Likelihood Non-Gaussianity in Large Scale Structure Analyses

ChangHoon Hahn, et al.
Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley CA 94720, USA
changhoon.hahn@lbl.gov

DRAFT --- 0d87906 --- 2017-12-13 --- NOT READY FOR DISTRIBUTION

ABSTRACT

abstract here

Subject headings: methods: statistical — galaxies: statistics — methods: data analysis — cosmological parameters — cosmology: observations — large-scale structure of universe

1. Introduction

- Talk about the use of Bayesian parameter inference and getting the posterior in LSS cosmology
- Explain the two major assumptions that go into evaluating the likelihood
- Emphasize that we are not talking about non-Gaussian contributions to the likelihood
- Emphasize the scope of this paper is to address whether one of the assumptions matters for galaxy clustering analyses.

2. Gaussian Likelihood Assumption

- Depending on Hogg's paper maybe a simple illustration of how the likelihood asumption

However, as we show in this paper, the assumption of likelihood Gaussianity is not necessary. In fact, we will show that the mock catalogs used in standard LSS analyses to estimate the covariance matrix for evaluating the Gaussian likelihood, can be used to quantify the non-Gaussianity. More important the mock catalogs can be used to construct an accurate estimator for the non-Gaussian likelihood.

3. Mock Catalogs

Mock catalogs are play an indispensable role in standard cosmological anslyses of LSS studies. They're used for testing analysis pipelines (Beutler et al. 2017; Grieb et al. 2017; Tinker & et al. in preparation), testing the effect of systematics (Guo et al. 2012; Vargas-Magaña et al. 2014; Hahn et al. 2017; Pinol et al. 2017; Ross et al. 2017), and, most relevantly for this paper, estimating the covariance matrix (Parkinson et al. 2012; Kazin et al. 2014; Grieb et al. 2017; Alam et al. 2017; Beutler et al. 2017; Sinha et al. 2017). In fact, nearly all current state-of-the-art LSS analyses use covariance matrices estimated from mocks to evaluate the likelihood for parameter inference.

While some argue for analytic estimates of the covariance matrix (e.g. Mohammed et al. 2017) or estimates directly from data by subsampling (e.g. Norberg et al. 2009), covariance matrices from mocks have a number of advantages. Mock catalogs allow us to incorporate detailed systematic errors present in the data and variance beyond the volume of the data. Even for analytic estimates, large ensembles of mocks are crucial for validataion (Slepian et al. 2017). Moreover, as we show later in this paper, mock catalogs present an additional advantage: they allow us to quantify the non-Gaussianity of the likelihood and more accurately estimate the true likelihood.

In this paper, we we focus on two LSS analyses: the powerspectrum multipole full shape analysis of Beutler et al. (2017) and group multiplicity function analysis of Sinha et al. (2017). Throughout the paper we will make extensive use of the mock catalogs used in these analyses. In this section, we give a brief description of these mocks and how the observables used in the analysis — the powerspectrum multipole (P_ℓ) and group multiplicity function ($\zeta(N)$) — are calculated from them. Afterwards, we will describe how we compute the covariance matrix, $\mathbb{C}$, and the pre-processed mock observable date, $\mathbf{X}^{\rm mock}$.

3.1. MultiDark-PATCHY Mock Catalog

In their powerspectrum multipole full shape analysis, Beutler et al. (2017) use the MultiDark-PATCHY mock catalogs from Kitaura et al. (2016). These mocks are generated using the PATCHY code (Kitaura et al. 2014, 2015). They rely on large-scale density fields generated using augmented Lagrangian Perturbation Theory (ALPT; Kitaura & Heß 2013) on a mesh. This mesh is then populated with galaxies based on a combined non-linear deterministic and stochastic biases. The mocks from the PATCHYcode are then calibrated to reproduce the galaxy clustering in the high-fidelity BigMultiDark N-body simulation (Rodríguez-Torres et al. 2016; Klypin et al. 2016).

The galaxies are then assigned stellar masses using the HADRON code (Zhao et al. 2015). And the SUGAR code (Rodríguez-Torres et al. 2016) is applied to combine different boxes, incorporate selection effects and masking to produce mock light-cone galaxy catalogs. The

statistics of the resultng mocks are then compared to observations and the process is iterated to reach desired accuracy. We refer readers to Kitaura et al. (2016) for further details.

In total, Kitaura et al. (2016) generated a 12,228 mock light-cone galaxy catalogs for BOSS Data Release 12: 2048 for each southern and northern galactic caps of LOWZ, CMASS, combined samples. In Beutler et al. (2017), they use 2045 and 2048 for the northern galactic cap (NGC) and souther galactic cap (SGC) of the LOWZ+CMASS combined sample. Beutler et al. (2017) excluded 3 mock realizations, due to notable issues, which have been since been addressed. Therefore, in our analysis we use all 2048 mocks for both the NGC and SGC of the LOWZ+CMASS combined sample.

3.2. Sinha et al. (2017) Mocks

The simulations used in the small-scale clustering analysis of Sinha et al. (2017) are from the Large Suite of Dark Matter Simulations project (LasDamas; McBride et al. 2009). More specifically Sinha et al. (2017) uses the Consuelo and Carmen configurations, which were designed to model SDSS galaxies with M_r thresholds of -19 and -21, respectively. The initial conditions for these simulations are derived from second order Lagrangian Perturbation Theory using the 2LTPIC code (Scoccimarro 1998; Crocce et al. 2006) and evolved using the N-body `GADGET` -2 code (Springel 2005). Halos are then identified from the dark matter distribution outputs using the `ntropy` $-$ `fofsv` code (Gardner et al. 2007), which uses a friend-of-friends algorithm (FoF; Davis et al. 1985) with a linking length of 0.2 times the mean inter-particle separation. The FoF halo masses are then adjusted using the Warren et al. (2006) correction. The Consuelo simulation contains 1400^3 dark matter particles with mass of $1.87\times10^9\,h^{-1}M_\odot$ in a cubic volume of $420\,h^{-1}Mpc$ per side and is evolved from $z_{\rm init}=99$. The Carmen simulation contains 1120^3 dark matter particles with mass of $4.938\times10^{10}\,h^{-1}M_\odot$ in a cubic volume of $1000\,h^{-1}Mpc$ per side and is evolved from $z_{\rm init}=49$. They use the following cosmological parameters, which are motivated by the WMAP3 constraints (Spergel et al. 2007): $\Omega_m=0.25, \Omega_\Lambda=0.75, \Omega_b=0.04, h=0.7, \sigma_8=0.8,$ and $n_s=1.0$.

The FoF halo catalogs are then populated with galaxies using the 'Halo Occupation Distribution' (HOD) framework. In this framework, the number, positions, and velocities of galaxies are described statistically by an HOD model. Sinha et al. (2017) adopts the 'vanilla' HOD model of Zheng et al. (2007), where the mean number of central and satellite galaxies are described by the halo mass and five HOD parameters: $M_{\rm min}, \sigma_{\log M}, M_0, M_1,$ and α. Finally, once the simulation boxes are populated with galaxies, observational systematic effects are imposed. The peculiar velocities of galaxies are used to impose redshift-space distortions. And galaxies that lie outside the redshift limits or sky footprint of the SDSS sample are removed. For further details regarding the LasDamas simulations or mock catalogs, we refer readers to Sinha et al. (2017).

To calculate their covariance matrix, Sinha et al. (2017) produced 200 independent mock

catalogs from 50 simulations using a single set of HOD model parameters. To take advantage of the methods we present in this work (Sections **ref**), we require a large number of mock catalogs. Our methods rely on sampling multidimensional distributions, so incorporating more mocks into the analysis drastically improves their accuracy. Therefore, we utilize an additional 19,800 mock catalogs made from the procedure. These mocks are not generated using the same set of HOD model parameters, but 200 mocks each from 99 sets of HOD parameters sampled from the MCMC chain used to produce the posterior probability distriubtion presented in Sinha et al. (2017).

3.3. Mock Observable $\mathbf{X}^{\rm mock}$ and Covariance Matrix $\mathbb{C}$

In Beutler et al. (2017) and Sinha et al. (2017) they analyze the powerspectrum multipoles measured from the BOSS DR12 galaxies and the group multiplicity function measured from the SDSS DR7 galaxies, respectively. To get from the mock catalogs described above to the covariance matrices used in these analyses, the observables must be consistently measured for each mock catalog. Below we briefly describe how $P_\ell(k)$ and $\zeta(N)$ and their covariance matrices are measured in Beutler et al. (2017) and Sinha et al. (2017). Furthermore, we describe how we pre-process the mock observables to be used in the methods we describe in the next sections.

To measure the powerspectrum multipoles of the BOSS DR12 galaxies and the MutliDark-PATCHY mocks (Section 3.1), Beutler et al. (2017) uses a fast fourier transform (FFT) based anisotropic powerspectrum estimator based on Bianchi et al. (2015) and **?**. This estimator estimates the monopole, quadrupole, and hexadecapole ($\ell = 0, 2,$ and4) of the powerspectrum using FFTs of the overdensity field multipoles for a given survey geometry. By using FFTs rather than counting all galaxy pairs, the estimator significantly reduces the computational costs to $\mathcal{O}(N \log N)$, where N is the number of grid cells used to bin the galaxy data. The powerspectrum multipoles are calculated in bins of $\Delta k = 0.01\,h{\rm Mpc}^{-1}$. The powerspectrum monopole and quadrupole are computed over the range $k = 0.01 - 0.15\,h{\rm Mpc}^{-1}$ while the hexadecapole is computed over the range $k = 0.01 - 0.10\,h{\rm Mpc}^{-1}$. For further details on the estimator we refer readers to Section 3 of Beutler et al. (2017).

Using the $P_\ell(k)$ of the MultiDark-PATCHY mocks, Beutler et al. (2017) then computes the covariance matrix of all multipoles as

$$\mathbb{C}_{xy} = \frac{1}{N_{\rm mock} - 1} \sum_{n=1}^{N_{\rm mock}} [P_{\ell,n}(k_i) - \overline{P}_\ell(k_i)] \times [P_{\ell',n}(k_j) - \overline{P}_{\ell'}(k_j)]. \quad (1)$$

$N_{\rm mock}$ is the number of mock catalogs and $\overline{P}_\ell$ is the mean of the mock powerspectra:

$$\overline{P}_\ell(k_i) = \frac{1}{N_{\rm mock}} \sum_{n=1}^{N_{\rm mock}} P_{\ell,n}(k_i). \quad (2)$$

The (x, y) element of $\mathbb{C}$ is given by $(x, y) = (n_b\frac{\ell}{2} + i, n_b\frac{\ell'}{2} + j)$, where n_b is the number of bins in each multipole power spectrum ($n_b = 14$ for the monopole and quadrupole; $n_b = 9$ for the hexadecapole). $\mathbb{C}$ is a 37×37 matrix.

In this work, we compute the $P_\ell(k)$ of the MultiDark-PATCHY mocks, using a similar FFT-based estimator of Hand et al. (2017) instead of the estimator in Beutler et al. (2017). Our choice was based on computational convenience. A python implementation of the Hand et al. (2017) estimator is publicly available in the `NBODYKIT` package[3]. We emphasize that the resulting $P_\ell(k)$s and covariance matrix from the two estimators have been confirmed to be consistent with one another.

Next, for the Sinha et al. (2017) group multiplicity function analysis, they start with the Berlind et al. (2006) friend-of-friend algorithm to identify groups in the SDSS and mock data. According to the algorithm, a galaxy pair is assigned to the same group if their projected and line-of-sight separations are both less than the corresponding linking length. Sinha et al. (2017) adopts the Berlind et al. (2006) linking lengths: $b_\perp = 0.14$ and $b_\parallel = 0.75$. These linking lengths are in units of mean inter-galaxy separation $n_g^{-1/3}$, where n_g is the number density of the sample. In comoving lengths, the linking lengths for the SDSS DR7 $M_r < -19$ sample correspond to $(r_\perp, r_\parallel) = (0.57, 3.05)h^{-1}$Mpc. Once the groups are identified in the SDSS and mock data, $\zeta(N)$ is derived by calculating the comoving number density of groups in bins of richness N — the number of galaxies in the group. For the $M_r < 19$ SDSS sample, Sinha et al. (2017) uses eight N bins: $(5-6), (7-9), (10-13), (14-19), (20-32), (33-52), (53-84), (85-220)$. For further details on the GMF calculation, we refer readers to Section 4.2 of Sinha et al. (2017).

From the GMFs of each mocks, Sinha et al. (2017) computes the covariance matrix for the analysis as

$$\mathbb{C}_{ij} = \frac{1}{N_{\rm mock} - 1} \sum_{n=1}^{N_{\rm mock}} [\zeta_n(N_i) - \bar{\zeta}(N_i)] \times [\zeta_n(N_j) - \bar{\zeta}(N_j)]. \tag{3}$$

In Sinha et al. (2017), they compute the covariance matrix using 200 mocks generated using a single fiducial set of HOD parameters. However, as we mention in Section 3.2, in this paper we use 20,000 mocks from 100 different set of HOD parameters sampled from the MCMC chain. Consistent with this, the GMF covariance matrix we use in this paper is computed as Eq. 3 but with the $N_{\rm mock} = 20,000$ mocks.

In the rest of this paper we investigate the non-Gaussianity of the likelihood and its impact on parameter inference for both Beutler et al. (2017) and Sinha et al. (2017). In order to discuss the two separate analyses in a consistent manner, we define the matrix $\mathbf{D}^{\rm mock}$ of the

[3]http://nbodykit.readthedocs.io/en/latest/index.html

mock observables (P_ℓ and ζ) as follows.

$$\mathbf{D}^{\rm mock} = \{\mathbf{D}_n^{\rm mock}\} \tag{4}$$

where $\mathbf{D}_n^{\rm mock} = [P_0(k)_n, P_2(k)_n, P_4(k)_n]$ for Beutler et al. (2017) and $\mathbf{D}_n^{\rm mock} = \zeta(N)_n$ for Sinha et al. (2017). The $\mathbf{D}^{\rm mock}$ matrix has dimensions of $N_{\rm mock} \times N_{\rm bin}$, where $N_{\rm bin}$ is the number of bins in the observable. For B2017, $N_{\rm mock} = 2048$ and $N_{\rm bin} = 37$. Meanwhile, for S2017, $N_{\rm mock} = 20,000$ and $N_{\rm bin} = 8$.

For the methods we later describe in Section **ref**, the mock observable data ($\mathbf{D}^{\rm mock}$) need to be pre-processed. This pre-processing involves two steps: mean-subtraction and whitening. For mean subtraction, the mean of the observable, $\bar{\mathbf{D}}^{\rm mock} = \overline{P_\ell}$ or $\bar{\zeta}$, is subtracted from $\mathbf{D}^{\rm mock}$. Then $\mathbf{D}^{\rm mock} - \bar{\mathbf{D}}^{\rm mock}$ is whitened using a linear transformation to remove the Gaussian correlation between the bins of $\mathbf{D}^{\rm mock}$:

$$\mathbf{X}^{\rm mock} = L\,(\mathbf{D}^{\rm mock} - \bar{\mathbf{D}}^{\rm mock}). \tag{5}$$

The linear transformation is derived so that covariance matrix of the whitened data, $\mathbf{X}^{\rm mock}$, is the identity matrix $\mathbb{I}$. Such a whitening linear transformation can be derived in infinite ways. One way to derive the linear transformation is through the eigen-decomposition of the covariance matrix (*e.g.* Hartlap et al. 2009; Sellentin et al. 2017). We, alternatively, derive the linear transformation $\mathbf{L}$ using Choletsky decompositon of the inverse covariance matrix (Press et al. 1992): $\mathbb{C}^{-1} = \mathbf{L}\,\mathbf{L}^T$. **We confirm tha the different whitening algorithms, does not impact the results of the paper.** Now that we have the pre-processed data of the mock observables, we proceed in the next section to quantifying the non-Gaussianity of the P_ℓ and ζ likelihoods using $\mathbf{X}^{\rm mock}$.

4. Quantifying the Likelihood non-Gaussianity

The standard approach to parameter inference in LSS studies neglects to account for likelihood non-Gaussianity (Section 2). However, we are not the first to investigate likelihood non-Gaussianity in LSS analyses. Nearly two decades ago, Scoccimarro (2000) examined the likelihood non-Gaussianity for the powerspectrum and reduced bispectrum using mock catalogs of the IRAS redshift catalogs. More recently, Hartlap et al. (2009) and Sellentin et al. (2017) examined the non-Gaussianity of the cosmic shear correlation function likelihood using simulations of the Chandra Deep Field South and CFHTLenS, respectively.

While these works present different methods for identifying likelihood non-Gaussianity, they do not present a concrete way of quantifying it. For instance, Hartlap et al. (2009) identifies the non-Gaussianity of the cosmic shear likelihood by looking at the statistical independence/dependence of PCA components of the mock observable datavector. In Sellentin et al. (2017), they use the Mean Integrated Squared Error (MISE) as a distance metric between Gaussian random variables and the whitened mock observable datavector to identify

non-Gaussian correlations between two elements of the data vector. These indirect measures of likelihood non-Gaussianity are challenging to interpret and extend more generally to LSS studies.

A more direct approach, however, can be taken to quantify the non-Gaussianity of the likelihood. We can calculate the divergence between the distribution of our observable, $p(x)$, and $q(x)$ a multivariate Gaussian described by the average of the mocks and the covariance matrix. The following are two of the most commonly used divergences: the Kullback-Leibler (hereafter KL) divergence

$$D_{KL}(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x \tag{6}$$

and the Rényi-α divergence

$$D_{R-\alpha}(p \parallel q) = \frac{1}{\alpha - 1} \log \int p^{\alpha}(x) q^{1-\alpha}(x) \, \mathrm{d}x. \tag{7}$$

In the limit as α approaches 1, the Rényi-α divergence is equivalent to the KL divergence.

Of course, in our case, we don't know $p(x)$ — *i.e.* the distribution of our observable. If we did, we would simply use that instead of bothering with the covariance matrix or this paper. We can, however, still estimate the divergence using nonparametric divergence estimators (Wang et al. 2009; Póczos et al. 2012; Krishnamurthy et al. 2014). These estimators, allows us to estimate the divergence directly from samples $X_{1:n} = \{X_1, ...X_n\}$ and $Y_{1:m} = \{Y_1, ...Y_m\}$ drawn from p and q respectively: $\hat{D}_{\alpha}(X_{1:n} \parallel Y_{1:m})$.

For instance, the estimator presented in Póczos et al. (2012) allows us to estimate the kernel function of the Rényi-α divergence,

$$D_{\alpha}(p \parallel q) = \int p^{\alpha}(x) q^{1-\alpha}(x) \, \mathrm{d}x. \tag{8}$$

using kth nearest neighbor density estimators. Let $\rho_k(x)$ denote the Euclidean distance of the kth nearest neighbor of x in the sample $X_{1:n}$ and $\nu_k(x)$ denote the Euclidean distance of the kth nearest neighbor of x in the sample $Y_{1:m}$. Then $D_{\alpha}(p \parallel q)$ can be estimated as

$$\hat{D}_{\alpha}(X_{1:n} \parallel Y_{1:m}) = \frac{B_{k,\alpha}}{n} \left(\frac{n-1}{m} \right)^{1-\alpha} \sum_{i=1}^{n} \left(\frac{\rho_k^d(X_i)}{\nu_k^d(X_i)} \right)^{1-\alpha}, \tag{9}$$

where $B_{k,\alpha} = \frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}$. Póczos et al. (2012) goes to further prove that this estimated kernel function is asymptotically unbias,

$$\lim_{n,m \to \infty} \mathbb{E}[\hat{D}_{\alpha}(X_{1:n} \parallel Y_{1:m})] = D_{\alpha}(p \parallel q). \tag{10}$$

Plugging $\hat{D}_\alpha(X_{1:n} \parallel Y_{1:m})$ into Eq. 7, we get an estimator for the Rényi-α divergence. A similar estimator (Wang et al. 2009) can also be derived for the KL divergence (Eq. 6). We note that while the divergence estimators converge to the true divergence with a large enough samples, with a limited number of samples from the distribution, the estimators are noisy.

These divergence estimates have been applied to Support Vector Machines and used extensively in the machine learning and astronomical literature with great success **elaborate a lot more**

- Compile papers that use this divergence, Ntampaka et al. (2015, 2016)

For more details on the non-parametric divergence estimators, we refer readers to Póczos et al. (2012) and Krishnamurthy et al. (2014).

Now we can use the divergence estimators above to quantify the non-Gaussianity of the likelihood. In other words, we're intersted in the divergence between the distribution $p(x)$ sampled by the mock observables $\mathbf{X}^{\rm mock}$ (Section 3.3) and the multivariate Gaussian "pseudo-likelihood" distribution assumed in standard analyses — $\mathcal{N}(\overline{\mathbf{X}}^{\rm mock}, \mathbb{C})$. Since $\mathbf{X}^{\rm mock}$ is a sample from $p(x)$, we draw a reference sample $\mathbf{Y}^{\rm ref}$ from $\mathcal{N}(\overline{\mathbf{X}}^{\rm mock}, \mathbb{C})$ to use in the estimators. Similar to the experiments detailed in Póczos et al. (2012), we construct $\mathbf{Y}^{\rm ref}$ with a comparable sample size as $\mathbf{X}^{\rm mock}$: 2000 and 10,000 for the P_ℓ and ζ analyses respectively. In Figure 3, we compare the distribution of Rényi-α(left) and KL (right) divergence estimates (orange) $\hat{D}_{R\alpha}$ and $\hat{D}_{KL}$ between the mock data $\mathbf{X}^{\rm mock}$ and a reference sample $\mathbf{Y}^{\rm ref}$ for the $P_\ell(k)$ (top) and $\zeta(N)$ (bottom) analyses. These distributions were constructed using **100** divergence estimates, where $\mathbf{Y}^{\rm ref}$ is resampled for each estimate. In order to illustrate the scatter of the estimators and also as a reference point for the comparison, we include (blue) the distribution of $\hat{D}_{R\alpha}(\mathbf{X}^{\rm ref} \parallel \mathbf{Y}^{\rm ref})$ and $\hat{D}_{KL}(\mathbf{X}^{\rm ref} \parallel \mathbf{Y}^{\rm ref})$ where $\mathbf{X}^{\rm ref}$ is a data vector with the same dimension as $\mathbf{X}^{\rm mock}$ sampled from $\mathcal{N}(\overline{\mathbf{X}}^{\rm mock}, \mathbb{C})$.

The discrepancy between the blue and orange distributions illustrates the non-Gaussianity of $p(x)$ sampled by $\mathbf{X}^{\rm mock}$.

- Describe the discrepancy.

5. Estimating the Non-Gaussian Likelihood

In the previous section, we estimate the divergence between the P_ℓ and ζ likelihoods sampled by mocks and their respective Gaussian pseudo-likelihoods. The divergence estimates quantify the non-Gaussianity of the likelihoods and discredit the Gaussian likelihood assumption in standard LSS studies. Our ultimate goal, however, goes beyond demonstrating this non-Gaussianity. We're interested in quantifying the impact of likelihood non-Gaussianity on

the final cosmological parameter constraints and also in developing more accurate methods for parameter inference in LSS analyses.

While the divergence estimates of the previous are useful for quantifying non-Gaussianity, it's not clear how they propagate onto the inferred parameter constraints. In this section, we present two methods for estimating the non-Gaussian likelihood distribution of P_ℓ and ζ from the corresponding mocks. These methods will be used later to quantify the impact of likelihood non-Gaussianity on the parameter constraints of B2017 and S2017. Moreover, they provide a general framework for more accurately estimating the likelihood distribution than the Gaussian pseudo-likelihood.

5.1. Gaussian Mixture Likelihood Estimation

When mock catalogs are used in LSS analyses to estimate the covariance matrix, they are used as data points sampling the likelihood distribution. For the pseudo-likelihood, this distribution is assumed to have a Gaussian functional form. However, the Gaussian functional form, or any functional form for that matter, is not necessary to estimate the likelihood distribution. Instead, the $N_{\rm bin}$-dimensional likelihood distribution can be directly estimated from the set of mock catalogs using — for instance, Gaussian mixture density estimation (McLachlan & Peel 2000). **talk about how this is used extensively in ML** In astronomy, Gaussian mixture density estimation has been used for inferring the velocity distribution of stars from the Hipparcos satellite (Bovy et al. 2011), classifying galaxies in the Galaxy And Mass Assembly Survey Taylor et al. (2015), and classifying pulsars (Lee et al. (2012); see also references in Kuhn & Feigelson (2017)).

Gaussian mixutre density estimation is a "semi-parametric" method that uses a weighted sum of k Gaussian component densities (a Gaussian mixture model)

$$p(\mathbf{x};\boldsymbol{\theta}) = \sum_{i=1}^{k} \pi_i \mathcal{N}(\mathbf{x};\boldsymbol{\theta}_i), \tag{11}$$

to estimate the density. The component weights (π_i; also known as mixing weights) and the component parameters $\boldsymbol{\theta}_i$ are free parameters of the mixture model. Given some data set $\mathbf{X}_N = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$, the free parameters of the gaussian mixture model are, most popularly, estimated through an expectation-maximization algorithm (EM; Dempster et al. 1977; Neal & Hinton 1998). The EM algorithm begins by randomly assigning $\boldsymbol{\theta}_i^0$ to the k Gaussian componenets. The algorithm then iterates between two steps. In the first step, the algorithm computes, for each data point $\mathbf{x}_n$, a probability of being generated by each component of the model. These probabilities can be thought of as weighted assignments of the points to the components. Next, given the assignment of $\mathbf{x}_n$ to the components, $\boldsymbol{\theta}_i^t$ of each component are updated to $\boldsymbol{\theta}_i^{t+1}$ to maximize the likelihood of the assigned points. At this point, the mixing

weights π_i can also be updated by summing up the assignment weights and normalizing it by N, the total number of data points. This entire process is repeated until convergence — *i.e.* when the log-likelihood of given the mixture model $\log \mathcal{L} = \log p(\mathbf{X}_N; \boldsymbol{\theta}^t)$ converges. The EM algorithm is guaranteed to converges to a local maximum of the likelihood (Wu 1983).

In practice, instead of arbitrarily assigning the initial $\boldsymbol{\theta}_i^0$ more commonly $\boldsymbol{\theta}_i^0$ is derived from a $\texttt{k}-\texttt{means}$ clustering algorithm (Lloyd 1982). Without going into details, the $\texttt{k}-\texttt{means}$ algorithm clusters the data $\mathbf{X}_N$ into k clusters, each described by the mean (or centroid) μ_i of the samples in the cluster. The algorithm then iteratively chooses centroids that minimize the average squared distance between points in the same cluster. In our use of Gaussian mixture modelling, we set the initial conditions of the EM algorithm using the $\texttt{k}-\texttt{means}{+}{+}$ algorithm of Arthur & Vassilvitskii (2007). In Figure 2, we illustrate Gaussian Mixture density estimation in action. We use Gaussian mixture models with $k = 1$ (top), 3 (middle), 10 (bottom) components to estimate the distribution of one dimensional data drawn from three separate Gaussia distributions. **As an illustration of Gaussian mixture density estimation, in Figure we show Gaussian mixture models of the highest N bin of $\mathbf{X}^{\rm mock}$.**

So far in our discussion of Gaussian mixture modeling, we have kept the number of componenets k fixed. However, k is a free parameter and selecting it is a crucial step in Gaussian mixture density estimation. With too many components the model may overfit the data; with too few components, the model may not be flexible enough to approximate the true underlying distribution. In order to address this model selection problem of selecting k, we make use of the Bayesian Information Criterion (BIC; Schwarz 1978). BIC has been widely used for determining the number of components in mixture modeling (Leroux 1992; Roeder & Wasserman 1997; Fraley & Raftery 1998; Steele & Raftery 2010) and for model selection in general in astronomy (*e.g.* Liddle 2007; Broderick et al. 2011; Wilkinson et al. 2015; Vakili & Hahn 2016). According to BIC, models with higher likelihood are preferred; however, to address the concern of overfitting, BIC introduces a penalty term for the number of parameters in the model:

$$\mathrm{BIC} = -2 \ln \mathcal{L} + N_{\rm par} \ln N_{\rm data}. \tag{12}$$

We select k based on the number of components in the model with the lowest BIC.

With the Gaussian mixture density estimation method described above we can directly estimate the likelihood distribution using the mock catalogs. We first fit Gaussian mixture models with $k < 30$ components to the whitened mock data $\mathbf{X}^{\rm mock}$ using the EM algorithm for each model. For each of the coverged Gaussian mixture models, we then calculate the BIC. Afterwards we select the model with the lowest BIC as the best density estimate of the likelihood distribution: $p_{\rm GMM}(\mathbf{X}^{\rm mock})$. The selected density estimate can then be used to calculate the likelihood and quantify the impact of likelihood non-Gaussianity on the parameter constraints of B2017 and S2017. But before we do that, we have to confirm

whether $p_{\rm GMM}(\mathbf{X}^{\rm mock})$ is in fact an improved estimate of the likelihood over the Gaussian pseudo-likelihood. To do this, we return to the divergence estimates of Section 4.

To estimate the divergence between our Gaussian mixture density estimate, $p_{\rm GMM}(\mathbf{X}^{\rm mock})$, and the likelihood distribution, we take the same approach as our $D(\mathbf{X}^{\rm mock} \parallel \mathbf{Y}^{\rm ref})$ calculation. We draw samples from $p_{\rm GMM}(\mathbf{X}^{\rm mock})$ with the same dimensions as $\mathbf{Y}^{\rm ref}$. Then we calculate k-NN Rényi-α and KL divergence estimates (Section 4) between this sample and $\mathbf{X}^{\rm mock}$. As we did in Figure 1, we repeat this process **100** times, resampling $p_{\rm GMM}(\mathbf{X}^{\rm mock})$ each time, in order get a distribution of divergence estimates that reflects the scatter in the estimator. In Figure 3, we present the resulting distribution of divergences between $p_{\rm GMM}(\mathbf{X}^{\rm mock})$ and the likelihood distribution in **green** for the $P_\ell(k)$ (top) and $\zeta(N)$ (bottom) analyses. For comparison, we also include the distributions from Figure 1.

For the $\zeta(N)$ analysis of S2017, our Gaussian mixture density estimate significantly improves the divergence discrepancy compared to the pseudo-likelihood. In other words, *our Gaussian mixture density estimate is a significant better estimate of the ζ likelihood distribution than the pseudo-likelihood*. On the other hand, our Gaussian mixture density estimate for the $P_\ell(k)$ analysis of B2017 does not significantly improve the divergence discrepancy. The difference in the performance of Gaussian mixture density estimation is not a surprise. One would expect a direct density estimation to be more effective for the S2017 case, where we are estimating a $N_{\rm bin} = 8$ dimensional distribution with $N_{\rm mock} = 20,000$ samples, compared to the B2017 case. For B2017 we are estimating a higher dimensional distribution ($N_{\rm bin} = 37$) with fewer samples ($N_{\rm mock} = 2048$). Given the unconvincing accuracy of the Gaussian mixture density estimate of the P_ℓ likelihood, in the next section, we present an alterative method for estimating the non-Gaussian likelihood.

5.2. Independent Component Analysis

As Figure 3 reveals, Gaussian mixture density estimation fails to accurately estimate the 37-dimensional P_ℓ likelihood distribution of the B2017 analysis. Rather than estimating the high dimensional likelihood distribution directly, following the approach from Hartlap et al. (2009), consider a linear transformation on the observable (P_ℓ) into $N_{\rm bin}$ statistically independent components. **equation** Since each component is statistically independent from the rest, the likelihood becomes the multiple of $N_{\rm bin}$ one dimensional distributions. **equation** For the B2017 case, this reduces the problem of estimating a 37 dimensional distribution with 2048 samples to a problem of estimating 37 one dimensional distributions with 2048 samples each. The challenge, however, is in finding the transformation.

Efforts in the past have similarly attempted to tackle the high-dimensionality problem of the observable space (*e.g.* Scoccimarro 2000; Eisenstein & Zaldarriaga 2001; Gaztañaga & Scoccimarro 2005; Norberg et al. 2009; Sinha et al. 2017). These works typically use singular value decomposition or principal component analysis (hereafter PCA; Press et al. 1992). For

a Gaussian likelihood, the resulting components from such a decomposition/transformation are statistically independent. However, when the likelihood is not Gaussian, as in our case (Section 4), the PCA components are uncorrelated but *not necessarily statistically independent* (Hartlap et al. 2009). Therefore, instead of PCA, we use Independent Component Analysis (hereafter ICA).

- describe ICA
- Some pedagogical figure that shows how ICA works.

describe ICA here

Using ICA, we decompose/transform $\mathbf{X}^{\rm mock}$ with the unmixing matrix $\mathbf{W}$ into $N_{\rm bin}$ independent components:

$$\mathbf{X}^{\rm ICA} = \mathbf{W}\,\mathbf{X}^{\rm mock} = \{\mathbf{X}^{\rm ICA}_1, ..., \mathbf{X}^{\rm ICA}_{N_{\rm bin}}\}. \tag{13}$$

The statistical indpendence of these components, allow us to write down the likelihood distribution as

$$\mathcal{L} \approx \prod_{n=1}^{N_{\rm bin}} p_{x^{\rm ICA}_n}(x) \tag{14}$$

where $p_{x^{\rm ICA}_n}(x)$ is the 1-dimensional distribution function of the $n^{\rm th}$ ICA component, which in our context is sampled by $\mathbf{X}^{\rm ICA}_n$. Based on Eq. 14, we can estimate the high-dimensional $\mathcal{L}$ from $\mathbf{X}^{\rm mock}$ by estimating the $p_{x^{\rm ICA}_n}$ distributions from $\mathbf{X}^{\rm ICA}_n$. For estimating the $p_{x^{\rm ICA}_n}$s, we use the kerenl density method (hereafter KDE; *e.g.* Hastie et al. 2009; Feigelson & Babu 2012) following Hartlap et al. (2009). With KDE, the density estimate, $\hat{p}_{x^{\rm ICA}_n}$, is constructed by smoothing the empirical distribution of the ICA componenet $x^{\rm ICA}_n$ using a smooth kernel. In our context, the density estimate is

$$\hat{p}_{x^{\rm ICA}_n}(x) = \frac{1}{N_{\rm mock} b} \sum_{j=1}^{N_{\rm mock}} K\left(\frac{x - \mathrm{X}^{(j),{\rm ICA}}_n}{b}\right). \tag{15}$$

b is the bandwidth and K is the kernel function. Following the choices of Hartlap et al. (2009), we use a Gaussian distribution for K and the "rule of thumb" bandwidth (also known as Scott's rule) for b.

- Kernel Density Estimation
- put it all together to estimate the likelihood
- Figure 3 showing the results.

6. Impact on Parameter Inference

With accurate estimates of the likelihood distribution now in hand, we can now quantify the impact of likelihood non-Gaussianity on the parameter constraints of B2017 and S2017. We begin by providing a brief overview of the MCMC frameworked used in the original B2017 and S2017. analyses.

- details of each of the MCMC runs
- equations explaining importance sampling framework

Figure 4

Figure 5

7. Discussion

- Will it matter for future surveys?
- Likelihood free inference (cite justin's paper)

8. Summary

Acknowledgements

It's a pleasure to thank Simone Ferraro, David W. Hogg, Emmaneul Schaan, Roman Scoccimarro Zachary Slepian

REFERENCES

Alam, S., Ata, M., Bailey, S., et al. 2017, Monthly Notices of the Royal Astronomical Society, 470, 2617

Arthur, D., & Vassilvitskii, S. 2007, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07 (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics), 1027

Berlind, A. A., Frieman, J., Weinberg, D. H., et al. 2006, The Astrophysical Journal Supplement Series, 167, 1

Beutler, F., Seo, H.-J., Saito, S., et al. 2017, Monthly Notices of the Royal Astronomical Society, 466, 2242

Bianchi, D., Gil-Marín, H., Ruggeri, R., & Percival, W. J. 2015, Monthly Notices of the Royal Astronomical Society, 453, L11

Bovy, J., Hogg, D. W., & Roweis, S. T. 2011, The Annals of Applied Statistics, 5, 1657

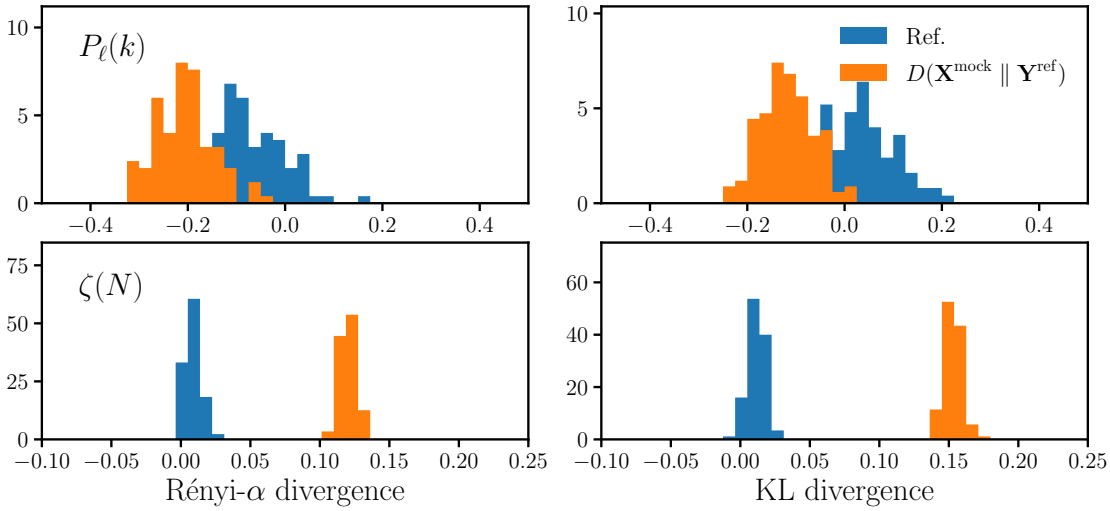

Fig. 1.— Rényi-α and KL divergence estimates, ($D_{R\alpha}$ and D_{KL}), between the mock data $\mathbf{X}^{\rm mock}$ and a reference sample $\mathbf{Y}^{\rm ref}$ for the $P_\ell(k)$ (left) and $\zeta(N)$ (right) analyses.

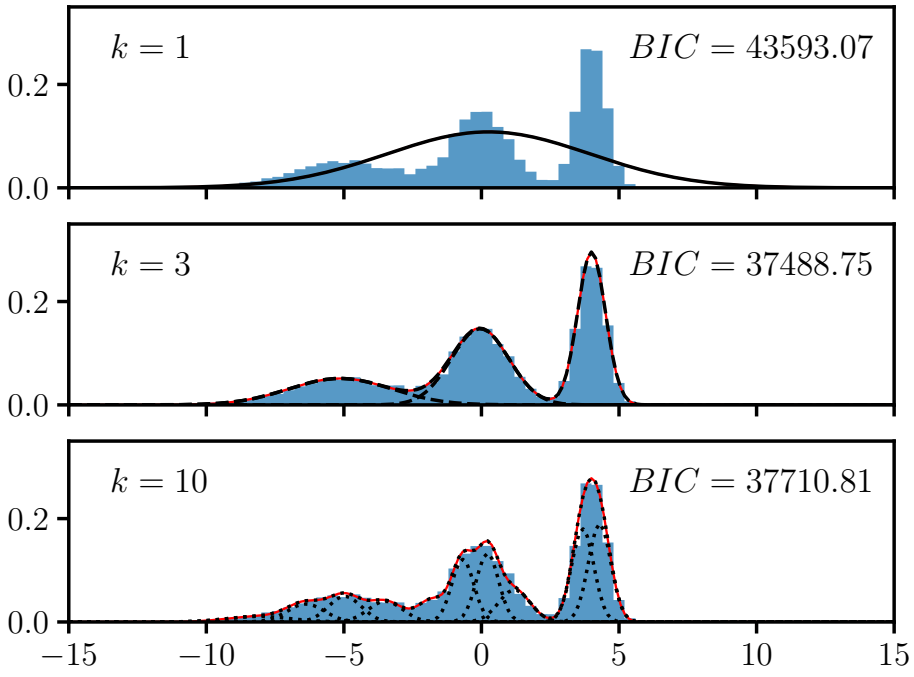

Fig. 2.— We use Gaussian mixture models with $k = 1$ (top), 3, (middle), 10 (bottom) components to estimate the distribution of data (blue) drawn from three Gaussian distributions.

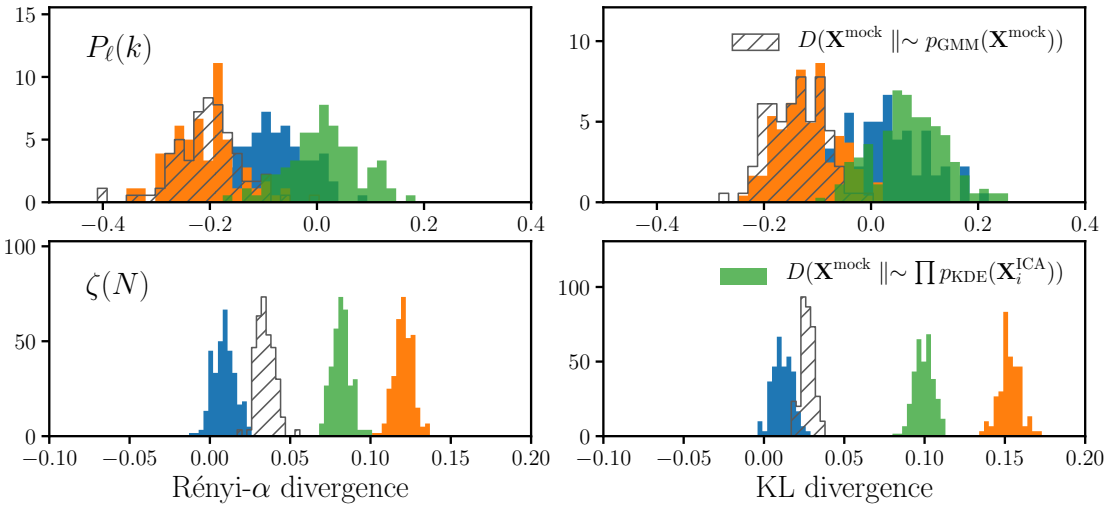
$P_\ell(k)$
$\zeta(N)$
$D(\mathbf{X}^{\rm mock} \| \sim p_{\rm GMM}(\mathbf{X}^{\rm mock}))$
$D(\mathbf{X}^{\rm mock} \| \sim \prod p_{\rm KDE}(\mathbf{X}_i^{\rm ICA}))$
Rényi-α divergence
KL divergence

Fig. 3.—

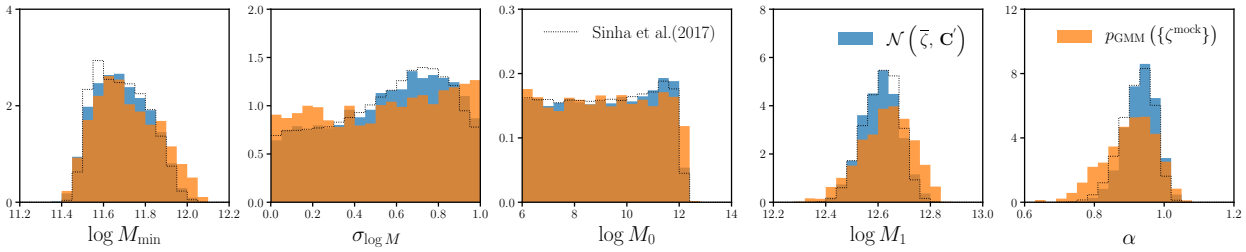
Sinha et al.(2017)
$\mathcal{N}\left(\bar{\zeta}, \mathbf{C}'\right)$
$p_{\rm GMM}\left(\{\zeta^{\rm mock}\}\right)$
$\log M_{\rm min}$
$\sigma_{\log M}$
$\log M_0$
$\log M_1$
α

Fig. 4.—

Broderick, A. E., Fish, V. L., Doeleman, S. S., & Loeb, A. 2011, The Astrophysical Journal, 735, 110

Crocce, M., Pueblas, S., & Scoccimarro, R. 2006, Monthly Notices of the Royal Astronomical Society, 373, 369

Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, The Astrophysical Journal, 292, 371

Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977, Journal of the Royal Statistical Society. Series B (Methodological), 39, 1

Eisenstein, D. J., & Zaldarriaga, M. 2001, The Astrophysical Journal, 546, 2

Feigelson, E. D., & Babu, G. J. 2012, Modern Statistical Methods for Astronomy

Fraley, C., & Raftery, A. E. 1998, The Computer Journal, 41, 578

Gardner, J. P., Connolly, A., & McBride, C. 2007, in Astronomical Data Analysis Software and Systems XVI, Vol. 376, 69

Gaztañaga, E., & Scoccimarro, R. 2005, Monthly Notices of the Royal Astronomical Society, 361, 824

Grieb, J. N., Sánchez, A. G., Salazar-Albornoz, S., et al. 2017, Monthly Notices of the Royal Astronomical Society, 467, 2085

Guo, H., Zehavi, I., & Zheng, Z. 2012, The Astrophysical Journal, 756, 127

Hahn, C., Scoccimarro, R., Blanton, M. R., Tinker, J. L., & Rodríguez-Torres, S. A. 2017, Monthly Notices of the Royal Astronomical Society, 467, 1940

Hand, N., Li, Y., Slepian, Z., & Seljak, U. 2017, Journal of Cosmology and Astro-Particle Physics, 07, 002

Hartlap, J., Schrabback, T., Simon, P., & Schneider, P. 2009, Astronomy and Astrophysics, 504, 689

Hastie, T., Tibshirani, R., & Friedman, J. 2009, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) (Springer)

Kazin, E. A., Koda, J., Blake, C., et al. 2014, Monthly Notices of the Royal Astronomical Society, 441, 3524

Kitaura, F.-S., Gil-Marín, H., Scóccola, C. G., et al. 2015, Monthly Notices of the Royal Astronomical Society, 450, 1836

Kitaura, F.-S., & Heß, S. 2013, Monthly Notices of the Royal Astronomical Society, 435, L78

Kitaura, F.-S., Yepes, G., & Prada, F. 2014, Monthly Notices of the Royal Astronomical Society, 439, L21

Kitaura, F.-S., Rodríguez-Torres, S., Chuang, C.-H., et al. 2016, Monthly Notices of the Royal Astronomical Society, 456, 4156

Klypin, A., Yepes, G., Gottlöber, S., Prada, F., & Heß, S. 2016, Monthly Notices of the

Royal Astronomical Society, 457, 4340
Krishnamurthy, A., Kandasamy, K., Poczos, B., & Wasserman, L. 2014, arXiv:1402.2966 [math, stat], arXiv:1402.2966 [math, stat]
Kuhn, M. A., & Feigelson, E. D. 2017, arXiv:1711.11101 [astro-ph, stat], arXiv:1711.11101 [astro-ph, stat]
Lee, K. J., Guillemot, L., Yue, Y. L., Kramer, M., & Champion, D. J. 2012, Monthly Notices of the Royal Astronomical Society, 424, 2832
Leroux, B. G. 1992, The Annals of Statistics, 20, 1350
Liddle, A. R. 2007, Monthly Notices of the Royal Astronomical Society, 377, L74
Lloyd, S. 1982, IEEE Transactions on Information Theory, 28, 129
McBride, C., Berlind, A., Scoccimarro, R., et al. 2009, in Bulletin of the American Astronomical Society, Vol. 213, 425.06
McLachlan, G., & Peel, D. 2000, Finite Mixture Models (Wiley-Interscience)
Mohammed, I., Seljak, U., & Vlah, Z. 2017, Monthly Notices of the Royal Astronomical Society, 466, 780
Neal, R. M., & Hinton, G. E. 1998, in Learning in Graphical Models, NATO ASI Series (Springer, Dordrecht), 355
Norberg, P., Baugh, C. M., Gaztañaga, E., & Croton, D. J. 2009, Monthly Notices of the Royal Astronomical Society, 396, 19
Ntampaka, M., Trac, H., Sutherland, D. J., et al. 2015, The Astrophysical Journal, 803, 50
—. 2016, The Astrophysical Journal, 831, 135
Parkinson, D., Riemer-Sørensen, S., Blake, C., et al. 2012, Physical Review D, 86, 103518
Pinol, L., Cahn, R. N., Hand, N., Seljak, U., & White, M. 2017, Journal of Cosmology and Astroparticle Physics, 2017, 008
Póczos, B., Xiong, L., Sutherland, D. J., & Schneider, J. 2012, in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2989
Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing (New York, NY, USA: Cambridge University Press)
Rodríguez-Torres, S. A., Chuang, C.-H., Prada, F., et al. 2016, Monthly Notices of the Royal Astronomical Society, 460, 1173
Roeder, K., & Wasserman, L. 1997, Journal of the American Statistical Association, 92, 894
Ross, A. J., Beutler, F., Chuang, C.-H., et al. 2017, Monthly Notices of the Royal Astronomical Society, 464, 1168
Schwarz, G. 1978, The Annals of Statistics, 6, 461
Scoccimarro, R. 1998, Monthly Notices of the Royal Astronomical Society, 299, 1097
—. 2000, The Astrophysical Journal, 544, 597
Sellentin, E., Jaffe, A. H., & Heavens, A. F. 2017, arXiv:1709.03452 [astro-ph, stat],

arXiv:1709.03452 [astro-ph, stat]
Sinha, M., Berlind, A. A., McBride, C. K., et al. 2017, arXiv:1708.04892 [astro-ph], arXiv:1708.04892 [astro-ph]
Slepian, Z., Eisenstein, D. J., Brownstein, J. R., et al. 2017, Monthly Notices of the Royal Astronomical Society, 469, 1738
Spergel, D. N., Bean, R., Doré, O., et al. 2007, The Astrophysical Journal Supplement Series, 170, 377
Springel, V. 2005, Monthly Notices of the Royal Astronomical Society, 364, 1105
Steele, R. J., & Raftery, A. E. 2010
Taylor, E. N., Hopkins, A. M., Baldry, I. K., et al. 2015, Monthly Notices of the Royal Astronomical Society, 446, 2144
Tinker, J. L., & et al. in preparation
Vakili, M., & Hahn, C. H. 2016, arXiv:1610.01991 [astro-ph], arXiv:1610.01991 [astro-ph]
Vargas-Magaña, M., Ho, S., Xu, X., et al. 2014, Monthly Notices of the Royal Astronomical Society, 445, 2
Wang, Q., Sanjeev, K., & Sergio, V. 2009, IEEE TRANSACTIONS ON INFORMATION THEORY, 55, 2392
Warren, M. S., Abazajian, K., Holz, D. E., & Teodoro, L. 2006, The Astrophysical Journal, 646, 881
Wilkinson, D. M., Maraston, C., Thomas, D., et al. 2015, Monthly Notices of the Royal Astronomical Society, 449, 328
Wu, C. F. J. 1983, The Annals of Statistics, 11, 95
Zhao, C., Kitaura, F.-S., Chuang, C.-H., et al. 2015, Monthly Notices of the Royal Astronomical Society, 451, 4266
Zheng, Z., Coil, A. L., & Zehavi, I. 2007, The Astrophysical Journal, 667, 760

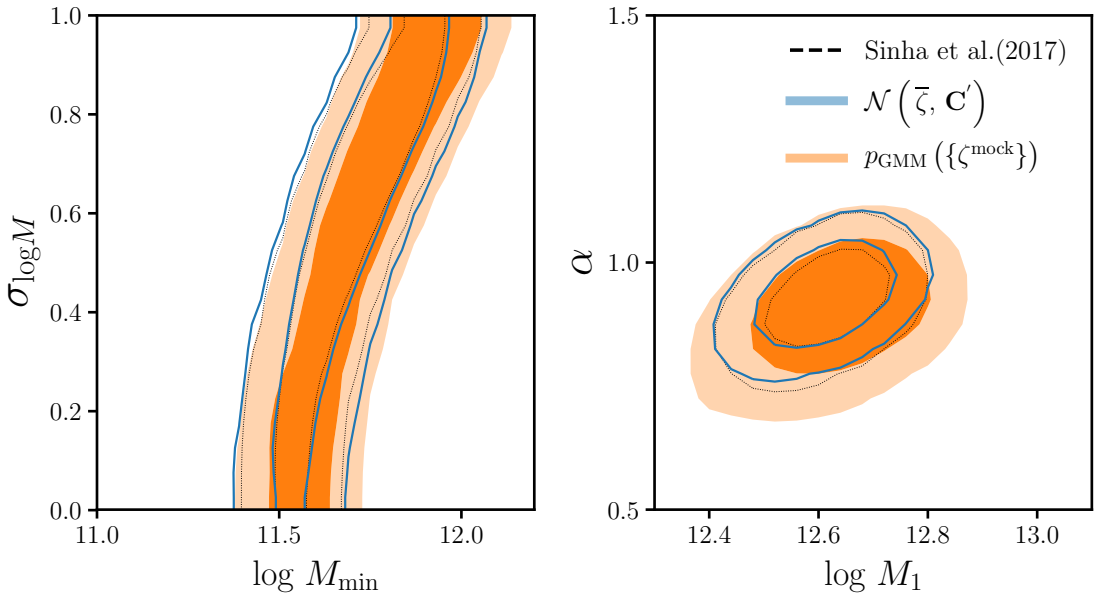
1.0
0.8
0.6
0.4
0.2
0.0
$\sigma_{\log M}$
11.0
11.5
12.0
log $M_{\rm min}$
1.5
1.0
0.5
α
12.4
12.6
12.8
13.0
log M_1
Sinha et al.(2017)
$\mathcal{N}\left(\overline{\zeta}, \mathbf{C}'\right)$
$p_{\rm GMM}\left(\{\zeta^{\rm mock}\}\right)$

Fig. 5.—