-moving volume per solid angle between z_{min} and z_{max} is:

$$\frac{dV_C}{d\Omega} = \int_{z_{min}}^{z_{max}} \left(\frac{dV_c}{dz d\Omega}\right) dz \approx \sum_{z=z_{min}}^{z_{max}} \left(\frac{dV_c}{dz d\Omega}\right)_z \Delta z$$

for small Δz, where Ω is solid angle, and

$$\frac{dV_c}{dz d\Omega} = \frac{c\ [d_L(z)]^2}{H_0(1+z)^2\sqrt{\Omega_M(1+z)^3 + \Omega_\Lambda}}$$

The source number in a given redshift and luminosity bin is

$$dN(L,z) = \Phi(L,z)\frac{dV_c}{dz d\Omega}(z) dz d\Omega d\log L$$

or, in terms of binned luminosity and redshift, with constant angular area,

$$N(L,z,\Omega) \approx \Phi(L,z)\frac{dV_c}{dz d\Omega}(z)\Omega\Delta z\Delta\log L$$

2.3.3 Observed Flux Densities

The SED templates give me a luminosity per unit frequency in the emitted frame, however I need to produce a flux density in the observed frame. The observed flux and emitted luminosity are related according to the relation

$$\frac{L(\nu_o)}{d\nu_o} = \frac{d\nu_e}{d\nu_o}\frac{L(\nu_e)}{d\nu_e} = (1+z)\frac{L(\nu_e)}{d\nu_e}$$

where ν_e and ν_o are emitted and observed frequencies ($\nu_e = [1 + z]\nu_o$). I thus compute the flux from our luminosity densities for a source at redshift z according to

$$f_\nu(\nu_o) = \frac{(1+z)}{4\pi d_L^2} L_\nu(\nu_e)$$

where D_L is the luminosity distance discussed in section 2.3.1. See Hogg et al. (2002) for a more involved derivation and discussion.

I simulate observational error by adding Gaussian noise to the flux density calculated using the above relation. The user specifies, for each band, the flux minimum and characteristic flux measurement error for the instrument with which the observations were conducted, with the characteristic error taken as the variance of our noise distribution. The detectability of the source is determined by whether the flux density lies above the minimum value after noise is added. This is a rough substitute for the use of specific filters with measured characteristics, but is important for producing the spread of colors seen in our fitting diagnostic.

2.4 Simulating a Galactic Population

For each set of model parameters, I simulate an entire population of galaxies following the general prescription described earlier in this section. The user determines the acceptable redshift range and bin width, and I step through each redshift bin sequentially. For each redshift, I calculate, from the luminosity function, how many galaxies should be detected at that redshift for each luminosity bin; these bins are set by the number and intrinsic luminosity of the SED templates. If color evolution is being applied to the library, I also compute the fraction by which intrinsic luminosities of the SED library should be scaled before selecting a template for conversion to flux values. I then find the emitted wavelength corresponding to the observed frame wavelength for a given redshift, and extract the luminosity for that emitted wavelength from the SED template, interpolating if necessary between model points and applying color-evolution corrections, to produce three luminosities, one per band.

I then convert these luminosities to fluxes using the calculating luminosity distance for the given redshift and applying the correction to observed flux according to the prescription in section 2.3.3. Finally, I add the random Gaussian deviation to each flux value corresponding to the typical flux error for the observed population in each band, and store the result in a source structure. This is performed for all three observation bands simultaneously, before any cuts are made based on lower flux limits. When each source is simulated, it is discarded if it would not make it into the survey, after it is added to the internal number counts calculation; I cannot compare that source to my band matched, observed sources if I could not observe it in the first place. After generating such a survey, the fitting routine takes over and compares the survey to the observed population, determining a goodness of fit statistic.

The final saved survey comes in the form of matched flux densities, and is directly comparable in form to the input survey. For each simulated survey, the program also computes observed source counts for each band if a source is detected in that band, regardless of whether it was detected in the other two bands. These are de-evolved counts calculated in the same manner as Clements et al. (2010), with the same binning, for comparison. These are discussed at greater length in section 4.

3 Monte Carlo Survey Simulation

Given a means by which to simulate a galaxy population, I want to be able to find optimal parameter values which produce as close as possible a survey identical to that observed. As described in Acquaviva et al. (2011), many astrophysical processes are described by multivariate models which are almost invariably highly degenerate in different parameters spaces, and thus model fitting through typical chi-square

minimization methods would almost always be highly dependent on initial input parameters and could inevitably only find local minima; chi-square methods do not allow us to sample the entirety of a model space, and thus in using them in our higher dimensional models one discards much of the information that might be gleaned by tracking parameter permutations; often exploring the entire model space illuminates flaws or successes of given models.

This is particularly true for models of large galactic populations; individual galaxies show such widespread variation that to describe their SED even partially requires specification of mass, luminosity, morphology, composition, age, and other such parameters. Adding to this complexity, if we want to truly reproduce observations of such populations in regimes which lie far away from the optical, we sacrifice much of the resolution which allows us to discriminate between different types of galaxies, such as particularly defined spectral features (Kirkpatrick et al., 2013). Monte Carlo simulation techniques allow us to sample the entire range of our models, explore degeneracy between all of our parameters, and determine best fit parameters and associated uncertainties from likelihood distributions.

In this section, I walk through the development of the fitting statistics, and the implementation of the Monte Carlo fitting technique I employ for this program. I will discuss Markov Chain Monte Carlo methods in particular as my fitting methodology and the algorithm I use to sample the parameter space. I will then discuss the implementation of the program as a whole, as well as how the fitting methods interact with simulation methods.

3.1 Fitting Statistics

With sets of simulated and empirical observations, we now need a robust, methodical procedure by which we can establish the correlation between the two samples. Most programs tend towards one dimensional data representations, as these are much easier to handle. Due to the often rich nature of the color-color space as a means to compare galaxies of different magnitudes and types, I decided to employ a two-dimensional color density metric. This data representation allows me to clearly identify multiple populations where present, as well as trends at both sides of the range of interest, regardless of the intrinsic brightness of a given galaxy.

The diagnostic produced from the simulated data is a two dimensional histogram, with the $250\mu m/500\mu m$ color as the x axis and the $350\mu m/500\mu m$ color as the y axis. There are in fact three unique color pairs that can be produced from fluxes in three distinct bands. This combination was chosen because the $500\mu m$ flux has a larger statistical error than the other two bands. This combination ensures that sources with largely biased $500\mu m$ fluxes will still exhibit the same general trend, but lie either lower or higher on the trend than they should be. This is not important for fitting purposes, as I add in simulated noise at comparable levels to that present in observation, which should serve to simulate any systematic error due to such noise. It does allow for better identification of potential multiple populations by eye within a given sample and minimizes the visual noise obscuration such that overall trends are minimally disturbed. An example of how my SED templates fill out such a space is shown in figure 6.

The major hurdle to working with higher dimensional metrics like this is the over abundance of zero-valued histogram bins. Higher dimensional spaces tend to become largely empty as data sets become smaller, thus making the traditional goodness of fit methods harder to implement, as many behave poorly at small values and are undefined at 0. To help counteract this problem, I employ the bin size selection method outline in Scott (1979), where the ideal bin size for a sample of size N with standard deviation σ is:

$$\Delta\alpha = \frac{3.49\sigma}{\sqrt[3]{N}}$$

This formula was experimentally found by Scott (1979) to best balance the need to minimize whitespace and statistical error while maintaining as much resolution as possible. An illustration of how this bin size mitigates these concerns can be seen in figure 7.

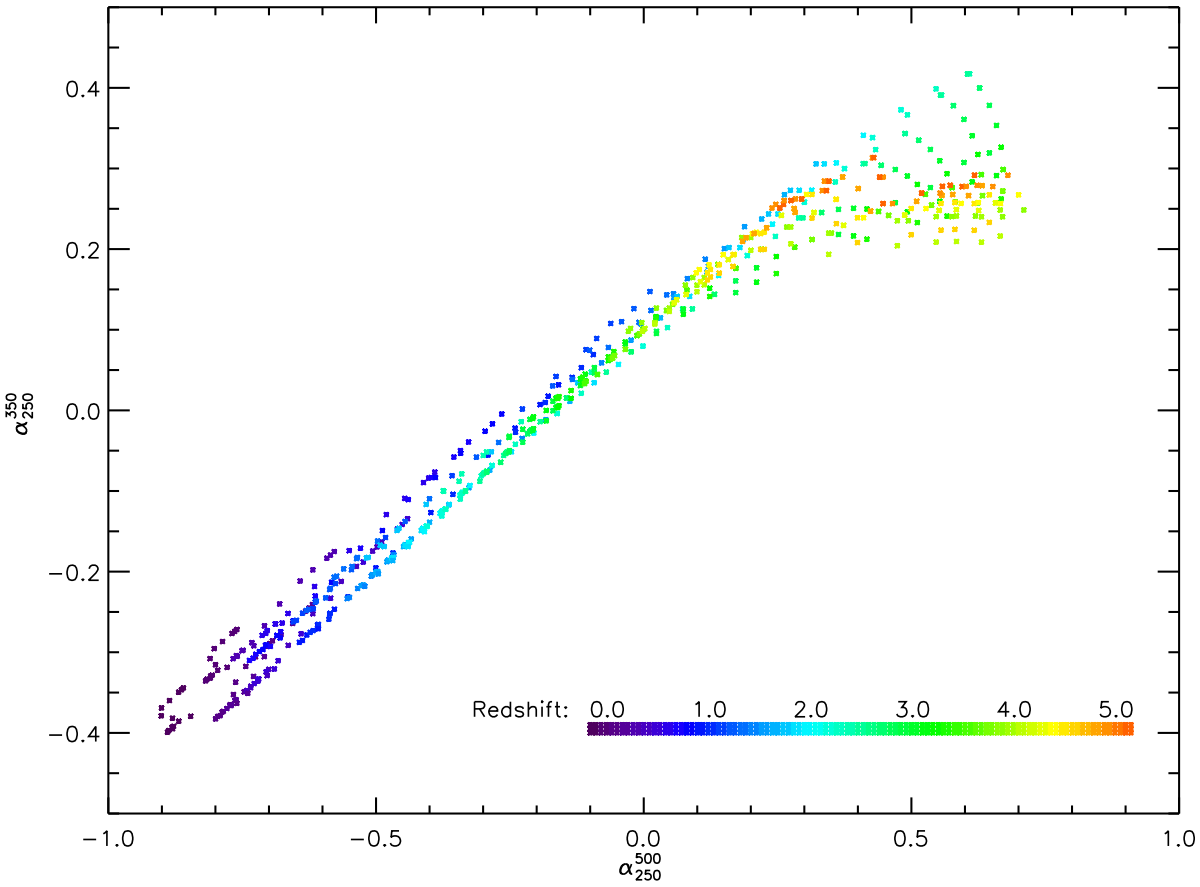


Figure 6: The spectral color pairs produced by the entire library of models for $0 < z < 5$, without added noise. Here we can see the general trend associated with redshift, however also a degeneracy between intrinsic luminosity and redshift along the trend line, as well as some natural dispersion due to natural variation in template form through the far-infrared region.

The goodness of fit statistic I employ is the popular χ^2 statistic in its non-reduced form, although a few minor alterations needed to be made to the general formula to obtain a statistic useful for our particular diagnostic, as well as to eliminate the "0" problem which plagues the traditional formula. To begin with, I assign the histogram bins standard Poisson errors, using tabulated values for bins less than 100, and otherwise assigning the bins $\sigma = \sqrt{N} + 1$ (Bagaud et al., 2005). Starting from the generalized chi-square formula:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{\sigma_i^2}$$

I get my fitting statistic by adding model and observation bin values in quadrature and substituting this for generic observation variance:

$$\sigma_i = \sqrt{\sigma_{E_i}^2 + \sigma_{O_i}^2}$$

By employing Poisson errors, which are non-zero for bins with zero values, I ensure that the χ^2 formula is

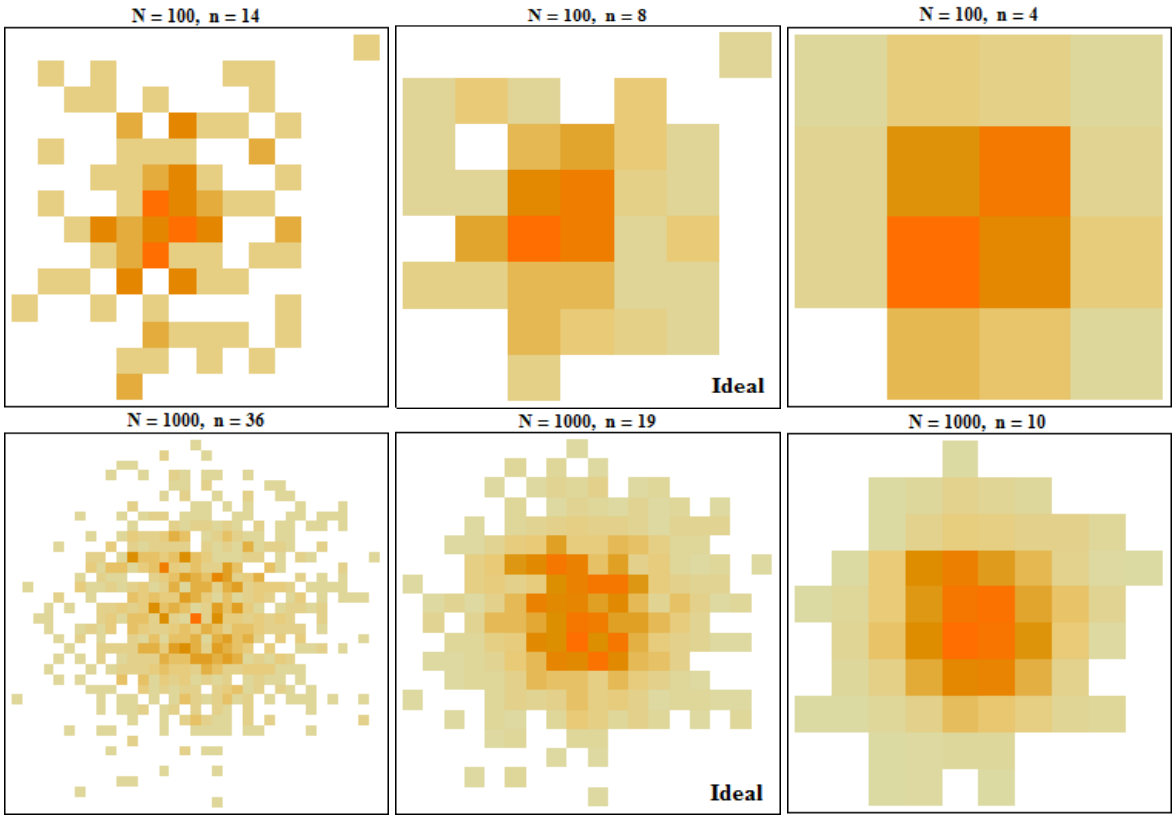


Figure 7: Above is an illustration of the effect of varying bin-width and sample size on the quality of the histogram produced. The middle column shows histograms sized according to the Scott (1979) formula, while the left hand column shows histograms with half this bin width, and the right hand column twice this binwidth. The histograms on the left are too fragmented and, considering that one expects relatively continuous functions in such plots, contain too much whitespace in what should be high density areas. The histograms on the right solve this problem, but sacrifice detail, reducing the resolution with which models can be compared against one another. The ideally binned histograms balance these concerns well; they minimize whitespace in high density regions while retaining the bulk of the data, and produce relatively continuous trends.

valid for all integer bin values. The resulting χ^2 formula then becomes, for the two-dimensional histogram:

$$\chi^2 = \sum_{i,j=1}^{n,n} \frac{(O_{i,j} - E_{i,j})^2}{\sigma^2_{E_{i,j}} + \sigma^2_{O_{i,j}}}$$

where i and j refer to column and row indices, and n is the side length in number of bins of my two-dimensional histogram. This approach ensures that proper histogram errors are taken into account, especially in cases where bins are near zero, and gives more weight to areas of higher density, as they have lower comparable error to those bins with only a few sources.

3.1.1 Auxiliary Diagnostics

The color-color diagnostic, while robust, does discard some vital information produced by our simulation, namely redshift distribution and number counts[10]. While these are not fundamental to every survey, when

[10]In this paper, I employ differential number counts, defined (as in Clements et al., 2010) as: $(dN/dS) * S\,[Jy]^{2.5}$

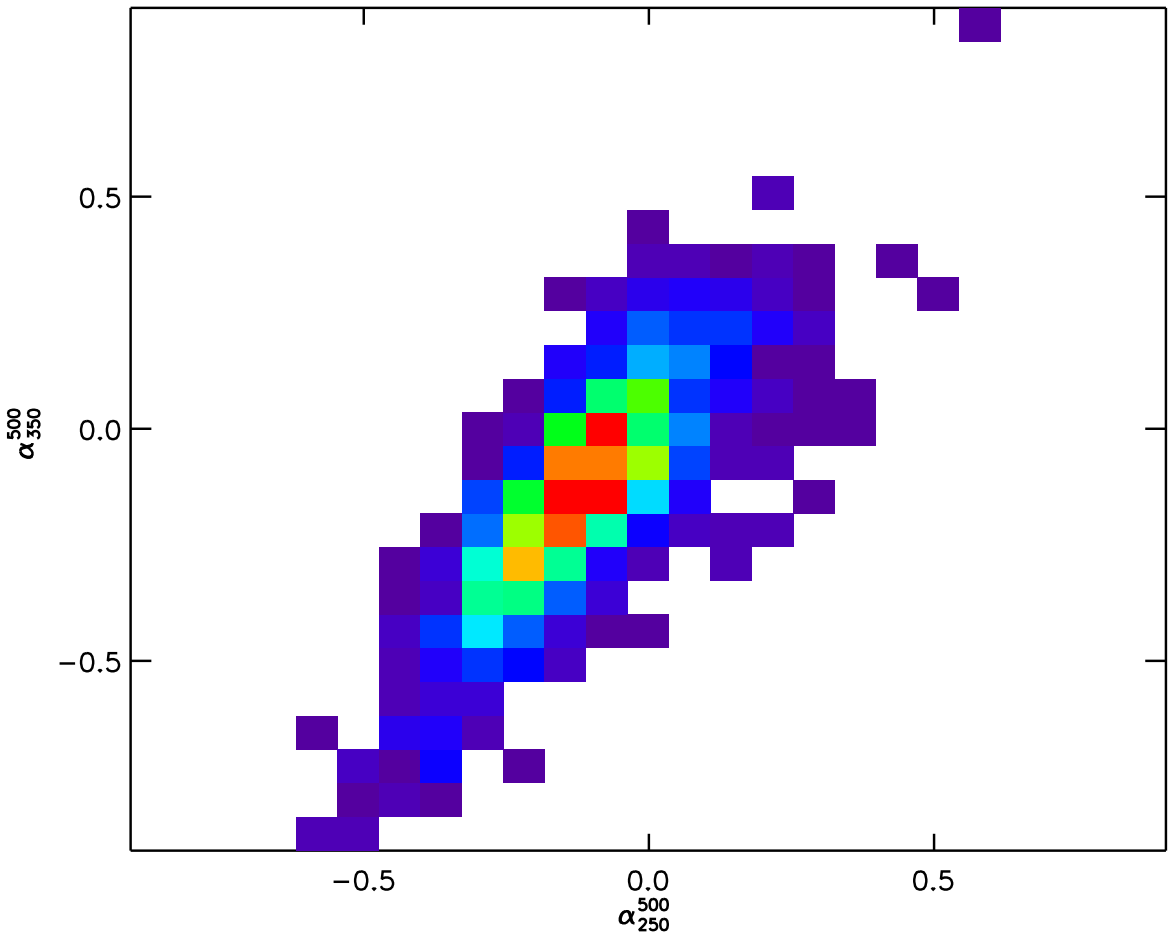


Figure 8: Example of how the example data set discussed later fills out such a diagnostic plot.

available they provide powerful secondary goodness of fit constraints by further constraining the model to produce realistic distributions in both redshift and flux density spaces. As an example, some of the fits I show later would prefer a higher redshift cutoff, however, I know from previous studies that the redshift evolution should stop around redshift 2. While I now make this constraint manually, given an input redshift distribution, I could produce a more constrained fit. As these are not the primary diagnostic available for any multi-spectral survey, they are not currently used, but are produced for comparison purposes; such constraint will probably be implemented in future versions.

3.2 Markov Chain Monte Carlo Fitting

MCMC algorithms consist of random deviations from an initial guess, and an algorithm which decides whether to accept these deviations as acceptable guesses. This is usually based upon the difference between the goodness of fit these parameters produce and the best goodness of fit obtained during the parameter space search; if the goodness of fit is better, it will automatically be accepted, otherwise the likelihood of acceptance will fall off exponentially. A detailed discussion of such algorithms as applied to cosmological parameter fitting can be found in Dunkley et al. (2005), and a more theoretical discussion is presented in Ross (2013).

3.2.1 Sampling

The most popular sampling algorithm, and arguably the simplest, is the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970). This algorithm is built off of four generalized logical steps (Johnson et al., 2013):

1. Draw a random deviate from some proposal distribution to displace each parameter from its current value, generating a next proposed step:

$$X_{i+1} = X_i + X_R(X_i)$$

2. Calculate the acceptance probability, which is 1 if the proposed parameters produce a better fit, or less than one (but non-zero) otherwise:

$$P_{acc}(X_{i+1}|X_i) = min\left(1, \frac{P(X_i)}{P(X_{i+1})}\right)$$

3. Generate a uniform random variate, and accept the guess if it is less than the acceptance probability:

$$\text{Accept}(X_{i+1}|U, P_{acc}) = \begin{cases} \text{True} & U < P_{acc} \\ \text{False} & U \geq P_{acc} \end{cases}$$

4. Repeat this process until you satisfy some end condition, to be determined based upon fitting criteria.

From here, the creation of a specific Markov Chain Monte Carlo requires either analytical or numerical prescriptions for each of the four functions; the method is made more robust not by complicating the above steps, but by the creation of robust algorithms for acceptance probability and random sampling. The most common algorithm for generating random deviates to create a new guess is by adopting a Gaussian proposal distribution for each parameter, and specifying a unique variance for this proposal distribution based on prior knowledge of the variable's acceptable range (Johnson et al., 2013):

$$X_{R,j} = N(\sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left[\frac{-x^2}{2\sigma_p^2}\right]$$

This variance, in the simplest case, will be chosen by the user at run-time, although a more robust algorithm would change it dynamically to ensure that the entire space is sampled. The trade-off is completeness of sampling versus resolution of sampling for a given time period; this depends on how confident the user is about the proposal distribution variance. A more robust and efficient algorithm is the combination of a Gaussian and an exponential Acquaviva et al. (2011):

$$X_{R,j} = gN(\sigma_j) + (1-g)\exp[-x]$$

which I consider implementing; the Gaussian from above is the special case of $g = 1$.

For our acceptance probability function I adopt an algorithm based on an analogy to a thermal system, using the Boltzmann factor; I consider the χ^2 for a model produced by the parameters as the "energy" of the system, and thus the probability of a system occupying such a state is given by

$$P(E\, or\, \chi^2) \propto \exp\left[\frac{-E}{kT}\right] = \exp\left[\frac{-\chi^2}{T_s}\right]$$

where of course T_s is not strictly a temperature, as χ^2 is a dimensionless quantity. The point here is just to show that, according to our algorithm, we can define a "heat" of our Monte Carlo, where a hotter run corresponds to greater probability of choosing χ^2 values further from the best value and thus a more thorough sampling of the space. The balance is choosing a temperature which allows for a small number of "worse" guesses to be accepted, allowing the Monte Carlo to randomly move about the best parameters, while making sure enough are rejected such that the Monte Carlo will still tend towards the best parameter region.

The starting value of this temperature has thus become an additional user-defined parameter, with a default value of 100, though this is adjusted by the program subsequently to obtain the correct acceptance rate, as discussed below. A possible future approach is simulated annealing, in which the simulation begins at a high temperature, samples the entire space, and then the temperature is lowered and the simulation samples a smaller space about the best parameters found in the high-temperature case. This strikes a balance between complete space exploration and the need to estimate parameter uncertainties, and sample the space about the best parameters very accurately.

3.2.2 Acceptance Optimization

The benefit of using MCMC methods is that one is able to sample all relevant areas of our parameter space by randomly accepting parameter values which produce worse fits than those one has previously found. For this process to be efficient, and maximize one's chances of both isolating the true global best-fit parameter vector while sampling the space close to this best fit model efficiently, one needs to optimize the rate by which one accepts such random guesses. Johnson et al. (2013) and Dunkley et al. (2005) cite an ideal acceptance rate of 25%, and they achieve this through variations upon the basic theme of comparing the mean acceptance rate to the ideal rate, and adjusting chain parameters accordingly.

I have chosen to implement a simple feedback mechanism to adjust the temperature of the metropolis-hastings sampler to maintain the acceptance rate within the desired range. I simply track all acceptance and rejections and compute the average acceptance rate at regular intervals, incrementing or decrementing the temperature according to whether the computed value is less than or greater than the desired value. In order to isolate the rest of the simulation from the "burn-in" period, the period during which the simulation reaches this acceptance rate which is marked by much higher acceptance than desired, I employ a separate burn-in phase during which the acceptance rate is optimized and the space is explored for the initial best fit parameters. This is not tracked like the chain during simulation and is discarded after it executes.

3.2.3 Convergence Testing

The subject of convergence testing is the most controversial aspect of Monte Carlo implementation, with many reviews coming to vastly different conclusions about the manner in which convergence should be evaluated (see e.g. Cowles & Carlin, 1996; Dunkley et al., 2005). The largest variation is seen on the basis of sampling method, however even within the most common methods, Cowles & Carlin (1996) find upwards of 6-10 methods each with strengths and weaknesses dependent on degeneracy of parameters and shaped of model likelihood distribution. In the case of the Metropolis-Hastings sampling algorithm, a chain or set of chains is generally said to converge when the ratio

$$r = \frac{\sigma_x^2}{\sigma_0^2}$$

descends below a specified value, tied to the requested precision to which the parameters are to be determined. Here σ_0^2 is the variance of the prior distribution, and $\sigma_x^2(N)$ is the variance of the sample means of independent chains with event length N. A chain is said to converge when r evaluates to be less than a given

constraint for all fitted parameters, where the constraint on r comes from the precision to which a parameter is desired.

The problem this metric causes is that we are required to have knowledge of the prior variance, which is one of the unknowns we is trying to determine. The major convergence tests represent means of approximating this ratio without knowledge of the underlying distribution variance. Convergence for a single chain can be determined either through spectral means or through repeated simulations for a given model (Dunkley et al., 2005; Allison & Dunkley, 2014), while convergence for a set of independent chains can be computed by comparing variance between chains to variance within chains. Specifically, Brooks & Gelman (1998) define the ratio $R \sim 1 + r$ which can be calculated from m independent chains of length $2n$ fairly simply from desired confidence intervals. For the confidence interval $100(1-\alpha)\%$ I compute the following from the last n chain iterations:

1. Calculate the empirical $100(1-\alpha)\%$ interval for each of the m chains from the last n chain links. I will call the mean of these intervals $\mathrm{CI_m}$.

2. Calculate the empirical $100(1-\alpha)\%$ interval for the entire set of mn observations (that is, for the combined n links of all m chains). I will call this $\mathrm{CI_T}$.

3. Calculate R from the above as
$$R = \frac{\mathrm{CI_T}}{\mathrm{CI_m}}$$

While this is not the most efficient or strictly most robust of the convergence tests in the literature, it is by far the easiest to implement, most intuitive to use, and behaves better than most criteria when the prior distribution is not a nicely behaved function with a well defined variance. By calculating empirical intervals, no assumption is made about the shape of the prior distribution, whereas other methods generally assume a Gaussian form (Brooks & Gelman, 1998).

As an example, if we want a set of chains which approximate the prior distribution to within a 95% confidence level, we set $r = 0.05$ and accordingly $R = 1.05$ as our convergence limit. The simulation is considered converged if all parameters have an R values below this limit.

3.3 Implementation

The full functionality of this MCMC fitting algorithm and the luminosity function based simulation approach are implemented as a combination of an IDL graphical user interface, for input and output, as well as the core MCMC and simulation code implemented as a C++ program, modularized to separate the different function types into different classes. In the following two subsections, I give an overview of the structure of both parts of the code, including how the functionality is divided within the C++ code as well as the inputs and outputs required to operate the entire package.

3.3.1 C++ Fitting Code

The C++ portion of this software package is built to run front to back without user input, utilizing the fits files for input and output operations. It has been built with a focus on modularization, with each of the primary functions (i.e. fitting, simulation, diagnostics, cosmological functions) implemented either as a class or within its own file with similar methods. This was done to maximize the degree to which the code may be modified to suit new types of analyses. For example, to modify the form of the luminosity function with a similar number of parameters, one simply needs to modify one function within lumdist.cpp, but all other declarations remain identical, and it is not necessary to dig through thousands of lines of code to find the proper modification. If someone merely wanted to use the Markov Chain Monte Carlo fitting

functionality, they could take the mc_util header and source code with the main loop and implement it to fit and entirely different function.

The C++ routine takes three or four arguments, specifying the fits files which contain the SED templates, observations, and model information as inputs. The SED template file contains all SEDs and associated luminosities, the observations fits file contains a table of matched fluxes, with a header specifying the various properties of the survey, and the model fits file contains the parameters and associated properties of each parameter, as well as information about how to run the fitting portion and the various other settings seen in the widget. Once created, these inputs can be passed to the C++ program repeatedly, and if the fourth argument, the output file, is changed, the output of successive simulations will be stored in successive files. If no fourth argument is specified, the default output file name is used.

The fitting functionality, including the Metropolis-Hastings acceptance, acceptance rate optimization, and the chain tracking and convergence testing are contained with two classes, called "MetropSampler" and "MCChains". The first class stores the results of all previous trials independent of chain and modifies the temperature of the metropolis algorithm if its falls outside of the acceptable range, which is custom set by the user to be a certain percentage above or below the ideal acceptance rate. For this iteration, I have determined our ideal acceptance rate to be 25%, as stated above, but have our acceptable range lie between 20% and 30%. If the acceptance rate falls below 20%, the temperature is incremented, or if the acceptance rises above 30%, the temperature is decremented. These ranges are fully adjustable through the user interface. I experimented with various feedback approaches to adjusting the temperature, however all of these approaches greatly increased time for the chains to converge.

The MCChains class is responsible for tracking the parameter values and acceptance information for each iteration of each chain, and computing convergence when requested by a class method. Each time the convergence is calculated, the result is stored and an internal counter is incremented such that all of the convergence tests can be saved at the completion of the fitting process. The MCChains class also keeps track of the best simulation iteration, such that at any point the program could recover and return to the best performing parameter values encountered during the simulation. This class also has a dedicated "save" method to allow it to write the chains and convergence information into fits extensions for later plotting and analysis.

The "simulator" class is responsible for running the full simulation, and aside from the luminosity function class, is the only of the model specific classes found within the main program loop. This class manages instances of hist_lib (the class which generates the fitting diagnostics) and model_lib (the class which handles the SED library), and uses knowledge of the location of the lumfunct class storing the luminosity function data to perform the full simulation as described in section 2.4. This class takes the various fits files as input from the main function, initializes its classes, and waits for the "simulate" command. It then runs the simulation and returns a structure containing information about the simulation, most notably a goodness of fit statistic. It is also responsible for saving the two-dimensional histograms for the best fit model, as well as the sizing of those histograms as described earlier in this section.

The main function reads the various initial parameter values and simulation variables from the fits files, instantiates the various classes, and runs the fitting. We have split the fitting into two main parts, the burn-in phase, where the chains are set to start randomly throughout the parameter space, and the main fitting phases, where the chains are run until convergence is reached. The burn-in phase is run for a small number of runs, or until we reach the target acceptance rate; it's function is to help calibrate the metropolis-hastings class to the range of chi-square values produced by the model. After completing, we set the initial chain positions close to the best guess from the burn-in phase, and run the main fitting process, checking for convergence at regular intervals, and continuing to anneal the metropolis-hastings algorithm for a brief time towards the beginning the fitting process. The periods for which each of these phases are run can be set in the user interface.

3.3.2 Graphical User Interface

While C may be efficient for fitting, I employed IDL for the user interaction as well as plotting functionality due to the wide array of available plotting utilities. The user's interaction with the program is mainly through the widget interface built with IDL, which is responsible for changing input parameters, and displays the SED library for the user to verify its accuracy before the start of fitting. The opening screen for the IDL widget is seen in figure 9, and a panel displaying additional sections, accessed via the menu bar, is seen in figure 10.

Upon running the simulation, the IDL widget is suspended until the simulation completes, and the progress of the simulation and fitting is output in the terminal window from which the widget was initially spawned. On completion of the program, all output plots are first created as postscript, and then the simulation output is generated, including luminosity function, redshift distribution, number counts, and the observed, model, and residual histograms from fitting. The widget, when prompted, will also display the model parameter space explored by the MCMC fitting routine, and the likelihood spaces for each pair of parameters as well as best fit values for the parameters. Convergence and chi-square plots are displayed along the left hand side of this panel. The plots for the last run simulation can always be displayed from the main widget panel as well. Both of these output panels will be presented in the next section, when results of the program are discussed.

I am also working on the creation of a python widget, which will replicate some of the basic functionality of this IDL widget, but operate at a more command-prompt based level, with minimal interactive functionality. I feel that having both options will open up this program to more users, due to the progressively growing popularity of python in astronomy and the fact that it can be downloaded for free, whereas IDL is proprietary.

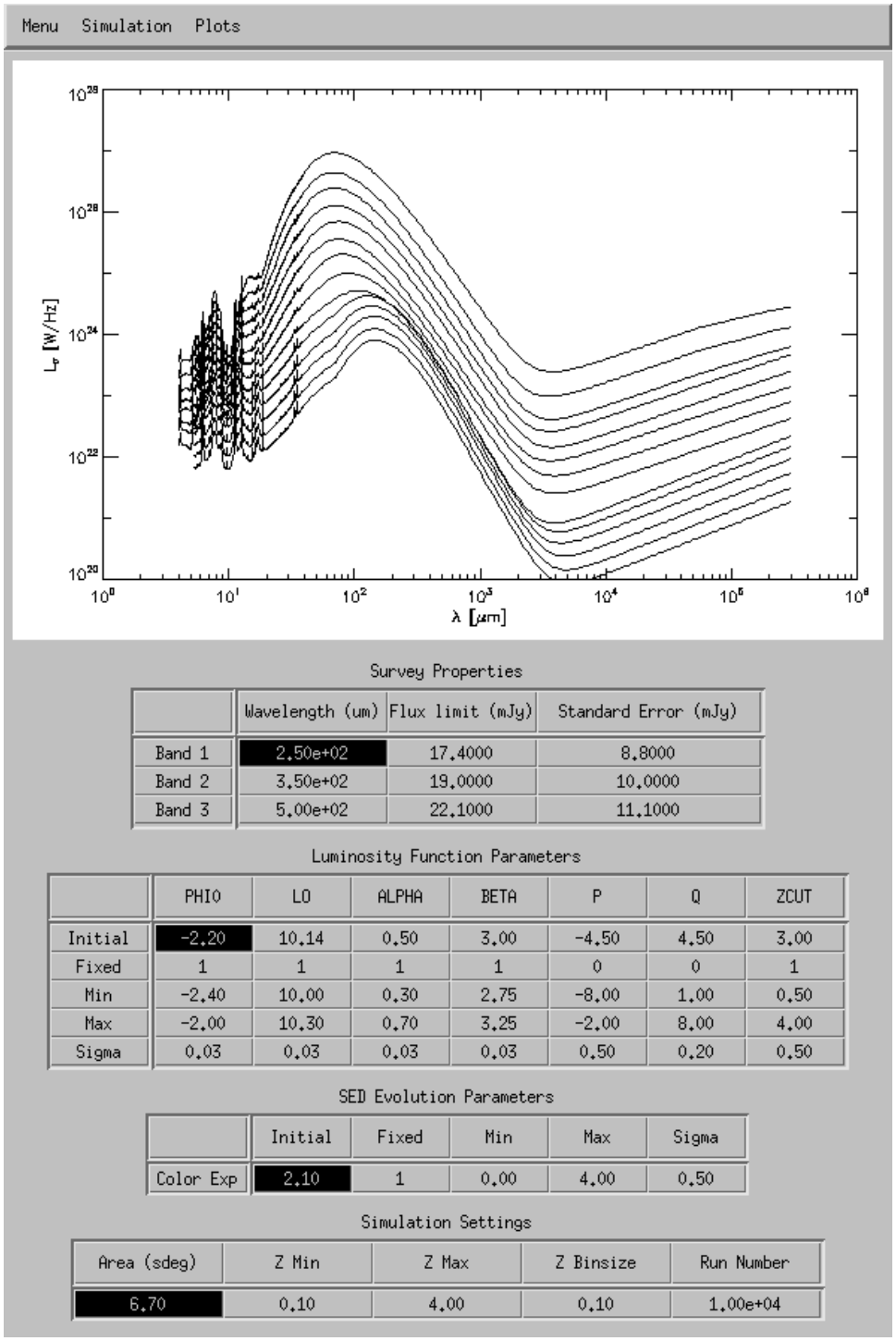
4 Application: *Herschel* SPIRE Sources

To test the fitting program, I used it to analyze sources from a Herschel SPIRE (Pilbratt et al., 2010; Griffin et al., 2010) survey of the Spitzer First Look Survey (FLS) field taken from the HerMES survey (Smith et al., 2012). In this section, I will show examples of fitting functionality applied to this data set, and discuss what the results tell us about the model and fitting functionality.

4.1 HerMES Far-Infrared Sample

The sample itself was, as mentioned, drawn from the first data release of the HerMES survey, a program conducted with the Herschel SPIRE instrument to take deep observations of various deep fields in 250, 350, and 500 micron band passes. I obtained the data separated by observed wavelength, and used the beam-size for the instrument from Smith et al. (2012) (full width half maximum (FWHM) of 18.15, 25.12, and 36.6 arc seconds respectively) to match sources across all three bands. Sources were considered a match if they lay within 75% of the combined FWHM for both frequencies being compared. The 500 micron observations only contained about 1400 sources, of which approximately 1000 had matches in the other two bands, and thus the testing sample contains around 1000 matched galaxies with measurements in all three bands. The survey covers an area of 6.7 square degrees. The observation settings used to simulate this sample were taken from the companion paper for the survey, Smith et al. (2012), and the flux limits correspond to the 50% completeness levels, while the band error corresponds to the overall (systematic and instrumental) error found for each band within that paper. The observed trend within our fitting diagnostic was previously presented in figure 8.

For the local luminosity function parameters, given that the local luminosity function is highly constrained, I took the values from Negrello et al. (2013), a local analysis of *Planck* sources at slightly higher



	Wavelength (um)	Flux limit (mJy)	Standard Error (mJy)
Band 1	2.50e+02	17.4000	8.8000
Band 2	3.50e+02	19.0000	10.0000
Band 3	5.00e+02	22.1000	11.1000

	PHI0	LO	ALPHA	BETA	P	Q	ZCUT
Initial	-2.20	10.14	0.50	3.00	-4.50	4.50	3.00
Fixed	1	1	1	1	0	0	1
Min	-2.40	10.00	0.30	2.75	-8.00	1.00	0.50
Max	-2.00	10.30	0.70	3.25	-2.00	8.00	4.00
Sigma	0.03	0.03	0.03	0.03	0.50	0.20	0.50

	Initial	Fixed	Min	Max	Sigma
Color Exp	2.10	1	0.00	4.00	0.50

Area (sdeg)	Z Min	Z Max	Z Binsize	Run Number
6.70	0.10	4.00	0.10	1.00e+04

Figure 9: Main input panel and the screen that appears when the IDL wrapper is launched. At the top are the SED templates. In the tables below the survey, LF, color evolution, and simulation parameters are shown and can be modified. The top left buttons are menus containing options to save parameters, quit, run, adjust additional settings, or display the output panels.

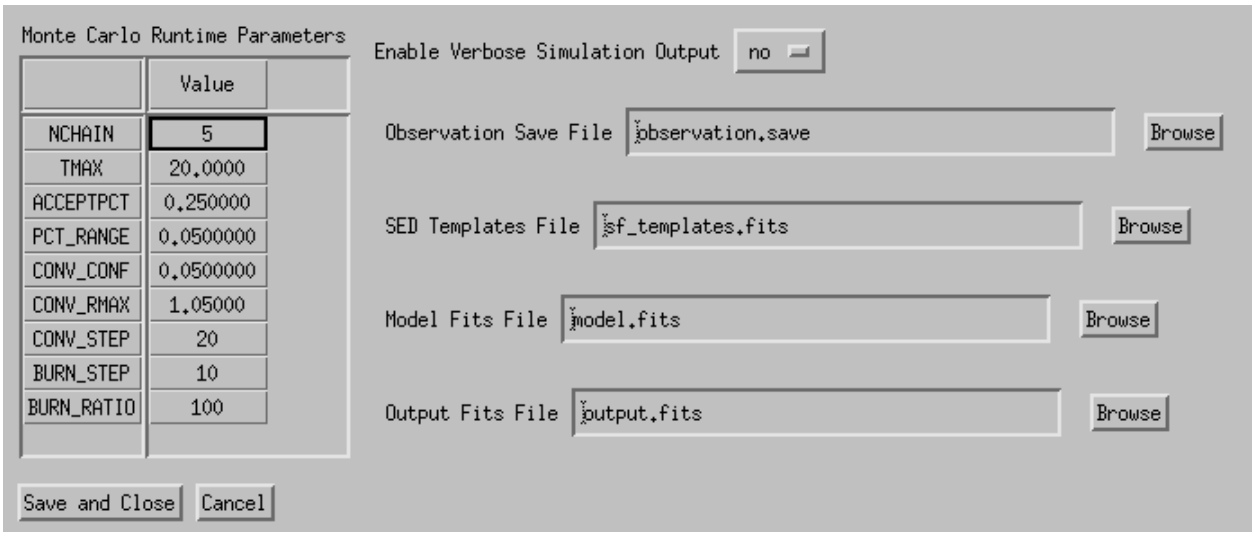


Figure 10: Additional settings parameter showing some of the additional parameters discussed in the earlier fitting section.

wavelengths, which should still be approximately accurate at our selected wavelengths. The initial redshift cut-off was a choice based on the findings of Marsden et al. (2011) and others, although I did attempt to fully fit this parameter as I did all other evolutionary parameters. I employed the Herschel ATLAS counts of Clements et al. (2010) for the number count comparisons, seen in figure 15. All of these default parameters can be seen in the main tables within the main widget screen in figure 9.

4.2 Fitting Results

Results from four different fitting runs can be seen in the figures to follow, with the main parameter space explorations in figure 11, fitted luminosity functions with evolution in figure 14, number count comparisons and output redshift distribution for our best-fit model in figure 15, and the diagnostic survey color-color plots and residuals in figures 16. The four fits were performed with the same input parameters, and I varied the number of parameters fit in each iteration to illustrate the range of fits which can be performed, and illustrate the effect of adding more parameters to a fitting analysis in terms of fit quality and MCMC performance. My most constrained and best-fit model was that obtained by varying only p and q, using best-fit parameters for redshift cutoff and color evolution obtained by the higher dimensional fits. The widget output plot containing the survey metrics can be seen in figure 12, and the MCMC output can be seen in figure 13.

In each of the four-panel figures, from left to right and top to bottom, you can see fits of 1) p and q alone, 2) p, q, and color evolution, 3) p, q, color evolution, and redshift cutoff, and 4) all parameters with the exception of color evolution. In the last case, color evolution was set to a value of 2.1. All four fits achieved roughly the same minimum chi-square of between 60 and 100, thought the fits tend to be better when the redshift cut-off is held constant but the other three evolutionary parameters are fixed. Note that the χ^2 here is non-reduced, and means that in the color diagnostic, around 60 sources out of 1000 are discrepant. In the case where all but color evolution are fit, I allowed the four local parameters to vary only within the range dictated by their quoted errors, although in principle I could have allowed them to vary freely; the constraint was done to keep them consistent with the local luminosity function measurement, and allow the simulation to complete in a matter of minutes as opposed to an hour or two.

I find that the best fit parameters are fairly consistent across all fitting variations; I find, for these models,

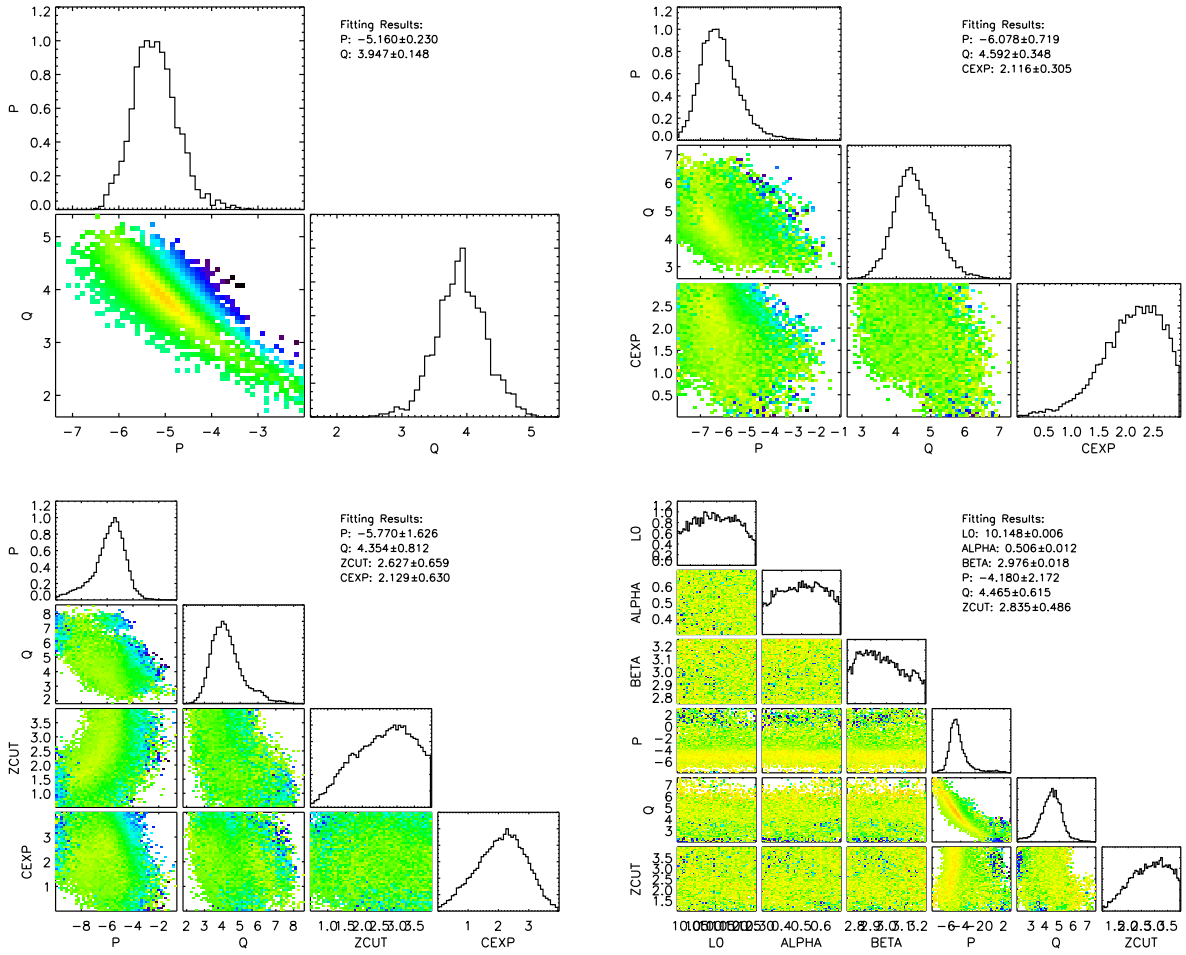


Figure 11: Fitting results for four different simulations run for this data set. Top left: fit p and q, color evolution set to 2.1, redshift cutoff set to 2.7; Top Right: p, q and color evolution fitted, redshift cutoff set to 2.7; Bottom left: p, q, redshift cutoff, and color evolution fit; Bottom right: all parameters except color evolution fit, with main LF parameters kept within range of standard error; color evolution set to 2.0.

best fit parameters as follows:

$$p = -6.0 \pm 0.7$$
$$q = 4.5 \pm 0.4$$
$$z_{cut} = 2.6 \pm 0.6$$
$$a = 2.1 \pm 0.6$$

As seen in figure 11, these numbers are fairly consistent across all fits, although at times the last two vary by slightly more. The p, q, and a as well as the p, q, z, and a fits were the best, and thus I used these fits to determine the best fit parameters.

In figure 15, I have shown the differential number counts and redshift distribution of the best fit parameters to illustrate the secondary metrics produced by the program which can be used as a sanity check on the simulation, and to help determine how well my metric constrains my model. Along with the calculated counts, I have included the counts determined by Clements et al. (2010) for comparison. These counts are

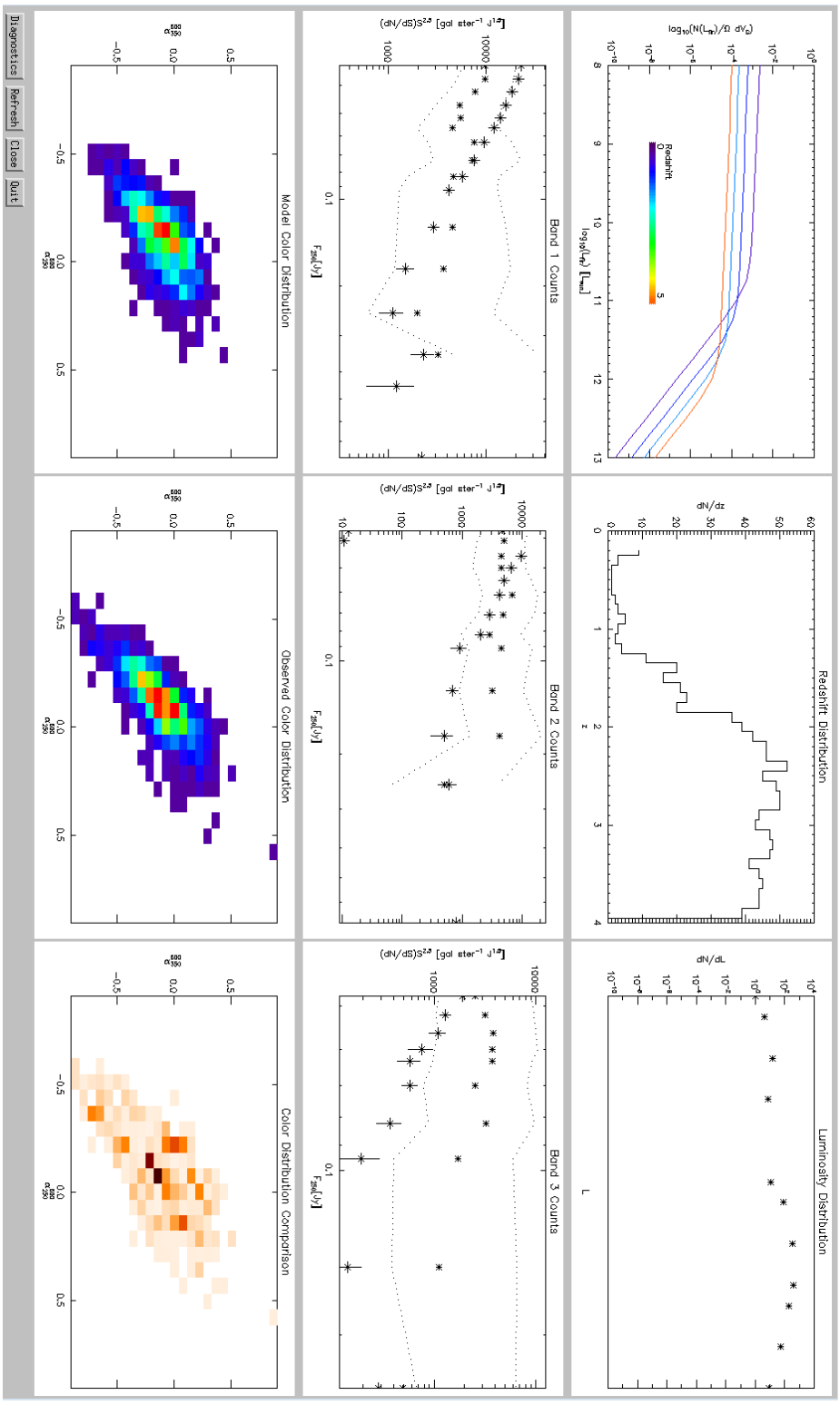


Figure 12: Simulation output corresponding to fitting only p and q, best parameters shown are those included in the text in figure 13. Top left is the luminosity function, top middle is the redshift distribution, top right is the intrinsic luminosity distribution, middle are number counts, de-evolved, and bottom are simulated, observed, and subtracted results.

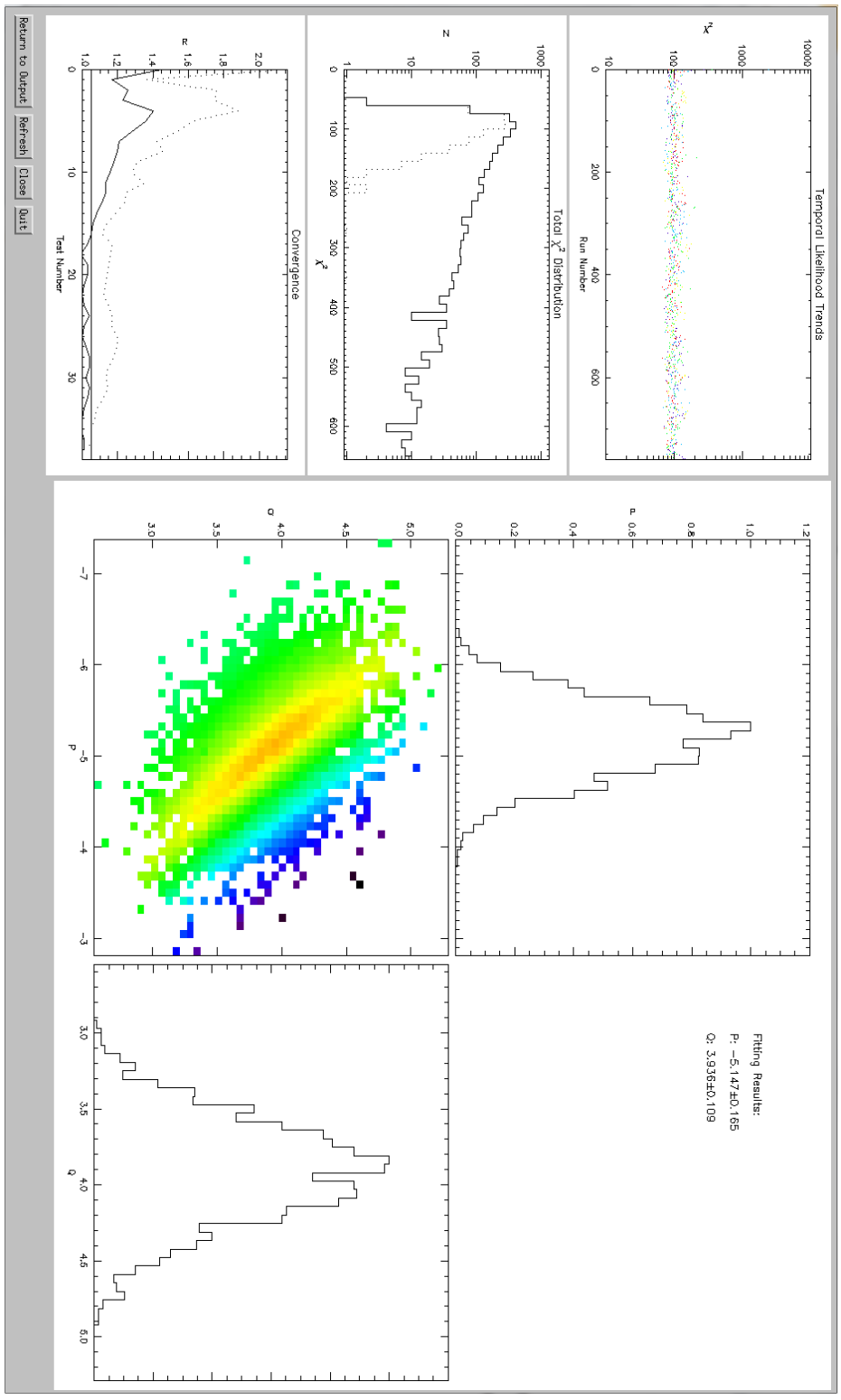


Figure 13: Simulation diagnostics corresponding to fitting only p and q. On the left are the simulation diagnostics, including (top to bottom) the temporal accepted chi-square trends, the chi-square distribution (solid is total, dotted is accepted), and the convergence for each parameter over the course of the simulation. The right panel displays the results of the MCMC fitting, including number density histograms for each parameter of chi-square above the median for the entire simulation, and a two-dimensional histogram of the p versus q color space. This panel will automatically size itself to produce plots of every combination of fitted parameters, as seen in figure 11

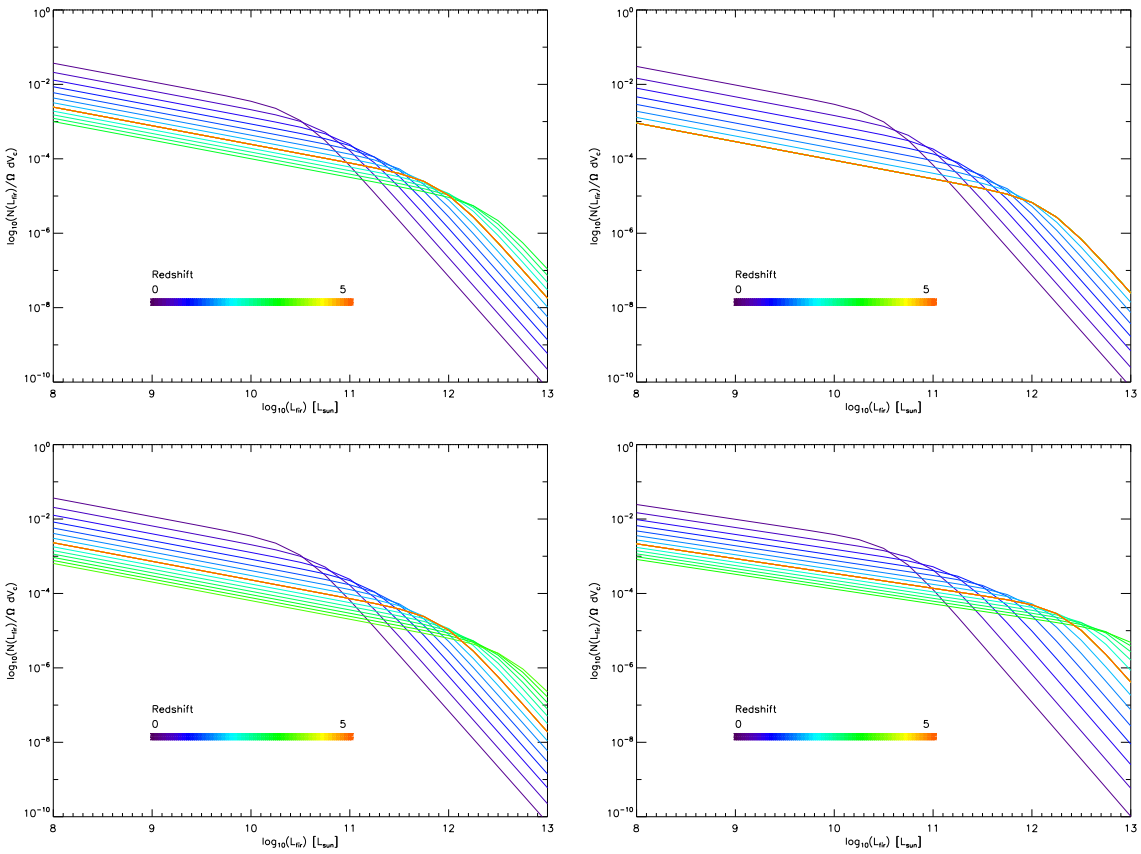


Figure 14: Best fit luminosity functions for four input parameter sets. Top left: fit p and q, color evolution set to 2.1, redshift cutoff set to 2.7; Top Right: p, q and color evolution fitted, redshift cutoff set to 2.7; Bottom left: p, q, redshift cutoff, and color evolution fit; Bottom right: all parameters except color evolution fit, with main LF parameters kept within range of standard error; color evolution set to 2.0. Here you can see the effect of extending the redshift cutoff in addition to changing the normalization and characteristic luminosity evolution in the expansion of the green trend lines.

based on a larger sample of HerMES sources, and thus I expect it to be more of an average and extend to higher fluxes than our relatively small sample. In addition to the best-fit counts, I have plotted the 68% confidence regions for the number counts computed using the stored number counts for all simulation iterations. With the rather large confidence regions included, it becomes clear that these counts are in agreement with the clements counts, if only because our model does not take counts into account directly and thus leaves them poorly constrained. The strength is that, for the 250 and 350 μm counts, the overall trend is comparable. It should be noted that the Clements et al. (2010) counts for the 500 μm are of low confidence due to the small sample size used and the high degree of incompleteness in the Clements et al. (2010) sample used to create them, and thus the larger discrepancy in this plot is not highly surprising. This being said, the faint end of the number counts do agree to as good a degree as in the other bands, which is an encouraging sign.

While the main diagnostic seems to be very well fit, for the most part, the number counts and redshift distribution are not quite as well matched. I expect, from comparison to a similar study conducted by Devlin et al. (2009), and from the results in Marsden et al. (2011), that the number of infrared galaxies should peak around redshift ~ 2, however I overpredict the number thereafter. I will discuss these considerations in more detail in the discussion.

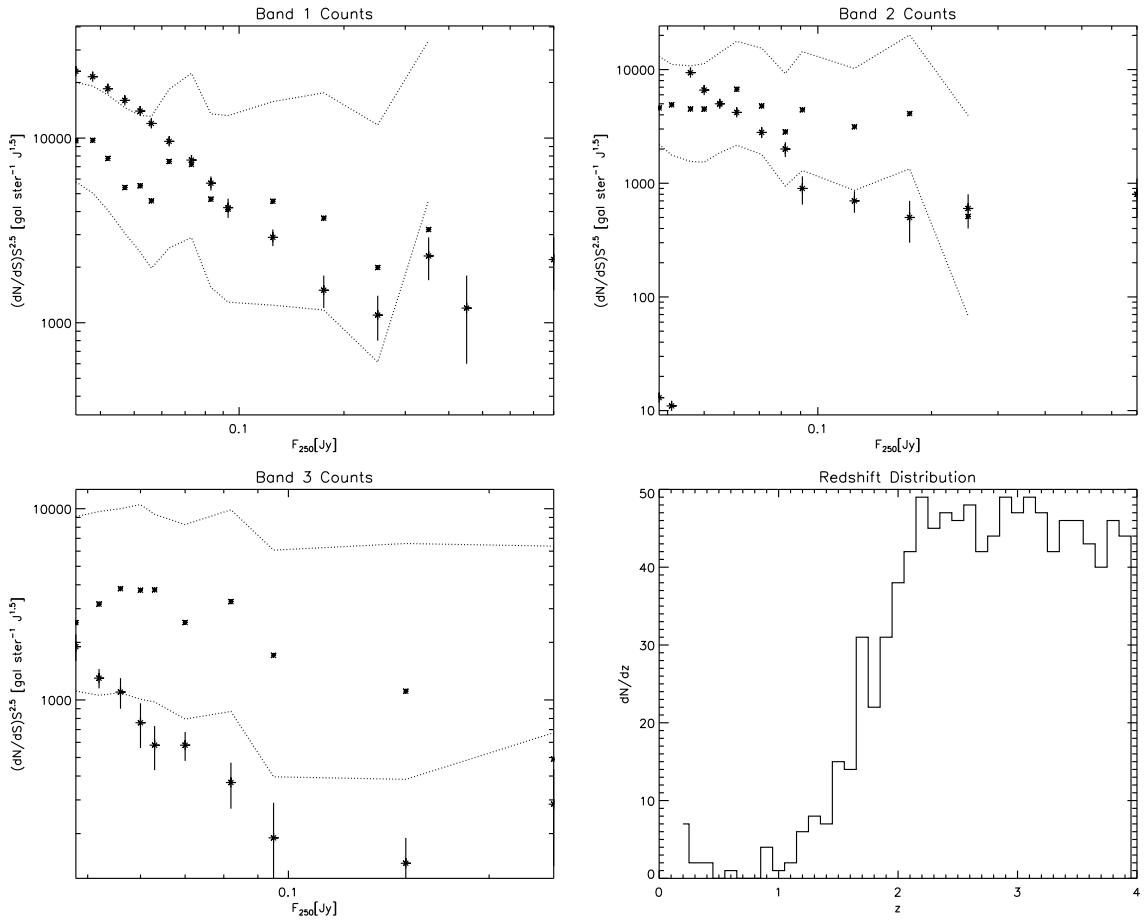


Figure 15: Number counts and redshift distribution for the simulation in which only p and q were fit. Here you see an example of how the fit produced does not entirely match other observables, as the starred data points (without error bars) do not fit the established counts produced by Clements et al. (2010) for a similar sample of Herschel sources. In the number counts plots, the dashed lines represent the 68% confidence region of the counts, which, on the contrary, does generally agree with the Clements et al. (2010) counts. On the bottom right, you also see a redshift distribution for this simulation; if I had a measured distribution, this would be an additional metric to include in order to better constrain my simulations.

4.2.1 MCMC Evaluation

One can evaluate the MCMC performance on the grounds of success of convergence, consistency of results, and quality of recovered likelihood distributions. Referring first to figure 17, you can see that the time taken for the simulation to converge (as evidenced by the number of evaluations until convergence along the x axis) is highly dependent on the number of parameters fit; this is expected from any analysis of typical MCMC trends (Brooks & Gelman, 1998; Dunkley et al., 2005). During fitting, I did find that the time taken for the burn-in sequence to complete was highly variable, and at times I would have to exit the burn-in sequence if it found itself in a region in which time per simulation was far too high. Additionally, the convergence trends here show multiple times in which parameters are close to convergence and subsequently move away from convergence. I believe this is due to an inefficient annealing procedure, and would be improved by making my procedure more robust. Dunkley et al. (2005) suggests multiple techniques for doing this which promise to increase efficiency and reliability of convergence during the main phase, if implemented. In light

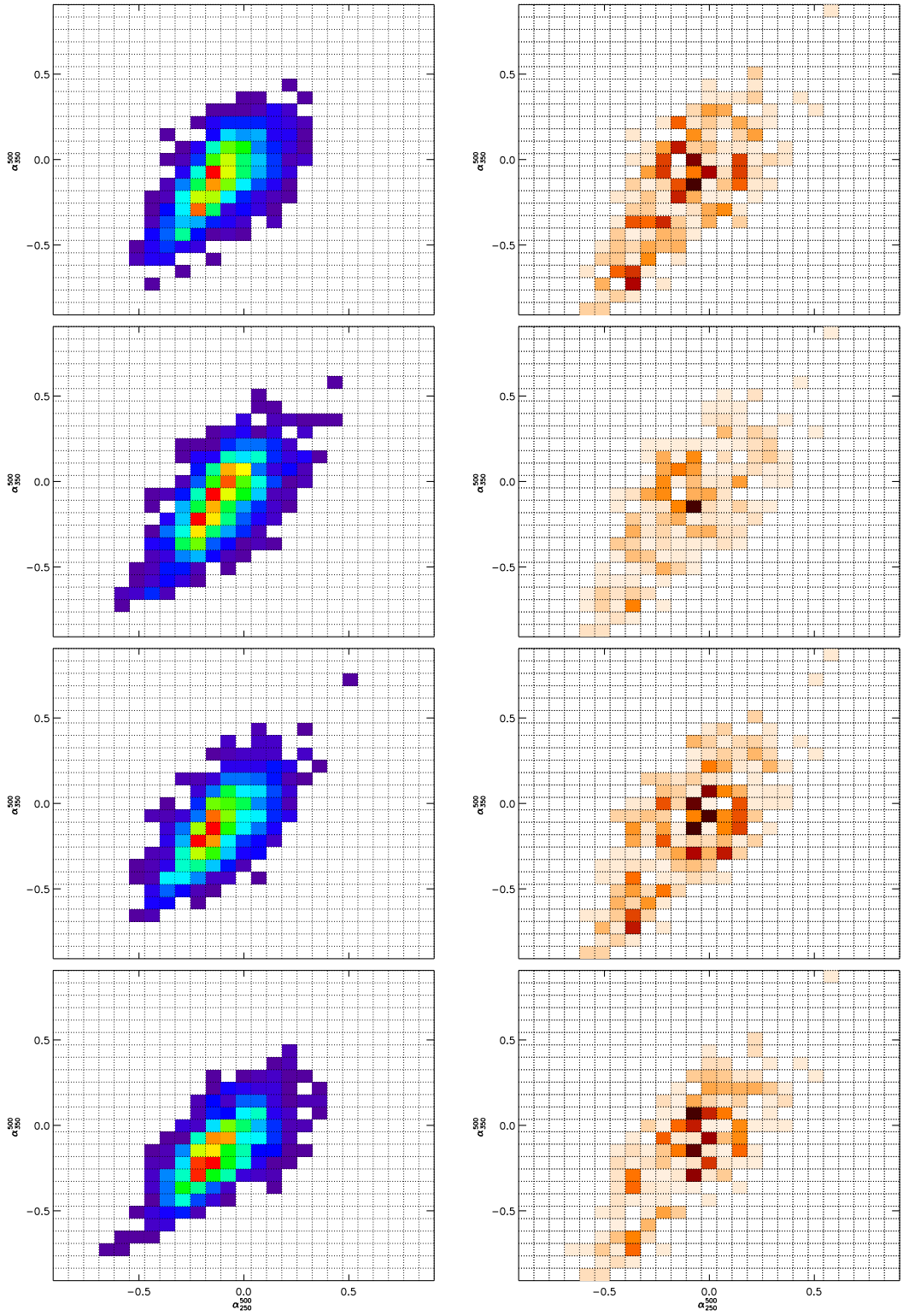


Figure 16: Model (left) and Residual (Model - Observed, right) histograms for four input parameter sets. First row: fit p and q, color evolution set to 2.1, redshift cutoff set to 2.7; Second row: p, q and color evolution fitted, redshift cutoff set to 2.7; Third row: p, q, redshift cutoff, and color evolution fit; Bottom row: all parameters except color evolution fit, with main LF parameters kept within range of standard error; color evolution set to 2.0.

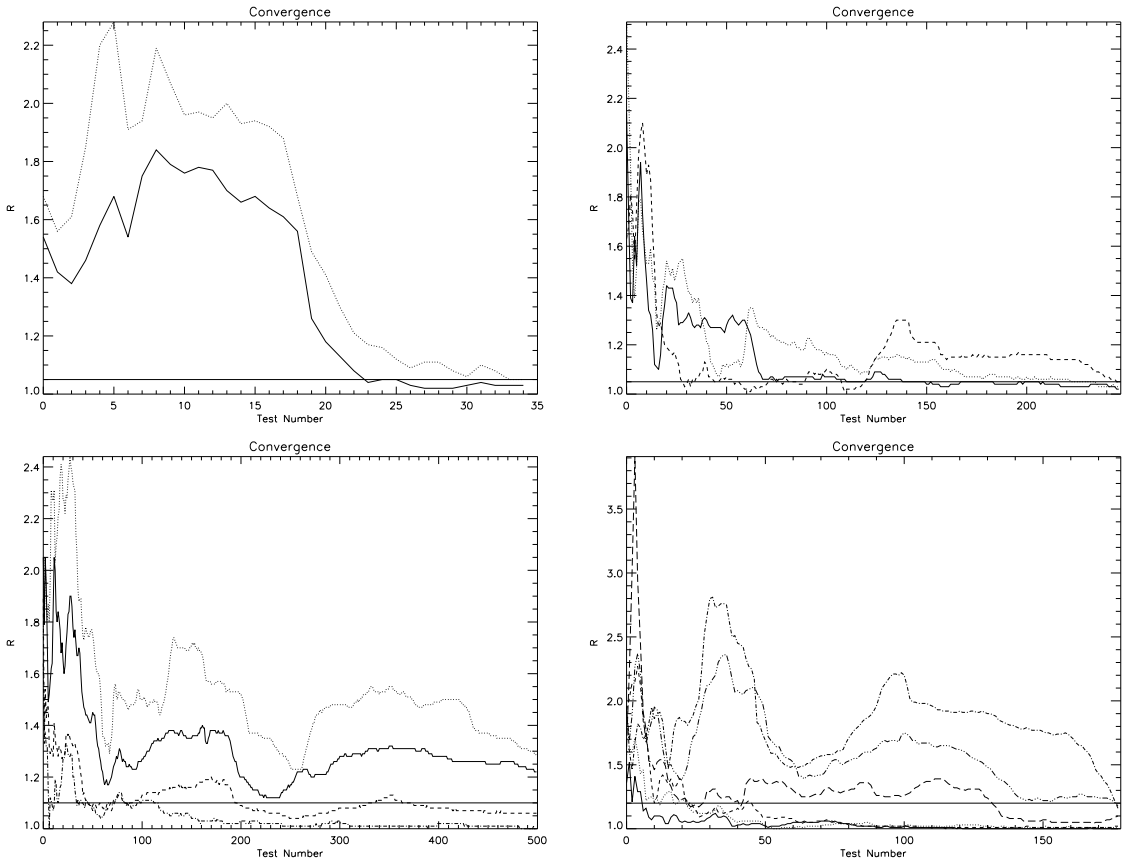


Figure 17: R convergence criterion versus convergence test number for four simulations with convergence intervals of 20. The horizontal line shows the maximum converged R value. The tendency away from convergence is due to the bimodal nature of the particular distribution and variation in acceptance rate. Top left: fit p and q, color evolution set to 2.1, redshift cutoff set to 2.7; Top Right: p, q and color evolution fitted, redshift cutoff set to 2.7; Bottom left: p, q, redshift cutoff, and color evolution fit; Bottom right: all parameters except color evolution fit, with main LF parameters kept within range of standard error; color evolution set to 2.0.

of the performance, and the fact that for convergence, half of the chain is ignored, in the future I would opt to combine burn-in and main phases, and determine a robust way by which chains can be drawn out of low likelihood, computationally slow regions; in addition, I would refine the annealing procedure to include all acceptance information, not just some number of the most recent trials.

In any case, the simulation is guaranteed to converge if given infinite time, and such considerations are only for the purpose of time efficiency (Brooks & Gelman, 1998). One might argue that such increase in run-time is nevertheless beneficial due to the greater degree of parameter space sampling and higher confidence statistics. Some intermediate position between these two considerations must be taken in future iterations of this MCMC code. Three of the four fitting attempts here converged to the confidence level I required within the time given, with the fourth not quite reaching convergence, although running it an earlier time did see it converge within the allotted time. The setting of an iteration limit is simply to force the program to exit at a given point such that the user might make changes which improve the efficiency of fitting.

One such change which does have a large impact on convergence time is the parameter distribution widths. Set too high, the fitting will convergence within a few iterations and no parameter fitting will occur,

and set too low and the program will take an prohibitive amount of time to converge. In all cases, I found the program capable of convergence within 10,000 chain iterations given that the parameters were assigned reasonable distribution widths. Brooks & Gelman (1998) find that the maximally efficient chain converges in N steps according to the formula $N \approx 300D$, where D is the dimensionality of the the parameter space. In other words, for a D dimensional space, you should expect to run your MCMC sampler for at least $N \approx 300D$ before achieving convergence, though in principle it will take longer than this number of iterations if a) your parameter space is highly irregular or b) your parameter covariance and acceptance criteria are not properly optimized. The limit of 10,000 is easily adjustable but in principle the user should not increase it, opting to re-run the simulation with better distribution widths if the program does not converge within 10,000 iterations. A future program iteration might benefit from automatically computing parameter covariance as in Dunkley et al. (2005) and Acquaviva et al. (2011).

4.2.2 SED Library Evaluation

Referring now to figure 16, I can use the typical discrepancies and comparison of my model to that of especially Marsden et al. (2011) to make a few conclusions about the robustness of the SED templates I employed. In all cases, I find that the sources furthest to the bottom left and top right are consistently underpredicted, and the center of the distribution while in general well fit at times is never quite correct. Looking back to the coverage of the color space by the SED models in figure 6, I can say that this is most likely due to the fact that the Rieke et al. (2009) templates are averages, and we have opted for an SED treatment which is undoubtedly oversimplified. Galaxies with a given luminosity may have very widely varying SEDs due to changing composition, varying morphology, and the potential for different modes of star formation, among other factors. My fitting results determine a color evolution parameter of $a \sim 2.1$, which is in agreement with the findings of e.g. Sajina et al. (2012), who find that the SEDs of LIRGS at $z \sim 2$ most closely resemble local local galaxies which are an order of magnitude less luminous. My evolutionary parameter fit gives me

$$\frac{L(z=2)}{L(z=0)} = (1 + (z=2))^{2.1} = 3^{2.1} \approx 10$$

This gives me high confidence that my SED models are correctly modeling, to a first degree, the galaxies one observes at high redshift.

I would conclude from these considerations that our SED templates are not robust enough to produce a highly accurate simulation, and the inclusion of models which vary within a luminosity bin would improve the ability to reproduce a wider range of potential galactic populations, and fill the metric space more completely. I feel that main limitation on our goodness of fit is the lack of diversity in our SED templates, although I will discuss a few other potential improvements in the discussion.

5 Discussion

The main finding of this effort was that the creation of such a simulation and fitting package has produced an incredibly powerful tool, which is nevertheless very limited by the robustness of its input models, and while it may operate in the absence of pre-existing knowledge of a sample, is necessarily bayesian and benefits greatly from earlier understanding of a sample. I do need to adjust our fitting statistics to take into account all of this existing information in the future, such as number counts and redshift distribution information, but in its absence this tool gives us a very good idea of the evolutionary history implied by even the most scattered and informationally sparse data sets, given a basic knowledge of the instrument used to conduct such measurements. As in the results section, the discussion of the strengths and weaknesses method fall into two categories, those due to the structure of the simulation and fitting code, and those due to the specific SED and LF models chosen.

5.1 Comparison to Similar Studies

Comparing our results mainly to those of Marsden et al. (2011) and Sajina et al. (2012) produced high confidence in our method, and many suggestions for areas to improve upon in the near future. I find that the density and luminosity evolution are entirely consistent with the power-law fits found in Marsden et al. (2011), who perform an analysis similar to ours on data taken with a nearly identical instrument, and conclude that our main fits are consistent with those found by other analyses. I also find that, given the SED library employed, the fitted color evolution predicts a luminosity increase exactly in line with that found in Sajina et al. (2012). In terms of the luminosity function evolution, I have reproduced in my analysis the same conclusions reached through other means. That being said, we could not reproduce the number counts calculated by Clements et al. (2010) to very high precision, and the redshift distribution appears to be skewed towards high redshift sources, which are not unreasonable but seem unlikely. The shape of the redshift distribution does tend in the right direction, however the number of high redshift sources is most likely overstated (see e.g. Devlin et al. (2009)).

Some improvements in the accuracy of these fits might be made if I had instead adopted the local luminosity function fitted by Vaccari et al. (2010), which differs slightly from that of Negrello et al. (2013) but is measured in exactly the regime we are analyzing, whereas the Negrello et al. (2013) results are for sources at slightly higher wavelength and thus should show slightly different behavior. I saw disagreement between my evolutionary trends and that of Caputi et al. (2007), which was initially confusing, until one considers the differences between samples. The Caputi et al. (2007) sample was a primarily mid to near IR sample at around 8 microns, and was thus more local than our sample; this combined with the known trend of much shallower evolution for local infrared galaxies, combined with the knowledge that galaxies falling within our bandpass will tend to exist at redshifts higher than 1.2 (Devlin et al., 2009) makes us reasonably certain that we should see stronger evolution. My fitted density and luminosity evolution are, in summary, in agreement with existing analysis, and the fits could be improved by more accurate local luminosity functions and a more robust SED template. In the near future, this analysis will benefit from the inclusion of a newer Herschel sample containing a much greater number of sources, covering a much larger spatial area, and updated local luminosity function.

A final improvement which might apply more broadly to all analyses which employ a similar LF parameterization is that a more robust model than a single power law may be needed to reproduce evolutionary trends. These trends are, after all, more complex, and perhaps a better approach would be to fit some generic polynomial for the power instead of a constant term. Given the density and luminosity evolutionary trends shown in e.g. Marsden et al. (2011), it is apparent that neither trend is truly a power law, the power law describes the average behavior over the considered redshift range. For this analysis to produce realistic surveys, I had to impose an artificial redshift cutoff, which when allowed to vary did in fact limit itself, lending credibility to the notion that either a) this power law should be flattened eventually in a more gradual way or b) the initial evolution is too strong, and it must be flattened to accurately reproduce high-redshift volumes which produce the majority of sources.

5.2 Future Improvements

The primary note on the implementation end is that despite fitting observed and simulated color-color metrics well, without an explicit place in the goodness of fit calculation, the number counts and redshift distribution cannot be fit to a high degree of accuracy. Given that our sample contained only 1000 sources, and the luminosity function had 10-16 bins depending upon observation band, means we couldn't hope for counts with incredibly high degrees of accuracy, however we see entire orders of magnitude in uncertainty across the entire range of flux densities. It is easy to stipulate that including the number counts, if available, in the goodness of fit will benefit the fitting routine and improve the confidence of fitted parameters if the model is

made to reproduce accurate number counts. Along this same vein, we would add the option to specify some template redshift distribution to help constrain this metric as well.

The MCMC fitting routine behaved incredibly well after some basic optimizations for my model, but has some room for improvement as well. It is common for routines such as this to adopt an adaptive covariance matrix, which adjusts the relative step sizes of the parameters to achieve and maintain an optimal acceptance ratio and uniformly drive all parameters towards convergence. I was relying on initial fits to give me an idea of parameter covariances, and inputting these by hand as the ideal step sizes, with the overall acceptance normalization performed internally by adjusting the sampling temperature. For this MCMC program to be truly competitive with similar modules, for example COSMOMC (Lewis & Bridle, 2002), it should have this functionality, and it will be implemented in the near future.

Future program iterations will see much added simulation functionality including the addition of instrument specific filters, which will be used to convolve flux densities, as well as more options in terms of luminosity function form and SED treatment. These will be determined by the model improvements we make in our own analyses. I will also polish this program for release to the public in a way that the IDL, python, and C++ code may be compiled, used, and customized in such a manner as to minimize the pre-existing infrastructure necessary for the user. I find this program to be immensely powerful and useful, and hope for the astronomical community to be able to benefit from its robustness and ability to quickly characterize any band-matched survey.

5.3 Broader Impacts

The constraint of LF evolution has burgeoned of late by a large increase in the size, depth, coverage, and availability with which extragalactic surveys have been conducted, both by ground, balloon borne, and space based experiments. The past decade has seen an increase in the number of high redshift galaxies being studied and discovered on a rich statistical scale. As current and future observatories continue to more finely resolve the cosmic infrared background into individual point sources, we will gain an increasingly rich dataset with which to constrain galactic luminosity evolution. The first set of experiments (e.g. BLAST, Herschel) drove pioneering work into applying these established methods to LF evolutionary constrain, as shown for Marsden et al. (2011) in the case of BLAST. As new datasets become available from future observatories such as JWST or EUCLID (Serjeant et al., 2012), we should be able to simply reapply these analyses to quickly produce better measurements, rather than having to completely re-develop them for each new effort. In this way new analyses may be undertaken earlier, and the scientific yield of short lived infrared observatories greatly enhanced.

The recent results from the BICEP2 experiment (BICEP2 Collaboration et al., 2014) discussed in the introduction, as well as past cases with the *Planck* (Planck Collaboration et al., 2013) and *WMAP* (Spergel et al., 2003; Dunkley et al., 2009) satellites, demonstrate the power of developing and employing a standard MCMC simulation and analysis package. The BICEP2 team used COSMOMC (Lewis & Bridle, 2002) to compare their results to that of Planck Collaboration et al. (2013) using the exact same analysis framework used by the latter team, illustrating the power of having a robust MCMC fitting package readily available for re-constraint of parameters given new data.The work of Wu et al. (2014), who were able to further evaluate these results using COSMOMC within a week of the BICEP2 results being publicly released, demonstrates the benefit of having such packages widely available, and in general use. I hope for this package to evolve to a similar point, and allow constraint of luminosity function evolution to be performed on similar timescales and adopted by the observational community.

6 Summary and Conclusions

I have created a MCMC simulation and fitting package which can constrain the redshift evolution of the luminosity function through comparison to an observed polychromatic survey. I find, through testing this package with a far-infrared sample, that:

1. The best fit parameters to our luminosity function and SED evolution models as presented in sections 2.2 and 2.1 are

$$\begin{aligned} p &= -6.0 \pm 0.7 \\ q &= 4.5 \pm 0.4 \\ z_{cut} &= 2.6 \pm 0.6 \\ a &= 2.1 \pm 0.6 \end{aligned}$$

   These are consistent with the results of both Marsden et al. (2011) and Sajina et al. (2012).

2. The Rieke et al. (2009) SED models are able to reproduce evolutionary trends seen in Sajina et al. (2012) but are not diverse enough to fully reproduce the space of colors seen in the sample.

3. The number counts as calculated for the best fit model agree with the Clements et al. (2010) counts to within uncertainties, however these uncertainties are very large, and would be decreased by including number counts in the calculation of goodness of fit.

4. The MCMC methods have been implemented successfully, showing full sampling of the model parameter space, high flexibility as to parameter number, and consistent convergence. There are areas in which the efficiency of sampling and convergence can and will be improved in the near future.

Given the potential benefit a general tool can have in this field, and the wide success similar tools have seen, I hope now to involve a broader community in the further refinement of this program, in the hopes that this will lend greater exposure to this effort and encourage others to include features which improve the overall yield for all future analyses. Our group at Tufts intends to use this program as a test bed for future SED libraries with explicit redshift dependence, and with such models this package will only improve on its ability to characterize future surveys. The results of this analysis, due to its highly statistical nature, are heavily dependent on the sample size employed for the observed metric and the additional data used to constrain the parameters (e.g. number counts and redshift distributions). I have shown that, given the statistical limitations, this program can extract evolutionary constraints from a dataset comparable to any other analysis, and once configured for a particular type of data, this program can improve upon its constraint as more data is added.

This package promises to be a powerful tool for constraining the evolutionary properties of galaxy samples, and I intend to implement it along with the changes discussed above to analyze larger samples covering different electromagnetic regimes to further demonstrate its flexibility and power. Above all, I have created a package only as powerful as the models it is given, which should allow those conducting such analyses to focus on the physical considerations relevant to modeling galaxies in lieu of worrying about statistical fitting and simulation concerns.

References

Acquaviva, V., Gawiser, E., & Guaita, L. 2011, ApJ, 737, 47
Ajello, M., Shaw, M. S., Romani, R. W., et al. 2012, ApJ, 751, 108
Allison, R. & Dunkley, J. 2014, MNRAS, 437, 3918
Bagaud, B., Martin, K., Abouelfath, A., et al. 2005, European Journal of Epidemiology, 20, 213
BICEP2 Collaboration, Ade, P. A. R., Aikin, R. W., et al. 2014, ArXiv e-prints
Binggeli, B., Sandage, A., & Tammann, G. A. 1988, ARAA, 26, 509
Brooks, S. P. & Gelman, A. 1998, Journal of Computational and Graphical Statistics, 7, 434
Caputi, K. I., Lagache, G., Yan, L., et al. 2007, ApJ, 660, 97
Clements, D. L., Rigby, E., Maddox, S., et al. 2010, A&A, 518, L8
Cowles, M. K. & Carlin, B. P. 1996, JASA, 91, pp. 883
Devlin, M. J., Ade, P. A. R., Aretxaga, I., et al. 2009, Nature, 458, 737
Dunkley, J., Bucher, M., Ferreira, P. G., Moodley, K., & Skordis, C. 2005, MNRAS, 356, 925
Dunkley, J., Komatsu, E., Nolta, M. R., et al. 2009, ApJS, 180, 306
Elvis, M., Wilkes, B. J., McDowell, J. C., et al. 1994, ApJS, 95, 1
Farrah, D., Lonsdale, C. J., Weedman, D. W., et al. 2008, ApJ, 677, 957
Griffin, M. J., Abergel, A., Abreu, A., et al. 2010, A&A, 518, L3
Hastings, W. K. 1970, Biometrika, 57, 97
Hogg, D. W., Baldry, I. K., Blanton, M. R., & Eisenstein, D. J. 2002, ArXiv Astrophysics e-prints
Hubble, E. 1929, PNAS, 15, 168
Johnson, S. P., Wilson, G. W., Tang, Y., & Scott, K. S. 2013, MNRAS, 436, 2535
Johnston, R. 2011, A&A Rev., 19, 1
Kirkpatrick, A., Pope, A., Charmandaris, V., et al. 2013, ApJ, 763, 123
Komatsu, E., Smith, K. M., Dunkley, J., et al. 2011, ApJS, 192, 18
Kurinsky, N., Sajina, A., Partridge, B., et al. 2013, A&A, 549, A133
Lewis, A. & Bridle, S. 2002, PRD, 66, 103511
Marsden, G., Chapin, E. L., Halpern, M., et al. 2011, MNRAS, 417, 1192
Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953, JCP, 21, 1087
Negrello, M., Clemens, M., Gonzalez-Nuevo, J., et al. 2013, MNRAS, 429, 1309
Pilbratt, G. L., Riedinger, J. R., Passvogel, T., et al. 2010, A&A, 518, L1
Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2013, ArXiv e-prints
Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2011, A&A, 536, A1
Rieke, G. H., Alonso-Herrero, A., Weiner, B. J., et al. 2009, ApJ, 692, 556
Ross, S. 2013, in Simulation, Fifth edn., ed. S. M. Ross (Academic Press), 271 – 302
Sajina, A., Yan, L., Fadda, D., Dasyra, K., & Huynh, M. 2012, ApJ, 757, 13
Sanders, D. B. & Mirabel, I. F. 1996, ARAA, 34, 749
Saunders, W., Rowan-Robinson, M., Lawrence, A., et al. 1990, MNRAS, 242, 318
Schechter, P. 1976, ApJ, 203, 297
Scott, D. W. 1979, Biometrika, 66, pp. 605
Serjeant, S., Buat, V., Burgarella, D., et al. 2012, ArXiv e-prints
Smith, A. J., Wang, L., Oliver, S. J., et al. 2012, MNRAS, 419, 377
Sommer, M. W., Basu, K., Pacaud, F., Bertoldi, F., & Andernach, H. 2011, A&A, 529, A124
Spergel, D. N., Verde, L., Peiris, H. V., et al. 2003, ApJS, 148, 175
Vaccari, M., Marchetti, L., Franceschini, A., et al. 2010, A&A, 518, L20
Wiklind, T. 2003, ApJ, 588, 736
Wu, F., Li, Y., Lu, Y., & Chen, X. 2014, ArXiv e-prints