

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 small to extremely large projects on a wide range of computing infrastructure, as well as

36 best-practise software development and community support to ensure longevity and
37 adaptability of the pipeline, keeping up with the field of metagenomics.

38 2 Introduction

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

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

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

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

73 A second challenge in taxonomic classification is a question of databases. As with tools,
74 there is no one set ‘gold standard’ database. Different questions and contexts require
75 different databases, such as when a researcher wants to search for both bacterial and
76 viral species in samples, and as an extension of this, taxonomic classifiers may need

different settings for each database. Furthermore, as genomic sequencing becomes cheaper and more efficient, the number of publicly available reference genomes is rapidly increasing (Nasko et al. 2018). Consequently, the size of reference databases of taxonomic classifiers is also growing, often outpacing the computational capacity available to researchers. In fact, while this was one of the main motivations behind classifiers such as Kraken2 (Wood, Lu, and Langmead 2019), these algorithmic techniques are already becoming insufficient (Wright, Comeau, and Langille 2023).

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

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

3 Description

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

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

It can perform a range of appropriate read preprocessing steps, such as adapter removal, read merging, low-sequence complexity filtering, host- or contamination removal,

and/or per-sample run merging. All of these steps are optional, and are aimed at removing possible sequencing artefacts that may result in false positive taxonomic classification hits or improve classification efficiency. Most of these steps also provide options of different tools to allow user preference.

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

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

All relevant preprocessing statistics are displayed in an interactive and dynamic MultiQC report (Ewels et al. 2020).

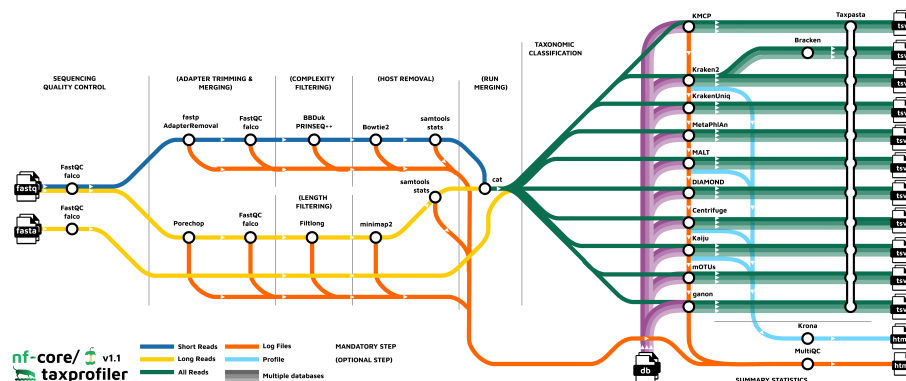


Figure 1: Visual overview of the nf-core/taxprofiler workflow. nf-core/taxprofiler can take in FASTQ (short or long reads) or FASTA files (long reads), that will optionally go through sequencing quality control (e.g. with FastQC), read preprocessing (e.g. removal of adapters), complexity filtering, host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

147 to support supplying multiple database for all of the tools in a single pipeline run.

Table 2: Comparison of functionality with four recent taxonomic pipelines with similar functionality. A more detailed textual comaprison can be found in the [Supplementary Information](#).

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
Information	Source code URL	https://github.com/cmrbio/s-tag-mwc	https://github.com/sunbeam-labs/sunbeam	https://github.com/ugene-neunipro/ugene	https://github.com/jkimlab/TAMA	https://github.com/nf-core/taxprofiler/
Information	Evaluated version	0.7.0	4	48	githash: 3a22c8f	1.1.0
Information	Last release date	2023-06-13	2023-08-08	2023-08-08	2022-03-02	2023-09-19
Information	Publication year	Unpublished	2019	2019	2020	This publication
Information	Publication DOI	Unpublished	10.1186/s40168-019-0658-x	10.1093/bioinformatics/bty250	10.1186/s12851-020-3533-7	This publication
Reproduction	Pipeline versioning	Yes	Yes	Yes	No	Yes
Reproduction	Software versioning	Yes	Yes	Yes	Yes	Yes
Reproduction	Multiple software environments or container engines supported	2	2	0	1	7
Accessibility	Installation documentation	Yes	Yes	Yes	Yes	Yes
Accessibility	Usage documentation	Yes	Yes	Yes	Yes	Yes
Accessibility	Output documentation	Yes	Yes	Yes	Yes	Yes
Accessibility	CLI execution interface	Yes	Yes	No	Yes	Yes
Accessibility	GUI execution interface	No	No	Yes	No	Yes
Accessibility	Integrability with scheduling systems	Yes	Yes	No	No	Yes

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
Portability	Compatibility with operating systems	2	1	3	1	2
Portability	Local machine integration	Yes	Yes	Yes	Yes	Yes
Portability	Highly scheduler integration	Yes	Yes	No	No	Yes
Portability	Cloud computing integration	Unsure	Unsure	No	No	Yes
Portability	Integration with multiple scheduling systems	Partial	Partial	No	No	Yes
Scalability	Per-process resource optimisation	Yes	Yes	Yes	No	Yes
Functionality	Short read support	Yes	Yes	Yes	Yes	Yes
Functionality	Long read support	No	No	Yes	No	Yes
Functionality	Read preprocessing	Yes	Yes	Yes	Yes	Yes
Functionality	Sequencing depth estimation	Yes	No	No	No	No
Functionality	Complexity filtering	No	Yes	No	No	Yes
Functionality	Host removal	Yes	Yes	Partial	No	Yes
Functionality	Any supported taxonomic classifiers/profilers	7	3	3	3	11
Functionality	Graphical run reports	Yes	No	No	No	Yes
Functionality	Standardised profiles	No	No	No	Yes	Yes
Functionality	Multiple database supported	Partial	No	No	No	Yes

Category	Criterion	StaG-mwc	sunbeam	Unipro UGENE	tama	nf-core/taxprofiler
Functional	Metagenomic assembly	No	Yes	No	No	No
Functional	Visualisation	No	No	No	No	Partial

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

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

172 Finally, once an overall taxonomic profile is generated, researchers often wish to
 173 validate hits through more sensitive and accurate methods such as with read-mapping
 174 alignment. While read alignment is supported by other pipelines such as StaG-mwc,
 175 this happens in-parallel to the taxonomic profiling and requires prior expectation of
 176 which reference genomes to map against. Instead, nf-core/taxprofiler could be easily
 177 extended to have a validation step similar to that of the ancient DNA metagenomic
 178 pipeline aMeta (Pochon et al. 2022) where, utilising Nextflow’s execution parallelism,
 179 the input sequences could be aligned back to the reference genomes of only those
 180 species with hits from the taxonomic classification with dedicated accurate short- or
 181 long-read aligners. In addition to the more precise classification, post-classification
 182 read-alignment could also be particularly useful for researchers in palaeogenomics
 183 who wish to use tools other than KrakenUniq for initial classification (as in aMeta),
 184 where alignment information can be used to authenticate ancient DNA within their

185 samples but also in clinical metagenomics to identify potential pathogens at much
186 finer resolution (e.g. down to strain level).

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

197 5 Conclusion

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

206 6 Data Availability

207 All data used in this publication

208 7 Code Availability

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

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

214 8 Supplementary Data

215 9 Acknowledgments

216 We thank Prof. Christina Warinner and the Microbiome Sciences group MPI-EVA for
217 original discussions that lead to the pipeline. We are also grateful for the nf-core

community for the original and ongoing support in the development in the pipeline, in particular for the contributions by Lauri Mesilaakso, Jianhong Ou, and Rafał Stepień.

10 Funding

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

11 Supplementary Information

11.1 Implementation

11.1.1 Input and Execution

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

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

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

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
```

(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 that will allow for more specific classification when paired-end alignment is not supported

¹For an ‘infamous’ case of adapter sequences in a published eukaryotic genome, see the following blog posts

Graham Etherington: <https://web.archive.org/web/20201219022000/http://grahametherington.blogspot.com/2014/09/why-you-should-qc-your-reads-and-your.html?m=1> why-you-should-qc-your-reads-and-your.html Sixing Huang: <https://web.archive.org/web/20220904205331/https://dgg32.medium.com/carp-in-the-soil-1168818d2191>

(Accessed 2023-08-25)

272 by a given classifier. For these tasks nf-core/taxprofiler can apply either fastp (Chen et
273 al. 2018) or AdapterRemoval2 (Schubert, Lindgreen, and Orlando 2016) for short reads,
274 and currently Porechop (Wick et al. 2017) for Oxford Nanopore long-read data. For
275 both short and long reads, FastQC or Falco is run again to allow assessment on the
276 performance of the adapter removal and/or pair-merging step.

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

291 Removing host DNA is another common preprocessing step in metagenomic studies.
292 This can help speed up run-time, particularly in microbiome studies, where detection
293 of microbes are of interest. Furthermore, host-contamination of reference genomes in
294 public databases is common (Longo, O'Neill, and O'Neill 2011; Kryukov and Imanishi
295 2016; Florian P. Breitwieser et al. 2019) and therefore the removal of such sequences
296 can also decrease the risk of false positive taxonomic assignment. To remove multiple
297 hosts or other sequences, all reference genomes can be combined into a single FASTA
298 reference file. Short read host removal can be carried out with Bowtie2 (Langmead
299 and Salzberg 2012; Langmead et al. 2019) and minimap2 (Li 2018) for long reads, both
300 in combination with SAMtools (Li et al. 2009; Danecek et al. 2021), where reads are
301 aligned against the reference genome and the off-target (unaligned) reads are then
302 converted back to FASTQ format for classification.

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

309 11.1.3 Profiling

310 There are many types of metagenomic profiling techniques, from profiling against
311 whole-genome references with alignment or k-mer based approaches, to methods
312 involving alignment to species-specific marker-gene families (Quince et al. 2017; Ye et
313 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 (Vågane et al. 2018), DIAMOND (Buchfink, Reuter, and Drost 2021), Kaiju (Menzel, Ng, and Krogh 2016), MetaPhlAn (Blanco-Míguez et al. 2023), mOTUs (Ruscheweyh et al. 2022), ganon (Piro et al. 2020), 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.

11.1.4 Post-profiling

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

One of the challenges that researchers face when comparing multiple taxonomic

356 classifiers or profilers is the heterogenous output formats that are produced, that often
357 require custom parsing and merging scripts for each tool to standardise. To facilitate
358 more user-friendly cross-comparisons between tools, nf-core/taxprofiler utilises the
359 TAXPASTA tool (Beber et al. 2023) to generate standardised profiles and generate
360 multi-sample tables.

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

366 11.1.5 Output

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

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

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

380 11.2 Comparison with other solutions

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

390 We searched Google Scholar for open-source pipelines published or released in the
391 last 5 years (at the time of writing, since 2018) that were designed primarily for
392 metagenomic classification screening, that supported at least 2 classifiers, had at least
393 one preprocessing step and were not specifically targeted at read classification of
394 specific domains of taxa (e.g. viruses or bacteriophages only). We also included an
395 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 supports execution on all three major operating systems (Linux, OSX, Windows), whereas StaG-mwc and nf-core/taxprofiler can be run on unix operating systems, and sunbeam and TAMA are only being supported on Linux. While all pipelines support 'local' machine execution (e.g. personal laptops or desktops), a large portion of academic users execute computationally intensive bioinformatic tasks on HPC clusters. In these contexts, pipeline task submissions are normally managed by job schedulers, thus integration with schedulers is an important 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.

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

440 In terms of reproducibility, all five pipelines are good at ensuring reproducibility in
441 terms of pipeline and software versioning (allowing re-execution of pipeline runs using
442 the same software), with only tama not having stable versioned releases. However,
443 installing software manually across different infrastructures can result in variability in
444 the execution of each software² (Di Tommaso et al. 2017). The current most popular
445 solution to the problem of inconsistent software environments is to use container
446 engines such as Docker or Apptainer to run container images which are isolated,
447 deterministic computing environments which can be executed by any system providing
448 a container runtime. Only Unipro UGENE does not document the use of a container
449 system, with nf-core/taxprofiler offering the biggest choice for users courtesy of
450 Nextflow (6 different engine systems at the time of writing).

451 Finally, we compared metagenomics related functionality between the pipelines. All
452 pipelines support short-read FASTQ input, but only nf-core/taxprofiler explicitly re-
453 ports long-read support, while the documentation in Unipro UGENE states that as-
454 sembled contigs are possible input to some of the profilers. All pipelines support read
455 preprocessing (adapter clipping, and merging). In terms of tools used for preprocessing,
456 Trimmomatic (Bolger, Lohse, and Usadel 2014) is popular across the other pipelines
457 but is not supported in nf-core/taxprofiler. Only sunbeam and nf-core/taxprofiler
458 support complexity filtering to remove low sequence diversity reads. In fact within
459 sunbeam, the authors developed their own dedicated, performant complexity filtering
460 tool Komplexity (Clarke et al. 2019). Most pipelines support some form of host removal
461 (only TAMA did not support this), and it is likely possible with Unipro UGENE through
462 user customisation of the workflow. In all cases, host removal consists of mapping
463 processed reads with an aligner and using the off-target reads for downstream profiling
464 (as implemented in nf-core/taxprofiler), however StaG-mwc has an additional separate
465 metagenomic host removal step with Kraken2. nf-core/taxprofiler supports by far the
466 largest number of taxonomic classifiers and profilers at 11 as of v1.1.0 - providing the
467 greatest choice to users - with StaG-mwc offering 7, and the remaining pipelines only 3.
468 Only nf-core/taxprofiler and partly StaG-mwc explicitly support running each profiler
469 with multiple databases. nf-core/taxprofiler is the only pipeline that supports running
470 an arbitrary number of different metagenomic profiler databases each with their own
471 settings - making it useful for tool parameter comparison, testing different databases,
472 or reducing the size of each database (e.g. per domain) to make it more flexibility for
473 running on smaller computational infrastructure. StaG-mwc allows multiple references
474 for their short-read alignment steps rather than the metagenomic profilers. For output,
475 nf-core/taxprofiler, StaG-mwc, and sunbeam (via an extension) support a singular run
476 report for summarising all preprocessing step. Only nf-core/taxprofiler and TAMA
477 produce standardised output for all taxonomic profilers (via TAXPASTA). However
478 Unipro UGENE additionally offers a 'consensus' profile using WEVOTE (Metwally et
479 al. 2016).

480 To summarise, many of the pipelines reviewed here offer similar functionality, with
481 particularly StaG-mwc having a strong overlap with nf-core/taxprofiler. Thus, users in

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)

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/10.1111/1755-0998.12399>.
- Blanco-Míguez, Aitor, Francesco Beghini, Fabio Cumbo, Lauren J McIver, Kelsey N Thompson, Moreno Zolfo, Paolo Manghi, et al. 2023. “Extending and Improving Metagenomic Taxonomic Profiling with Uncharacterized Species Using MetaPhlAn 4.” *Nature Biotechnology*, February, 1–12. <https://doi.org/10.1038/s41587-023-01688-w>.
- Bolger, Anthony M, Marc Lohse, and Bjoern Usadel. 2014. “Trimmomatic: A Flexible Trimmer for Illumina Sequence Data.” *Bioinformatics (Oxford, England)* 30 (15): 2114–20. <https://doi.org/10.1093/bioinformatics/btu170>.
- Boulund, Fredrik, Aron Arzoomand, Justine Debelius, chrbs, and Lisa Olsson. 2023. “Ctmbio/Stag-Mwc: StaG v0.7.0.” Zenodo. <https://doi.org/10.5281/ZENODO.80324>

- Breitwieser, F P, D N Baker, and S L Salzberg. 2018. "KrakenUniq: Confident and Fast Metagenomics Classification Using Unique k-Mer Counts." *Genome Biology* 19 (1): 198. <https://doi.org/10.1186/s13059-018-1568-0>.
- Breitwieser, Florian P, Jennifer Lu, and Steven L Salzberg. 2019. "A Review of Methods and Databases for Metagenomic Classification and Assembly." *Briefings in Bioinformatics* 20 (4): 1125–36. <https://doi.org/10.1093/bib/bbx120>.
- Breitwieser, Florian P, Mihaela Pertea, Aleksey Zimin, and Steven L Salzberg. 2019. "Human Contamination in Bacterial Genomes Has Created Thousands of Spurious Proteins." *Genome Research* 29 (May): 954–60. <https://doi.org/10.1101/gr.245373.118>.
- Buchfink, Benjamin, Klaus Reuter, and Hajk-Georg Drost. 2021. "Sensitive Protein Alignments at Tree-of-Life Scale Using DIAMOND." *Nature Methods* 18 (4): 366–68. <https://doi.org/10.1038/s41592-021-01101-x>.
- Bushnell, Brian. 2022. "BBMap." <https://sourceforge.net/projects/bbmap/>.
- Cantu, Vito Adrian, Jeffrey Sadural, and Robert Edwards. 2019. "PRINSEQ++, a Multi-Threaded Tool for Fast and Efficient Quality Control and Preprocessing of Sequencing Datasets." e27553v1. PeerJ Preprints; PeerJ Inc. <https://doi.org/10.7287/peerj.preprints.27553v1>.
- Chen, Shifu, Yanqing Zhou, Yaru Chen, and Jia Gu. 2018. "Fastp: An Ultra-Fast All-in-One FASTQ Preprocessor." *Bioinformatics* 34 (17): i884–90. <https://doi.org/10.1093/bioinformatics/bty560>.
- Chiu, Charles Y, and Steven A Miller. 2019. "Clinical Metagenomics." *Nature Reviews. Genetics* 20 (6): 341–55. <https://doi.org/10.1038/s41576-019-0113-7>.
- Clarke, Erik L, Louis J Taylor, Chunyu Zhao, Andrew Connell, Jung-Jin Lee, Bryton Fett, Frederic D Bushman, and Kyle Bittinger. 2019. "Sunbeam: An Extensible Pipeline for Analyzing Metagenomic Sequencing Experiments." *Microbiome* 7 (1): 46. <https://doi.org/10.1186/s40168-019-0658-x>.
- Danecek, Petr, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, et al. 2021. "Twelve Years of SAMtools and BCFTools." *GigaScience* 10 (2). <https://doi.org/10.1093/gigascience/giab008>.
- DeSantis, T Z, P Hugenholtz, N Larsen, M Rojas, E L Brodie, K Keller, T Huber, D Dalevi, P Hu, and G L Andersen. 2006. "Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB." *Applied and Environmental Microbiology* 72 (7): 5069–72. <https://doi.org/10.1128/AEM.03006-05>.
- Di Tommaso, Paolo, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. "Nextflow Enables Reproducible Computational Workflows." *Nature Biotechnology* 35 (4): 316–19. <https://doi.org/10.1038/nbt.3820>.
- Dilthey, Alexander T, Chirag Jain, Sergey Koren, and Adam M Phillippy. 2019. "Strain-Level Metagenomic Assignment and Compositional Estimation for Long Reads with MetaMaps." *Nature Communications* 10 (1): 3066. <https://doi.org/10.1038/s41467-019-10934-2>.
- Eloe-Fadros, Emiley A, Natalia N Ivanova, Tanja Woyke, and Nikos C Kyrpides. 2016. "Metagenomics Uncovers Gaps in Amplicon-Based Detection of Microbial Diversity." *Nature Microbiology* 1 (4): 15032. <https://doi.org/10.1038/nmicrobiol.2015.32>.
- Ewels, Philip A, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg,

570 Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen.
571 2020. "The Nf-Core Framework for Community-Curated Bioinformatics Pipelines."
572 *Nature Biotechnology* 38 (3): 276–78. <https://doi.org/10.1038/s41587-020-0439-x>.

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

576
577 Grüning, Björn, Ryan Dale, Andreas Sjödin, Brad A Chapman, Jillian Rowe, Christopher
578 H Tomkins-Tinch, Renan Valieris, Johannes Köster, and Bioconda Team. 2018.
579 "Bioconda: Sustainable and Comprehensive Software Distribution for the Life
580 Sciences." *Nature Methods* 15 (7): 475–76. <https://doi.org/10.1038/s41592-018-0046-7>.

581
582 Hillmann, Benjamin, Gabriel A Al-Ghalith, Robin R Shields-Cutler, Qiyun Zhu, Daryl
583 M Gohl, Kenneth B Beckman, Rob Knight, and Dan Knights. 2018. "Evaluating
584 the Information Content of Shallow Shotgun Metagenomics." *mSystems* 3 (6).
585 <https://doi.org/10.1128/mSystems.00069-18>.

586 Kim, Daehwan, Li Song, Florian P Breitwieser, and Steven L Salzberg. 2016. "Centrifuge:
587 Rapid and Sensitive Classification of Metagenomic Sequences." *Genome Research*
588 26 (12): 1721–29. <https://doi.org/10.1101/gr.210641.116>.

589 Krakau, Sabrina, Daniel Straub, Hadrien Gourel, Gisela Gabernet, and Sven Nahnsen.
590 2022. "Nf-Core/Mag: A Best-Practice Pipeline for Metagenome Hybrid Assembly
591 and Binning." *NAR Genomics and Bioinformatics* 4 (1). <https://doi.org/10.1093/nargab/lqac007>.

592
593 Kryukov, Kirill, and Tadashi Imanishi. 2016. "Human Contamination in Public Genome
594 Assemblies." *PloS One* 11 (9): e0162424. <https://doi.org/10.1371/journal.pone.0162424>.

595
596 Langmead, Ben, and Steven L Salzberg. 2012. "Fast Gapped-Read Alignment with
597 Bowtie 2." *Nature Methods* 9 (4): 357–59. <https://doi.org/10.1038/nmeth.1923>.

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

601 Li, Heng. 2018. "Minimap2: Pairwise Alignment for Nucleotide Sequences." *Bioinfor-*
602 *matics* 34 (18): 3094–3100. <https://doi.org/10.1093/bioinformatics/bty191>.

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

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

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

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

615

616 Marcelino, Vanessa R, Philip T L C Clausen, Jan P Buchmann, Michelle Wille, Jonathan
617 R Iredell, Wieland Meyer, Ole Lund, Tania C Sorrell, and Edward C Holmes.
618 2020. "CCMetagen: Comprehensive and Accurate Identification of Eukaryotes
619 and Prokaryotes in Metagenomic Data." *Genome Biology* 21 (1): 103. <https://doi.org/10.1186/s13059-020-02014-2>.
620
621 McIntyre, Alexa B R, Rachid Ounit, Ebrahim Afshinnkoo, Robert J Prill, Elizabeth Hé-
622 naff, Noah Alexander, Samuel S Minot, et al. 2017. "Comprehensive Benchmarking
623 and Ensemble Approaches for Metagenomic Classifiers." *Genome Biology* 18 (1):
624 182. <https://doi.org/10.1186/s13059-017-1299-7>.
625 Menzel, Peter, Kim Lee Ng, and Anders Krogh. 2016. "Fast and Sensitive Taxonomic
626 Classification for Metagenomics with Kaiju." *Nature Communications* 7 (April):
627 11257. <https://doi.org/10.1038/ncomms11257>.
628 Metwally, Ahmed A, Yang Dai, Patricia W Finn, and David L Perkins. 2016. "WEVOTE:
629 Weighted VOTing Taxonomic idEntification Method of Microbial Sequences." *PloS*
630 *One* 11 (9): e0163527. <https://doi.org/10.1371/journal.pone.0163527>.
631 Meyer, Fernando, Adrian Fritz, Zhi-Luo Deng, David Koslicki, Till Robin Lesker, Alexey
632 Gurevich, Gary Robertson, et al. 2022. "Critical Assessment of Metagenome
633 Interpretation: The Second Round of Challenges." *Nature Methods* 19 (4): 429–40.
634 <https://doi.org/10.1038/s41592-022-01431-4>.
635 Mitchell, Alex L, Alexandre Almeida, Martin Beracochea, Miguel Boland, Josephine
636 Burgin, Guy Cochrane, Michael R Crusoe, et al. 2019. "MGnify: The Microbiome
637 Analysis Resource in 2020." *Nucleic Acids Research*, November. <https://doi.org/10.1093/nar/gkz1035>.
638
639 Mölder, Felix, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H
640 Tomkins-Tinch, Vanessa Sochat, Jan Forster, et al. 2021. "Sustainable Data Analysis
641 with Snakemake." *F1000Research* 10 (January): 33. [https://doi.org/10.12688/f1000re
642 search.29032.2](https://doi.org/10.12688/f1000research.29032.2).
643 Morais, Diego A A, João V F Cavalcante, Shênia S Monteiro, Matheus A B Pasquali, and
644 Rodrigo J S Dalmolin. 2022. "MEDUSA: A Pipeline for Sensitive Taxonomic Classi-
645 fication and Flexible Functional Annotation of Metagenomic Shotgun Sequences."
646 *Frontiers in Genetics* 13 (March): 814437. <https://doi.org/10.3389/fgene.2022.814437>.
647 Nasko, Daniel J, Sergey Koren, Adam M Phillippy, and Todd J Treangen. 2018. "RefSeq
648 Database Growth Influences the Accuracy of k-Mer-Based Lowest Common An-
649 cestor Species Identification." *Genome Biology* 19 (1): 165. [https://doi.org/10.1186/
650 s13059-018-1554-6](https://doi.org/10.1186/s13059-018-1554-6).
651 Nayfach, Stephen, and Katherine S Pollard. 2016. "Toward Accurate and Quantitative
652 Comparative Metagenomics." *Cell* 166 (5): 1103–16. [https://doi.org/10.1016/j.cell.2
653 016.08.007](https://doi.org/10.1016/j.cell.2016.08.007).
654 Ondov, Brian D, Nicholas H Bergman, and Adam M Phillippy. 2011. "Interactive
655 Metagenomic Visualization in a Web Browser." *BMC Bioinformatics* 12 (1): 385.
656 <https://doi.org/10.1186/1471-2105-12-385>.
657 Piro, Vitor C, Temesgen H Dadi, Enrico Seiler, Knut Reinert, and Bernhard Y Renard.
658 2020. "Ganon: Precise Metagenomics Classification Against Large and up-to-Date
659 Sets of Reference Sequences." *Bioinformatics (Oxford, England)* 36 (Suppl_1): i12–20.
660 <https://doi.org/10.1093/bioinformatics/btaa458>.
661 Pochon, Zoé, Nora Bergfeldt, Emrah Kirdök, 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 Hartmann, Emily B Hollister, Ryan A Lesniewski, et al. 2009. “Introducing Mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities.” *Applied and Environmental Microbiology* 75 (23): 7537–41. <https://doi.org/10.1128/AEM.01541-09>.

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

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

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

- 708 Sena Brandine, Guilherme de, and Andrew D Smith. 2021. "Falco: High-Speed FastQC
709 Emulation for Quality Control of Sequencing Data." *F1000Research* 8 (1874): 1874.
710 <https://doi.org/10.12688/f1000research.21142.2>.
- 711 Sharpton, Thomas J. 2014. "An Introduction to the Analysis of Shotgun Metagenomic
712 Data." *Frontiers in Plant Science* 5 (June): 209. [https://doi.org/10.3389/fpls.2014.0020](https://doi.org/10.3389/fpls.2014.00209)
713 9.
- 714 Shen, Wei, Hongyan Xiang, Tianquan Huang, Hui Tang, Mingli Peng, Dachuan Cai,
715 Peng Hu, and Hong Ren. 2023. "KMCP: Accurate Metagenomic Profiling of Both
716 Prokaryotic and Viral Populations by Pseudo-Mapping." *Bioinformatics* 39 (1):
717 btac845. <https://doi.org/10.1093/bioinformatics/btac845>.
- 718 Sim, Mikang, Jongin Lee, Daehwan Lee, Daehong Kwon, and Jaebum Kim. 2020.
719 "TAMA: Improved Metagenomic Sequence Classification Through Meta-Analysis."
720 *BMC Bioinformatics* 21 (1): 185. <https://doi.org/10.1186/s12859-020-3533-7>.
- 721 Straub, Daniel, Nia Blackwell, Adrian Langarica-Fuentes, Alexander Peltzer, Sven
722 Nahnsen, and Sara Kleindienst. 2020. "Interpretations of Environmental Micro-
723 bial Community Studies Are Biased by the Selected 16S rRNA (Gene) Ampli-
724 con Sequencing Pipeline." *Frontiers in Microbiology* 11 (October): 550420. <https://doi.org/10.3389/fmicb.2020.550420>.
- 725 Sun, Zheng, Shi Huang, Meng Zhang, Qiyun Zhu, Niina Haiminen, Anna Paola Carrieri,
726 Yoshiki Vázquez-Baeza, et al. 2021. "Challenges in Benchmarking Metagenomic
727 Profilers." *Nature Methods* 18 (6): 618–26. [https://doi.org/10.1038/s41592-021-](https://doi.org/10.1038/s41592-021-01141-3)
728 01141-3.
- 729 Tamames, Javier, and Fernando Puente-Sánchez. 2018. "SqueezeMeta, a Highly
730 Portable, Fully Automatic Metagenomic Analysis Pipeline." *Frontiers in Micro-*
731 *biology* 9: 3349. <https://doi.org/10.3389/fmicb.2018.03349>.
- 732 Uritskiy, Gherman V, Jocelyne DiRuggiero, and James Taylor. 2018. "MetaWRAP-a
733 Flexible Pipeline for Genome-Resolved Metagenomic Data Analysis." *Microbiome* 6
734 (1): 158. <https://doi.org/10.1186/s40168-018-0541-1>.
- 735 Vågø, Åshild J, Alexander Herbig, Michael G Campana, Nelly M Robles García,
736 Christina Warinner, Susanna Sabin, Maria A Spyrou, et al. 2018. "Salmonella
737 Enterica Genomes from Victims of a Major Sixteenth-Century Epidemic in Mexico."
738 *Nature Ecology & Evolution* 2 (3): 520–28. [https://doi.org/10.1038/s41559-017-0446-](https://doi.org/10.1038/s41559-017-0446-6)
739 6.
- 740 Veiga Leprevost, Felipe da, Björn A Gruning, Saulo Alves Aflitos, Hannes L Röst,
741 Julian Uszkoreit, Harald Barsnes, Marc Vaudel, et al. 2017. "BioContainers: An
742 Open-Source and Community-Driven Framework for Software Standardization."
743 *Bioinformatics (Oxford, England)* 33 (16): 2580–82. [https://doi.org/10.1093/bioinform](https://doi.org/10.1093/bioinformatics/btx192)
744 [matics/btx192](https://doi.org/10.1093/bioinformatics/btx192).
- 745 Wick, Ryan R, Louise M Judd, Claire L Gorrie, and Kathryn E Holt. 2017. "Completing
746 Bacterial Genome Assemblies with Multiplex MinION Sequencing." *Microbial*
747 *Genomics* 3 (10): e000132. <https://doi.org/10.1099/mgen.0.000132>.
- 748 Wood, Derrick E, Jennifer Lu, and Ben Langmead. 2019. "Improved Metagenomic
749 Analysis with Kraken 2." *Genome Biology* 20 (1): 257. [https://doi.org/10.1186/s13059-](https://doi.org/10.1186/s13059-019-1891-0)
750 019-1891-0.
- 751 Wratten, Laura, Andreas Wilm, and Jonathan Göke. 2021. "Reproducible, Scalable, and
752 Shareable Analysis Pipelines with Bioinformatics Workflow Managers." *Nature*
753

754 *Methods* 18 (10): 1161–68. <https://doi.org/10.1038/s41592-021-01254-9>.

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

758

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

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

766