

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

Sofia Stamouli¹, Moritz E. Beber², Tanja Normark³, Thomas A. Christensen II⁴, Lili Andersson-Li⁵, Maxime Borry⁶, Mahwash Jamy⁷, nf-core community⁸, James A. Fellows Yates⁹

¹Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

¹Department of Clinical Microbiology, Karolinska University Hospital

²Unseen Bio ApS

³Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

³Department of Clinical Microbiology, Karolinska University Hospital

⁴Veterinary Diagnostic Laboratory, Kansas State University College of Veterinary Medicine

⁵Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

⁵Department of Clinical Microbiology, Karolinska University Hospital

⁶Department of Archaeogenetics, Max Planck Institute for Evolutionary Anthropology

⁶Research Unit Archaeogenetics, Leibniz Institute for Natural Product Research and Infection Biology Hans Knöll Institute (current address)

⁷Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

⁷Department of Clinical Microbiology, Karolinska University Hospital

⁸

⁹Department of Archaeogenetics, Max Planck Institute for Evolutionary Anthropology

⁹Research Unit Archaeogenetics, Leibniz Institute for Natural Product Research and Infection Biology Hans Knöll Institute (current address)

⁹Research Unit Paleobiotechnology, Leibniz Institute for Natural Product Research and Infection Biology Hans Knöll Institute (current address)

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 read-processing and taxonomic classification pipeline. It is designed for the automated and simultaneous clas-

sification and/or profiling of both short- and long-read metagenomic sequencing libraries against a 11 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, accommodating from small to extremely large projects on a wide range of computing infrastructure. It has been developed following best-practise software development practises and community support to ensure longevity and adaptability of the pipeline, to help keep it up to date with the field of metagenomics.

2 Introduction

Whole-genome, metagenomic sequencing 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 used 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, for example, shotgun sequencing. These metagenomic analyses with whole microbial genome as references have been shown to provide a similar level of taxonomic resolution (Hillmann et al. 2018). However they also have the added benefit of having greater reusability potential of the data, such as for whole genome and/or functional classification (Sharpton 2014; Quince et al. 2017).

Taxonomic classifiers (sometimes referred to as taxonomic bidders) aim to identify 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 reads (sequenced DNA molecules) against hundreds or thousands of reference genomes either via sequence alignment or ‘k-mer matching’ (Sharpton 2014; Sun et al. 2021). The reference genome with the most similar match to the read is then considered the most likely original ‘source’ organism of that sequence. In this article we will also refer to ‘taxonomic profilers’. We consider these as classifiers that also try to infer sequence abundance (i.e. re-assignment of counts to the most likely source based on the distribution of other hits) or biological relative abundance of the organism in the original sample (by coverage of expected marker genes, copy number estimations etc.), in addition to the simple read classification (Nayfach and Pollard 2016). We will use classifiers and profilers interchangeably throughout the publication.

Having to identify the original source of the many DNA sequences out of the many reference genomes in a time and computationally efficient manner is a difficult problem. In many cases biologists are not just interested as to which organism of each DNA sequence comes from, but also in using this information to infer the original ‘cellular’ (or natural) abundance of each organism of the given environment - something that is very difficult due to the biases inherent to DNA extraction and sequencing. Therefore a plethora of tools have been developed to address these challenges, all with their own biases and specific contexts (Sczyrba et al. 2017; Meyer et al. 2022). Furthermore,

78 each tool often produces tool-specific output formats making it difficult to efficiently
79 cross compare results. Thus, no established ‘gold standard’ classifier tool or method
80 currently exists.

81 One solution to addressing the problem of choice among the range of different tools
82 is to run all of them in parallel, and cross compare the results. This can be useful both
83 for benchmarking studies (e.g. Sczyrba et al. 2017; Meyer et al. 2022), but also to
84 build consensus profiles whereby confidence of a particular taxonomic identification
85 can be increased when it is detected by multiple tools (McIntyre et al. 2017; Ye et al.
86 2019).

87 A second challenge in taxonomic classification (and arguably a larger one) is a ques-
88 tion of databases. As with tools, there is no one set ‘gold standard’ database. Different
89 questions and contexts require different databases, such as when a researcher wants
90 to search for both bacterial and viral species in samples, but as an extension of this,
91 taxonomic classifiers often will need different settings for each database. Further-
92 more, as genomic sequencing becomes cheaper and more efficient, the number of
93 publicly available reference genomes is rapidly increasing (Nasko et al. 2018). Conse-
94 quently, the size of reference databases of taxonomic classifiers is also growing, often
95 outpacing the computational capacity available to researchers. In fact, while this was
96 one of the main motivations behind classifiers such as Kraken2 (Wood, Lu, and Lang-
97 mead 2019), these algorithmic techniques are already becoming insufficient (Wright,
98 Comeau, and Langille 2023).

99 Finally, with the decrease of costs, the possibility for larger and larger metagenomic
100 sequencing datasets increases, leading to increasing sample sizes in studies. This is
101 exemplified by the doubling of the number of metagenomes on the European Bioin-
102 formatic Institute’s MGnify database within just two years (Mitchell et al. 2019).

103 Altogether this highlights the need for methods to efficiently profile many samples
104 using many tools. Manually setting up bioinformatic jobs for classification tasks for
105 each database and settings against different tools on traditional academic computing
106 infrastructure (e.g. high performance computing clusters or ‘HPC’ clusters) can be
107 very tedious. Additionally, particularly for very large sample sets, there is increas-
108 ing use of cloud platforms that have greater scalability than traditional HPCs. Being
109 able to reliably and reproducibly execute taxonomic classification tasks across infras-
110 tructure with minimal intervention would therefore be a boon for the metagenomics
111 field.

112 In recent years, workflow managers such as Nextflow (Di Tommaso et al. 2017) or
113 Snakemake (Mölder et al. 2021) have become highly popular in bioinformatics. These
114 frameworks provide for developers robust workflow execution with different HPC
115 scheduling tools and software provisioning systems, ensuring maximum portability
116 and efficient in different computational contexts. While a range of metagenomic
117 pipelines already exist (a non-exhaustive list being for example, StaG-mwc by
118 Boulund et al. 2023; MetaMeta by Piro, Matschkowski, and Renard 2017; TAMA by
119 Sim et al. 2020; UGENE by Rose et al. 2019; and Sunbeam by Clarke et al. 2019), few
120 leverage workflow managers to make multi-step workflows easier to use in HPC or

cloud infrastructure. Furthermore, often these pipelines aim to carry out multiple different types of metagenomic analyses (e.g. also performing functional or assembly analyses, such as Morais et al. 2022; Boulund et al. 2023) of which each step has fewer options of tools and may execute functionality unwanted by the end user.

Here we present nf-core/taxprofiler (<https://nf-co.re/taxprofiler>), a pipeline designed to allow users to efficiently and simultaneously taxonomically classify or profile short- and long-read sequencing data. At the time of writing it supports 11 classifiers and an arbitrary number of databases per classifier 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. It includes detailed documentation and a graphical-user-interface (GUI) execution interface in addition to a standard command-line-interface (CLI).

3 Description

nf-core/taxprofiler aims to facilitate three main steps of a typical whole-genome, metagenomic sequencing analysis workflow (Chiu and Miller 2019, Figure 1). A longer description of the available functionality and motivations can be seen in the [Supplementary Information](#).

In brief, nf-core/taxprofiler can accept short- (e.g. Illumina) and/or long-read (e.g. Nanopore) FASTQ or FASTA files. These are supplied to the pipeline in the form of a TSV file that includes basic sample and sequencing library metadata. The pipeline can then be executed either via a standard Nextflow command-line-interface execution or graphical-user-interface through either the open-source and free nf-core launch page (<https://nf-co.re/launch>) or the commercial (with free-tier) Nextflow tower (<https://tower.nf>) solution. Examples of the command-line execution and nf-core launch GUI can be seen in the [Supplementary Information](#).

The pipeline can perform a range of metagenomics appropriate read preprocessing steps, such as adapter removal, read merging, low-sequence complexity filtering, host- or contamination removal, and/or per-sample run merging. All of these steps are optional, and are aimed at removing possible sequencing artefacts that may result in false positive taxonomic classification hits or improve classification efficiency. Most of these steps also provide options of different tools to account for user preference.

After pre-processing, nf-core/taxprofiler can perform simultaneous profiling of pre-processed reads with up to as many as 11 different taxonomic classifiers or profilers (Table 1). Additionally on top of this, also simultaneously for each of the classifiers, an arbitrary number of databases as supplied by the user. 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 (Vågane 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), and

162 KMCP (Shen et al. 2023). Databases are also supplied via a input TSV file, which
 163 also allows per-database custom classification parameters - meaning a given database
 164 can be supplied multiple times each with different parameters or multiple different
 165 databases per profiler. All classifiers with secondary steps to generate or convert to
 166 additional output file formats are also included.

167 Post-processing of taxonomic profiles include standardisation and aggregation of pro-
 168 files , i.e. merging of multiple profiles into a single multi-sample table for easier com-
 169 parison between profilers, with the tool TAXPASTA (Beber et al. 2023), and visualisa-
 170 tion of profiles with Krona (Ondov, Bergman, and Phillippy 2011) where supported.

171 All relevant preprocessing statistics are displayed in an interactive and dynamic Mul-
 172 tiQC report (Ewels et al. 2020).

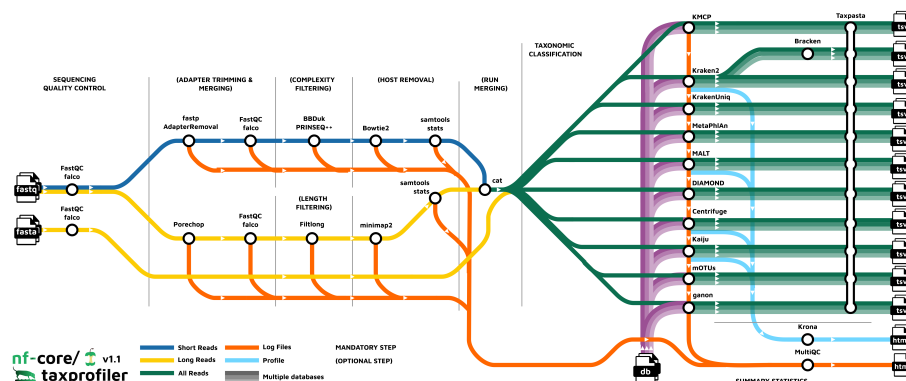


Figure 1: Visual overview of the nf-core/taxprofiler workflow. nf-core/taxprofiler can take in FASTQ (short or long reads) or FASTA files (long reads), that will optionally go through sequencing quality control (e.g. with FastQC), read preprocessing (e.g. removal of adapters), complexity filtering, host 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.

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. 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. Sequencing matching type refers to which ‘molecular alphabet’ is primarily used for matching between a query (read) and a reference (genome/gene).

Tool	Primary Algorithm	Reference Type	Sequence Matching Type
Kraken2	k-mer based	whole-genome	Nucleotide
Kaiju	k-mer based	whole-genome	Amino Acid

Tool	Primary Algorithm	Reference Type	Sequence Matching Type
Bracken	k-mer based	whole-genome	Nucleotide
KrakenUniq	k-mer based	whole-genome	Nucleotide
ganon	k-mer based	whole-genome	Nucleotide
KMCP	k-mer based	whole-genome	Nucleotide
MALT	alignment based	whole-genome	Nucleotide/Amino Acid
DIAMOND	alignment based	whole-genome	Amino Acid
Centrifuge	alignment based	whole-genome	Nucleotide
MetaPhlAn	alignment based	marker-gene	Nucleotide
mOTUS	alignment based	marker-gene	Nucleotide

173 nf-core/taxprofiler comes with extensive documentation for general usage, short- and
 174 long- parameter help texts, and output file descriptions. To ensure maximum accessi-
 175 bility, these are available in pipeline results as markdown files ([https://github.com/nf-
 176 core/taxprofiler](https://github.com/nf-core/taxprofiler)), on the nf-core website (<https://nf-co.re/taxprofiler>) and for the pa-
 177 rameter help texts on the command line via standard --help. The output documen-
 178 tation also aims to guide users as the most suitable files for different types of down-
 179 stream analysis

180 **4 Discussion**

181 A range of pipelines already exists for taxonomic profiling, however, each have
 182 their own particular purpose and capabilities. We compared the functionality
 183 of nf-core/taxprofiler against four other recently published or released pipelines,
 184 selected based on their similarity of purpose to nf-core/taxprofiler. The selection
 185 criteria and a more detailed comparison between the five pipelines can be seen
 186 in the [Supplementary Information](#). Overall, while there was a general similarity
 187 across all pipelines, nf-core/taxprofiler showed the largest number of options for
 188 pipeline execution accessibility, and user choice. This is facilitated through the
 189 use of an established workflow manager (with Nextflow supporting 7 software
 190 environment/container systems), support for both CLI and GUI execution, and by the
 191 number of supported classifiers. Furthermore, it is unique in that is the only pipeline
 192 to support supplying multiple database for all of the tools in a single pipeline run.

Table 2: Comparison of functionality with four recent taxonomic pipelines with similar functionality. A more detailed textual comparison can be found in the [Supplementary Information](#). Category keys are as follows: I - Information, R - Reproducibility, A - Accessibility, P - Portability, S - Scalability, F - Functionality.

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
I	Source code URL	https://github.com/ctmrbio/stag-mwc	https://github.com/sunbeam-labs/sunbeam	https://github.com/ugeneunipro/ugene	https://github.com/kimlab/TAMA	https://github.com/nf-core/taxprofiler/
I	Evaluated version	0.7.0	4	48	githash: 3a22c8f	1.1.0
I	Last release date	2023-06-13	2023-08-08	2023-08-08	2022-03-02	2023-09-19
I	Publication year	Unpublished	2019	2019	2020	This publication
I	Publication DOI	Unpublished	10.1186/s40168-019-0658-x	10.1093/bioinformatics/bt184	10.1093/bioinformatics/bt184	This publication
R	Pipeline versioning	Yes	Yes	Yes	No	Yes
R	Software versioning	Yes	Yes	Yes	Yes	Yes
R	Nr. software environments or container engines supported	2	2	0	1	7
A	Installation documentation	Yes	Yes	Yes	Yes	Yes
A	Usage documentation	Yes	Yes	Yes	Yes	Yes
A	Output documentation	Yes	Yes	Yes	Yes	Yes
A	CLI execution interface	Yes	Yes	No	Yes	Yes
A	GUI execution interface	No	No	Yes	No	Yes

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
A/S	Integration a scheduling systems	Yes	Yes	No	No	Yes
P/A	Nr. supported operating systems	2	1	3	1	2
P	Local machine integration	Yes	Yes	Yes	Yes	Yes
P/S	HPC scheduler integration	Yes	Yes	No	No	Yes
P/S	Cloud computing integration	Unsure	Unsure	No	No	Yes
P/S	Integration with multiple scheduling systems	Partial	Partial	No	No	Yes
S	Per-process resource optimisation	Yes	Yes	Yes	No	Yes
F	Short read support	Yes	Yes	Yes	Yes	Yes
F	Long read support	No	No	Yes	No	Yes
F	Read preprocessing	Yes	Yes	Yes	Yes	Yes
F	Sequencing depth estimation	Yes	No	No	No	No
F	Complexity filtering	No	Yes	No	No	Yes
F	Host removal	Yes	Yes	Partial	No	Yes
F	Nr. supported taxonomic classifiers/profilers	7	3	3	3	11
F	Graphical run reports	Yes	No	No	No	Yes
F	Standardised profiles	No	No	No	Yes	Yes

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
F	Multiple database supported	Partial	No	No	No	Yes
F	Metagenomic assembly	No	Yes	No	No	No
F	Visualisation	No	No	No	No	Partial

Another important advantage of nf-core/taxprofiler is that it is being developed within the nf-core community (<https://nf-co.re>), that provides strong long-term support for the continued community-based development and maintenance of its pipelines. In this framework, we will continue to add additional preprocessing, metagenomic classification, and profiling tools as they become established and as requested by the metagenomics community. For example, we feel that the inclusion of steps such as sequencing saturation estimation as already being performed by a similar pipeline StaG-mwc (<https://github.com/ctmrbio/stag-mwc>) would be beneficial to the nf-core/taxprofiler workflow (possibly with dedicated tools such as Nonpareil, Rodriguez-R et al. 2018), and/or more performant complexity filtering tools such as Komplexity as offered by the sunbeam metagenomics pipeline (Clarke et al. 2019). Additional tools that could be added for short-read classification could include sourmash (Titus Brown and Irber 2016) that provides scalable sequence to sequence comparison or other marker gene reference tools such as tools such as METAXA2 (Bengtsson-Palme et al. 2015) that use shotgun sequencing reads to recover 16S sequences from metagenomic samples. Adding additional classifiers 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 are typically 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,

227 this happens in-parallel to the taxonomic profiling and requires prior expectation of
228 which reference genomes to map against. Instead, nf-core/taxprofiler could be eas-
229 ily extended to have a validation step similar to the approach of the ancient DNA
230 metagenomic pipeline aMeta (Pochon et al. 2022). Utilising Nextflow’s execution par-
231 allelism, the input sequences could be aligned back to the reference genomes of only
232 those species with hits resulting from the taxonomic classification, but with dedicated
233 accurate short- or long-read aligners. In addition to the more precise classification,
234 post-classification read-alignment could also be particularly useful for researchers in
235 palaeogenomics who wish to use tools other than KrakenUniq for initial classification
236 (as in aMeta), where alignment information can be used to authenticate ancient DNA
237 within their samples, but also in clinical metagenomics to identify potential pathogens
238 at much finer resolution (e.g. down to strain level).

239 Another motivation for developing nf-core/taxprofiler, despite the large number of ex-
240 isting metagenomics pipelines, is that by establishing a taxonomic profiling pipeline
241 within the nf-core ecosystem, it is possible to begin building both standalone but
242 also an integrated suite of powerful interconnected pipelines for the major stages
243 of metagenomic workflows. Existing microbial- and metagenomics- related pipelines
244 within the nf-core initiative include nf-core/ampliseq (Straub et al. 2020), nf-core/mag
245 (Krakau et al. 2022), and nf-core/funcscan (<https://nf-co.re/funcscan>). We expect over
246 time the ability to link inputs and outputs of each workflow to develop comprehensive
247 metagenomic analyses, while still maintaining powerful standalone pipelines, provid-
248 ing maximal user choice but with familiar interfaces.

249 5 Conclusion

250 nf-core/taxprofiler is an accessible, efficient, and scalable pipeline for metagenomic
251 taxonomic classification and profiling that can be executed on anywhere from laptops
252 to the cloud. To our knowledge, the pipeline offers the largest number of taxonomic
253 profilers across similar pipelines, providing flexibility for users not just on choice of
254 profiling tool but also with databases and database settings within a single run. With
255 the development within the open and welcoming nf-core community and with best-
256 practise development infrastructure, we look forward to further contributions and in-
257 volvement of the wider metagenomics community, and also we hope that through de-
258 tailed documentation and a range of execution options, nf-core/taxprofiler will make
259 reproducible and high-throughput metagenomics more accessible for a wide range of
260 disciplines.

261 6 Code Availability

262 nf-core/taxprofiler source code is available on GitHub at [https://github.com/nf-core/](https://github.com/nf-core/taxprofiler)
263 [taxprofiler](https://github.com/nf-core/taxprofiler), and each release is archived on Zenodo (latest version DOI: [10.5281/zen-](https://doi.org/10.5281/zenodo.7728364)
264 [odo.7728364](https://doi.org/10.5281/zenodo.7728364))

265 The version of the pipeline described in this paper is version 1.1.0 (release specific

266 Zenodo archive DOI: [10.5281/zenodo.8358147](https://doi.org/10.5281/zenodo.8358147))

267 **7 Acknowledgments**

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

272 **8 Funding**

273 S.S. and L.A-L. were supported by Rapid establishment of comprehensive laboratory
274 pandemic preparedness – RAPID-SEQ. This material is based upon work supported by
275 the U.S. Department of Agriculture, Agricultural Research Service, under agreement
276 No. 58-3022-0-001 (T.A.C II). M.B. and J.A.F.Y were supported by the Max Planck So-
277 ciety. M.B. was supported by the Deutsche Forschungsgemeinschaft (DFG, German
278 Research Foundation) under Germany’s Excellence Strategy – EXC 2051 – Project-ID
279 390713860 (Balance of the Microverse). J.A.F.Y was supported by the Werner Siemens-
280 Stiftung (“Paleobiotechnology”, Awarded to Prof. Pierre Stallforth and Prof. Christina
281 Warinner).

282 9 Supplementary Information

283 9.1 Implementation

284 9.1.1 Input and Execution

285 The pipeline can be executed via typical Nextflow commands (Code Block 1), or using the standard nf-core 'launch' GUI (Figure 2), making the pipeline accessible for
286 both computationally experienced as well as less experienced 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
287 what each parameter does, both in short- and long-form, with long-form parameter descriptions additionally describing which tool-specific parameters are being modified for each pipeline parameter (<https://nf-co.re/launch/?pipeline=taxprofiler>). The
288 GUI also includes controlled user input by providing strict drop-down lists and input validation prior execution of the pipeline (Figure 2) to reduce the risk of typos and
289 other mistakes, which is in contrast to the command-line interface that only includes validation at pipeline run-time.

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 MetaPhlAn 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_metaphlan \  
  --run_krona \  
  --run_profile_standardisation
```

297 An example nf-core command line execution of the pipeline can be seen in Code
298 Block 1, where two input files are supplied: one file specifying paths of FASTQ files
299 of metagenomic samples and necessary metadata for preprocessing (such as sample
300 ID and sequencing platform), and the second file specifying paths to the user-defined
301 databases with per-database classification parameters. Various parameters are available to select different preprocessing steps, and provide additional configuration such
302 as tool selection and value options. Note that even if a user supplies a given database
303 in the database input sheet, the corresponding profiling tool must still be activated
304

Preprocessing short-read QC options

Launch

--shortread_qc_minlength

15

?

Specify the minimum length of reads to be retained

Specifying a minimum 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: --min length

--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_entropy

?

Specify the minimum sequence entropy level for complexity filtering

--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

305 with the corresponding pipeline parameter (e.g. `--run_kraken2`). Per-classifier flags
306 are also available for the optional saving of additional non-profile output files. Alter-
307 natively to command line flags, parameters can be specified via pre-configured YAML
308 format files, with which (provided no hardcoded paths are included) can be re-used
309 across pipeline runs.

310 All nf-core pipelines are strictly versioned (specified with the Nextflow `-r` flag), and to
311 ensure reproducibility, each version of the pipeline has a fixed set of software used for
312 each step of the pipeline. The fixed set of software are controlled through the use of
313 the conda package manager or containers (Docker, or Apptainer - previously known
314 as Singularity, etc) from the stable Bioconda (Grüning et al. 2018) or BioContainers
315 (Veiga Leprevost et al. 2017) repositories. This, coupled with the intrinsic Nextflow
316 ability to execute on most infrastructure whether that is a local laptop (resource re-
317 quirements permitting), traditional HPC, as well across common cloud providers also
318 makes nf-core/taxprofiler a very portable pipeline that can be used in many contexts.

319 9.1.2 Preprocessing

320 Preprocessing steps in nf-core/taxprofiler are aimed at removing laboratory and se-
321 quencing artefacts that may influence taxonomic profiling, either for computing re-
322 source consumption or and/or false-positive or false-negative classification reasons.
323 First sequencing quality control with FastQC (Andrews 2010) or Falco (Sena Brandine
324 and Smith 2021) is carried out. Falco was included for reduced memory requirements,
325 in particular for long read sequencing data. Artificial library adapter sequences added
326 during sequencing reduce sequencing matching accuracy by reducing sequence speci-
327 ficity, and in some cases, may result in false-positive hits due to adapter sequence con-
328 tamination in reference genomes (Schäffer et al. 2018; F. P. Breitwieser, Baker, and
329 Salzberg 2018)¹. Additionally, paired-end merging may provide longer sequences
330 that will allow for more specific classification when paired-end alignment is not sup-
331 ported by a given classifier. For these tasks nf-core/taxprofiler can apply either fastp
332 (Chen et al. 2018) or AdapterRemoval2 (Schubert, Lindgreen, and Orlando 2016) for
333 short reads, and currently Porechop (Wick et al. 2017) for Oxford Nanopore long-read
334 data. For both short and long reads, FastQC or Falco is run again to allow assessment
335 on the performance of the adapter removal and/or pair-merging step.

336 Low complexity sequences, e.g. sequences containing long stretches of mono- or
337 di-nucleotide repeats provide little specific genetic information that contribute to
338 taxonomic identification, as they can align to many different reference genomes
339 (Schmieder and Edwards 2011; Clarke et al. 2019). Including such reads during
340 taxonomic profiling can increase run-time and memory usage for little gain, as
341 during lowest-common-ancestor (LCA) classification steps they will be assigned to

¹For 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=1> why-you-should-qc-your-reads-and-your.html
Sixing Huang: <https://web.archive.org/web/20220904205331/https://dgg32.medium.com/carp-in-the-soil-1168818d2191>

(Accessed 2023-08-25)

high-level taxonomic ranks (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 desirable 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). Therefore, the removal of such sequences can help 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 multiple FASTQ files from the same sample but have been sequenced over multiple lanes are concatenated together 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.

9.1.3 Profiling

There are many types of metagenomic profiling techniques, from profiling against whole-genome references with alignment or k-mer based approaches, to methods involving alignment to species-specific marker-gene families (Quince et al. 2017; 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.

The following classifiers and profilers are supported in version 1.1.0 of nf-core/taxprofiler: 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),

384 Kaiju (Menzel, Ng, and Krogh 2016), MetaPhlAn (Blanco-Míguez et al. 2023), mOTUs
385 (Ruscheweyh et al. 2022), ganon (Piro et al. 2020), KMCP (Shen et al. 2023).

386 By default, nf-core/taxprofiler produces the default per-sample taxonomic classifica-
387 tion profile output from a tool or a tool’s report generation tool. The output is nor-
388 mally in the form of counts per reference sequencing, with additional statistics about
389 the hits of a particular organism (estimated sequence abundance, taxonomic level etc.).
390 Users can also optionally request output of per-read classification output and output
391 such as classified and unclassified reads in FASTQ format, where supported.

392 The pipeline provides high efficiency, particularly during the metagenomic classifica-
393 tion stage, through the inherent parallelisation provided by Nextflow. While metage-
394 nomic classification is comparatively computationally intensive (in terms of mem-
395 ory and execution time; due to a combination of sequencing depth and number of
396 reference genomes), Nextflow automatically optimises the execution order of all the
397 steps in pipeline, maximising the number parallel running of multiple profilers and/or
398 databases at any given time point, as far as the available computational resources al-
399 low. For local machines such as laptops or desktops, Nextflow will automatically
400 detect all available computational resources, but this is customisable using Nextflow
401 configuration files. For HPC and cloud infrastructure, users typically have to define
402 the computational infrastructural environment the pipeline is being executed on (CPU
403 or memory limitations, queues, instance types, etc.). To facilitate the pipeline compu-
404 tational configuration, nf-core/taxprofiler supports use of more than 90 pre-defined
405 centralised generic and pipeline-specific institutional Nextflow configurations as pro-
406 vided by nf-core/configs (<https://nf-co.re/configs>). However, of course users are still
407 welcome to supply their own custom configuration files as with any typical Nextflow
408 run, further refining computational limitations or execution specifications.

409 **9.1.4 Post-profiling**

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

415 One of the challenges that researchers face when comparing multiple taxonomic clas-
416 sifiers or profilers is the heterogenous output formats that are produced, that often
417 require custom parsing and merging scripts for each tool to standardise. To facilitate
418 more user-friendly cross-comparisons between tools, nf-core/taxprofiler utilises the
419 TAXPASTA tool (Beber et al. 2023) to generate standardised profiles and generate
420 multi-sample tables.

421 Summary statistics for the entire pipeline are visualised and displayed in a customis-
422 able MultiQC report (Ewels et al. 2020). When supported, quality control of data and
423 pipeline runs are shown for manual verification. Krona plots (Ondov, Bergman, and
424 Phillippy 2011) can also optionally be generated for supported tools to help provide
425 further visualisation of taxonomic profiles.

426 9.1.5 Output

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

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

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

440 9.2 Comparison with other solutions

441 nf-core/taxprofiler has been specifically developed for the analysis of whole-genome,
442 *metagenomic* sequencing data. While other types of taxonomic profiling data such
443 as 16S amplicon sequencing are well established fields with a range of popular high-
444 quality and best-practise tools pipelines (e.g. Blanco-Míguez et al. 2023; Schloss et
445 al. 2009) and databases (DeSantis et al. 2006; Yilmaz et al. 2014), ‘gold standard’
446 tools and databases for metagenomics remain much less established. Thus, the need
447 for highly-multiplexed classification is more desirable for the newer metagenomics
448 methods.

449 We searched Google Scholar for open-source pipelines published or released in the last
450 5 years (at the time of writing, since 2018) that were designed primarily for metage-
451 nomic classification screening, that supported at least 2 classifiers, had at least one
452 preprocessing step and were not specifically targeted at read classification of spe-
453 cific domains of taxa (e.g. viruses or bacteriophages only). We also included an addi-
454 tional open-source but unpublished pipeline at the recommendations of the authors
455 of the pipeline due to the functional overlap to nf-core/taxprofiler. We then evalu-
456 ated the pipelines based on their publications and documentation for typical metage-
457 nomic profiling workflow steps. We used a range of criteria related to expectations of
458 modern bioinformatic workflows that can be summarised in the following four cate-
459 gories: reproducibility, accessibility, scalability, and portability (Wratten, Wilm, and
460 Göke 2021). After searching, we selected the following pipelines for comparison with
461 nf-core/taxprofiler that matched the specific criteria described above: sunbeam (v4,
462 Clarke et al. 2019), Unipro UGENE (v48, Rose et al. 2019), TAMA (githash: 3a22c8f,
463 Sim et al. 2020), and StaG-mwc (0.7.0, Boulund et al. 2023).

464 In terms of accessibility, all pipelines have documentation describing the installation
465 steps, usage instructions, and output files. However, there are varying levels of de-
466 tail 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 which 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 overlap with accessibility, as it implies options for and ease of different users running on different types of computing infrastructure, whether that is on their own laptop, on an HPC cluster, or in the cloud. Unipro UGENE is the only pipeline that explicitly states support for execution on all three major operating systems (Linux, OSX, Windows), whereas StaG-mwc and nf-core/taxprofiler can be run on unix operating systems (albeit possibly on Windows via Windows Subsystem for Linux (WSL)), 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 criterion for running large multi-step and parallelised pipelines. The three pipelines leveraging workflow managers (Snakemake 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 for users in terms of which HPC infrastructure they can execute their pipeline on. As an extension of this, only nf-core/taxprofiler has explicit support for cloud computing (e.g. AWS, GCP, or Microsoft Azure) as provided by Nextflow, again maximising user choice and portability when it comes to running the pipeline.

In terms of scalability, the aforementioned integration with schedulers and cloud computing support implicitly maximises efficiency and parallelisation 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. Furthermore, 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 software² (Di Tommaso et al. 2017). The current most

²As 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)

popular solution to the problem of inconsistent software environments is to use container engines such as Docker or Apptainer to run container images which are isolated, deterministic computing environments which can be executed by any system providing a container runtime. Only Unipro UGENE does not document the use of a container system, with nf-core/taxprofiler offering the biggest choice for users, again, courtesy of Nextflow with 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 sunbeam, the authors developed their own dedicated, performant complexity filtering tool Komplexity (Clarke et al. 2019). Most pipelines support some form of host removal (only TAMA did not support this), and it is likely possible with Unipro UGENE (although not directly described). In all cases, host removal consists of mapping processed reads with an aligner and using the off-target reads for downstream profiling (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 classifiers 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 support 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. This makes it a 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, the former with the dedicated standalone tool TAXPASTA (Beber et al. 2023). However Unipro UGENE additionally offers a ‘consensus’ profile 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 framework they feel most comfortable with. However the advantages of nf-core/taxprofiler mainly come from the offering of the greatest choice of tools, as well the particular benefits provided by Nextflow. It provides the greatest number of computational infrastructure types the pipeline can be executed on, and container systems can be used to ensure reproducibility, as well 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 over

552 time to add new tools classifiers.

553 The functionality offered by other pipelines not currently supported by nf-
554 core/taxprofiler include sequencing saturation estimation (StaG-mwc), taxonomy-
555 free composition comparison (StaG-mwc), functional profiling (StaG-mwc), *de novo*
556 assembly (sunbeam), and reference mapping (StaG-mwc, sunbeam). We do not
557 plan to support *de novo* assembly or functional profiling in nf-core/taxprofiler as
558 we feel these are already better served by other existing dedicated pipelines within
559 the nf-core ecosystem: nf-core/mag for *de novo* assembly, (Krakau et al. 2022)
560 and nf-core/funcscan for functional profiling (<https://nf-co.re/funcscan>), as well as
561 elsewhere e.g. MetaWRAP (Uritskiy, DiRuggiero, and Taylor 2018).

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. "Ctmbio/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 Perte, 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-Fadros, 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 Gourel, 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>.
- 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>.

699 Morais, Diego A A, João V F Cavalcante, Shênia S Monteiro, Matheus A B Pasquali,
700 and Rodrigo J S Dalmolin. 2022. "MEDUSA: A Pipeline for Sensitive Taxonomic
701 Classification and Flexible Functional Annotation of Metagenomic Shotgun Se-
702 quences." *Frontiers in Genetics* 13 (March): 814437. [https://doi.org/10.3389/fgene.](https://doi.org/10.3389/fgene.2022.814437)
703 [2022.814437](https://doi.org/10.3389/fgene.2022.814437).

704 Nasko, Daniel J, Sergey Koren, Adam M Phillippy, and Todd J Treangen. 2018. "Ref-
705 Seq Database Growth Influences the Accuracy of k-Mer-Based Lowest Common
706 Ancestor Species Identification." *Genome Biology* 19 (1): 165. [https://doi.org/10.](https://doi.org/10.1186/s13059-018-1554-6)
707 [1186/s13059-018-1554-6](https://doi.org/10.1186/s13059-018-1554-6).

708 Nayfach, Stephen, and Katherine S Pollard. 2016. "Toward Accurate and Quantitative
709 Comparative Metagenomics." *Cell* 166 (5): 1103–16. [https://doi.org/10.1016/j.cell.](https://doi.org/10.1016/j.cell.2016.08.007)
710 [2016.08.007](https://doi.org/10.1016/j.cell.2016.08.007).

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

714 Piro, Vitor C, Temesgen H Dadi, Enrico Seiler, Knut Reinert, and Bernhard Y Renard.
715 2020. "Ganon: Precise Metagenomics Classification Against Large and up-to-Date
716 Sets of Reference Sequences." *Bioinformatics (Oxford, England)* 36 (Suppl_1): i12–
717 20. <https://doi.org/10.1093/bioinformatics/btaa458>.

718 Piro, Vitor C, Marcel Matschkowski, and Bernhard Y Renard. 2017. "MetaMeta: Inte-
719 grating Metagenome Analysis Tools to Improve Taxonomic Profiling." *Microbiome*
720 5 (1): 101. <https://doi.org/10.1186/s40168-017-0318-y>.

721 Pochon, Zoé, Nora Bergfeldt, Emrah Kirdök, Mário Vicente, Thijessen Naidoo, Tom
722 van der Valk, N Ezgi Altınışık, et al. 2022. "aMeta: An Accurate and Memory-
723 Efficient Ancient Metagenomic Profiling Workflow." *bioRxiv*. [https://doi.org/10.](https://doi.org/10.1101/2022.10.03.510579)
724 [1101/2022.10.03.510579](https://doi.org/10.1101/2022.10.03.510579).

725 Portik, Daniel M, C Titus Brown, and N Tessa Pierce-Ward. 2022. "Evaluation of
726 Taxonomic Classification and Profiling Methods for Long-Read Shotgun Metage-
727 nomic Sequencing Datasets." *BMC Bioinformatics* 23 (1): 541. [https://doi.org/10.](https://doi.org/10.1186/s12859-022-05103-0)
728 [1186/s12859-022-05103-0](https://doi.org/10.1186/s12859-022-05103-0).

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

732 Rodriguez-R, Luis M, Santosh Gunturu, James M Tiedje, James R Cole, and Konstanti-
733 nos T Konstantinidis. 2018. "Nonpareil 3: Fast Estimation of Metagenomic Cov-
734 erage and Sequence Diversity." *mSystems* 3 (3). [https://doi.org/10.1128/mSystems.](https://doi.org/10.1128/mSystems.00039-18)
735 [00039-18](https://doi.org/10.1128/mSystems.00039-18).

736 Rose, Rebecca, Olga Golosova, Dmitrii Sukhomlinov, Aleksey Tiunov, and Mattia
737 Proserpi. 2019. "Flexible Design of Multiple Metagenomics Classification
738 Pipelines with UGENE." *Bioinformatics (Oxford, England)* 35 (11): 1963–65.
739 <https://doi.org/10.1093/bioinformatics/bty901>.

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

744

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

749 Schloss, Patrick D, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hart-
750 mann, Emily B Hollister, Ryan A Lesniewski, et al. 2009. "Introducing Mothur:
751 Open-Source, Platform-Independent, Community-Supported Software for De-
752 scribing and Comparing Microbial Communities." *Applied and Environmental*
753 *Microbiology* 75 (23): 7537–41. <https://doi.org/10.1128/AEM.01541-09>.

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

756 Schubert, Mikkel, Stinus Lindgreen, and Ludovic Orlando. 2016. "AdapterRemoval v2:
757 Rapid Adapter Trimming, Identification, and Read Merging." *BMC Research Notes*
758 9 (February): 88. <https://doi.org/10.1186/s13104-016-1900-2>.

759 Sczyrba, Alexander, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen,
760 Johannes Dröge, Ivan Gregor, et al. 2017. "Critical Assessment of Metagenome
761 Interpretation-a Benchmark of Metagenomics Software." *Nature Methods* 14 (11):
762 1063–71. <https://doi.org/10.1038/nmeth.4458>.

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

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

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

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

775 Straub, Daniel, Nia Blackwell, Adrian Langarica-Fuentes, Alexander Peltzer, Sven
776 Nahnsen, and Sara Kleindienst. 2020. "Interpretations of Environmental Micro-
777 bial Community Studies Are Biased by the Selected 16S rRNA (Gene) Amplicon
778 Sequencing Pipeline." *Frontiers in Microbiology* 11 (October): 550420. <https://doi.org/10.3389/fmicb.2020.550420>.

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

782 Titus Brown, C, and Luiz Irber. 2016. "Sourmash: A Library for MinHash Sketching
783 of DNA." *Journal of Open Source Software* 1 (5): 27. <https://doi.org/10.21105/joss.00027>.

784 Uritskiy, Gherman V, Jocelyne DiRuggiero, and James Taylor. 2018. "MetaWRAP-a
785 Flexible Pipeline for Genome-Resolved Metagenomic Data Analysis." *Microbiome*

6 (1): 158. <https://doi.org/10.1186/s40168-018-0541-1>.

Vågane, 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, André 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>.