

## Interleaved Paired Read Splitter

Some sequencing platforms (e.g., Illumina) will create read pairs (forward/reverse) that may be interleaved together into one file with the forward read immediately followed by the reverse read or the reads may be in two separate files like `foo_1.fastq` and `foo_2.fastq` where `_1` is the forward read file and `_2` contains the reverse reads (or sometimes `_R1/_R2`).

Write a Python program called `au_pair.py` that accepts a list of positional arguments that are FASTA sequence files in interleaved format and splits them into `_1/_2` files in a `-o|--outdir` argument (default `split`). You should use the original extension of the file, e.g., `inputs/reads1.fa` should be split into `outdir/reads1_1.fa` and `outdir/reads1_2.fa` while `inputs/reads2.fasta` should be split into `outdir/reads2_1.fasta` and `outdir/reads2_2.fasta`.

As always, the program should provide usage statements on `-h|--help` or when run with no arguments. If one of the positional arguments is not a file, print `"<file>" is not a file` to `STDERR` and continue processing. If the `--outdir` does not exist, create it.

For the purposes of this exercise, assume the reads are properly interleaved such that the first read is forward and the second read is its reverse mate. Do not worry about testing the read IDs for forward/reverse or mate pair information. Also assume all input files are in FASTA format and should be written in FASTA format.

## Expected Behavior

```
$ ./au_pair.py
usage: au_pair.py [-h] [-o DIR] FILE [FILE ...]
au_pair.py: error: the following arguments are required: FILE
$ ./au_pair.py -h
usage: au_pair.py [-h] [-o DIR] FILE [FILE ...]
```

Split interleaved/paired reads

positional arguments:

|      |               |
|------|---------------|
| FILE | Input file(s) |
|------|---------------|

optional arguments:

|                                   |   |
|-----------------------------------|---|
| <code>-h, --help</code>           | show this help message and exit                 |
| <code>-o DIR, --outdir DIR</code> | Output directory (default: <code>split</code> ) |

```
$ ./au_pair.py foo
"foo" is not a file
$ ./au_pair.py inputs/reads1.fa
```

```

1: reads1.fa
  Split 4 sequences to dir "split"
$ ./au_pair.py inputs/reads2.fasta -o out
1: reads2.fasta
  Split 500 sequences to dir "out"
$ ./au_pair.py inputs/* -o all
1: reads1.fa
  Split 4 sequences to dir "all"
2: reads2.fasta
  Split 500 sequences to dir "all"

```

## Test Suite

A passing test suite looks like this:

```

$ make test
pytest -v test.py
===== test session starts =====
platform darwin -- Python 3.6.8, pytest-4.2.0, py-1.7.0, pluggy-0.8.1 -- /anaconda3/bin/python
cachedir: .pytest_cache
rootdir: /Users/kyclark/work/worked_examples/2019_spring_finals/read_pair_separator, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 4 items

test.py::test_usage PASSED [ 25%]
test.py::test_bad_input PASSED [ 50%]
test.py::test_good_input1 PASSED [ 75%]
test.py::test_good_input2 PASSED [100%]

===== 4 passed in 1.29 seconds =====

```