SDSS$_{C}onformityDocumentation$

Release 0.1

Victor Calderon

Feb 11, 2019

Contents

Analysis for the signature of **galactic conformity** in the Sloan Digital Sky Survey (SDSS) Data Release 7 (DR7).

The documentation details the codes used in Calderon et al. (2018) for the analysis of 1- and 2-halo galactic conformity using a set of different statistics on SDSS DR7 and synthetic catalogues.

This documentation is part of the repository SDSS_Conformity_Analysis.

CHAPTER 1

Contents

1.1 Getting started

Repository for the analysis of **Galactic Conformity** in SDSS DR7.

Author: Victor Calderon (victor.calderon@vanderbilt.edu)

1.1.1 Downloading Repository

The first thing that needs to be done is to download the repository from Github: https://github.com/vcalderon2009/SDSS_Conformity_Analysis:

```
git clone https://github.com/vcalderon2009/SDSS_Conformity_Analysis.git
```

This will download all of the necessary scripts to run the analysis on the SDSS DR7 catalogues.

1.1.2 Installing Environment & Dependencies

To use the scripts in this repository, you **must have Anaconda installed** on the system that will be running the scripts. This will simplify the process of installing all the dependencies.

For reference, see: Anaconda - Managing environments

The package counts with a **Makefile** with useful functions. You must use this Makefile to ensure that you have all the necessary *dependencies*, as well as the correct **conda environment**.

Once Anaconda has been installed, you can use the **Makefile** to

- Install the Anaconda environment `conformity`.
- Update the project environment `conformity`.
- Install the `src` package via *pip*.

Makefile functions

- Show all available functions in the *Makefile*

```
$:  make show-help

  1_halo_fracs_calc   1-halo Quenched Fractions - Calculations
  1_halo_mcf_calc     1-halo Marked Correlation Function - Calculations
  2_halo_fracs_calc   2-halo Quenched Fractions - Calculations
  2_halo_mcf_calc     2-halo Marked Correlation Function - Calculations
  clean               Deletes all build, test, coverage, and Python artifacts
  clean-build         Remove build artifacts
  clean-pyc           Removes Python file artifacts
  clean-test          Remove test and coverage artifacts
  cosmo_utils_install Installing cosmo-utils
  cosmo_utils_remove  Removing cosmo-utils
  cosmo_utils_upgrade Upgrading cosmo-utils
  download_dataset    Download required Dataset
  environment         Set up python interpreter environment - Using environment.yml
  lint                Lint using flake8
  plot_figures        Figures
  remove_calc_screens Remove Calc. screen session
  remove_catalogues   Remove downloaded catalogues
  remove_environment  Delete python interpreter environment
  remove_plot_screens Remove Plot screen session
  src_env             Import local source directory package
  src_remove          Remove local source directory package
  src_update          Updated local source directory package
  test_environment    Test python environment is setup correctly
  update_environment  Update python interpreter environment
```

- **Create** the environment from the *environment.yml* file:

```
$:  make environment
```

- **Activate** the new environment **conformity**.

```
$:  source activate conformity
```

- To **update** the *environment.yml* file (when the required packages have changed):

```
$:  make update_environment
```

- **Deactivate** the new environment:

```
$:  source deactivate
```

Auto-activate environment

To make it easier to activate the necessary environment, one can check out *conda-auto-env* which activates the necessary environment automatically.

1.1.3 Download Dataset

In order to be able to run the scripts in this repository, one needs to first **download** the required datasets. One can do that by running the following command from the main directory and using the *Makefile*:

```
$: make download_dataset
```

This command will download the required catalogues for the analysis to the `data/external/` directory.

Depending on the variables used for the analysis, one can download different sets of catalogues, depending on what kind of catalogues they want to use it for.

Note: In order to make use of this commands, one will need wget. If *wget* is not available, one can download the files from http://lss.phy.vanderbilt.edu/groups/data_vc/DR7/sdss_catalogues/ and put them in `/data/external/SDSS`.

1.1.4 Steps and Commands

By running the following commands, one is able to replicate the results found in Calderon et al. (2018).

```
git clone https://github.com/vcalderon2009/SDSS_Conformity_Analysis.git
cd SDSS_Conformity_Analysis/
make environment
source activate conformity
python test_environment.py
make download_dataset
make 1_halo_fracs_calc
make 1_halo_mcf_calc
make 2_halo_fracs_calc
make 2_halo_mcf_calc
make plot_figures
open /reports/figures/SDSS/Paper_Figures/
```

This is the sequence of commands used to create the results shown in Calderon et al. (2018). The scripts already have default values. If one wishes to perform the analysis using a different set of parameters, these can be changed in the **Makefile**, or by simply calling the functions in the *Makefile* as:

```
make SAMPLE="20" download_dataset
```

This command will download the datasets for the `Mr20` galaxy and group galaxy catalogues.

In the *Steps and Commands* section, one can run the commands shown in order to reproduce the results in *Calderon et al. (2017)*.

1.2 Commands

This project analyzes **1-halo** and **2-halo** conformity on SDSS DR7 data.

After having downloaded the dataset by running the command .. code:

```
make download_dataset
```

you can start analyzing the dataset. This command will download the required catalogues for the analysis to `data/external/`.

1.2.1 1-halo

There are **2** types of analysis for the 1-halo conformity.These are

- 1-halo Quenched Fractions calculations
- 1-halo Marked correlation function (MCF)

One can run these two analyses by running the following commands:

```
make 1_halo_fracs_calc
make 1_halo_mcf_calc
```

1.2.2 2-halo

There are **2** types of analysis for the 1-halo conformity.These are

- 2-halo Central Quenched Fractions calculations
- 2-halo Marked Correlation Function (MCF)

One can run these two analyses by running the following commands:

```
make 2_halo_fracs_calc
make 2_halo_mcf_calc
```

Note: These functions make use of a fraction of your CPU, so it is better to run them **one by one**. One can modify the allowed fraction of the CPU in the Makefile by setting the `CPU_FRAC` variable to be from 0 to 1.

1.2.3 Making Plots

Once **all of the analyses** for 1-halo and 2-halo are *done*, i.e. after having run the **4 commands** above, one can plot all of the results by running the following command:

```
make plot_figures
```

This will produce the plots for **data** and **mocks** for all of the 4 different analyses for 1- and 2-halo conformity. The figures will be saved in: `/reports/figures/SDSS/Paper_Figures/`

Note: The scripts have default values that were used in Calderon et al. (2018). If one wishes to perform the analyses using a different set of parameters, these can be changed in **Makefiles**, or be given as input variables to the Makefile. Take into account that not all combinations of parameters are allowed for the analysis.

1.3 Methods

1.3.1 Marked Correlation Function

In the analysis of galactic conformity, we make use of the *marked correlation function* (MCF, see Skibba et al (2006) for more information).

The MCF has the format of

$$\mathcal{M}(r_p) = \frac{1 + W(r_p)}{1 + \xi(r_p)} \equiv \frac{WW}{DD}$$

where $\xi(r_p)$ is the usual two-point correlation function with pairs summed up in bins of projected separation, r_p, and $W(r_p)$ is the same except that galaxy pairs are weighted by the product of their marks. The estimator used in the equation can also be written as WW/DD, where DD is the raw number of galaxy pairs separated by r_p and WW is the weighted number of pairs. In the conformity analysis using MCF, we normalize by the mean of the galaxy property and then compute the MCF results.

1.3.2 Quenched Fractions

In this analysis, we also look at the **quenched fractions**. For the 1-halo analysis, we look at the *quenched fraction* of satellites around their host central galaxy, as a function of group mass.

For the 2-halo analysis, we look at the *quenched fraction* of centrals around other centrals, as function of their projected distance, r_p.

We perform this statistic on three galaxy properties, i.e. *specific star formation rate* (sSFR), *Sersic index* (n), and $(g-r)$ color.

1.4 Project Organization

```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for␣
↪ordering),
│                         the creator's initials, and a short `-` delimited␣
↪description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis␣
↪environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── environment.yml    <- The requirements file for reproducing the analysis␣
↪environment, e.g.
│                         the Anaconda environment used in this project.
│
├── test_environment.py   <- Script that checks that you are running the correct␣
↪python environment.
│
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   └── data           <- Scripts to download or generate data
│
```

(continues on next page)

(continued from previous page)

```
│       ├── make_dataset.py
│       │
│       ├── One_halo_conformity <- Scripts to analyze 1-halo conformity
│       │
│       ├── Two_halo_conformity <- Scripts to analyze 2-halo conformity
│       │
│       └── utilities_python    <- Scripts to analyze 1-halo conformity
│           └── pair_counter_rp <- Scripts used throughout both analyses.
│
└── tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```

Project based on the cookiecutter data science project template