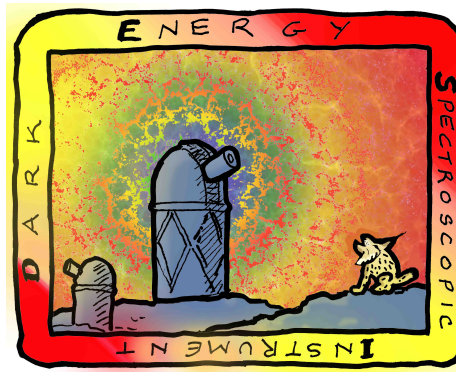




# Documentation

Documentation of all the scripts and functions used to investigate redshift incompleteness due to galaxies that are not assigned fibres



Part of the *DESI* (Dark Energy Spectroscopic Instrument) project [desi.lbl.gov](http://desi.lbl.gov) and its *Bright Galaxy Survey* group.

HUSNI ALMOUBAYYED

Email: [Husni@Physics.org](mailto:Husni@Physics.org)

## 1. Code

### 1.1. *make\_catalogues.py*

`make_catalogues.py` is a function to write the catalogues suitable to be used in CUTE. `make_cat()`, the function inside `make_catalogues.py` assumes that the full mock catalogue `aardvark_mockcat.hdf5`, and the full random catalogue `rancat.hdf5` are in the working directory and it writes the new catalogues to the working directory. The function takes in the following arguments.

**1.1.1. *magnitude*** The r-band magnitude for the catalogues.

**3** for  $r = 19.3$

**5** for  $r = 19.5$

**7** for  $r = 19.7$

**9** for  $r = 19.9$

### 1.1.2. *weights*

**weights=0** does not assign any weights to the new catalogues.

**weights=1** assigns unity weights for all the galaxies in the catalogues.

**weights=2** assigns redshift-dependant weights to every galaxy given by the equation:

$$w_i = \frac{1}{1 + 4\pi J_3 N(z)} \quad (1)$$

it uses the same weights for the random catalogue.

**weights=3** uses the function `cell_weights()` from `grid.py` to break the catalogue into cells of  $ra$  and  $\sin(dec)$ . At the moment this is set to 22 cells in  $ra$  and 12 cells in  $\sin(dec)$  (about 6 degrees in each) assign weights for each galaxy in a given cell using the ratio in equation 2 for the same cell.

$$w_c = \frac{\text{Number of galaxies in complete catalogue}}{\text{Number of galaxies in targeted catalogue}} \quad (2)$$

**weights=4** is a combination of 2 and 3.

**1.1.3. *debug*** using **debug = True** would get the function to plot the weights as it is going. This is very recommended to ensure the weights are being assigned as intended. There is also an option **superdebug = True** that would print the input/output catalogues (default is *False*)

**1.1.4. *skip\_randoms*** **skip\_randoms = True** does not assign weights for the random catalogues (default is *False*)

1.1.5. *same\_weights* **same\_weights = True** is used to assign the same weights that are calculated to the complete catalogue to the targeted catalogue (default is *False*).

### 1.2. *woftheta.c*

Under `/cosma/home/icc-sum1/CUTE2/CUTE/src/` you can find the function `woftheta.c` and its header `woftheta.h`, this function is implemented in CUTE2 to add pair weights of

$$w_{\theta} = \frac{1 + \omega(\theta)|_{all}}{1 + \omega(\theta)|_{targeted}} \quad (3)$$

depending on the angular separation between each two galaxies. The function takes a set of values in variable *yy* for this ratio and interpolates between them. *yy* can be changed for different magnitude limits etc. `woftheta.c` is implemented in  $\xi(r)$ ,  $\omega(\theta)$ , and  $\xi(\sigma, \pi)$ . It is defined in `common.h` and `Makefile` and edited into `correlator.c`.

**CUTE**, or Correlation Utilities and Two-point Estimates, is written by David Alonso (c). For more information on CUTE, see: [arXiv:1210.1833v2](https://arxiv.org/abs/1210.1833v2)

### 1.3. *grid.py*

Under `/cosma/home/icc-sum1/grid/` you can find the function `cell_counts` in `grid.py`. By inputting the number of cells wanted in ra and sin(dec) and the paths to the complete and targeted catalogues (more comments in `grid.py`), this function will count the number of galaxies in each cell and output the ratio of number of galaxies in the complete catalogue over the targeted catalogue as well as the boundaries of the cells in ra and sin(dec). `make_catalogues.py` calls this function to assign cell weights  $w_c$  to galaxies. This function takes the following arguments

1.3.1. *complete* Path to the desired complete catalog.

1.3.2. *targeted* Path to the desired targeted catalog.

1.3.3. *numra* The number of cells in right ascension

1.3.4. *numdec* The number of cells in sin (declination)

1.3.5. *debug=False* Optional argument; plots the central cell if set to *True*.

The function outputs a tuple including the weights, boundaries in ra, boundaries in sin(dec), number of galaxies per cell in complete catalog, number of galaxies per cell in targeted catalog; in that order.

#### 1.4. *grid\_output.py*

Under `/cosma/home/icc-sum1/grid/` there is also a function `grid_output.py` written to make it easier to plot the outputs from `grid.py`. This function takes the number of cells in ra and sin(dec), magnitude, bins (default=20), interpolation (default='none') and plots the ratio and the number per cell per square degrees in the both catalogues and the galaxy distribution. More information on how `grid_output` reads the catalogues and so on in `grid_output.py`.

#### 1.5. *ang\_correction\_calc.py*

Under `/gpfs/data/icc-sum1/Catalogues` I wrote a function named `ang_cor()` in `ang_correction_calc.py` to create the  $\omega_\theta$  correction (see section 1.2). The function assumes the correlation text files are under the respective subfolders in Catalogues/ and uses logarithmic binning. The function takes the following arguments:

**1.5.1. *magnitude*** The r-band magnitude. **3, 5, 7 or 9** (for `r = 19.magnitude`)

**1.5.2. *cell\_weights*** If set to `cell_weights = True`, it will use correlations that have been made after the cell correction.  $w_c$ . This is recommended as to not double count.

**1.5.3. *j3\_weights*** If set to `j3_weights = True`, it will use correlations that have been made after the  $w_i$ . Applying  $w_i$  weights are not very meaningful for  $\omega(\theta)$  so this option is mainly for comparison of the goodness of  $\omega(\theta)$ .

**1.5.4. *debug*** If set to `debug = True` the function will plot the  $\omega_\theta$  ratio against  $\theta$ .

This function returns an array of the  $\omega(\theta)$  values that can be used to replace `yy` in `woftheta.c` above (subsection 1.2).

#### 1.6. *show\_results.py*

Under `/gpfs/data/icc - sum1/Catalogues/`, `show_results.py` has two functions. `show_results_single()` and `show_results_triple` to plot and compare the correlations. The latter has two extra arguments to plot 3 correlations (complete; and targeted before and after the correction). The function takes the following arguments:

**magnitude** 3, 5, 7 or 9.

**cortype** Type of correlation, 'Angular' or 'Monopole'.

**binning** Type of binning, 'LogBins' or 'LinBins'

**correlation** For the triple function, correlation is the complete correlation, the remaining arguments for the triple function are:

**targeted\_raw**

**targeted\_corrected** The last three arguments should be one of the following strings:

'all\_weights', 'j3\_weights', 'j3\_cell\_weights', 'j3\_woftheta\_weights', 'cell\_weights',  
'no\_weights', 'woftheta\_weights', 'woftheta\_cell\_weights';

with Complete/ or Targeted/ added before these strings for the single function.

## 2. Correlations and Reproducibility of Results

I have created all possible combinations of corrections and variations of correlation functions under */gpps/data/icc-sum1/Catalogues*. They are organised in hierarchically depending on magnitude, type of correlation, type of binning, completeness, and type(s) of corrections, in that order. The options to change the binning is in the **Makefile** and the option to change the type of correlation is in the **param.txt** file. In every subdirectory there is all the files needed to recreate that correlation and a **Readme** file describing these files. Each subdirectory also has the version of CUTE that is suitable to reproduce it. To change the angular separation-dependant correction  $w_\theta$ , you will need to recompile CUTE (see section 1.2 above).