User Guide

Installation of Dependencies

Firstly ensure python version 3.6 or later is installed. Download R_analyse and the requirements text file from https://github.com/arap2/Package-for-REVOLVER-analysis and set the directory you installed them into as your working directory. Run the following command in the terminal (linux) to batch install all the packages from the requirements file:

```
python3 -m pip install -r requirements.txt
```

Data Requirements

R_analyse accepts three inputs per full run of the pipeline, these are required for mapping ETs, training decision trees, and making predictions. It is not necessary to store the data in the same directory as the application but it is recommended to store the data in a logical locations (i.e. a parent or sub-directory of R_analyse). Note when running the application that a GUI will be generated for file selection, this will default to the current working directory. Consider this when arranging your files. All files should correspond to the format of the files detailed on the github page. In order.

Running R_analyse

Initialisation

Simply navigate to the directory containing R_analyse in the terminal and execute the following command:

```
python3 revolver_analysis.py
```

You will be prompted to decide how much of the pipeline you want to run. Choice 1 only requires transitional data to be input, choice 2 requires transitional data and alteration data, choice 3 requires the previous inputs as well as a separate dataset for stratification.

An output directory named analysis-output is automatically generated to store the outputs of this run. This name is incrementally numbered by +1 for each new run of the application as long as the file already exists in the working directory to prevent overwriting outputs.

Mapping ETs

Enter 1 when prompted and select your transitional data from the file browser that should be displayed. You will then be asked to enter a value for how many times an alteration must occur in your transitional dataset for a given cluster to map to the ET.

Note that high cut-off values may cause the application to error if the ETs are empty. When running pipeline 2 or 3, the sum of unique alterations between all ETs must be > 4 as this is currently the minimum number required for generating the decision trees. If you must train decision trees using fewer features this can be adjusted in the source code by changing the min_selection_count of the pick() function.

After entering an appropriate cut-off (for the data on our github we suggest a value of 3) a figure for each unique evolutionary cluster in your dataset will be output to the analysis_output folder.

Training Decision Trees

The recurrent features found across all the ETs are passed into the second section of the application for decision tree training. These are provided in a list in the terminal which you can choose from by pressing space on the alterations you want to include as candidates for node selection. Selected features are highlighted, press enter when you have selected all the features you want to use.

After selecting the features, the file browser will re-open and you can enter the alteration dataset which will be used to train the decision tree. The graphs will be output to the analysis_output folder.

Making Predictions

Once the decision tree has been graphed, you can use it to stratify independent datasets automatically. The file browser will open a final time where you should select the independent data you want to make cluster predictions on. These predictions will be output in a text file in the analysis_output directory.