merge$_{f}astq$Documentation

Release 0.1.7

Ronak Shah

Oct 25, 2019

CONTENTS:

CHAPTER

ONE

MERGE_FASTQ

Package to merge multiple pair of pair-end fastq data

- Free software: Apache Software License 2.0
- Documentation: https://merge-fastq.readthedocs.io.

1.1 Features

- Given multiple pair-end fastq data merge them into single pair-end fastq w.r.t each READ1 and READ2

1.2 Usage

> merge_fastq –help

Usage: merge_fastq [OPTIONS]

Console script for merge_fastq.

Options:

-fp1, --fastq1 PATH Full path to gziped READ1 fastq files, can be specified multiple times for example: –fastq1 test_part1_R1.fastq.gz –fastq1 test_part2_R1.fastq.gz [required]

-fp2, --fastq2 PATH Full path to gziped READ2 fastq files, can be specified multiple times for example: –fastq2 test_part1_R2.fastq.gz –fastq2 test_part2_R2.fastq.gz [required]

-op, --output-path PATH Full path to write the output files (default: Current working directory)

-of1, --out-fastq1 TEXT Name of the merged output READ1 fastq file (default: merged_fastq_R1.fastq.gz)

-of2, --out-fastq2 TEXT Name of the merged output READ2 fastq file (default: merged_fastq_R2.fastq.gz)

--help Show this message and exit.

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER

TWO

INSTALLATION

2.1 Stable release

To install merge_fastq, run this command in your terminal:

```
$ pip install merge_fastq
```

This is the preferred method to install merge_fastq, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for merge_fastq can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/rhshah/merge_fastq
```

Or download the tarball:

```
$ curl -OJL https://github.com/rhshah/merge_fastq/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER

THREE

USAGE

To use merge_fastq in a project:

```
import merge_fastq
merge_fastq(fastq1,fastq2)
```

To use merge_fastq for command line:

```
> merge_fastq --help

Usage: merge_fastq [OPTIONS]

Console script for merge_fastq.

Options:
-fp1, --fastq1 PATH      Full path to gziped READ1 fastq files, can be
                         specified multiple times for example: --fastq1
                         test_part1_R1.fastq.gz --fastq1
                         test_part2_R1.fastq.gz  [required]
-fp2, --fastq2 PATH      Full path to gziped READ2 fastq files, can be
                         specified multiple times for example: --fastq2
                         test_part1_R2.fastq.gz --fastq2
                         test_part2_R2.fastq.gz  [required]
-op, --output-path PATH  Full path to write the output files (default:
                         Current working directory)
-of1, --out-fastq1 TEXT  Name of the merged output READ1 fastq file
                         (default: merged_fastq_R1.fastq.gz)
-of2, --out-fastq2 TEXT  Name of the merged output READ2 fastq file
                         (default: merged_fastq_R2.fastq.gz)
--help                   Show this message and exit.
```

Example commandline:

- Using default option for multiple fastq1 and fastq2 files

```
$ merge_fastq \
--fastq1 test_part1_R1.fastq.gz \
--fastq1 test_part2_R1.fastq.gz \
--fastq2 test_part1_R2.fastq.gz \
--fastq2 test_part2_R2.fastq.gz \
```

- Using custom option for multiple fastq1 and fastq2 files

```
$ merge_fastq \
--fastq1 test_part1_R1.fastq.gz \
```

(continues on next page)

(continued from previous page)

```
--fastq1 test_part2_R1.fastq.gz \
--fastq2 test_part1_R2.fastq.gz \
--fastq2 test_part2_R2.fastq.gz \
--output-path /path/to/where/you/want/output
--out-fastq1 test_merged_R1.fastq.gz
--out-fastq2 test_merged_R2.fastq.gz
```

CHAPTER

FOUR

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/rhshah/merge_fastq/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

merge_fastq could always use more documentation, whether as part of the official merge_fastq docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/rhshah/merge_fastq/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *merge_fastq* for local development.

1. Fork the *merge_fastq* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/merge_fastq.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv merge_fastq
$ cd merge_fastq/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 merge_fastq tests
$ python setup.py test or pytest
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.5, 3.6 and 3.7, and for PyPy. Check https://travis-ci.org/rhshah/merge_fastq/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ pytest tests.test_merge_fastq
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER

FIVE

CREDITS

5.1 Development Lead

- Ronak Shah <rons.shah@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER

SIX

HISTORY

6.1 0.1.0 (2019-10-21)

- First release on PyPI.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search