

Easy353 —— Short Manual

zzhen

zzhen0302@163.com

May 22,2022

Easy353 —— Short Manual

- 1. About Easy353
- 2. Easy353 and Graphical User Interface
- 3. Downloading Easy353
- 4. Installing Easy353
 - Get Easy353-GUI
 - Easy353-cmd
 - Option 1. Using the setup.py
 - Option 2. In situ configuration
- 5. Input File Format
 - 5.1 Reads Files
 - 5.2 Reference files
- 6. Parameters
 - 6.1 Parameters to be specified
 - 6.2 Parameters with default value
- 7. Output
- 8. Example

1. About Easy353

Easy353 is a tool for recovering Angiosperms353 gene set(AGS), which can filter and de novo assemble reads from sequencing data based on data from <https://treeoflife.kew.org/>, helping users capture AGS accurately and effectively.

When using Easy353 please cite:

XXXXXX

2. Easy353 and Graphical User Interface

Easy353 is an easy-to-use pure Python program and is designed to be compatible with versions higher than 3.6.

Easy353 has a command line version and a GUI version; users on Windows, macOS, and Linux computers could run it with cmd, and we also provide a user-friendly graphical interface for Windows and macOS users.

3. Downloading Easy353

Easy353 is open source under **MIT**. It is distributed via plant720 github repository: <https://github.com/plant720/Easy353> where you can always download the most up to date version. Make sure to watch the github repository to remain up to date regarding code changes.

Version numbers follow the notation **x.y.z** where **x** changes with major code reorganizations, **y** changes when new features are added and **z** changes with bug fixes.

4. Installing Easy353

Easy353 comes in two different flavors: Easy353-cmd and Easy353-GUI. Thus, before installation you need to know on what kind of system you intend to execute Easy353. The Easy353-cmd can be run at popular systems (Linux, macOS and Windows); while the easy353-GUI is designed for Windows and macOS.

When working with small amounts of data on a Windows or Mac computer, we advise using the Easy353-GUI; larger amounts of data should be processed using the Easy353-cmd.

Get Easy353-GUI

You can download Easy353-GUI from <https://github.com/plant720/Easy353/release> and Easy353-GUI can be run directly by double-clicking.

Easy353-cmd

There are several generally 2 ways to install Easy353-cmd:

- Option 1 **Using the setup.py**
- Option 2 **In situ configuration**

Option 1. Using the setup.py

You should use git to download the entire Easy353 repository and install the Easy353 using the setup.py.

```
1 # get a local copy of easy353 source code
2 git clone https://github.com/plant720/Easy353.git
3 # install the code
4 cd Easy353
5 python setup.py install --user
```

Using the setup.py, **you should have Python library setuptools installed** (`sudo apt install -y python-setuptools` or `sudo yum install -y python-setuptools` or `pip install setuptools`).

For some Linux nad macOS systems, after above commands you still cannot execute `build_database.py` and `easy353.py` in a new terminal directly, meaning `~/.local/bin` was not added to the \$PATH, you have to manually add `~/.local/bin` :

```
1 # add ~/.local/bin to PATH
2 echo "export PATH=~/.local/bin:\$PATH" >> ~/.bashrc
3 source ~/.bashrc
```

Option 2. In situ configuration

Use git to download the entire Easy353 repository.

```
1 # Supposing you are going to install it at ~/Applications
2 mkdir ~/Applications # create directories if not existed
3 cd ~/Applications
4 git clone https://github.com/plant720/Easy353.git
```

Use following commands to make Easy353 scripts executable

```
1 chmod 755 Easy353/build_database.py
2 chmod 755 Easy353/easy353.py
```

Add Easy353 to the \$PATH.

```
1 echo "export PATH=~/.Applications/Easy353:\$PATH" >> ~/.bashrc
2 source ~/.bashrc
```

At last, install python libraries biopython,psutil, requests, and beautifulsoup4 using pip or conda.

```
1 # install required libs
2 pip install biopython psutil requests beautifulsoup4
```

5. Input File Format

5.1 Reads Files

Easy353 assembles any High-throughput Sequencing(HTS) reads, given in the **FASTQ** format. Paired or unpaired reads are OK. And Easy353 can directly read files compressed with gzip. Compressed files should end with `.gz`.

[illegible]

5.2 Reference files

DNA sequences are stored in reference files in **FASTA** format. Each reference file contains one or more DNA sequence, which is the homologous gene from closely related species. Here we provide an example: <https://github.com/plant720/Easy353Test/tree/master/data>

Each sequence's ID should contain the gene name and species name, separated by the `_`.

```
1 >Litchi_chinensis_6946
2 CAAACCAAAATACACAATATAGGGGCAACACTTGTGGGGTTGATAAATTTGGTAACAAGTATTATGAGAAA
  CTTGGAAGACATAGGTGGGTGAATATGCAGAGAAAAG
3 >Tristiropsis_sp._6946
4 GGAAGACATAGGTGGGTGAATATGCAGAGAAAGGTCGCTACAATGCCTCTCAGGTGCCTCCGGAATGGCAT
  GGTGGCTTCACTTTATAACTGATCACACAGGAGATG
```

6. Parameters

One of the most important command is the Easy353 help option that displays all options. You can get on-line help type: `easy353.py -h` ,

and you will get the following, very long listing, that will be discussed at length below:

```
1  usage: easy353.py [options]
2
3  Easy353 zzhen@sculab
4
5  optional arguments:
6  -h, --help            show this help message and exit
7  -1 FQ_FILE_1 [FQ_FILE_1 ...]
8                        Input file(s) with forward paired-end reads
9                        (*.fq/.gz/.tar.gz).
10 -2 FQ_FILE_2 [FQ_FILE_2 ...]
11                       Input file(s) with reverse paired-end reads
12                       (*.fq/.gz/.tar.gz).
13 -u UNPAIRED_FQ_FILE [UNPAIRED_FQ_FILE ...]
14                       Input file(s) with unpaired (single-end) reads.
15 -r REFERENCE           Input a file(directory) with references.
16 -o OUTPUT_DIR         Output directory.
17 -k1 FILTER_KMER       Kmer setting for filtering reads. Default:31
18 -k2 ASSEMBLE_KMER     Kmer setting for assembling reads. Default:41
19 -s STEP_LENGTH        Step length of the sliding window on the reads.
20                       Default:1
21 -t1 FILTER_THREAD     Threads setting for filtering reads. Default:4
22 -t2 ASSEMBLE_THREAD   Threads setting for assembling reads. Default:4
23 -kmer_limit KMER_LIMIT
24                       Limit of kmer count. Default:2
25 -f FUNCTION_MODE      0:all,1:filter,2:assemble. Default:0
26 -min MINIMUM_LENGTH_RATIO
27                       The minimum ratio of contig length to reference
28                       average length. Default:1.0
29 -max MAXIMUM_LENGTH_RATIO
30                       The maximum ratio of contig length to reference
31                       average length. Default:2.0
32 -change_seed CHANGE_SEED
33                       Times of changing seed. Default:32
34 -reference_number REFERENCE_NUMBER
35                       The number of the reference sequences used to
36 build
37                       hash table. Default:all
38 -fast                 Whether to use fast mode.
```

6.1 Parameters to be specified

The following options need to be specified:

- `fq_file_1` and `fq_file_2` : The input files with paired-end reads, given in FASTQ format.
- `unpaired_fq_file` : The input file with unpaired-end reads.
- `reference` : A directory recording 353 gene sequences from closely related species, given in FASTA format.
- `output_dir` : The output directory.

6.2 Parameters with default value

The following options have default values and do not need to be specified in general.

- `filter_kmer` : K-mer length setting for filtering, default value is 31. The `filter_kmer` is the length of the shared DNA segment between the reads and the reference sequences. Easy353 looks for a shared K-length DNA segment (K-mer) between a read and a reference to assess whether they are related. Consequently, when the value of `filter_kmer` is set to 31, it means to search for reads that have 31bp in common with the reference sequence.
- `assemble_kmer` : K-mer length setting for assembly, default value is 41. The `assemble_kmer` is the length of the nodes in the de Bruijn graph (DBG). In reads assembly, the filtered reads are divided into K-mers; the k-mers are employed as the nodes of the DBG. It strongly depends on the input dataset. For Illumina reads(150bp) with sufficient coverage (> 40x), we have good results with $k = 41$.
- `kmer_limit` : Limit of K-mer count, default value is 2. The `kmer_limit` is used to remove erroneous, low-abundance K-mers. This parameter also strongly depends on the dataset. It corresponds to the smallest amount of times a correct k-mer appears in the reads. A typical value is 2, which means only K-mers with at least 3 occurrences will be used for assembly. If the dataset has high coverage, try larger values.
- `filter_thread` : Threads setting for reads filtering, default value is 1. This value is advised to ideally not exceed 4. And for Windows and macOS, the `filter_thread` should be set to 1.
- `assemble_thread` : Threads setting for reads assembly, default value is 4.
- `reference_number` : The number of the reference sequences used to build hash table, and the full reference sequence will be used by default. When one target gene has more than 100 homologous reference sequences, we advise setting this `reference_number` to 100 since the amount of sequences in the fasta file will effect the memory size needed by Easy353.
- `fast` : The switch to control fast mode. If typing `-fast` in command line, Easy353 will run faster, but will use nearly twice as much memory.
- `step_length` : The length of the interval when splitting the reads into K-mer, default value is 1. With a sequence is AGTTACGTCA, when `step_length` is 1 and `kmer_size` is 5, we can get AGTTA, GTTAC, TTACG, TACGT, ACGTC and CGTCA; when `step_length` is 2, we will get AGTTA, TTACG, ACGTC. This parameter can be used to reduce the program runtime when the dataset is large with sufficient coverage.

- `change_seed` : The setting for the number of seed changes, default value is 32. Actually, `change_seed` is the amount of times the assembly's beginning point can be changed. The seeds are high-abundance K-mers selected from filtered reads that serve as the beginning point for de novo assembly. When the assembled gene's length is less than the set value, Easy353 will alter the assembly beginning point.
- `minimum_length_ratio` : The minimum ratio of recovered gene's length to reference gene's average length, default value is 1.0. Easy353 will decide that the gene recovery has failed when the actual length ratio is less than this value.
- `maximum_length_ratio` : The maximum ratio of recovered gene's length to reference gene's average length, default value is 2.0. When the actual assembled gene's length is longer, Easy353 will use stronger limitations, such as higher `assemble_kmer` and `kmer_limit`.

7. Output

The output directory contains the `filtered_reads` and the `target_genes` .

- The directory, `filtered_reads` , contains filtered reads associated with target 353 genes.
- And the directory, `target_genes` , is the most important result directory of Easy353, which stores 353 gene sequences recovered by Easy353.
 - The files under `target_genes` are recovered successfully, and the `unrecovered_genes` directory contains 353 genes that were not recovered successfully, because the recovered genes are shorter in length.

8. Example

- Download the simulation data of *Glycine max*:

```
1 wget
  https://github.com/plant720/Easy353Test/raw/master/data/Gmax_sim_1.fastq
  .gz
2 wget
  https://github.com/plant720/Easy353Test/raw/master/data/Gmax_sim_2.fastq
  .gz
```

- After installation of Easy353 and downloading the simulation data of *Glycine max*, please download the AGS of related species as the reference

```

1 # Download the AGS data according to taxonomy: Glycine max is a species
  from Glycine genus in Fabaceae, so we download species from Fabaceae as
  the reference.
2 # The reference sequences are downloaded from
  https://treeoflife.kew.org/,so keep your devices connected to the
  network. And if you the build_database.py, please cite:
  https://doi.org/10.1093/sysbio/syab035.
3 build_database.py -o 353_ref_Fabaceae -c Fabaceae -t 10 -exclude
  Glycine_max -generate
4 # The final reference sequences can be found at
  353_ref_Fabaceae/353gene after downloading
5
6 ## Explanation of parameters
7 -o: the output directroy
8 -c: the taxonomy of species that used as reference
9 -t: the thread used to download files
10 -exclude: exclud species that are not used as reference
11 -generate: generate a csv that records the info of downloaded species

```

- Then do the recovery of Angiosperms353 gene set(AGS)

```

1 # use easy353.py to filter and assemble reads to get target genes
2 easy353.py -1 Gmax_sim_1.fastq.gz -2 Gmax_sim_2.fastq.gz -r
  353_ref_Fabaceae/353gene -o test_package -k1 31 -k2 41 -t1 1 -t2 4 -
  reference_number 100

```

- Now, you can view the result of Easy353 in output directory