ECO Mocks Catls Documentation

Release 0.1

Victor Calderon

Feb 22, 2018

Contents

This is the documentation for the ECO Mocks Catalogue repository. In here, you will find the structure and functions used in this repository, as well as information regarding the three different surveys.

Contents:

CHAPTER 1

Mock Catalogues

This is a brief overview of the different aspects of the synthetic catalogues produced for ECO RESOLVE-A and RESOLVE-B surveys

For a **more** comprehensive discussion on how the synthetic mocks were created, you can read the ECO and Resolve Synthetic Catalogue guide.

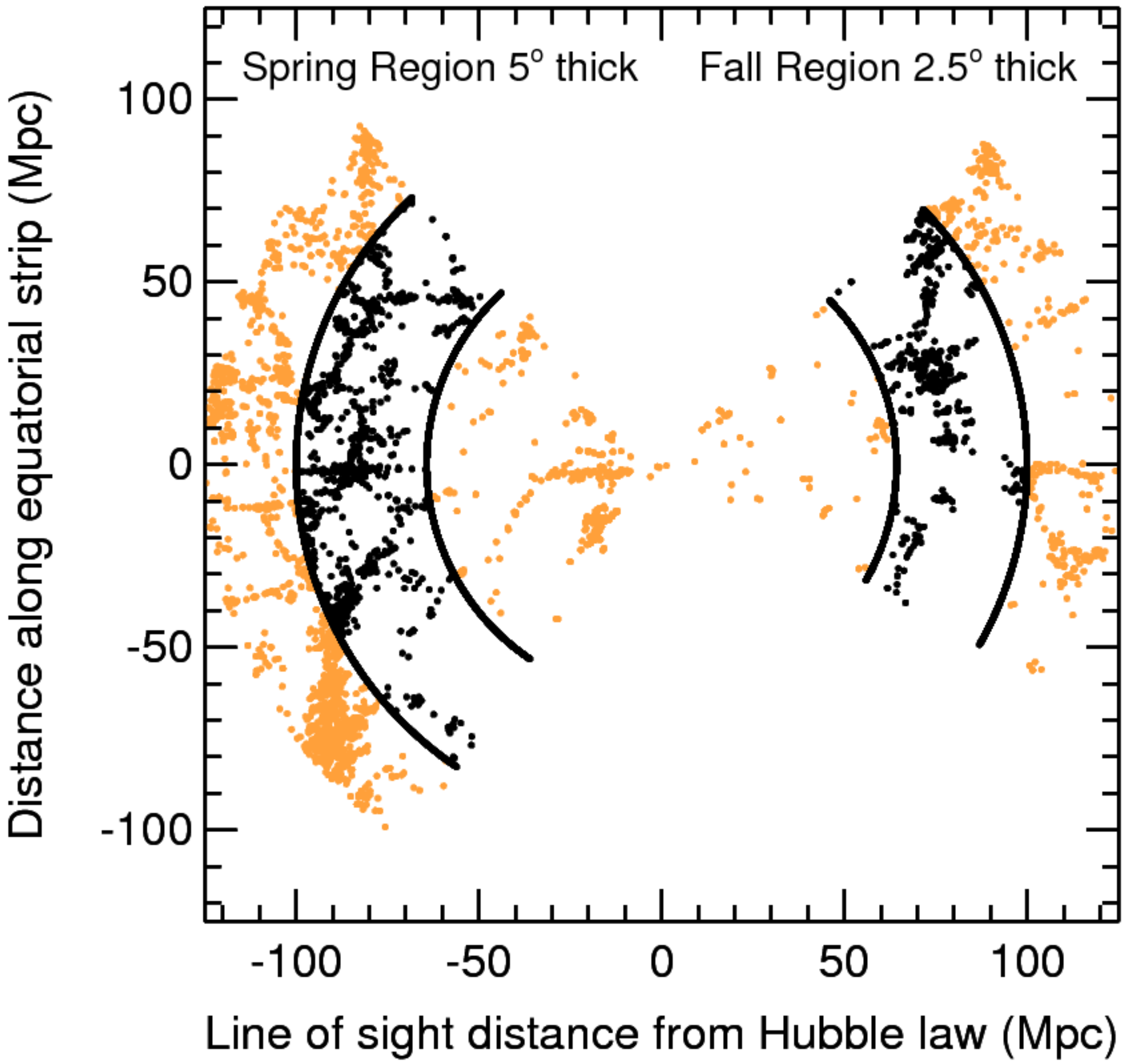
Table of Contents

1.1 ECO and RESOLVE

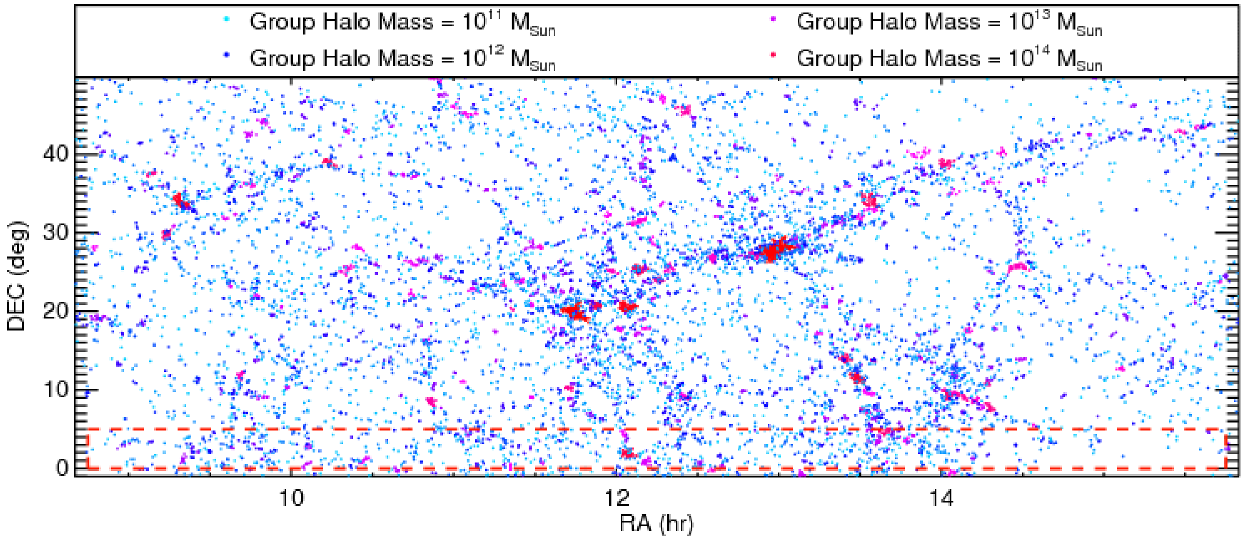
We construct a set of synthethic (mock) catalogues that have the same geometries as the **Environmental COntext** (ECO), **RESOLVE-A**, and **RESOLVE-B** galaxy surveys.

REsolved Spectroscopy Of a Local VolumE (RESOLVE) is a volume-limited census of stellar, gas, and dynamical mass as well as star formation and merging within >50,000 cubic Mpc of the nearby cosmic web, reaching down to the dwarf galaxy regime and up to structures on tens of Mpc scales such as filaments, walls, and voids.

The Environmental COntext (ECO) catalog around RESOLVE is a much larger, purely archival data set with pipelines and methods matched to RESOLVE, enabling statistically robust analyses of environmental trends and calibration of cosmic variance.



This shows the right-ascension (RA) and declination (DEC) of galaxies in RESOLVE-A and RESOLVE-B galaxy redshift surveys.



RESOLVE-A (footprint demarcated by red dashed lines) embedded within ECO (entire plot showing current footprint, with ECO-B in preparation)

For more information on how the data for the different galaxy surveys were taken, go to the Main ECO and RESOLVE website.

1.2 Constructing catalogues

We design the *synthetic* catalogues to have the exact same geometries and redshift limits as those of the ECO, RESOLVE-A, and RESOLVE-B galaxy surveys.

This is a summary of the values used to create the synthetic galaxy catalogues. These catalogues are taking a *buffer* regions, which is an *extra* buffer region along the *cz* (velocity) direction in redshift-space.

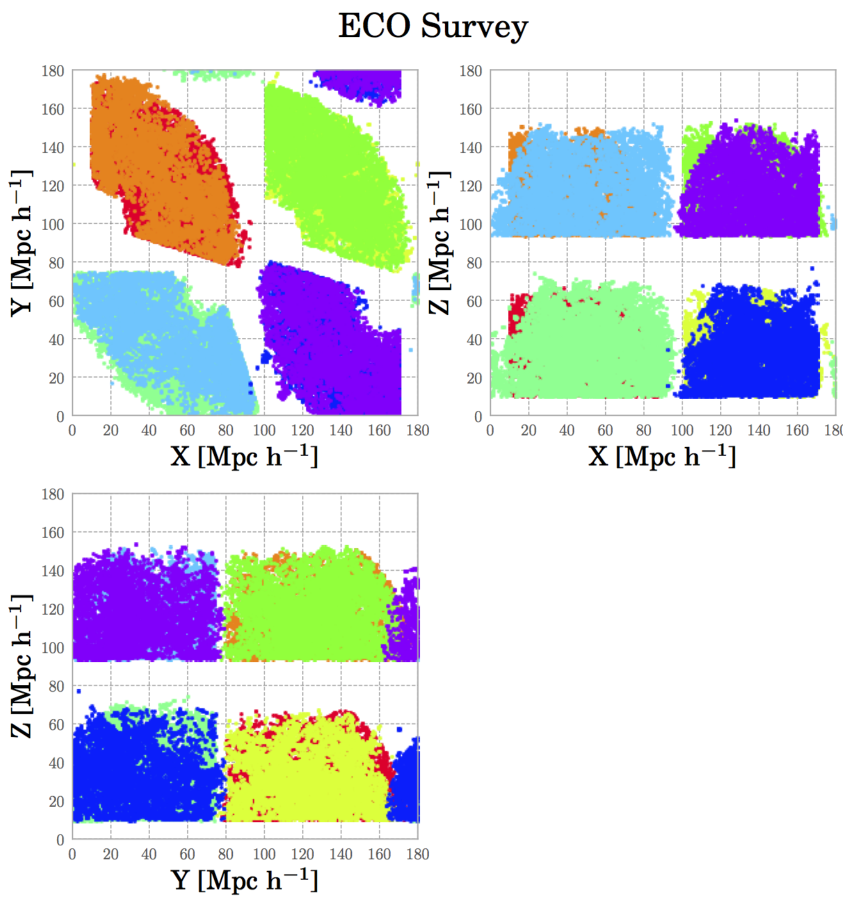
Survey	RA (deg)	RA range	DEC (deg)	DEC range	zmin	zmax	Vmin (km/s)	Vmax (km/s)	Dist (Mpc)
A	(131.25, 236.25)	105.0	(0 ,+5)	5	0.00844	0.0249	2532	7470.	(25.32,70.02)
B	(330.0 , 45.0)	75.0	(-1.25,+1.25)	2.5	0.01416	0.024166	4250	7250.	(42.5 , 72.5)
ECO	(130.05, 237.45)	107.4	(-1, +49.85)	50.85	0.00844	0.0249	2532	7470.	(25.32,70.02)

The next table provides the number of synthetic catalogues per cubic box of **L = 180 Mpc/h**, where $h = 1$.

Survey	Number Mocks
A	59
B	104
ECO	8

1.3 Distribution of catalogues in simulation box

In order to maximize the number of catalogues per simulation, we have to fit as many catalogues as we can, while keeping a distance of ~10 Mpc/h between catalogues. We chose this distance of 10 Mpc/h in order to avoid using the same galaxy for different catalogues, and also to make the catalogues as independent from each other as possible.



This figure shows how the catalogues for ECO surveys are organized within the simulation box used for this analysis.

1.4 Downloading and reading in data from catalogues

The mock catalogues are located at http://lss.phy.vanderbilt.edu/groups/data_eco_vc/Mock_Catalogues/.

These catalogues can be downloaded as *tar* files, and be read by the Python package Pandas.

After having downloaded your file, you can read them in the following way:

```
#! /usr/bin/env python

import pandas as pd
import os

def reading_catls(filename, catl_format='.hdf5'):
    """
    Function to read ECO/RESOLVE catalogues.
```

```
    Parameters
    ----------
    filename: string
        path and name of the ECO/RESOLVE catalogue to read

    catl_format: string, optional (default = '.hdf5')
        type of file to read.
        Options:
            - '.hdf5': Reads in a catalogue in HDF5 format

    Returns
    -------
    mock_pd: pandas DataFrame
        DataFrame with galaxy/group information

    Examples
    --------
    # Specifying `filename`
    >>> filename = 'ECO_catl.hdf5'

    # Reading in Catalogue
    >>> mock_pd = reading_catls(filename, format='.hdf5')

    >>> mock_pd.head()
               x          y         z          vx          vy          vz  \
    0  10.225435  24.778214  3.148386  356.112457 -318.894409  366.721832
    1  20.945772  14.500367 -0.237940  168.731766   37.558834  447.436951
    2  21.335835  14.808488  0.004653  967.204407 -701.556763 -388.055115
    3  11.102760  21.782235  2.947002  611.646484 -179.032089  113.388794
    4  13.217764  21.214905  2.113904  120.689598  -63.448833  400.766541

       loghalom  cs_flag  haloid  halo_ngal    ...        cz_nodist      vel_tot  \
    0    12.170        1  196005          1    ...      2704.599189   602.490355
    1    11.079        1  197110          1    ...      2552.681697   479.667489
    2    11.339        1  197131          1    ...      2602.377466  1256.285409
    3    11.529        1  199056          1    ...      2467.277182   647.318259
    4    10.642        1  199118          1    ...      2513.381124   423.326770

           vel_tan     vel_pec     ra_orig  groupid    M_group g_ngal  g_galtype  \
    0   591.399858 -115.068833  215.025116        0  11.702527      1          1
    1   453.617221  155.924074  182.144134        1  11.524787      4          0
    2  1192.742240  394.485714  182.213220        1  11.524787      4          0
    3   633.928896  130.977416  210.441320        2  11.502205      1          1
    4   421.064495   43.706352  205.525386        3  10.899680      1          1

       halo_rvir
    0   0.184839
    1   0.079997
    2   0.097636
    3   0.113011
    4   0.057210
    """
    ## Checking if file exists
    if not os.path.exists(filename):
        msg = '`filename`: {0} NOT FOUND! Exiting..'.format(filename)
        raise ValueError(msg)
    ## Reading file
    if catl_format=='.hdf5':
```

```
        mock_pd = pd.read_hdf(filename)
        else:
                msg = '`catl_format` ({0}) not supported! Exiting...'.format(catl_
↪format)
                raise ValueError(msg)

    return mock_pd

def main():
    # Specifying filename
    filename = 'ECO_catl.hdf5'
    # Reading in ECO/RESOLVE catalogue
    mock_pd = reading_catls(filename)


if __name__=='__main__':
    main()
```

1.5 Description of the *fields* in the catalogues

Each mock catalogues contains information about the **galaxy**, **group galaxy**, **host halo**, and more. We will denote *dark matter* as *DM*.

Note: The descriptions for the variables are somewhat long, so don't forget to scroll to the **right** to see *more*.

1.5.1 *Main* Galaxy Properties

Table 1.1: List of Parameters

Field	Description	Units
`ra`	Right Ascension	degrees
`dec`	Declination	degrees
`cz`	Velocity of the galaxy (**with redshift-space distortions)	km/s
`M_r`	r-band absolute magnitude of the galaxy	magnitudes
`haloid`	Dark matter halo ID, as taking from the simulation	None
`loghalom`	logarithmic value of the DM's mass	log(Msun/h) where h=1
`halo_ngal`	Total number of galaxies in DM halo. Number of galaxies in the mock may differ from this value	None
`cs_flag`	Type of galaxy. **Halo central** = 1, **Halo satellite** = 0	None
`cz_nodist`	Velocity of the galaxy (*without* redshift-space distortions)	km/s
`dist_c`	*Real* distance between halo's central galaxy and the galaxy.	Mpc/h with h=1
`vel_tot`	Total velu for **peculiar** velocity	km/s
`vel_tan`	Tangential component of the peculiar velocity	km/s
`morph`	Galaxy's morphology. 'LT': *Late Type*; 'ET': *Early type*. Used either *goodmorph* (ECO) or *MORPH* (RESOLVE) keys. '-9999' if no matched galaxy	None
`logmstar`	Log value of galaxy's stellar mass. Used either '*rpgoodmstarsnew*' (ECO) or '*MSTARS*' (RESOLVE) keys in the files	log(Msun)
`rmag`	r-band *apparent* magnitude. Used either 'rpsmoothrestrmagnew' (ECO) or '*SMOOTHRESTRMAG*' (RESOLVE) keys in the files.	magnitudes
`umag`	u-band *apparent* magnitude. Used either 'rpsmoothrestumagnew' (ECO) or '*SMOOTHRESTUMAG*' (RESOLVE) keys in the files.	magnitudes
`fsmgr`	Stellar mass produced over last Gyr divided by pre-existing stellar mass from new model set. Used '*rpmeanssfr*' (ECO) or '*MODELFSMGR*' (RESOLVE) keys.	(1/Gyr)
`survey_flag`	Survey name, from which the properties of the **real** matched galaxy were extracted.	None
`u_r`	Color of the matched galaxy, i.e. (`umag` - `rmag`)	magnitudes
`mhi`	HI mass in galaxy. Used the *predicted* HI massed (matched to the ECO file, i.e. `eco_wresa_050815.dat`) and the key "*MHI*" (RESOLVE). To compute MHI masses using *ECO*, we used the formula: 10^(`MHI` + `logmstar`)	Msun
`groupid`	Group ID, to which the galaxy belongs after running *Berlind2006* FoF group finder.	None
`g_ngal`	Number of galaxies in a group of galaxies	None
`halo_rvir`	Virial radius of the DM halo, to which the galaxy belongs.	Mpc/h with *h* = 1.
`M_group`	Abundance matched mass of the galaxy group. This was calculated by assuming a monotonic relation between DM halo mass `logM_halo` and the group *total* luminosity. For RESOLVE-B, we used a modified version of the *ECO* group luminosity function.	Msun/h with *h* = 1
`g_galtype`	Type of galaxy. **Group central** = 1, **Group satellite** = 0	None

Note: The relationship between velocities (`cz`'s') is the following: (`cz` - `cz_nodist`)^2 + (`vel_tan`)^2 =

(vel_tot)^2.

1.5.2 Halos Filaments

Author: Roberto Gonzales [regonzar@astro.puc.cl or regonzar@oddjob.uchicago.edu]

Affiliation: The University of Chicago, Universidad Católica de Chile

This file includes data about the filaments found in the simulation box used for these synthetic catalogues.

The catalogue can be found at http://lss.phy.vanderbilt.edu/groups/data_eco_vc/Halo_Filaments/.

Table 1.2: List of Parameters - Halo Filaments

Field	Description	Units
`Halo ID`	Halo ID number for the given DM halo in the simulation box.	
`log(MHalo)`	Logarithmic value of the DM halo's mass, as	log(Msun/h) with *h* = 1
`ID/Type`	ID of the DM halo's environment. '0': Not in a filament; '1': filament node; '2': part of a filament skeleton; '3': within a clode radius of a filament.	
`Fil`	ID of the halo's filament. (-1 if not in a filament)	
`Fil. Quality`	Quality of the filament, i.e. probability that the filament is *real*	