Red$_{S}equence_{E}volution Documentation$

Release 0.0.dev0

Victor Calderon <victor.calderon@vanderbilt.edu>

Oct 11, 2019

CONTENTS

Demonstration of Red Sequence Evolution for Galaxy Clusters in Color-Magnitude diagrams

Author: Victor Calderon (victor.calderon@vanderbilt.edu) and Gourav Khullar (gkhullar@oddjob.uchicago.edu)

This documentation is part of the repository Red_Sequence_Evolution.

Part I

Contents

CHAPTER

ONE

GETTING STARTED

Demonstration of Red Sequence Evolution for Galaxy Clusters in Color-Magnitude diagrams

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

1.1 Downloading and Installing

Demonstration of Red Sequence Evolution for Galaxy Clusters in Color-Magnitude diagrams

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

1.1.1 Downloading repository

This documentation is part of the repository Red_Sequence_Evolution.

To download the repository to your computer, follow the following commands.

```
cd /path/to/where/you/want/to/download/repo
git clone https://github.com/vcalderon2009/Red_Sequence_Evolution.git
cd Red_Sequence_Evolution
```

The next step is to install and activate the project environment before being able to run any of the project's commands.

See *Using the Project's environment* for more information.

Project based on the modified version of cookiecutter data science project template

1.2 Using the Project's environment

Demonstration of Red Sequence Evolution for Galaxy Clusters in Color-Magnitude diagrams

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

1.2.1 Installing Environment & Dependencies

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

For reference, see: Manage Anaconda Environments

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

Show all available functions in the Makefile

You can use the *Makefile* for running common tasks like *updating environments*, *cleaning extra files*, and more.

To show all available functions in the Makefile, run:

```
make show-help

Available rules:

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
environment         Set up python interpreter environment - Using environment.yml
lint                Lint using flake8
remove_environment  Delete python interpreter environment
test_environment    Test python environment is setup correctly
update_environment  Update python interpreter environment
```

Create environment

In order to properly run the commands of this project, you should install the **necessary packages** before. For this, you will to have installed **Anaconda**, because otherwise you will not be able to use this command.

The name of the environment and its dependencies are explicitely shown in the `environment.yml` file. In order to create the environment, you must run:

```
make environment
```

The main file that lists all of the dependencies for the project can be found as `environment.yml`.

Activating the environment

Once the environment has been **installed**, you can now *activate* the environment by typing

```
source activate red_sequence_evolution
```

Note: Depending on your installation of Anaconda, you might have to use the command:

```
conda activate red_sequence_evolution
```

instead.

Updating environment

You can always update the project's environment. The package dependencies are handled by the `environment.yml` file, and sometimes these packages need to updaetd.

You can updated the project's environments by running:

```
make update_environment
```

This will update the versions of each of the necessary packages.

Deactivating environment

Once you are done running the scripts of this project, you should **deactivate** the environment. To do so, run:

```
source deactivate
```

Note: Depending on your installation of Anaconda, you might have to use the command:

```
conda deactivate
```

instead.

Auto-activate environment

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

See the link above for more information!

Project based on the modified version of cookiecutter data science project template

Project based on the modified version of cookiecutter data science project template

CHAPTER

TWO

COMMANDS

The Makefile contains the central entry points for common tasks related to this project.

This section is dedicated towards the functions used through the analysis.

Project based on the modified version of cookiecutter data science project template

CHAPTER

THREE

PROJECT STRUCTURE

The organization of the project is the following:

```
├── 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
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── 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`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── 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 Anaconda environment requirements file for reproducing the analysis␣
↪environment.
│                         This file is used by Anaconda to create the project environment.
│
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   │
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
```

(continues on next page)

(continued from previous page)

```
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```

Project based on the modified version of cookiecutter data science project template

Project based on the modified version of cookiecutter data science project template