

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

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

⁹Department 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, as well
37 as best-practise software development and community support to ensure longevity
38 and adaptability of the pipeline, keeping up with the field of metagenomics.

39 2 Introduction

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

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

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

69 One solution to addressing the problem of choice among the range of different tools
70 is to run all of them in parallel, and cross compare the results. This can be useful both
71 for benchmarking studies (e.g. Sczyrba et al. 2017; Meyer et al. 2022), but also to
72 build consensus profiles whereby confidence of a particular taxonomic identification
73 can be increased when it is detected by multiple tools (McIntyre et al. 2017; Ye et al.
74 2019).

75 A second challenge in taxonomic classification is a question of databases. As with
76 tools, there is no one set ‘gold standard’ database. Different questions and contexts

77 require different databases, such as when a researcher wants to search for both bacte-
78 rial and viral species in samples, and as an extension of this, taxonomic classifiers
79 may need different settings for each database. Furthermore, as genomic sequenc-
80 ing becomes cheaper and more efficient, the number of publicly available reference
81 genomes is rapidly increasing (Nasko et al. 2018). Consequently, the size of reference
82 databases of taxonomic classifiers is also growing, often outpacing the computational
83 capacity available to researchers. In fact, while this was one of the main motivations
84 behind classifiers such as Kraken2 (Wood, Lu, and Langmead 2019), these algorithmic
85 techniques are already becoming insufficient (Wright, Comeau, and Langille 2023).

86 Finally, with the decrease of costs, the possibility for larger and larger metagenomic
87 sequencing datasets increases, leading to increasing sample sizes in studies, as ex-
88 emplified by the doubling of the number of metagenomes on the European Bioin-
89 formatic Institute’s MGnify database in two years (Mitchell et al. 2019). Altogether
90 this highlights the need for methods to efficiently profile many samples using many
91 tools. Manually setting up bioinformatic jobs for classification tasks for each database
92 and settings against different tools on traditional academic computing infrastructure
93 (e.g. high performance computing clusters or ‘HPC’ clusters) can be very tedious. Ad-
94 ditionally, particularly for very large sample sets, there is increasing use of cloud plat-
95 forms that have greater scalability than traditional HPCs. Being able to reliably and
96 reproducibly execute taxonomic classification tasks across infrastructure with mini-
97 mal intervention would therefore be a boon for the metagenomics field.

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

106 3 Description

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

111 In brief, nf-core/taxprofiler can accept short- (e.g. Illumina) and/or long-read
112 (e.g. Nanopore) FASTQ or FASTA files. This is provided in the form of a TSV file that
113 includes basic sample and sequencing library information. The pipeline can then
114 be executed either via a standard Nextflow command-line-interface (CLI) execution
115 or graphical-user-interface (GUI) through either the open-source and free nf-core
116 launch page (<https://nf-core/launch>) or the commercial (with free-tier) Nextflow
117 tower (<https://tower.nf>) solution. Examples of the command-line execution and

118 nf-core launch GUI can be seen in the [Supplementary Information](#).

119 It can perform a range of appropriate read preprocessing steps, such adapter removal,
120 read merging, low-sequence complexity filtering, host- or contamination removal,
121 and/or per-sample run merging. All of these steps are optional, and are aimed at
122 removing possible sequencing artefacts that may result in false positive taxonomic
123 classification hits or improve classification efficiency. Most of these steps also pro-
124 vide options of different tools to allow user preference.

125 After pre-processing, nf-core/taxprofiler can perform simultaneous profiling of pre-
126 processing reads as many as 11 different taxonomic classifiers or profilers (Table 1),
127 and on top of this, simultaneous for each of these an arbitrary number of databases
128 supplied by the user. Databases are also supplied via an input TSV file, that also allows
129 per-database custom classification parameters - meaning a given database can be sup-
130 plied multiple times each with different parameters. All classifiers with secondary
131 steps to generate or convert to additional output file formats are also included.

132 Post-processing of taxonomic profiles include aggregation (i.e., merging of multiple
133 profiles into a single multi-sample table), standardisation of profiles for easier com-
134 parison between profilers with the tool TAXPASTA (developed originally for the nf-
135 core/taxprofiler project, Beber et al. 2023), and visualisation of profiles with Krona
136 (Ondov, Bergman, and Phillippy 2011) for supported classifiers.

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

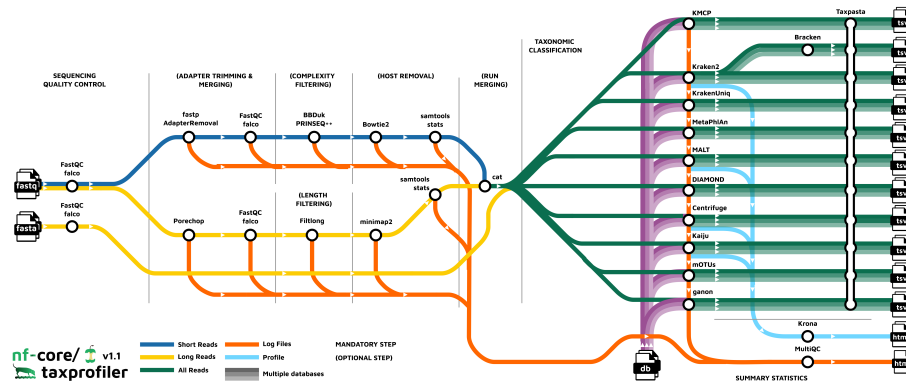


Figure 1: Visual overview of the nf-core/taxprofiler workflow. nf-core/taxprofiler can take in FASTQ (short or long reads) or FASTA files (long reads), that will optionally go through sequencing quality control (e.g. with FastQC), read preprocessing (e.g. removal of adapters), complexity filtering, host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

4 Discussion

A range of pipelines already exists for taxonomic profiling, however, each have their own particular purpose and capabilities. We compared the functionality of nf-core/taxprofiler against four other recently published or released pipelines, selected based on their similarity of purpose to nf-core/taxprofiler. The selection criteria and a more detailed comparison between the five pipelines can be seen in the [Supplementary Information](#), however overall, while there was a general similarity across all pipelines, nf-core/taxprofiler showed the greatest accessibility and user choice, through the use of an established workflow manager (Nextflow supporting 7 software environment/container systems), supporting both CLI and GUI execution, and the number of supported classifiers. Furthermore, it is unique in that is the only

150 pipeline to support supplying multiple database for all of the tools in a single pipeline
 151 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/gkimlab/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/btz275	10.1093/bioinformatics/btz275	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

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
A	GUI execution interface	No	No	Yes	No	Yes
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

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

152 An important advantage of nf-core/taxprofiler is that it is being developed within the
 153 nf-core community (<https://nf-co.re>), that provides strong long-term support for the
 154 continued community-based development and maintenance of its pipelines. In this
 155 framework, we will continue to add additional preprocessing, metagenomic classifi-
 156 cation, and profiling tools as they become established and as requested by the metage-
 157 nomics community, for example, we feel that the inclusion of steps such as sequenc-
 158 ing saturation estimation as already being performed by a similar pipeline StaG-mwc
 159 (<https://github.com/ctmrbio/stag-mwc>) would be beneficial to the nf-core/taxprofiler
 160 workflow (possibly with dedicated tools such as Nonpareil, Rodriguez-R et al. 2018),
 161 and/or more performant complexity filtering tools such as Komplexity as offered by
 162 the sunbeam metagenomics pipeline (Clarke et al. 2019). This also applies to extend
 163 support to other sequencing platforms; nf-core/taxprofiler already supports Nanopore
 164 long-read data, however the use of long-read PacBio data for metagenomic data is
 165 growing in interest (Portik, Brown, and Pierce-Ward 2022). We are therefore consid-
 166 ering adding dedicated preprocessing steps for this type of sequencing data.

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

177 Finally, once an overall taxonomic profile is generated, researchers often wish to val-
 178 idate hits through more sensitive and accurate methods such as with read-mapping
 179 alignment. While read alignment is supported by other pipelines such as StaG-mwc,
 180 this happens in-parallel to the taxonomic profiling and requires prior expectation of
 181 which reference genomes to map against. Instead, nf-core/taxprofiler could be easily
 182 extended to have a validation step similar to that of the ancient DNA metagenomic
 183 pipeline aMeta (Pochon et al. 2022) where, utilising Nextflow’s execution parallelism,

184 the input sequences could be aligned back to the reference genomes of only those
185 species with hits from the taxonomic classification with dedicated accurate short- or
186 long-read aligners. In addition to the more precise classification, post-classification
187 read-alignment could also be particularly useful for researchers in palaeogenomics
188 who wish to use tools other than KrakenUniq for initial classification (as in aMeta),
189 where alignment information can be used to authenticate ancient DNA within their
190 samples but also in clinical metagenomics to identify potential pathogens at much
191 finer resolution (e.g. down to strain level).

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

202 5 Conclusion

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

211 6 Data Availability

212 All data used in this publication

213 7 Code Availability

214 nf-core/taxprofiler source code is available on GitHub at <https://github.com/nf-core/taxprofiler>,
215 and each release is archived on Zenodo (latest version DOI: [10.5281/zenodo.7728364](https://doi.org/10.5281/zenodo.7728364))
216

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

219 **8 Supplementary Data**

220 **9 Acknowledgments**

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

225 **10 Funding**

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

232 **11 Supplementary Information**

233 **11.1 Implementation**

234 **11.1.1 Input and Execution**

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

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

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

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
```

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

11.1.2 Preprocessing

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

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-

276 that will allow for more specific classification when paired-end alignment is not sup-
277 ported by a given classifier. For these tasks nf-core/taxprofiler can apply either fastp
278 (Chen et al. 2018) or AdapterRemoval2 (Schubert, Lindgreen, and Orlando 2016) for
279 short reads, and currently Porechop (Wick et al. 2017) for Oxford Nanopore long-read
280 data. For both short and long reads, FastQC or Falco is run again to allow assessment
281 on the performance of the adapter removal and/or pair-merging step.

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

296 Removing host DNA is another common preprocessing step in metagenomic studies.
297 This can help speed up run-time, particularly in microbiome studies, where detection
298 of microbes are of interest. Furthermore, host-contamination of reference genomes in
299 public databases is common (Longo, O'Neill, and O'Neill 2011; Kryukov and Imanishi
300 2016; Florian P. Breitwieser et al. 2019) and therefore the removal of such sequences
301 can also decrease the risk of false positive taxonomic assignment. To remove multiple
302 hosts or other sequences, all reference genomes can be combined into a single FASTA
303 reference file. Short read host removal can be carried out with Bowtie2 (Langmead
304 and Salzberg 2012; Langmead et al. 2019) and minimap2 (Li 2018) for long reads, both
305 in combination with SAMtools (Li et al. 2009; Danecek et al. 2021), where reads are
306 aligned against the reference genome and the off-target (unaligned) reads are then
307 converted back to FASTQ format for classification.

308 Finally, nf-core/taxprofiler can optionally perform run merging where libraries have
309 been sequenced over multiple lanes to generate one profile per sample or library. The
310 final set of reads used for profiling can be optionally saved for downstream re-use.
311 Throughout all steps, relevant statistics and log files are generated and used both for
312 the final pipeline run report as well as saved into the results directory of the pipeline
313 run for further inspection where necessary.

your.html Sixing Huang: <https://web.archive.org/web/20220904205331/https://dgg32.medium.com/carp-in-the-soil-1168818d2191>
(Accessed 2023-08-25)

11.1.3 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.

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

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.

354 **11.1.4 Post-profiling**

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

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

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

371 **11.1.5 Output**

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

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

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

385 **11.2 Comparison with other solutions**

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

393 methods. Despite this, tools such as METAXA2 (Bengtsson-Palme et al. 2015) that
394 use shotgun sequencing reads to recover 16S sequences from metagenomic samples.

395 We searched Google Scholar for open-source pipelines published or released in the last
396 5 years (at the time of writing, since 2018) that were designed primarily for metage-
397 nomic classification screening, that supported at least 2 classifiers, had at least one
398 preprocessing step and were not specifically targeted at read classification of specific
399 domains of taxa (e.g. viruses or bacteriophages only). We also included an additional
400 pipeline at the recommendations of the authors of the pipeline due to the functional
401 overlap to nf-core/taxprofiler. We then evaluated the pipelines based on their publi-
402 cations and documentation for typical metagenomic profiling workflow steps, and a
403 range of criteria related to expectations of modern bioinformatic workflows that can
404 be summarised in the following four criteria: reproducibility, accessibility, scalabil-
405 ity, and portability (Wratten, Wilm, and Göke 2021). After searching, we selected the
406 following pipelines for comparison with nf-core/taxprofiler: sunbeam (v4, Clarke et
407 al. 2019), Unipro UGENE (v48, Rose et al. 2019), TAMA (githash: 3a22c8f, Sim et al.
408 2020), and StaG-mwc (0.7.0, Boulund et al. 2023).

409 In terms of accessibility, all pipelines have documentation describing the installation
410 steps, usage instructions, and output files. However, there are varying levels of de-
411 tail and comprehensiveness. In particular, StaG-mwc and nf-core/taxprofiler have
412 the most detailed descriptions of all possible output files for every supported mod-
413 ule, whereas Unipro UGENE and sunbeam have very minimal to possibly unfinished
414 output documentation. For execution options, most of the pipelines provide CLI ex-
415 ecution, except for Unipro UGENE which offers only GUI-based pipeline set-up (de-
416 spite a command-line execution of the GUI generated configuration). In particular, nf-
417 core/taxprofiler is the only pipeline providing both CLI and GUI interfaces for pipeline
418 run execution.

419 Criteria covering portability also overlap with accessibility, as it implies options for
420 and ease of different users running on different types of computing infrastructure,
421 whether that is on their own laptop, on an HPC cluster, or in the cloud. Unipro
422 UGENE is the only pipeline that supports execution on all three major operating sys-
423 tems (Linux, OSX, Windows), whereas StaG-mwc and nf-core/taxprofiler can be run
424 on unix operating systems, and sunbeam and TAMA are only being supported on
425 Linux. While all pipelines support ‘local’ machine execution (e.g. personal laptops or
426 desktops), a large portion of academic users execute computationally intensive bioin-
427 formatic tasks on HPC clusters. In these contexts, pipeline task submissions are nor-
428 mally managed by job schedulers, thus integration with schedulers is an important
429 criterion for running large multi-step and parallelised pipelines. The three pipelines
430 leveraging workflow managers (Snakemake (Mölder et al. 2021) and Nextflow) sup-
431 port integration with schedulers (StaG-mwc, sunbeam, and nf-core/taxprofiler) with
432 nf-core/taxprofiler supporting the most by far ([>10 scheduling systems](#)) as natively
433 offered by Nextflow. This allows the greatest possible choice for users in terms of
434 which HPC infrastructure they can execute their pipeline on. As an extension of this,
435 only nf-core/taxprofiler has explicit support for cloud computing (e.g. AWS, GCP, or
436 Microsoft Azure), again maximising user choice and portability when it comes to run-

437 ning the pipeline.

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

446 In terms of reproducibility, all five pipelines are good at ensuring reproducibility in
447 terms of pipeline and software versioning (allowing re-execution of pipeline runs us-
448 ing the same software), with only tama not having stable versioned releases. However,
449 installing software manually across different infrastructures can result in variability
450 in the execution of each software ² (Di Tommaso et al. 2017). The current most popu-
451 lar solution to the problem of inconsistent software environments is to use container
452 engines such as Docker or Apptainer to run container images which are isolated, deter-
453 ministic computing environments which can be executed by any system providing a
454 container runtime. Only Unipro UGENE does not document the use of a container sys-
455 tem, with nf-core/taxprofiler offering the biggest choice for users courtesy of Nextflow
456 (6 different engine systems at the time of writing).

457 Finally, we compared metagenomics related functionality between the pipelines. All
458 pipelines support short-read FASTQ input, but only nf-core/taxprofiler explicitly re-
459 ports long-read support, while the documentation in Unipro UGENE states that assem-
460 bled contigs are possible input to some of the profilers. All pipelines support read pre-
461 processing (adapter clipping, and merging). In terms of tools used for preprocessing,
462 Trimmomatic (Bolger, Lohse, and Usadel 2014) is popular across the other pipelines
463 but is not supported in nf-core/taxprofiler. Only sunbeam and nf-core/taxprofiler sup-
464 port complexity filtering to remove low sequence diversity reads. In fact within sun-
465 beam, the authors developed their own dedicated, performant complexity filtering
466 tool Komplexity (Clarke et al. 2019). Most pipelines support some form of host re-
467 moval (only TAMA did not support this), and it is likely possible with Unipro UGENE
468 through user customisation of the workflow. In all cases, host removal consists of
469 mapping processed reads with an aligner and using the off-target reads for down-
470 stream profiling (as implemented in nf-core/taxprofiler), however StaG-mwc has an
471 additional separate metagenomic host removal step with Kraken2. nf-core/taxprofiler
472 supports by far the largest number of taxonomic classifiers and profilers at 11 as of
473 v1.1.0 - providing the greatest choice to users - with StaG-mwc offering 7, and the
474 remaining pipelines only 3. Only nf-core/taxprofiler and partly StaG-mwc explicitly
475 support running each profiler with multiple databases. nf-core/taxprofiler is the only
476 pipeline that supports running an arbitrary number of different metagenomic profiler
477 databases each with their own settings - making it useful for tool parameter compari-

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)

son, testing different databases, or reducing the size of each database (e.g. per domain) to make it more flexibility for running on smaller computational infrastructure. StaG-mwc allows multiple references for their short-read alignment steps rather than the metagenomic profilers. For output, nf-core/taxprofiler, StaG-mwc, and sunbeam (via an extension) support a singular run report for summarising all preprocessing step. Only nf-core/taxprofiler and TAMA produce standardised output for all taxonomic profilers (via TAXPASTA). However Unipro UGENE additionally offers a ‘consensus’ profile 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, the benefits provided by Nextflow whereby it provides the greatest number of computational infrastructure types the pipeline can be executed on, and container systems can be used to ensure reproducibility, and the support of the nf-core community due to the centralised pool of ‘plug-and-play’ modules to make it easier to update the pipeline over time to add new tool.

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

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

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/10.1111/1365-3113.12403>.

520 [//doi.org/10.1111/1755-0998.12399](https://doi.org/10.1111/1755-0998.12399).

521 Blanco-Míguez, Aitor, Francesco Beghini, Fabio Cumbo, Lauren J McIver,
 522 Kelsey N Thompson, Moreno Zolfo, Paolo Manghi, et al. 2023. "Extend-
 523 ing and Improving Metagenomic Taxonomic Profiling with Uncharacter-
 524 ized Species Using MetaPhlAn 4." *Nature Biotechnology*, February, 1–12.
 525 <https://doi.org/10.1038/s41587-023-01688-w>.

526 Bolger, Anthony M, Marc Lohse, and Bjoern Usadel. 2014. "Trimmomatic: A Flexible
 527 Trimmer for Illumina Sequence Data." *Bioinformatics (Oxford, England)* 30 (15):
 528 2114–20. <https://doi.org/10.1093/bioinformatics/btu170>.

529 Boulund, Fredrik, Aron Arzoomand, Justine Debelius, chrbs, and Lisa Olsson. 2023.
 530 "Ctmbio/Stag-Mwc: StaG v0.7.0." Zenodo. [https://doi.org/10.5281/ZENODO.](https://doi.org/10.5281/ZENODO.8032462)
 531 [8032462](https://doi.org/10.5281/ZENODO.8032462).

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

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

538 Breitwieser, Florian P, Mihaela Perte, Aleksey Zimin, and Steven L Salzberg. 2019.
 539 "Human Contamination in Bacterial Genomes Has Created Thousands of Spurious
 540 Proteins." *Genome Research* 29 (May): 954–60. [https://doi.org/10.1101/gr.245373.](https://doi.org/10.1101/gr.245373.118)
 541 [118](https://doi.org/10.1101/gr.245373.118).

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

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

546 Cantu, Vito Adrian, Jeffrey Sadural, and Robert Edwards. 2019. "PRINSEQ++, a Multi-
 547 Threaded Tool for Fast and Efficient Quality Control and Preprocessing of Se-
 548 quencing Datasets." e27553v1. PeerJ Preprints; PeerJ Inc. [https://doi.org/10.7287/](https://doi.org/10.7287/peerj.preprints.27553v1)
 549 [peerj.preprints.27553v1](https://doi.org/10.7287/peerj.preprints.27553v1).

550 Chen, Shifu, Yanqing Zhou, Yaru Chen, and Jia Gu. 2018. "Fastp: An Ultra-Fast All-
 551 in-One FASTQ Preprocessor." *Bioinformatics* 34 (17): i884–90. [https://doi.org/10.](https://doi.org/10.1093/bioinformatics/bty560)
 552 [1093/bioinformatics/bty560](https://doi.org/10.1093/bioinformatics/bty560).

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

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

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

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

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

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

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

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

581 Govender, Kumeren N, and David W Eyre. 2022. "Benchmarking Taxonomic Classi-
582 fiers with Illumina and Nanopore Sequence Data for Clinical Metagenomic Diag-
583 nostic Applications." *Microbial Genomics* 8 (10): 000886. <https://doi.org/10.1099/mgen.0.000886>.

584
585
586 Grüning, Björn, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe,
587 Christopher H Tomkins-Tinch, Renan Valieris, Johannes Köster, and Bio-
588 conda Team. 2018. "Bioconda: Sustainable and Comprehensive Soft-
589 ware Distribution for the Life Sciences." *Nature Methods* 15 (7): 475–76.
590 <https://doi.org/10.1038/s41592-018-0046-7>.

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

594
595 Kim, Daehwan, Li Song, Florian P Breitwieser, and Steven L Salzberg. 2016. "Cen-
596 trifuge: Rapid and Sensitive Classification of Metagenomic Sequences." *Genome*
597 *Research* 26 (12): 1721–29. <https://doi.org/10.1101/gr.210641.116>.

598 Krakau, Sabrina, Daniel Straub, Hadrien Gourel, Gisela Gabernet, and Sven Nahnsen.
599 2022. "Nf-Core/Mag: A Best-Practice Pipeline for Metagenome Hybrid Assembly
600 and Binning." *NAR Genomics and Bioinformatics* 4 (1). <https://doi.org/10.1093/nargab/lqac007>.

601
602 Kryukov, Kirill, and Tadashi Imanishi. 2016. "Human Contamination in Public
603 Genome Assemblies." *PloS One* 11 (9): e0162424. <https://doi.org/10.1371/journal.pone.0162424>.

604
605 Langmead, Ben, and Steven L Salzberg. 2012. "Fast Gapped-Read Alignment with
606 Bowtie 2." *Nature Methods* 9 (4): 357–59. <https://doi.org/10.1038/nmeth.1923>.

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

610 Li, Heng. 2018. "Minimap2: Pairwise Alignment for Nucleotide Sequences." *Bioinform-*
611 *atics* 34 (18): 3094–3100. <https://doi.org/10.1093/bioinformatics/bty191>.

Li, Heng, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. 2009. "The Sequence Alignment/Map Format and SAMtools." *Bioinformatics* 25 (16): 2078–79. <https://doi.org/10.1093/bioinformatics/btp352>.

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

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

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

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

McIntyre, Alexa B R, Rachid Ounit, Ebrahim 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>.

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>.

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

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

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

Schloss, Patrick D, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hart-

mann, Emily B Hollister, Ryan A Lesniewski, et al. 2009. "Introducing Mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities." *Applied and Environmental Microbiology* 75 (23): 7537–41. <https://doi.org/10.1128/AEM.01541-09>.

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

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

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

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

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

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

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

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

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

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

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

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

0446-6.

- Veiga Leprevost, Felipe da, Björn A Grüning, Saulo Alves Aflitos, Hannes L Röst, Julian Uszkoreit, Harald Barsnes, Marc Vaudel, et al. 2017. "BioContainers: An Open-Source and Community-Driven Framework for Software Standardization." *Bioinformatics (Oxford, England)* 33 (16): 2580–82. <https://doi.org/10.1093/bioinformatics/btx192>.
- Wick, Ryan R, Louise M Judd, Claire L Gorrie, and Kathryn E Holt. 2017. "Completing Bacterial Genome Assemblies with Multiplex MinION Sequencing." *Microbial Genomics* 3 (10): e000132. <https://doi.org/10.1099/mgen.0.000132>.
- Wood, Derrick E, Jennifer Lu, and Ben Langmead. 2019. "Improved Metagenomic Analysis with Kraken 2." *Genome Biology* 20 (1): 257. <https://doi.org/10.1186/s13059-019-1891-0>.
- Wratten, Laura, Andreas Wilm, and Jonathan Göke. 2021. "Reproducible, Scalable, and Shareable Analysis Pipelines with Bioinformatics Workflow Managers." *Nature Methods* 18 (10): 1161–68. <https://doi.org/10.1038/s41592-021-01254-9>.
- Wright, Robyn J, 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>.