hundo Documentation

Release 1.2

Joe Brown

Aug 08, 2019

Contents

CHAPTER 1

Installation

This protocol leverages the work of Bioconda and depends on `conda`. For complete setup of these, please see:

https://bioconda.github.io/#using-bioconda

Really, you just need to make sure `conda` is executable and you've set up your channels (steps 1 and 2). Then:

```
conda install python>=3.6 click \
    pyyaml snakemake>=5.1.4 biopython
pip install hundo
```

To update to the newest version of Hundo, run

```
pip install --upgrade hundo
```

Alternatively, if you do not want any new executables in your environment you can install into a new conda environment, e.g. hundo_env:

```
conda create --name hundo_env python=3 click pyyaml snakemake biopython
conda activate hundo_env
pip install hundo
```

To leave the environment:

```
conda deactivate
```

CHAPTER 2

Usage

Running samples through annotation requires that input FASTQs be paired-end, named in a semi-conventional style starting sample ID, contain “_R1” (or “_r1”) and “_R2” (or “_r2”) index identifiers, and have an extension “.fastq” or “.fq”. The files may be gzipped and end with “.gz”. By default, both R1 and R2 need to be larger than 10K in size, which corresponds to around 100 reads in a compressed fastq file. This cutoff is arbitrary and can be set using `--prefilter-file-size`.

Using the example data of the mothur SOP located in our tests directory, we can annotate across SILVA using:

```
cd example
hundo annotate \
    --filter-adapters qc_references/adapters.fa.gz \
    --filter-contaminants qc_references/phix174.fa.gz \
    --out-dir mothur_sop_silva \
    --database-dir annotation_references \
    --reference-database silva \
    mothur_sop_data
```

The data directory can optionally be a pattern containing a wildcard, such as:

```
hundo annotate \
    --filter-adapters qc_references/adapters.fa.gz \
    --filter-contaminants qc_references/phix174.fa.gz \
    --out-dir mothur_sop_silva \
    --database-dir annotation_references \
    --reference-database silva \
    'mothur_sop_data/F3D14*S20*.fastq.gz'
```

The string must be contained between single quotes so it isn’t expanded into a space delimited list.

Or when data is spread across multiple directories, you can use a combination of paths and patterns in a comma separated list, like:

```
hundo annotate \
    --filter-adapters qc_references/adapters.fa.gz \
```

(continues on next page)

(continued from previous page)

```
    --filter-contaminants qc_references/phix174.fa.gz \
    --out-dir mothur_sop_silva \
    --database-dir annotation_references \
    --reference-database silva \
    'collection1/LM_*.fastq.gz,collection2/rawdata'
```

Or if you have a case where you have lots of data directories, you can specify `--input-dir` multiple times:

```
hundo annotate \
    --filter-adapters qc_references/adapters.fa.gz \
    --filter-contaminants qc_references/phix174.fa.gz \
    --out-dir mothur_sop_silva \
    --database-dir annotation_references \
    --reference-database silva \
    --input-dir collection2/rawdata \
    --input-dir collection3/rawdata \
    'collection1/LM_*.fastq.gz'
```

Dependencies are installed by default in the results directory defined on the command line as `--out-dir`. If you want to re-use dependencies across many analyses and not have to re-install each time you update the output directory, use Snakemake's `--conda-prefix`:

```
hundo annotate \
    --out-dir mothur_sop_silva \
    --database-dir annotation_references \
    --reference-database silva \
    mothur_sop_data \
    --conda-prefix /Users/brow015/devel/hundo/example/conda
```

Tip: In instances where compute nodes do not have access to the internet, download the reference databases and conda packages in advance.

To download the references for SILVA, run:

```
hundo download --database-dir annotation_references \
    --jobs 5 --reference-database silva
```

To download the conda environment:

```
hundo annotate \
    --out-dir mothur_sop_silva \
    --database-dir annotation_references \
    --reference-database silva \
    mothur_sop_data \
    --conda-prefix /Users/brow015/devel/hundo/example/conda \
    --create-envs-only
```

CHAPTER 3

Annotation Parameters

Argument list, definitions and default values for `hundo annotate`:

Argument	Type or Choice	Description	Default
`--prefilter-file-size`	INTEGER	Any FASTQ file size smaller than this in bytes is omitted from being processed.	100000
`--jobs`	INTEGER	Use at most this many cores in parallel. The total running tasks at any given time will be jobs divided by threads.	auto
`--out-dir`	TEXT	Results output directory.	current directory
`--no-conda`		Do not use conda environments. Requires that all dependencies are installed and executable.	FALSE
`--dryrun`		Do not execute anything, just show the commands that will be executed by Snakemake.	FALSE
`--author`	TEXT	Will show in footer of summary HTML document.	`uname`
`--aligner`	[blast|vsearch]	Local aligner; *blast* is more sensitive while *vsearch* is much faster	blast
`--threads`	INTEGER	When a step is multi-threaded, use this many threads. This is all or a subset of `--jobs`.	8
`--database-dir`	TEXT	Directory containing reference data or new directory into which to download reference data.	'references'
`--filter-adapters`	TEXT	File path to adapters FASTA to use for trimming read ends.	None
`--filter-contaminants`	TEXT	File path to FASTA to use for filtering reads.	None
`--allowable-kmer-mismatches`	INTEGER	Kmer mismatches allowed during adapter trim process.	1
`--reference-kmer-match-length`	INTEGER	Length of kmer to search against contaminant sequences.	27
`--reduced-kmer-min`	INTEGER	Look for shorter kmers at read tips down to this length; 0 disables.	8
`--minimum-passing-read-length`	INTEGER	Passing single-end read length prior to merging.	100
`--minimum-base-quality`	INTEGER	Regions with average quality below this will be trimmed.	10
`--minimum-merge-length`	INTEGER	Minimum allowable read length after merging.	150
`--allow-merge-stagger`		Allow merging of staggered reads by VSEARCH.	FALSE
`--max-diffs`	INTEGER	Maximum number of different bases allowable in overlap.	5
`--min-overlap`	INTEGER	When merging, the minimum length of overlap between reads.	16
`--maximum-expected-error`	FLOAT	After merging, the allowable limit of erroneous bases.	1
`--reference-chimera-filter`	TEXT	Define a file path or set to true to use BLAST reference database.	TRUE
`--minimum-sequence-abundance`	INTEGER	When clustering, do not create any clusters with fewer than this many representative sequences.	2
`--percent-of-allowable-difference`	FLOAT	Maximum difference between an OTU member sequence and the representative sequence of that OTU.	3
`--reference-database`	[silva|greengenes|unite]	Two 16S databases are supported, SILVA and GreenGenes, along with Unite for ITS. References will be downloaded as needed during the execution of the workflow to the location set using `--database-dir`.	'silva'
`--blast-minimum-bitscore`	INTEGER	Filter out alignments below this bitscore threshold and do not use them in the LCA calculation.	100
`--blast-top-fraction`	FLOAT	When calculating LCA, only use this fraction of HSPs from the best scoring alignment.	0.95
`--read-identity-requirement`	FLOAT	When mapping reads back to OTU seed sequences for quantification, require this fraction of sequence identity between sequence and reference.	0.97
`--min-pid`	FLOAT	Minimum percent ID required from VSEARCH hits in order to be retained for LCA calculation	0.85

CHAPTER 4

Annotation Output

An interactive example is available at https://pnnl.github.io/hundo/.

OTU.biom

Biom table with raw counts per sample and their associated taxonomic assignment formatted to be compatible with downstream tools like phyloseq.

OTU.fasta

Representative DNA sequences of each OTU.

OTU.tree

Newick tree representation of aligned OTU sequences.

OTU.txt

Tab-delimited text table with columns OTU ID, a column for each sample, and taxonomy assignment in the final column as a comma delimited list.

OTU_aligned.fasta

OTU sequences after alignment using MAFFT.

all-sequences.fasta

Quality-controlled, dereplicated DNA sequences of all samples. The header of each record identifies the sample of origin and the count resulting from dereplication.

blast-hits.txt

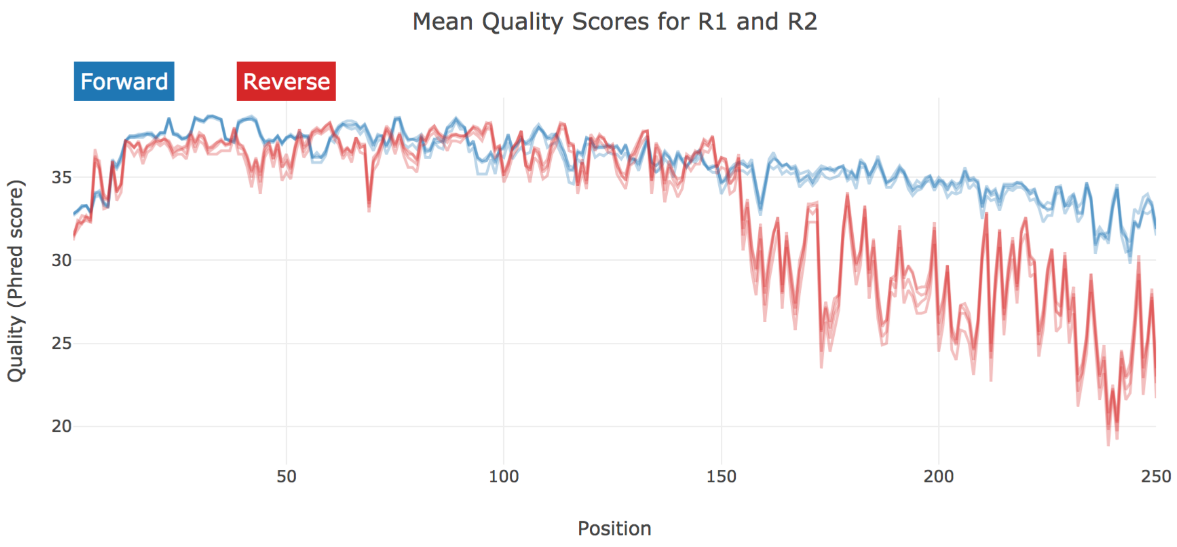
The BLAST assignments per OTU sequence.

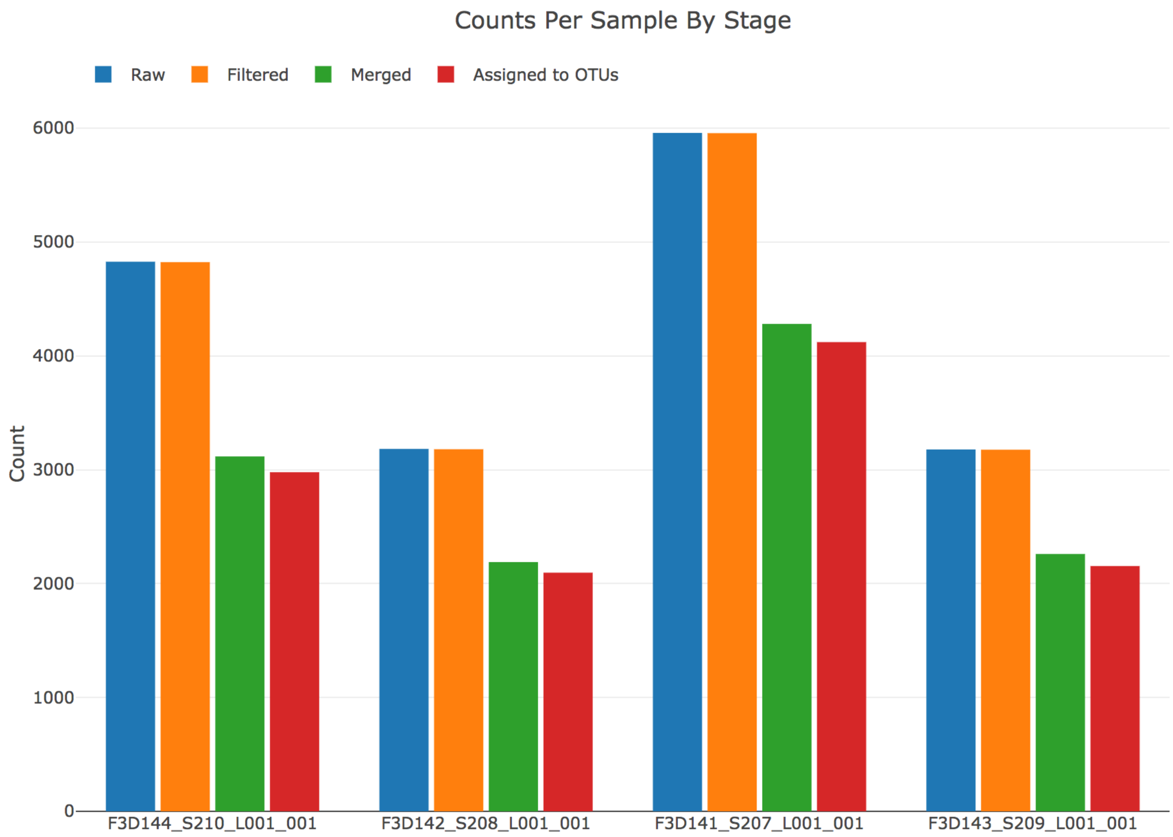
summary.html

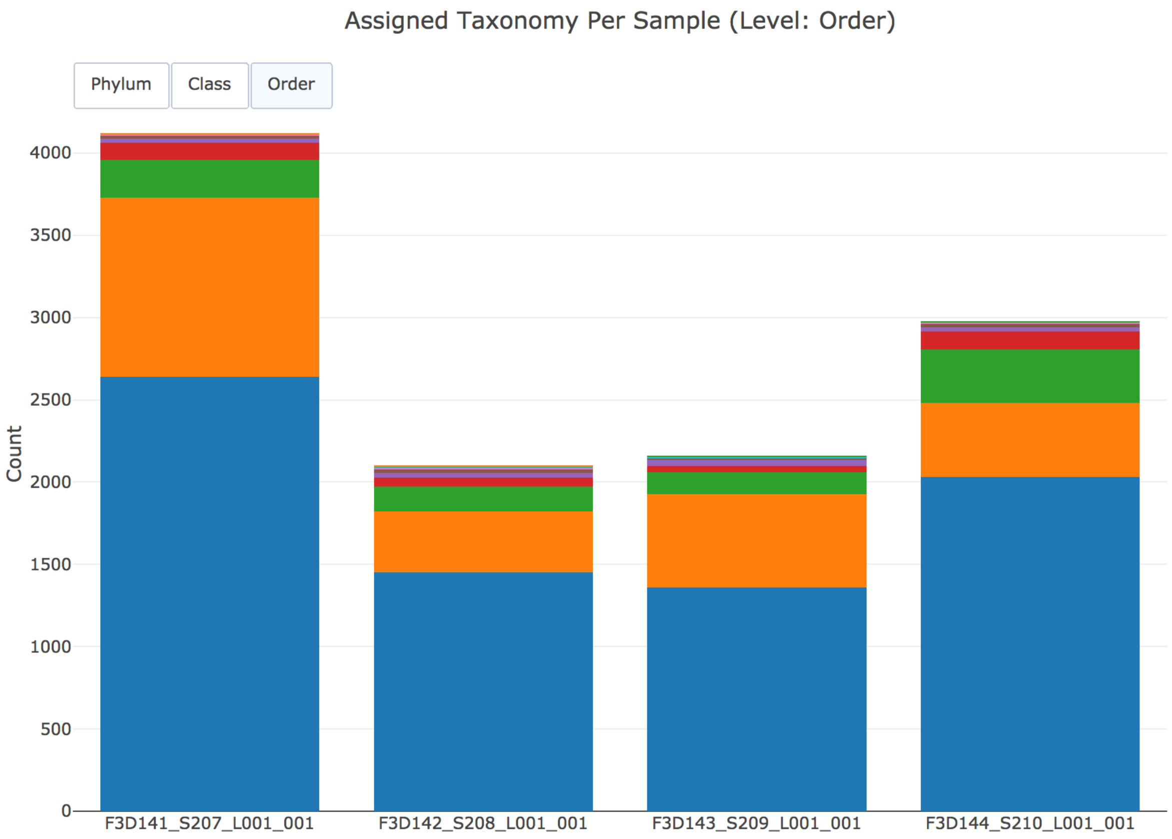
Captures and summarizes data of the experimental dataset. Things like sequence quality:

And counts per sample at varying stages of pre-processing:

Taxonomies are also summarized per sample across phylum, class, and order:


Mean Quality Scores for R1 and R2
Forward
Reverse
Quality (Phred score)
35
30
25
20
50
100
150
200
250
Position


Counts Per Sample By Stage
Raw
Filtered
Merged
Assigned to OTUs
Count
6000
5000
4000
3000
2000
1000
0
F3D144_S210_L001_001
F3D142_S208_L001_001
F3D141_S207_L001_001
F3D143_S209_L001_001


Assigned Taxonomy Per Sample (Level: Order)
Phylum
Class
Order
Count
0
500
1000
1500
2000
2500
3000
3500
4000
F3D141_S207_L001_001
F3D142_S208_L001_001
F3D143_S209_L001_001
F3D144_S210_L001_001

CHAPTER 5

Results Example

https://pnnl.github.io/hundo/

CHAPTER 6

Summary

Snakemake-based amplicon processing protocol for 16S and ITS sequences:

- Performs quality control based on quality, can trim adapters, and remove sequences matching a contaminant database;
- Handles paired-end read merging;
- Integrates *de novo* and reference-based chimera filtering;
- Clusters sequences and annotates using databases that are downloaded as needed;
- Generates standard outputs for these data like a newick tree, a tabular OTU table with taxonomy, and .biom.

This workflow is built using Snakemake and makes use of Bioconda to install its dependencies.