EasyDIVER

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

User's guide
Version 1.0

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

EasyDIVER

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. Submitted.

For feedback, suggestions, technical support, etc., please email us at blanco@ucsb.edu or sverbanic@ucsb.edu.

When reporting bugs, please include the full output (eror 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 -i input.directory [-o output.directory] [-p
forward.primer] [-q reverse.primer] [-a] [-r] [-T threads] [-e] [-h]
```

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.

-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.

-T threads

Optional. Number of threads used for computation. Default value is 1.

-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

-h

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

INSTALLATION

To use the pipeline, first install the required dependencies.

The pipeline and the translator must be placed in /usr/local/bin/ upon download.

EasyDIVER can be used with or without installation. To install EasyDIVER, execute from the local directory where it's stored (the first command makes it executable, the second comman installs EasyDIVER):

```
chmod +x easydiver.sh
```

```
sudo install easydiver.sh
```

If EasyDIVER is not installed, then the command bash and the full script name (easydiver.sh) must be used to run the pipeline (e.g. `bash easydiver.sh -i [-o -p -q -h -a -r -T -e]`).

EXAMPLE

An example of command to run the pipeline using the test data provided in the GitHub repository:

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

The above command works if it is:

1. Run from the same directory where the raw data is located.
2. The script easydiver.sh has been moved to /usr/local/bin, made executable and installed. For information on how to install EasyDIVER, see the README file available in the GitHub repository.

INPUT

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.

OUTPUT

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 contaisn 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 derided (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.