

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

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

<sup>1</sup>Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

<sup>1</sup>Department of Clinical Microbiology, Karolinska University Hospital

<sup>2</sup>Unseen Bio ApS

<sup>3</sup>Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

<sup>3</sup>Department of Clinical Microbiology, Karolinska University Hospital

<sup>4</sup>Veterinary Diagnostic Laboratory, Kansas State University College of Veterinary Medicine

<sup>5</sup>Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

<sup>5</sup>Department of Clinical Microbiology, Karolinska University Hospital

<sup>6</sup>Department of Archaeogenetics, Max Planck Institute for Evolutionary Anthropology

<sup>7</sup>Department of Microbiology, Tumor and Cell Biology, Karolinska Institutet

<sup>7</sup>Department of Clinical Microbiology, Karolinska University Hospital

<sup>8</sup>

<sup>9</sup>Department of Archaeogenetics, Max Planck Institute for Evolutionary Anthropology

<sup>9</sup>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, accommodating from

36 small to extremely large projects on a wide range of computing infrastructure. It has  
37 been developed following best-practise software development practises and commu-  
38 nity support to ensure longevity and adaptability of the pipeline, to help keep it up to  
39 date with the field of metagenomics.

## 40 2 Introduction

41 Whole-genome, metagenomic sequencing offers strong benefits to the taxonomic clas-  
42 sification of DNA samples over targeted approaches (Eloe-Fadrosh et al. 2016; Florian  
43 P. Breitwieser, Lu, and Salzberg 2019). While metabarcoding approaches targeting the  
44 16S rRNA or other marker genes are widely used due to low cost and large, diverse  
45 reference databases (Yilmaz et al. 2014; Lynch and Neufeld 2015), metagenomic ap-  
46 proaches have been gaining popularity with the increasingly lower costs of, for exam-  
47 ple, shotgun sequencing. These metagenomic analyses have been shown to provide  
48 a similar resolution on microbial genomes during taxonomic classification (Hillmann  
49 et al. 2018), with the added benefit of having greater reusability potential of the data,  
50 via whole genome reconstruction and also functional classification of metagenomics  
51 (Sharpton 2014; Quince et al. 2017).

52 Taxonomic classifiers (sometimes referred to as taxonomic bidders) aim to identify  
53 the original ‘taxonomic source’ of a given DNA sequence (Ye et al. 2019; Meyer et al.  
54 2022; Govender and Eyre 2022). In metagenomics, this typically consists of comparing  
55 millions of DNA reads (sequenced DNA molecules) against hundreds or thousands of  
56 reference genomes either via sequence alignment or ‘k-mer matching’ (Sharpton 2014;  
57 Sun et al. 2021), with the most close match being considered the most likely original  
58 ‘source’ organism of that sequence. We will also refer to taxonomic profilers that  
59 are classifiers that also try to infer *species* abundance of the organism in the original  
60 sample, in addition to the typical sequence abundance (Nayfach and Pollard 2016). We  
61 will use classifiers and profilers interchangeably throughout the publication.

62 Having to identify the original source of the many DNA sequences out of the many  
63 reference genomes, but in a time and computationally *efficient* manner, is a difficult  
64 problem. In many cases biologists are not just interested as to which organism of each  
65 DNA sequence comes from, but rather using this information to *infer* the original  
66 natural abundance of each organism of the given environment - something that is  
67 very difficult due to the biases inherent to DNA extraction and sequencing. Therefore  
68 a plethora of tools have been developed to address these challenges, all with their own  
69 biases and specific contexts (Sczyrba et al. 2017; Meyer et al. 2022). Furthermore,  
70 each tool often produces tool-specific output formats making it difficult to efficiently  
71 cross compare results. Thus, no established ‘gold standard’ classifier tool or method  
72 currently exists.

73 One solution to addressing the problem of choice among the range of different tools  
74 is to run all of them in parallel, and cross compare the results. This can be useful both  
75 for benchmarking studies (e.g. Sczyrba et al. 2017; Meyer et al. 2022), but also to  
76 build consensus profiles whereby confidence of a particular taxonomic identification

77 can be increased when it is detected by multiple tools (McIntyre et al. 2017; Ye et al.  
78 2019).

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

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

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

104 In reason years, workflow managers such as Nextflow (Di Tommaso et al. 2017) or  
105 Snakemake (Mölder et al. 2021) have become highly popular in bioinformatics. These  
106 frameworks provide for developers robust workflow execution with different HPC  
107 scheduling tools and software provisioning systems, ensuring maximum portability  
108 and efficient in different computational contexts. While a range of metagenomic  
109 pipelines already exist (a non-exhaustive list being for example, Boulund et al. 2023;  
110 Piro, Matschkowski, and Renard 2017; Sim et al. 2020; Rose et al. 2019; Clarke et al.  
111 2019), few leverage workflow managers to make multi-step workflows easier to use  
112 in HPC or cloud infrastructure. Furthermore, often these pipelines aim to carry out  
113 multiple different types of metagenomic analyses Boulund et al. (2023) of which each  
114 step has fewer options of tools and may be unwanted by the end user.

115 Here we present nf-core/taxprofiler, a pipeline designed to allow users to efficiently  
116 and simultaneously taxonomically classify and profile short- and long-read sequenc-  
117 ing data against (at the time of writing) 11 classifiers and databases in a single pipeline  
118 run. nf-core/taxprofiler utilises Nextflow (Di Tommaso et al. 2017) to ensure effi-  
119 ciency, portability, and scalability, and has been developed within the nf-core ini-

120 tiative of Nextflow pipelines (Ewels et al. 2020) to ensure high quality coding prac-  
121 tises and user accessibility, including detailed documentation and a graphical-user-  
122 interface (GUI) execution interface.

### 123 3 Description

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

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

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

142 After pre-processing, nf-core/taxprofiler can perform simultaneous profiling of  
143 preprocessing reads as many as 11 different taxonomic classifiers or profilers  
144 (Table 1), and on top of this, simultaneous for each of these an arbitrary number  
145 of databases supplied by the user. As of version 1.1.0, the following classifiers and  
146 profilers are available: Kraken2 (Wood, Lu, and Langmead 2019), Bracken (Lu et al.  
147 2017), KrakenUniq (F. P. Breitwieser, Baker, and Salzberg 2018), Centrifuge (Kim  
148 et al. 2016), MALT (Vågene et al. 2018), DIAMOND (Buchfink, Reuter, and Drost  
149 2021), Kaiju (Menzel, Ng, and Krogh 2016), MetaPhlAn (Blanco-Míguez et al. 2023),  
150 mOTUs (Ruscheweyh et al. 2022), ganon (Piro et al. 2020), KMCP (Shen et al. 2023).  
151 Databases are also supplied via a input TSV file, that also allows per-database custom  
152 classification parameters - meaning a given database can be supplied multiple times  
153 each with different parameters. All classifiers with secondary steps to generate or  
154 convert to additional output file formats are also included.

155 Post-processing of taxonomic profiles include standardisation and aggregation of pro-  
156 files, i.e., merging of multiple profiles into a single multi-sample table, for easier  
157 comparison between profilers with the tool TAXPASTA (Beber et al. 2023), and visu-  
158 alisation of profiles with Krona (Ondov, Bergman, and Phillippy 2011) for supported  
159 classifiers.

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

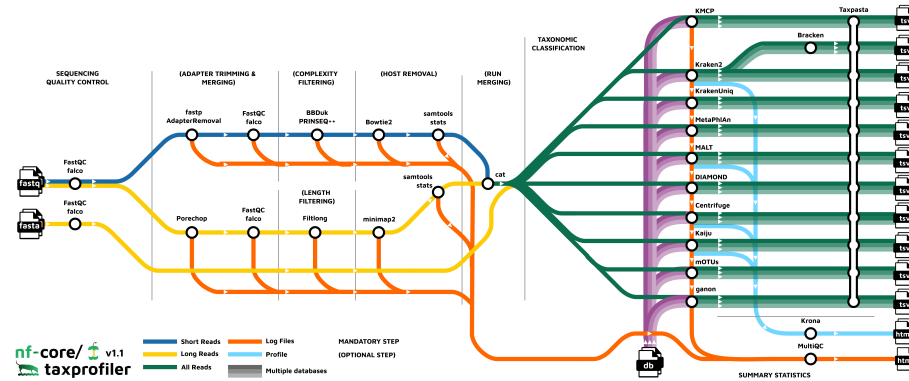


Figure 1: Visual overview of the nf-core/taxprofiler workflow. nf-core/taxprofiler can take in FASTQ (short or long reads) or FASTA files (long reads), that will optionally go through sequencing quality control (e.g. with FastQC), read preprocessing (e.g. removal of adapters), complexity filtering, host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

Tool	Primary Algorithm	Reference Type	Method	Sequence Matching Type
Kraken2	k-mer based	whole-genome	classifier	Nucleotide
Kaiju	k-mer based	whole-genome	classifier	Amino Acid
Bracken	k-mer based	whole-genome	profiler	Nucleotide
KrakenUniq	k-mer based	whole-genome	profiler	Nucleotide

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

162 nf-core/taxprofiler comes with extensive documentation for general usage, short- and  
 163 long- parameter help texts, and output file descriptions. To ensure maximum accessi-  
 164 bility, these are available in pipeline results as markdown files, on the nf-core website  
 165 and for the parameter help texts on the command line via standard `--help`. The out-  
 166 put documentation also aims to guide users as the most suitable files for different  
 167 types of downstream analysis

## 168 4 Discussion

169 A range of pipelines already exists for taxonomic profiling, however, each have  
 170 their own particular purpose and capabilities. We compared the functionality  
 171 of nf-core/taxprofiler against four other recently published or released pipelines,  
 172 selected based on their similarity of purpose to nf-core/taxprofiler. The selection  
 173 criteria and a more detailed comparison between the five pipelines can be seen in the  
 174 [Supplementary Information](#). Overall, while there was a general similarity across all  
 175 pipelines, nf-core/taxprofiler showed the largest number of functionality for pipeline  
 176 execution accessibility and user choice, through the use of an established workflow  
 177 manager (with Nextflow supporting 7 software environment/container systems),  
 178 supporting both CLI and GUI execution, and the number of supported classifiers.  
 179 Furthermore, it is unique in that is the only pipeline to support supplying multiple  
 180 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	<a href="https://github.com/ctmrbio/stag-mwc">https://github.com/ctmrbio/stag-mwc</a>	<a href="https://github.com/sunbeam-labs/sunbeam">https://github.com/sunbeam-labs/sunbeam</a>	<a href="https://github.com/ugeneunipro/ugene">https://github.com/ugeneunipro/ugene</a>	<a href="https://github.com/kimlab/TAMA">https://github.com/kimlab/TAMA</a>	<a href="https://github.com/nf-core/taxprofiler/">https://github.com/nf-core/taxprofiler/</a>
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	<a href="https://doi.org/10.1186/s40168-019-0658-x">10.1186/s40168-019-0658-x</a>	<a href="https://doi.org/10.1093/bioinformatics/bty184">10.1093/bioinformatics/bty184</a>	<a href="https://doi.org/10.1093/bioinformatics/bty184">10.1093/bioinformatics/bty184</a>	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 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

215 which reference genomes to map against. Instead, nf-core/taxprofiler could be eas-  
216 ily extended to have a validation step similar to the approach of the ancient DNA  
217 metagenomic pipeline aMeta (Pochon et al. 2022). Utilising Nextflow’s execution par-  
218 allelism, the input sequences could be aligned back to the reference genomes of only  
219 those species with hits resulting from the taxonomic classification, but with dedicated  
220 accurate short- or long-read aligners. In addition to the more precise classification,  
221 post-classification read-alignment could also be particularly useful for researchers in  
222 palaeogenomics who wish to use tools other than KrakenUniq for initial classification  
223 (as in aMeta), where alignment information can be used to authenticate ancient DNA  
224 within their samples, but also in clinical metagenomics to identify potential pathogens  
225 at much finer resolution (e.g. down to strain level).

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

## 236 5 Conclusion

237 nf-core/taxprofiler is an accessible, efficient, and scalable pipeline for metagenomic  
238 taxonomic classification and profiling that can be executed on anywhere from lap-  
239 tops to the cloud. To our knowledge, the pipeline offers the largest number of tax-  
240 onomic profilers across similar pipelines, providing flexibility for users not just on  
241 choice of profiling tool but also with databases and database settings, with any num-  
242 ber being able to be supplied to the pipeline in a single run. With the development  
243 within the open and welcoming nf-core community and with best-practise develop-  
244 ment infrastructure, we look forward to further contributions and involvement of the  
245 wider metagenomics community, and also we hope that through detailed documenta-  
246 tion and a range of execution options, nf-core/taxprofiler will make reproducible and  
247 high-throughput metagenomics more accessible for a wide range of disciplines.

## 248 6 Code Availability

249 nf-core/taxprofiler source code is available on GitHub at [https://github.com/nf-core/](https://github.com/nf-core/taxprofiler)  
250 [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)  
251 [odo.7728364](https://doi.org/10.5281/zenodo.7728364))

252 The version of the pipeline described in this paper is version (1.1.0) (release specific  
253 Zenodo archive DOI: [10.5281/zenodo.8358147](https://doi.org/10.5281/zenodo.8358147))

## 254 7 Acknowledgments

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

## 259 8 Funding

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

## 266 9 Supplementary Information

### 267 9.1 Implementation

#### 268 9.1.1 Input and Execution

269 The pipeline can be executed via typical Nextflow commands, or using the standard  
270 nf-core ‘launch’ GUI (<https://nf-co.re/taxprofiler/launch>), making the pipeline acces-  
271 sible for both computationally experienced as well as less experienced researchers. In  
272 addition to the general usage and parameter documentation of the pipeline ([https://nf-](https://nf-co.re/taxprofiler)  
273 [co.re/taxprofiler](https://nf-co.re/taxprofiler)). The GUI offers immediate assistance and guidance to users on what  
274 each parameter does, both in short- and long-form, with long-form parameter descrip-  
275 tions additionally describing which tool-specific parameters are being modified for  
276 each pipeline parameter (Figure 2). The GUI also includes controlled user input by  
277 providing strict drop-down lists and input validation prior execution of the pipeline  
278 to reduce the risk of typos and other mistakes (in contrast to the command-line inter-  
279 face (CLI) that only includes validation at pipeline run-time).

280 An example nf-core command line execution of the pipeline can be seen in Code  
281 Block 1, where two input files are supplied: one file specifying paths of FASTQ files  
282 of metagenomic samples and necessary metadata for preprocessing (such as sample  
283 ID and sequencing platform), and the second file specifying paths to the user-defined  
284 databases with per-database classification parameters. Various parameters are avail-  
285 able to select different preprocessing steps, and provide additional configuration such  
286 as tool selection and value options. Note that even if a user supplies a given database  
287 in the database input sheet, the corresponding profiling tool must still be activated  
288 with the corresponding pipeline parameter (e.g. `--run_kraken2`). Per-classifier flags  
289 are also available for the optional saving of additional non-profile output files. Alter-  
290 natively to command line flags, parameters can be specified via pre-configured YAML

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

291 format files, with which (provided no hardcoded paths are included) can be re-used  
292 across pipeline runs.

---

**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
```

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

### 302 9.1.2 Preprocessing

303 Preprocessing steps in nf-core/taxprofiler are aimed at removing laboratory and se-  
304 quencing artefacts that may influence taxonomic profiling, either for computing re-  
305 source consumption or and/or false-positive or false-negative classification reasons.  
306 First sequencing quality control with FastQC (Andrews 2010) or Falco (Sena Brandine  
307 and Smith 2021) is carried out. Falco was included for reduced memory requirements,  
308 in particular for long read sequencing data. Artificial library adapter sequences added  
309 during sequencing reduce sequencing matching accuracy by reducing sequence speci-  
310 ficity, and in some cases, may result in false-positive hits due to adapter sequence con-  
311 tamination in reference genomes (Schäffer et al. 2018; F. P. Breitwieser, Baker, and

Salzberg 2018)<sup>1</sup>. Additionally, paired-end merging may provide longer sequences that will allow for more specific classification when paired-end alignment is not 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 of 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 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) 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

---

<sup>1</sup>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=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)

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), 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.

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 resources but this is customisable using 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.

#### 392 9.1.4 Post-profiling

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

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

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

#### 409 9.1.5 Output

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

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

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

## 423 9.2 Comparison with other solutions

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



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 least one preprocessing step and were not specifically targeted at read classification of specific domains of taxa (e.g. viruses or bacteriophages 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 evaluated the pipelines based on their publications and documentation for typical metagenomic profiling workflow steps, and a range of 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 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 (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 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), again maximising user choice and portability when it comes to running the pipeline.

476 In terms of scalability, the aforementioned integration with schedulers and cloud com-  
477 puting support implicitly maximises efficiency and parallelisation of pipeline runs,  
478 providing good scalability for varying numbers of input files and steps in the pipeline.  
479 Again, the three workflow manager based pipelines provide scalability, whereas there  
480 is no mention neither Unipro UGENE nor TAMA in reference to parallel task execu-  
481 tion. Furthermore, all pipelines except TAMA, allowed per-process customisation of  
482 computational resources, something critical for maximising efficient scalability to en-  
483 sure only the necessary resources for a given step of a pipeline are requested.

484 In terms of reproducibility, all five pipelines are good at ensuring reproducibility in  
485 terms of pipeline and software versioning (allowing re-execution of pipeline runs us-  
486 ing the same software), with only tama not having stable versioned releases. However,  
487 installing software manually across different infrastructures can result in variability  
488 in the execution of each software <sup>2</sup> (Di Tommaso et al. 2017). The current most popu-  
489 lar solution to the problem of inconsistent software environments is to use container  
490 engines such as Docker or Apptainer to run container images which are isolated, deter-  
491 ministic computing environments which can be executed by any system providing a  
492 container runtime. Only Unipro UGENE does not document the use of a container sys-  
493 tem, with nf-core/taxprofiler offering the biggest choice for users courtesy of Nextflow  
494 (6 different engine systems at the time of writing).

495 Finally, we compared metagenomics related functionality between the pipelines. All  
496 pipelines support short-read FASTQ input, but only nf-core/taxprofiler explicitly re-  
497 ports long-read support, while the documentation in Unipro UGENE states that assem-  
498 bled contigs are possible input to some of the profilers. All pipelines support read pre-  
499 processing (adapter clipping, and merging). In terms of tools used for preprocessing,  
500 Trimmomatic (Bolger, Lohse, and Usadel 2014) is popular across the other pipelines  
501 but is not supported in nf-core/taxprofiler. Only sunbeam and nf-core/taxprofiler sup-  
502 port complexity filtering to remove low sequence diversity reads. In fact within sun-  
503 beam, the authors developed their own dedicated, performant complexity filtering  
504 tool Komplexity (Clarke et al. 2019). Most pipelines support some form of host re-  
505 moval (only TAMA did not support this), and it is likely possible with Unipro UGENE  
506 (although not directly described). In all cases, host removal consists of mapping pro-  
507 cessed reads with an aligner and using the off-target reads for downstream profiling  
508 (as implemented in nf-core/taxprofiler), however StaG-mwc has an additional separate  
509 metagenomic host removal step with Kraken2. nf-core/taxprofiler supports by far the  
510 largest number of taxonomic classifiers and profilers at 11 as of v1.1.0 - providing the  
511 greatest choice to users - with StaG-mwc offering 7, and the remaining pipelines only  
512 3. Only nf-core/taxprofiler and partly StaG-mwc explicitly support running each pro-  
513 filer with multiple databases. nf-core/taxprofiler is the only pipeline that supports  
514 running an arbitrary number of different metagenomic profiler databases each with  
515 their own settings - making it useful for tool parameter comparison, testing differ-  
516 ent databases, or reducing the size of each database (e.g. per domain) to make it  
517 more flexibility for running on smaller computational infrastructure. StaG-mwc al-

<sup>2</sup>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)

518 lows multiple references for their short-read alignment steps rather than the metage-  
 519 nomic profilers. For output, nf-core/taxprofiler, StaG-mwc, and sunbeam (via an ex-  
 520 tension) support a singular run report for summarising all preprocessing step. Only  
 521 nf-core/taxprofiler and TAMA produce standardised output for all taxonomic profil-  
 522 ers, the former with the dedicated standalone tool TAXPASTA (Beber et al. 2023).  
 523 However Unipro UGENE additionally offers a ‘consensus’ profile using WEVOTE  
 524 (Metwally et al. 2016).

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

535 The functionality offered by other pipelines not currently supported by nf-  
 536 core/taxprofiler include sequencing saturation estimation (StaG-mwc), taxonomy-  
 537 free composition comparison (StaG-mwc), functional profiling (StaG-mwc), *de novo*  
 538 assembly (sunbeam), and reference mapping (StaG-mwc, sunbeam). We do not  
 539 plan to support *de novo* assembly or functional profiling in nf-core/taxprofiler as  
 540 we feel these are already better served by other existing dedicated pipelines within  
 541 the nf-core ecosystem [nf-core/mag for *de novo* assembly, Krakau et al. (2022),  
 542 and nf-core/funcscan for functional profiling <https://nf-co.re/funcscan>], as well as  
 543 elsewhere [e.g. MetaWrap Uritskiy, DiRuggiero, and Taylor (2018);].

## 544 References

- 545 Andrews, Simon. 2010. “FastQC: A Quality Control Tool for High Throughput Se-  
 546 quence Data.” <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>.
- 547 Beber, Moritz E, Maxime Borry, Sofia Stamouli, and James A Fellows Yates. 2023.  
 548 “TAXPASTA: TAXonomic Profile Aggregation and STandardisation.” *Journal of*  
 549 *Open Source Software* 8 (87): 5627. <https://doi.org/10.21105/joss.05627>.
- 550 Bengtsson-Palme, Johan, Martin Hartmann, Karl Martin Eriksson, Chandan Pal, Kaisa  
 551 Thorell, Dan Göran Joakim Larsson, and Rolf Henrik Nilsson. 2015. “METAXA2:  
 552 Improved Identification and Taxonomic Classification of Small and Large Subunit  
 553 rRNA in Metagenomic Data.” *Molecular Ecology Resources* 15 (6): 1403–14. <https://doi.org/10.1111/1755-0998.12399>.
- 554 Blanco-Míguez, Aitor, Francesco Beghini, Fabio Cumbo, Lauren J McIver,  
 555 Kelsey N Thompson, Moreno Zolfo, Paolo Manghi, et al. 2023. “Extend-  
 556 ing and Improving Metagenomic Taxonomic Profiling with Uncharacter-  
 557 ized Species Using MetaPhlAn 4.” *Nature Biotechnology*, February, 1–12.  
 558 <https://doi.org/10.1038/s41587-023-01688-w>.

560 Bolger, Anthony M, Marc Lohse, and Bjoern Usadel. 2014. "Trimmomatic: A Flexible  
561 Trimmer for Illumina Sequence Data." *Bioinformatics (Oxford, England)* 30 (15):  
562 2114–20. <https://doi.org/10.1093/bioinformatics/btu170>.

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

565

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

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

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

575

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

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

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

583

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

586

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

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

593 Danecek, Petr, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Mar-  
594 tin O Pollard, Andrew Whitwham, et al. 2021. "Twelve Years of SAMtools and  
595 BCftools." *GigaScience* 10 (2). <https://doi.org/10.1093/gigascience/giab008>.

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

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

604 Dilthey, Alexander T, Chirag Jain, Sergey Koren, and Adam M Phillippy. 2019. "Strain-  
605 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 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 Afshinnkoo, 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. "Ref-Seq 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>.

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

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 Proserpi. 2019. “Flexible Design of Multiple Metagenomics Classification Pipelines with UGENE.” *Bioinformatics (Oxford, England)* 35 (11): 1963–65. <https://doi.org/10.1093/bioinformatics/bty901>.

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

744 Interpretation-a Benchmark of Metagenomics Software.” *Nature Methods* 14 (11):  
745 1063–71. <https://doi.org/10.1038/nmeth.4458>.

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

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

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

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

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

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

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

767 Uritskiy, Gherman V, Jocelyne DiRuggiero, and James Taylor. 2018. “MetaWRAP-a  
768 Flexible Pipeline for Genome-Resolved Metagenomic Data Analysis.” *Microbiome*  
769 6 (1): 158. <https://doi.org/10.1186/s40168-018-0541-1>.

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

774 Veiga Leprevost, Felipe da, Björn A Gruning, Saulo Alves Aflitos, Hannes L  
775 Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, et al. 2017. “Bio-  
776 Containers: An Open-Source and Community-Driven Framework for Soft-  
777 ware Standardization.” *Bioinformatics (Oxford, England)* 33 (16): 2580–82.  
778 <https://doi.org/10.1093/bioinformatics/btx192>.

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

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



- 790 Wratten, Laura, Andreas Wilm, and Jonathan Göke. 2021. “Reproducible, Scalable,  
791 and Shareable Analysis Pipelines with Bioinformatics Workflow Managers.” *Nature Methods* 18 (10): 1161–68. <https://doi.org/10.1038/s41592-021-01254-9>.
- 792 Wright, Robyn J, André M Comeau, and Morgan G I Langille. 2023. “From Defaults to  
793 Databases: Parameter and Database Choice Dramatically Impact the Performance  
794 of Metagenomic Taxonomic Classification Tools.” *Microbial Genomics* 9 (3). <https://doi.org/10.1099/mgen.0.000949>.
- 795 Ye, Simon H, Katherine J Siddle, Daniel J Park, and Pardis C Sabeti. 2019. “Bench-  
796 marking Metagenomics Tools for Taxonomic Classification.” *Cell* 178 (4): 779–94.  
797 <https://doi.org/10.1016/j.cell.2019.07.010>.
- 798 Yilmaz, Pelin, Laura Wegener Parfrey, Pablo Yarza, Jan Gerken, Elmar Pruesse, Chris-  
799 tian Quast, Timmy Schweer, Jörg Peplies, Wolfgang Ludwig, and Frank Oliver  
800 Glöckner. 2014. “The SILVA and ‘All-Species Living Tree Project (LTP)’ Taxo-  
801 nomic Frameworks.” *Nucleic Acids Research* 42 (Database issue): D643–8. <https://doi.org/10.1093/nar/gkt1209>.
- 802  
803  
804