pynnotate: a flexible tool for retrieving and processing GenBank data in molecular evolution research and education

Fernanda S. Caron[1*], Felipe de M. Magalhães[2*], Matheus M. A. Salles[3], & Fabricius M. C. B. Domingos[4]

[1]Programa de Pós-Graduação em Ecologia e Conservação, Universidade Federal do Paraná, Curitiba, Paraná, Brazil.

[2]Programa de Pós-Graduação em Ciências Biológicas, Centro de Ciências Exatas e da Natureza, Universidade Federal da Paraíba, João Pessoa, Paraíba, Brazil.

[3]Programa de Pós-Graduação em Zoologia, Universidade Federal do Paraná, Curitiba, Paraná, Brazil.

[4]Departamento de Zoologia, Universidade Federal do Paraná, Curitiba, Paraná, Brazil.

[*]Co-first authorship.

Corresponding author: Fernanda S. Caron. Email: fernandacaron@ufpr.br

Running title: pynnotate: a tool for molecular evolution studies

Acknowledgments

FSC was funded through a graduate scholarship from CAPES (#88887.923452/2023-00). FMM acknowledges fellowship and research support from the Brazilian National Council for Scientific and Technological Development (CNPq; #315066/2025-3). FMCBD acknowledges a productivity fellowship from CNPq (#311997/2025-2).

Data availability

The ‘pynnotate’ public repository is available at https://github.com/fernandacaron/pynnotate.

Conflict of interest

The authors have no conflicts of interest to declare.

Author contribution statements

FSC (Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Writing—original draft, Writing—review & editing), FMM (Conceptualisation, Methodology, Software, Validation, Formal analysis, Investigation, Writing—review & editing), MMAS (Conceptualisation, Methodology, Validation, Writing—original draft, Writing—review & editing), and FMCBD (Conceptualisation, Methodology, Validation, Writing—review & editing, Project Administration).

Abstract

Pynnotate is a Python-based tool designed for automated retrieval, parsing, and extraction of annotated gene sequences from GenBank records. The tool addresses the common challenges researchers face when working with GenBank data, including inconsistent gene nomenclature, redundant sequences, and the need for standardised gene extraction across multiple taxa. Pynnotate operates through both a graphical user interface and a command-line interface, making it accessible to users with varying levels of bioinformatics experience. The tool supports flexible sequence retrieval through manually defined accession numbers or NCBI query terms, and offers three distinct filtering modes: unconstrained (all sequences), strict (one sequence per species prioritising gene completeness), and flexible (multiple sequences per species when contributing different genes). Key features include synonym resolution for gene names, customizable sequence headers, metadata tracking, and automated gene extraction into separate files. Built-in dictionaries support animal and plant mitochondrial DNA, chloroplast DNA, and ribosomal DNA, and allow users to provide custom synonym dictionaries. The tool generates structured output including FASTA files, metadata matrices, and detailed logs, facilitating integration with downstream analyses. Designed for speed and scalability, pynnotate efficiently handles large datasets, allowing quick retrieval and extraction of annotated sequences across multiple taxa. Finally, pynnotate serves as a valuable resource for both research applications and educational settings, particularly benefiting educators conducting bioinformatics analyses with students with limited command-line experience.

Keywords: bioinformatics; comparative genomics; feature extraction; molecular evolution; phylogenetics; Python; sequence annotation.

1. Introduction

Public molecular databases have revolutionised molecular evolution research by providing access to vast amounts of genomic data across diverse taxa. Pioneering the field, Margaret Dayhoff and her colleagues were the first to publish an online database of molecular sequences in 1965, aiming to assist research and make protein data easily accessible (Hagen, 2011). This *Atlas of Protein Sequence and Structure* was particularly important because it inspired the creation of GenBank. Since its first release in 1982, the amount of sequence data in GenBank has roughly doubled every 18 months, reflecting the increasing scale of molecular data generation worldwide (https://www.ncbi.nlm.nih.gov/genbank/statistics/). Currently, GenBank hosts hundreds of millions of publicly available nucleotide sequences, but despite this wealth of data, retrieving and organising specific genetic features for targeted evolutionary analyses remains challenging, particularly for non-model organisms.

A key contribution of GenBank to molecular research is the availability of stable accession numbers and standardised records, which facilitate data traceability and reproducibility. Making molecular data publicly available has enabled collaborative science and has allowed researchers to replicate previously published results (Leray et al., 2019). However, to successfully replicate molecular evolution research, the processing and manipulation of sequences must be performed almost exactly to achieve a comparable alignment (Baykal et al., 2024). An issue arises when large numbers of these sequences must be analysed simultaneously because GenBank records are not necessarily standardised. For instance, different sequences may use various names for the same genes (e.g., the cytochrome oxidase I may be named COI, or COX1, or CO1), or sequences related to one feature might be mistakenly registered under another feature's name. While it is standard to address these problems when curating datasets for molecular evolution research projects, the specific methods used to handle these issues are not always clearly defined, limiting replicability.

There is a growing demand for flexible tools that streamline the retrieval of sequence features for evolutionary analysis. It is common practice in molecular evolutionary studies to use complete mitogenomes or similar records with annotated features, analysing these features separately. Currently, GenBank does not provide an option to download each feature from these records separately, other than manually. This limitation becomes a major bottleneck in large-scale comparative studies involving hundreds or thousands of annotated sequences. Moreover, the tools available for separate downloads by feature have several limitations that may prevent the full utilisation of GenBank data, such as not retrieving reference sequences (e.g., Borstein & O'Meara, 2018), potentially resulting in incomplete datasets.

Beyond research applications, there is growing interest in using bioinformatics tools for educational purposes, particularly those that combine graphical and command-line interfaces to support progressive instruction across different levels of expertise. Such tools have proven effective in promoting active learning and improving bioinformatics and evolutionary biology literacy among students with limited programming experience (Emery & Morgan, 2017; Harris et al., 2020). Instructors can employ these platforms to teach essential skills—such as accessing public databases, resolving gene-name inconsistencies, curating metadata, and extracting annotated features—while introducing broader conceptual topics in molecular evolution, including homology, alignment quality, and annotation errors. The integration of reproducible workflows into teaching environments also provides a valuable opportunity to emphasise open science practices and data transparency (e.g., Cokelaer et al., 2023; Toelch & Ostwald, 2018). Consequently, user-friendly and reproducible bioinformatics tools can serve as powerful educational resources, bridging the gap between theoretical concepts and hands-on data analysis in molecular evolution.

To address these research and educational gaps, we developed pynnotate, a flexible Python-based tool that standardises the extraction and processing steps, allowing for

reproducible outputs across diverse datasets. Pynnotate is not intended to perform genome assembly or large-scale annotation pipelines. Instead, it facilitates targeted sequence retrieval and curation of already annotated loci, particularly mitochondrial (mtDNA), plastid (cpDNA), ribosomal (rDNA), and selected nuclear markers commonly used in molecular evolution and systematics. In addition to its command-line interface, pynnotate also includes a graphical version, enabling use by researchers and students without experience in Python, including in classroom settings. Inspired by similar tools and packages developed in the fields of ecology and evolution, we named our module pynnotate, a reference to the Python programming language and its function related to gene feature annotation. It supports batch processing of large datasets by retrieving and parsing multiple GenBank records sequentially, allowing users to extract multiple features across many species in a single run. Here, we describe the implementation of pynnotate and demonstrate its utility through easy-to-follow tutorials. Pynnotate is publicly available and can be easily adapted for various studies using GenBank data.

2. Package overview

a. Overview

Pynnotate is a Python-based tool designed for automated retrieval, parsing, and extraction of annotated gene sequences from GenBank records, with a focus on mitochondrial and chloroplast genomes. The module is designed for workflows requiring standardised gene extraction across multiple taxa, with support for synonym resolution, customizable headers, and metadata tracking. Pynnotate operates on both manually defined accession numbers and NCBI query terms, enabling flexible sequence retrieval. Its extraction method recognises and parses multiple annotation fields (e.g., gene, product, note), detecting aliases using synonym dictionaries, and organises the output by gene. Currently implemented on pynnotate are

dictionaries for animal and plant mtDNA, cpDNA, rDNA, and some animal nuclear genes. Although not complete, an advantage of pynnotate is its flexibility, which allows the user to provide a custom synonym dictionary, and its functionalities will apply to any other sequence. Specific logic is implemented for tRNA-Leu and tRNA-Ser genes to differentiate positional isoforms based on genomic context. Optional filters allow control over gene overlap, record length, and the number of individuals per species, making the tool suitable for low- and high-throughput genomic data processing. In this context, pynnotate has three current filtering modes: (1) download all sequences matching the search query (unconstrained mode); (2) download one individual per species, choosing those that maximize the number of genes downloaded per species (strict); and (3) downloads many individuals per species, but choosing only one individual per gene (i.e., supermatrix) (flexible). Limitations include dependency on the accuracy and completeness of GenBank annotations.

b. Implementation

Pynnotate is written in Python (≥3.8) and can be executed on Windows, macOS, or Linux. Pynnotate relies primarily on the Biopython library for parsing GenBank records, and several other external dependencies (https://github.com/fernandacaron/pynnotate). These dependencies are automatically installed and require no additional compilation steps. Bugs and technical issues can be reported directly through the GitHub repository's issues page.

In the command-line, all execution settings of pynnotate are specified via a YAML configuration file. This file defines the data source (accession numbers or search terms), desired genes, sequence filters (e.g., minimum length, unique species), and output options such as header customisation and log generation. All results are saved in a structured directory containing FASTA files, logs, and metadata matrices, facilitating integration with downstream analytical tools. In addition to storing extracted sequences, pynnotate logs all retrieved records

and explicitly reports those that were not extracted, together with the reason for exclusion (e.g., missing annotations or applied filters). This ensures full traceability of dataset assembly and allows users to review which records contributed to the final output. As we recognise that running Python scripts can be a barrier for many users in bioinformatics, we have developed a graphical user interface (GUI) alongside the module, which users can download and install locally (on Windows, macOS, or Linux). This application allows all module functionalities to be executed through an intuitive and user-friendly interface. In this way, we hope this tool will serve as a flexible and accessible resource for researchers to retrieve data from GenBank in an automated and efficient manner, while also supporting the reproducibility of the results. Furthermore, we believe that the graphical interface is particularly valuable for educators who wish to conduct bioinformatics analyses in classroom settings with students who have no prior experience with command-line tools, coding, or bioinformatics.

3. Graphic interface workflow

a. Getting started

The pynnotate interface was designed to make it easy for users to get all functionalities in a simple step-by-step workflow (Figure 1A). The first required parameter to retrieve GenBank sequences is the genome type (Figure 1B). This genome type is used subsequently for gene extraction and filtering, so not all functionalities will be affected by this choice. More specifically, when gene extraction is disabled, all sequences matching your search will be downloaded regardless of genome type. The genome type can be one of (1) animal mitochondria, (2) plant mitochondria, (3) chloroplast, or (4) other. For the first three options, some synonyms are already built into pynnotate. These synonyms are used to identify which sequences in GenBank databases correspond to each gene. Since GenBank lacks gene names standardisation, not all gene synonyms are included in our dictionary. Therefore, users can

specify new synonyms using the "Add gene synonyms" option in Step 1 (Figure 1B). We recommend running the program first to identify any unrecognised gene synonyms. Alternatively, if options 1-3 do not meet the user's needs, a new dictionary can also be provided using the genome type "Other" and specifying which synonyms should be considered in the "Add gene synonyms" option.

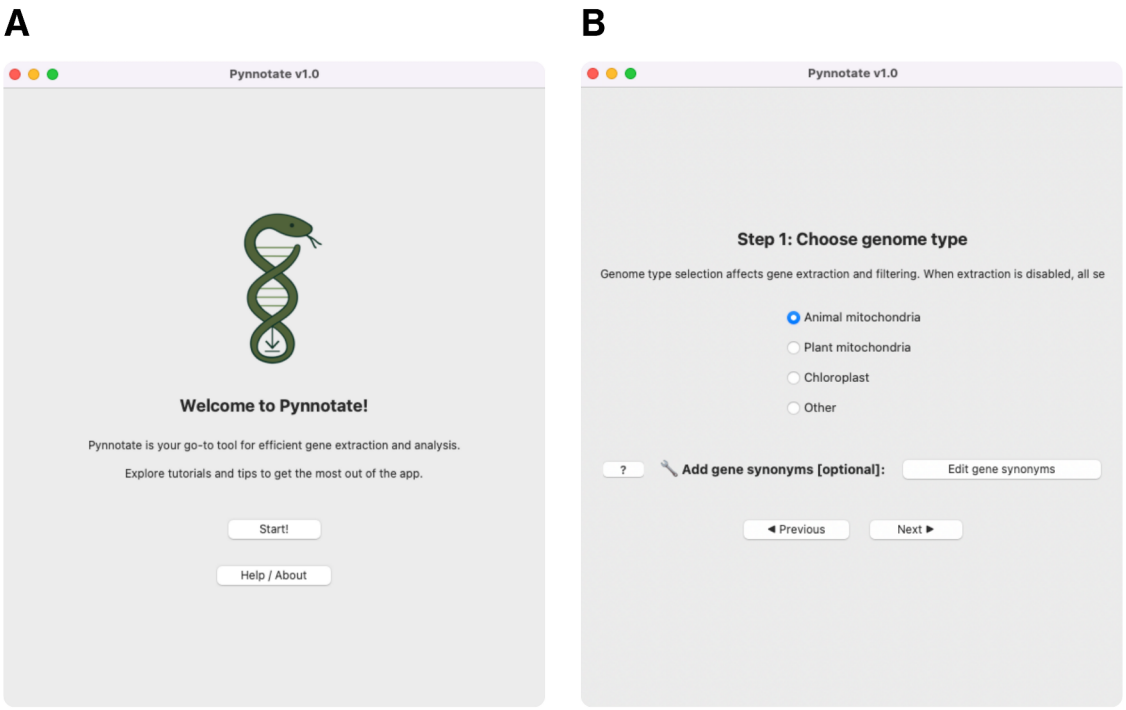

Figure 1. Pynnotate graphic interface. **A.** Welcome page including citation information. **B.** Step 1: Defining genome type and gene synonyms.

b. Assembling a search query

Pynnotate searches for and downloads sequences from GenBank either by assembling a search query with user-defined parameters or directly using accession numbers provided by the user. In Step 2 (Figure 2), users can specify these accession numbers in the first field. Alternatively, a search query can be assembled using the remaining available fields. When

choosing this approach, users can specify particular genes, organism names (e.g., families, genera, species), and publication terms. An additional field allows users to specify any other query components. Furthermore, users can restrict the query to mitochondrial genes, the mitogenome, or the chloroplast genome. As a side note, since GenBank does not explicitly support such restricted filtering, some sequences outside the intended scope may still be downloaded when using these options. Users can also choose to exclude unannotated sequences ('UNVERIFIED' in GenBank) at this step. Finally, all fields are combined into a single query. The search shown in Figure 2, for example, would generate '(COI) AND (Anura[Organism]).'

Figure 2. Pynnotate graphic interface. Step 2: Build a custom search query.

c. Filtering and other options

Besides sequence search, pynnotate offers many filtering options that can be applied to searches before results are returned to the user (Figure 3A). One of the most important features is the different filtering modes pynnotate allows. First, 'unconstrained' includes all available sequences regardless of redundancy, useful when users want to manually explore or curate all records. Second, 'strict' selects a single record per species by maximising the number of requested genes present in the annotation table. When multiple records contain the same number of target genes, pynnotate prioritises records with longer cumulative gene length and complete feature annotations. Ties are resolved deterministically to ensure reproducibility. Finally, 'flexible' selects one representative sequence per species for each gene independently. As a consequence, different genes within a multi-gene dataset may derive from distinct individuals of the same species. This approach maximizes gene occupancy across loci (e.g., in supermatrix analyses), but should be used with awareness of its implications for downstream phylogenetic concatenation. Both strict and flexible modes rely on the synonym dictionary to extract genes from each species. Furthermore, in Step 3, users can specify the header format for each sequence (e.g., organism name and accession number) and set minimum and maximum sequence lengths in base pairs.

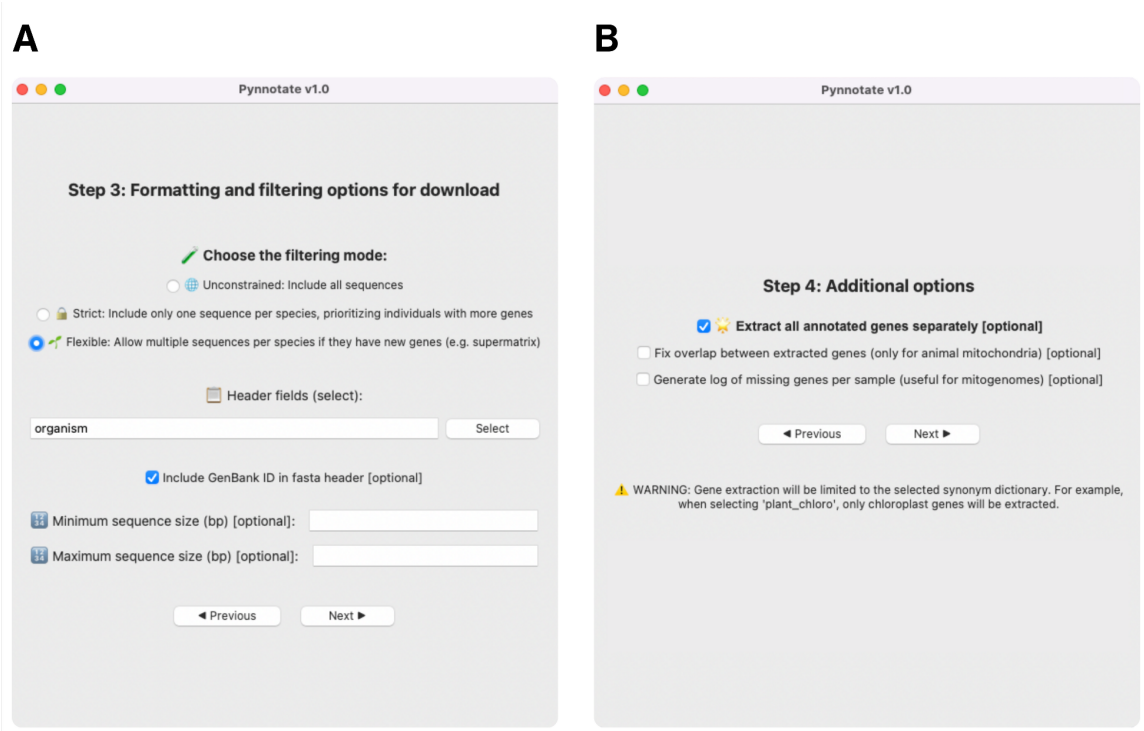

Figure 3. Pynnotate graphic interface. **A.** Step 3: Formatting and filtering options for download. **B.** Step 4: Additional options.

The additional options available in Step 4 control how results are presented (Figure 3B). The first particularly useful option is the extraction of all annotated genes separately, grouping different organisms into respective files for each gene. Importantly, gene extraction will again be limited by the selected synonym dictionary. The next two options allow adjustment of the overlap between extracted mitochondrial genes and the generation of a log documenting missing organisms per sample.

d. Download and output

Finally, Step 5 includes saving and configuration options (Figure 4). First, users must provide their email address to access the NCBI database. Then, users can select the output folder and specify the folder name where pynnotate will save the results. Upon clicking on 'Search and download sequences', several windows are displayed to provide informative

messages to users. When the download begins, a progress bar appears, indicating the download status. Finally, when the results are ready, an information window notifies the user.

Figure 4. Pynnotate graphic interface. Step 5: Final options.

4. Terminal workflow

For users who prefer command-line execution or need to integrate pynnotate into automated pipelines, the tool also provides a terminal-based interface. The terminal version offers the same functionality as the graphical interface but uses a YAML configuration file (example available at https://github.com/fernandacaron/pynnotate/blob/main/pynnotate/examples/config.yaml) to define all parameters, making it suitable for batch processing and reproducible analyses.

a. Configuration file setup

All execution parameters are specified in a YAML configuration file, which centralises the settings and eliminates the need for multiple command-line arguments. The configuration

file includes the same parameters available in the GUI: genome type, search terms or accession numbers, filtering options, output preferences, and file paths. An example configuration file is provided in the repository's examples folder.

b. Execution and output

Once the configuration file is prepared, pynnotate can be executed with a single command:

```
bash pynnotate --config config.yaml
```

The terminal version generates the same output structure as the graphic interface. Progress information and any errors are displayed directly in the terminal, providing immediate feedback during execution. This approach is particularly valuable for high-throughput analyses, integration with other bioinformatics tools, and ensuring reproducibility through version-controlled configuration files.

5. Discussion

Pynnotate fills a gap in the phylogenetic annotation workflow by providing a simple yet flexible tool for extracting and organising gene sequences across multiple taxa. Pynnotate offers both an intuitive graphical interface and a fully scriptable command-line mode, making it accessible to users with or without programming experience. Because it is implemented as a modular Python package, it can be easily incorporated into larger bioinformatic pipelines or expanded by advanced users.

The main strengths of Pynnotate lie in its automated handling of repetitive annotation tasks and its ability to standardise output across heterogeneous GenBank records. Its focus on

batch annotation and feature retrieval makes it highly efficient for common preparatory steps in phylogenetic and comparative genomics studies. Future improvements may include broader support for additional feature types, identifying unannotated GenBank sequences through BLAST, and integration with alignment and phylogenetic inference tools. By combining a graphical interface for teaching environments with a command-line interface for reproducible research workflows, pynnotate bridges the gap between educational use and research-scale dataset assembly. This integration lowers the barrier to dataset preparation and has the potential to streamline early stages of evolutionary and molecular analyses.

6. References

Baykal, P. I., Łabaj, P. P., Markowetz, F., Schriml, L. M., Stekhoven, D. J., Mangul, S., & Beerenwinkel, N. (2024). Genomic reproducibility in the bioinformatics era. *Genome Biology*, *25*(1), 213. https://doi.org/10.1186/s13059-024-03343-2

Borstein, S. R., & O'Meara, B. C. (2018). AnnotationBustR: an R package to extract subsequences from GenBank annotations. *PeerJ*, *6*, e5179. https://doi.org/10.7717/peerj.5179

Cokelaer, T., Cohen-Boulakia, S., & Lemoine, F. (2023). Reprohackathons: Promoting reproducibility in bioinformatics through training. *Bioinformatics*, *39*(Supplement_1), i11–i20. https://doi.org/10.1093/bioinformatics/btad227

Emery, L. R., & Morgan, S. L. (2017). The application of project-based learning in bioinformatics training. *PLOS Computational Biology*, *13*(8), e1005620. https://doi.org/10.1371/journal.pcbi.1005620

Hagen, J. B. (2011). The Origin and Early Reception of Sequence Databases. In M. Hamacher, M. Eisenacher, & C. Stephan (Eds.), *Data Mining in Proteomics* (Vol. 696, pp. 61–77). Humana Press. https://doi.org/10.1007/978-1-60761-987-1_4

315 Harris, B. N., McCarthy, P. C., Wright, A. M., Schutz, H., Boersma, K. S., Shepherd, S. L.,
316 Manning, L. A., Malisch, J. L., & Ellington, R. M. (2020). From panic to pedagogy:
317 Using online active learning to promote inclusive instruction in ecology and
318 evolutionary biology courses and beyond. *Ecology and Evolution*, *10*(22), 12581–
319 12612. https://doi.org/10.1002/ece3.6915
320 Leray, M., Knowlton, N., Ho, S.-L., Nguyen, B. N., & Machida, R. J. (2019). GenBank is a
321 reliable resource for 21st century biodiversity research. *Proceedings of the National*
322 *Academy of Sciences*, *116*(45), 22651–22656.
323 https://doi.org/10.1073/pnas.1911714116
324 Toelch, U., & Ostwald, D. (2018). Digital open science—Teaching digital tools for
325 reproducible and transparent research. *PLOS Biology*, *16*(7), e2006022.
326 https://doi.org/10.1371/journal.pbio.2006022
327