GRAFree Manual

Aritra Mahapatra and Jayanta Mukherjee
Indian Institute of Technology, Kharagpur, India 721302

November 11, 2019

1 About GRAFree

GRAFree (GRaphical footprint based Alignment-Free method) is a python based program for deriving the phylogenetic tree from the genomic sequence data of different species based on the graphical representation of the sequences. The program consists of deriving phylogenetic tree from the set of genomic sequences, generating bootstrap sequences, and generate the bootstrap trees from the bootstrap sequences.

2 Required packages

The GRAFree is developed on python 2.7 and Linux based system. So this program is only compatible for python 2.7. The other prerequisites are following,

- Numpy
- Biopython v1.74
- Dendropy v4.4.0
- Matplotlib
- java-jdk 8.45.14

3 Create the environment

For executing the GRAFree we recommend to create a separate environment in anaconda. The image of the required environment (.yml) is given in this package.
First create the environment and activate it by executing the following commands,

```
conda env create -f myEnv.yml
conda activate grafreeEnv
```

We recommend to verify the installed package list in the `grafreeEnv` by executing `conda env list`.

4 Input directory

The input files should be kept in the following manner,

```
Dataset
 |__Species1
 |   |__filename
 |__Species2
     |__filename
```

The genomic sequences are stored as the fasta format in the input file. The filename should be same for every species.

5 The GRAFree options

GRAFree can only be executed through the command prompt. It is necessary to make all the python file executable before running the program (use `chmod +x *.py`). The command to execute the GRAFree from the command prompt is as `./GRAFree.py` with the command line arguments. The options are as follows,

`-d` DATASET_DIR. Location of the dataset. The dataset contains the directory named as the corresponding species name. Inside this directory the input sequence file (fasta format) having same name is placed.

`-i` INPUT_FILENAME. Name of the input file placed inside each species directory. It is to be mentioned that the input filename should be same for all the species but places inside separate directory.

`-o` OUTPUT_DIR. Mention the output directory.

`-l` BLOCK_SIZE. Mention the size of the block or window to compute the drift.

`-f` NO_OF_FRAGMENTS. Mention the number of fragments of the drift.

`-a` ASTRAL_PKG (OPTIONAL). As the combination of the trees is done by the ASTRAL, hence to overcome the version compatibility issue, we are providing the ASTRAL package with the GRAFRee. If this option is omitted then the program will execute by taking the default ASTRAL package.

`-x` exectimefile (OPTIONAL). The execution time for different modules are stored in this file.

`-w` WRITE (OPTIONAL). This is a boolean option. If this option is provided the the program will store the intermediate values, such as GFP, drift, distance matrix, etc and stored in the corresponding output directory.

`-p` PLOT (OPTIONAL). This is a boolean option. If this option is provided the the program will plot the GFPs and drift of each species and stored in the corresponding output directory.

`-s` SAVE_DATAFORMAT (OPTIONAL). This option will be activated once the `-p` is on. Default the plots are stored as the pdf file. But they can be stored as jpg, png, etc. format too by this option.

`--redo` REDO (OPTIONAL). This is a boolean option. If this option is provided the the program will regenerate trees by ignoring the previous intermediate outputs.

6 The bootstrap options

The bootstrap of the sequence is done by considering the probability of the variation of the sequence. This program impose the insertion, deletion, and mutation within the target sequence at a particular rate of variation. This code can be executed by `./Proposed_Bootstrapping.py` with some options. Once the code is executed it will generate a single copy of the bootstrap sequence for the whole dataset. Hence, for generating `N` number of bootstrap copy, this code should be executed for the `N` number of times. The options are as follows,

`-d` DATASET_DIR. Location of the target dataset directory.

`-o` BS_DATASET_DIR. Location of the generated bootstrapped sequences.

`-i` INPUT_FILENAME. Name of the target sequence file (fasta format). The name of the file should be same for all the species.

`-v` variation_probability. Mention the probability of the variation of the sequence. This option takes the floating type value range from 0 to 1.

7 Genome composition

This additional feature makes a list of the composition of the genome sequences considered in the study. This program derives the length of the sequences, fraction of occurrence of A, T, G, C, AT-skew, GC-skew, etc. This code can be executed by `./Genome_Composition.py` with some options. The options are as follows,

`-d` DATASET_DIR. Location of the target dataset directory.

`-i` INPUT_FILENAME. Name of the target sequence file (fasta format). The name of the file should be same for all the species.

`-o` OUTPUT_DIR. Mention the output directory.

A sample command file is provided with this package. The shell script file name `SampleCode.sh`, contains the sample code and can be executed by changing the variables accordingly.