The Journal of Open Source Software

fqfa: A pure Python package for genomic sequence files

Alan F. Rubin[1, 2]

1 The Walter and Eliza Hall Institute of Medical Research **2** The University of Melbourne

DOI: 10.21105/joss.02076

Software

- Review
- Repository
- Archive

Editor: Lorena Pantano

Reviewers:

- @natir
- @FlorianThibord

Submitted: 03 February 2020
Published: 01 March 2020

Summary

Modern bioinformatics requires the use of many field-specific file formats. Two of the most prevalent formats for representing biological sequences are FASTA (Pearson & Lipman, 1988) and FASTQ (Cock, Fields, Goto, Heuer, & Rice, 2010). While multiple feature-rich Python bioinformatics libraries exist that can process biological sequence files (Cock et al., 2009; scikit-bio Development Team, 2013), they require complex compiled dependencies that may limit their use in non-Unix environments. Other FASTA or FASTQ specific Python libraries (Du, 2019; Hunt, 2013; Pedersen, 2010; Shirley, Ma, Pedersen, & Wheelan, 2015) are outdated, require runtime dependencies, or make heavy use of C extensions that prioritize speed over readability and portability.

fqfa is a pure Python package that aims to fill the needs of bioinformatics and computational biology researchers who want a simple and efficient solution for working with files in FASTA and FASTQ formats. It has no dependencies outside of the Python standard library (with the exception of backported dataclasses (Smith, 2017) for Python 3.6 users) and makes use of newer language features such as type hinting and f-strings to improve readability. These implementation details make fqfa highly suitable for use in notebooks and projects that have simple requirements, with underlying code that is easy for novice bioinformaticians and students to understand and explore.

Although fqfa is written in pure Python, its performance is comparable to modules using C extensions like pyfastx (Du, 2019) for tasks such as processing a FASTQ file sequentially and collecting or filtering on quality statistics from the high-throughput sequencing reads. Detailed benchmarking results and usage examples comparing fqfa and pyfastx (Du, 2019) are available as part of the fqfa documentation in static format as well as in Jupyter notebooks (Kluyver et al., 2016).

fqfa is released under the BSD 3-Clause License and is available from GitHub and PyPI.

Acknowledgements

Thank you to Matthew Wakefield for helpful discussion and code review. The research benefited by support from the Victorian State Government Operational Infrastructure Support and Australian Government NHMRC Independent Research Institute Infrastructure Support. AFR was supported by the National Human Genome Research Institute of the NIH under award number RM1HG010461.

References

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., et al. (2009). Biopython: Freely available Python tools for computational molecular biol-

ogy and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. doi:10.1093/bioinformatics/btp163

Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L., & Rice, P. M. (2010). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, *38*(6), 1767–1771. doi:10.1093/nar/gkp1137

Du, L. (2019, March). Lmdu/pyfastx. Retrieved from https://github.com/lmdu/pyfastx

Hunt, M. (2013, September). Sanger-pathogens/Fastaq. Pathogen Informatics, Wellcome Sanger Institute. Retrieved from https://github.com/sanger-pathogens/Fastaq

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. (F. Loizides & B. Schmidt, Eds.). IOS Press.

Pearson, W. R., & Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, *85*(8), 2444–2448. doi:10.1073/pnas.85.8.2444

Pedersen, B. (2010, July). Brentp/pyfasta. Retrieved from https://github.com/brentp/pyfasta

scikit-bio Development Team. (2013, December). Biocore/scikit-bio. biocore. Retrieved from https://github.com/biocore/scikit-bio

Shirley, M. D., Ma, Z., Pedersen, B. S., & Wheelan, S. J. (2015). *Efficient "pythonic" access to FASTA files using pyfaidx* (No. e1196). PeerJ Inc. doi:10.7287/peerj.preprints.970v1

Smith, E. V. (2017, May). Ericvsmith/dataclasses. Retrieved from https://github.com/ericvsmith/dataclasses