

---

# **Aemulus\_data Documentation**

***Release 0.1***

**Tom McClintock, Joe DeRose, Zhongxu Zhai, Sean McLaughlin, R**

**Jun 05, 2018**



---

## Contents:

---

<b>1</b>	<b>Cosmologies</b>	<b>3</b>
<b>2</b>	<b>Particles and Halos</b>	<b>5</b>
<b>3</b>	<b>Halo Mass Function</b>	<b>7</b>
<b>4</b>	<b>Galaxy-galaxy correlation function at <math>z = 0.57</math></b>	<b>9</b>
<b>5</b>	<b>Halo Occupation Distribution with Assembly Bias</b>	<b>11</b>
<b>6</b>	<b>Halo Mass Function Data</b>	<b>13</b>
<b>7</b>	<b>HOD Configurations</b>	<b>15</b>
<b>8</b>	<b>aemulus_data</b>	<b>17</b>
8.1	aemulus_data package . . . . .	17
<b>9</b>	<b>Citation Information</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>



The Aemulus simulations are a suite of simulations used for cosmological inference. They form the building blocks of a set of predictors, cosmic emulators, used to model various cosmological observables.

The emulators are located in various repositories associated to this one and include:

- Halo mass function  $\frac{dn}{dM}(M, z)$
- Galaxy-galaxy correlation function at  $\xi_{gg}(r, z = 0.57)$
- Halo occupation distribution (HOD) and assembly bias models  $P(N|M, x)$
- Matter power spectrum  $P(k)$  (in development)
- Matter-matter correlation functions  $\xi_{mm}$  (in development)
- Halo-matter correlation functions  $\xi_{hm}$  (in development)
- Scale dependent halo bias  $b(M, r)$  (in development)

Here you will find details on how to use these various emulators and some of the theory behind them. In addition, this repository contains the data used to train the Gaussian Processes that make up the emulators as well as the routines used to create some of the figures used in our papers.



# CHAPTER 1

---

## Cosmologies

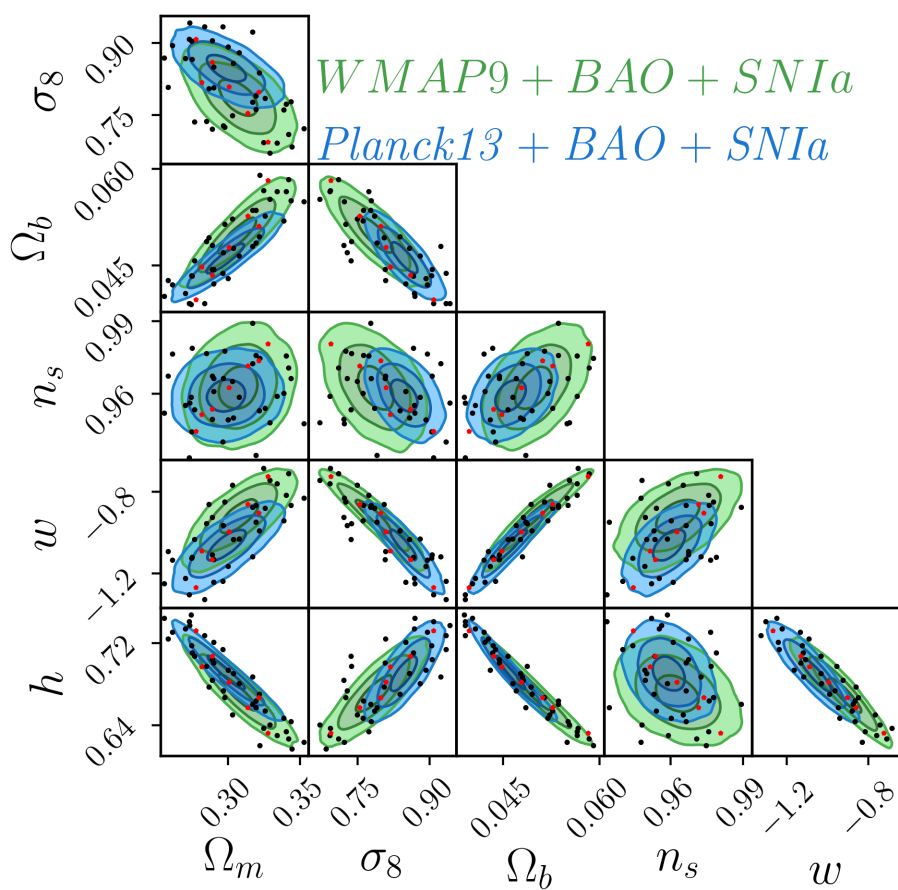
---

The Aemulus simulations used to build the emulators were all run with different sets of cosmological parameters. As detailed in DeRose et al. (in prep.), these cosmologies were designed to span the space of [Planck13](#) + [WMAP9](#) + [BAO](#) + [SNIa](#). The available parameters are  $\Omega_b, \Omega_m, w, n_s, H_0, \sigma_8, N_{\text{eff}}$ .

To access these cosmologies you can do

```
import aemulus_data as AD
import numpy as np
path = AD.path_to_building_box_cosmologies()
cosmologies = np.genfromtxt(path)
```

The figure below shows the locations of all parameters except  $N_{\text{eff}}$  which was not sampled by the Planck/WMAP chains.





## CHAPTER 2

---

### Particles and Halos

---

Particle catalogs and full halo catalogs are available on SLAC servers (in progress).

This repository does provide reduced halo catalogs for all cosmologies containing high mass halos of  $M > 5 \times 10^{14} \text{ M}_{\odot} h^{-1}$ .

in progress



## CHAPTER 3

---

### Halo Mass Function

---

in progress



---

### Galaxy-galaxy correlation function at $z = 0.57$

---

in progress



---

## Halo Occupation Distribution with Assembly Bias

---

in progress





## CHAPTER 6

---

### Halo Mass Function Data

---

in progress



## CHAPTER 7

---

### HOD Configurations

---

in progress



## 8.1 aemulus\_data package

### 8.1.1 Module contents

This file provides simple functions to get paths to various data.

`aemulus_data.big_box_binned_mass_function(box, snapshot)`

The binned mass function for a snapshot of a big box.

Units are Msun/h. Columns are M\_low, M\_high, Number, Total\_Mass. To get the average mass of halos in a bin divide Total\_Mass/Number.

**Parameters**

- **box** (*int*) – Index of the big box; from 0-6.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** Nbinsx4 array of binned mass function data.

**Return type** numpy.array

`aemulus_data.big_box_binned_mass_function_covariance(box, snapshot)`

The covariance matrix for the binned mass function for a snapshot of a big box.

Units are Msun/h.

**Parameters**

- **box** (*int*) – Index of the big box; from 0-39.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** NbinsxNbins symmetric covariance matrix.

**Return type** numpy.array

`aemulus_data.building_box_binned_mass_function(box, snapshot)`

The binned mass function for a snapshot of a box.

Units are Msun/h. Columns are M\_low, M\_high, Number, Total\_Mass. To get the average mass of halos in a bin divide Total\_Mass/Number.

**Parameters**

- **box** (*int*) – Index of the simulation box; from 0-39.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** Nbinsx4 array of binned mass function data.

**Return type** numpy.array

`aemulus_data.building_box_binned_mass_function_covariance(box, snapshot)`

The covariance matrix for the binned mass function for a snapshot of a simulation box.

Units are Msun/h.

**Parameters**

- **box** (*int*) – Index of the simulation box; from 0-39.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** NbinsxNbins symmetric covariance matrix.

**Return type** numpy.array

`aemulus_data.building_box_cosmologies()`

Cosmologies for the building boxes aka the aemulus simulations.

Columns are: Omega\_bh^2 Omega\_ch^2 w0 ns ln10As H0[km/s/Mpc] Neff sigma8.

**Returns** 40 by 8 array of the cosmologies for each simulation.

**Return type** numpy.array

`aemulus_data.highres_box_binned_mass_function(box, snapshot)`

The binned mass function for a snapshot of a highres box.

Units are Msun/h. Columns are M\_low, M\_high, Number, Mean\_Mass.

**Parameters**

- **box** (*int*) – Index of the medium box; 11 or 14.
- **snapshot** (*int*) – Index of the snapshot; from 0-13.

**Returns** Nbinsx4 array of binned mass function data.

**Return type** numpy.array

`aemulus_data.highres_box_binned_mass_function_covariance(box, snapshot)`

The covariance matrix for the binned mass function for a snapshot of a highres box.

Units are Msun/h.

**Parameters**

- **box** (*int*) – Index of the medium box; 11 or 14.
- **snapshot** (*int*) – Index of the snapshot; from 0-13.

**Returns** symmetric covariance matrix.

**Return type** numpy.array

`aemulus_data.highres_box_cosmologies()`

Cosmologies for the highres boxes.

Note: this doesn't contain sigma8.

Columns are: Omega\_bh^2 Omega\_ch^2 w0 ns ln10As H0[km/s/Mpc] Neff

**Returns** 40 by 7 array of the cosmologies for each simulation.

**Return type** numpy.array

`aemulus_data.highres_scale_factors()`

Scale factors of snapshots of the highres simulations.

Note: these are not the same scale factors as those of the building and test boxes.

**Returns** Scale factors of highres snapshots.

**Return type** array

`aemulus_data.individual_test_box_binned_mass_function(box, snapshot, realization)`

The binned mass function for a snapshot of a test box.

Units are Msun/h. Columns are M\_low, M\_high, Number, Total\_Mass. To get the average mass of halos in a bin divide Total\_Mass/Number.

**Parameters**

- **box** (*int*) – Index of the test box; from 0-6.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.
- **realization** (*int*) – Index of the realization; from 0-4.

**Returns** Nbinsx4 array of binned mass function data.

**Return type** numpy.array

`aemulus_data.individual_test_box_binned_mass_function_covariance(box, snapshot, realization)`

The covariance matrix for the binned mass function for a snapshot of a test box.

Units are Msun/h.

**Parameters**

- **box** (*int*) – Index of the test box; from 0-39.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.
- **realization** (*int*) – Index of the realization; from 0-4.

**Returns** NbinsxNbins symmetric covariance matrix.

**Return type** numpy.array

`aemulus_data.medium_box_binned_mass_function(box, snapshot)`

The binned mass function for a snapshot of a medium box.

Units are Msun/h. Columns are M\_low, M\_high, Number, Total\_Mass. To get the average mass of halos in a bin divide Total\_Mass/Number.

**Parameters**

- **box** (*int*) – Index of the medium box; from 0-6.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** Nbinsx4 array of binned mass function data.

**Return type** numpy.array

`aemulus_data.medium_box_binned_mass_function_covariance(box, snapshot)`

The covariance matrix for the binned mass function for a snapshot of a medium box.

Units are Msun/h.

**Parameters**

- **box** (*int*) – Index of the medium box; from 0-39.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** NbinsxNbins symmetric covariance matrix.

**Return type** numpy.array

`aemulus_data.scale_factors()`

Scale factors of snapshots.

**Returns** Scale factors of the snapshots.

**Return type** array

`aemulus_data.test_box_binned_mass_function(box, snapshot)`

The binned mass function for a snapshot of a test box.

Units are Msun/h. Columns are M\_low, M\_high, Number, Total\_Mass. To get the average mass of halos in a bin divide Total\_Mass/Number.

**Parameters**

- **box** (*int*) – Index of the test box; from 0-6.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** Nbinsx4 array of binned mass function data.

**Return type** numpy.array

`aemulus_data.test_box_binned_mass_function_covariance(box, snapshot)`

The covariance matrix for the binned mass function for a snapshot of a test box.

Units are Msun/h.

**Parameters**

- **box** (*int*) – Index of the test box; from 0-39.
- **snapshot** (*int*) – Index of the snapshot; from 0-9.

**Returns** NbinsxNbins symmetric covariance matrix.

**Return type** numpy.array

`aemulus_data.test_box_cosmologies()`

Cosmologies for the test boxes.

Columns are: Omega\_bh^2 Omega\_ch^2 w0 ns ln10As H0[km/s/Mpc] Neff

**Returns** 7 by 8 array of the cosmologies for each simulation.

**Return type** numpy.array



## CHAPTER 9

---

### Citation Information

---

If you use any of these emulators in your work please cite DeRose et al. (in prep) which introduced the simulations and the associated paper for that emulator.

- Mass function - McClintock et al. (in prep.)
- Galaxy-galaxy correlation function at  $z = 0.57$  - Zhai et al. (in prep.)
- HOD - McLaughlin et al. (in prep.)



### **a**

`aemulus_data`, [17](#)



## A

`aemulus_data` (module), [17](#)

## B

`big_box_binned_mass_function()` (in module `aemulus_data`), [17](#)

`big_box_binned_mass_function_covariance()` (in module `aemulus_data`), [17](#)

`building_box_binned_mass_function()` (in module `aemulus_data`), [17](#)

`building_box_binned_mass_function_covariance()` (in module `aemulus_data`), [18](#)

`building_box_cosmologies()` (in module `aemulus_data`), [18](#)

## H

`highres_box_binned_mass_function()` (in module `aemulus_data`), [18](#)

`highres_box_binned_mass_function_covariance()` (in module `aemulus_data`), [18](#)

`highres_box_cosmologies()` (in module `aemulus_data`), [18](#)

`highres_scale_factors()` (in module `aemulus_data`), [19](#)

## I

`individual_test_box_binned_mass_function()` (in module `aemulus_data`), [19](#)

`individual_test_box_binned_mass_function_covariance()` (in module `aemulus_data`), [19](#)

## M

`medium_box_binned_mass_function()` (in module `aemulus_data`), [19](#)

`medium_box_binned_mass_function_covariance()` (in module `aemulus_data`), [20](#)

## S

`scale_factors()` (in module `aemulus_data`), [20](#)

## T

`test_box_binned_mass_function()` (in module `aemulus_data`), [20](#)

`test_box_binned_mass_function_covariance()` (in module `aemulus_data`), [20](#)

`test_box_cosmologies()` (in module `aemulus_data`), [20](#)