Brief Report

TaxoFlow: A Step-by-Step Tutorial to Build a Nextflow Pipeline for Metagenomics Taxonomic Classification

Jeferyd Yepes-García and Laurent Falquet *

Posted Date: 29 May 2026

doi: 10.20944/preprints202512.1989.v2

Keywords: bioinformatics; software and workflows; metagenomics; pipeline; nextflow; tutorial

 doi:10.20944/preprints202512.1989.v2

Brief Report

TaxoFlow: A Step-by-Step Tutorial to Build a Nextflow Pipeline for Metagenomics Taxonomic Classification

Jeferyd Yepes-García [1,2] and Laurent Falquet [1,*]

[1] Department of Biology, University of Fribourg, Fribourg, Canton of Fribourg, 1700, Switzerland
[2] Swiss Institute of Bioinformatics, Lausanne, Vaud, 1015, Switzerland
* Correspondence: laurent.falquet@unifr.ch

Abstract

Reproducibility challenges scientific reporting, including metagenomics, where increasingly complex bioinformatics pipelines hinder transparency, comparability, and customization for life science students globally. To address this demanding task, we built an open, interactive and web-based tutorial that guides scholars with basic command-line skills through the detailed development of a validated and reproducible Nextflow metagenomics classification pipeline. As important features, the tutorial emphasizes simplicity, modularity, and containerization, which empowers users with both conceptual understanding and practical implementation skills. Noteworthy, this tutorial provides all the required files, databases, dependencies, software and environment for users to run it without the need of local installation or computational adaptations elsewhere. Finally, by offering a fully reproducible pipeline with a step-by-step developing tutorial, this work aims to lower technical barriers in microbiome bioinformatics and promote best practices in metagenomics data analysis. TaxoFlow is freely available at https://taxoflow.work/.

Keywords: bioinformatics; software and workflows; metagenomics; pipeline; nextflow; tutorial

1. Statement of Need

The constant complexity growth of computational analyses in metagenomics research has brought the necessity of reproducible, scalable and transparent workflows [1]. In this sense, several authors have highlighted the central concern regarding reproducibility across various scientific disciplines as usually the same analysis with the same data yields different results, a situation that has escalated until a denominated reproducibility crisis [2]. The bioinformatics field is especially sensitive to a lack of reproducibility given the wide variety of inconsistent sources such as software versions, parameter settings, or execution environments [3]. Notwithstanding, to mitigate these issues, workflow management systems such as Nextflow [4] or Snakemake [5] have emerged as powerful tools for structuring, automating, and documenting analysis pipelines [6,7]. Specifically, Nextflow enables researchers to design modular, scalable, portable and controlled workflows that are fully reproducible across different computing environments using containers and/or dependency management systems. Being so, not only do these features facilitate scalability and reproducibility, but also enhance transparency, allowing analyses to be easily shared, inspected, and reused [7,8].

On the other hand, metagenomics provides an unprecedented insight into microbial communities without the need for cultivation given the possibility to capture the entire genetic diversity present in a determined environment [9]. Moreover, the strategies to analyze metagenomics data are divided mainly into two categories: *i)* assembly-based approaches, which use the original reads to build longer sequences (contigs) and reconstruct afterwards Metagenome Assembled Genomes (MAGs); and *ii)* read-based taxonomic classification, which classifies sequencing reads directly by leveraging reference or indexed databases [10]. In the case of assembly-based methodologies, they enable detailed genomic and functional characterization of individual microorganisms, albeit demanding substantial

 doi:10.20944/preprints202512.1989.v2

computational resources and high sequencing depth [11]. Meanwhile, taxonomic classification allows a rapid and accurate community composition estimation especially for large-scale study implementation or complex environments given its efficiency in terms of CPU and memory usage [12–14].

Furthermore, there is a broad landscape of tools developed for short-read metagenomics taxonomic classification that employ distinct algorithmic strategies. For instance, Kraken2 [15], a faster and more sensitive version of Kraken [16], classifies reads based on exact k-mer matches to a reference database. Frequently, Kraken2 is complemented with the execution of Bracken [17], a tool that refines the initial classifications by re-estimating species abundances through Bayesian reallocation of ambiguous reads. Consistently, Kraken2 and Bracken rank among the top performers in terms of classification accuracy and runtime efficiency across simulated and real metagenomics datasets [10,18,19].

To promote reproducibility and scalability during metagenomics taxonomic classification, Nextflow-based pipelines that encompass Kraken2 and/or Bracken have been released, including nf-core/taxprofiler [20], kraken-nf [21], wf-metagenomics [22], nxf-kraken2 [23], 16S-Metatranscriptomic-Analysis [24] and specific modules within the Bactopia [25] and nf-core [26] suites. These workflows provide comprehensive implementations of metagenomics taxonomic classification software, integrating multiple classifiers, database management steps, and reporting modules. Nonetheless, such pipelines feature a high number of parameters and an internal complex structure, making them challenging for inexperienced users to understand, modify, or adapt to specific requirements. As a result, this "black box" nature of these robust but sophisticated workflows can hinder learning and flexibility. In addition, although there have been efforts to document training material to perform metagenomics data analysis [27,28] or to develop assembly-based Nextflow pipelines [29], there is a growing need for educational resources that provide the technical knowledge, materials, standardized computing environments and practical implementations to develop reproducible metagenomics-focused pipelines wrapped with workflow managers.

In this context, we created an open, interactive and web-based tutorial (TaxoFlow) that guides learners step-by-step through the creation of a simple yet complete Nextflow metagenomics pipeline. This tutorial is built and extends the reference protocol proposed by Lu et al. (2022) [30] to perform read quality trimming and evaluation, remove host reads, taxonomic classification, species abundance re-estimation and generation of interactive reports in a container-based environment. The tutorial emphasizes conceptual understanding, modular pipeline design, and reproducibility, providing a practical entry point for researchers seeking to build or customize their own workflows.

1.1. Content and Learning Objectives

TaxoFlow offers a hand-on and educational framework for developing a reproducible Nextflow workflow dedicated to metagenomics taxonomic classification. Through this tutorial, we are aiming at teaching both the conceptual foundations and technical implementation of reproducible data analysis, while guiding the students to adapt the pipeline to obtain biologically meaningful results. To achieve this, we use established tools such as FastQC [31], Trim Galore [32], Bowtie2 [33], Kraken2, Bracken, Krona plots [34] and MultiQC [35]. The resource is hosted at https://taxoflow.work/, depicting an interactive, step-by-step learning experience complemented by commented code examples, schematic representations, and example datasets. Being so, TaxoFlow's logic is designed for early-career researchers and students with basic bioinformatics notions seeking to learn how workflow management systems can enhance reproducibility, scalability, and transparency in microbiome-related projects. Noteworthy, TaxoFlow assumes a basic familiarity with command-line interface (CLI) and provides links to complementary educational resources for users who wish to strengthen their background in Linux, Nextflow, or metagenomics data handling.

1.1.1. Educational Scope and Learning Objectives

TaxoFlow presents the process of building a linear pipeline that performs read quality control, host read removal, taxonomic classification and species abundance re-estimation through a progressive learning path to introduce important workflow concepts. This initial exercise establishes a foundation

 doi:10.20944/preprints202512.1989.v2

for understanding process definition, parameterization, and file handling in Nextflow. Afterwards, the users then learn how to manage domain-specific outputs to ensure reproducible downstream analyses.

A central feature of TaxoFlow is the introduction of Nextflow's dataflow paradigm, which enables dynamic parallelization of analyses across multiple samples. In this sense, learners first execute the pipeline for a single dataset and then generalize it to handle multiple input samples simultaneously, a process that depicts how channels and operators manage dependencies and data exchange between processes. This tutorial encompasses important considerations regarding workflow design and implementations [6,7,36,37], and it is complemented with additional sections to demonstrate conditional execution and the use of logical operators to dynamically adapt the pipeline according to user-defined parameters or input availability. Moreover, TaxoFlow shows the modularization of the workflow by separating individual processes into reusable components. Nonetheless, although this modular design pursues nf-core guidelines, its scope is constrained in terms of fully compatibility with existing nf-core modules which limits the re-use of the pipeline's modules in broader community-standardized contexts. Notably, all modules encompassed by TaxoFlow are fully compatible with the pipelines that are part of the official Nextflow training v3.5 documentation.

Further, TaxoFlow illustrates how to integrate custom scripts, maintaining portability through containerized environments. As a result, the tutorial promotes the adoption of FAIR principles for research software [38], and adheres to general recommendations to organize computational biology projects [39,40].

2. Implementation

2.1. Instructional Design

2.1.1. Part 1 - Pipeline

The workflow implemented during the tutorial development is presented in **Figure 1a**. The example dataset used in the tutorial consists of paired-end reads recovered from an oligotrophic, phosphorus-deficient pond in Cuatro Ciénegas, Mexico [41]. Before building an executing the pipeline, TaxoFlow guides users to download an indexed genome of *Arabidopsis thaliana* required by Bowtie2 (indexes maintained by Langmead Lab), and provides instructions to retrieve a custom database with 54 bacterial species for Kraken2 and Bracken. Noteworthy, this default execution using a mock database and a defined indexed genome aims at demonstrating TaxoFlow's educational purpose only, with no real biological interpretation of the outcomes if run on environmental or clinical data sets. In consequence, at the introductory section of the tutorial, the users are advised to not run the pipeline in its default state for real-world data set unless they include the proper resources. Thus, they are provided with links to ready-to-use databases and indexed genomes, as well as specific tips, to adapt the pipeline accordingly.

The workflow then takes as input raw FASTQ files from one or multiple metagenomics samples to perform an initial quality assessment through FasTQC (v0.12.1), as well as read trimming and adapter removal with Trim Galore (v0.6.10); quality is re-evaluated after these steps. The pipeline then continues with host read removal by alignment against the downloaded reference genome using Bowtie2 (v2.5.4). Afterwards, the filtered and cleaned reads are subjected to taxonomic classification with Kraken2 (v2.14), followed by species abundance re-estimation using Bracken (v3.1), producing refined taxonomic profiles for each sample. Later, the resulting Bracken reports are visualized through Krona plots (v2.8.1). The workflow is bifurcated if multiple samples are used as input to automatically concatenate Bracken outputs and convert them into a Biological Observation Matrix (BIOM) file (Kraken-BIOM v1.2.0 [42]), which is subsequently imported into R as a Phyloseq object (v1.50.0 [43]) for the generation of a diversity report. This is achieved through the execution of a custom script that generates a self-contained HTML report with basic exploratory summaries of the community, including α-diversity, Bray–Curtis–based β-diversity (PCoA), and simple co-occurrence networks at the genus level. These outputs are intended solely for visualization and qualitative inspection of patterns in the data; the users are prompted to review additional resources to perform proper metagenomics statistical

 doi:10.20944/preprints202512.1989.v2

analyses. The workflow ends with the aggregation of the outputs from FastQC, Trim Galore, Bowtie2 and Kraken2 into a comprehensive MultiQC report.

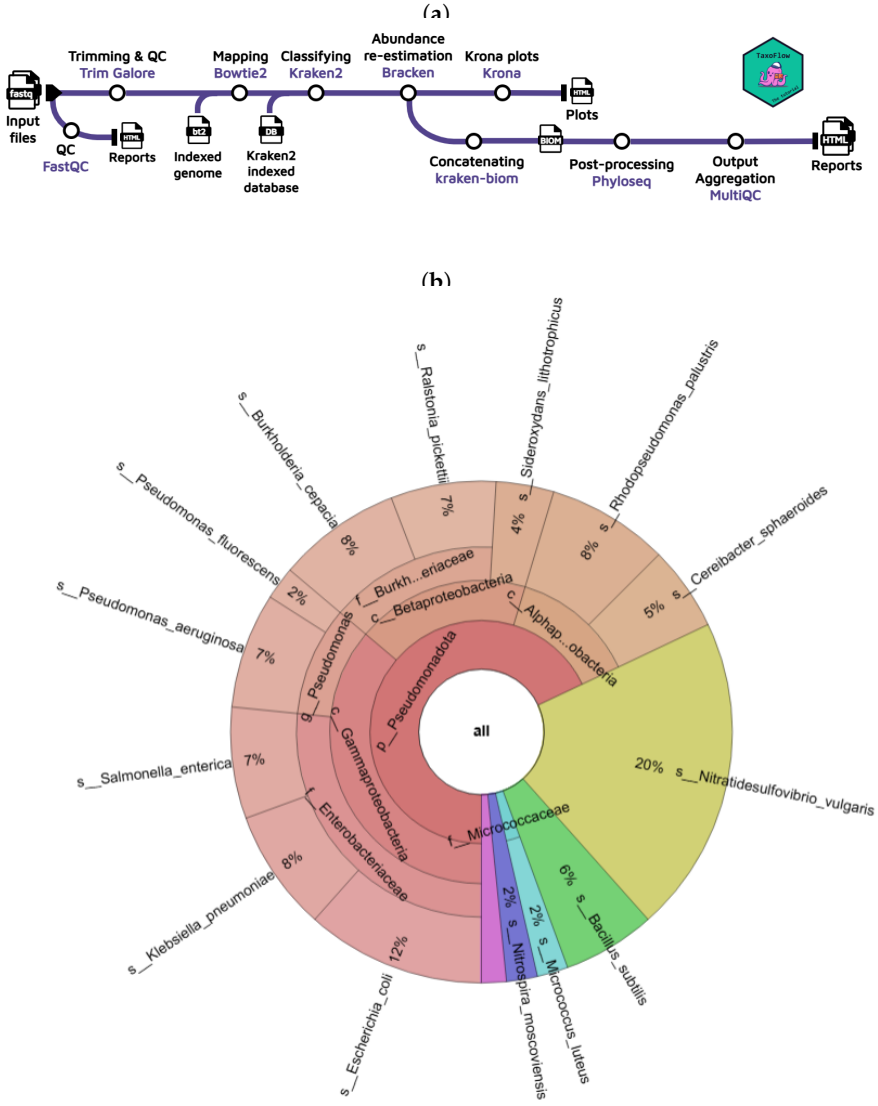

Figure 1. Workflow and single-sample result obtained by Taxoflow. **(a)** Schematic representation of the educational pipeline built through TaxoFlow. **(b)** Snapshot of the Krona plot obtained through the single-sample execution of the pipeline during the Part 2 of the tutorial. **Results not biologically meaningful**.

Furthermore, the pipeline execution relies on Seqera container images pulled and launched by Docker. To achieve maximum portability, TaxoFlow features an Environment section, where users can find detailed instructions to execute the pipeline using local Dev Containers, adapting the workflow to High Performance Computing (HPC) clusters, and even how to use CodeSandbox as an alternative for Codespaces. The HPC section provides instructions to generate Singularity/Apptainer containers to execute the pipeline in HPC environments.

2.1.2. Part 2 - Single Sample

This section introduces the essentials of workflow design in Nextflow by walking learners through the construction of a metagenomics pipeline. The learners are presented with core concepts, including processes, channels, configuration files, and the dataflow programming model, all framed within the minimal computational environment required to run the pipeline; these concepts are explained through hands-on development, where users build the functional workflow step by step. The building

 doi:10.20944/preprints202512.1989.v2

process includes defining modules for each process, centralizing the execution in a single workflow file that handles input channels for raw FASTQ files, databases and connection among processes. Each component is intentionally constructed incrementally to clarify how data moves among tasks, and how scripts run inside processes. Here, good workflow engineering practices are also emphasized by encouraging learners to modularize their code, name processes consistently, and adopt conventions inspired by nf-core.

The section concludes with guidance on running the workflow for a single sample, ensuring that learners understand not only how the workflow functions but also how to adapt it. **Figure 1b** shows an example of the resulting output after single-sample execution of the developed pipeline.

2.1.3. Part 3 - Multi-Sample

This part expands on the foundational skills developed in Part 2 by teaching learners how to scale the workflow to handle multiple samples and generate integrated metagenomics outputs. This section begins by explaining how the dataflow paradigm allows workflows to process many samples automatically and in parallel, without manually iterating through files. Learners modify their input definitions so that the workflow recognizes and processes an arbitrary number of FASTQ files. We also introduced more advanced workflow-control features, such as implementing conditional execution paths, using operators to coordinate outputs from different processes, and structuring pipeline logic to accommodate both optional and mandatory steps. Moreover, TaxoFlow shows how to perform pipeline enhancements by including the concatenation individual taxonomic reports into an abundance matrix, converting the matrix to BIOM format, generating a Phyloseq object, and producing a suite of basic ecological analyses such as α- and β-diversity metrics (**Figure 2**); users are warned regarding the biological and statistical interpretation of this demonstrative execution. The workflow ultimately produces automated HTML reports that summarize the results for each input sample, illustrating how Nextflow can orchestrate complete end-to-end analyses. Alongside technical expansion, this sections aims at reinforcing best practices, even with the incorporation of custom scripts, while maintaining portability and reproducibility.

3. Validation

The workflow proposed in this tutorial has been validated by Lu et al. (2022) [30]. In addition, we used TaxoFlow to analyze the sequences (SRR32316197) belonging to the mock community (Zymo-BIOMICS™ Microbial Community DNA Standard D6305) in combination with the Kraken2 database PlusPFP (Standard plus Refseq protozoa, fungi & plant) v04/09/2024. This analysis shows the correct estimation of the community member proportions according to the manufacturer specifications. The resulting files of this analysis were deposited on Zenodo under the identifier 16947911 and with the filename *mock_community.tar.gz*.

4. Conclusions

TaxoFlow serves both as an educational resource and a functional analytical tool, enabling scalable analysis from raw reads to ecological interpretation. The pipeline obtained through this tutorial facilitates reproducibility and provides an accessible entry point into the design principles of community standards such as nf-core guidelines. Likewise, the resulting workflow offers a lightweight, transparent, and customizable alternative for researchers who wish to understand or adapt taxonomic profiling pipelines from the ground up, while adhering to best practices in reproducible computational metagenomics. Finally, the tutorial demonstrates how accessible and well documented pipelines can bridge the gap between learning and research, empowering users to develop and adapt reproducible bioinformatics tools for diverse metagenomics applications.

5. Availability of Source Code and Requirements

- Project name: TaxoFlow

 doi:10.20944/preprints202512.1989.v2

- Project home page: https://taxoflow.work/
- Source code: https://github.com/jeffe107/taxoflow_tutorial.
- Archived version: TaxoFlow v2.0 on Zenodo: doi.org/10.5281/zenodo.20398657
- Platform: GitHub Codespaces or CodeSandbox.
- Programming language: Nextflow v25.10.4 (pinned version across environment to ensure portability and compatibility).
- Licenses: CC BY-NC-SA 4.0. This license covers both the tutorial and the derived pipeline. Specific licenses per tool used by Taxoflow are included on Table 1.
- Other requirements: Modern browser with Internet access. The software versions and container images are presented on **Table 1**.

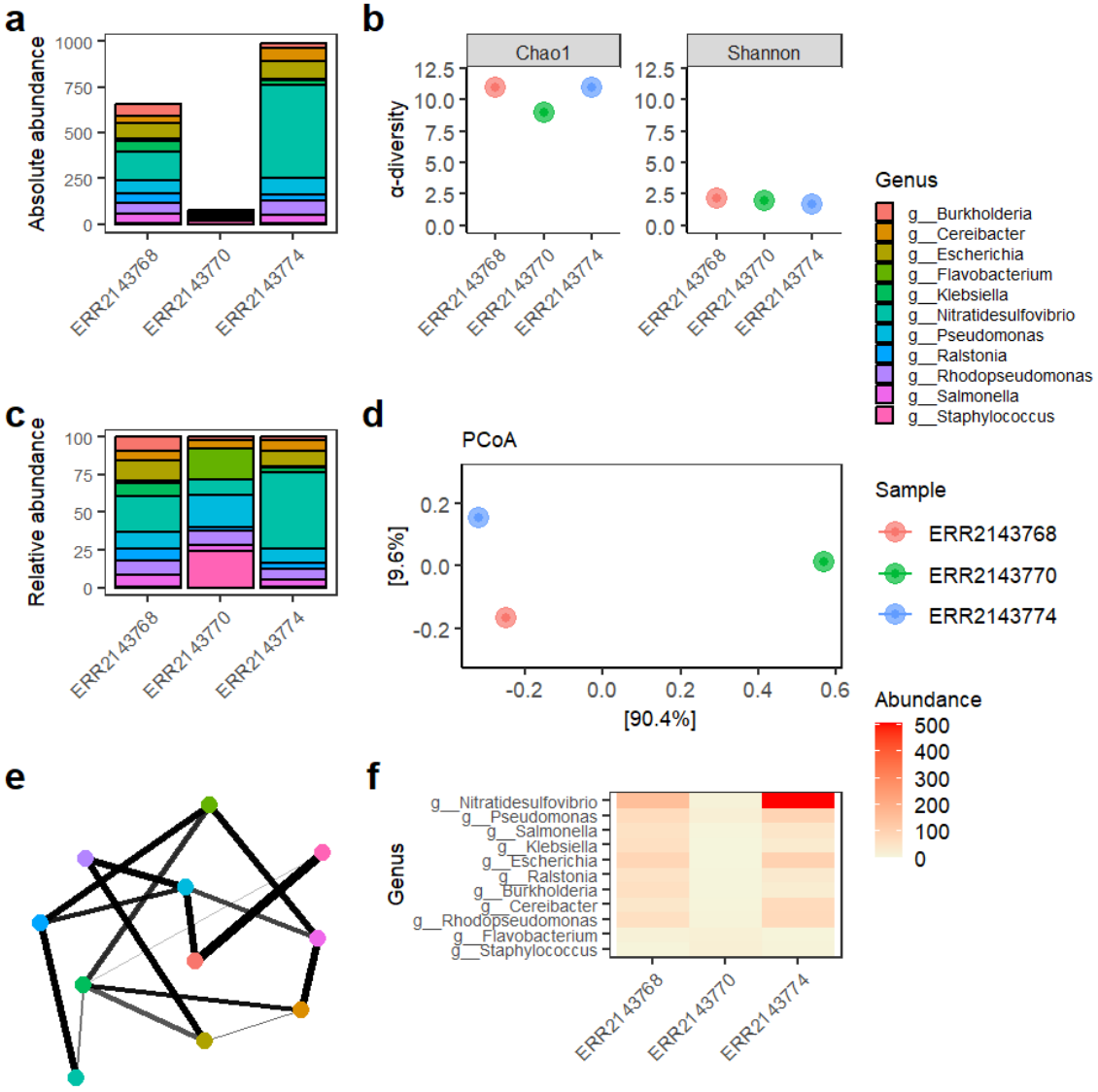

Figure 2. Taxonomic composition and diversity analyses of environmental samples using TaxoFlow. Absolute (**a**) and relative (**c**) abundance plots show the distribution of dominant genera across samples. α-diversity (**b**) is estimated using Chao1 and Shannon indices. β-diversity (**d**) is assessed by Principal Coordinates Analysis (PCoA) using Bray–Curtis distance. A co-occurrence network (**e**) shows relationships among genera based on Bray–Curtis dissimilarity (maxdist = 0.9). A heatmap (**f**) displays genus abundance patterns across samples, ordered according to Bray–Curtis dissimilarity and PCoA. Low-abundance genera (<3% mean relative abundance) are removed before diversity and network analyses. **Results not biologically meaningful**.

 doi:10.20944/preprints202512.1989.v2

Table 1. Specific package and software versions used by TaxoFlow. The Seqera container images are also included.

Tool	Version	Container* (Seqera**)	License
FastQC [31]	0.12.1	trim-galore:0.6.10–1bf8ca4e1967cd18	GPL-3.0
Trim Galore [32]	0.6.10	trim-galore:0.6.10–1bf8ca4e1967cd18	GPL-3.0
Bowtie2 [33]	2.5.4	bowtie2:2.5.4–d51920539234bea7	GPL-3.0
Kraken2 [15]	2.14	kraken2:2.14–83aa57048e304f01	MIT
Bracken [17]	3.1	bracken:3.1–22a4e66ce04c5e01	GPL-3.0
KrakenTools [30]	1.2	krakentools:1.2–db94e0b19cfa397b	GPL-3.0
Krona [34]	2.8.1	krona:2.8.1–2f750080982f027e	BSD
Kraken-BIOM [42]	1.2.0	kraken-biom:1.2.0–f040ab91c9691136	MIT
MultiQC [35]	1.33	pip_multiqc:a3c26f6199d64b7c	GPL-3.0
R	4.4.3	bioconductor-phyloseq_knit_r-base_r-ggplot2_r-rmdformats:6efceb52eb05eb44	GPL-2 | GPL-3
ggplot2 [44]	3.5.1	bioconductor-phyloseq_knit_r-base_r-ggplot2_r-rmdformats:6efceb52eb05eb44	MIT
Phyloseq [43]	1.50.0	bioconductor-phyloseq_knit_r-base_r-ggplot2_r-rmdformats:6efceb52eb05eb44	AGPL-3.0

*Containers found at: *community.wave.seqera.io/library/*. **Containers compatible with Docker engine.

Author Contributions: JYG: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. LF: Conceptualization, Funding acquisition, Project administration, Supervision, Resources, Writing – review & editing.

Funding: This work was supported by the Centenary Research Fund of the University of Fribourg (FC-22-912 0857), and by the *Fondation de Recherche en Biochimie*, Epalinges, Switzerland.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The required Kraken2/Bracken database for the tutorial can be retrieved from Zenodo under identifier: 17708950. The sequences reads used as input for the demonstrative execution of TaxoFlow are found at the NCBI Sequence Read Archive (SRA) with run accession numbers ERR2143768, ERR2143770 and ERR2143774, covered by the BioProject PRJEB22811.

Acknowledgments: JYG specially thanks the Federal Commission for Scholarships for Foreign Students (FCS) for their support through the Swiss Government Excellence Scholarship. We also acknowledge Geraldine Van der Auwera from the Nextflow Training Team for her valuable contribution to conceive the idea of the tutorial and for her insightful feedback to implement it.

Conflicts of Interest: The author(s) declare no competing interests.

Abbreviations

MAG, Metagenome-Assembled Genome; CLI, command-line interface; FAIR, Findable, Accessible, Interoperable, Reusable; BIOM, Biological Observation Matrix; HPC, High Perfomance Computing.

References

1. Kim, N.; Ma, J.; Kim, W.; Kim, J.; Belenky, P.; Lee, I. Genome-resolved metagenomics: a game changer for microbiome medicine. *Experimental & Molecular Medicine* **2024**, *56*, 1501–1512. https://doi.org/10.1038/s12276-024-01262-7.
2. Baker, M. 1, 500 scientists lift the lid on reproducibility. *Nature* **2016**, *533*, 452–454. https://doi.org/10.1038/533452a.
3. Yang, C.; Chowdhury, D.; Zhang, Z.; Cheung, W.K.; Lu, A.; Bian, Z.; Zhang, L. A review of computational tools for generating metagenome-assembled genomes from metagenomic sequencing data. *Computational and Structural Biotechnology Journal* **2021**, *19*, 6301–6314. https://doi.org/10.1016/j.csbj.2021.11.028.
4. Langer, B.E.; Amaral, A.; Baudement, M.O.; Bonath, F.; Charles, M.; Chitneedi, P.K.; Clark, E.L.; Di Tommaso, P.; Djebali, S.; Ewels, P.A.; et al. Empowering bioinformatics communities with Nextflow and nf-core. *Genome Biology* **2025**, *26*. https://doi.org/10.1186/s13059-025-03673-9.

 doi:10.20944/preprints202512.1989.v2

5. Mölder, F.; Jablonski, K.P.; Letcher, B.; Hall, M.B.; van Dyken, P.C.; Tomkins-Tinch, C.H.; Sochat, V.; Forster, J.; Vieira, F.G.; Meesters, C.; et al. Sustainable data analysis with Snakemake. *F1000Research* **2025**, *10*, 33. https://doi.org/10.12688/f1000research.29032.3.
6. Wratten, L.; Wilm, A.; Göke, J. Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers. *Nature Methods* **2021**, *18*, 1161–1168. https://doi.org/10.1038/s41592-021-01254-9.
7. Roach, M.J.; Pierce-Ward, N.T.; Suchecki, R.; Mallawaarachchi, V.; Papudeshi, B.; Handley, S.A.; Brown, C.T.; Watson-Haigh, N.S.; Edwards, R.A. Ten simple rules and a template for creating workflows-as-applications. *PLOS Computational Biology* **2022**, *18*, e1010705. https://doi.org/10.1371/journal.pcbi.1010705.
8. Ahmed, A.E.; Allen, J.M.; Bhat, T.; Burra, P.; Fliege, C.E.; Hart, S.N.; Heldenbrand, J.R.; Hudson, M.E.; Istanto, D.D.; Kalmbach, M.T.; et al. Design considerations for workflow management systems use in production genomics research and the clinic. *Scientific Reports* **2021**, *11*. https://doi.org/10.1038/s41598-021-99288-8.
9. Navgire, G.S.; Goel, N.; Sawhney, G.; Sharma, M.; Kaushik, P.; Mohanta, Y.K.; Mohanta, T.K.; Al-Harrasi, A. Analysis and Interpretation of metagenomics data: an approach. *Biological Procedures Online* **2022**, *24*. https://doi.org/10.1186/s12575-022-00179-7.
10. Edwin, N.R.; Fitzpatrick, A.H.; Brennan, F.; Abram, F.; O'Sullivan, O. An in-depth evaluation of metagenomic classifiers for soil microbiomes. *Environmental Microbiome* **2024**, *19*. https://doi.org/10.1186/s40793-024-00561-w.
11. Wajid, B.; Anwar, F.; Wajid, I.; Nisar, H.; Meraj, S.; Zafar, A.; Al-Shawaqfeh, M.K.; Ekti, A.R.; Khatoon, A.; Suchodolski, J.S. Music of metagenomics—a review of its applications, analysis pipeline, and associated tools. *Functional & Integrative Genomics* **2021**, *22*, 3–26. https://doi.org/10.1007/s10142-021-00810-y.
12. Quince, C.; Walker, A.W.; Simpson, J.T.; Loman, N.J.; Segata, N. Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology* **2017**, *35*, 833–844. https://doi.org/10.1038/nbt.3935.
13. Liu, Y.X.; Qin, Y.; Chen, T.; Lu, M.; Qian, X.; Guo, X.; Bai, Y. A practical guide to amplicon and metagenomic analysis of microbiome data. *Protein & Cell* **2020**, *12*, 315–330. https://doi.org/10.1007/s13238-020-00724-8.
14. Yepes-García, J.; Falquet, L. 2Pipe starts with a question: matching you with the correct pipeline for MAG reconstruction. *mSystems* **2026**, *0:e00844-25*. https://doi.org/10.1128/msystems.00844-25.
15. Wood, D.E.; Lu, J.; Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biology* **2019**, *20*, 1–13. https://doi.org/10.1186/s13059-019-1891-0.
16. Wood, D.E.; Salzberg, S.L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology* **2014**, *15*. https://doi.org/10.1186/gb-2014-15-3-r46.
17. Lu, J.; Breitwieser, F.P.; Thielen, P.; Salzberg, S.L. Bracken: Estimating species abundance in metagenomics data. *PeerJ Computer Science* **2017**, p. e104. https://doi.org/10.7717/PEERJ-CS.104/SUPP-5.
18. Timilsina, M.; Chundru, D.; Pradhan, A.K.; Blaustein, R.A.; Ghanem, M. Benchmarking Metagenomic Pipelines for the Detection of Foodborne Pathogens in Simulated Microbial Communities. *Journal of Food Protection* **2025**, *88*, 100583. https://doi.org/10.1016/j.jfp.2025.100583.
19. Pusadkar, V.; Azad, R.K. Benchmarking Metagenomic Classifiers on Simulated Ancient and Modern Metagenomic Data. *Microorganisms* **2023**, *11*, 2478. https://doi.org/10.3390/microorganisms11102478.
20. Stamouli, S.; Beber, M.E.; Normark, T.; Christensen, T.A.; Andersson-Li, L.; Borry, M.; Jamy, M.; Community, N.C.; Yates, J.A.F. nf-core/taxprofiler: highly parallelised and flexible pipeline for metagenomic taxonomic classification and profiling, 2023. https://doi.org/10.1101/2023.10.20.563221.
21. Borry, M. kraken-nf, 2019. Accessed 2025-12-17.
22. EPI2ME. wf-metagenomics, 2021. Accessed 2025-12-17.
23. Angelov, A. nxf-kraken2, 2020. Accessed 2025-12-17.
24. Terrón-Camero, L.C.; Gordillo-González, F.; Salas-Espejo, E.; Andrés-León, E. Comparison of Metagenomics and Metatranscriptomics Tools: A Guide to Making the Right Choice. *Genes* **2022**, *13*, 2280. https://doi.org/10.3390/genes13122280.
25. Petit, R.A.; Read, T.D. Bactopia: a Flexible Pipeline for Complete Analysis of Bacterial Genomes. *mSystems* **2020**, *5*. https://doi.org/10.1128/msystems.00190-20.
26. Ewels, P.A.; Peltzer, A.; Fillinger, S.; Patel, H.; Alneberg, J.; Wilm, A.; Garcia, M.U.; Tommaso, P.D.; Nahnsen, S. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology* **2020**, *38*, 276–278. https://doi.org/10.1038/s41587-020-0439-x.
27. Kruchten, A.E. A Curricular Bioinformatics Approach to Teaching Undergraduates to Analyze Metagenomic Datasets Using R. *Frontiers in Microbiology* **2020**, *11*. https://doi.org/10.3389/fmicb.2020.578600.

 doi:10.20944/preprints202512.1989.v2

28. Zirión-Martínez, C.; Garfias-Gallegos, D.; Arellano-Fernandez, T.V.; Espinosa-Jaime, A.; Bustos-Díaz, E.D.; Lovaco-Flores, J.A.; Tejero-Gómez, L.G.; Avelar-Rivas, J.A.; Sélem-Mojica, N. A Data Carpentry- Style Metagenomics Workshop. *Journal of Open Source Education* **2024**, *7*, 209. https://doi.org/10.21105/jose.00209.
29. Telatin, A. nextflow-example, 2022. Accessed 2025-12-17.
30. Lu, J.; Rincon, N.; Wood, D.E.; Breitwieser, F.P.; Pockrandt, C.; Langmead, B.; Salzberg, S.L.; Steinegger, M. Metagenome analysis using the Kraken software suite. *Nature Protocols* **2022**, *17*, 2815–2839. https://doi.org/10.1038/s41596-022-00738-y.
31. Andrews, S. FASTQC. A quality control tool for high throughput sequence data, 2010.
32. Krueger F, Trim Galore. Zenodo; 2026. https://www.trimgalore.com.
33. Langmead, B.; Salzberg, S.L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **2012**, *9*, 357–359. https://doi.org/10.1038/nmeth.1923.
34. Ondov, B.D.; Bergman, N.H.; Phillippy, A.M. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics* **2011**, *12*, 1–10. https://doi.org/10.1186/1471-2105-12-385.
35. Ewels, P.; Magnusson, M.; Lundin, S.; Käller, M. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **2016**, *32*, 3047. https://doi.org/10.1093/bioinformatics/btw354.
36. Kadri, S.; Sboner, A.; Sigaras, A.; Roy, S. Containers in Bioinformatics: Applications, Practical Considerations, and Best Practices in Molecular Pathology. *The Journal of Molecular Diagnostics* **2022**, *24*, 442–454. https://doi.org/10.1016/j.jmoldx.2022.01.006.
37. Jackson, M.; Kavoussanakis, K.; Wallace, E.W.J. Using prototyping to choose a bioinformatics workflow management system. *PLOS Computational Biology* **2021**, *17*, e1008622. https://doi.org/10.1371/journal.pcbi.1008622.
38. Barker, M.; Chue Hong, N.P.; Katz, D.S.; Lamprecht, A.L.; Martinez-Ortiz, C.; Psomopoulos, F.; Harrow, J.; Castro, L.J.; Gruenpeter, M.; Martinez, P.A.; et al. Introducing the FAIR Principles for research software. *Scientific Data* **2022**, *9*, 622. https://doi.org/10.1038/s41597-022-01710-x.
39. Noble, W.S. A Quick Guide to Organizing Computational Biology Projects. *PLOS Computational Biology* **2009**, *5*, e1000424. https://doi.org/10.1371/journal.pcbi.1000424.
40. Noble, W.S. Ten simple rules for defining a computational biology project. *PLOS Computational Biology* **2023**, *19*, e1010786. https://doi.org/10.1371/journal.pcbi.1010786.
41. Okie, J.G.; Poret-Peterson, A.T.; Lee, Z.M.; Richter, A.; Alcaraz, L.D.; Eguiarte, L.E.; Siefert, J.L.; Souza, V.; Dupont, C.L.; Elser, J.J. Genomic adaptations in information processing underpin trophic strategy in a whole-ecosystem nutrient enrichment experiment. *eLife* **2020**, *9*, e49816. https://doi.org/10.7554/eLife.49816.
42. McDonald, D.; Clemente, J.C.; Kuczynski, J.; Rideout, J.R.; Stombaugh, J.; Wendel, D.; Wilke, A.; Huse, S.; Hufnagle, J.; Meyer, F.; et al. The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. *GigaScience* **2012**, *1*, 2047–217X–1–7. https://doi.org/10.1186/2047-217X-1-7.
43. McMurdie, P.J.; Holmes, S. phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. *PLOS ONE* **2013**, *8*, e61217. https://doi.org/10.1371/journal.pone.0061217.
44. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*; Springer-Verlag New York, 2016.