First look at Constantin's cosmological parameter fitting code

Narei (with help from Vincent)

April 2nd 2024

Code explanation

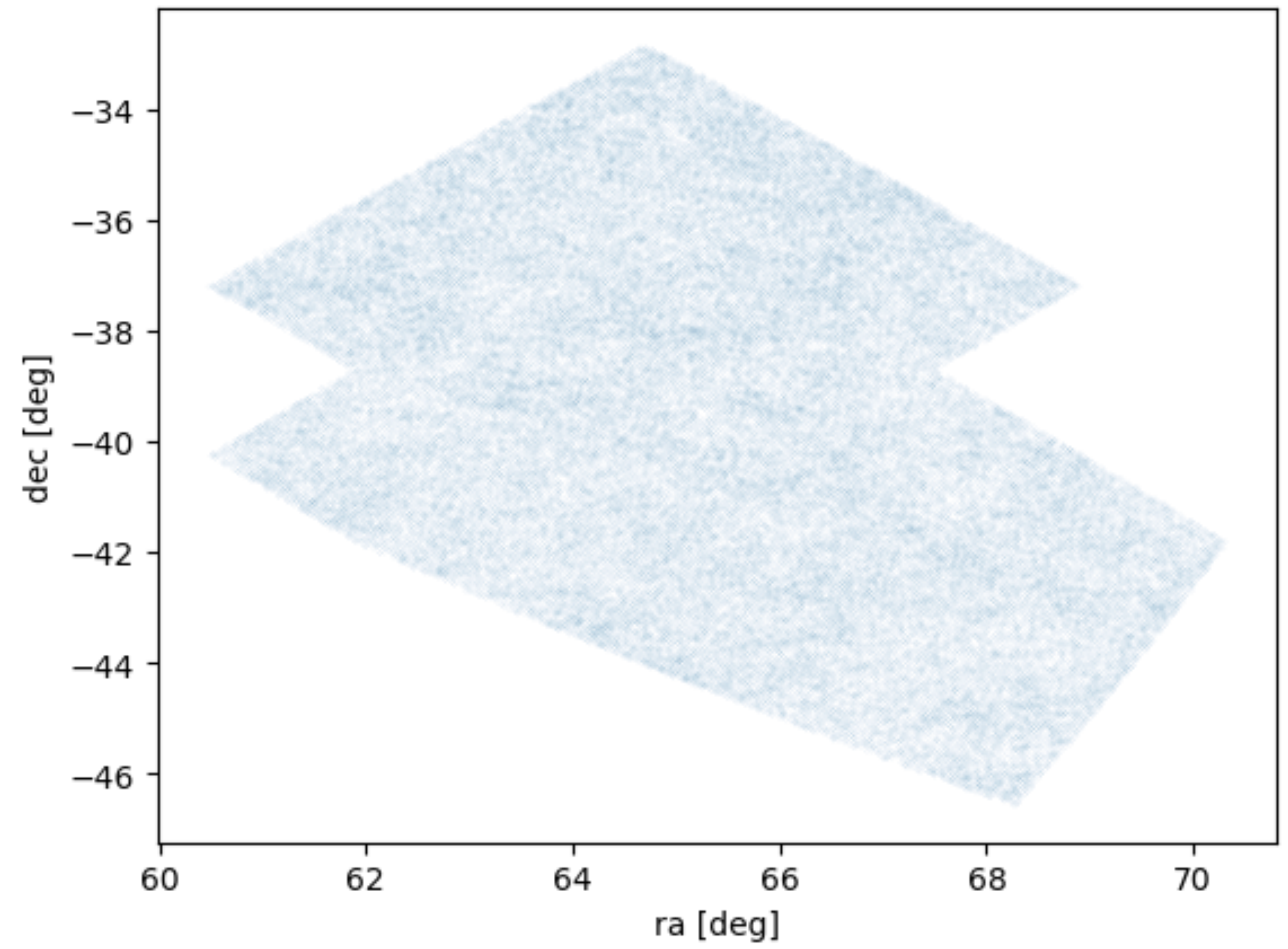
Constantin's code: https://github.com/LSSTDESC/CLCosmo_Sim/blob/main/notebooks/produce_chi2_map.ipynb

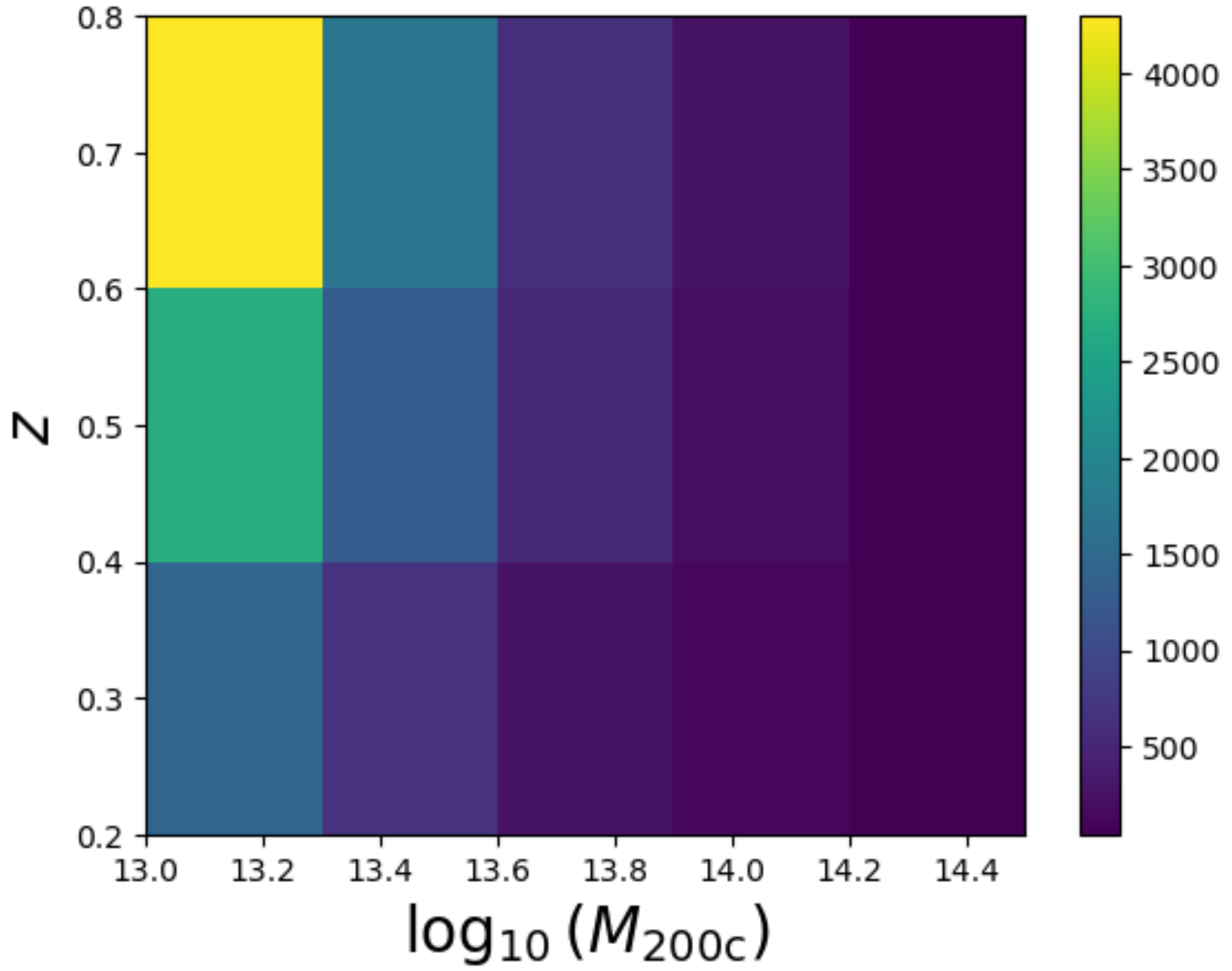
1. Building the binned observed number of clusters

How: Take a given simulated catalog (here *skysim_v1.1.1_small*), perform bins in redshift and masseur clusters (*logm_bin:{13, 13.3, 13.6, 13.9, 14.2, 14.5}* ; *z_bin{0.2, 0.4, 0.6, 0.8}* and create a 2D map from that

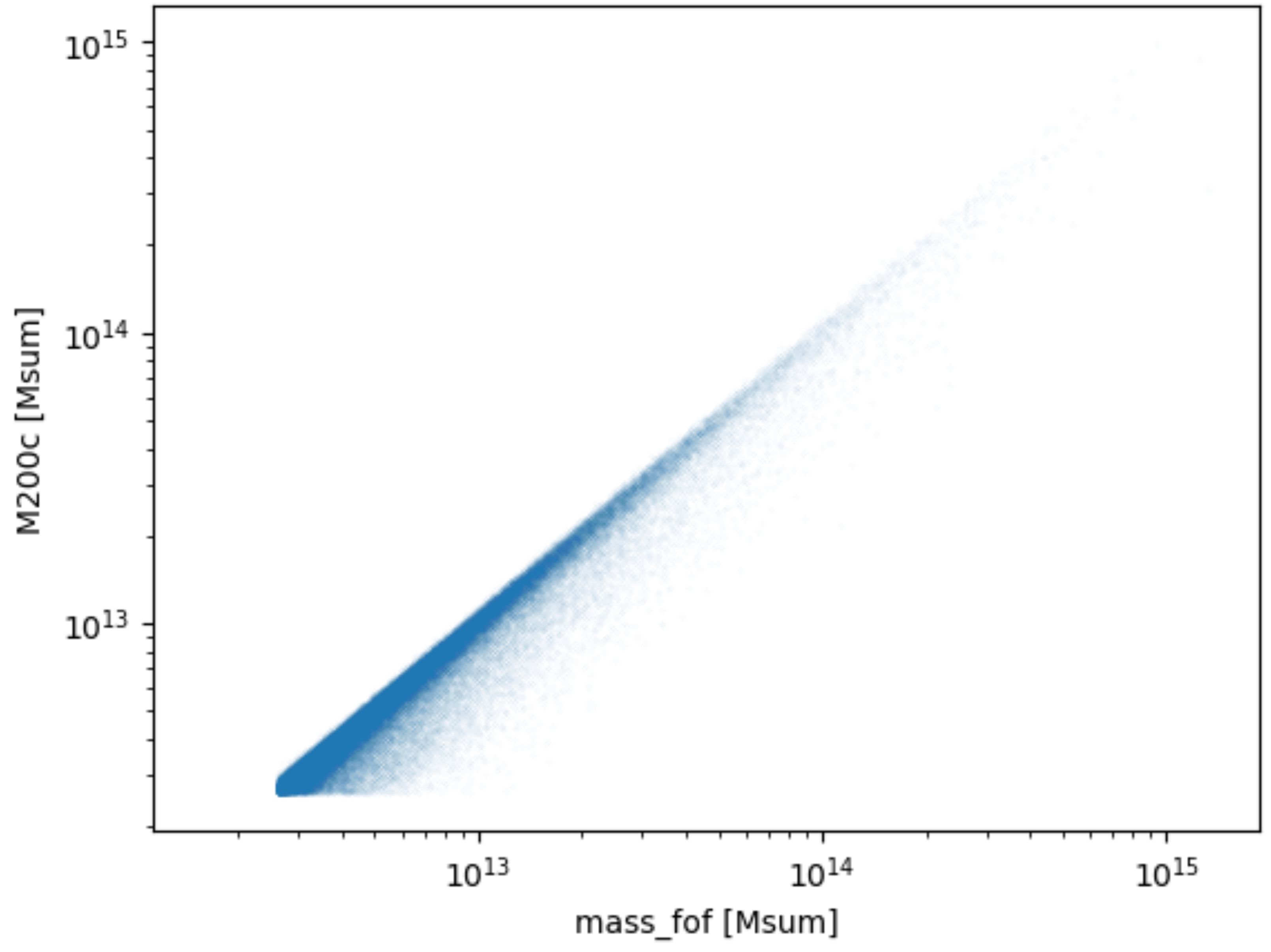
- **why not using larger/smaller z and masses ?**
- **using here M200c, could use also mass_fof**

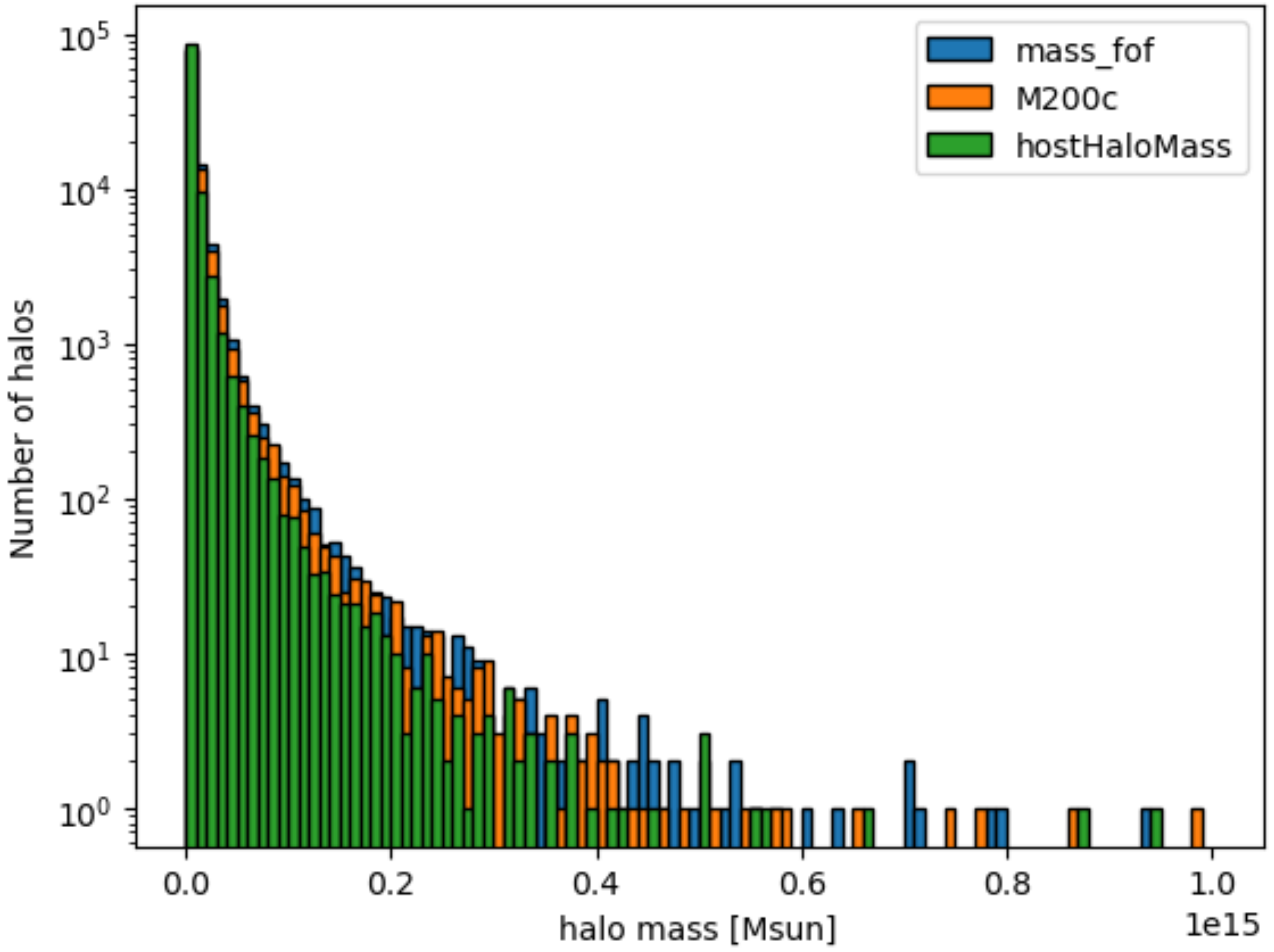
–> This provides the « data » sample

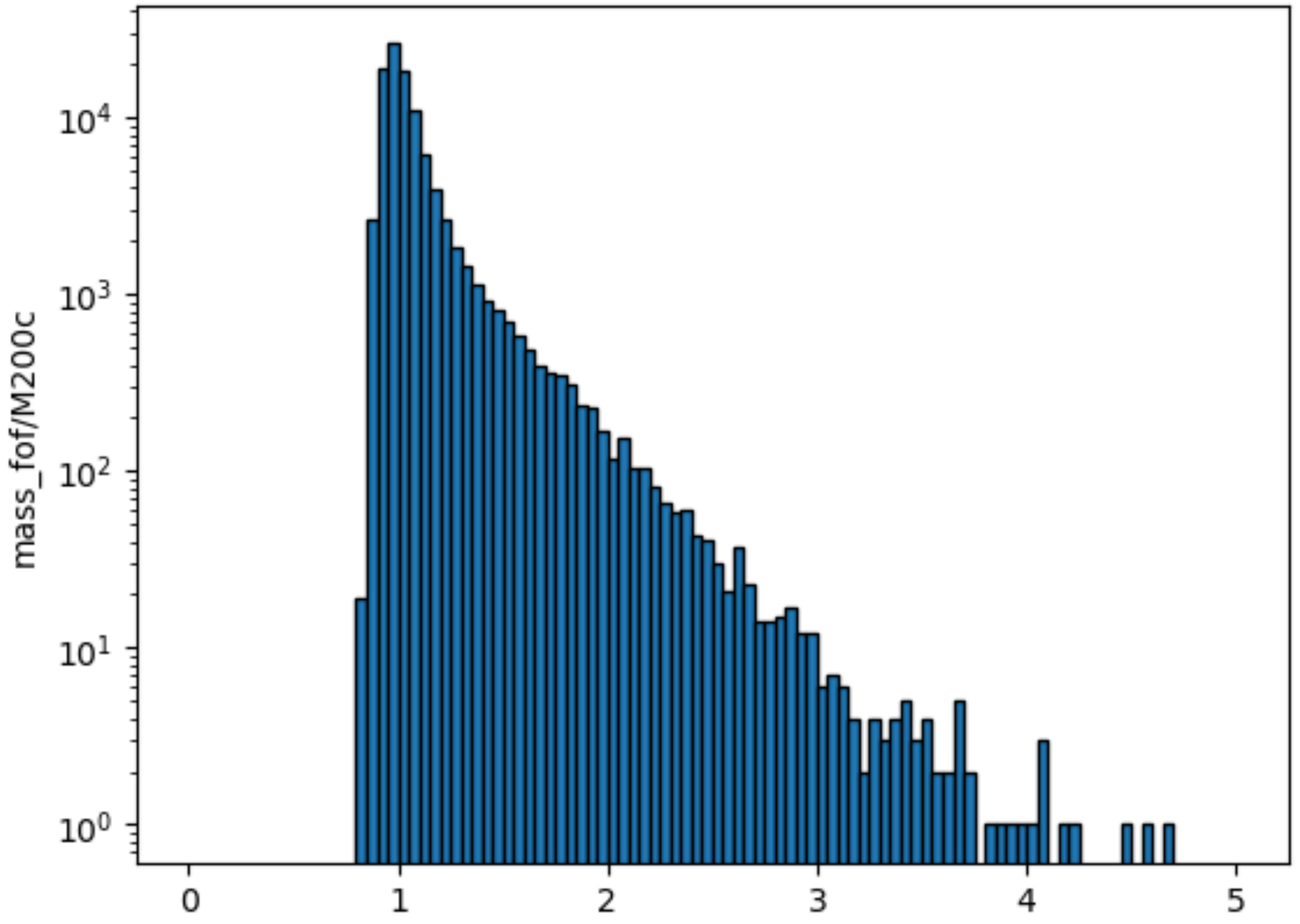




Mass differences







- What is hostHaloMass ?

Code explanation

2. Building the binned predicted number of clusters

How: using the formula below, and computing each term (differential comoving volume

renormalise to full sky area (here we have only a partial sky)

$$N_{ij}^{\mathrm{th}} = \Omega_{\mathrm{SkySim}} \int_{z_i}^{z_{i+1}} dz \frac{dV(z)}{dz d\Omega} \int_{\log_{10} M_j}^{\log_{10} M_{j+1}} \frac{dn(M,z)}{d\log_{10} M} d\log_{10} M$$

integration on redshift
differential comoving volume

integration on mass
Mass function

Comoving volume

- Definition found in: http://arxiv.org/abs/astro-ph/9905116v4
- Depends on which cosmology you assume !

Hubble distance

$$D_H \equiv \frac{c}{H_0} = 3000\, h^{-1} \text{ Mpc} = 9.26 \times 10^{25}\, h^{-1} \text{ m}$$

angular diameter distance
DM=DC for Ωk=0

$$D_A = \frac{D_M}{1+z}$$

$$dV_C = D_H \frac{(1+z)^2\, D_A^2}{E(z)}\, d\Omega\, dz$$

line-of-sight comoving distance, h/H0

$$E(z) \equiv \sqrt{\Omega_M\,(1+z)^3 + \Omega_k\,(1+z)^2 + \Omega_\Lambda}$$

Chosen cosmology

- Cosmology of skysim5000 (or cosmoDC2)
  - taken from OuterRim -> supposedly WMAP-7 combined results (https://arxiv.org/abs/1904.11970)
  - Parameters found in simu vs parameters found in literature:
    - **not sure to understand why the skysim parameters are different ?**

	sskysim	Planck 2015	2018	WMAP combined
Om0	0.2648	0.3089	0.315	0.236
Ob0	0.0448	0.0486	-	0.0461
sigma80	0.8	0.8159	0.811	0.817
h0	0.71	0.677	0,674	0.697
ns0	0.963	0.9667	0.965	0.9646

WMAP 7-year Cosmological Interpretation

TABLE 1
SUMMARY OF THE COSMOLOGICAL PARAMETERS OF ΛCDM MODEL[a]

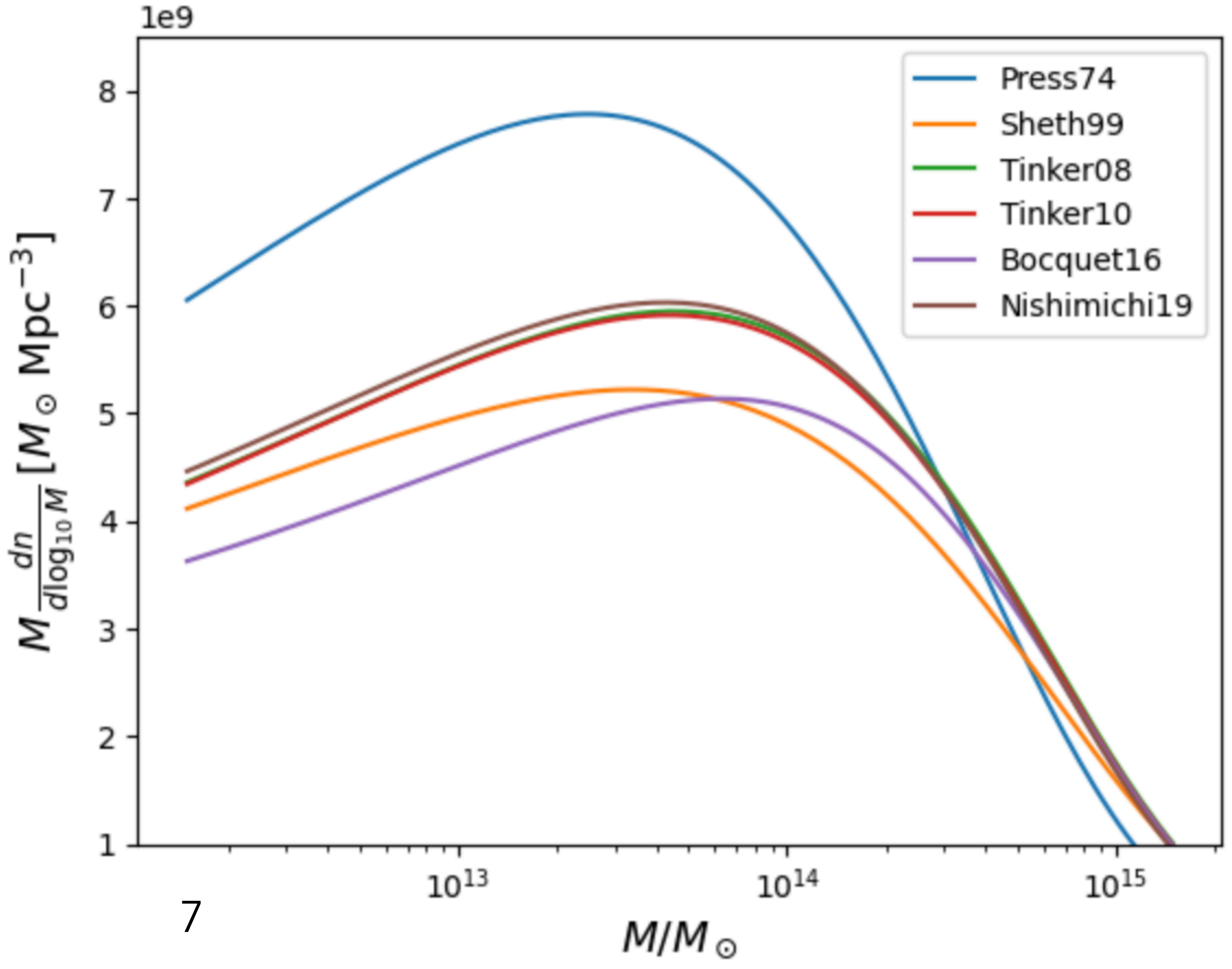
Class	Parameter	*WMAP* 7-year ML[b]	*WMAP*+BAO+H_0 ML	*WMAP* 7-year Mean[c]	*WMAP*+BAO+H_0 Mean
Primary	$100\Omega_b h^2$	2.227	2.253	$2.249^{+0.056}_{-0.057}$	2.255 ± 0.054
	$\Omega_c h^2$	0.1116	0.1122	0.1120 ± 0.0056	0.1126 ± 0.0036
	Ω_Λ	0.729	0.728	$0.727^{+0.030}_{-0.029}$	0.725 ± 0.016
	n_s	0.966	0.967	0.967 ± 0.014	0.968 ± 0.012
	τ	0.085	0.085	0.088 ± 0.015	0.088 ± 0.014
	$\Delta^2_{\mathcal{R}}(k_0)$[d]	2.42×10^{-9}	2.42×10^{-9}	$(2.43 \pm 0.11) \times 10^{-9}$	$(2.430 \pm 0.091) \times 10^{-9}$
Derived	σ_8	0.809	0.810	$0.811^{+0.030}_{-0.031}$	0.816 ± 0.024
	H_0	70.3 km/s/Mpc	70.4 km/s/Mpc	70.4 ± 2.5 km/s/Mpc	70.2 ± 1.4 km/s/Mpc
	Ω_b	0.0451	0.0455	0.0455 ± 0.0028	0.0458 ± 0.0016
	Ω_c	0.226	0.226	0.228 ± 0.027	0.229 ± 0.015
	$\Omega_m h^2$	0.1338	0.1347	$0.1345^{+0.0056}_{-0.0055}$	0.1352 ± 0.0036
	$z_{\rm reion}$[e]	10.4	10.3	10.6 ± 1.2	10.6 ± 1.2
	t_0[f]	13.79 Gyr	13.76 Gyr	13.77 ± 0.13 Gyr	13.76 ± 0.11 Gyr

ns0 is the spectral index of primordial fluctuations : how density fluctuations vary with scale. An index of 1 means that the variations are the same on all scale

Mass function

- The number of cluster cat a given mass and redshift bin is provided a posteriori by several authors (Bocquet, Tinker, …)
- Several mass functions are available in ccl libraries (CCL=Core Cosmology Library :github, arxiv)
- Possibility to work with different definition of mass (including mass_fof)

```
# Press & Schechter mass function
hmfs.append(ccl.halos.MassFuncPress74(mass_def=hmd_fof))

# Sheth & Tormen mass function
hmfs.append(ccl.halos.MassFuncSheth99(mass_def=hmd_fof))

# Tinker 2008 mass function
hmfs.append(ccl.halos.MassFuncTinker08(mass_def=hmd_200m))

# Tinker 2010 mass function
hmfs.append(ccl.halos.MassFuncTinker10(mass_def=hmd_200m))

# Bocquet 2016 mass function
hmfs.append(ccl.halos.MassFuncBocquet16(mass_def=hmd_200m))

# Nishimichi 2019 mass function
# To use this mass function you need the dark emulator:
# https://dark-emulator.readthedocs.io/en/latest/
hmfs.append(ccl.halos.MassFuncNishimichi19(mass_def=hmd_200m,extrapolate=True))
```



Tinker mass function

- From Tinker's paper: https://iopscience.iop.org/article/10.1086/591439/pdf

$$\frac{dn}{dM} = f(\sigma)\frac{\bar{\rho}_m}{M}\frac{d\ln\sigma^{-1}}{dM}. \qquad (2)$$

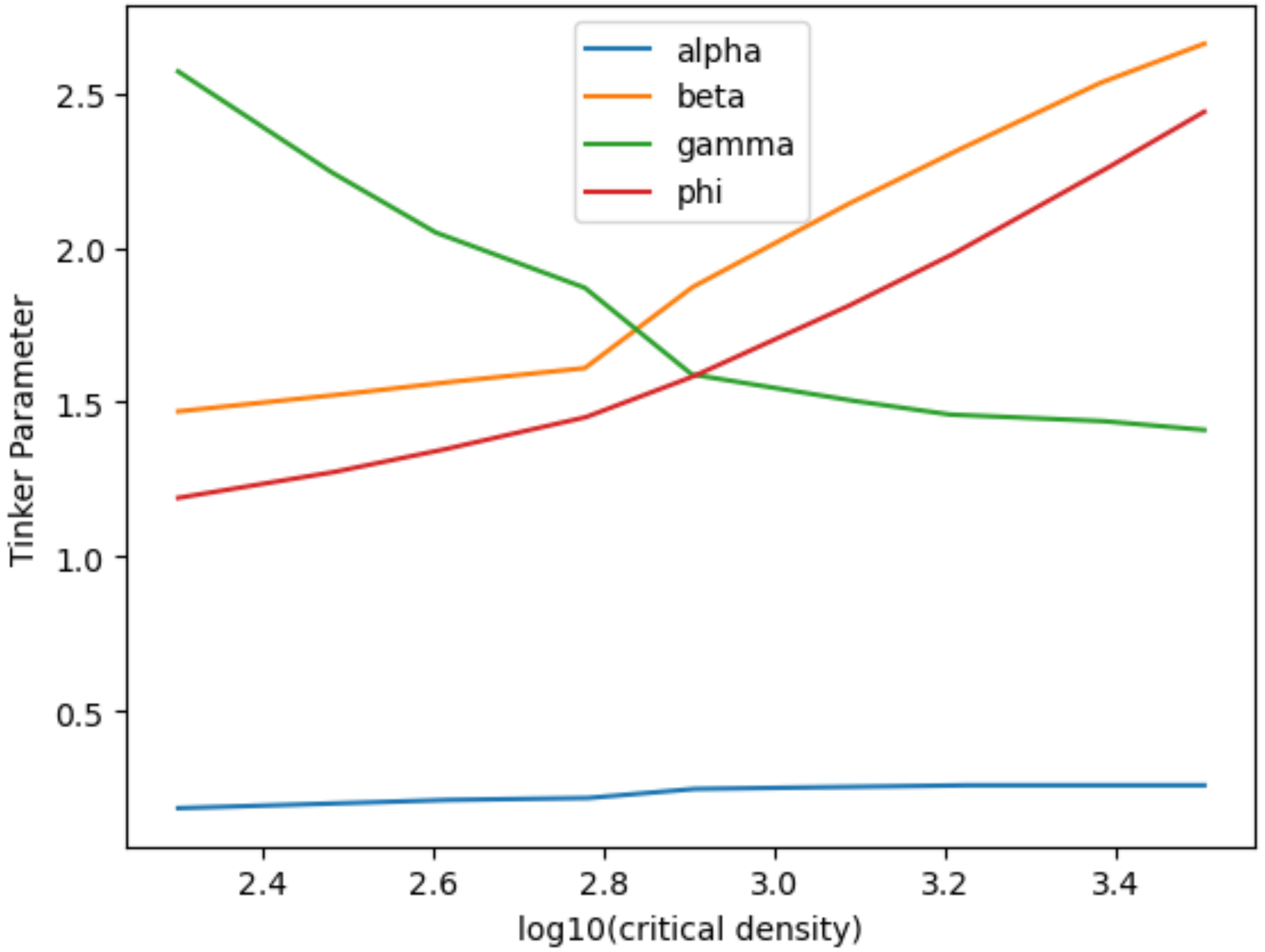
Here, the function $f(\sigma)$ is expected to be universal to the changes in redshift and cosmology and is parameterized as

$$f(\sigma) = A\left[\left(\frac{\sigma}{b}\right)^{-a} + 1\right]e^{-c/\sigma^2} \qquad (3)$$

where

$$\sigma = \int P(k)\hat{W}(kR)k^2dk, \qquad (4)$$

and $P(k)$ is the linear matter power spectrum as a function of wavenumber k, and $\hat{W}$ is the Fourier transform of the real-space top-hat window function of radius R. It is convenient to

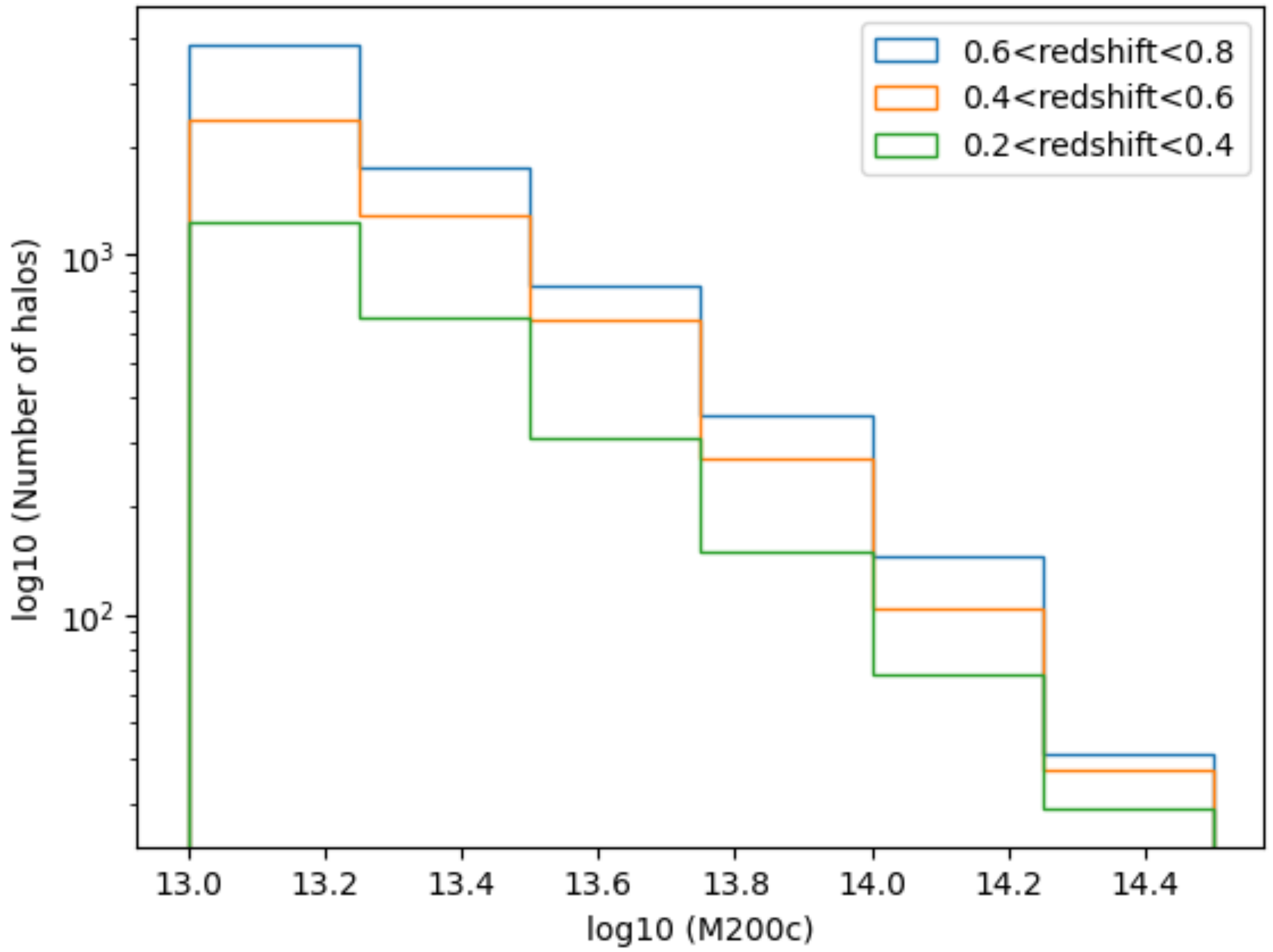


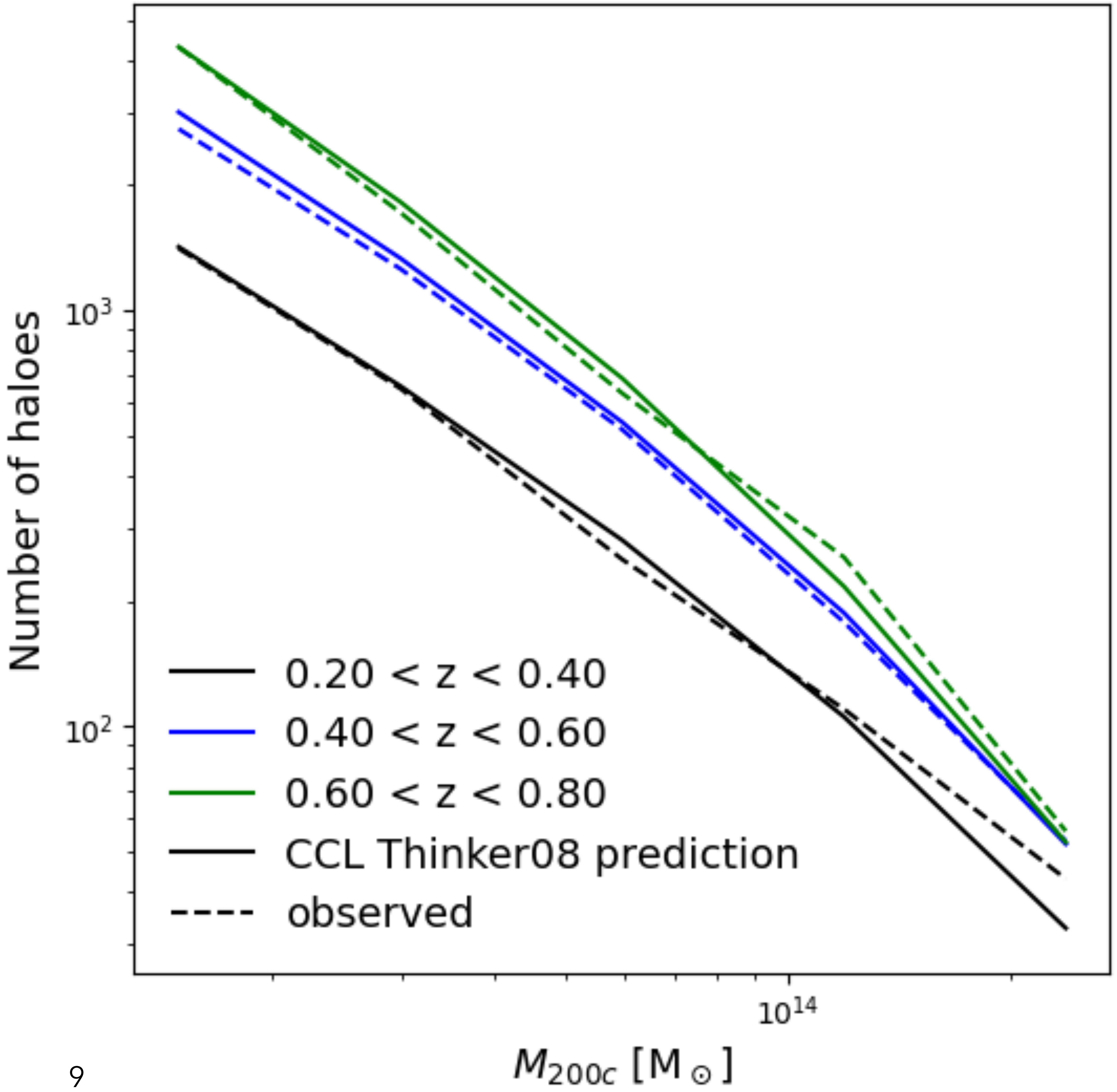
A, a, b, c : are parameters to be fitted (called alpha, beta, gamma, phi here)

« critical density » :
overdensity within a sphere of radius *Rdelta with respect to* the mean density of the Universe at the epoch of analysis

Predicted number of haloes vs « observed »

- « Observed » = from simulation





Building log-likelihood

- Assuming Gaussian or Poisson statistics
- Likelihood: probably that a given event is produced after N trials-> product of single N probabilities (assuming independents events)
- Log likelihood -> the product becomes a sum
- Likelihood made dependent on on sigma8 and OmegaM (since Nth depends on cosmology)
    - all other parameters kept the same !
    - computed for 10x10 points for **sigma8=[0.68-0.9], OmegaM=[0.2-0.3]**

$$P(X = x|\theta) = f(x) = e^{-\theta}\frac{\theta^x}{x!}, \quad x \in \{0, 1, \ldots, \infty\}, \theta > 0$$

$$Log(L)(\Omega_m, \sigma_8) \propto \sum_{i,j} N_{ij}^{\mathrm{obs}} \log(N_{ij}^{\mathrm{th}}) - N_{ij}^{\mathrm{th}}$$

Building the chi2 and plotting it

- The minimization is made using a chi2 = -2lnL
- Between two points (sigma8, omegaM), an interpolation is performed
  - warning: interpolation used by Constantin is now deprecated ! (interp2d). Used instead RectBivariateSpline
- 2D contours are plotted, using levels: levels = list(np.arange(1,4)**2)