




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](https://doi.org/10.21105/joss.02076)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Lorena Pantano](#) 

Reviewers:

- [@natir](#)
- [@FlorianThibord](#)

Submitted: 03 February 2020

Published: 27 February 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

Modern bioinformatics requires the use of many field-specific file formats. Two of the most prevalent 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 (Cock et al., 2009; scikit-bio Development Team, 2013), they require complex compiled dependencies that can limit their use in non-Unix environments. Other FASTA/FASTQ specific Python libraries (Du, 2019; Hunt, 2013; Pedersen, 2010; Shirley, Ma, Pedersen, & Wheelan, 2015) are either outdated, require runtime dependencies, or make heavy use of C extensions that prioritize speed over readability and portability.

fqfa is a pure Python package for working with files in FASTA and FASTQ formats. It has no dependencies outside of the Python standard library and makes use of newer language features such as type hinting to improve readability. This makes it more suitable for use in notebooks or projects with simple requirements, as well as easier to understand by novice bioinformaticians and students.

Although it is written in pure Python, fqfa's performance is comparable to modules using C extensions like pyfastx (Du, 2019) for tasks such as processing a FASTQ file sequentially. Timing results and example usage are available as part of the fqfa documentation.

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 biology and bioinformatics. *Bioinformatics*, 25(11), 1422–1423. doi:[10.1093/bioinformatics/btp163](https://doi.org/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](https://doi.org/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>
- 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](https://doi.org/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](https://doi.org/10.7287/peerj.preprints.970v1)