The UniverseMachine

Science/Documentation Paper: https://arxiv.org/abs/1806.07893

Contents

1 Overview **2**

2 Precalculated Data Products **2**
2.1 Observed versus True Stellar Masses and Star Formation Rates 3
2.2 Meaning of Values and Uncertainties 3
2.3 Correlation Functions 3
2.4 Cosmic Star Formation Rates 4
2.5 Ex-Situ Fractions 4
2.6 Halo Mass Functions 4
2.7 Infall and Quenching Distribution Statistics for Satellites . . . 4
2.8 Observations and Best-fit Model 5
2.9 Stellar Mass–Halo Mass Relations 7
2.9.1 Median Measurements from the Simulation 7
2.9.2 Median Fits 7
2.9.3 Average Measurements from the Simulation 8
2.10 Quenched Fractions 8
2.11 Rejuvenation Statistics 8
2.12 Average Star Formation Histories 9
2.13 Stellar Mass Functions and Satellite Fractions 9
2.14 Average Specific Star Formation Rates 9
2.15 UV Luminosity Functions 9
2.16 UV–Stellar Mass Relations 9
2.17 Weak Lensing 10

3 Mock Catalogs and Lightcones **10**
3.1 Halo and Galaxy Properties 10
3.2 Star Formation Histories 11
3.3 CANDELS Lightcones 11

4 Running Basic Analyses **11**
4.1 Compiling 11
4.2 Making New Lightcones 12
4.3 Making New Catalogs 12

5 Running on Different Simulations **13**

6 Parameter Space Exploration **14**
6.1 Observational Constraints 14
6.2 Setting Up the Config File 15
6.2.1 Simulation/Data Setup 15
6.2.2 Box Size/Cosmology 16
6.2.3 Parallel Node Setup/MCMC 16
6.3 Finding a Best-Fit Solution 17
6.4 MCMC Analysis 18

1 Overview

The UniverseMachine applies simple empirical models of galaxy formation to dark matter halo merger trees. For each model, it generates an entire mock universe, which it then observes in the same way as the real Universe to calculate a likelihood function. It includes an advanced MCMC algorithm to explore the allowed parameter space of empirical models that are consistent with observations.

2 Precalculated Data Products

The latest data release is available at `http://www.peterbehroozi.com/data.html`. The UniverseMachine can generate many data products (e.g., stellar mass functions, stellar mass—halo mass relations, star formation histories, etc.) and the list is growing continuously. This section describes some of the available data products, which are all in the `data` subdirectory of the data release tarball. Except where otherwise specified, errors represent the uncertainties in the model posterior distribution.

2.1 Observed versus True Stellar Masses and Star Formation Rates

The UniverseMachine keeps track of two stellar masses – the "true" stellar mass, and the "observed" stellar mass. The true stellar mass is the physically self-consistent stellar mass given by the integral of past star formation minus stellar mass loss. The observed stellar mass includes systematic offsets as well as scatter that both change as a function of redshift. Which one is more useful depends on the purpose—those wanting physically self-consistent star formation histories should use the true stellar mass; those wanting to compare with other observations should use the observed stellar mass. A similar distinction applies to star formation rates, where the observed rates include additional scatter and systematic offsets, but are the closest match to other observations; the true SFRs are by contrast guaranteed to be physically self-consistent with the true stellar masses.

2.2 Meaning of Values and Uncertainties

Almost all data products include the bestfit value as well as the 68% confidence interval from the model posterior space. The column `Err+` gives the difference between the 84th-percentile model and the bestfit model; the column `Err-` gives the difference between the bestfit model and the 16th-percentile model. Hence, if the best-fit value, `Err+`, and `Err-` columns are A, B, and C, respectively, the 68% confidence interval ranges from A-C to A+B. Exceptions to this rule are always noted in the data files.

As a general rule, *Bolshoi-Planck* becomes increasingly incomplete for satellite galaxies with peak halo masses $M_{\mathrm{peak}} < 10^{10.5}\mathrm{M}_{\odot}$ and for central galaxies with peak halo masses $M_{\mathrm{peak}} < 10^{10}\mathrm{M}_{\odot}$. The incompleteness for massive halos varies with redshift; most data files include galaxy/halo counts so that it is clear where the statistics become unreliable.

2.3 Correlation Functions

Found in `data/corrs`. Correlation functions are in `corr_sm*`; the filename gives the observed stellar mass range and the scale factor at which the correlation function was calculated. Correlation functions for all, star-forming, and quenched galaxies are included, as is the star-forming x quenched cross-correlation function. Values for, e.g., π_{max}, redshift errors, etc., depend on the parameter file used, but are documented in each file. Ratios of correlation functions are in `corr_ratios_sm*`. The meaning of the filename is the same (stellar mass range and the scale factor at which the correlation function was calculated). Ratios of quenched to star-forming, star-forming to all, quenched to all, and the cross-correlation to all galaxies are included.

2.4 Cosmic Star Formation Rates

Found in `data/csfrs`. This includes the total observed CSFR, the observed CSFR for galaxies with $M_{1500} < -17$ (AB), and the true CSFR. For *Bolshoi-Planck*, the $M_{1500} < -17$ CSFR is almost identical to the total CSFR, as the simulation becomes increasingly incomplete for $M_{1500} > -19$. This also suggests that the "total" CSFRs at $z > 8$ are underestimates of the true total CSFRs.

2.5 Ex-Situ Fractions

Found in `data/ex_situ`. Ex-situ fractions (i.e., fractions of mass accreted in mergers) as a function of observed stellar mass are in `ex_situ_a*`, where the filename includes the scale factor. Ex-situ fractions as a function of M_{peak} are in `ex_situ_hm_a*`.

2.6 Halo Mass Functions

Found in `data/hmfs`. Both the total mass function (including satellites) as well as the satellite fractions are in `hmf_a*`, where the filename includes the scale factor. While central halos are invariant for a given simulation, the satellite fraction can vary as a result of the orphan threshold (see the UniverseMachine paper).

2.7 Infall and Quenching Distribution Statistics for Satellites

Found in `data/infall_stats`. The time distributions since satellite first infall (i.e., the 50^{th}, 84^{th}, and 16^{th} percentiles) as a function of observed satellite stellar mass are in `infall_delay_times_a*`, where the filename includes the scale factor. These percentiles refer to the distribution for individual galaxies; each percentile is accompanied by uncertainties across the model posterior distribution. I.e., the median time since infall is reported for the best-fit model, followed by the 68% confidence interval on the median across model posterior space, followed by the 84th percentile time since infall, followed by the 68% confidence interval on the 84th percentile across model posterior space, and so on. The time distributions are recorded for satellites of Milky Way–mass hosts, group-mass hosts, and cluster-mass hosts; the definitions for each host mass are given in the file.

Quenching delay times (i.e., the time delay between satellite infall and satellite quenching for quenched satellites) as a function of observed satellite stellar mass are in `quenching_delay_times_a*`. As with infall delay times, the 50th, 84th, and 16th percentiles of the distribution for individual satellites are given, for satellites of Milky Way–mass hosts, group-mass hosts, and cluster-mass hosts. Similarly, infall SSFR distributions as a function of observed satellite stellar mass are given in `infall_ssfrs_a*`. Only a restricted set of scale factors are available for the latter to reduce the disk space necessary for postprocessing.

2.8 Observations and Best-fit Model

Found in `data/obs.txt`. The top line contains the best-fit model, which may be used to generate new catalogs with the `make_sf_catalog` command (§4.3). The remaining lines contain one data point per line, including both the observed value and the modeled value. The data point type can be one of the following:

- `smf`: observed stellar mass function (i.e., galaxy number density); units of comoving $\mathrm{Mpc}^{-3}\ \mathrm{dex}^{-1}$.
- `uvlf`: $\mathrm{M}_{1500,UV}$ luminosity function; units of comoving $\mathrm{Mpc}^{-3}\ \mathrm{mag}^{-1}$.
- `qf`: quenched fraction as a function of observed stellar mass, using the Moustakas/PRIMUS quenching threshold.
- `qf11`: quenched fraction as a function of observed stellar mass, using a threshold of $\mathrm{SSFR} < 10^{-11}\ \mathrm{yr}^{-1}$.
- `qf_uvj`: UVJ quenched fraction as a function of observed stellar mass.
- `ssfr`: average observed specific star formation rate (for all galaxies) as a function of observed stellar mass; units of yr^{-1}.

- `csfr`: total observed cosmic star formation rate; units of $M_\odot$ yr^{-1} comoving Mpc^{-3}.
- `csfr_(uv)`: total observed cosmic star formation rate with $M_{1500} < -17$ (AB); same units as `csfr`.
- `correlation`: projected autocorrelation functions; units of comoving Mpc.
- `cross correlation`: projected cross-correlation functions; units of comoving Mpc.
- `conformity`: central galactic conformity, currently unused.
- `lensing`: weak lensing, currently unused.
- `cdens_fsf`: fraction of star-forming central galaxies, as a function of environment density.
- `cdens_ssfr_sf`: observed specific star formation rates for star-forming central galaxies, as a function of environmental density; currently unused.
- `uvsm`: median stellar mass as a function of M_{1500}; units of $M_\odot$.
- `irx`: average infrared excess as a function of M_{1500}; $\log_{10}$ units.

The data point subtype is typically "a" for "all galaxies," but for correlation functions, it can also be "s" for star-forming galaxies and "q" for quenched galaxies. The columns `Z1` and `Z2` contain the redshift range of the observation; `step1` and `step2` are the corresponding range of simulation snapshot numbers used. `SM1` and `SM2` are the stellar mass bin, except for `csfr` (where the stellar mass bin is meaningless), `uvlf` (where the UV magnitude bin is given instead), and `cdens_fsf` (where the bin for the number of neighbors is given instead). `smb1` and `smb2` give the internal stellar mass/UV/environment bin indices used by the UniverseMachine. `R1` and `R2` are the range of radii used, which is only relevant for `correlation` functions.

The observed data point is given by the `Val` column, with +/− uncertainties given in the `Err_h` and `Err_l` columns, respectively. All values and uncertainties are given in $\log_{10}$ units, except for quenched and star-forming fractions (`qf`, `qf_uvj`, `cdens_fsf`), which are in linear units. The best-fit model result is given in the `Model_Val` column, with the +/− 68% confidence interval of the model posteriors in the `MV+` and `MV-` columns. The best-fit χ^2 and 68% range are given in the next three columns. Here, χ^2 values may be zero if the model result is within the calculation error tolerance of the observed value (to prevent over-fitting). Correlation functions and other observations that use covariance matrices may not have a direct relation between χ^2 values and model – observed differences for individual data points.

2.9 Stellar Mass–Halo Mass Relations

2.9.1 Median Measurements from the Simulation

Direct measurements of the median relations binned on halo peak mass are found in `data/smhm/median_raw`. The median SMHM ratios are in `smhm_a*`, and are available for both observed and true stellar masses, as well as subsamples (e.g., centrals, satellites, quenched, star-forming, etc.); the filename includes the scale factor. The halo mass column gives $\log_{10}(M_{\text{peak}}/M_\odot)$, the SMHM ratio columns give the median $\log_{10}(M_*/M_{\text{peak}})$, and the error columns give the +/− uncertainties in dex. Errors on the observed stellar mass ratios should be interpreted as *statistical* errors; errors on the true stellar mass ratios should be interpreted as *statistical+systematic* errors. Ratios of subsamples (e.g., central galaxies vs. all galaxies) are in `ratios_a*`. As the model currently applies the same offset between observed and true stellar masses to all galaxies, the stellar mass ratios are the same for observed and true stellar masses; hence, no distinction is made in the file. Finally, measurements of the scatter in $\log_{10}(M_*/M_{\text{peak}})$ (in dex) are in `smhm_scatter_a*`.

2.9.2 Median Fits

Fits to the measurements above are found in `data/smhm/median_fits` in pretabulated form for both `smhm_a*` and `ratios_a*`; the filename includes the scale factor. Residuals with the direct measurements are found in `smhm_residuals*`; these should be examined if using the fits for a rare population (e.g., high-redshift quiescent galaxies).

The fit parameters listed in the paper are found in `data/smhm/params`, along with a Python script to generate SMHM ratios at arbitrary redshifts.

2.9.3 Average Measurements from the Simulation

Averages of $\log_{10}(M_*/M_{\rm peak})$ (both for observed and true stellar mass) as a function of peak halo mass are found in data/smhm/averages/sm_averages_a*; the filename includes the scale factor. Average halo masses as a function of observed stellar mass are found in data/smhm/averages/hm_averages_a*. Here, several different averages are available, including the linear average peak halo mass ($\langle M_{\rm peak}\rangle$), the log average peak halo mass ($\langle\log_{10}(M_{\rm peak})\rangle$), and the weak lensing-averaged halo mass ($\langle M_{\rm peak}^{2/3}\rangle^{3/2}$).

2.10 Quenched Fractions

Found in data/qfs. Basic quenched fractions according to three different quenching definitions (Moustakas/PRIMUS, SSFR $< 10^{-11}$ yr^{-1}, and UVJ) are found as a function of observed stellar mass in qf_a* and as a function of peak halo mass in qf_hm_a*; the filename includes the scale factor. Statistics for the quenched fraction of all centrals and all satellites, as well as satellites of Milky Way-mass hosts, group-mass hosts, and cluster-mass hosts are found as a function of observed stellar mass in qf_groupstats_a* and of peak (satellite) halo mass in qf_hm_groupstats_a*. The fractions of quenched satellites that were quenched after infall for Milky Way-mass, group-mass, and cluster-mass hosts are found as a function of observed stellar mass in qf_groupstats_infall_a* and of peak (satellite) halo mass in qf_hm_groupstats_infall_a*. The fractions of quenched satellites that were quenched *due to* infall (calculated as $f_{q,sat} - f_{q,cen}$) for all satellites as well as satellites of Milky Way-mass hosts, group-mass hosts, and cluster-mass hosts are found as a function of observed stellar mass in qf_quenched_infall_a* and of peak (satellite) halo mass in qf_hm_quenched_infall_a*. The fractions of galaxies' most-massive progenitors that were quenched as a function of cosmic time for both currently star-forming and currently quenched galaxies are found for bins of observed stellar mass in qf_sm_histories_sm* and for bins of peak halo mass in qf_hm_histories_hm*. The exact range of stellar masses or halo masses in each bin is detailed in the file header.

2.11 Rejuvenation Statistics

Found in data/rejuvenation. The fractions of galaxies that rejuvenated (i.e., were quenched for at least 300 Myr and then were star-forming for at least 300 Myr thereafter) are found as a function of observed stellar mass in rejuv_a* and as a function of peak halo mass in rejuv_hm_a*; the filename includes the scale factor.

2.12 Average Star Formation Histories

Found in data/sfhs. Average star formation histories for all galaxies, centrals, satellites, star-forming, and quenched galaxies are found in bins of observed stellar mass in sfh_sm* and in bins of peak halo mass in sfh_hm*; the filename includes the mass bin and the scale factor.

2.13 Stellar Mass Functions and Satellite Fractions

Found in data/smfs. The stellar mass function (i.e., galaxy number density) as well as the satellite fraction as a function of observed stellar mass are found in smf_a*, where the filename includes the scale factor. The *Bolshoi-Planck* simulation is incomplete for low-mass galaxies and halos. This incompleteness is significant below $10^7 M_\odot$ at $z = 0$ and $10^{8.5} M_\odot$ at $z = 8$.

2.14 Average Specific Star Formation Rates

Found in data/ssfrs. The average linear ratio of observed SFR to observed stellar mass as a function of observed stellar mass is found in ssfr_a*, where the filename includes the scale factor.

2.15 UV Luminosity Functions

Found for $z \geq 4$ in data/uvlfs and $z < 4$ in data/uvlfs_uncalibrated; the directory naming reflects that $z < 4$ UV luminosities do not have proper dust calibration in the model and are likely incorrect. Galaxy number densities as a function of UV magnitude ($M_{1500,AB}$) are found in uvlf_a*; the filename includes the scale factor. The Bolshoi-Planck simulation becomes increasingly incomplete for $M_{1500} > -19$.

2.16 UV–Stellar Mass Relations

Found for $z \geq 4$ in data/uvsm and $z < 4$ in data/uvsm_uncalibrated; the directory naming reflects that $z < 4$ UV luminosities do not have proper dust calibration in the model and are likely incorrect. Median observed stellar masses as a function of UV magnitude ($M_{1500,AB}$) are found in uvsm_z*; the filename includes the redshift range. The Bolshoi-Planck simulation becomes increasingly incomplete for $M_{1500} > -19$.

2.17 Weak Lensing

Found in data/weak_lensing. Galaxy-galaxy weak lensing shear predictions are found for all galaxies, star-forming galaxies, and quenched galaxies in wl_sm*. The filename includes the scale factor and the lower observed stellar mass threshold; i.e., only galaxies with masses above the threshold in the filename are included. Ratios of weak lensing predictions for quenched to star-forming, star-forming to all galaxies, and quenched to all galaxies are found in wl_ratios_sm*.

3 Mock Catalogs and Lightcones

Mock catalogs and lightcones are available for the best-fit model (see http://www.peterbehroozi.com/data.html), as described below.

3.1 Halo and Galaxy Properties

Halo and galaxy properties at every simulation snapshot are available at SFR/sfr_catalog_* and SFR_ASCII/sfr_catalog_*; the filename includes the scale factor. Halo properties include the halo ID (for cross-matching to *Bolshoi-Planck* halo catalogs), descendant ID, parent ID (for satellites), position, velocity, current mass and $v_{\max}$, peak mass, and $v_{\max}$ at peak mass. Galaxy properties include the true stellar mass in the galaxy, true stellar mass in the intrahalo light, observed stellar mass, observed SFR, observed SSFR, true stellar mass / halo mass ratio, and observed UV luminosity (valid for $z > 4$). Both binary (SFR) and text (SFR_ASCII) versions are available. The binary version is consecutive catalog_halo structures; Python and C loaders are provided in the same directory. See halo.h for the structure definition and print_sm_catalog.c in the UniverseMachine source code for a more advanced example of how to read the binary version. You can also use HaloTools to load the binary catalogs.

3.2 Star Formation Histories

Catalogs with star formation histories are available at specific redshifts (e.g., $z = 0, 1, 2$) in `SFH/sfh_catalog_*`; the filename includes the scale factor. Besides the halo and galaxy properties in §3.1, these files contain star formation histories for the present-day stellar population in the galaxy (i.e., including all merged progenitors), star formation histories for the present-day population in the intrahalo light, the main progenitor galaxy's stellar mass history, the main progenitor's intrahalo light history, the main progenitor's halo mass history, the main progenitor's SFR history (i.e., excluding any mergers), the main progenitor's v_{Mpeak} (i.e., $v_{\max}$ at peak mass), and the main progenitor's $\Delta v_{\max}$ rank (expressed in units of standard deviations). These files are split into many pieces (144 for *Bolshoi-Planck*) to make them easier to analyze in parallel.

3.3 CANDELS Lightcones

Lightcones for the CANDELS fields (EGS, COSMOS, UDS, GOODS-N/S) are available in `CANDELS_Lightcones/survey_*`. The filenames include the field, the redshift range, the width of the lightcone in arcminutes ("x"), the height of the lightcone in arcminutes ("y"), and the lightcone index. Eight lightcones are available for each field—these are separate realizations of each field to aid in estimating sample variance. These lightcones contain the galaxy sky position (RA, Dec, z), the halo ID, lightcone 3D position, velocity, halo mass and $v_{\max}$, galaxy true/observed stellar mass, intrahalo light, true/observed SFR, observed SSFR, true stellar mass to halo mass ratio, observed UV luminosity (only valid at $z > 4$), and UV attenuation (only valid at $z > 4$).

4 Running Basic Analyses

4.1 Compiling

If you use the GNU C compiler version 4.0 or above on a 64-bit machine, compiling should be as simple as typing "`make`" at the command prompt. If you use the Intel C compiler, uncomment the lines `CC=icc` and `OPT_FLAGS=-fast` in the Makefile before running "`make`".

The UniverseMachine does not support compiling on 32-bit machines and has not been tested with other compilers. Additionally, it does not support non-Unix environments. (Mac OS X is fine; Windows is not). If you use the code to convert new merger trees to UniverseMachine format, you will need the GNU Scientific Library (GSL) installed; to compile this code, you should run "`make treereg`".

4.2 Making New Lightcones

Lightcones for arbitrary fields can be generated with the `lightcone` command after compiling. You will need the binary SFR catalogs (§3.1), the config file, and the list of snapshots (`snaps.txt`). After downloading, you'll have to edit the config file so that `INBASE` is the directory path where you've downloaded `snaps.txt` and `OUTBASE` is the directory where the binary SFR catalogs are located. Running the `lightcone` command gives a brief usage statment. `z_low` and `z_high` give the redshift range to generate the lightcone, `x_arcmin` gives the width of the lightcone in arcminutes, `y_arcmin` gives the height of the lightcone in arcminutes, `samples` gives the number of lightcone realizations to generate, and `id_tag` gives optional text to add to the output filename. `do_collision_test`, if specified as 1, will ensure that the lightcone doesn't overlap with itself. This is inadvisable except with very small lightcones; e.g., most volumes of interest will be of comparable size to *Bolshoi-Planck*. `ra` and `dec` give the center of the lightcone on the sky; `theta` gives the additional rotation (in degrees) of the lightcone around this central axis. Finally, `rseed` allows specifying the random seed for generating lightcone positions and orientations within the simulation. This is helpful if you want to generate a lightcone using the same halos for a different UniverseMachine model.

4.3 Making New Catalogs

The script `scripts/make_sf_catalog.pl` will generate new catalogs (as in §3.1) as well as, optionally, new star formation histories for a specified model. Run `perl scripts/make_sf_catalog.pl` for usage information.

You will need the base simulation data (`base/*.*`), the config file, as well as the model parameters (e.g., those in the data release, in `data/obs.txt`). You will need to edit the config file so that `INBASE` is the directory where the base simulation data is located and `OUTBASE` is the directory to which the binary SFR catalogs (as in §3.1) and text SFHs (if specified) should be written. The script will use as many threads as you specify to generate the final catalog; the memory used is about 3GB per thread for `Bolshoi-Planck`. The final SFH catalogs will be in text format, but the final halo/galaxy property catalogs will be in binary format. To convert the latter to text format, you can use the `print_sm_catalog` command. For testing purposes, you can also use the `make_sf_catalog` command directly; this will generate a single piece of the catalog at a time. Run `make_sf_catalog` without options to see usage information.

5 Running on Different Simulations

Currently, preprocessed simulation files are available for *Bolshoi-Planck*, *VSMDPL*, *SMDPL*, and *MDPL2* (see `http://www.cosmosim.org`). To run on another simulation, you will need to start with trees and catalogs generated by a recent version of CONSISTENT TREES (i.e., the catalogs should have a `DeltaLogVmax` column). You will then need to do the following steps:

- In a terminal, print out the first line of one of the `hlist` catalogs for your simulation (`head -n1 hlist_XYZ.list`).
- Open `sims/rhs_bolshoi.h` in a text editor. Follow the instructions in the file for how to modify it to match the first (header) line of your hlist.
- Run "`make treereg`" in the UNIVERSEMACHINE source directory to generate the tree-processing executables. This requires GSL to be installed.
- Run "`perl /path/to/universemachine/scripts/parallel_fit_splines.pl <threads>`" in the directory containing all your hlist files, where `<threads>` is the number of hlist files that can fit into memory simultaneously. This will create a directory called `splines`, with information about the rank ordering of all halos' `DeltaLogVmax` parameters.
- Run "`/path/to/universemachine/split_halo_trees_phase1 <hlist_a1.list> <box_size> <nx> <ny> 1 /path/to/forests.list > boxlist.txt`" Here, `<hlist_a1.list>` is the full name of the $a = 1$ halo catalog, `<box_size>` is the box length in Mpc/h, `<nx>` and `<ny>` are the number of pieces in which the simulation volume will be chopped in the

x and y directions, and `/path/to/forests.list` is the path to the `forests.list` file generated by CONSISTENT TREES (often, `../trees/forests.list`).

- Run "`perl /path/to/universemachine/scripts/parallel_split_halo_trees.pl <box_size> <Om> <h> /path/to/scales.txt /path/to/splines /path/to/boxlist <threads> /path/to/trees/tree_*.dat`." Here, `<box_size>` is the box length in Mpc/h, `<Om>` is the value of Ω_M, `<h>` is the value of h, `/path/to/scales.txt` is the path to the `scales.txt` file used by CONSISTENT TREES (SCALEFILE), `/path/to/splines` is the path to the `splines` directory generated by the `parallel_fit_splines.pl` script, `/path/to/boxlist` is the path to the `boxlist.txt` file generated in the previous step, `<threads>` is the number of trees to read in parallel (each uses about 20 GB of memory), and `/path/to/trees/tree_*.dat` is the path to all the tree files generated by CONSISTENT TREES (often, `../trees/tree_*.dat`). This step reads in all the tree files, gravitationally simulates the paths of orphan subhalos (i.e., subhalos that have been potentially destroyed too early in the simulation), reorders the halos into depth-first order, and prints out binary files in the format that UNIVERSEMACHINE expects. It will generate a directory called `catalog_bin` that contains the binary catalogs, a list of snapshots, and lists of halo counts at each redshift for each binary catalog.

The `catalog_bin` directory generated above can then be used as the `INBASE` directory in your config file. The number of subboxes generated will be the product of `<nx>` and `<ny>`; this should be set as `NUM_BLOCKS` in your config file. You will also have to modify your config file to have the correct `BOX_SIZE` and cosmology (`Om`, `Ol`=Ω_Λ, and `h`).

6 Parameter Space Exploration

6.1 Observational Constraints

Observational constraints are automatically read from the `obs/` subdirectory for the UniverseMachine. You can add new constraints by placing new files in this directory. All files provided should ideally have the same assumptions employed: specifically, a Chabrier IMF, a Calzetti dust law, and the BC03 SPS model. If these are not consistent, the output constraints will not necessarily be physically meaningful.

Each file must be in ASCII format, with at least one header line that specifies the file type (i.e., "`#type: XYZ`"). As of this writing, the allowed file types are: `smf`, `uvlf`, `qf`, `qf11`, `qf_uvj`, `ssfr`, `cosmic sfr`, `cosmic sfr (uv)`, `correlation`, `cross correlation`, `conformity`, `lensing`, `cdens fsf`, `cdens ssfr_sf`, `uvsm`, and `irx`, which indicate the observation type as in Section 2.8. Note that some type names above include spaces, whereas the observation types in Section 2.8 do not; the type names listed here are the correct ones to use. The units are the same as listed in Section 2.8. For types other than cosmic SFRs, the file should also include two lines that specify the redshift range (`#zlow: z1` and `#zhigh: z2`).

The format of the file depends on its type. Each line contains a separate data point. For cosmic SFRs, the format is:

```
z1 z2 log10(csfr) err+ err-
```

where `z1` and `z2` are the redshift range over which the measurement was performed, and `err+` and `err-` are the uncertainties in dex.

For most other files, the format is:

```
m1 m2 log10(obs) err+ err-
```

where `m1` and `m2` are the luminosity or stellar mass range (in magnitudes or $\log_{10}$ solar masses, respectively), and `err+` and `err-` are the errors in dex. The exception is quenched fractions, where the observation and the errors are assumed to be linear instead of logarithmic.

Correlation functions require special treatment, so if adding/replacing existing data, please contact the authors.

6.2 Setting Up the Config File

It is best to start with a sample config file (e.g., `scripts/sample.cfg`) and modify it so that all desired fields are included.

6.2.1 Simulation/Data Setup

```
INBASE = /path/to/simulation/data
```

This should be the directory where all the simulation data files (e.g., `cat.box*`) are stored. It does not need to be writable.

```
OUTBASE = /path/to/output/directory
```

This should be a directory where you would like all outputs to be saved.

```
NUM_BLOCKS = 144
```

This should be the total number of blocks into which the simulation data has been divided. That is, it should be the number of different `cat.box*` files. Note that the indexing of the `cat.box*` files starts from zero, which means that the number of `cat.box*` files will be one more than the largest filename number (e.g., `cat.box143.dat`, in this example).

6.2.2 Box Size/Cosmology

```
BOX_SIZE = 250 #In Mpc/h
Om = 0.307     #Omega_matter
Ol = 0.693     #Omega_lambda
h0 = 0.68      #h0 = H0 / (100 km/s/Mpc)
fb = 0.158     #cosmic baryon fraction
```

These parameters capture the box length and the cosmology. The parameters above are appropriate for the *Bolshoi-Planck* simulation; if using a different one, you may have to change these parameters appropriately.

6.2.3 Parallel Node Setup/MCMC

If performing fits or MCMC exploration, the UniverseMachine needs to be run on a parallel system with enough memory to store all the simulation data files at once. For MCMC exploration, it is preferable to use as many nodes as possible.

```
NUM_NODES = 6          #Total number of nodes used
```

This specifies the total number of nodes used in the calculation.

```
BLOCKS_PER_NODE = 24          #Parallel tasks per node
```

This specifies how many parallel tasks are performed per node. Often, `BLOCKS_PER_NODE` will be the number of cores per node. However, it does not need to be identical, which may be a good idea if, for example, the memory per node is not sufficient. Each node should have memory at least 1.3 x `BLOCKS_PER_NODE` x `B`, where `B` is the average size of a single `cat.box` input file.

The UniverseMachine will generate an integer number of universes in parallel; in addition, each node can only work on one universe at a time. Hence, `BLOCKS_PER_NODE` must divide `NUM_BLOCKS`, and the product of `NUM_NODES` x `BLOCKS_PER_NODE` must be divisible by `NUM_BLOCKS`. E.g., if `NUM_BLOCKS` = 144 and each node has 24 processors, then it would be appropriate to set `BLOCKS_PER_NODE` = 24, and to set `NUM_NODES` to be a multiple of 6 (because 24x6 = 144).

6.3 Finding a Best-Fit Solution

The UniverseMachine interfaces with `scipy` to find best-fitting solutions. Currently, finding the best-fit solution is not parallelized beyond generating a single universe at a time, so `NUM_NODES` should be set to `NUM_BLOCKS / BLOCKS_PER_NODE` in your config file. To perform fits, the following config options should be specified:

```
FITTING_MODE=1
EXTERNAL_FITTING_MODE=1
```

You should compile the UniverseMachine code by running "`make`" in the UniverseMachine directory. You can then start the master fitting process by running (from the UniverseMachine directory):

```
python minimize.py /path/to/um_config.cfg > fit_log.txt
```

This will generate an `auto-um.cfg` file in your `OUTBASE` directory. You should then run the following command once on each node:

```
./box_server /path/to/OUTBASE/auto-um.cfg
```

This is best accomplished using the computing system's distributed execution command, typically `srun`, `mpirun`, `ibrun`, or similar.

If successful, the best-fitting results will be saved in `fit_log.txt`. You can find an example script and config file in `scripts/um_fitting.sh` and `scripts/um_fit.cfg`.

6.4 MCMC Analysis

You should compile the UniverseMachine code by running "`make`" in the UniverseMachine directory. You can then start the MCMC process by running (from the UniverseMachine directory):

```
./master_server /path/to/um_config.cfg > server.log 2>&1
```

This will generate an `auto-um.cfg` file in your `OUTBASE` directory. You should then run the following command once on each node:

```
./box_server /path/to/OUTBASE/auto-um.cfg
```

This is best accomplished using the computing system's distributed execution command, typically `srun`, `mpirun`, `ibrun`, or similar. You can find an example script and config file in `scripts/um_mcmc.sh` and `scripts/sample.cfg`.

During the fitting process, the master server will keep a log file in `OUTBASE/master_server.log` You can view this to check the progress of loading the data files, burning in, and performing MCMC exploration. Burn-in steps are saved in `OUTBASE/burn_in_steps.dat`, and the final MCMC steps are stored in `OUTBASE/mcmc_steps.dat`.

By default, the MCMC algorithm uses 100 MCMC walkers to explore parameter space. (Note that this number is independent from the number of universes generated simultaneously). This number can be adjusted by setting `NUM_WALKERS` to a different number in the config file.