

---

# **process***fastqDocumentation*

***Release 2.1.3***

**Ronak Hasmukh Shah**

**Mar 30, 2020**



**CONTENTS:**

- 1 process\_fastq 1**
  - 1.1 Usage . . . . . 1
  - 1.2 Features . . . . . 1
  - 1.3 Credits . . . . . 1
- 2 Installation 3**
  - 2.1 Stable release . . . . . 3
  - 2.2 From sources . . . . . 3
- 3 Usage 5**
- 4 Contributing 7**
  - 4.1 Types of Contributions . . . . . 7
  - 4.2 Get Started! . . . . . 8
  - 4.3 Pull Request Guidelines . . . . . 8
  - 4.4 Tips . . . . . 9
  - 4.5 Deploying . . . . . 9
- 5 Credits 11**
  - 5.1 Development Lead . . . . . 11
  - 5.2 Contributors . . . . . 11
- 6 History 13**
  - 6.1 2.1.0 (2020-03-12) . . . . . 13
  - 6.2 0.1.0 (2019-07-25) . . . . . 13
- 7 Indices and tables 15**



## PROCESS\_FASTQ

This package will help process, merge and link fastq in user specified directory from manifest file

- Free software: Apache Software License 2.0
- Documentation: <https://process-fastq.readthedocs.io>.

### 1.1 Usage

Usage can be found here: <https://process-fastq.readthedocs.io/en/latest/usage.html>

### 1.2 Features

1. Given Manifest file, path to location of raw fastq, path to where they need to linked:
  - a. Get all the folders for the samples and the fastq file
  - b. Check quickly the lenght of the reads if read length is not the same use the shorter read length and trim the fastq
  - c. Merge the final fastq if comming from multiple runs.
  - d. Link all the fastq with the folder structure in user provided location

### 1.3 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.



## INSTALLATION

### 2.1 Stable release

To install `process_fastq`, run this command in your terminal:

```
$ pip install process_fastq
```

This is the preferred method to install `process_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 `process_fastq` can be downloaded from the [Github repo](#).

You can either clone the public repository:

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

Or download the [tarball](#):

```
$ curl -OL https://github.com/rhshah/process_fastq/tarball/master
```

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

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





## USAGE

To use `process_fastq` in a project:

```
import process_fastq
process_fastq.run(sample_id, request_id, run_id, fastq_path, output_path, cutadapt_
↳ path)
```

To use `process_fastq` from cli:

```
Usage: process_fastq [OPTIONS]

Options:
  -s, --sample-id TEXT          Sample id to get the fastq files can be
                                either IGO sample id or Investigator sample
                                id (eg: -s EDD_ret_pt049_cf02) [required]
  -p, --request-id TEXT         IGO request id to get the fastq files.
                                (eg: -p Project_05500_GB or -p 05500_GB)
  -r, --run-id TEXT             Run id to get the fastq files, can be
                                specified multiple times (eg: -r PITT_0376 -r
                                PITT_0378)
  -fp, --fastq-path PATH        Full path to fastq files [required]
  -op, --output-path PATH       Full path to where we link the output files
                                [required]
  -cp, --cutadapt-path PATH     Full path to location of cutadapt executable
                                [required]
  -l, --expected-read-length INTEGER Expected read length from the fastq file
  --version                     Show the version and exit.
  -v, --verbosity LVL          Either CRITICAL, ERROR, WARNING, INFO or
                                DEBUG
  --help                       Show this message and exit.
```

Example commandline:

```
$ process_fastq \
-p request_id \
-s smaple_name \
-r RunID \
-fp /path/to/fastq/directory \
-op /path/to/output/directory \
-cp /path/to/cutadapt
```

To use `link_fastq_juno.py` from cli:

Usage: `link_fastq_juno.py` [OPTIONS]

**Options:**

- m, --manifest-file PATH** Manifest file having information about run id and sample id to get the fastq files (eg: -m Project\_05500\_GB\_manifest.xlsx) [required]
- p, --request-id TEXT** IGO request id to get the fastq files. (eg:-p Project\_05500\_GB or -p 05500\_GB) [required]
- fp, --fastq-path PATH** Full path to fastq files [required]
- op, --output-path PATH** Full path to where we link the output files [required]
- cp, --cutadapt-path PATH** Full path to location of cutadapt executable [required]
- pfp, --process-fastq-path PATH** Full path to location of cutadapt executable [required]
- l, --expected-read-length INTEGER** Expected read length from the fastq file
- version** Show the version and exit.
- v, --verbosity LVL** Either CRITICAL, ERROR, WARNING, INFO or DEBUG
- help** Show this message and exit.

Example commandline:

```
$ python3 link_fastq_juno.py \  
-p request_id \  
-m /path/to/manifest.xlsx \  
-pfp /path/to/process_fastq \  
-fp /path/to/fastq/directory \  
-op /path/to/output/directory \  
-cp /path/to/cutadapt
```

## 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/process\\_fastq/issues](https://github.com/rhshah/process_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

`process_fastq` could always use more documentation, whether as part of the official `process_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/process\\_fastq/issues](https://github.com/rhshah/process_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 *process\_fastq* for local development.

1. Fork the *process\_fastq* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/process_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 process_fastq
$ cd process_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 process_fastq tests
$ python setup.py test or py.test
$ 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, 3.4, 3.5 and 3.6, and for PyPy. Check [https://travis-ci.org/rhshah/process\\_fastq/pull\\_requests](https://travis-ci.org/rhshah/process_fastq/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_process_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:

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

Travis will then deploy to PyPI if tests pass.



CREDITS

## 5.1 Development Lead

- Ronak Hasmukh Shah <[rons.shah@gmail.com](mailto:rons.shah@gmail.com)>

## 5.2 Contributors

None yet. Why not be the first?





## HISTORY

### 6.1 2.1.0 (2020-03-12)

- Modifying to suite new access workflow.

### 6.2 0.1.0 (2019-07-25)

- First release on PyPI.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`