Galaxy Population Models for z = 0-4 based on Dark Matter Halos

Emma Klemets

Summer Undergrad Research Student 2020

McGill Cosmic Dawn Intensity Mapping Group, McGill University

Department of Physics

Supervisor: Adrian Liu, with support from Jordan Mirocha

August 12, 2020

Contents

1 Introduction

My work is building to look at UV and optical luminosity functions to be used to remove foregrounds, so I am looking at galaxy populations between redshift 0 - 4. The main work has been building the ARES galaxy class GalaxyHOD, that contain various functions of a galaxy population model, explained in section 2. These galaxies are based on the masses and densities of the dark matter halos that contain galaxies.

Many of the models have free parameters in them. The default parameters have either been taken from papers or just trial and error. To get better results I ran MCMCs to fit the models to known data, which can be seen in section 4.

Terms and conventions used here

- stellar mass: m or SM
- halo mass: M or HM
- (Dark) halo mass function: HMF = $\frac{dn}{dM}$,
- stellar mass function: SMF
- luminosity function: LF
- absolute magnitude: mag or Mag
- star-forming: sf
- quiescent: q

2 ARES Work

2.1 GalaxyHOD.py

The default values for this class can be found in \input\litdata\emma.py, to make a population object with the defaults, do:

```
base_pars = ares.util.ParameterBundle('emma:model1')
pop = ares.populations.GalaxyPopulation(**base_pars)
```

For a brief overview of the models in this file, see table 2.2.

For most functions, any masses are expected to be $\log_{10}$ values, and some have the optional Boolean tag to switch from stellar mass input to halo mass input.

Luminosity Function

```
LuminosityFunction(self, z, x, text=False)
```

Reconstructs the luminosity function from a simple model base on the HMF. I first tried relating $LF\ [mag] = c \cdot HMF$ where [c] = [solar mass], but as the HMF curve decreases with mass (bigger galaxies are rarer) and the expected LF curves increase with mass (less bright galaxies are more common), this did not result in a good LF model.

Instead I used the relation of luminosity to absolute magnitude. Starting with a linear relation between luminosity and halo mass: $L\ [lum] = c \cdot mass$ where [c] = [solar mass s /J]. In ARES, c ia actually a Parameterized Quantity, however the default is a linear function (with respect to z) where only the constant is non-zero. Then using the conversion to absolute magnitude (using Mag_o and L_o constants for reference, for ease let $Mag_o = 0$ and $L_o = 1$):

$$Mag - Mag_o = -2.5 log_{10}(\frac{c \cdot mass}{L_o}) \tag{1}$$

This gives the new x axis values, in absolute magnitude. Then to rebin and change the LF from a function of mass bins to magnitude bins, I used the change of variables for probability distribution, assuming a linear relation between magnitudes and masses: $mag = a \cdot mass$.

$$LF(mag) = \frac{LF(mass)}{(\frac{d(mag)}{d(mass)})} \quad \text{where} \quad \frac{d(mag)}{d(mass)} = |\frac{-2.5}{ln(10) \cdot mass}| \tag{2}$$

$$LF = |\frac{ln(10) \cdot mass}{-2.5}|\ HMF(Mag(mass)) \tag{3}$$

The work done to build this model can be seen in WorkingFromHalo.ipynb .

Stellar Mass Function

```
StellarMassFunction(self, z, logbins, text=False, **kwargs)
```

This model was made following the paper Moster2010 [1], based on the relation of galaxy masses to halo masses. The main idea is a ratio between the galaxy and halo mass:

$$r = \frac{m(M)}{M} = 2N_1[(\frac{M}{M_1})^{-\beta} + (\frac{M}{M_1})^{\gamma}]^{-1} \tag{4}$$

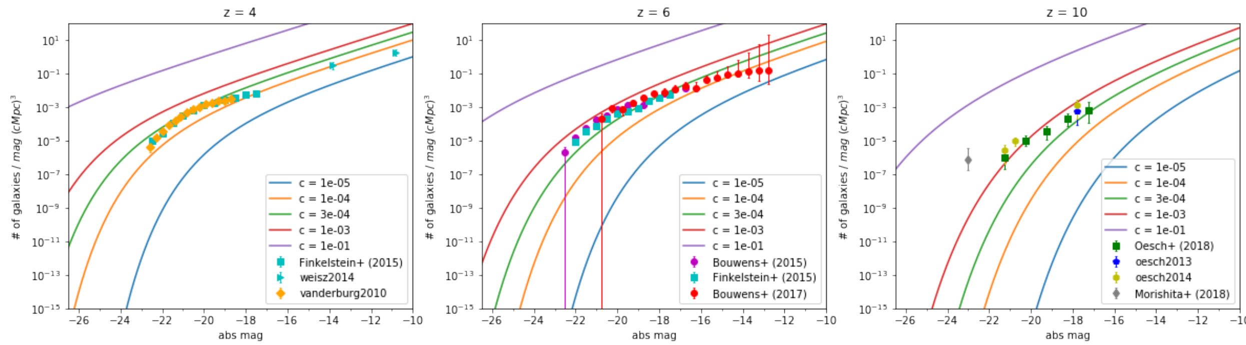

Figure 1: Results from LuminosityFunction() for z = 4, 6 and 10 for a couple values of the free parameter c, along with literature values.

Where N_1, M_1, β, γ are fit parameters, parametrized to change with redshift according to Eqs. 7.

The SMF is found with:

$$\phi\ [\ /(Mpc)^3 dex^{-1}] = \frac{dM}{dlog(m(M))} \cdot \frac{dn}{dM} \tag{5}$$

$\frac{dn}{dM}$ is simply the halo number density, already built into ARES halo objects, so only the first derivative needed to be coded. This is done analytically in `_dlogm_dM()`, using Equ 4:

$$\frac{dM}{dlog(m(M))} = -1\frac{(\gamma-1)(\frac{M}{M_1})^{\gamma+\beta} - \beta - 1}{logM\frac{M}{M_1})^{\gamma+\beta} + 1} \tag{6}$$

With redshift-dependent parameterizations

$$\begin{aligned} M_1 &= 10^{logM_0} \cdot (z+1)^{\mu}\ ^{1} \\ N_1 &= N_0 \cdot (z+1)^{\nu} \\ \gamma &= \gamma_0 \cdot (z+1)^{\gamma_1} \\ \beta &= \beta_1 \cdot z + \beta_0 \end{aligned} \tag{7}$$

First the SMF is found for the entire halo mass array, and the corresponding stellar masses

[1]This was changed from the equation used in Moster2010: $logM_1 = logM_0 \cdot (z+1)^{\mu}$ to decrease the sensitivity to μ as it was compensating for a better fit at high z and messing with the fit of the normalization parameters.

are calculated in _SM_fromHM() using:

$$m = 10^{log_{10}(r)+log_{10}(M)} \tag{8}$$

Then the logbins values that are passed into the function are compared to the calculated stellar masses, if they match up exactly, the corresponding SMF values are returned. If not, interpolate.interp1d() is used and values that are out of the range will return as -inf.

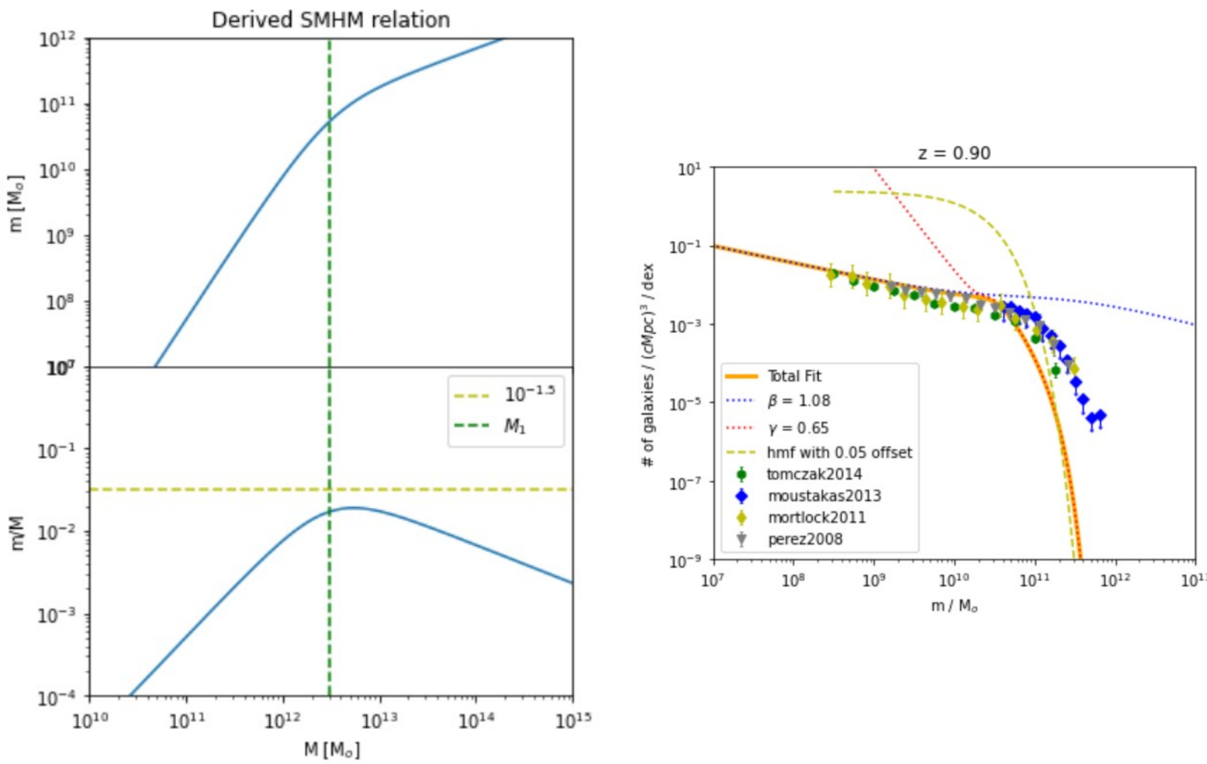

Figure 2: Right: Relation between stellar mass and halo mass, using default parameters for z = 1 (match Fig 4 in Moster2010). Left: The behaviour of the gamma and beta terms in the double power law for z = 0.9, this is using the old M_1 equation.

The work done to build this model can be seen in HMFtoSMF_nonLinear.ipynb

(Specific) Stellar Formation Rate and Density

Work in SFRs.ipynb

```
SFR(self, z, logmass, haloMass=False, log10=True)
```

This function is based on Speagle2014 [2], specifically: 5.1 The Evolution of the Galaxy

"Main Sequence". It is a linear relation with respect to stellar mass and time:

$$log_{10}\Psi(m,t) = (a - b \cdot t) \cdot log_{10}m - (c - d \cdot t) \quad (9)$$

where m is the stellar mass, t is the age of universe in Gyr and a, b, c, d are the free parameters. The default values are taken from Speagle2014, seen in Table 2.1

a	0.84(2)	b	0.026(3)
c	6.51(24)	d	0.11(3)

Table 1: Default parameters for Equ 2.1, from fits in Speagle2014. These values are for mass range $\log_{10}$m = 9.7 – 11.1, meant for z $\sim$ 0.25 – 2.75, but gives a decent fit out to z = 5.

In the class, the first and second term are parametrized as functions of t, so more complicated functions could be used here instead.

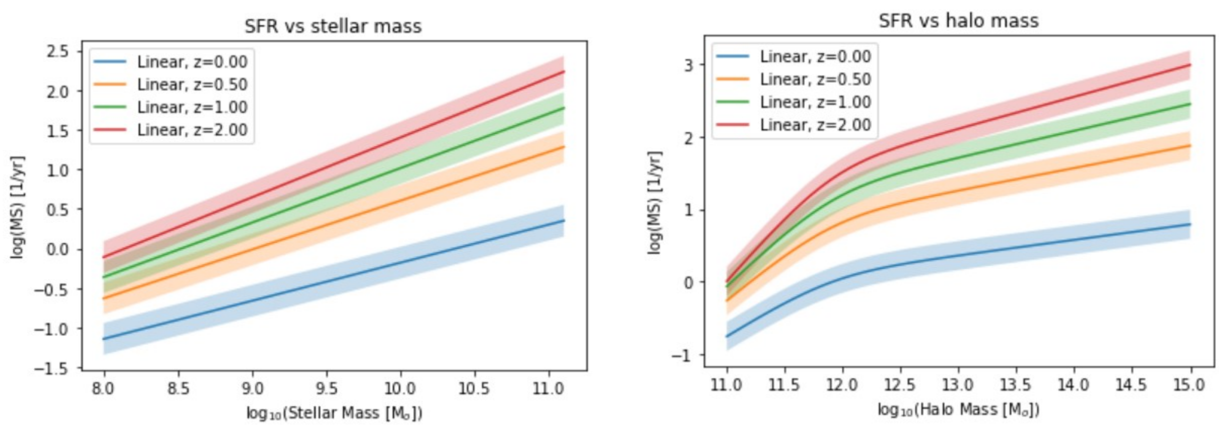

Figure 3: SFR with respect to stellar mass (right), and halo mass (left) with default parameters for a range of redshifts.

```
SSFR(self, z, logmass, haloMass=False)
```

This is simply $SSFR = \frac{SFR}{SM}$

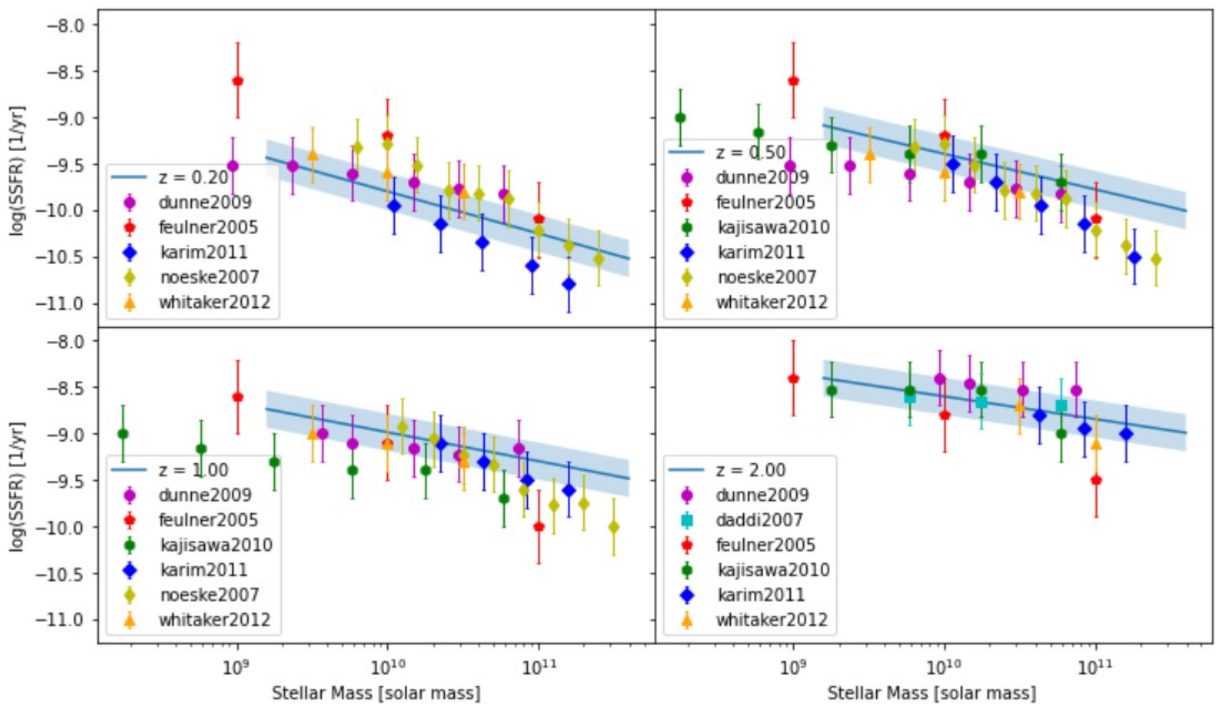

Figure 4: SSFR model, error and literature data for a number of redshifts.

`SFRD(self, z)`

The density of the SFR with respect to redshift. This is found by integrating SFR $\cdot$ number density (really the SMF) of each mass over all masses:

$$SFRD(z) = \int SFR(z,m) \cdot SMF(z,m) dm \approx \sum_m [SFR(z,m) \cdot SMF(z,m) \cdot dm] \qquad (10)$$

I am using the approximation, as I can easily find dm.

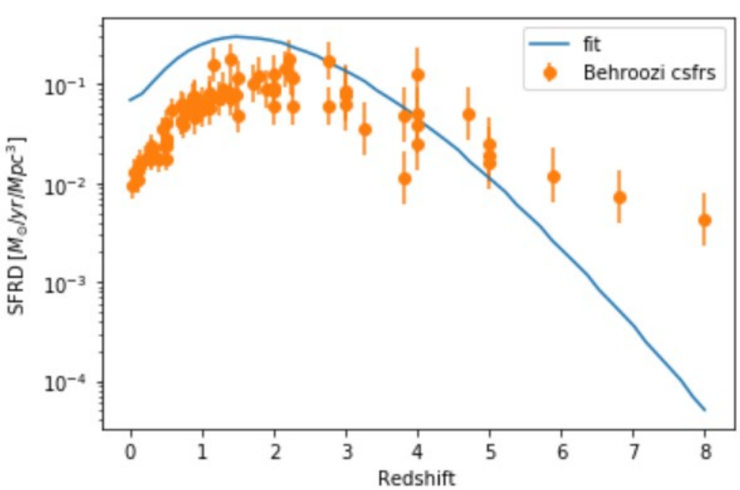

Figure 5: SFRD fit with default parameters and cSFRS data [3]

2.2 Other files

Edited files

GalaxyPopulation.py

Added PlotSSFR(): to plot literature values for the SSFR.

ParameterizedQuantity.py

Added two equations:

PowerLaw10 ('pl_10') $= 10^{p[0]} \cdot (\frac{x}{p[1]})^{p[2]}$

PointsLinear ('p_linear') $= \frac{(p[3]-p[2])}{(p[1]-p[0])} \cdot (x - p[1]) + p[3]$

Data files - input/litdata/

SMF - Corrected as seen in Behroozi2013 [3], Table 3

- marchesini2009_10.py
- mortlock2011.py
- perez2008.py
- moustakas2013.py
- tomczak2014.py - added lower z values [4]

SSFR - Corrected as seen in Behroozi2013 [3], Table 5

- daddi2007.py
- dunne2009.py
- feulner2005.py
- gonzalez2012.py
- kajisawa2010.py
- karim2011.py
- noeske2007.py
- whitaker2012.py

emma.py - My ParameterBundle for the default parameters (model1) for an HOD galaxy object. Use with caution as many of these values are not the best fit for the model. See table 2.2 for the different options.

Model	**SMF - DPL values**	SF_{fract} **function**	SF_{fract} **values**
model1	default parmeters	a, b dpl; c, b linear	default parmeters
model2	best fit parameters	a, b dpl; c, b linear	default parmeters
model3	best fit parameters	all dpl	default parmeters
model4	default parmeters	a, b linear; c, b constant	default parmeters
model5	best fit parameters	a, b dpl; c, b linear	best fit parameters

3 Star-Forming vs Quiescent Galaxies

To properly look at the galaxy populations, along with looking at the total population, we also need to be able to separate the star-forming (blue) and quiescent (red, dead) galaxies [5]. This is done by having a fraction which gives the percent of star-forming to quiescent galaxies that is simply multiply into the SMF. So now we have:

$$\phi\ [\ /(Mpc)^3 dex^{-1}] = \frac{dM}{dlog(m(M))} \cdot \frac{dn}{dM} \cdot \begin{cases} 1 & \text{total pop} \\ SF_{fract} & \text{sf pop} \\ 1 - SF_{fract} & \text{q pop} \end{cases} \tag{11}$$

3.1 Star-Forming Fraction

To get the actual function for SF_{fract} I went through a couple different equations. I first started with a linear relation: $SF_{fract} = m \cdot log10(halo\ mass) + b$, however this did not get the curvature of the quiescent galaxies at lower masses. Instead I settled on:

$$SF_{fract} = \frac{1}{c}[tanh(a \cdot (log_{10}(SM) + b) + d] \tag{12}$$

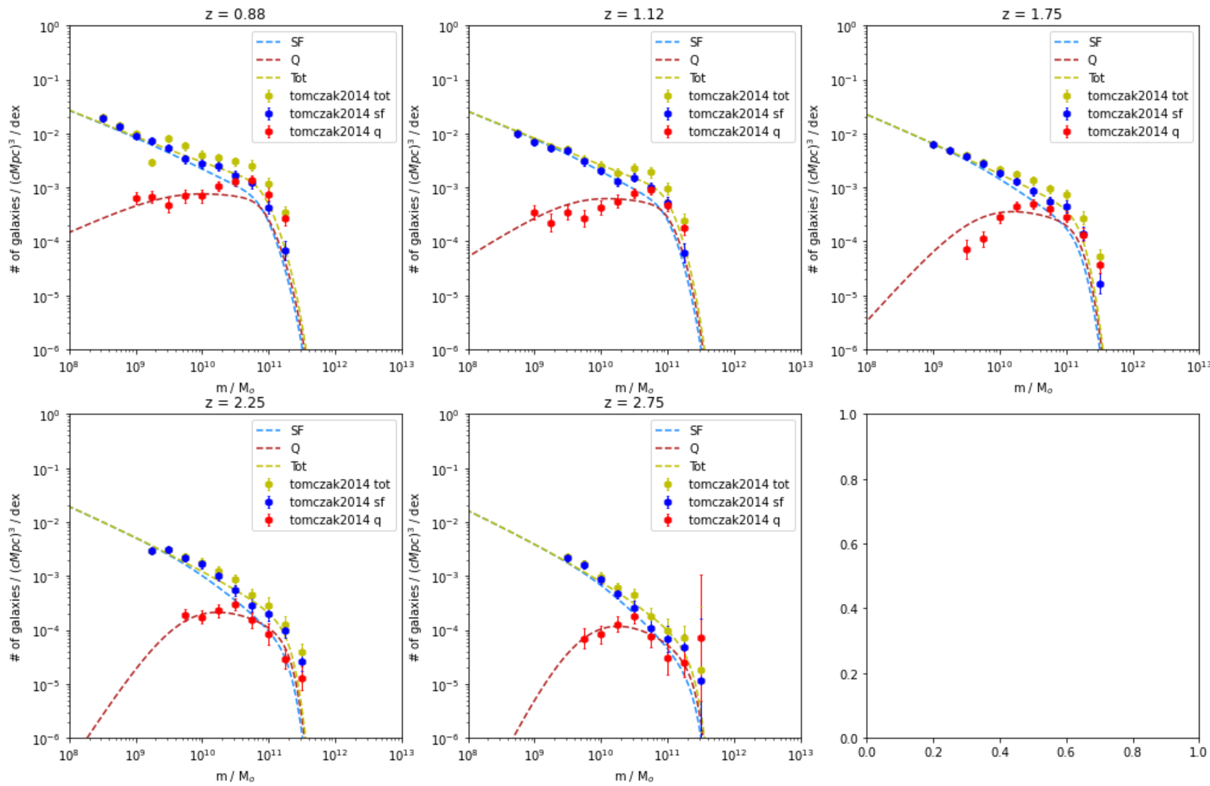

Figure 7: Attempt on using SF_{fract} to get the three different galaxy population SMFs. Here the values were just fit by hand by trial and error.

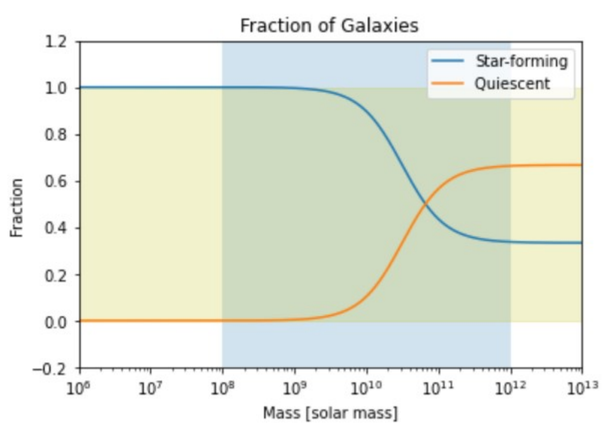

Figure 6: The SF_{fract} (and it's inverse for the quiescent galaxies) using Equ 3.1. z = 1.75, a=-1.7, b=-10.5, c = 3.0, d = 2.0.

I then needed to parametrize the constants with respect to z. I started with a linear function of a and b, and played around with the values by hand, resulting in some decent fits as seen in Fig 7. Work for this is found in SMF_SFvsQ.ipynb.

3.2 Generalized LF

3.2.1 Star-Forming

For the luminosity function for star forming galaxies, the assumption is that they've been forming stars at a constant rate for quite awhile. The luminosity for a given wavelength is found with the ARES function $L_{perSFR}(\lambda)$. So we simply use the SFR(z, Hm) to get an array of luminosities and then rebin to get the luminosity function, which is done numerically.

$$Lum_\lambda = L_{perSFR}(\lambda) \cdot SFR(z, Hm) \implies \phi(M_\lambda) = \frac{dM_h}{dLum_\lambda} \cdot \frac{dn}{dM_h} \tag{13}$$

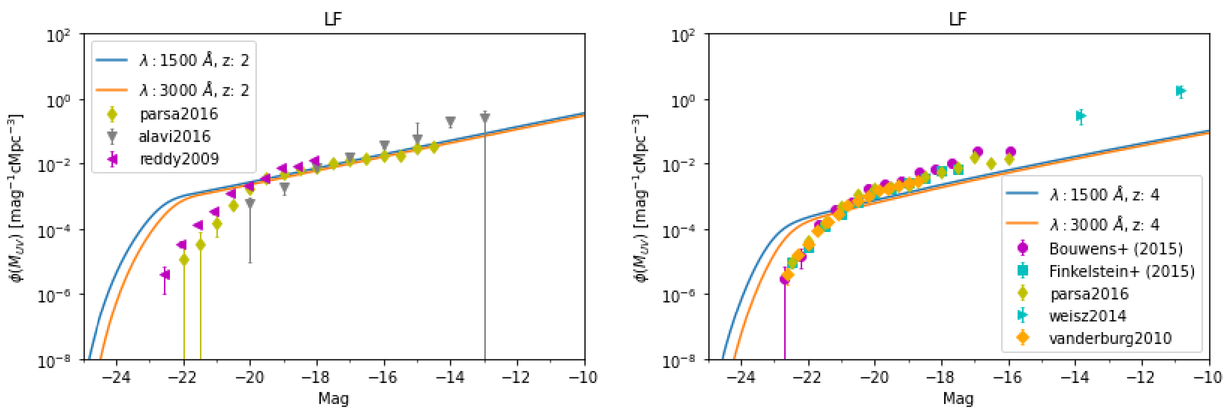

Figure 8: Example of the luminosity function for two different redshifts and wavelengths. Note the literature data is not all for the same wavelength.

Work for this is found in Gen_LF.ipynb. This is implemented in my ARES class as: `Gen_LuminosityFunction(self, z, x, Lambda)`. Right now it is only made for star-forming populations. Any population objects that will use this function need two additional parameters added to base_pars for the source:

pop_sed = 'eldridge2009' - literature data for source

pop_tsf = # - population age [Myr]

Quiescent

I did not get to making the generalized luminosity function for quiescent galaxies. This is a little more complicated to implement, assumptions are needed about how long the galaxy has been dead for to get it's luminosity. Then the model should be built off this distribution.

4 MCMC Fitting

All fitting was done with the MCMC method, working with emcee through ARES. I started off with this example.

4.1 LF

Here I was fitting the linear function for c, (the constant) seen in Equ 2.1. We have three free parameters:

pq_func_par0[4]: const

pq_func_par1[4]: offset such that $c(z) = \text{slope} \cdot (z - \text{offset}) + \text{const}$

pq_func_par2[4]: slope

The set up of the MCMC can be found in this file, with only one blob, for the LuminosityFunction(). All parameters have a uniform prior of (0, 1), and initial guesses of: const = 3×10^{-4}, offset, slope = 0.

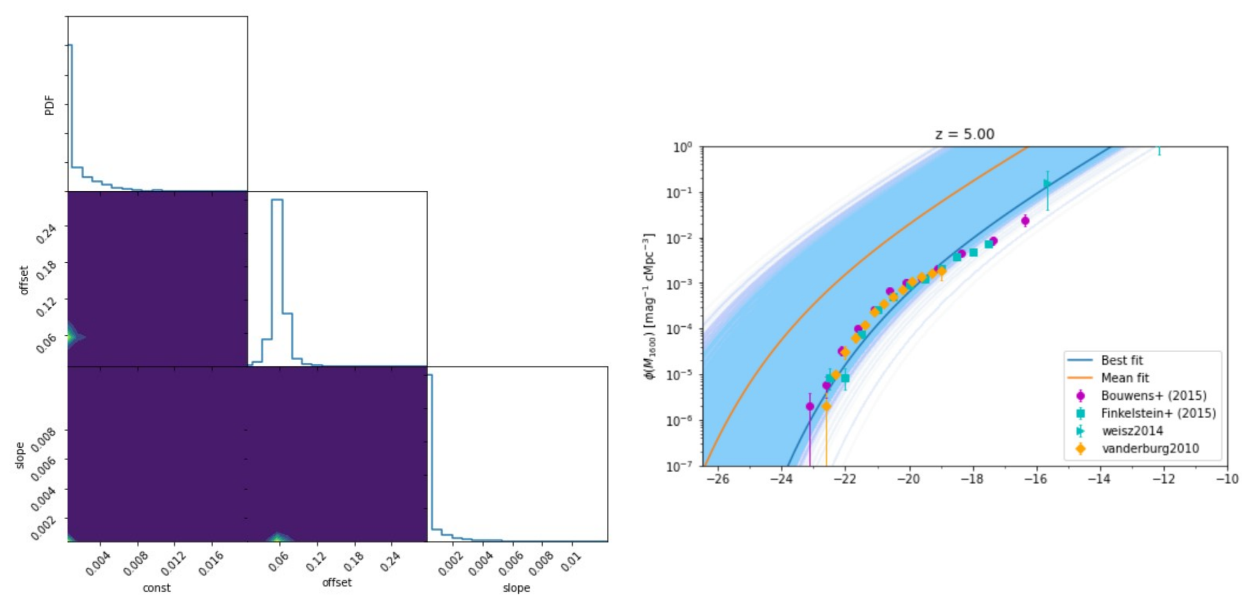

Figure 9: Triangle plot and SMF results from fitting the LF. THis gave best fit results of: const = 1.52×10^{-6}, offset = 0.13, slope = 6.69×10^{-5}. MCMC files can be found here

4.2 SMF

The fitting for the SMF was a lot more involved than that of the LF, as I now had 8 parameters to fit, two for each redshift-dependant parameter of the double power law, as seen in Equ. 7. After a long road of errors, bad likelihoods and improving my SMF function, it produces decent fits now.

Most of the MCMCs were run on the Cedar computer cluster so that multiple cores could be used to speed up the process. The script to run these as well as basic analysis are briefly explained in section 5.1.

I played around a lot with the priors used for the parameters, the ones I settled on using can be seen in Table 4.2, along with the initial guesses. Some of the work for this can be found in the start of CheckingFitGuesses.ipynb.

Labels	**Parameters**	**Prior Bounds**	**Initial Guesses**	**Best Fit Results**
β_0	pq_func_par0[0]	(-1, 4)	1.06	0.88
β_1	pq_func_par2[0]	(-1, 1)	0.17	-0.034
N_0	pq_func_par0[1]	(0, 2)	0.0282	0.010
ν	pq_func_par2[1]	(-1, 1)	-0.72	0.29
γ_0	pq_func_par0[2]	(0, 1)	0.556	0.56
γ_1	pq_func_par2[2]	(-3, 1)	0.556	0.18
$log_{10}M_0$	pq_func_par0[3]	(10.0, 14.0)	11.88	11.7
μ	pq_func_par2[3]	(-2, 10)	2.0	1.86

Table 2: Prior bounds and initial guesses for fits for the SMF. Most initial guesses are the parameter values found in Moster2010 [1], except for μ as the equation changed from the paper. The best fit values are those as seen in Fig 4.2.

The following is an example result, from this SMF MCMC output, files: smf_29_07_12-08_0.0-4.0. Done with 128 walkers and 400 steps, fitting to SMF total data from tomczak2014, mortlock2011, moustakas2013 and marchesini2009_10, for z = 0 - 4. The path of the walkers can be found in the Appendix, Fig 19. As some of the initial guesses are not in a high probability region, there is a significant number of steps that some of the walkers spend at the start, even after the burn in, getting to a better parameter space. To account for this, the first 100 steps of each walker are removed to avoid double peaked histograms.

I also saved the SFRD as a blob for these fits, as it depends on the SMF. The corresponding results for this can be seen in Fig 14. It's not a great fit and is in general too high, but I also only have one set of data to compare to here.

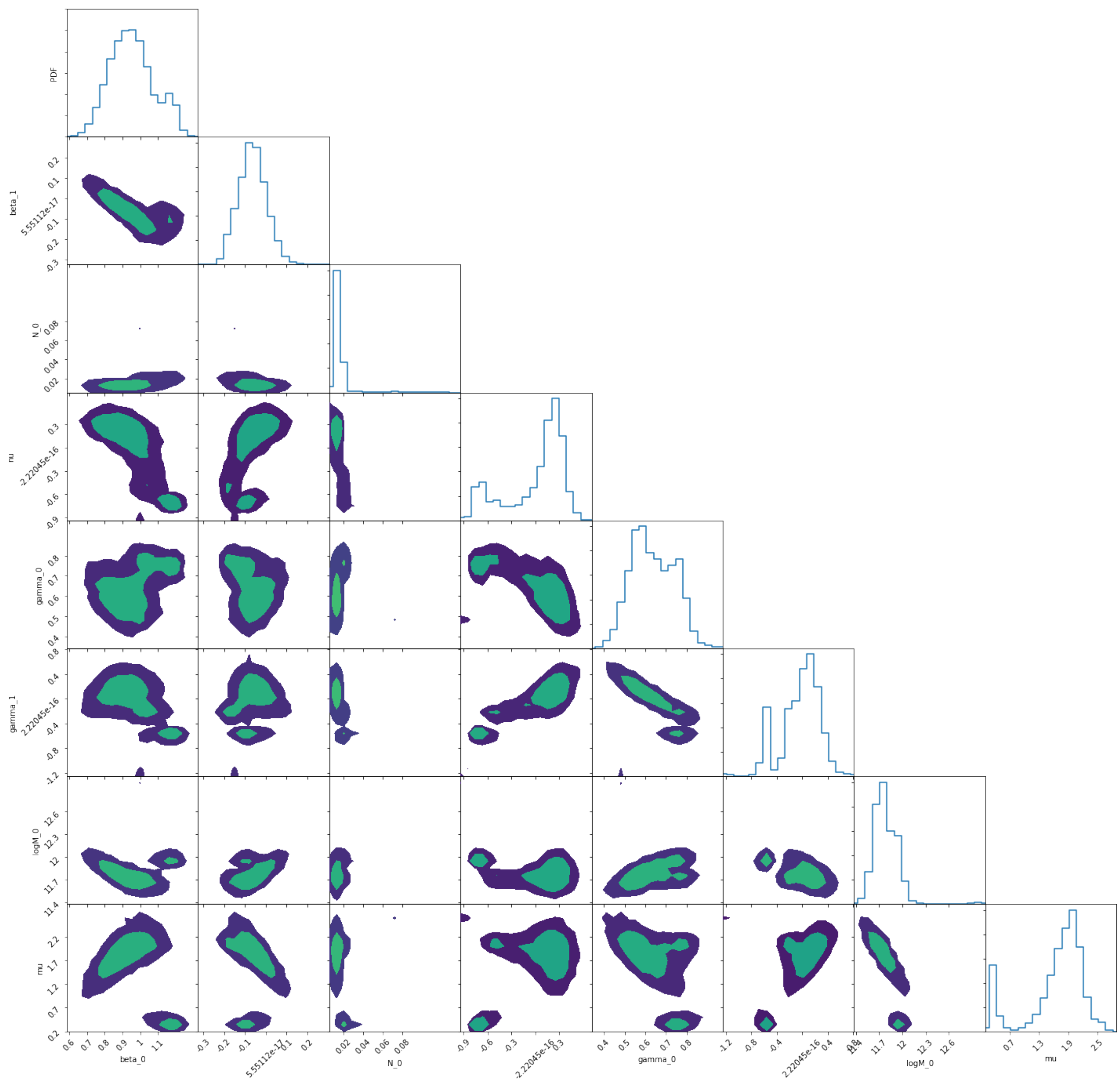

Figure 10: Triangle plots, before cutting the first 100 steps.

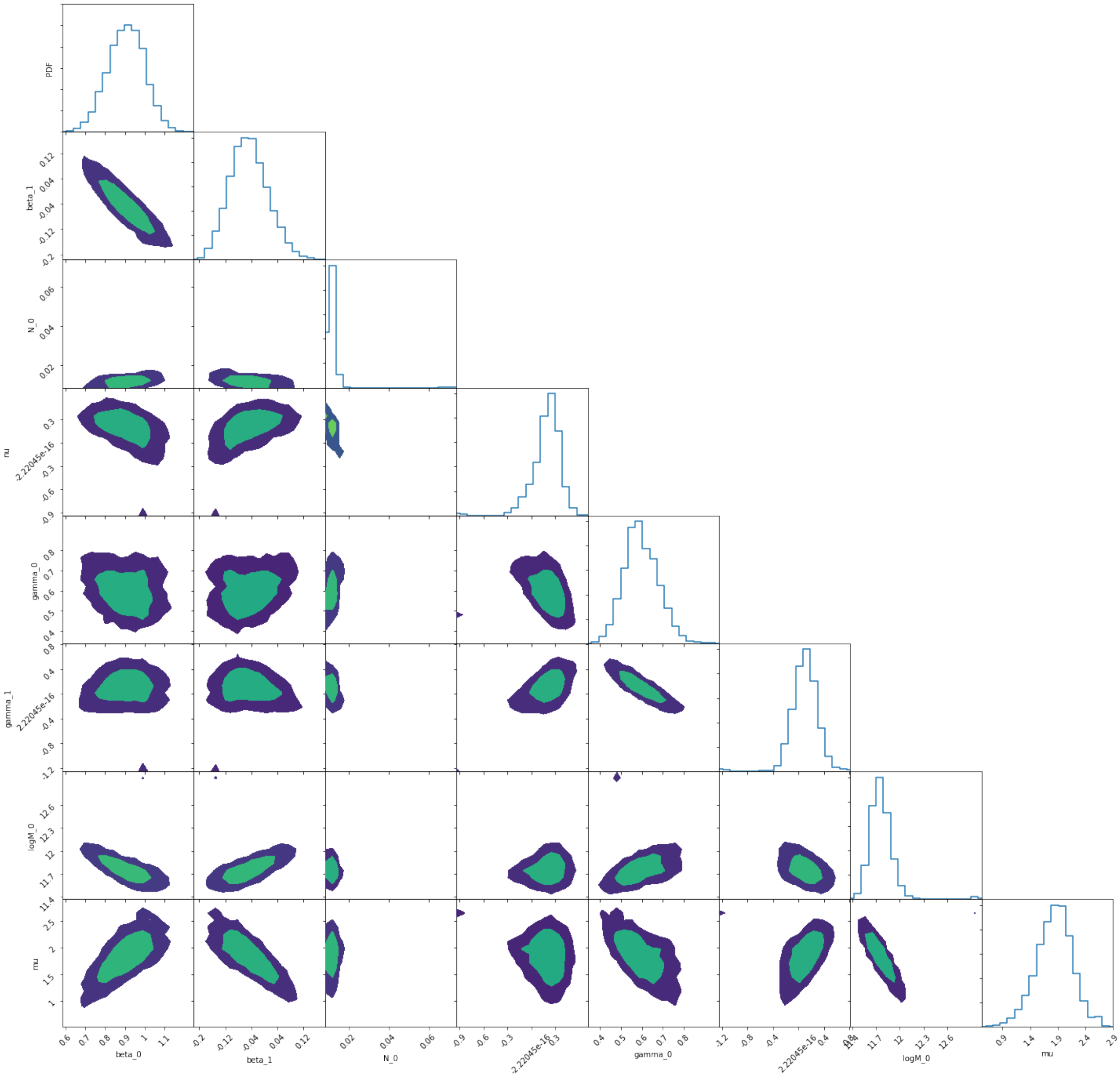

Figure 11: Triangle plots, after cutting the first 100 steps.

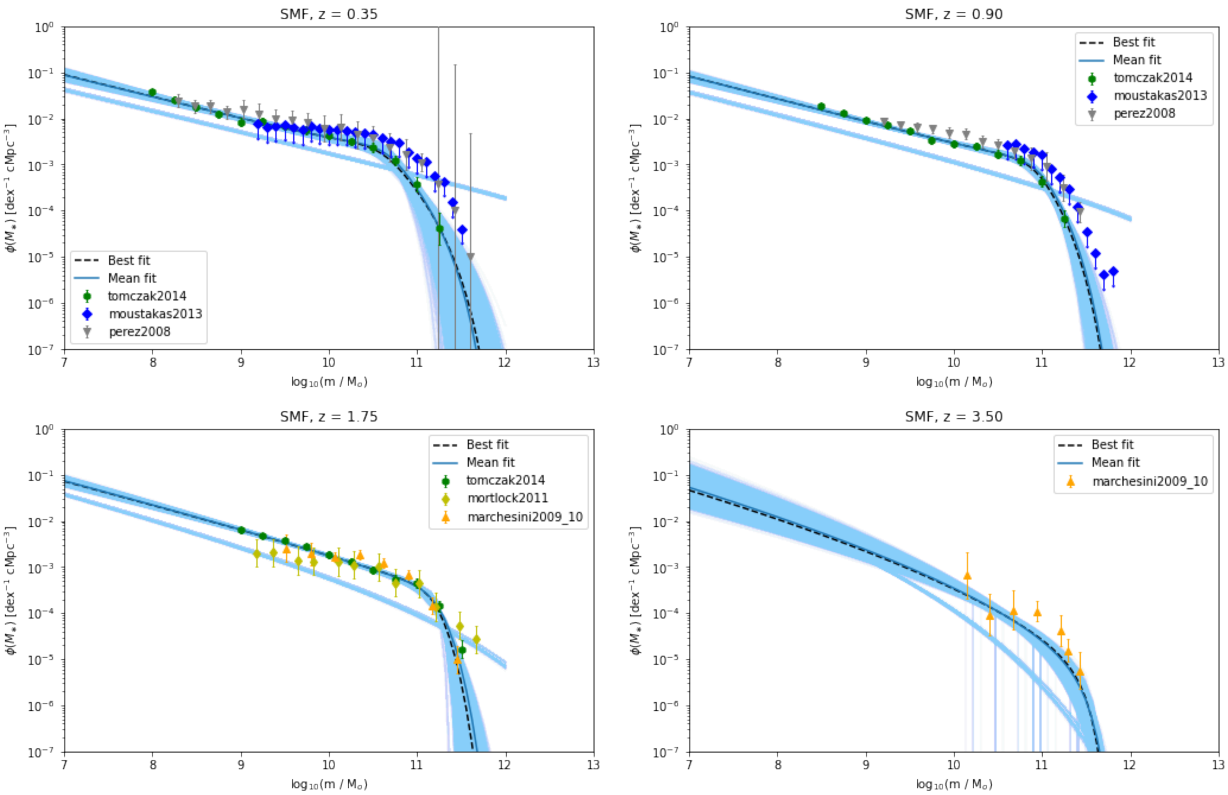

Figure 12: SMF fit results, with data, best and mean likelihood parameter models.

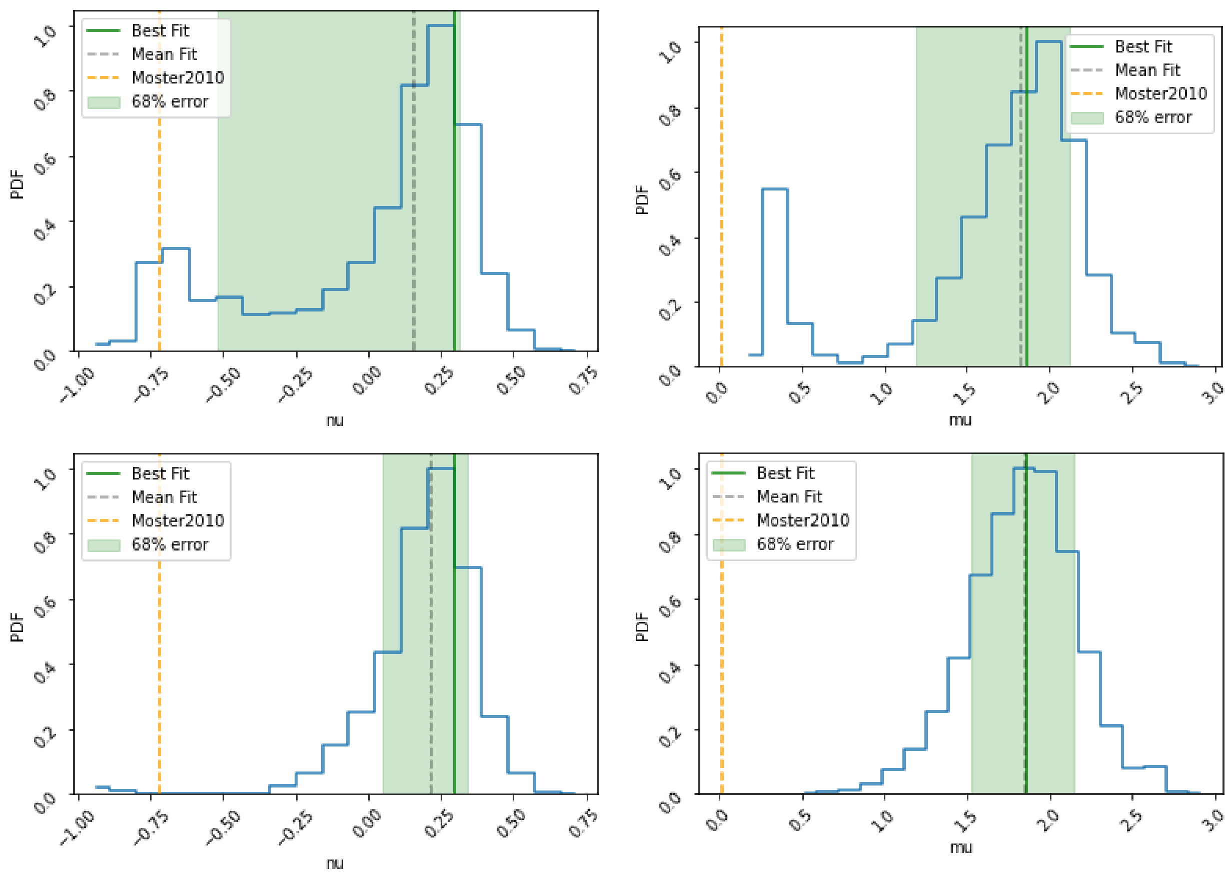

Figure 13: Probability distribution with best and mean fit values marked, of the normalization exponent ν and the peak mass exponent μ before (top) and after (bottom) cutting the first 100 steps.

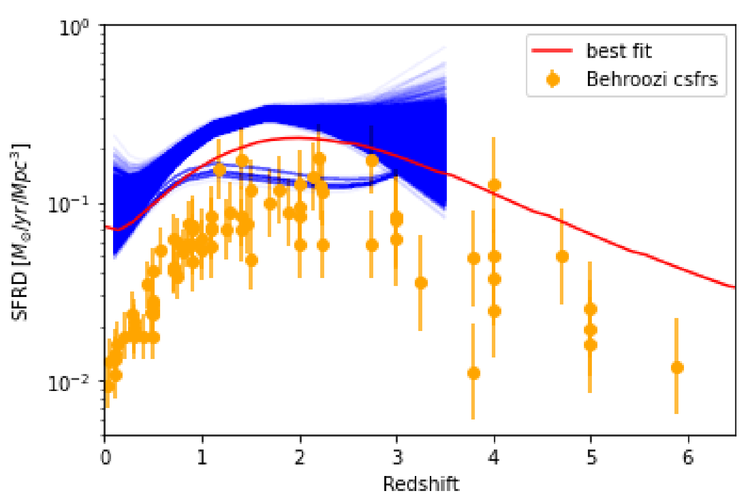

Figure 14: SFRD resulting from fit, best fit model and data.

I did a lot of runs on Cedar, with a variety of redshift ranges. The values from a run with different redshift ranges can be seen in Fig 4.2. I repeated many of the runs to get averages of the best fits parameter values. The final parameter values can be seen in Fig. 4.2. This work was all done in CompareParams.ipynb.

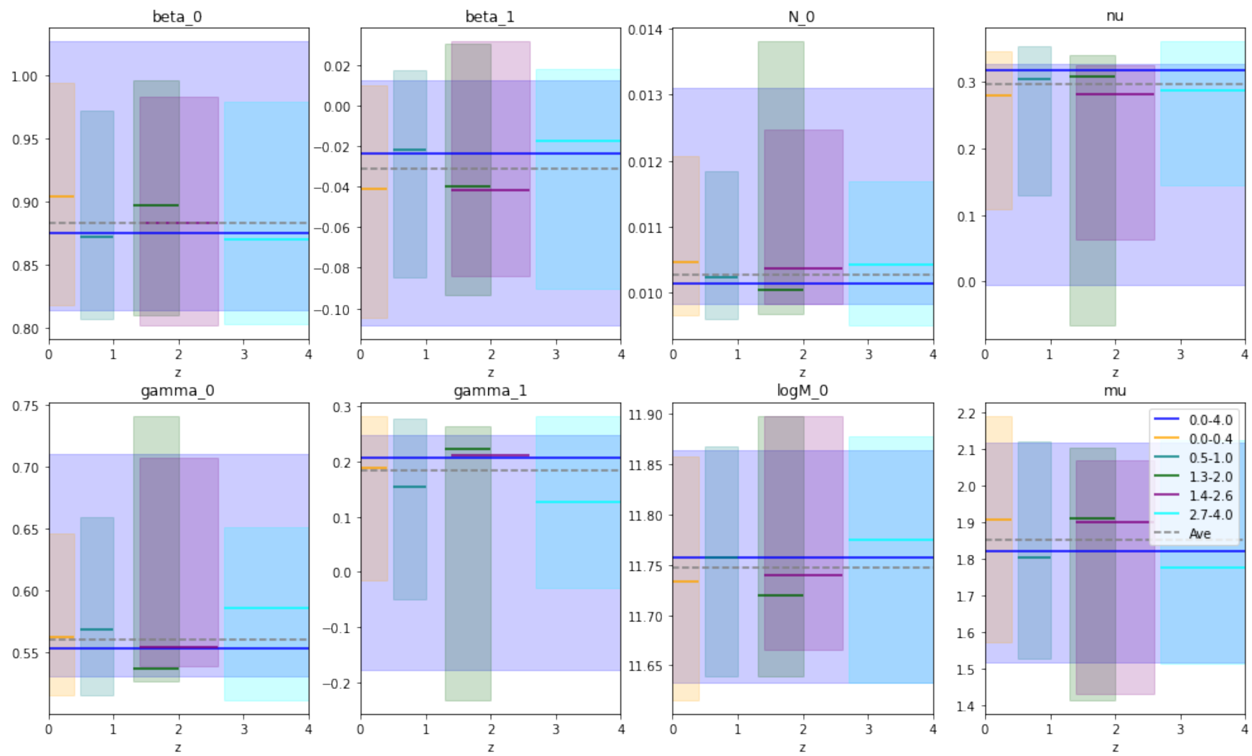

Figure 15: Best fit parameter values for a number of runs with different redshift ranges they where fitting over (x axis).

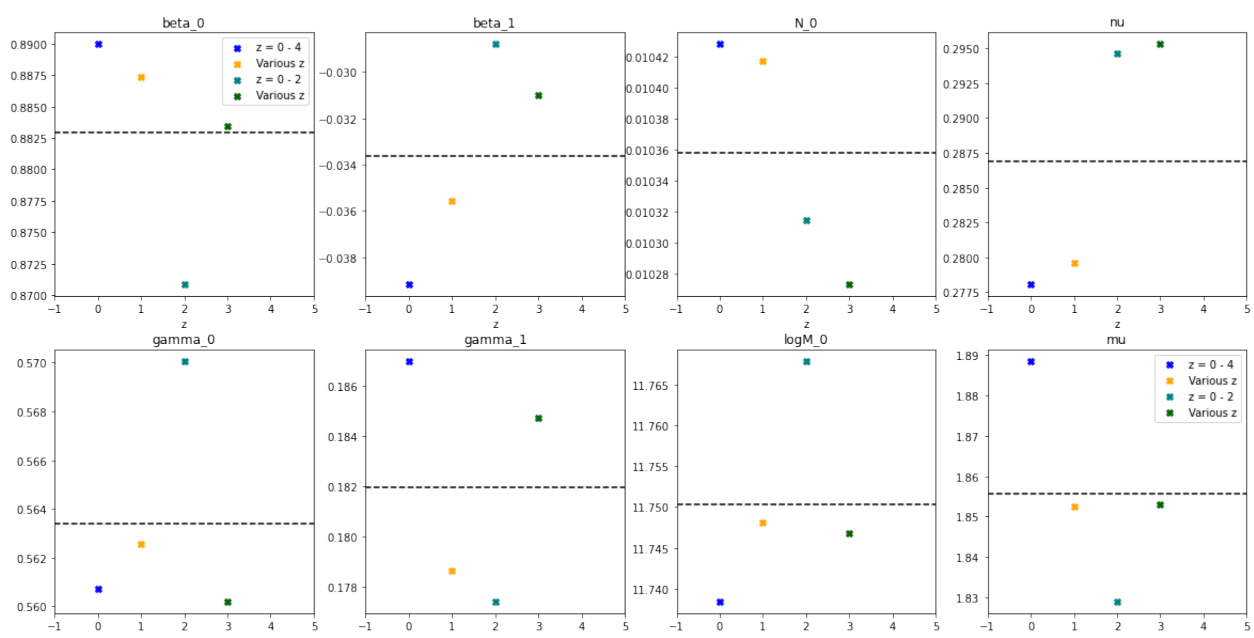

Figure 16: Average parameter values over many different runs, the overall average values can be seen in the last column of Table 4.2.

4.3 SMF with SF-Fraction

See section 3.1 for the introduction to what is being fit here.

I've tried a number of different redshift parametrizations here for the SF_{fract}. First I used (model4 in emma.py):

$$\begin{aligned} a &= a_2 \cdot (z - a_1) + a_0 \\ b &= b_2 \cdot (z - b_1) + b_0 \\ c, d &= \text{constants} \end{aligned} \tag{14}$$

An example of this fit can be seen in Fig 17, using sf_smf_07_08_06-58_0.0-3.0 files in SMF MCMC output. This fit is pretty good for low z, but at higher z for high mass, all the fits are too low as well the populations cross where they should not. I think this was partially because I was using the default DPL SMF values, and not the best fit values from the previous section.

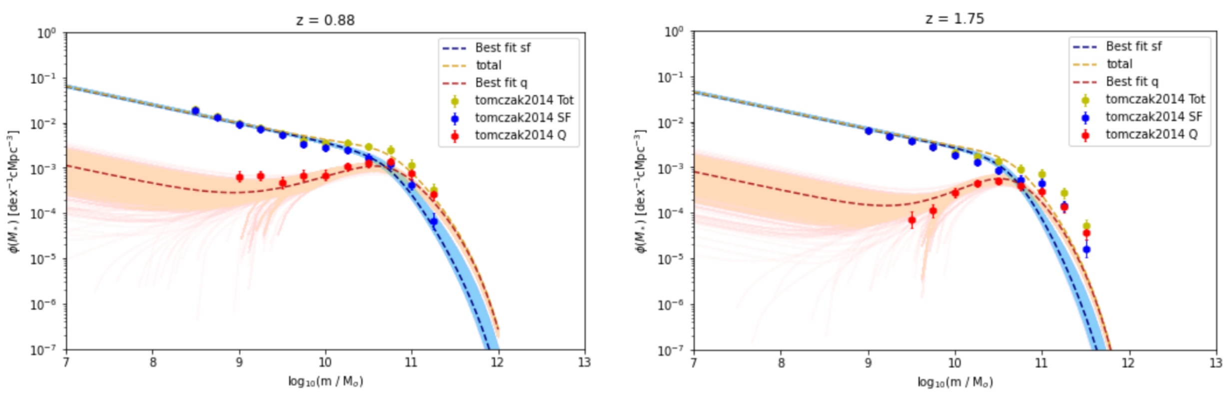

Figure 17: SF_{fract} fit results for Equ 14, with data and best likelihood parameter models for both star-forming and quiescent populations. Best fit values: a_0 = -1.22, a_1 = 0.62, a_2 = -0.36, b_0 = -10.65, b_1 = -0.025, b_2 = -0.0329, c = 2.13, d = 1.09.

To give more flexibility to c, d as well as less linear dependence on z, I then changed the equation to (model1 in emma.py):

$$\begin{aligned} a &= a_0 \cdot (z + 1)^{a_1} \\ b &= b_0 \cdot (z + 1)^{b_1} \\ c &= c_1 \cdot z + c_0 \\ d &= d_1 \cdot z + d_0 \end{aligned} \tag{15}$$

This improved the curvature at the lower mass, but not at the higher mass end, see Fig 18, using sf_smf_07_08_08-19_0.0-3.0 files in SMF MCMC output.

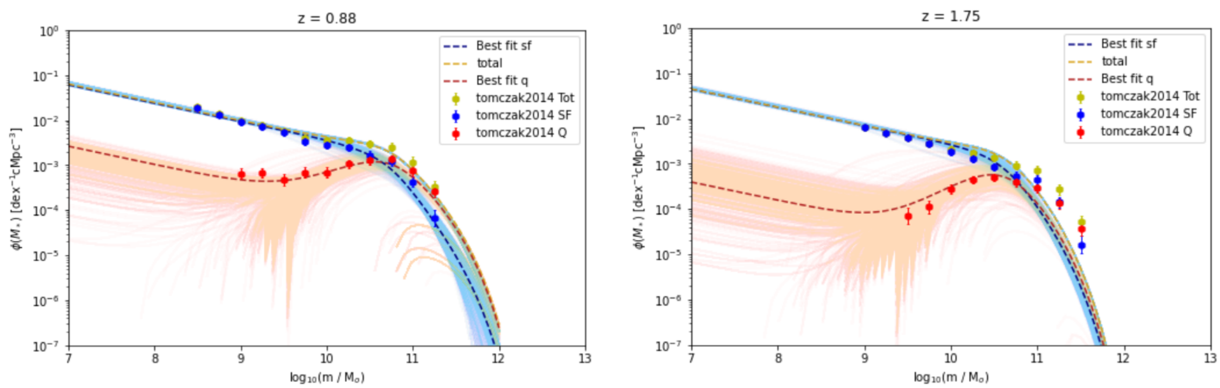

Figure 18: SF_{fract} fit results for Equ 15, with data and best likelihood parameter models for both star-forming and quiescent populations. Best fit values: a_0 = -1.18, a_1 = 0.42, b_0 = -10.7, b_1 = -0.022, c_0 = 1.71, c_1 = 0.88. d_0 = 0.537, d_1 = 0.96

I though the problem with all the models being too low at the high mass end was caused by the parameters I was using for the SMF double power law. The two previous fits were done with the default parameters for Equ. 7, and not the best fit values found in section 4.2. However, with using the best fit parameters, the fits did not turn out very good and I'm not really sure why as the SMF model, especially for the higher mass values is much closer to the data. I also tried these fits for Equ 15 as well as all power laws (see sf_smf_10_08_09-49_0.0-3.0 files in SMF MCMC output), but both of these fits gave parameters that made $SF_{fract} = 1$ for all masses so there was no quiescent galaxies. This may be a fault in the SF_{fract} function that only appeared in these fits, or more fundamentally in the SMF.

5 Miscellaneous Scripts

DataCompile.ipynb: Script to read and format data from "behroozi-2013-data-compilation" for more SMF and SSFR lit values [3].

UsefulHODFunctions.py: To hold small functions I'm using in various files for analysis or getting lit values.

5.1 Cedar Scripts - CedarScripts/

I wrote a whole bunch of bash scripts for MCMC batch runs on Cedar as well as looking at/saving the results.

Script	**Input**	**Function**
test_job_new.sh, runMCMC.sh		Run a MCMC
name_results.sh	slurm-#.out	Prints out name and parms of a run given ID
runMCMC_save.sh		Run MCMC, plot, put params in output and save to csv like file
2runMCMC_save.sh	General file name	Puts params in output and save to csv like file
Arr1_runMCMC_save.sh		Runs array batch, but repeating only one file.
Arr_runMCMC_save.sh		Runs array batch, using file: ./ArrScript/input_#.txt

Table 3: Bash scripts

Script	**Input**	**Function**
MCMC_fitScript.py	General file name, z	Sets up and runs MCMC for SMF
MCMC_results.py	General file name, z	Plots smf for given z, trig, walks and likelihood for one param
MCMC_fitScript_new.py	General file name, min z, max z	Sets up and runs MCMC for SMF with specified min and max z
MCMC_results_new.py	General file name, z	Plots smf for given z, prints out best fit params
params_results.py	General file name	Prints out best fit params
MCMC_fitScript_M.py		Sets up and runs MCMC for SMF with specified min and max z for new μ equation
fractSMF_fitScript.py		Sets up and runs MCMC for SF fraction with specified min and max z
FR_MCMC_results.py	General file name, z	Plots smf from SF fraction fit for given z, trig, walks and likelihood for one param

Table 4: Python scripts

6 Notes

Future Work

- Adding quiescent populations option for the generalized luminosity function.
- Improving fits for the SMF, particularly for higher masses
- Get more data to fit SF_{fract}, as well as improve the equations used for redshift parametrization
- Running MCMCs for the SMF and SF_{fract} at the same time

References

[1] B. P. Moster, R. S. Somerville, C. Maulbetsch, F. C. van den Bosch, A. V. Macciò, T. Naab, and L. Oser, "CONSTRAINTS ON THE RELATIONSHIP BETWEEN STELLAR MASS AND HALO MASS AT LOW AND HIGH REDSHIFT," *The Astrophysical Journal*, vol. 710, no. 2, pp. 903–923, jan 2010. [Online]. Available: https://doi.org/10.1088%2F0004-637x%2F710%2F2%2F903 2, 12

[2] J. S. Speagle, C. L. Steinhardt, P. L. Capak, and J. D. Silverman, "A highly consistent framework for the evolution of the star-forming "main sequence" from z 0-6," *The Astrophysical Journal Supplement Series*, vol. 214, no. 2, p. 15, Sep 2014. [Online]. Available: http://dx.doi.org/10.1088/0067-0049/214/2/15 4

[3] P. S. Behroozi, R. H. Wechsler, and C. Conroy, "The average star formation histories of galaxies in dark matter halos fromz= 0-8," *The Astrophysical Journal*, vol. 770, no. 1, p. 57, May 2013. [Online]. Available: http://dx.doi.org/10.1088/0004-637X/770/1/57 6, 7, 19, 23

[4] A. R. Tomczak, R. F. Quadri, K.-V. H. Tran, I. Labbé, C. M. S. Straatman, C. Papovich, K. Glazebrook, R. Allen, G. B. Brammer, G. G. Kacprzak, and et al., "Galaxy stellar mass functions from zfourge/candels: An excess of low-mass galaxies sincez= 2 and the rapid buildup of quiescent galaxies," *The Astrophysical Journal*, vol. 783, no. 2, p. 85, Feb 2014. [Online]. Available: http://dx.doi.org/10.1088/0004-637X/783/2/85 7

[5] Y.-j. Peng, S. J. Lilly, K. Kovač, M. Bolzonella, L. Pozzetti, A. Renzini, G. Zamorani, O. Ilbert, C. Knobel, A. Iovino, and et al., "Mass and environment as drivers of galaxy evolution in sdss and zcosmos and the origin of the schechter function," *The Astrophysical Journal*, vol. 721, no. 1, p. 193–221, Aug 2010. [Online]. Available: http://dx.doi.org/10.1088/0004-637X/721/1/193 8

A Appendix

Data files from Peter Behroozi, main sources listed in detail in [3].

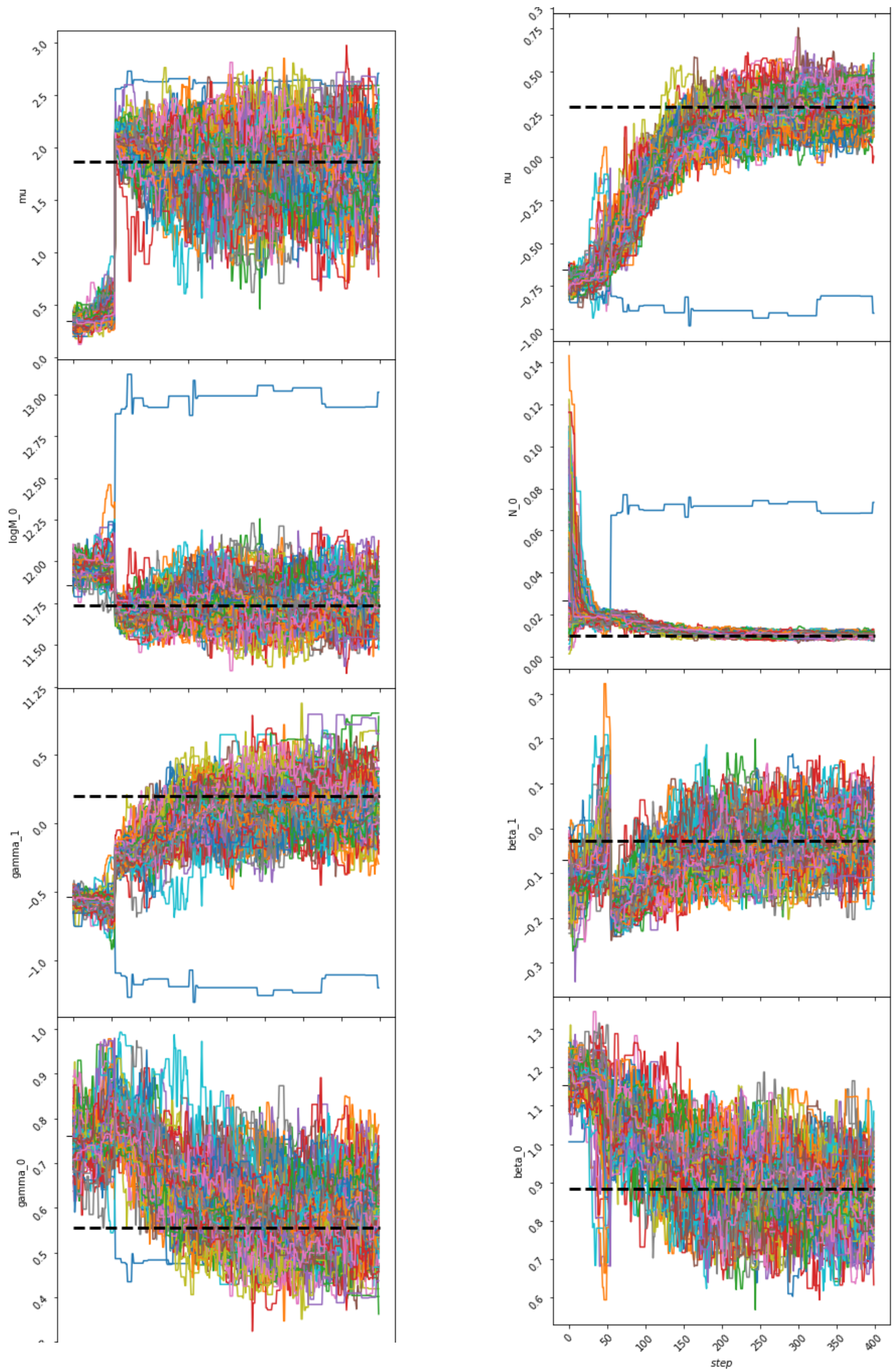

Figure 19