Easy and interactive taxonomic profiling with Metabuli App

SunJae Lee,[1,2,3,*] **Jaebeom Kim**,[2,4,*] **Milot Mirdita**[2] and **Martin Steinegger**[2,4,5,6,†]

[1]Department of Computer Science and Engineering, Seoul National University, Seoul, South Korea, [2]School of Biological Sciences, Seoul National University, Seoul, South Korea, [3]College of Liberal Studies, Seoul National University, Seoul, South Korea, [4]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, South Korea, [5]Institute of Molecular Biology and Genetics, Seoul National University, Seoul, South Korea and [6]Artificial Intelligence Institute, Seoul National University, Seoul, South Korea

*Equal contribution. †To whom correspondence should be addressed.

Abstract

Summary: Accurate metagenomic taxonomic profiling is critical for understanding microbial communities. However, computational analysis often requires command-line proficiency and high-performance computing resources. To lower these barriers, we developed Metabuli App, an all-in-one, stand-alone desktop application for taxonomic analysis. It features user-friendly interfaces for custom database creation, taxonomic profiling, and interactive result visualization.

Availability and implementation: GPLv3 licensed source code and pre-built apps for Windows, macOS, and Linux are available at `https://github.com/steineggerlab/Metabuli-App`. The Sankey-based taxonomy visualization component is available at `https://github.com/steineggerlab/taxoview` for easy integration into other web projects.

Contact: martin.steinegger@snu.ac.kr

Introduction

Metagenomic taxonomic classification allows researchers to understand the taxonomic structure of microbial communities by analyzing environmental DNA. However, it often requires command-line proficiency and high-performance computing resources.

Various tools have been developed to lower these barriers. For example, MEGAN (1) and Pavian (2) provide user-friendly interfaces for metagenomic analysis and visualization. However, MEGAN relies on alignment results, and Pavian requires taxonomic profiling results, both generated by separate software that often demands server-level hardware and command-line expertise.

Web-based tools like CZ ID (3) and BugSeq (4) eliminate these hurdles by hosting servers to perform computations and providing complete analyses. In particular, CZ ID offers free access to metagenomic analysis and pathogen detection, promoting equity in global health studies. BugSeq, while providing detailed reports, operates on a paid-subscription basis. However, neither supports custom databases, often necessary for advanced usage.

Building on prior efforts to improve accessibility in metagenomic analysis, we present the Metabuli App—the first stand-alone, user-friendly desktop application that supports the full suite of functionalities: database curation, taxonomic profiling, and interactive, customizable visualization.

At the core of the app is Metabuli (5, combining "meta" for metagenomic with the Korean "분리," meaning binning), a taxonomic classifier that integrates sensitive homology detection of amino acid-based search and point mutation level resolution of DNA-based analysis. We optimized Metabuli for desktop environments, enabling the app to classify ∼22 million paired-end reads of a human gut metagenome against a database of 36,203 genomes across 8,465 prokaryotic species in just 35 minutes on a MacBook. Additionally, it offers highly customizable visualizations, inspired by Pavian (2), including Sankey (6) and Krona (7) plots. Moreover, it offers interfaces for creating a new database and expanding an existing one with new sequences. By lowering computational barriers, Metabuli App makes advanced metagenomic profiling accessible to a broad range of researchers and practitioners, including those in microbiome research and clinical pathogen detection.

Methods

Metabuli App implementation

Metabuli App is built using the Electron framework (electronjs.org), enabling cross-platform compatibility on Windows, macOS, and Linux operating systems. The front-end utilizes the Vue framework (vuejs.org) for reactive, dynamic user interfaces and D3 (8) to render the interactive Sankey plots and Krona charts. The app takes input parameters required to execute command-line version of Metabuli and transparently processes the data.

Sankey plot generation

A Sankey plot is created by parsing Metabuli's taxonomy report in Kraken format (9). The algorithm hierarchically extracts each taxon along with its rank, proportion, and clade read count. The data is returned in JSON format and visualized using a D3-based (8) Sankey diagram, intuitively displaying taxonomic distribution and hierarchical relationships. To verify the accuracy of the Sankey plot, a taxonomy report is regenerated using the processed data, then compared with the original Kraken report.

In extension, we developed a separate stand-alone Vue.js plugin for this Sankey visualization component, TaxoView (10), allowing integration into other projects such as webservers of Foldseek search (11) and cluster (12).

Filtering and customization. Users can apply minimum proportion, clade count, and most abundant taxa per rank as filtering thresholds. Applying them dynamically updates all visualizations.

Subtree Sankey plot. The user-specified node becomes a root, and all descendant nodes and links are recursively gathered, applying the same filtering criteria used in the main Sankey plot. The filtered set of nodes are then processed again by the D3 component to update the Sankey plot.

New Metabuli commands

Taxon-specific read extraction

The 'extract' command retrieves reads classified to a specified taxon or its descendants. It takes a taxonomy ID, scans the classification results (which retains read order), and collects the line numbers of matching classifications. Reads corresponding to these numbers are then extracted into a separate file.

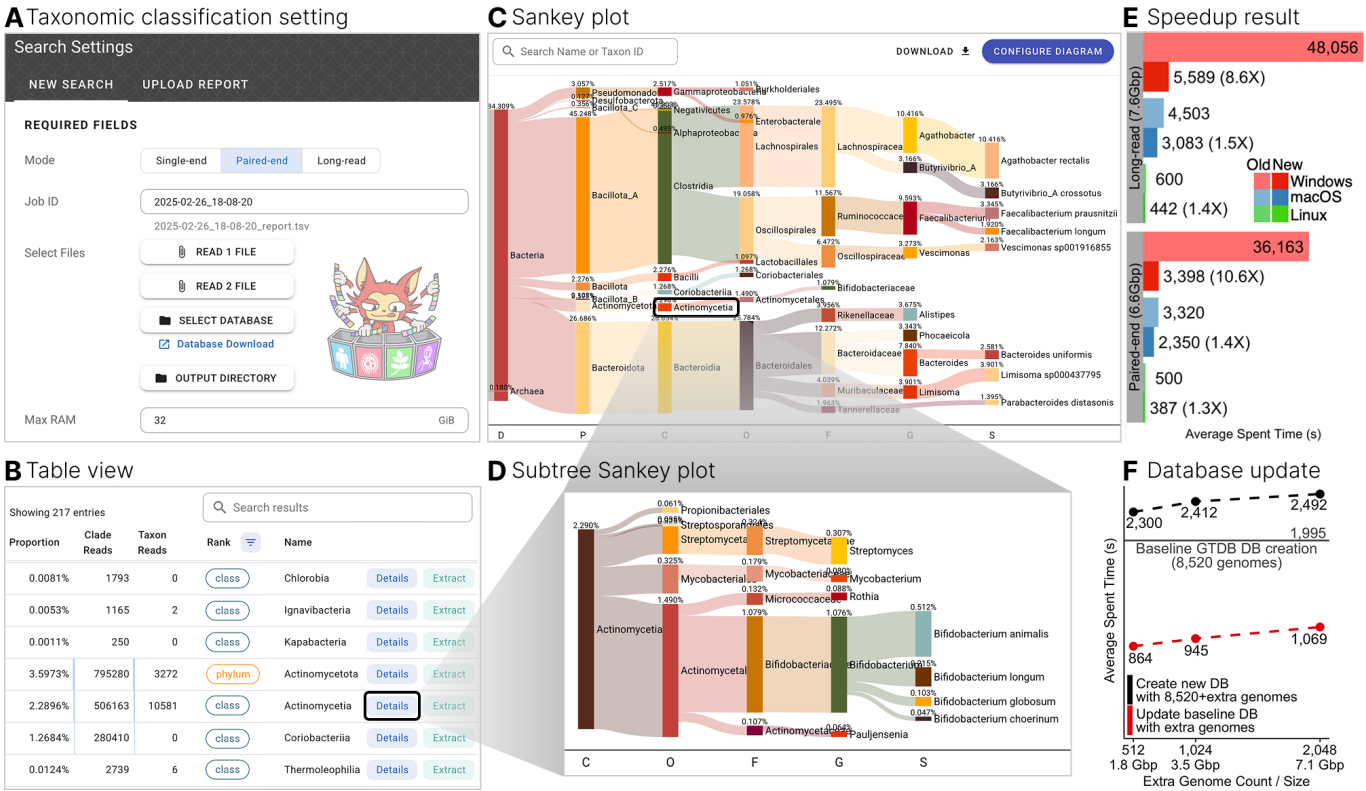

Figure 1. Metabuli App interface and Metabuli optimization results. (A) Classification setting interface. **(B)** Table view of classified proportions and read counts of detected taxa. Only the phylum and class are displayed using rank filtering. **(C)** Sankey plot summarizing taxonomic classification results. **(D)** Subtree Sankey plot of a selected taxon, accessed via the 'Details' button in (B) or a node in (C). **(E)** Classification speedup in v1.1.0 across different platforms. **(F)** Comparison of processing time for updating an existing database with extra genomes versus creating a new one from scratch using both original and extra genomes.

Updating a database with new sequences

The 'updateDB' command computes a sorted k-mer list from new sequences and merges it with that of an existing database, deduplicating identical k-mers from the same species. The merged k-mer list becomes the updated database.

Metabuli optimization

Windows support and optimized memory use

We added Windows support using Cygwin, a POSIX compatibility layer that allows software written for Unix-like systems to run on Windows. However, it introduced overhead, particularly in memory-intensive, parallelized calculations. To mitigate this, we optimized memory use by replacing non-linear associative containers with linear memory structures, reusing allocated memory, and minimizing allocations within parallel blocks.

Faster multi-threaded query k-mer extraction

In the previous version, parallel query k-mer extraction was suboptimal, as read loading and k-mer extraction steps were repeated sequentially, creating synchronization bottlenecks after both steps. Specifically, k-mer extraction began after all threads loaded, and new sequences were loaded after all threads finished extraction. Moreover, one read per thread was processed per iteration, making it even slower.

Now, read loading and k-mer extraction occur concurrently, with an increased granularity of one thousand reads. The main thread continuously identifies idle threads and loads one thousand reads into their pre-allocated memory. Once a thread has its reads loaded, it immediately starts extracting k-mers without waiting for other threads.

Faster k-mer match assembly

Assembling k-mer matches to score each species was the main bottleneck in classification. The process identifies consecutive matches—shifted by one codon with identical overlap—from a k-mer match list sorted by read coordinate. These matches are then assembled by selecting one per codon shift to maximize the score of the covered region.

The previous approach constructed a directed graph, where nodes represented matches and edges linked consecutive matches. Paths from source to terminal nodes were explored using depth-first search to identify the highest-scoring path. The start and end positions of this path, along with its score, were then used to compute the species score. This method required scanning the match list twice—once to build the graph and again to traverse it.

However, recognizing that consecutive matches could be connected independently to earlier ones, we devised a non-graph-based, cache-efficient algorithm that requires only a single scan of the sorted match list. First, matches at n and $n+1$ are identified. For each match at $n+1$, its consecutive matches at n are found, and it is connected to the one with the highest score. Specifically, when connecting two matches, the score of the preceding match is added to the score of the next match's additional codon to form the next match's score, with the start position of the match chain also passed along. As this process continues with $n+1$ and $n+2$, match chains emerge, extend, or terminate. These match chains are equivalent to the highest-scoring paths in the previous approach.

Speed Measurement

The time required for database creation, database update, and classification was measured using Metabuli v1.1.0 on a Windows desktop (Intel i9-9900, 32GB RAM), a MacBook notebook (M2 Pro, 32GB RAM), and a Linux server (64-core AMD EPYC 7742 CPU, 1TB RAM). A database of quality-filtered representative genomes of 8520 species clusters from Genome Taxonomy Database (13, GTDB Release 220; Complete Genome or Chromosome-level assemblies, CheckM completeness > 90% and contamination < 5%) was created and updated with viral genomes of 13253 species of Master Species List 39.4 of International Committee on Taxonomy of Viruses (ICTV) following ICTV's taxonomy (14). Additionally, the same database was updated on the Linux server with 512, 1024, and 2048 extra GTDB genomes meeting the same quality criteria but at the Scaffold-level assembly.

For classification, a GTDB database compatible to both v1.0.6 and v1.1.0 is used to measure the classification speed optimization. The database includes 36,203 genomes across 8,465 prokaryotic species (15; 16). Paired-end (SRR24315757, ∼6.6 Gbp) and long-read (SRR15489014, ∼7.6 Gbp) human gut metagenomes were classified. The average wall-clock time over five measurements is reported, except for the classifications on Windows machine with the Metabuli previous version v1.0.6, which was measured only once due to its long computation time.

Results

Database creation

Users can create a database by providing a list of sequence files, taxonomy dump files, and a mapping from accession to taxonomy ID. Taxonkit (17)-generated taxonomy dump files for Genome Taxonomy Database (13) and International Committee on Taxonomy of Viruses, as well as NCBI Taxonomy (18) dump files, can be utilized. Creating a GTDB database (details in Methods) took 33, 103, 153 minutes on a Linux server, MacBook, and Windows desktop, respectively.

Database sequence addition

New sequences can be added to an existing database by providing a list of sequence files and a mapping from accession to taxonomy ID. This allows users to download a pre-built database containing thousands of genomes and expand it by adding specific sequences of interest. Updating an existing database with extra genomes is also faster than creating a new one from scratch using both the original and extra genomes (Figure 1F).

Moreover, the taxonomy of an existing database can be extended using a different taxonomy, using the '--new-taxa' option. For example, up on the GTDB DB created above, we added genomes of 13253 viral species using ICTV's taxonomy. It took 12, 14, 64 minutes on the same systems above, respectively.

Taxonomic profiling

Users are prompted to select a mode (single-end, paired-end, or long-read) and specify the read file(s), database, and output directory (Figure 1A). Advanced settings allow adjustments to the number of threads, maximum RAM, and a minimum score threshold for classification. The output includes classification results for each read, a Kraken-format report file, and a Krona HTML file.

To measure the classification speed, we downloaded a pre-built GTDB database containing 36,203 genomes across 8,465 prokaryotic species (15; 16). Classifying a paired-end human gut metagenome against it took approximately 6, 35, and 57 minutes on the Linux server, MacBook, and Windows desktop, respectively, while the long-read human gut data required approximately 7, 45, and 93 minutes on the same systems. It is ∼1.3-1.4x, ∼1.6-1.7x, and ∼8.6-10.6x acceleration on the same systems, respectively, compared to the previous version (Figure 1E).

Customizable and interactive result visualization

Upon completion of profiling, the results are visualized as a table, a Sankey plot, and a Krona chart. Users can also upload a previous report to generate a table and a Sankey plot.

Table view. It displays the proportion and number of classified reads for each detected taxon. The results can be filtered by taxonomic rank, sorted by specific columns, and searched for a given taxon (Figure 1B). Moreover, the 'Details' button allows for a focused investigation of the selected taxon (see the 'Detailed View' section below).

Sankey plot. Sankey plots help users intuitively visualize the flow of classified reads across hierarchical taxonomic relationships (Figure 1C). Users can hover over nodes to highlight lineages, search for specific taxa, and click nodes to access the 'Detailed View' (Figure 1D) of a taxon. Customization options include filtering low-abundance taxa by read counts or proportions, limiting the number of taxa per rank, and adjusting visual settings. The plot can be exported in JPG, PNG, or HTML format.

Detailed view. A detailed view, accessible from the Sankey plot or table view, displays a subtree Sankey plot for the selected taxon (Figure 1D). Full customization and download options are also available for this subtree plot. Moreover, when an NCBI taxonomy-based database is used, external links to the corresponding NCBI taxonomy and genome web pages assist users in conducting downstream analysis.

Taxon-specific read extraction

Using the "Extract" button in the table view and detailed view, users can extract reads assigned to a specific taxon at any rank and save them in a separate file.

Conclusion

Metabuli App addresses two key barriers in taxonomic profiling: the need for server-level computing resources and command-line expertise. Its user-friendly graphical interface allows users to create or update customized databases, perform taxonomic profiling, visualize results interactively with Sankey and Krona plots, customize Sankey plots for publication-ready figures, and extract reads classified into specific taxa for downstream analyses—all without requiring command-line expertise. The desktop-optimized Metabuli efficiently classifies real-world metagenomes against a comprehensive whole-genome database on standard computers. We anticipate that Metabuli App will enable a wide range of researchers and practitioners, including those in microbiome research and clinical pathogen detection, to perform advanced metagenomic profiling without requiring command-line proficiency or specialized computing infrastructure.

Acknowledgement

M.S. acknowledges support by the National Research Foundation of Korea grants (2020M3-A9G7-103933, 2021-R1C1-C102065 and 2021-M3A9-I4021220, RS-2024-00396026), Samsung DS research fund, Novo Nordisk Foundation (NNF24SA0092560). Creative-Pioneering Researchers Program and AI-Bio Research Grant through Seoul National University. M.M. acknowledges support from the National Research Foundation of Korea (RS-2023-00250470).

Data availability

Scripts used in benchmarks are available at `https://github.com/jaebeom-kim/metabuli-app-analysis`.

Author contributions

All authors contributed to the conceptualization of the application. S.L. developed the graphical user interface under the guidance of M.M., while J.K. optimized Metabuli's performance. S.L. and J.K. conducted speedup measurements and, along with M.M. and M.S., co-authored the manuscript.

Competing interests

M.S. acknowledges outside interest in Stylus Medicine. The remaining authors declare no competing interests.

References

1. Huson, D. H. *et al.* MEGAN community edition-interactive exploration and analysis of large-scale microbiome sequencing data. *PLoS Computational Biology* **12**, e1004957 (2016).
2. Breitwieser, F. P. & Salzberg, S. L. Pavian: interactive analysis of metagenomics data for microbiome studies and pathogen identification. *Bioinformatics* **36**, 1303–1304 (2020).
3. Simmonds, S. E. *et al.* CZ ID: a cloud-based, no-code platform enabling advanced long read metagenomic analysis. *bioRxiv* 2024.02.29.579666v1 (2024).
4. Fan, J., Huang, S. & Chorlton, S. D. BugSeq: a highly accurate cloud platform for long-read metagenomic analyses. *BMC Bioinformatics* **22**, 160 (2021).
5. Kim, J. & Steinegger, M. Metabuli: sensitive and specific metagenomic classification via joint analysis of amino acid and DNA. *Nature Methods* **21**, 971–973 (2024).
6. Platzer, A. *et al.* BioSankey: Visualization of Microbial Communities Over Time. *Journal of Integrative Bioinformatics* **15**, 20170063 (2018).
7. Ondov, B. D., Bergman, N. H. & Phillippy, A. M. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics* **12**, 385 (2011).
8. Bostock, M., Ogievetsky, V. & Heer, J. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* **17**, 2301–2309 (2011).
9. Wood, D. E. & Salzberg, S. L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology* **15**, R46 (2014).
10. Lee, S. Taxoview (2025). URL `https://doi.org/10.5281/zenodo.14892413`. 10.5281/zenodo.14892413.
11. Van Kempen, M. *et al.* Fast and accurate protein structure search with Foldseek. *Nature Biotechnology* **42**, 243–246 (2024).
12. Barrio-Hernandez, I. *et al.* Clustering predicted structures at the scale of the known protein universe. *Nature* **622**, 637–645 (2023).
13. Parks, D. H. *et al.* GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic Acids Research* **50**, D785–D794 (2022).
14. Walker, P. J. *et al.* Recent changes to virus taxonomy ratified by the International Committee on Taxonomy of Viruses (2022). *Archives of Virology* **167**, 2429–2440 (2022).
15. Kim, J. Metabuli GTDB database part 1 (2024). URL `https://doi.org/10.5281/zenodo.12207057`. 10.5281/zenodo.12207057.
16. Kim, J. Metabuli GTDB database part 2 (2024). URL `https://doi.org/10.5281/zenodo.12207189`. 10.5281/zenodo.12207189.
17. Shen, W. & Ren, H. TaxonKit: A practical and efficient NCBI taxonomy toolkit. *Journal of Genetics and Genomics* **48**, 844–850 (2021).
18. Schoch, C. L. *et al.* NCBI Taxonomy: a comprehensive update on curation, resources and tools. *Database* **2020**, baaa062 (2020).