nf-core/taxprofiler: highly parallelised and flexible pipeline for metagenomic taxonomic classification and profiling

Sofia Stamouli[1], Moritz E. Beber[2], Tanja Normark[3], Thomas A. Christensen II[4], Lili Andersson-Li[5], Maxime Borry[6], Mahwash Jamy[7], nf-core community[8], James A. Fellows Yates[9]

[1]Department of Microbiology, Tumor and Cell Biology, Karolinkska Institutet
[1]Department of Clinical Microbiology, Karolinska University Hospital
[2]Unseen Bio ApS
[3]Department of Microbiology, Tumor and Cell Biology, Karolinkska Institutet
[3]Department of Clinical Microbiology, Karolinska University Hospital
[4]Veterinary Diagnostic Laboratory, Kansas State University College of Veterinary Medicine
[5]Department of Microbiology, Tumor and Cell Biology, Karolinkska Institutet
[5]Department of Clinical Microbiology, Karolinska University Hospital
[6]Department of Archaeogenetics, Max Planck Institute for Evolutionary Anthropology
[7]Department of Microbiology, Tumor and Cell Biology, Karolinkska Institutet
[7]Department of Clinical Microbiology, Karolinska University Hospital
[8]
[9]Department of Archaeogenetics, Max Planck Institute for Evolutionary Anthropology
[9]Department of Paleobiotechnology, Leibniz Institute for Natural Product Research and Infection Biology Hans Knöll Institute

1 Abstract

Metagenomic classification tackles the problem of characterising the taxonomic source of all DNA sequencing reads in a sample. A common approach to address the differences and biases between the many different taxonomic classification tools is to run metagenomic data through multiple classification tools and databases. This, however, is a very time-consuming task when performed manually - particularly when combined with the appropriate preprocessing of sequencing reads before the classification.

Here we present nf-core/taxprofiler, a highly parallelised taxonomic classification and processing pipeline that allows for automated and simultaneous classification and/or profiling of both short- and long-read metagenomic sequencing libraries against a large number of taxonomic classifiers and profilers as well as databases within a single pipeline run. Implemented in Nextflow and as part of the nf-core initiative, the pipeline benefits from high levels of scalability and portability, allowing for large and

small projects on a wide range of computing infrastructure, as well as best-practise software development and community support to ensure longevity and adaptability of the pipeline, keeping up with the field of metagenomics.

2 Introduction

Shotgun metagenomics offers strong benefits to the taxonomic classification of DNA samples over targeted approaches (Eloe-Fadrosh et al. 2016; Florian P. Breitwieser, Lu, and Salzberg 2019). While metabarcoding approaches targeting the 16S rRNA or other marker genes are widely due to low cost and large, diverse reference databases (Yilmaz et al. 2014; Lynch and Neufeld 2015), metagenomic approaches have been gaining popularity with the increasingly lower costs of shotgun sequencing. These metagenomic analyses have been shown to provide similar resolution on microbial genomes during taxonomic classification (Hillmann et al. 2018), with the added benefit of having greater reusability potential of the data, via whole genome reconstruction and also functional classification of metagenomics (Sharpton 2014; Quince et al. 2017).

Taxonomic classifiers (sometimes referred to as taxonomic binners) consists of identifying the original 'taxonomic source' of a given DNA sequence (Ye et al. 2019; Meyer et al. 2022; Govender and Eyre 2022). In metagenomics, this typically consists of comparing millions of DNA sequences against hundreds or thousands of reference genomes either via alignment or 'k-mer matching' (Sharpton 2014; Sun et al. 2021), with the most close match being considered the most likely original 'source' organism of that sequence. Taxonomic profilers additionally will also try to infer species abundance of the organism in the original sample, based on the sequence abundance (Nayfach and Pollard 2016). We will use classifiers and profilers interchangeably throughout the publication.

Due to the scale of the problem, taxonomic profiling remains an 'unresolved problem' in bioinformatics. Having to identify the original source of many sequences out of many reference genomes, but in an *efficient* manner, is understandably a difficult problem. Therefore a plethora of tools have been developed to address this challenge, all with their own biases and specific contexts (Sczyrba et al. 2017; Meyer et al. 2022). Additionally, each tool often produces tool-specific output formats making it difficult to efficiently cross compare results. Thus, no established 'gold standard' method currently exists.

One solution to address the range of different tools is to run all of them in parallel, and cross compare the results. This can be useful both for benchmarking studies (e.g. Sczyrba et al. 2017; Meyer et al. 2022), but also to build consensus profiles whereby confidence of a particular taxonomic identification can be increased when it is detected by multiple tools (McIntyre et al. 2017; Ye et al. 2019).

A second challenge in taxonomic classification is a question of databases. As with tools, there is no one set 'gold standard' database. Different questions and contexts may require different databases, such as when a researcher wants to search for both bacterial and virus species in samples, and as an extension of this, taxonomic classifiers may need different settings for each database. Furthermore, as genomic sequencing becomes cheaper and more efficient, the number of publicly available reference genomes are rapidly increasing (Nasko et al. 2018), making the size of databases taxonomic classifiers also much larger and often outpacing the computational capacity available to researchers. In fact, while this was one of the main motivations behind classifiers such as Kraken2 (Wood, Lu, and Langmead 2019), these algorithmic techniques are already becoming insufficient (Wright, Comeau, and Langille 2023).

Finally, with the decrease of costs, the possibility for larger and larger metagenome sequencing datasets increases, leading to increasing sample sizes in studies, as exemplified by the doubling of the number of metagenomes on the European Bioinformatic Institute's MGnify database in two years (Mitchell et al. 2019). Altogether this highlights the need for methods to efficiently profile many samples using many tools. Manually setting up bioinformatic jobs for classification tasks for each database and settings against different tools on traditional academic computing infrastructure (e.g. high performance computing clusters or 'HPC' clusters) can be very tedious. Additionally, particularly for very large sample sets, there is increasing use of cloud platforms that have greater scalability than traditional HPCs. Being able to reliably and reproducibly execute taxonomic classification tasks across infrastructure with minimal intervention would therefore be a boon for the metagenomics field.

Here we present nf-core/taxprofiler, a pipeline designed to allow users to efficiently and simultaneously taxonomically classify and profile short- and long-read sequencing data against multiple tools and databases in a single pipeline run. nf-core/taxprofiler utilises Nextflow (Di Tommaso et al. 2017) to ensure efficiency, portability, and scalability, and has been developed within the nf-core initiative of Nextflow pipelines (Ewels et al. 2020) to ensure high quality coding practises and user accessibility, including detailed documentation and a graphical-user-interface (GUI) execution interface.

3 Implementation

nf-core/taxprofiler aims to facilitate three main steps of a typical shotgun metagenomic workflow (Chiu and Miller 2019). Taking in short- (e.g. Illumina) or long-read (e.g. Nanopore) FASTQ or FASTA files, it can (1) perform a range of appropriate read preprocessing steps, (2) perform taxonomic classification and profiling against a range of different tools depending on user preferences, and finally (3) perform post-classification aggregation and standardisation of the resulting profiles with the possibility of visualisation to the outputs (Figure 1). All relevant preprocessing statistics are displayed in an interactive and dynamic MultiQC report (Ewels et al. 2020).

3.1 Input and Execution

The pipeline can be executed via typical Nextflow commands, or using the standard nf-core 'launch' graphical-user-interface (GUI) (https://nf-co.re/taxprofiler/launch), making the pipeline accessible for both computationally experienced as well as less ex-

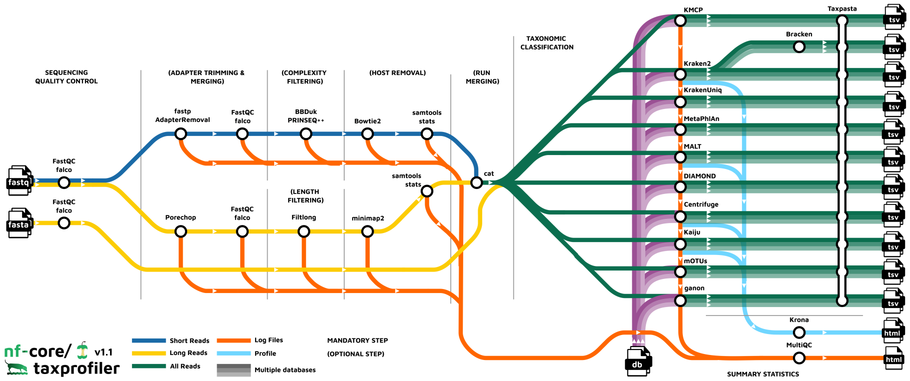
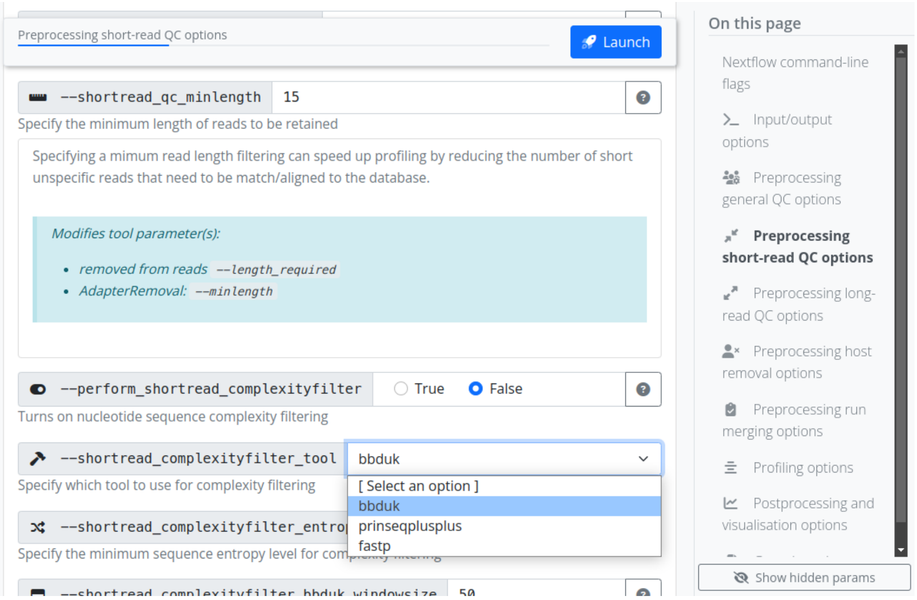


Figure 1: Visual overview of the nf-core/taxprofiler workflow. nf-core/taxprofiler can take in FASTQ (short or long reads) or FASTA files (long reads), that will optionally go through sequencing quality control (e.g. with FastQC), read preprocessing (e.g. removal of adapters), complexity filtering, host removal, and run merging before performing taxonomic classification and/or profiling with a user-selected range of tools and databases. Output from all classifiers and profilers are standardised into a common taxon table format, and when supported visualisations of the profiles are generated.

perienced researchers. In addition to the general usage and parameter documentation of the pipeline (https://nf-co.re/taxprofiler). The GUI offers immediate assistance and guidance to users on what each parameter does, both in short- and long-form, with long-form parameter descriptions additionally describing which tool-specific parameter(s) are being modified for each pipeline parameter (Figure 2). The GUI also includes controlled user input by providing strict drop-down lists and input validation prior execution of the pipeline to reduce the risk of typos and other mistakes (in contrast to the command-line interface (CLI) that only includes validation at pipeline run-time).

An example nf-core command line execution of the pipeline can be seen in figure (Code Block 1), where two input files are supplied: one file specifying paths of FASTQ files of metagenomic samples and necessary metadata for preprocessing (such as sample ID and sequencing platform), and the second file specifying paths to the user-defined databases with per-database classification parameters. Various parameters are available to optionally turn on different preprocessing steps, and provide additional configurations such as tool selection and value options. Note that even if a user supplies a given database in the database input sheet, the corresponding profiling tool must still be activated with the corresponding pipeline parameter (e.g. `--run_kraken2`). Per-classifier flags are also available for the optional saving of additional non-profile output files.

All nf-core pipelines are strictly versioned (specified with the Nextflow `-r` flag), and to ensure reproducibility, each version of the pipeline has a fixed set of software used for each step of the pipeline. The fixed set of software are controlled through the use



Preprocessing short-read QC options

Launch

`--shortread_qc_minlength` 15

Specify the minimum length of reads to be retained

Specifying a mimum read length filtering can speed up profiling by reducing the number of short unspecific reads that need to be match/aligned to the database.

Modifies tool parameter(s):

- *removed from reads* `--length_required`
- *AdapterRemoval:* `--minlength`

`--perform_shortread_complexityfilter` ○ True ● False

Turns on nucleotide sequence complexity filtering

`--shortread_complexityfilter_tool` bbduk

Specify which tool to use for complexity filtering

[Select an option]
bbduk
prinseqplusplus
fastp

`--shortread_complexityfilter_entro`

Specify the minimum sequence entropy level for com

`--shortread_complexityfilter_bbduk_windowsize` 50

On this page
Nextflow command-line flags
Input/output options
Preprocessing general QC options
Preprocessing short-read QC options
Preprocessing long-read QC options
Preprocessing host removal options
Preprocessing run merging options
Profiling options
Postprocessing and visualisation options
Show hidden params

Figure 2: Screenshot of the nf-core pipeline launch graphical user interface with nf-core/taxprofiler options displayed. The web browser-based interface provides guidance for how to configure each pipeline parameter by providing both short and long help descriptions to help guide users in which contexts to configure each parameter. Additional elements such as radio buttons, drop down menus, and background regular expressions check for validity of input. When pressing launch, a prepared configuration file and command is provided that can be copied and pasted by the user into the terminal

Listing 1 Example nf-core/taxprofiler command for running short-read quality control, removal of host DNA and executing the k-mer based Kraken2 and marker gene alignment MetaPhlAn3 tools.

```
$ nextflow run nf-core/taxprofiler -r 1.1.0 \
-profile singularity,<institute> \

--input <samplesheet.csv> --databases <database.csv> \
--perform_shortread_qc  --shortread_qc_minlength 20 --preprocessing_qc_tool falco \

--run_host_removal --hostremoval_reference 'host_genome.fasta' \
--run_kraken2 --kraken2_save_reads \

--run_metaphlan3 \
--run_krona \

--run_profile_standardisation \
```

of the conda package manager and containers (e.g., Docker, Apptainer [previously known as Singularity]) from the stable Bioconda (Grüning et al. 2018) and BioContainers (Veiga Leprevost et al. 2017) repositories. This, coupled with the intrinsic Nextflow ability to execute on most infrastructure whether that is a local laptop (resource requirements permitting), traditional HPC, as well across common cloud providers also makes nf-core/taxprofiler a very portable pipeline that can be used across many contexts.

3.2 Preprocessing

Preprocessing steps in nf-core/taxprofiler are aimed at removing laboratory and sequencing artefacts that may influence taxonomically profiling, either for computing resource consumption and/or false-positive or false-negative classification reasons. First sequencing quality control with FastQC (Andrews 2010) or Falco (Sena Brandine and Smith 2021) is carried out. Falco was included in for reduced memory requirements, in particular for long reads sequencing. Artificial library adapter sequences added during sequencing reduces accuracy during alignment by reducing sequence specificity, and in some cases, may result in false-positive hits due to adapter sequence contamination in reference genomes (Schäffer et al. 2018; F. P. Breitwieser, Baker, and Salzberg 2018) [1]. Additionally, the paired-end merging may provide longer sequences that will allow more specific classification when paired-end alignment is

[1] For an 'infamous' case of adapter sequences in a published eukaryotic genome, see the following blog posts

Graham Etherington: https://web.archive.org/web/20201219022000/http://grahametherington.blogspot.com/2014/09/why-you-should-qc-your-reads-and-your.html?m=1why-you-should-qc-your-reads-and-your.html Sixing Huang: https://web.archive.org/web/20220904205331/https://dgg32.medium.com/carpnot supported by a given classifier. For these tasks nf-core/taxprofiler can apply either fastp (Chen et al. 2018) or AdapterRemoval2 (Schubert, Lindgreen, and Orlando 2016) for short-reads, and currently Porechop (Wick et al. 2017) for Oxford Nanopore long-read data. For both short- and long-reads FastQC or Falco is run again to allow assessment on the performance of the adapter removal and/or pair-merging step.

Low complexity sequences, e.g. sequences containing long stretches mono- or di-nucleotide repeats provide little specific genetic information that contribute to taxonomic identification, as they can align to many different reference genomes (Schmieder and Edwards 2011; Clarke et al. 2019). Including such reads during taxonomic profiling can increase run-time and memory usage for little gain, as during lowest-common-ancestor (LCA) classification steps they will be assigned to high-level taxonomic levels (e.g. Kingdom). nf-core/taxprofiler performs removal of these reads through complexity filtering algorithms as provided by fastp, BBDuk (Bushnell 2022), or PRINSEQ++ (Cantu, Sadural, and Edwards 2019). Long read sequences often do not have such reads as lengths are sufficient enough to capture greater sequence diversity - but it is sometimes desired to only classify reads longer than a certain length - as these provide more precise taxonomic information (Dilthey et al. 2019; Portik, Brown, and Pierce-Ward 2022). Therefore, nf-core/taxprofiler can remove reads shorter than a user-defined length using Filtlong.

Removing host DNA is another common preprocessing step in metagenomic studies. This can help speed up run-time, particularly in microbiome studies, where detection of microbes are of interest. Furthermore, host-contamination of reference genomes in public databases is common (Longo, O'Neill, and O'Neill 2011; Kryukov and Imanishi 2016; Florian P. Breitwieser et al. 2019) and therefore the removal of such sequences can also decrease the risk of false positive taxonomic assignment. To remove multiple hosts or other sequences, all reference genomes can be combined into a single FASTA reference file. Short-read host removal can be carried out with Bowtie2 (Langmead and Salzberg 2012; Langmead et al. 2019) and minimap2 (Li 2018) for long-reads, both in combination with SAMtools (Li et al. 2009; Danecek et al. 2021), where reads are aligned against the reference genome and the off-target (unaligned) reads are then converted back to FASTQ format for classification.

Finally, nf-core/taxprofiler can optionally perform run merging where libraries have been sequenced over multiple lanes to generate one profile per sample or library. The final set of reads used for profiling can be optionally saved for downstream re-use.

Throughout all steps, relevant statistics and log files are generated and used both for the final pipeline run report as well as saved into the results directory of the pipeline run for further inspection where necessary.

3.3 Profiling

There are many types of metagenomic profiling techniques, from profiling against whole-genome references with alignment or k-mer based approaches, to methods

in-the-soil-1168818d2191
(Accessed 2023-08-25)

involving alignment to species-specific marker-gene families Ye et al. (2019). nf-core/taxprofiler aims to support and include all established classification or profiling tools as requested by the community. The choice of tools used in a pipeline run is up to the user, with a tool being executed when both the corresponding database and `--run_<tool>` flag is provided. Specific classification settings for each tool and database are specified in the database CSV input sheet. Some tools also have pipeline level command-line flags for controlling certain aspects of output files.

As of version 1.1.0, the following classifiers and profilers are available: Kraken2 (Wood, Lu, and Langmead 2019), Bracken (Lu et al. 2017), KrakenUniq (F. P. Breitwieser, Baker, and Salzberg 2018), Centrifuge (Kim et al. 2016), MALT (Vågene et al. 2018), DIAMOND (Buchfink, Reuter, and Drost 2021), Kaiju (Menzel, Ng, and Krogh 2016), MetaPhlAn (Blanco-Míguez et al. 2023), mOTUs (Ruscheweyh et al. 2022), ganon (Piro et al. 2020), KMCP (Shen et al. 2023). Table 1 summarises the category and reference database type for each tool.

Table 1: List of nf-core/taxprofiler supported taxonomic/classifiers profilers as of version 1.1 and their approximate method and supported input database types. Sequencing matching type refers to which 'molecular alphabet' is primarily used for matching between a query (read) and a reference (genome/gene). Primary algorithm refers to the algorithm type used for sequencing matching. Reference type refers to the typical sequence type used in database construction of the tool. Method refers to whether the tool performs just read classification (classifier) or additionally abundance estimation (profiler)

Sequence Matching Type	Primary Algorithm	Reference Type	Method	Tool
Nucleotide	k-mer based	whole-genome	classifier	Kraken2
Amino Acid	k-mer based	whole-genome	classifier	Kaiju
Nucleotide	k-mer based	whole-genome	profiler	Bracken
Nucleotide	k-mer based	whole-genome	profiler	KrakenUniq
Nucleotide	k-mer based	whole-genome	profiler	ganon
Nucleotide	k-mer based	whole-genome	profiler	KMCP
Nucleotide/Amino Acid	alignment based	whole-genome	classifier	MALT
Amino Acid	alignment based	whole-genome	classifer	DIAMOND
Nucleotide	alignment based	whole-genome	profiler	Centrifuge

Sequence Matching Type	Primary Algorithm	Reference Type	Method	Tool
Nucleotide	alignment based	marker-gene	profiler	MetaPhlAn
Nucleotide	alignment based	marker-gene	profiler	mOTUS

By default, nf-core/taxprofiler produces the per-sample main taxonomic classification profile from a tool or a tool's report generation tool. The output is normally in the form of counts per reference sequencing, with additional statistics about the hits of a particular organism (estimated abundance, taxonomic level etc.). Users can also optionally request output of per-read classification output, and output such as classified and unclassified reads in FASTQ format, where supported.

The pipeline provides high efficiency, particularly during the metagenomic classification stage, through the inherent parallelisation provided by Nextflow. While metagenomic classification is comparatively computationally intensive (in terms of memory and execution time; due to a combination of sequencing depth and number of reference genomes), Nextflow automatically optimises the execution order of all the steps in pipeline, maximising the number parallel running of multiple profilers and/or databases at any given time point, as far as the available computational resources allow. For local machines such as laptops or desktops, Nextflow will automatically detect all available computational sources but this is customisable using custom Nextflow configuration files. For HPC and cloud infrastructure, users typically have to define the computational infrastructural environment the pipeline is being executed on (CPU or memory limitations, queues, instance types etc.). To facilitate the pipeline set-up, nf-core/taxprofiler supports pre-defined centralised generic and pipeline-specific institutional Nextflow configurations as provided by nf-core/configs (https://nf-co.re/configs; more than 90 institutions at the time of writing). However, users are still welcome to supply their own custom configuration files, further refining computational limitations or execution specifications.

3.4 Post-profiling

In metagenomic studies, it is common practise to compare the profiles among many samples, and the results of multiple profiles are normally stored in 'taxon tables', i.e, counts per reference (rows), for each sample (columns). When available, nf-core/taxprofiler supports the option to produce the 'native' taxon table of each classification tool when multiple samples are run.

One of the challenges that researchers face when comparing multiple taxonomic classifiers or profilers is the heterogenous output formats that are produced, and often requires custom parser and merging scripts for each tool to standardise. To facilitate more user-friendly cross-comparison between tools, nf-core/taxprofiler utilises the TAXPASTA tool (Beber et al. 2023) to generate standardised profiles and generate multi-sample tables.

Summary statistics for the entire pipeline are visualised and displayed in a customisable MultiQC report (Ewels et al. 2020), when supported, to support user quality control of data and pipeline runs. Krona plots (Ondov, Bergman, and Phillippy 2011) can also optionally be generated for supported tools to help provide further visualisation of taxonomic profiles.

3.5 Output

To summarise, the main default output from nf-core/taxprofiler are both classifier 'native' and standardised single- and multi-sample taxonomic profiles with counts per-taxon and an interactive MultiQC run report with all run statistics, in addition to the raw log files themselves where available.

The MultiQC run report displays statistics and summary visualisations for all steps of the pipeline where possible, lists of versions for all tools of each step of the pipeline, and provides a dynamically-constructed text for the recommendeded 'methods' text for reporting how the pipeline was executed (including relevant citations) that users can use in their own publications.

Optional outputs can include other types of profiles (e.g. per read classification) and in other formats as produced by the tools themselves, as well as raw reads from preprocessing steps and output visualisations from Krona. Nextflow resource usage and trace reports are also by default produced for users to check pipeline performance.

4 Discussion

4.1 Comparison with other solutions

nf-core/taxprofiler has been specifically developed for the analysis of *metagenomic* data. While other types of taxonomic profiling data such as 16S amplicon sequencing are well established fields with a range of popular high-quality and best-practise tools pipelines (e.g. (Blanco-Míguez et al. 2023; Schloss et al. 2009)) and databases (DeSantis et al. 2006; Yilmaz et al. 2014), whereas 'gold standard' tools and databases for metagenomics remains are much less established - thus the need for highly-multiplexed classification is more desirable for the newer metagenomics methods. Despite this, tools such as METAXA2 (Bengtsson-Palme et al. 2015) that use shotgun sequencing reads to recover 16S sequences from metagenomic samples.

A range of pipelines already exist for taxonomic profiling, however each have their own particular purpose and abilities. Here we compare the functionality of nf-core/taxprofiler against four other recently published or released pipelines, selected based on their similarity of purpose to nf-core/taxprofiler. We searched Google Scholar for open-source pipelines published or released in the last 5 years (at the time of writing, since 2018) that were designed primarily for metagenomic classification screening, that supported at least 2 classifiers, had at laest one preprocessing step and was not specificaly targeted at read classification of specific domains of taxa (e.g. virus or bacteriophage only). We also included an additional pipeline at the recommendations of the authors of the pipeline due to the functional overlap to nf-core/taxprofiler. We then evaluted the pipelines based on their publications and documentation for typical metagenomic profiling workflow steps, and a range criteria related to expectations of modern bioinformatic workflows that can be summarised in the following four criteria: reproducibility, accessibility, scalability, and portability (Wratten, Wilm, and Göke 2021). After searching, we selected the following pipelines for comparison with nf-core/taxprofiler: sunbeam [v4; Clarke et al. (2019)], Unipro UGENE [v48; Rose et al. (2019)], TAMA [githash: 3a22c8f; Sim et al. (2020)], and StaG-mwc [0.7.0; Boulund et al. (2023)].

In terms of accessibility, all pipelines have documentation describing the installation steps, usage instructions, and output files. However, there are varying levels of detail and comprehensiveness. In particular, StaG-mwc and nf-core/taxprofiler have the most detailed descriptions of all possible output files for every supported module, whereas Unipro UGENE and sunbeam have very minimal to possibly unfinished output documentation. For execution options, most of the pipelines provide CLI execution, except for Unipro UGENE offers only GUI-based pipeline set-up (despite a command-line execution of the GUI generated configuration). In particular, nf-core/taxprofiler is the only pipeline providing both CLI and GUI interfaces for pipeline run execution.

Criteria covering portability also overlaps with accessibility, as it implies the options and ease that different users can run on different types of computing infrastructure, whether that is on their own laptop, on a HPC cluster or in the cloud. Unipro UGENE is the only pipeline that supports execution on all three major operating systems (Linux, OSX, Windows), whereas StaG-mwc and nf-core/taxprofiler can be run on unix operating systems, and sunbeam and TAMA are only being supported on Linux. While all pipelines support 'local' machine execution (e.g. personal laptops or desktops), a large portion of academic users execute computationally intensive bioinformatic tasks on HPC clusters. In these contexts, pipeline task submissions are normally managed by job schedulers, thus integration with schedulers is an important criteria for running large multi-step and parallelised pipelines. The three pipelines leveraging workflow managers (Snakemake (Mölder et al. 2021) and Nextflow) support integration with schedulers (StaG-mwc, sunbeam, and nf-core/taxprofiler) with nf-core/taxprofiler supporting the most by far (>10 scheduling systems) as natively offered by Nextflow. This allows the greatest possible choice or users in terms of which HPC infrastructure they could execute their pipeline on. As an extension of this, only nf-core/taxprofiler has explicitly described support for cloud computing (e.g. AWS or Microsoft Azure), again maximising user choice and accessibility when it comes to running the pipeline.

In terms of scalability, the aforementioned integration with schedulers and cloud computing support implicit maximises efficiency and parallellisation of pipeline runs, providing good scalability for varying numbers of input files and steps in the pipeline. Again, the three workflow manager based pipelines provide scalability, whereas there is no mention neither Unipro UGENE nor TAMA in reference to parallel task execution. Furthemore, all pipelines except TAMA, allowed per-process customisation of computational resources, something critical for maximising efficient scalability to ensure only the necessary resources for a given step of a pipeline are requested.

In terms of reproducibility, all five pipelines are good at ensuring reproducibility in terms of pipeline and software versioning (allowing re-execution of pipeline runs using the same software), with only tama not having stable versioned releases. However, installing software manually across different infrastructures can result in variability in the execution of each software [2] (Di Tommaso et al. 2017).The current most popular solution to the problem of inconsistent software environments is to use container engines such as Docker or Apptainer. These allow 'snapshotting' of a operating system and all configuration and fixed versions of all software required for the pipeline. Only Unipro UGENE does not document the use of a container system with nf-core/taxprofiler offering the biggest choice for users courtsey of Nextflow (6 different engine systems at the time of writing).

Finally we compared metagenomics related functionality between the pipelines. All pipelines support short-read FASTQ input, but only nf-core/taxprofiler explicitly reports long-read support, while the documentation in Unipro UGENE states that assembled contigs are possible input to some of the profilers. All pipelines support read preprocessing (adapter clipping, and merging). In terms of tools used for preprocessing, Trimmomatic (Bolger, Lohse, and Usadel 2014) is popular across the other pipelines but is not supported in nf-core/taxprofiler. Only sunbeam and nf-core/taxprofiler support complexity filtering to remove low sequence diversity reads. In fact within the development of sunbeam, the developers developed their own dedicated performant complexity filtering tool Komplexity (Clarke et al. 2019), which may be of interest for adding to nf-core/taxprofiler if containers are created. Most pipelines support some form of host removal (only TAMA did not support this), and it is likely possible with Unipro UGENE through user customisation of the workflow. In all cases, this consists of mapping processed reads with an aligner and taking the off-target reads (as implemented in nf-core/taxprofiler), however StaG-mwc has an additional separate metagenomic host removal step with Kraken2. nf-core/taxprofiler supports by far the largest number of taxonomic classifers and profilers at 11 as of v1.1.0 - providing the greatest choice to users - with StaG-mwc offering 7, and the remaining pipelines only 3. Only nf-core/taxprofiler and partly StaG-mwc explicitly supports running each profiler with multiple databases.nf-core/taxprofiler is the only pipeline that supports running an arbitrary number of different metagenomic profiler databases each with their own settings - making it useful for tool parameter comparison, testing different databases, or reducing the size of each database (e.g. per domain) to make it more flexibility for running on smaller computational infrastructure. StaG-mwc allows multiple references for their short-read alignment steps rather than the metagenomic profilers. For output, nf-core/taxprofiler, StaG-mwc, and sunbeam (via an extension) support a singular run report for summarising all preprocessing step. Only nf-core/taxprofiler and tama produce standardised output for all taxonomic profilers (via TAXPASTA). However Unipro UGENE additionally offers a 'consensus' profile

[2] As demonstrated in this blogpost from Paweł Przytuła: https://web.archive.org/web/20230320223436/https://appsilon.com/reproducible-research-when-your-results-cant-be-reproduced/ (Accessed 2023-08-25)

using WEVOTE (Metwally et al. 2016).

To summarise, many of the pipelines reviewed here offer similar functionality, with particularly StaG-mwc having a strong overlap with nf-core/taxprofiler. Thus, users in most cases will be able to select the pipeline depending on which pipeline framework they feel most comfortable with. However the advantages of nf-core/taxprofiler mainly comes from the offering of the greatest choice of tools, the benefits provided by Nextflow whereby it provides the greatest number of computational infrastructure types the pipeline can be executed on and container systems can be used to ensure reproducibility, and the support of the nf-core community due to the centralised pool of 'plug-and-play' modules to make it easier to update the pipeline overtime to add new tool.

The functionality offered by other pipelines not currently supported by nf-core/taxprofiler include sequencing saturation estimation (StaG-mwc), taxonomy-free composition comparison (StaG-mwc), functional profiling (StaG-mwc), *de novo* assembly (sunbeam), and reference mapping (StaG-mwc, sunbeam). We do not plan to support *de novo* assembly or functional profiling in nf-core/taxprofiler as we feel this better served by other existing dedicated pipelines (e.g. Uritskiy, DiRuggiero, and Taylor 2018; Krakau et al. 2022).

We note there exists a range of other pipelines that also include some form of taxonomic classification. However often these pipelines have been developed with a different main purpose (e.g. Assembly and binning for nf-core/mag (Krakau et al. 2022), MetaWRAP (Uritskiy, DiRuggiero, and Taylor 2018), SqueezeMeta (Tamames and Puente-Sánchez 2018), or MEDUSA (Morais et al. 2022); Metagenomic read alignment with CCMetaGen (Marcelino et al. 2020) and Wochenende (Rosenboom et al. 2022)).

4.2 Development roadmap

An important advantage of nf-core/taxprofiler is that it is being developed within the nf-core community (https://nf-co.re), that provides a strong and long term support for the continued community-based development and maintenance of the pipeline. In this framework, we will continue to add additional preprocessing and metagenomic classification and profiling tools as they become established and as requested by the metagenomics community. For example, we feel inclusion of steps such as sequencing saturation estimation as already being performed by StaG-mwc would be beneficial to the nf-core/taxprofiler workflow (possibly with dedicated tools such as Nonpariel (Rodriguez-R et al. 2018) once gzipped FASTQ input is supported), and/or more performant complexity filtering tools such as Komplexity as offered by sunbeam (once software containers are offered for this tool). This also applies to extend support to other sequencing platforms; nf-core/taxprofiler already supports Nanopore long-read data, however the use of long-read PacBio data for metagenomic data is growing in interest (Portik, Brown, and Pierce-Ward 2022). We are therefore considering adding dedicated preprocessing steps for this type of sequencing data.

A remaining major challenge for metagenomics researchers (and not supported in the same workflow by any of the compared pipelines above) is the construction of databases for each profiling tool. Given there still are no curated, high-quality 'gold standard' databases in metagenomics, and while nf-core/taxprofiler allows the profiling against multiple databases and settings in parallel, currently the pipeline still requires users to construct these manually and to supply to the pipeline. While we feel this is currently a reasonable investment as such databases can be repeatedly re-used, we are exploring the possibility to add an additional complementary workflow in the pipeline to allow automated database construction of all classification tools, given a set of FASTA reference files.

Finally, once an overall taxonomic profile is generated, researchers often wish to validate hits through more sensitive and accurate methods such as with read-mapping alignment. While read alignment is supported by other pipelines such as StaG-mwc, this happens in-parallel to the taxonomic profiling and requires prior expectation of which reference genomes to map against. Instead, nf-core/profiler could be easily extended to have a validation step similar to that of the ancient DNA metagenomic pipeline aMeta (Pochon et al. 2022) where, utilising Nextflow's execution parallelism, the input sequences could be aligned back to the reference genomes of only those species with hits from the taxonomic classification with dedicated accurate short- or long-read aligners. In addition to the more precise classification, post-classification read-alignment could also be particularly useful for researchers in palaeogenomics who wish to use other tools other than KrakenUniq for initial classification (as in aMeta), where alignment information can be used to authenticate ancient DNA within their samples but also in clinical metagenomics to identify potential pathogens at a much finer resolution (e.g. down to strain level).

Another motivation for developing nf-core/taxprofiler, despite the large number of existing metagenomics pipelines is by establishing a taxonomic profiling pipeline within the nf-core ecosystem, it is possible to begin building both standalone but also an integrated suite of powerful interconnected pipelines for the major stages of metagenomic workflows. Existing microbial- and metagenomics- related pipelines within the nf-core initiative include nf-core/ampliseq, nf-core/mag, and nf-core/funcscan. We expect over time the ability to link inputs and outputs of each workflow to develop comphrensive metagenomic analyses, while still maintaing powerful standalone pipelines, providing maximal user choice.

5 Conclusion

nf-core/taxprofiler is an accessible, efficient, and scalable pipeline for metagenomic taxonomic classification and profiling that can be executed on anywhere from laptops to the cloud. Offering, to our knowledge, the largest number of taxonomic profilers across similar pipelines, it provides flexibility for users not just on choice of profiling tool but also with databases and database settings, with any number being able to be supplied to the pipeline in a single run. We hope that through detailed documentation and a range of execution options, nf-core/taxprofiler will make reproducible and high-throughput metagenomics more accessible for a wide range of disciplines.

6 Data Availability

All data used in this publication

7 Code Availability

nf-core/taxprofiler source code is available on GitHub at https://github.com/nf-core/taxprofiler, and each release is archived on Zenodo (latest version DOI: 10.5281/zenodo.7728364)

The version of the pipeline described in this paper is version (1.1.0) (release specific Zenodo archive DOI: 10.5281/zenodo.8358147)

8 Supplementary Data

9 Acknowledgments

We thank Prof. Christina Warinner and the Microbiome Sciences group MPI-EVA for original discussions that lead to the pipeline. We are also grateful for the nf-core community for the original and ongoing support in the development in the pipeline, in particular for the contributions by Lauri Mesilaakso, Jianhong Ou, and Rafał Stępień.

10 Funding

S.S. and L.A-L. were supported by Rapid establishment of comprehensive laboratory pandemic preparedness – RAPID-SEQ. This material is based upon work supported by the U.S. Department of Agriculture, Agricultural Research Service, under agreement No. 58-3022-0-001 (T.A.C II). M.B. and J.A.F.Y were supported by the Max Planck Society. J.A.F.Y was supported by the Werner Siemens-Stiftung (“Paleobiotechnology”, Awarded to Prof. Pierre Stallforth and Prof. Christina Warinner).

References

Andrews, Simon. 2010. “FastQC: A Quality Control Tool for High Throughput Sequence Data.” http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

Beber, Moritz E, Maxime Borry, Sofia Stamouli, and James A Fellows Yates. 2023. “TAXPASTA: TAXonomic Profile Aggregation and STAndardisation.” *Journal of Open Source Software* 8 (87): 5627. https://doi.org/10.21105/joss.05627.

Bengtsson-Palme, Johan, Martin Hartmann, Karl Martin Eriksson, Chandan Pal, Kaisa Thorell, Dan Göran Joakim Larsson, and Rolf Henrik Nilsson. 2015. “METAXA2: Improved Identification and Taxonomic Classification of Small and Large Subunit rRNA in Metagenomic Data.” *Molecular Ecology Resources* 15 (6): 1403–14. https://doi.org/10.1111/1755-0998.12399.

Blanco-Míguez, Aitor, Francesco Beghini, Fabio Cumbo, Lauren J McIver, Kelsey N Thompson, Moreno Zolfo, Paolo Manghi, et al. 2023. "Extending and Improving Metagenomic Taxonomic Profiling with Uncharacterized Species Using MetaPhlAn 4." *Nature Biotechnology*, February, 1–12. https://doi.org/10.1038/s41587-023-01688-w.

Bolger, Anthony M, Marc Lohse, and Bjoern Usadel. 2014. "Trimmomatic: A Flexible Trimmer for Illumina Sequence Data." *Bioinformatics (Oxford, England)* 30 (15): 2114–20. https://doi.org/10.1093/bioinformatics/btu170.

Boulund, Fredrik, Aron Arzoomand, Justine Debelius, chrsb, and Lisa Olsson. 2023. "Ctmrbio/Stag-Mwc: StaG v0.7.0." Zenodo. https://doi.org/10.5281/ZENODO.8032462.

Breitwieser, F P, D N Baker, and S L Salzberg. 2018. "KrakenUniq: Confident and Fast Metagenomics Classification Using Unique k-Mer Counts." *Genome Biology* 19 (1): 198. https://doi.org/10.1186/s13059-018-1568-0.

Breitwieser, Florian P, Jennifer Lu, and Steven L Salzberg. 2019. "A Review of Methods and Databases for Metagenomic Classification and Assembly." *Briefings in Bioinformatics* 20 (4): 1125–36. https://doi.org/10.1093/bib/bbx120.

Breitwieser, Florian P, Mihaela Pertea, Aleksey Zimin, and Steven L Salzberg. 2019. "Human Contamination in Bacterial Genomes Has Created Thousands of Spurious Proteins." *Genome Research* 29 (May): 954–60. https://doi.org/10.1101/gr.245373.118.

Buchfink, Benjamin, Klaus Reuter, and Hajk-Georg Drost. 2021. "Sensitive Protein Alignments at Tree-of-Life Scale Using DIAMOND." *Nature Methods* 18 (4): 366–68. https://doi.org/10.1038/s41592-021-01101-x.

Bushnell, Brian. 2022. "BBMap." https://sourceforge.net/projects/bbmap/.

Cantu, Vito Adrian, Jeffrey Sadural, and Robert Edwards. 2019. "PRINSEQ++, a Multi-Threaded Tool for Fast and Efficient Quality Control and Preprocessing of Sequencing Datasets." e27553v1. PeerJ Preprints; PeerJ Inc. https://doi.org/10.7287/peerj.preprints.27553v1.

Chen, Shifu, Yanqing Zhou, Yaru Chen, and Jia Gu. 2018. "Fastp: An Ultra-Fast All-in-One FASTQ Preprocessor." *Bioinformatics* 34 (17): i884–90. https://doi.org/10.1093/bioinformatics/bty560.

Chiu, Charles Y, and Steven A Miller. 2019. "Clinical Metagenomics." *Nature Reviews. Genetics* 20 (6): 341–55. https://doi.org/10.1038/s41576-019-0113-7.

Clarke, Erik L, Louis J Taylor, Chunyu Zhao, Andrew Connell, Jung-Jin Lee, Bryton Fett, Frederic D Bushman, and Kyle Bittinger. 2019. "Sunbeam: An Extensible Pipeline for Analyzing Metagenomic Sequencing Experiments." *Microbiome* 7 (1): 46. https://doi.org/10.1186/s40168-019-0658-x.

Danecek, Petr, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, et al. 2021. "Twelve Years of SAMtools and BCFtools." *GigaScience* 10 (2). https://doi.org/10.1093/gigascience/giab008.

DeSantis, T Z, P Hugenholtz, N Larsen, M Rojas, E L Brodie, K Keller, T Huber, D Dalevi, P Hu, and G L Andersen. 2006. "Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB." *Applied and Environmental Microbiology* 72 (7): 5069–72. https://doi.org/10.1128/AEM.03006-05.

Di Tommaso, Paolo, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. "Nextflow Enables Reproducible Computational Workflows." *Nature Biotechnology* 35 (4): 316–19. https://doi.org/10.1038/nbt.3820.

Dilthey, Alexander T, Chirag Jain, Sergey Koren, and Adam M Phillippy. 2019. "Strain-Level Metagenomic Assignment and Compositional Estimation for Long Reads with MetaMaps." *Nature Communications* 10 (1): 3066. https://doi.org/10.1038/s41467-019-10934-2.

Eloe-Fadrosh, Emiley A, Natalia N Ivanova, Tanja Woyke, and Nikos C Kyrpides. 2016. "Metagenomics Uncovers Gaps in Amplicon-Based Detection of Microbial Diversity." *Nature Microbiology* 1 (4): 15032. https://doi.org/10.1038/nmicrobiol.2015.32.

Ewels, Philip A, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen. 2020. "The Nf-Core Framework for Community-Curated Bioinformatics Pipelines." *Nature Biotechnology* 38 (3): 276–78. https://doi.org/10.1038/s41587-020-0439-x.

Govender, Kumeren N, and David W Eyre. 2022. "Benchmarking Taxonomic Classifiers with Illumina and Nanopore Sequence Data for Clinical Metagenomic Diagnostic Applications." *Microbial Genomics* 8 (10): 000886. https://doi.org/10.1099/mgen.0.000886.

Grüning, Björn, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe, Christopher H Tomkins-Tinch, Renan Valieris, Johannes Köster, and Bioconda Team. 2018. "Bioconda: Sustainable and Comprehensive Software Distribution for the Life Sciences." *Nature Methods* 15 (7): 475–76. https://doi.org/10.1038/s41592-018-0046-7.

Hillmann, Benjamin, Gabriel A Al-Ghalith, Robin R Shields-Cutler, Qiyun Zhu, Daryl M Gohl, Kenneth B Beckman, Rob Knight, and Dan Knights. 2018. "Evaluating the Information Content of Shallow Shotgun Metagenomics." *mSystems* 3 (6). https://doi.org/10.1128/mSystems.00069-18.

Kim, Daehwan, Li Song, Florian P Breitwieser, and Steven L Salzberg. 2016. "Centrifuge: Rapid and Sensitive Classification of Metagenomic Sequences." *Genome Research* 26 (12): 1721–29. https://doi.org/10.1101/gr.210641.116.

Krakau, Sabrina, Daniel Straub, Hadrien Gourlé, Gisela Gabernet, and Sven Nahnsen. 2022. "Nf-Core/Mag: A Best-Practice Pipeline for Metagenome Hybrid Assembly and Binning." *NAR Genomics and Bioinformatics* 4 (1). https://doi.org/10.1093/nargab/lqac007.

Kryukov, Kirill, and Tadashi Imanishi. 2016. "Human Contamination in Public Genome Assemblies." *PloS One* 11 (9): e0162424. https://doi.org/10.1371/journal.pone.0162424.

Langmead, Ben, and Steven L Salzberg. 2012. "Fast Gapped-Read Alignment with Bowtie 2." *Nature Methods* 9 (4): 357–59. https://doi.org/10.1038/nmeth.1923.

Langmead, Ben, Christopher Wilks, Valentin Antonescu, and Rone Charles. 2019. "Scaling Read Aligners to Hundreds of Threads on General-Purpose Processors." *Bioinformatics* 35 (3): 421–32. https://doi.org/10.1093/bioinformatics/bty648.

Li, Heng. 2018. "Minimap2: Pairwise Alignment for Nucleotide Sequences." *Bioinformatics* 34 (18): 3094–3100. https://doi.org/10.1093/bioinformatics/bty191.

Li, Heng, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor

Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. 2009. "The Sequence Alignment/Map Format and SAMtools." *Bioinformatics* 25 (16): 2078–79. https://doi.org/10.1093/bioinformatics/btp352.

Longo, Mark S, Michael J O'Neill, and Rachel J O'Neill. 2011. "Abundant Human DNA Contamination Identified in Non-Primate Genome Databases." *PloS One* 6 (2): e16410. https://doi.org/10.1371/journal.pone.0016410.

Lu, Jennifer, Florian P Breitwieser, Peter Thielen, and Steven L Salzberg. 2017. "Bracken: Estimating Species Abundance in Metagenomics Data." *PeerJ. Computer Science* 3 (e104): e104. https://doi.org/10.7717/peerj-cs.104.

Lynch, Michael D J, and Josh D Neufeld. 2015. "Ecology and Exploration of the Rare Biosphere." *Nature Reviews. Microbiology* 13 (4): 217–29. https://doi.org/10.1038/nrmicro3400.

Marcelino, Vanessa R, Philip T L C Clausen, Jan P Buchmann, Michelle Wille, Jonathan R Iredell, Wieland Meyer, Ole Lund, Tania C Sorrell, and Edward C Holmes. 2020. "CCMetagen: Comprehensive and Accurate Identification of Eukaryotes and Prokaryotes in Metagenomic Data." *Genome Biology* 21 (1): 103. https://doi.org/10.1186/s13059-020-02014-2.

McIntyre, Alexa B R, Rachid Ounit, Ebrahim Afshinnekoo, Robert J Prill, Elizabeth Hénaff, Noah Alexander, Samuel S Minot, et al. 2017. "Comprehensive Benchmarking and Ensemble Approaches for Metagenomic Classifiers." *Genome Biology* 18 (1): 182. https://doi.org/10.1186/s13059-017-1299-7.

Menzel, Peter, Kim Lee Ng, and Anders Krogh. 2016. "Fast and Sensitive Taxonomic Classification for Metagenomics with Kaiju." *Nature Communications* 7 (April): 11257. https://doi.org/10.1038/ncomms11257.

Metwally, Ahmed A, Yang Dai, Patricia W Finn, and David L Perkins. 2016. "WEVOTE: Weighted VOting Taxonomic idEntification Method of Microbial Sequences." *PloS One* 11 (9): e0163527. https://doi.org/10.1371/journal.pone.0163527.

Meyer, Fernando, Adrian Fritz, Zhi-Luo Deng, David Koslicki, Till Robin Lesker, Alexey Gurevich, Gary Robertson, et al. 2022. "Critical Assessment of Metagenome Interpretation: The Second Round of Challenges." *Nature Methods* 19 (4): 429–40. https://doi.org/10.1038/s41592-022-01431-4.

Mitchell, Alex L, Alexandre Almeida, Martin Beracochea, Miguel Boland, Josephine Burgin, Guy Cochrane, Michael R Crusoe, et al. 2019. "MGnify: The Microbiome Analysis Resource in 2020." *Nucleic Acids Research*, November. https://doi.org/10.1093/nar/gkz1035.

Mölder, Felix, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, et al. 2021. "Sustainable Data Analysis with Snakemake." *F1000Research* 10 (January): 33. https://doi.org/10.12688/f1000research.29032.2.

Morais, Diego A A, João V F Cavalcante, Shênia S Monteiro, Matheus A B Pasquali, and Rodrigo J S Dalmolin. 2022. "MEDUSA: A Pipeline for Sensitive Taxonomic Classification and Flexible Functional Annotation of Metagenomic Shotgun Sequences." *Frontiers in Genetics* 13 (March): 814437. https://doi.org/10.3389/fgene.2022.814437.

Nasko, Daniel J, Sergey Koren, Adam M Phillippy, and Todd J Treangen. 2018. "RefSeq Database Growth Influences the Accuracy of k-Mer-Based Lowest Common

Ancestor Species Identification.” *Genome Biology* 19 (1): 165. https://doi.org/10.1186/s13059-018-1554-6.

Nayfach, Stephen, and Katherine S Pollard. 2016. “Toward Accurate and Quantitative Comparative Metagenomics.” *Cell* 166 (5): 1103–16. https://doi.org/10.1016/j.cell.2016.08.007.

Ondov, Brian D, Nicholas H Bergman, and Adam M Phillippy. 2011. “Interactive Metagenomic Visualization in a Web Browser.” *BMC Bioinformatics* 12 (1): 385. https://doi.org/10.1186/1471-2105-12-385.

Piro, Vitor C, Temesgen H Dadi, Enrico Seiler, Knut Reinert, and Bernhard Y Renard. 2020. “Ganon: Precise Metagenomics Classification Against Large and up-to-Date Sets of Reference Sequences.” *Bioinformatics (Oxford, England)* 36 (Suppl_1): i12–20. https://doi.org/10.1093/bioinformatics/btaa458.

Pochon, Zoé, Nora Bergfeldt, Emrah Kırdök, Mário Vicente, Thijessen Naidoo, Tom van der Valk, N Ezgi Altınışık, et al. 2022. “aMeta: An Accurate and Memory-Efficient Ancient Metagenomic Profiling Workflow.” *bioRxiv*. https://doi.org/10.1101/2022.10.03.510579.

Portik, Daniel M, C Titus Brown, and N Tessa Pierce-Ward. 2022. “Evaluation of Taxonomic Classification and Profiling Methods for Long-Read Shotgun Metagenomic Sequencing Datasets.” *BMC Bioinformatics* 23 (1): 541. https://doi.org/10.1186/s12859-022-05103-0.

Quince, Christopher, Alan W Walker, Jared T Simpson, Nicholas J Loman, and Nicola Segata. 2017. “Shotgun Metagenomics, from Sampling to Analysis.” *Nature Biotechnology* 35 (9): 833–44. https://doi.org/10.1038/nbt.3935.

Rodriguez-R, Luis M, Santosh Gunturu, James M Tiedje, James R Cole, and Konstantinos T Konstantinidis. 2018. “Nonpareil 3: Fast Estimation of Metagenomic Coverage and Sequence Diversity.” *mSystems* 3 (3). https://doi.org/10.1128/mSystems.00039-18.

Rose, Rebecca, Olga Golosova, Dmitrii Sukhomlinov, Aleksey Tiunov, and Mattia Prosperi. 2019. “Flexible Design of Multiple Metagenomics Classification Pipelines with UGENE.” *Bioinformatics (Oxford, England)* 35 (11): 1963–65. https://doi.org/10.1093/bioinformatics/bty901.

Rosenboom, Ilona, Tobias Scheithauer, Fabian C Friedrich, Sophia Pörtner, Lisa Hollstein, Marie-Madlen Pust, Konstantinos Sifakis, et al. 2022. “Wochenende - Modular and Flexible Alignment-Based Shotgun Metagenome Analysis.” *BMC Genomics* 23 (1): 748. https://doi.org/10.1186/s12864-022-08985-9.

Ruscheweyh, Hans-Joachim, Alessio Milanese, Lucas Paoli, Nicolai Karcher, Quentin Clayssen, Marisa Isabell Keller, Jakob Wirbel, et al. 2022. “Cultivation-Independent Genomes Greatly Expand Taxonomic-Profiling Capabilities of mOTUs Across Various Environments.” *Microbiome* 10 (1): 212. https://doi.org/10.1186/s40168-022-01410-z.

Schäffer, Alejandro A, Eric P Nawrocki, Yoon Choi, Paul A Kitts, Ilene Karsch-Mizrachi, and Richard McVeigh. 2018. “VecScreen_plus_taxonomy: Imposing a Tax(onomy) Increase on Vector Contamination Screening.” *Bioinformatics (Oxford, England)* 34 (5): 755–59. https://doi.org/10.1093/bioinformatics/btx669.

Schloss, Patrick D, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hartmann, Emily B Hollister, Ryan A Lesniewski, et al. 2009. “Introducing Mothur:

Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities." *Applied and Environmental Microbiology* 75 (23): 7537–41. https://doi.org/10.1128/AEM.01541-09.

Schmieder, Robert, and Robert Edwards. 2011. "Quality Control and Preprocessing of Metagenomic Datasets." *Bioinformatics (Oxford, England)* 27 (6): 863–64. https://doi.org/10.1093/bioinformatics/btr026.

Schubert, Mikkel, Stinus Lindgreen, and Ludovic Orlando. 2016. "AdapterRemoval v2: Rapid Adapter Trimming, Identification, and Read Merging." *BMC Research Notes* 9 (February): 88. https://doi.org/10.1186/s13104-016-1900-2.

Sczyrba, Alexander, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Dröge, Ivan Gregor, et al. 2017. "Critical Assessment of Metagenome Interpretation-a Benchmark of Metagenomics Software." *Nature Methods* 14 (11): 1063–71. https://doi.org/10.1038/nmeth.4458.

Sena Brandine, Guilherme de, and Andrew D Smith. 2021. "Falco: High-Speed FastQC Emulation for Quality Control of Sequencing Data." *F1000Research* 8 (1874): 1874. https://doi.org/10.12688/f1000research.21142.2.

Sharpton, Thomas J. 2014. "An Introduction to the Analysis of Shotgun Metagenomic Data." *Frontiers in Plant Science* 5 (June): 209. https://doi.org/10.3389/fpls.2014.00209.

Shen, Wei, Hongyan Xiang, Tianquan Huang, Hui Tang, Mingli Peng, Dachuan Cai, Peng Hu, and Hong Ren. 2023. "KMCP: Accurate Metagenomic Profiling of Both Prokaryotic and Viral Populations by Pseudo-Mapping." *Bioinformatics* 39 (1): btac845. https://doi.org/10.1093/bioinformatics/btac845.

Sim, Mikang, Jongin Lee, Daehwan Lee, Daehong Kwon, and Jaebum Kim. 2020. "TAMA: Improved Metagenomic Sequence Classification Through Meta-Analysis." *BMC Bioinformatics* 21 (1): 185. https://doi.org/10.1186/s12859-020-3533-7.

Sun, Zheng, Shi Huang, Meng Zhang, Qiyun Zhu, Niina Haiminen, Anna Paola Carrieri, Yoshiki Vázquez-Baeza, et al. 2021. "Challenges in Benchmarking Metagenomic Profilers." *Nature Methods* 18 (6): 618–26. https://doi.org/10.1038/s41592-021-01141-3.

Tamames, Javier, and Fernando Puente-Sánchez. 2018. "SqueezeMeta, a Highly Portable, Fully Automatic Metagenomic Analysis Pipeline." *Frontiers in Microbiology* 9: 3349. https://doi.org/10.3389/fmicb.2018.03349.

Uritskiy, Gherman V, Jocelyne DiRuggiero, and James Taylor. 2018. "MetaWRAP-a Flexible Pipeline for Genome-Resolved Metagenomic Data Analysis." *Microbiome* 6 (1): 158. https://doi.org/10.1186/s40168-018-0541-1.

Vågene, Åshild J, Alexander Herbig, Michael G Campana, Nelly M Robles García, Christina Warinner, Susanna Sabin, Maria A Spyrou, et al. 2018. "Salmonella Enterica Genomes from Victims of a Major Sixteenth-Century Epidemic in Mexico." *Nature Ecology & Evolution* 2 (3): 520–28. https://doi.org/10.1038/s41559-017-0446-6.

Veiga Leprevost, Felipe da, Björn A Grüning, Saulo Alves Aflitos, Hannes L Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, et al. 2017. "BioContainers: An Open-Source and Community-Driven Framework for Software Standardization." *Bioinformatics (Oxford, England)* 33 (16): 2580–82. https://doi.org/10.1093/bioinformatics/btx192.

Wick, Ryan R, Louise M Judd, Claire L Gorrie, and Kathryn E Holt. 2017. “Completing Bacterial Genome Assemblies with Multiplex MinION Sequencing.” *Microbial Genomics* 3 (10): e000132. https://doi.org/10.1099/mgen.0.000132.

Wood, Derrick E, Jennifer Lu, and Ben Langmead. 2019. “Improved Metagenomic Analysis with Kraken 2.” *Genome Biology* 20 (1): 257. https://doi.org/10.1186/s13059-019-1891-0.

Wratten, Laura, Andreas Wilm, and Jonathan Göke. 2021. “Reproducible, Scalable, and Shareable Analysis Pipelines with Bioinformatics Workflow Managers.” *Nature Methods* 18 (10): 1161–68. https://doi.org/10.1038/s41592-021-01254-9.

Wright, Robyn J, Andrè M Comeau, and Morgan G I Langille. 2023. “From Defaults to Databases: Parameter and Database Choice Dramatically Impact the Performance of Metagenomic Taxonomic Classification Tools.” *Microbial Genomics* 9 (3). https://doi.org/10.1099/mgen.0.000949.

Ye, Simon H, Katherine J Siddle, Daniel J Park, and Pardis C Sabeti. 2019. “Benchmarking Metagenomics Tools for Taxonomic Classification.” *Cell* 178 (4): 779–94. https://doi.org/10.1016/j.cell.2019.07.010.

Yilmaz, Pelin, Laura Wegener Parfrey, Pablo Yarza, Jan Gerken, Elmar Pruesse, Christian Quast, Timmy Schweer, Jörg Peplies, Wolfgang Ludwig, and Frank Oliver Glöckner. 2014. “The SILVA and ‘All-Species Living Tree Project (LTP)’ Taxonomic Frameworks.” *Nucleic Acids Research* 42 (Database issue): D643–8. https://doi.org/10.1093/nar/gkt1209.