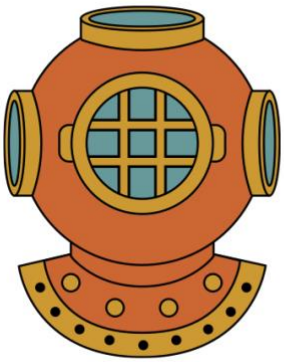


EasyDIVER

Easy pre-processing and **D**ereplication of **I**n **V**itro **E**volution **R**eads

User's guide
Version 2.0
(May 2020)

Celia Blanco[1,2], Samuel Verbanic[2,3], Burckhard Seelig[4,5] and Irene A. Chen[1,2,3]

1. Department of Chemistry and Biochemistry 9510, University of California, Santa Barbara, CA 93106
2. Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095
3. Program in Biomolecular Sciences and Engineering, University of California, Santa Barbara, CA 93106
4. Department of Biochemistry, Molecular Biology and Biophysics, University of Minnesota, Minneapolis, MN, 55455, USA
5. BioTechnology Institute, University of Minnesota, St. Paul, MN, 55108, USA

If you use Easydiver, please cite the paper:

Celia Blanco*, Samuel Verbanic*, Burckhard Seelig and Irene A. Chen. EasyDIVER: a pipeline for assembling and counting high throughput sequencing data from in vitro evolution of nucleic acids or peptides. *Under review*.

For feedback, suggestions, technical support, etc., please email us at celiablanco@ucla.edu or verbanic@ucla.edu.

When reporting bugs, please include the full output (error messages included) printed in the terminal when running the pipeline.

DESCRIPTION

EasyDIVER converts raw, paired-end, demultiplexed Illumina read files into processed, dereplicated data ready for analysis. The algorithm performs the following:

1. Joins the raw data using PANDAseq.
2. Extracts the insert sequence based on user-supplied primer sequences.
3. Optionally translates into amino acids.
4. Generates counts files.
5. Collects sequence length distributions (histos).
6. Creates a log file.

SYNOPSIS

```
easydiver.sh -i input.directory [-o output.directory] [-p
forward.primer] [-q reverse.primer] [-T threads] [-a] [-r] [-e] [-h]
```

Alternatively, a more user-friendly (but less versatile) solution can be optionally used. If no flags are provided the user will be prompted for input values in the command line in verbose form. To use the prompted input version, run:

```
easydiver.sh
```

OPTIONS

-i input.directory

Required. Input directory path and name. If no value is provided, an error message will be printed in the terminal: `ERROR: No input filepath supplied` and no further action will be performed.

-o output.directory

Optional. Output directory path and name. If no value is provided, the default value `/pipeline.output` will be used.

-p forward.primer

Optional. Extraction forward DNA primer. If a forward primer sequence is provided, the pipeline strips out the primer from the start of the sequence. Any sequence before the provided primer will be discarded.

-q reverse.primer

Optional. Extraction reverse DNA primer. If a reverse primer sequence is provided, the pipeline strips out the primer at the start of the sequence. Any sequence after the provided primer will be discarded.

-T threads

Optional. Number of threads used for computation. Default value is 1.
The number of threads that may be used is dependent on the user's CPU (for example, if using a machine with 16 threads, 14 could be a desirable number). The default value of 1 would be suboptimal for multi-core machines.

-a

Optional. Translation into amino acids is performed. DNA sequences are translated using the standard genetic code, and the resulting sequences are dereplicated. By default, translation is not performed.

-r

Optional. Files for individual lanes are retained. By default, the script will suppress outputs from individual lanes.

-e

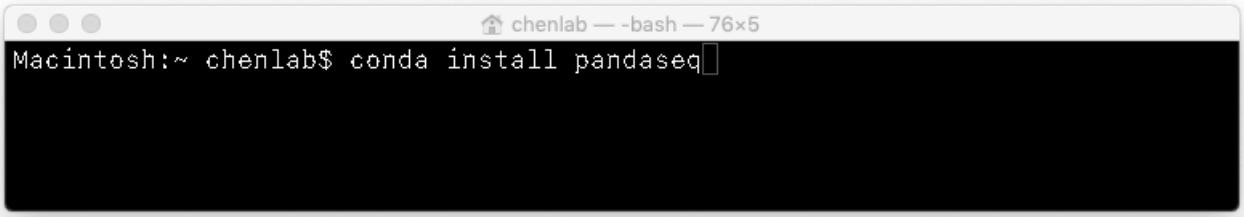
Optional. Additional internal PANDAseq flags. Values must be entered in quotation marks (e.g. `-e "-L 50"`). Default value is `"-l 1 -d rbfkms"`. For more information see the PANDAseq manual.

-h

If used, a help message will be printed in the terminal and no further action will be performed.

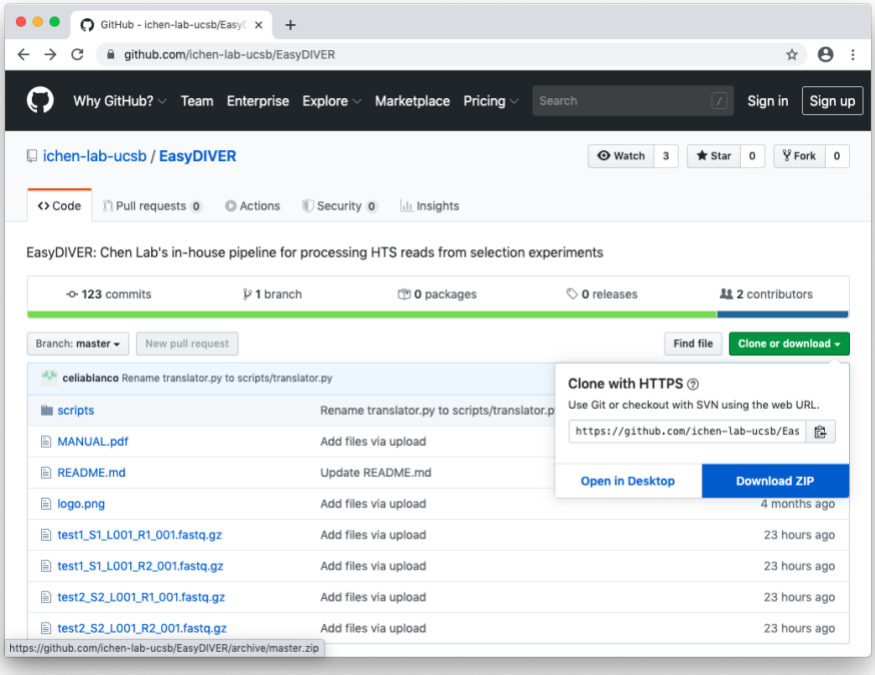
DEPENDENCIES

Before you can use EasyDIVER you need to install Python and PANDASeq. To install Python follow the instructions here. To install PANDAseq follow the instructions here or keep reading. We recommend using the Anaconda distribution of python, and adding the Bioconda channel to Anaconda's package manager, conda. See the Anaconda documentation for installation. After installing Anaconda with Bioconda, PANDASeq is easily installed using conda as conda install pandaseq:

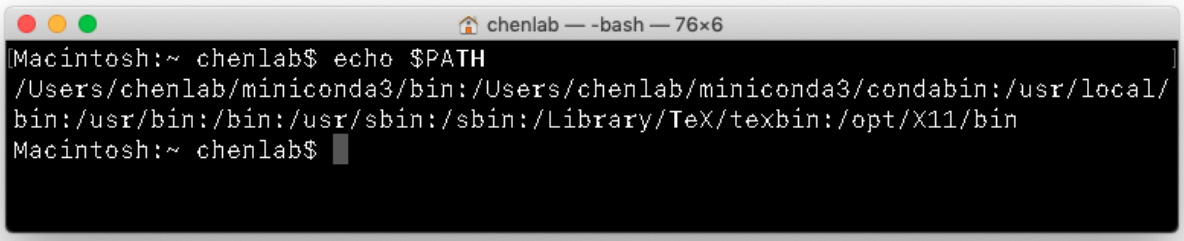


DOWNLOAD AND SET UP

To start using EasyDIVER, download the entire package from GitHub (click on the green button Clone or download, then click on Download ZIP). Once it's downloaded, go to the Downloads folder and unzip the package (this can usually be done by double clicking on it).

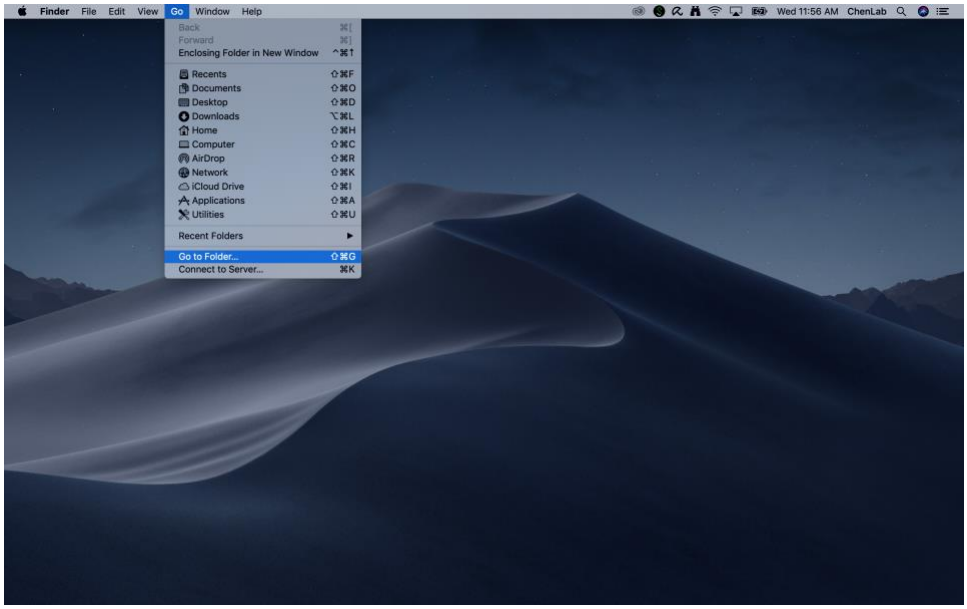


PATH is an environment variable on most operating systems that specifies where executable programs are located. An executable file located in a directory on the PATH variable can be called from anywhere in the system. To find these directories, you can enter echo $PATH in a terminal emulator (e.g. the Terminal app on Mac machines):

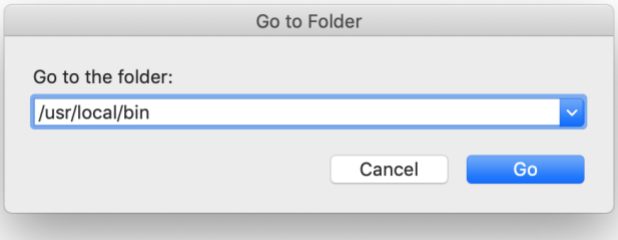


All the directories in the PATH variable are listed, separated by a colon. Placing the EasyDIVER scripts (easydiver.sh and translator.py) in one of these directories ensures that the system can find them from wherever you are calling them (even if you don't specify their location when calling them).

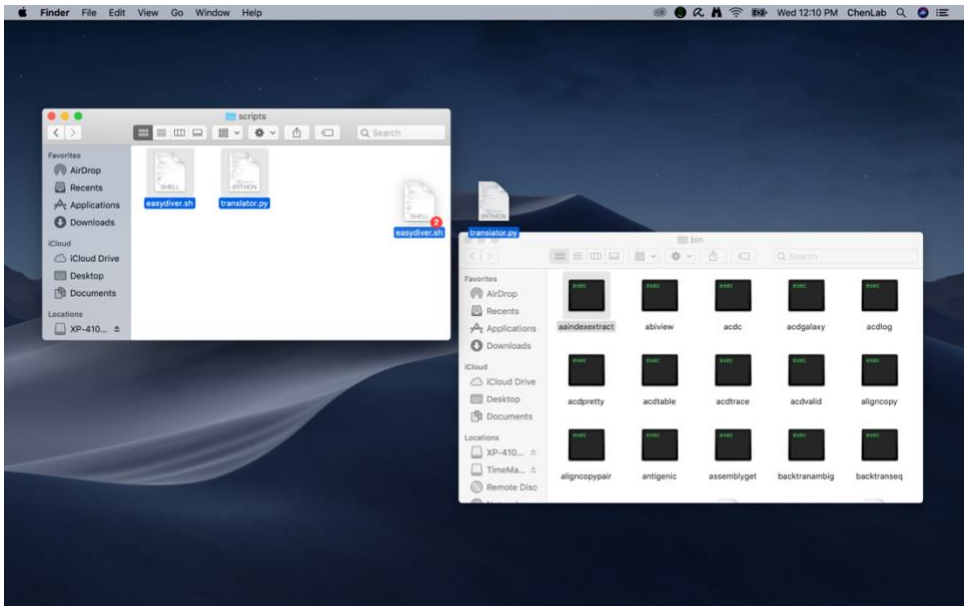
Here, we will place the EasyDIVER scripts under */usr/local/bin.* This can be done using the command line. For the sake of simplicity, here we show how to move them using the graphical user interface on a Mac. To open the */usr/local/bin* folder, we can click on Go in the menu bar and then click on "Go to folder":



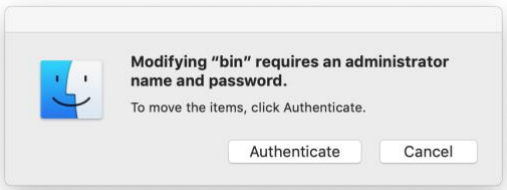
Enter the name of the directory in the PATH variable. In this case, */usr/local/bin:*

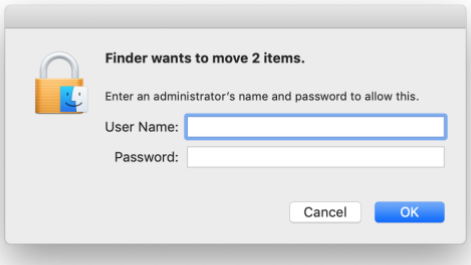


Move easydiver.sh and translator.py from *Downloads/EasyDIVER-master/scripts* to */usr/local/bin* by dragging them from one folder to the other:

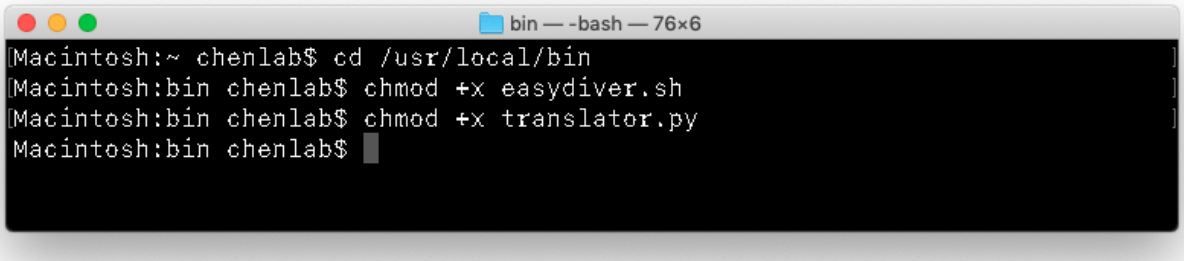


If a window pops-up asking for administrator credential, click on Authenticate and provide User name and Password:





To make the scripts executable, we first need to go to the directory where they are placed, in this case */usr/local/bin* with the command cd /usr/local/bin. To make the easydiver script executable enter chmod +x easydiver.sh. To make the translator script executable enter chmod +x translator.py:



EXAMPLE

To test the pipeline and troubleshoot potential issues, we provide two samples of test data in the GitHub repository. An example of command to run the pipeline using the test data provided in the GitHub repository:

```
easydiver.sh -i ./ -o ./output -p GGCGGAAAGCACATCTGC -T 14 -a -r
```

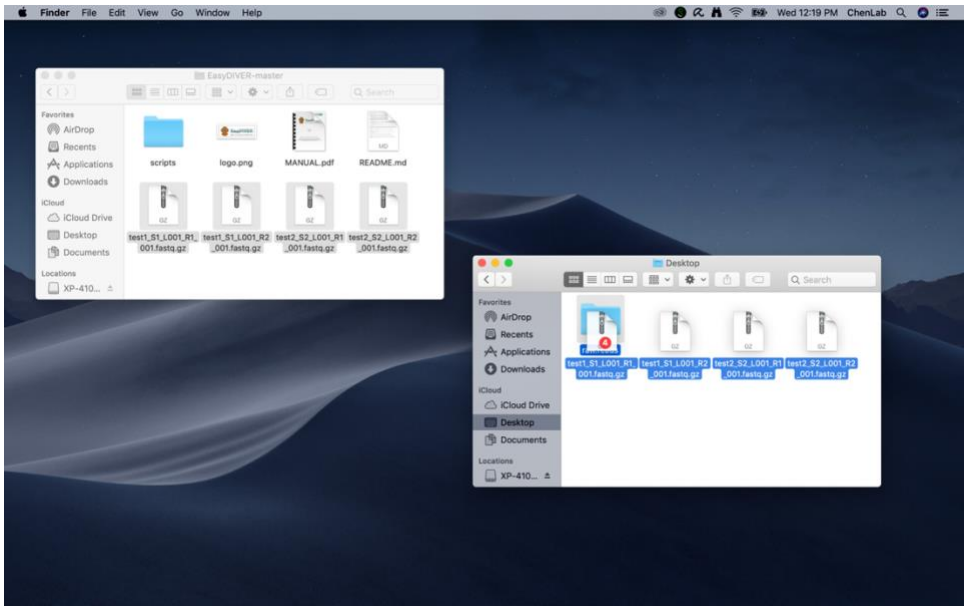
Alternatively, the prompted input version can be requested by not using any flags:

```
easydiver.sh
```

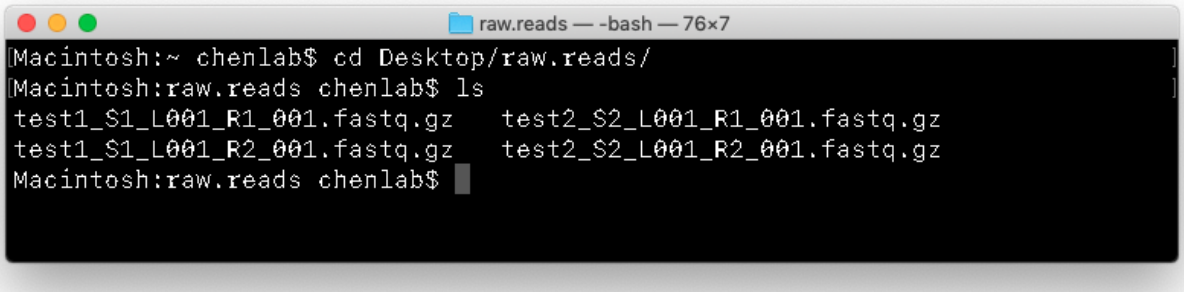
All input files must be:

1. Located in the same directory (even reads from separate lanes).
2. In FASTQ format
3. Named using the standard Illumina naming scheme: `sample-name_S#_L00#_R#_001.fastq`
4. In either .fastq or .fastq.gz extensions.

Create a directory to locate the test files. For example, here we create the folder *raw.reads* in the Desktop. Then move the test files to that folder by dragging them from one folder to the other.

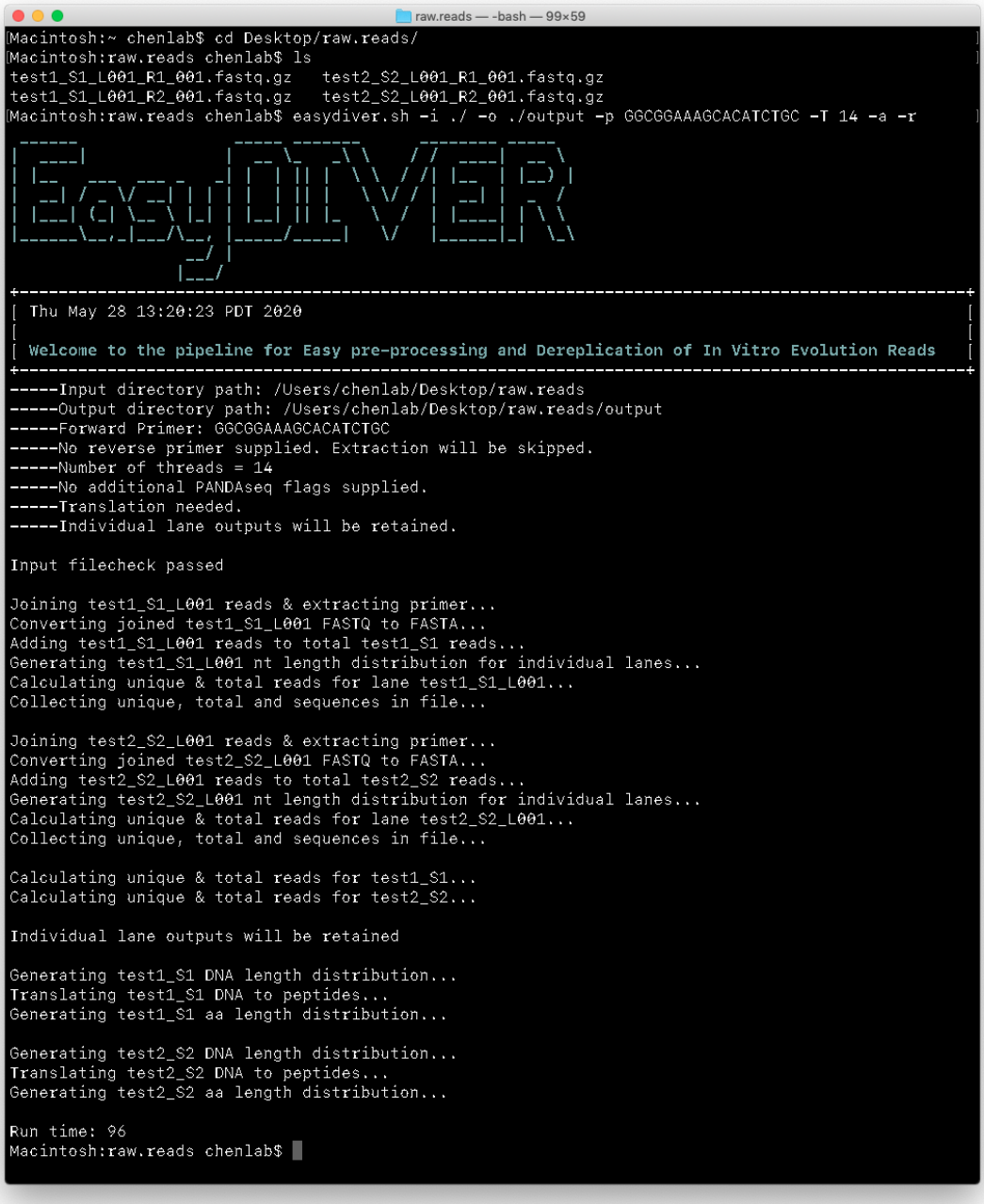


To run EasyDIVER, go to the location where the data is stored (*Desktop/raw.reads* in this case). To do this, enter cd Desktop/raw.reads in the command line. By typing ls you can see the contents of the directory. Make sure your test files are in this location.



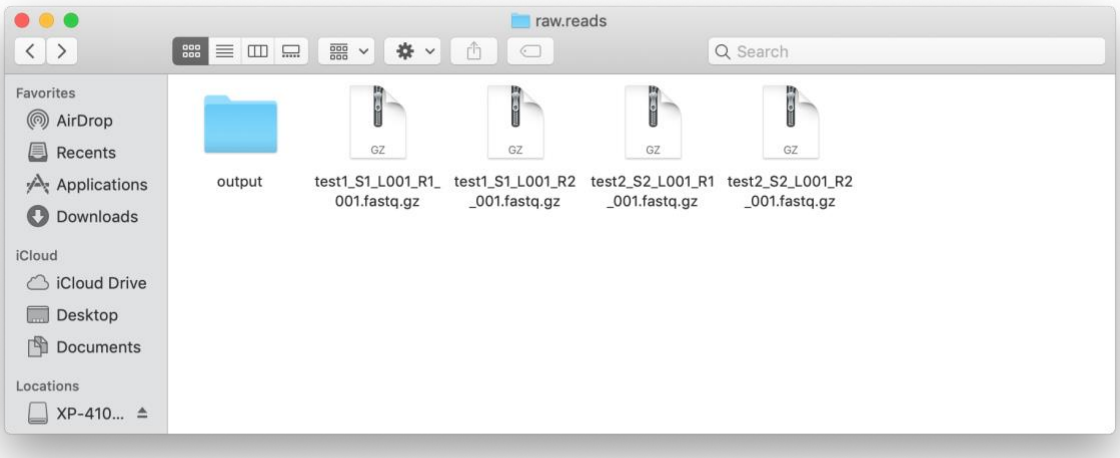
You are all set to run EasyDIVER!

Type easydiver.sh -i ./ -o ./output -p GGCGGAAAGCACATCTGC -T 14 -a -r to run the example above. For a detailed explanation of the choice of parameters, check the section TIPS at the end of this manual.

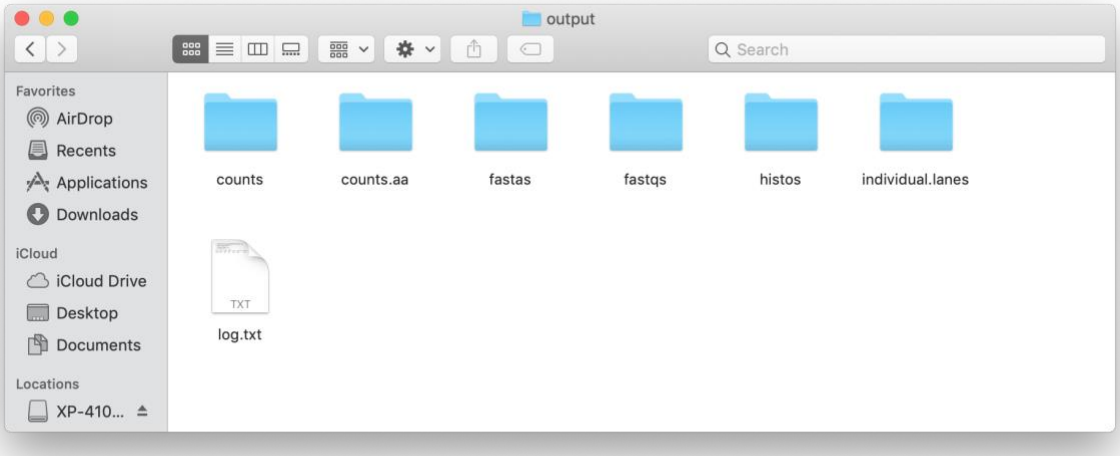


If the prompted input version is requested (no flags provided), the terminal output will look like this instead (note that the rest of the output has been omitted for the sake of simplicity, as it is the same as for the non-prompted input version):

Once it finished, you should have a new subfolder in the *raw.reads* folder (in this case it should be called *output*, because that's the name we gave it in the command line):



The subfolder output contains all the files generated by Easydiver:



For each sample, the pipeline combines the reads from every lane, and redirects the outputs to the following sub-directories:

fastqs contains the joined fastq files
fastas contains the joined fasta files
counts contains DNA counts files for every sample
counts.aa contains peptide count files for every sample (if translation is requested using the flag `-a`)
histos contains the DNA sequence length distributions and the peptide sequence length distribution (if translation is required)
individual.lanes contains the files (joined fasta files joined fastq files, text counts files and text histograms) corresponding to the individual lanes (if requested using the flag `-r`)

If translation to amino acids is desired (indicated by the use of the flag `-a`) the counts files are translated using the standard genetic code, and the resulting sequences are dereplicated. Count files and length distributions are created for the amino acid sequences as well. All sequence length distributions are redirected to the directory **histos**.

By default, the script will suppress outputs from individual lanes. If you wish to retain the individual lane outputs, use the `-r` flag. If the flag `-r` is used, files corresponding to the individual lanes (joined fasta files joined fastq files, text counts files and text histograms) are retained and redirected to the subdirectory called **individual.lanes**.

For the record, and to monitor the success of the run, a single log text file with the parameters used and the number of sequences in the fastq and counts files is created at the end of the process.

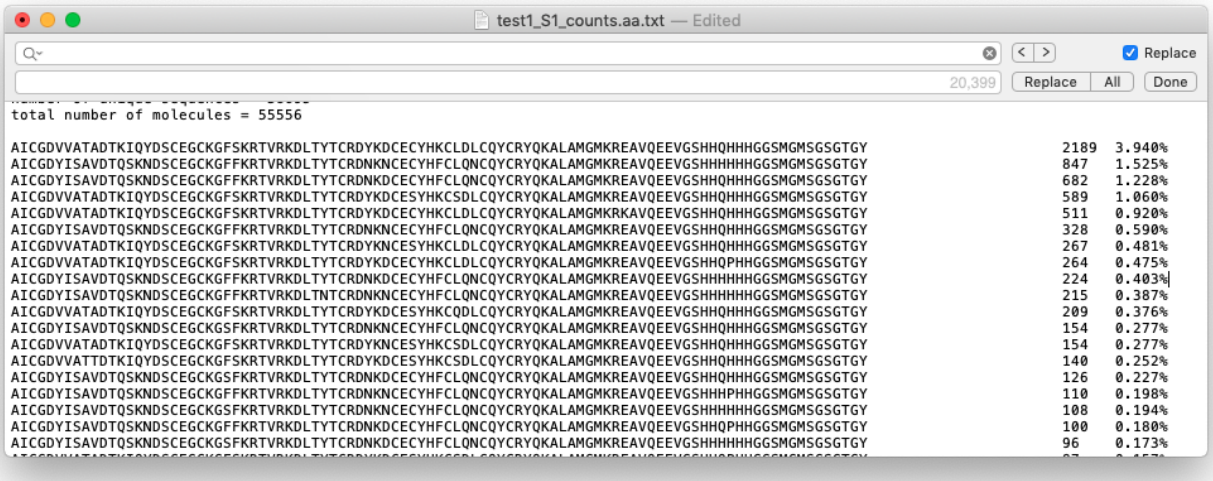
The outcoming file log.txt, for the same example parameters:

log.txt

```
-----Input directory path: /Users/cblanco/Desktop/test_pipeline/raw.reads.midBS.10
-----Output directory path: /Users/cblanco/Desktop/test_pipeline/raw.reads.midBS.10/output_CB
-----Forward Primer: GGCGGAAAGCACATCTGC
-----No reverse primer supplied.
-----Number of threads = 14
-----No additional PANDAseq flags.
-----Translation needed.
-----Individual lane outputs retained.

sample    fastq_R1  fastq_R2  unique_nt  total_nt  recovered_nt(%)  unique_aa  total_aa  recovered_aa(%)
test1_S1  64516     64516     54168      55576     86.14%           38695      55556     86.11%
test2_S2  53541     53541     45131      46605     87.05%           31147      46593     87.02%
```

The first lines of the outcoming peptide count file for sample sub1_S1 (sub1_S1_counts.aa.txt) will look as follow:



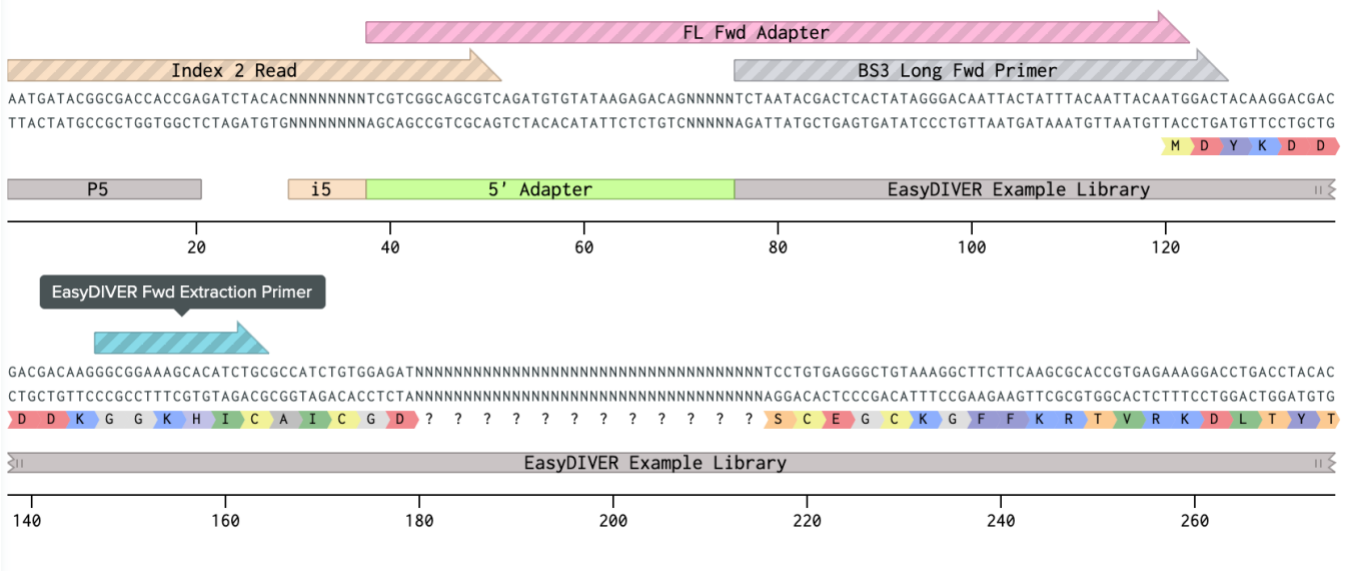
For more detailed information on EasyDIVER, please see our publication, or the README file available in the GitHub repository.

TIPS

To successfully process your data with EasyDIVER, the correct parameters must be selected. Here we will provide guidance on how to select parameters for your data.

- **Primer selection:** Your choice of primers will determine how your target sequences are extracted and set the reading frame for translation. You should select primers that will target conserved sequences in your library and place your extracted sequences in the desired reading frame.

    - **Example:** In the example command provided in this manual, the forward extraction primer sequence is entered as `-p GGCGGAAAGCACATCTGC`. Using the schematic below, we can see why this sequence was selected. It is in a conserved portion of the library, and should be present in every sequence. The extracted sequence will be in our desired reading frame, starting with the amino acids `AICGD`, followed by the random portion of the library.



- **Threads:** Modern processors possess the capability to run processes in parallel by using multiple 'threads'. Certain processes in EasyDIVER (such as joining with PANDAseq) can utilize this capability to run faster. As a rule of thumb, you may set the number of threads equal to the number of cores on your machine for optimal performance. For further optimization, look up the specifications of your hardware and adjust your thread count accordingly.

    - **Example:** In the example command provided in this manual, the thread count is set as `-T 14`, which would be a desirable choice for a machine with 16 threads. To be safe, and assuming your machine is at least a quad-core CPU, use `-T 4`.