The Journal of Open Source Software

Gene Fetch: A Python tool for sequence retrieval from GenBank across the tree of life

Daniel A. J. Parsons [1] **and Benjamin Price** [1]

1 Natural History Museum, Cromwell Road, London, SW7 5BD, United Kingdom

DOI: 10.21105/joss.08456

Software

- Review
- Repository
- Archive

Editor: Abhishek Tiwari

Reviewers:

- @johnvusich
- @hrhotz
- @ManavalanG

Submitted: 07 May 2025
Published: 14 August 2025

Summary

`Gene Fetch` is an open-source Python tool that automates the retrieval of sequence data from the National Center for Biotechnology Information (NCBI) GenBank sequence database (Benson et al., 2012; Sayers et al., 2024). It simplifies sequence acquisition for biological research by retrieving both protein and nucleotide sequences for user-specified targets from across the tree of life, accepting NCBI taxonomic identifiers or hierarchical taxonomic information as input.

Statement of need

Comparative genetic analyses within or between taxonomic groups often requires researchers to gather large numbers of gene or protein sequences from public repositories such as GenBank, which currently holds over 4.7 billion nucleotide sequences from over 580,000 formally described species (Sayers et al., 2024). Streamlined access to this invaluable database is essential for comprehensive sequence data analysis, however, this task presents several significant challenges: (1) Limited database curation leading to inconsistent sequence annotations. (2) Variable sequence representation across taxonomic groups on the databases. (3) Time-intensive manual retrieval processes that do not scale efficiently.

While several existing tools enable sequence retrieval from NCBI sequence repositories, they often require considerable bioinformatics expertise, are limited in functionality or data scope, or are suited for slightly different applications. E-utilities like Entrez Direct (Kans, 2024) offer broad NCBI database access via several APIs but rely on manual NCBI search term construction and significant scripting expertise, burdening the user with navigating database-specific syntax and structure. Similarly, other tools like CRABS: Creating Reference databases for Amplicon-Based Sequencing (Jeunen et al., 2023) and RESCRIPt: REference Sequence annotation and CuRatIon Pipeline (Robeson II et al., 2021) offer bulk, programmatic retrieval of sequences from several databases (e.g. NCBI, BOLD, etc.), yet they also require manual search terms construction, operate on a single taxon, and often require substantial post-processing by the user. NCBI Datasets (O'Leary et al., 2024), while facilitating web- and programmatic-access to NCBI sequence data, is restricted to the curated RefSeq database (a small subset of sequences available on GenBank), limited to species-level queries, and lacks sequence filtering and batch processing capabilities.

In contrast, `Gene Fetch` offers an accessible, high-throughput solution that automates and simplifies sequence retrieval from GenBank by matching sought after targets to feature table annotations in GenBank records, and requires no prior NCBI syntax knowledge. The tool integrates robust logging, error handling, checkpointing, and a standardised output format, making it suited for reproducible, efficient, and biologically-informed sequence retrieval at scale. `Gene Fetch` directly addresses the challenge of variable sequence representation by systematically traversing taxonomic hierarchies when target sequences are unavailable at the initially specified taxonomic rank (e.g., species → genus → family, etc), documenting the matched rank. This is especially valuable for researchers working with non-model organisms or taxonomic groups with limited sequence data, facilitating retrieval of the taxonomically closest available sequence.

`Gene Fetch` supports 'batch' and 'single' query modes across both protein and nucleotide sequences, with automated CDS extraction, customisable length filtering, and fallback mechanisms for atypical GenBank annotations. The integrated 'batch' mode processes multiple input taxa and retrieves the single 'best' sequence per taxon, whilst 'single' mode exhaustively searches for all target sequences for a specified taxon. Collectively, these modes enable efficient retrieval of sequence data for genomic and phylogenetic studies across diverse taxa. It can also process variable GenBank features, including complementary strands, joined sequences, and whole genome shotgun entries, enabling extraction regions of interest from variable feature annotations (e.g., COI from mitogenome records). Cross-validation of retrieved NCBI taxonomy against the input taxonomy prevents taxonomic homonyms matches (identical names referring to different organisms across the tree of life). At release, the tool is optimised for 18 common targets, including "barcoding" genes, with curated synonyms for improved search specificity. Users can also specify additional markers to those 18 targets, and optionally retrieve corresponding GenBank records for each fetched sequence.

Implementation

`Gene Fetch` is implemented in Python `(>=3.9)` and leverages two main libraries: Biopython (Cock et al., 2009), which, through subpackages of the Bio package (Bio.Entrez, Bio.Seq, Bio.SeqIO, and Bio.SeqRecord), provides the foundation for NCBI database access, sequence parsing and manipulation; and RateLimit, which manages NCBI API rate constraints (beyond those provided in Bio.Entrez) to prevent request throttling.

The tool follows a modular design with four primary components: configuration manager (handles search parameters and target-specific search term generation), Entrez handler (manages NCBI API interactions with comprehensive error handling), sequence processor (implements core logic for sequence extraction and validation), and output manager (controls file generation and reporting). Detailed logs of parameter and search progress are produced, and checkpoint recovery enables interrupted runs to be resumed. `Gene Fetch` utilises batch processing and taxonomic lineage caching to maximise efficiency, and can process hundreds to thousands of samples (in 'batch' mode) with modest computational resources, as outlined in the GitHub repository.

Availability

`Gene Fetch` is distributed as a Python package on PyPI and a Bioconda package, with the source code, testing modules, and a standalone script available under an MIT license through the GitHub repository. A supplementary shell wrapper script is also provided in the GitHub repository for submitting a `Gene Fetch` job to a High-Performance Computing (HPC) cluster running the SLURM job scheduler.

Acknowledgments and contributions

Contributions following the project's guidelines are welcomed. Users are encouraged to report bugs and suggest enhancements through the GitHub issue tracker. We thank Maria Kamouyiaros and Oliver White (Natural History Museum, London) for their contributions to the conceptualisation and testing of `Gene Fetch`. This work is supported by Biodiversity Genomics Europe (Grant no.101059492), funded by Horizon Europe under the Biodiversity, Circular Economy and Environment call (REA.B.3); co-funded by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract numbers 22.00173 and 24.00054;

and by the UK Research and Innovation (UKRI) under the Department for Business, Energy and Industrial Strategy's Horizon Europe Guarantee Scheme.

References

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2012). GenBank. *Nucleic Acids Research*, *41*(D1), D36–D42. https://doi.org/10.1093/nar/gks1195

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & Hoon, M. J. L. de. (2009). Biopython: Freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Jeunen, G.-J., Dowle, E., Edgecombe, J., von Ammon, U., Gemmell, N. J., & Cross, H. (2023). CRABS—a software program to generate curated reference databases for metabarcoding sequencing data. *Molecular Ecology Resources*, *23*(3), 725–738. https://doi.org/10.1111/1755-0998.13741

Kans, J. (2024). Entrez direct: E-utilities on the unix command line. In *Entrez programming utilities help [internet]*. National Center for Biotechnology Information (US). https://www.ncbi.nlm.nih.gov/books/NBK179288/

O'Leary, N. A., Cox, E., Holmes, J. B., Anderson, W. R., Falk, R., Hem, V., Tsuchiya, M. T., Schuler, G. D., Zhang, X., Torcivia, J., Ketter, A., Breen, L., Cothran, J., Bajwa, H., Tinne, J., Meric, P. A., Hlavina, W., & Schneider, V. A. (2024). Exploring and retrieving sequence and metadata for species across the tree of life with NCBI datasets. *Scientific Data*, *11*(1), 732. https://doi.org/10.1038/s41597-024-03571-y

Robeson II, M. S., O'Rourke, D. R., Kaehler, B. D., Ziemski, M., Dillon, M. R., Foster, J. T., & Bokulich, N. A. (2021). RESCRIPt: Reproducible sequence taxonomy reference database management. *PLOS Computational Biology*, *17*(11), 1–37. https://doi.org/10.1371/journal.pcbi.1009581

Sayers, E. W., Cavanaugh, M., Frisse, L., Pruitt, K. D., Schneider, V. A., Underwood, B. A., Yankie, L., & Karsch-Mizrachi, I. (2024). GenBank 2025 update. *Nucleic Acids Research*, *53*(D1), D56–D61. https://doi.org/10.1093/nar/gkae1114