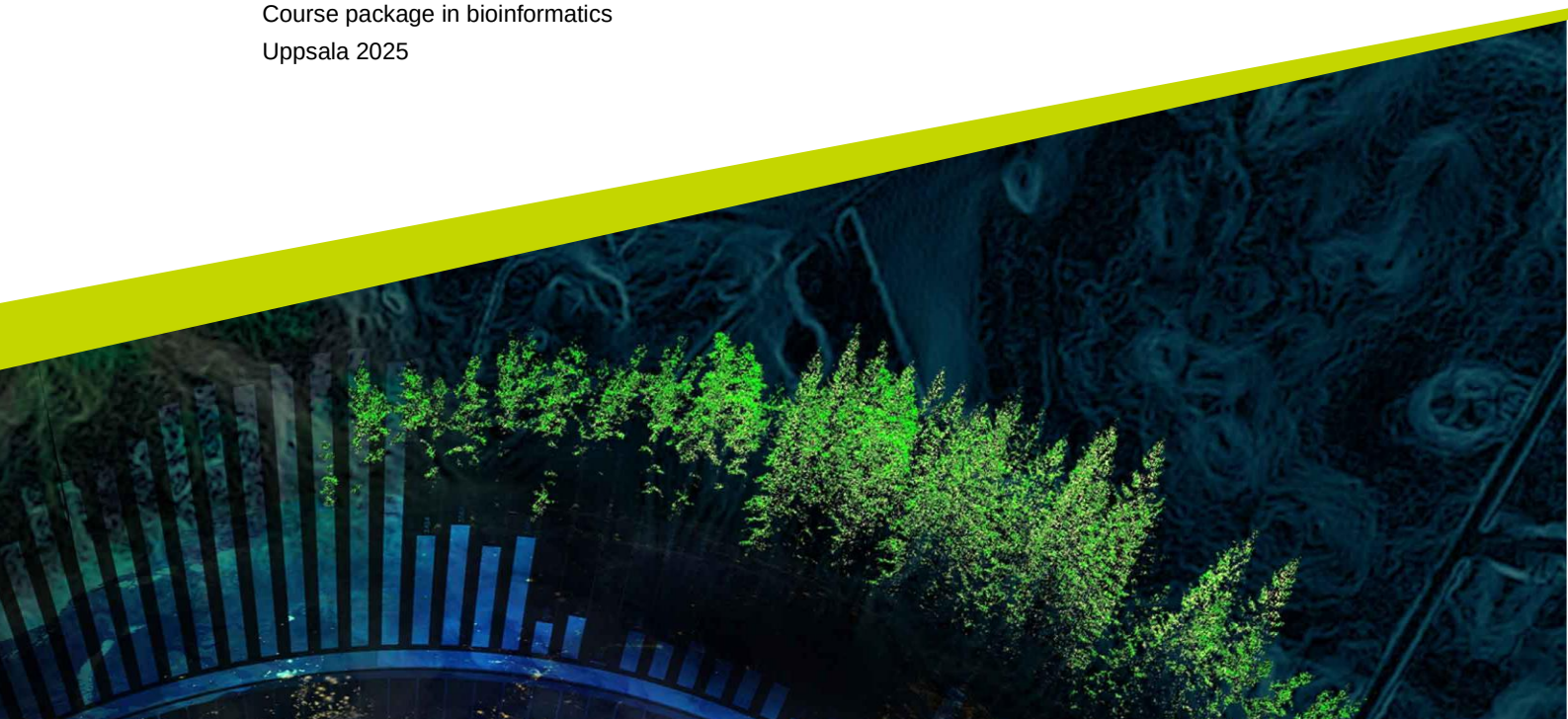
Benchmarking taxonomic classifiers for metagenomics using nanopore sequencing

Benjamin Walsh

Degree project/Independent project • 30 credits
Swedish University of Agricultural Sciences, SLU
Department of Animal Biosciences
Course package in bioinformatics
Uppsala 2025



Benchmarking taxonomic classifiers for metagenomics using nanopore sequencing.

Utvärdering av taxonomiska klassificerare för metagenomik med nanoporesekvensering.

Benjamin Walsh

Supervisor:	**Tobias Allander, Karolinska Institutet, Department of Microbiology, Tumor and Cell Biology**
Assistant supervisor:	Sofia Stamouli, Karolinska Institutet, Department of Microbiology, Tumor and Cell Biology
Assistant supervisor:	Lili Andersson-Li, Karolinska Institutet, Department of Microbiology, Tumor and Cell Biology
Examiner:	Stefan Bertilsson, Sveriges lantbruksuniversitet, Department of Aquatic Sciences and Assessment

Credits:	30
Level:	Second cycle, A2E
Course title:	Independent project in Bioinformatics
Course code:	EX1002
Programme/education:	Course package in Bioinformatics
Course coordinating dept:	Department of Animal Biosciences
Place of publication:	Uppsala
Year of publication:	2025
Keywords:	Bioinformatics, Metagenomics, Nanopore, Benchmarking, Microbiology, Taxonomic classifier

Swedish University of Agricultural Sciences
Faculty of Veterinary Medicine and Animal Science
Department of Animal Biosciences

Abstract

Metagenomic Next Generation Sequencing is increasingly being adopted for clinical diagnostic use as a valuable complement to traditional methods of pathogen detection. After major improvements in accuracy, Oxford Nanopore sequencing represents a viable alternative to Illumina. With longer read lengths, lower cost and shorter turnaround times, Nanopore can reduce the time to diagnosis and improve patient outcomes. To realize this potential, sensitive taxonomic classifiers with support for long-read data are needed. In this study we benchmark the performance of different taxonomic classifiers on Nanopore-sequenced viral data from both mock and real clinical samples. Custom databases for the *Kraken2*, *DIAMOND*, *Metabuli*, *MetaCache* and *Sylph* classifiers were built from the same reference sequence data. Classifications were compared to *BLAST* alignments as a gold standard and the classifiers were evaluated in terms of sensitivity and precision.

Metabuli and *MetaCache* were the most sensitive across datasets and for different viruses, at the cost of long processing times and *high* memory requirements respectively. *Kraken2* showed excellent precision but was the least sensitive. *DIAMOND* performed well on the mock data but had lower species-level sensitivity on the shorter read length clinical data, likely reflecting the limited specificity of protein-based classifiers. *Sylph* was highly computationally efficient and in combination with *Minimap2* performed well for most viruses, but was unable to detect some low-coverage genomes at default thresholds. Given enough memory, *MetaCache* may be the most suitable classifier for a diagnostic workflow, possibly in combination with *DIAMOND* to leverage the benefits of both DNA- and protein-based classifiers.

Keywords: Bioinformatics, Metagenomics, Nanopore, Benchmarking, Microbiology, Taxonomic classifier

Acknowledgements

I would like to thank my supervisors Tobias Allander, Sofia Stamouli and Lili Andersson-Li for the ideas, guidance, engagement and support they've provided throughout the course of this project.

Table of contents

List of figures

Abbreviations

Abbreviation	Description
AA	Amino acid
CNS	Central nervous system
HSV-1	Human simplexvirus type 1
HSV-2	Human simplexvirus type 2
LCA	Lowest common ancestor
mNGS	Metagenomic Next Generation Sequencing
ONT	Oxford Nanopore Technologies
Taxid	Taxonomic ID
VZV	Varicella zoster virus

1. Introduction

In recent years, the use of metagenomic Next Generation Sequencing (mNGS) for clinical diagnostic purposes has seen rapid development. As a broad-range diagnostic test, it offers untargeted detection of nucleic acid from all microbial pathogens, and it may further be used in the managing of immunocompromised patients or to predict antimicrobial resistance genes (Fourgeaud *et al.*, 2024; Gan *et al.*, 2024). Additionally, it can be used before starting treatment with immunosuppressants to minimize the risk of complications from an ongoing infection (Fourgeaud *et al.*, 2024). One area where mNGS has already proven valuable is for use in patients with central nervous system (CNS) infections, for whom the differential diagnosis is often broad and the supply of sample material limited. In a 7-year study conducted in the USA mNGS was found to offer superior sensitivity to indirect serologic and direct detection testing (Benoit *et al.*, 2024).

Implementing mNGS in routine use has the potential to transform the diagnostic landscape. For example, a British diagnostic method for bacterial lower respiratory infections using mNGS was recently described, cutting turnaround times down to 6h for diagnostic results, compared to typical culture times of 48-72h (Charalampous *et al.*, 2019). The key to achieving such rapid turnaround times was the use of Oxford Nanopore Technologies (ONT) long-read sequencing, which allows for real-time sequence analysis (Greninger *et al.*, 2015). Traditionally, the short-read Illumina sequencing technology has been the standard mNGS platform, largely because of a lower error rate and a higher throughput. With the introduction of the most modern ONT technology and chemistry however, Nanopore sequencing appears to have largely closed the quality gap, with substantially reduced error rates. (Ratcliff *et al.*, 2024; Sanderson *et al.*, 2024).

In order to leverage the improvements in ONT sequencing fully for diagnostic metagenomics, highly accurate software is needed to analyse sample reads and detect pathogens. Although several taxonomic classifiers are available, until recently few were designed specifically to make use of ONT long read data. Today the list of long read classifiers include *MetaMaps* (Dilthey *et al.*, 2019), MMSeqs2 (Steinegger and Söding, 2017), *MetaCache* (Müller *et al.*, 2017), *Metabuli* (Kim and Steinegger, 2024), *Sylph* (Shaw and Yu, 2024) and others. Past studies comparing taxonomic classifier performance on long-read data have often made use of methods designed for short-reads (Leidenfrost *et al.*, 2020). This is beginning to change, as several long-read taxonomic classifiers have been included in benchmarking papers in recent years, reflecting the rapid development of classifiers for ONT data (Portik, Brown and Pierce-Ward, 2022; Buddle *et al.*, 2024).

As metagenomic Nanopore sequencing is increasingly being adapted for clinical diagnostic use, there is a growing need for the evaluation of long-read taxonomic classifiers. In particular, these tools need to be evaluated specifically for the clinical use case. Generally speaking, clinical metagenomics aims to identify a single or a few pathogens in a human sample, rather than to characterize the composition of an environmental microbiome for example. It is therefore important that any classifier intended for routine diagnostic use be evaluated thoroughly on clinically relevant sample types and pathogenic species.

1.1 Taxonomic classifiers

A common class of taxonomic classifiers are the DNA-based classifiers, which work by matching nucleotide reads to reference genomes at the nucleotide level. This category includes k-mer-based tools such as *Kraken2* (Wood, Lu and Langmead, 2019) and *CLARK* (Ounit *et al.*, 2015) or alignment-based methods such as *MegaBLAST* (Morgulis *et al.*, 2008). The DNA-based classifiers are effective at distinguishing between closely related and well-studied taxa, using the relatively fast pace of change at the genomic level for increased specificity.

Another category is made up of protein-based classifiers, which translate nucleotide reads for comparison against protein reference sequences. This category includes classifiers such as *DIAMOND* (Buchfink, Reuter and Drost, 2021), *Kaiju* (Menzel, Ng and Krogh, 2016) or *MMseqs2* (Steinegger and Söding, 2017). Protein-based classifiers are more effective at detecting the homology of novel or underrepresented species to closely related taxa, using the higher conservation of sequence at the amino acid level. DNA- and protein-based classifiers are sometimes combined in a hybrid method to leverage the benefits of both types. Typically, this will consist of a classification with a DNA-based classifier first, followed by processing unclassified reads with a protein-based classifier (Yang, Jiang and Zhang, 2014).

Marker-based classifiers make up a third category, making use of a curated set of marker genes through which different species can be differentiated. The advantages of marker-based methods include smaller databases and faster runtimes, at the cost of less flexible database customization. Classifiers in this category include MetaPhlAn2(Truong *et al.*, 2015) (16) and mOTUs2 (Milanese *et al.*, 2019).

One of the most important factors for classifier performance is the choice of database with respect to the quantity and quality of the included reference sequences. While most classifiers have pre-built databases available, the databases for different classifiers will not necessarily be built from the same reference data. For any comparison of classifiers to accurately reflect their differences in capabilities and to avoid introducing potential biases, it is therefore important to use custom classifier databases built from a common set of reference sequences

(Van Uffelen *et al.*, 2024). Marker-based classifiers are an exception to this rule, since they are typically designed to make use of highly specific databases.

1.1.1 Kraken2

Kraken2 is a widely used k-mer-based classifier, designed primarily for short reads (Wood, Lu and Langmead, 2019). It works by splitting a sequenced read into k-mers, or subsequences of length k. Each k-mer is then compared to the reference genomes in a database and mapped to the lowest common ancestor (LCA) taxon of the matching genomes. Read classification proceeds on a weighted taxonomic subtree, made up of the LCA taxa and their ancestor nodes, where each taxon is weighted by the number of k-mers it contains. The weights in each root-to-leaf path is then summed, and the read is assigned to the taxon corresponding to the leaf node of the highest scoring path. Ties are resolved by assigning the read to the LCA taxon of all tied leaf nodes (Wood and Salzberg, 2014).

1.1.2 Metacache

MetaCache is another DNA- and k-mer-based method that was developed to address shortcomings of classifiers like the original *Kraken* (Müller *et al.*, 2017), such as the need to sacrifice either sensitivity or precision when choosing a k-mer length. *MetaCache* is designed to work with short or long reads. It uses a pair of hash functions to efficiently map a read to local subsequences on the reference genomes, using a subset of the read k-mers. The read is then assigned to the genome containing the region with the peak k-mer count intensity. In the case of a tie, the LCA of the tied genomes is used. In the original *MetaCache* paper, *MetaCache* is reported to outperform *Kraken* in terms of sensitivity and precision on bacterial Illumina data.

1.1.3 Metabuli

The ONT-ready *Metabuli* classifier introduces a novel type of k-mer, the metamer, which combinees a DNA 24-mer and its translated amino acid (AA) 8-mer sequence (Kim and Steinegger, 2024). The database is constructed from reference genomes and gene prediction is used to identify open reading frames (ORFs) for translation and metamer extraction. For the sequence reads, metamers are generated using the six-frame translations of the DNA sequence. Read classification works by matching read metamers to reference metamers with identical amino acid sequence, using the similarity at the DNA level to filter out too distant matches. Matches are grouped by species, translation frame and location on the read, and each species is scored by looking at paths of continuous matches covering the read, with differences on the DNA level used to weigh

matches. Similarly to *Kraken2*, the read is classified as the species with the highest score, or in the case of a tie to the LCA of the species with equal scores.

The main benefit promised by *Metabuli* is that it can leverage the strengths of both DNA-based and protein-based classifiers. In benchmarks on synthetic and real metagenomes *Metabuli* was shown to perform better than individual DNA- or protein-based classifiers, and to match optimal choices of hybrid classifiers (Kim and Steinegger, 2024).

1.1.4 Sylph

Sylph is a DNA- and k-mer based classifier, which constructs a taxonomic profile based on estimating the containment average nucleotide identity (ANI) of reference genomes within a sample metagenome (Shaw and Yu, 2024). *Sylph* profiling works by first subsampling k-mers from the reference genomes and the sample reads to sketches, and then computing the k-mer containment of the reference genome sketches in the sample sketch. For each reference genome, the k-mer coverage in the sample is modelled using a zero-inflated Poisson statistical model, with the inflated frequency of 0-coverage k-mers being due to mutations in the reference genome compared to the sample metagenome. The λ parameter of the Poisson distribution is known as the effective coverage, which is a function of the true genome coverage, k-mer lengths, read lengths, the sequencing error rate and sequencing depth. It is used to compute the coverage adjusted ANI, an estimate of the true genome ANI. The last profiling step is the reassignment of sample k-mers that are shared between reference genomes to the genome with the highest estimated ANI. The final profile includes all reference genomes with an estimated ANI above a threshold of 95% by default.

Unlike classifiers like *Kraken2*, *Sylph* does not output read-level classifications. *Sylph's* reported features include rapid processing times and a small footprint in terms of memory and storage space (Shaw and Yu, 2024).

1.1.5 DIAMOND

DIAMOND is a protein-based aligner with support for long reads (Buchfink, Reuter and Drost, 2021). It was designed as a faster alternative to *BLAST* at the cost of sensitivity, and intended for metagenomic applications. In its blastx (translated search) mode *DIAMOND* aligns translated sample reads to a database of protein reference sequences. To accurately detect genes for noisy and error-prone long read data, *DIAMOND* blastx makes use of frameshift alignments. This means that *DIAMOND* will align the translated sequence from all three reading frames at the same time against the reference, allowing arbitrary frameshifts to be tolerated at the cost of a scoring penalty. For Illumina short-read data *DIAMOND's* blastx mode was reported to be around 2000-20 000 times faster than *BLASTX*, depending on the use of *DIAMOND's* fast or sensitive mode

(Buchfink, Xie and Huson, 2015). The reported sensitivity was around 75% and 90% of *BLASTX* respectively.

1.2 Aim

The aim of this project is to benchmark the performance of the taxonomic classifiers *Kraken2*, *DIAMOND*, *Metabuli*, *MetaCache* and *Sylph* for shotgun metagenomics using long-read metagenomics viral data sequenced with Oxford Nanopore. It is hypothesized that the long-read *DIAMOND*, *Metabuli*, *MetaCache* and *Sylph* classifiers will perform competitively with the widely used *Kraken2* in terms of sensitivity and precision.

2. Materials and methods

2.1 Taxonomic classifier software

All the taxonomic classifiers used in this benchmarking study were installed from the Bioconda software distribution (Grüning *et al.*, 2018) using the Conda package manager. The evaluated classifiers are summarised in Table 1, including the versions used throughout this study.

Table 1. A list of benchmarked taxonomic classifiers.

Classifier	Type	Version used
DIAMOND	Protein-based	2.1.11
Kraken2	DNA-based	2.1.3
Metabuli	Mixed	1.1.0
MetaCache	DNA-based	2.4.3
Sylph	DNA-based	0.8.0

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at UPPMAX, funded by the Swedish Research Council through grant agreement no. 2022-06725. Most analyses were performed on the high-performance computing system UPPMAX Bianca.

2.2 Viral mock community dataset

Publicly available Oxford Nanopore Technologies (ONT) data was acquired from ENI (project PRJEB74559), consisting of 10 DNA samples (accessions ERR13488549-555, ERR13488749-750 and ERR13485822) and 9 RNA samples (accessions ERR13488556-563 and ERR13488751). The data was derived from viral mock community samples prepared by Buddle *et al*. (Buddle *et al.*, 2024), designed to resemble clinical samples with known viral composition. The dataset consists of dilutions of the MSA-1008 Virome Nucleic Acid Mix (ATCC) in background human DNA or RNA. The dataset will be referred to as ONT MSA-1008. It consists of a total of 47 million reads between 76 bp and 185 kbp long, with most reads concentrated between 1500 to 4000 bp (Figure 1) and with an N50 of 3669 bp.

Table 2. A list of the viral species composition of the ONT MSA-1008 dataset. *The genome lengths for the viruses included in the ATCC MSA-1008 virome mix are taken from the manufacturer. The genome lengths for Bacteriophage MS2 and Lambdavirus are taken from the GCF_000847485.1 and GCF_000840245.1 RefSeq genomes respectively.*

Species	Baltimore classification	Genome length (bp)

Bacteriophage MS2	+RNA	3569
Human betaherpesvirus 5	DNA	216 303
Human mastadenovirus F	DNA	34 392
Human orthopneumovirus	-RNA	15 228
Influenza B virus	-RNA	14 520
Lambdavirus	DNA	48 500
Mammalian orthoreovirus 3	dsRNA	23 416
Zika virus	+RNA	10 952

The MSA-1008 virome mix contains two DNA viruses *(Human betaherpesvirus 5* and *Human mastadenovirus F*) and four RNA viruses (*Mammalian orthoreovirus, human orthopneumovirus, Zika Virus* and *Influenza B virus*) at equal concentrations (see Table 2). The ONT MSA-1008 samples represent dilutions of the virome mix at 0, 60, 600, 6000 and 60 000 genome copies (gc) per mL, with a constant human background. The samples additionally contained spike-in internal controls consisting of *Lambdavirus* DNA for the DNA samples and *Bacteriophage MS2* RNA for the RNA samples. The ONT MSA-1008 dataset was generated on R.10.4.1 PromethION flow cells on a P2 Solo connected to a GridION, with Q20+ Kit V14 chemistry.

2.3 Clinical samples dataset

A dataset of ONT-sequenced clinical patient samples was obtained from Karolinska University Hospital. The samples were previously confirmed positive for one of 5 different DNA viruses (*adenovirus, bocavirus, herpes simplex virus type 1* (HSV-1), *herpes simplex virus type 2* (HSV-2) or *varicella zoster virus* (VZV)) using PCR. Of the samples, 3 were from nasopharyngeal tissue, 2 were from bronchoalveolar lavage fluid, 5 were from cerebrospinal fluid and 5 were from serum. For eight of the samples, the DNA had been prepared using both a standard extraction protocol and a human-depletion protocol consisting of three initial steps of centrifugation, bead beating and HL-SAN treatment. HL-SAN is a salt-active endonuclease which has been shown to be able to contribute to substantial depletion of human DNA (Charalampous *et al.*, 2019). After extraction, DNA was amplified using the REPLI-g kit (Qiagen). Sequencing libraries were prepared using Nanopore Rapid Barcoding Kit 96 V14 and sequencing performed using Promethion flow cells on a P2 Solo instrument.

The clinical samples dataset will be referred to as ONT Clinical. It is made up of 73 million reads between 1 bp and 735 kbp in length. The read length distribution is heavily skewed to the lower end of the range (Figure 1), with an N50 of 1001 bp.

Note that after performing the analyses on the ONT Clinical dataset, it was discovered that there was no HSV-2 reference genome included in the list of

RefSeq assemblies used to build the custom classifier databases. The HSV-2 samples were therefore not included in the results section of this report.

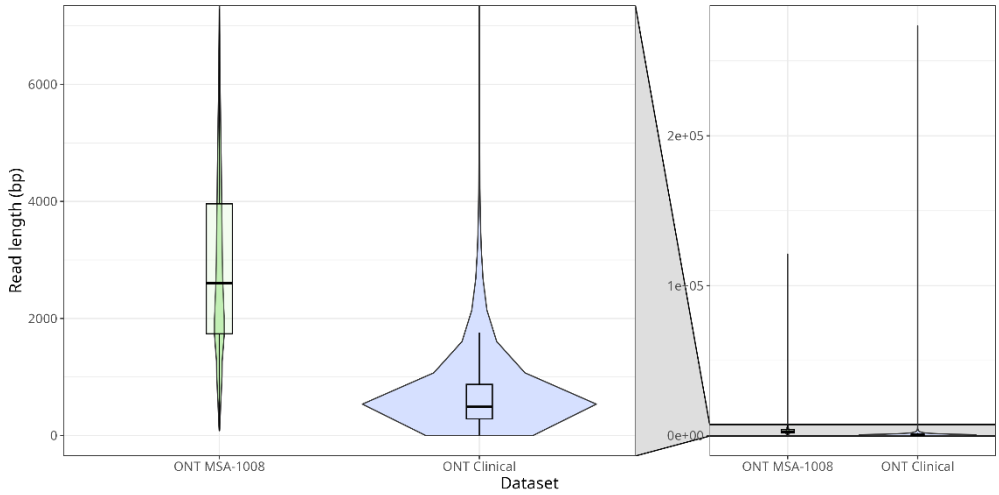


Figure 1. Read length distribution of clinical and mock viral community ONT datasets. *The distributions of 5.64 million subsampled reads from the ONT Clinical dataset and 4.35 million subsampled reads from the ONT MSA-1008 dataset are shown as violin plots. The boxplots show the median and first and third quartiles. The whiskers extend to 1.5 times the interquartile length (IQR) from the nearest hinge. The left pane shows a zoomed in view of the right pane.*

2.4 Building custom classifier databases

Custom databases for the different classifiers were constructed from a same set of sequences, corresponding to those found in the *Kraken2 PlusPF* database (https://benlangmead.github.io/aws-indexes/k2, release 9/4/2024). The *PlusPF* database contains RefSeq human (GRCh38 and T2T-CHM13), bacterial, archaeal, protozoal, fungal, viral and plasmid genomic sequences, as well as synthetic sequences from the Univec database (2). All FASTA files were decompressed before building the databases.

Files containing taxonomic information needed to build the custom databases were downloaded from NCBI (February 2025). The NCBI taxonomy name and tree files were downloaded from https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/. The accession to taxonomic ID mapping files *nucl_gb.accession2taxid*, *nucl_wgs.accession2taxid* and *prot.accession2taxid.FULL* were downloaded from https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/accession2taxid, and *assembly_summary_refseq.txt* was downloaded from https://ftp.ncbi.nlm.nih.gov/genomes/refseq/.

2.4.1 Building the *Kraken2* database

The Kraken2 custom database was built according to the *Kraken2* manual (https://github.com/DerrickWood/kraken2/blob/master/docs/MANUAL.markdow

n). After populating a taxonomy subdirectory with NCBI taxonomy name, tree and accession to taxid mapping files, the build process consisted of two main steps. First, FASTA files were processed to mask low-complexity regions and to create a library of hashed files. For this step, the k2 wrapper script added in *Kraken 2.1.3* was used. To speed up the library construction, multiple FASTA files were processed in parallel using xargs, as shown below.

```
cat <list-of-FASTA-files> \
          | xargs -P 16 -I{} -n 1 \
          | k2 add-to-library --db <db directory> --threads
1 --masker-threads 1 --file {}
```

In the second step, the database was built from the assembled FASTA library. For this step the k2 wrapper script was not used due to a bug (https://github.com/DerrickWood/kraken2/issues/942). Therefore, the older kraken2-build command was used instead.

```
kraken2-build --build --db <db directory> --threads 16
```

2.4.2 Building the *Metabuli* database

An NCBI-based custom *Metabuli* database was built according to instructions (https://github.com/steineggerlab/Metabuli?tab=readme-ov-file#ncbi-or-custom-taxonomy-based-database). Since *Metabuli* requires a single accession to taxid mapping file when building a database, the *nucl_gb.accession2taxid* and *nucl_wgs.accession2taxid* files were combined and the merged map file used as input to *Metabuli* to maximize the number of sequences mapped to taxa. The database was then built as below.

```
metabuli build <db-dir> \
          <list of fastas>.txt <merged>.accession2taxid --
taxonomy-path <taxonomy> --threads 16
```

where `<taxonomy>` contains the NCBI taxonomy name and tree files.

2.4.3 Building the *MetaCache* database

The *MetaCache* database was built according to the *MetaCache* authors' instructions (https://muellan.github.io/metacache/building.html). First, all FASTA files to be included in the database were copied to a new directory and the database was then built as below.

```
metacache build \
           <db name> <directory with FASTA files> -taxonomy
<taxonomy> -taxpostmap <taxonomy>/nucl_*.accession2taxid
```

where `<taxonomy>` is a directory containing the NCBI taxonomy files.

2.4.4 Building the *DIAMOND* database

To achieve equal taxonomic representation between the DNA-based databases and *DIAMOND's* amino-acid-based database, protein sequences of the RefSeq plasmids and the coding sequences of the genomic assemblies included in the *Kraken2 PlusPF* database were downloaded from NCBI. Those sequences were then used to construct the custom *DIAMOND* database. As some protein sequences were duplicated between the plasmid collections, the assembly sequences and assemblies from related species, all protein sequences were first merged and processed with *SeqKit* ((Shen *et al.*, 2016)3) to remove duplicate sequences.

```
seqkit rmdup <duplicated FASTA> \
          > <deduplicated FASTA>
```

The deduplicated FASTA sequences were then used to build the DIAMOND custom database.

```
diamond makedb --in <deduplicated FASTA> \
          --db <db name> \
          --taxonmap prot.accession2taxid.FULL \
          --taxonnodes nodes.dmp \
          --taxonnames names.dmp \
          --threads 16
```

2.4.5 Building the Sylph database

The *Sylph* database was built according to the *Sylph* cookbook (https://github.com/bluenote-1577/sylph/wiki/sylph-cookbook). Following the recommendation to sketch small genomes at lower subsampling rates (`-c` flag), all viral genomes were sketched at `-c 1` and all other genomic sequences were sketched at `-c 100`. Sketching all genomes at `-c 1` was unfeasible as the system ran out of memory after sketching only a small fraction of the genomes.

Sylph requires a tsv file that provides the taxonomic lineage for each FASTA file used to build the database. Prior to building the database, custom bash scripts were therefore used to extract the taxid of every genome assembly using the information in the NCBI *assembly_summary_refseq.txt* file, as well as to extract all plasmid and UniVec sequences to new FASTA files. This resulted in a new library of FASTA files containing sequences of only one taxid each. These FASTA files were used to build the *Sylph* database as below.

```
sylph sketch -c 1 -l <viral genome FASTAs>.txt \
          -t 16 -o <viral output>
sylph sketch -c 100 -l < non-viral genome FASTAs>.txt \
          -t 16 -o <non-viral output>
```

The NCBI *names.dmp* and *nodes.dmp* taxonomy files were then used together with custom python scripts to generate a custom *Sylph* taxonomy tsv-file to allow for the later conversion of Sylph genome profiles of metagenomic samples to taxonomic profiles.

2.5 Read preprocessing

The preprocessing of the ONT sequencing data was carried out in accordance with the workflow of the *nf-core/taxprofiler* pipeline (Stamouli *et al.*, 2023). The *nf-core/taxprofiler* pipeline is intended to act as a unified front-end to several different metagenomic taxonomic profilers and databases within a single pipeline run for both Illumina and ONT data. The recommended preprocessing workflow consists of quality control, optional adapter removal and length- and/or read quality-based filtering, followed by removal of human host reads.

2.5.1 Viral mock community data

To increase comparability across samples, all reads in the ONT MSA-1008 dataset were randomly subsampled to 5Gbps using *Rasusa* (Hall, 2022) before further preprocessing. Since the samples showed distributions of predominantly long and high-quality reads, no length- or quality-based filtering was performed. No adapter trimming was performed either, as all adapters in the data have been removed prior it being uploaded to ENI. Removal of host reads was performed by mapping the samples to the T2T-CHM13v2.0 human reference genome with *Minimap2* (Li, 2018) and discarding all mapped reads.

2.5.2 Viral clinical sample data

Since the ONT Clinical data represent samples from different patients, consist of different sample materials and were DNA-extracted using two different protocols, no initial subsampling of the sample reads was performed. The data showed a high degree of extremely short reads, down to 1bp in length. To decrease the number of uninformative reads, the samples were therefore filtered using *nanoq* (Steinig and Coin, 2022) to exclude all reads shorter than 50 bp. Adapter trimming was not performed and host removal was carried out the same way as for the ONT MSA-1008 data.

2.6 Taxonomic classification of sample reads

Following removal of human host reads, each sample was processed by all five classifiers to generate taxonomic profiles. For each classifier, default options were used. In some cases some non-standard values were used to increase sensitivity or to speed up the processing time. Such non-standard options are noted below for

each classifier. All input sample files were gzipped fastq files and all classifiers were configured to make use of all 16 available CPU threads.

2.6.1 Classification with *Kraken2*

To reduce the classification time when running multiple samples, the *Kraken2* custom database was preloaded into memory prior to read processing and samples were then classified with *Kraken2* using the `--memory-mapping` flag. This prevents *Kraken2* from loading the database in and out of memory between each sample, allowing *Kraken2* to act in a batch-like mode for more rapid processing. Individual samples were classified as below.

```
k2 classify \
       --db <db in shared memory> --threads 16 --memory-
mapping --output <output file> --report <report file>
<sample fastq file>
```

2.6.2 Classification with *DIAMOND*

Samples were classified using *DIAMOND's* blastx mode, aligning translated reads to the reference protein sequences. Due to issues with processing some samples, the fastq input files were first converted to FASTA format before classification. The `--long-reads` flag was used to optimize ONT read processing. To reduce the sample processing time at the expense of higher memory use, the `--block-size` and `--index-chunks` parameters were set to non-default values. Classifications were performed as:

```
diamond blastx \
          --db <database> --out <output> --threads 16 --
outfmt 102 --long-reads --block-size 6 --index-chunks 1 --
query <FASTA>
```

where `–outfmt 102` ensures that a taxonomy is assigned to each read using an LCA algorithm.

2.6.3 Classification with *Metabuli*

The *Metabuli* classifier was run in sequencing mode 3 `(--seq-mode 3`), corresponding to long read data. Samples were processed as below.

```
metabuli classify \
      --seq-mode 3 <input fastq> <database> <output> <output
prefix> --threads 16 --taxonomy-path <NCBI taxonomy>
```

2.6.4 Classification with *MetaCache*

Similarly to *Kraken2*, *MetaCache* was run in a batch-like mode, processing samples one after another without unloading the database in between. In

MetaCache this is known as "interactive mode" and is activated when no query is supplied as an argument. To enable processing samples in batches, queries are added to a shell variable, one line per query, which is then piped to *MetaCache* to be processed as a batch. An example run with two queries is executed according to

```
queries="-out <output query 1> <fastq query 1>\n"
queries="${queries} -out <ouput query 2> <fastq query 2>\n"
echo -e ${queries} \
          | MetaCache query <database> -threads 16 -taxids -
separate-cols -lowest species
```

where the `-taxids` and `-separate-cols` flags alters the output file format to include taxids in a separate column, and `-lowest species` tells *MetaCache* to assign classifications at the species level as the lowest taxonomic rank.

2.6.5 Classification with *Sylph*

Sample fastq files were sketched at a subsampling rate (`-c`) of 1 according to

```
sylph sketch -t 16 -c 1 -r <fastq files>
```

and the sketched files were then profiled with *Sylph* using

```
sylph profile \
          <db directory>/*.syldb *.sylsp -u --min-number-
kmers 3 -t 16 -o <output>
```

where `--min-number-kmers` is used to lower the default (50) number of k-mers that are needed to get a result, as recommended for small genomes such as viruses. The `-u` flag makes it so that *Sylph* estimates the true genome coverage instead of the effective coverage, making the estimated sequence abundance proportional to the total number of sample reads. Finally, the *Sylph* genomic profiles are converted to taxonomic profiles using the *sylph-tax* utility and the *Sylph* taxonomy generated previously in the database construction section.

```
sylph-tax taxprof <Sylph results file> -t <Sylph taxonomy>
```

2.6.6 Combined *Sylph* and *Minimap2* method

To compensate for *Sylph's* limitation in not providing read-level classifications, an alternative combined method was devised utilizing *Minimap2* in addition to *Sylph*. For each sample profiled with *Sylph*, the detected genomes were matched to a species taxonomy using the *assembly_summary_refseq.txt* file. A genome was considered positive if it belonged to an expected species of the processed sample. The sample reads were mapped to each positive genome using *Minimap2* and the alignments were saved as a sorted BAM file using *SAMtools* (Danecek *et al.*, 2021):

```
minimap2 -a -x map-ont -t 16 <genome> <input fastq> \
          | samtools view -b - \
          | samtools sort -m 7G -o <BAM output>
```

If multiple genomes of the same species were detected, the genome with the highest breadth of coverage at a minimum depth of 1x was selected for that species. The breadth of coverage was computed by dividing the number of bases with coverage ≥ 1x by the genome length. The reads mapped to the selected genome were then extracted using *SAMtools* to acquire a set of individually classified reads.

```
Samtools view -F 0x4 <BAM output> | cut -f 1 > <reads file>
```

2.7 Classifier species-level viral abundances

2.7.1 Absolute abundances from normalized read counts

Normalized viral abundances were derived from the individually classified reads output by the *Kraken2*, *Metabuli*, *MetaCache*, *DIAMOND* and *Sylph + Minimap2* classifiers. For each metagenomic sample, the read classifications were compared against the viruses expected to be found in the sample. Only taxa at the species-level were considered: classifications at lower taxonomic levels were summarised to their ancestor species node while any classification above the species-level was disregarded. In other words, any classification at or below the species level of an expected virus was counted as a positive read for that species.

For the ONT MSA-1008 dataset, the expected viral composition was known beforehand exactly down to the species level, with every expected virus corresponding to exactly one taxon. For the ONT clinical dataset, the expected species was known beforehand only for the PCR-confirmed HSV-1, HSV-2 and VZV samples, while the exact species of *adenovirus* or *bocavirus* present were not known beforehand. To avoid biasing the results, reads from these samples were therefore considered positive if they were classified as any *adenovirus* or *bocavirus* species known to infect humans, as determined from the NCBI taxonomy browser (Schoch *et al.*, 2020). In total, 63 such *adenoviruses*, including *Human Mastadenovirus A-F*, as well as 5 different *bocavirus* species were included.

For the ONT MSA-1008 data, positive DNA and RNA virus reads were only counted from the DNA and RNA samples respectively. The total positive read counts were computed per species and viral load, and the raw read counts were normalized by the total number of sequenced sample reads and converted to reads per million (RPM). ONT Clinical samples were treated similarly: samples were grouped by PCR-verified species and the extraction protocol used (+/- HL-SAN method). The within-group positive read counts were then summed and normalized to RPM abundances.

2.7.2 Estimation of *Sylph* sequence abundances

Unlike the other benchmarked classifiers, *Sylph* does not provide read-level classifications but instead estimates the containment of reference genomes within a metagenomic sample. Abundances in terms of normalized read counts therefore could not be calculated directly, but were instead estimated. For each of q genomes assigned to a profiled metagnome, *Sylph* outputs a sequence abundance estimate according to

$$\frac{\lambda_i \cdot GL_i}{\sum_{i=1}^{q} \lambda_i \cdot GL_i},$$

where λ_i is the effective coverage of genome i and GL_i is the corresponding genome length in bp. The total fraction of classified sample reads is estimated as

$$\frac{\sum_{i=1}^{q} \delta_i \cdot GL_i}{\sum_{i=1}^{n} RL_i},$$

where δ_i is the true coverage of the genome and RL_i is the read length in bp of the ith read of a total of n sample reads. When run with the `-u` flag, *Sylph* estimates the true coverage instead of the effective coverage and the sequence abundance is multiplied by the estimated fraction of classified reads, becoming

$$\frac{\delta_i \cdot GL_i}{\sum_{i=1}^{n} RL_i},$$

which is the fraction of all sample base pairs assigned to genome i. This is then taken as an approximation of the fraction of sample reads assigned to the genome, and the read count was estimated by multiplying this fraction with the total number of input reads. The estimated read counts were then normalized to RPM as described previously.

2.7.3 Mock viral community relative abundances

For each species in the MSA-1008 virome mix, genome length normalized read counts were calculated for each taxonomic classifier at the different viral loads by dividing the classified read count with the genome length (see Table 2). The normalized read counts were divided by the sum of normalized read counts for all species to generate relative genome length adjusted abundances.

2.8 Estimation of Sensitivity, Precision and F1 score

2.8.1 Verification of read-level classifications using *BLAST*

For each host-filtered sample, all unmapped reads were aligned to a subset of the *nt_viruses* database using *BLAST+* (version 2.15.0) with the MegaBLAST (Morgulis *et al.*, 2008) module. To speed up the alignments, a list of NCBI taxids was used to limit the search space to viruses within the same taxonomic order as the viral species that were expected to be found in the sample. To filter out spurious hits, thresholds were set to limit detection to sequences with at least 90% nucleotide identity, an e-value of 10^{-5} and a minimum 50% query coverage of the reference. The `-duster` flag was also used to mask low-complexity regions, and the number of hits per read was limited to 10. A read was considered a true positive if at least one hit sequence belonged to a taxon at or below the species level of one of the expected viruses for the sample of that read, otherwise the read was considered negative.

The read classifications of each taxonomic classifier were compared to the BLAST results. If a read was positive for the same species using both the classifier and *BLAST*, it was considered a true positive (TP). If a positive classification was not confirmed by *BLAST* (either negative or positive for a different species) the read classification was considered a false positive (FP), and if the classification was negative but there was an expected *BLAST* hit the read was counted as a false negative (FN) for that classifier.

2.8.2 Computation of detection metrics

The taxonomic classifier performance was quantified by computing read-level sensitivity, precision and F1 scores for each classifier. Using the true positives, false positives and false negatives from the *BLAST* verifications, the sensitivity was computed as the rate of true positive reads, according to

$$sensitivity = \frac{TP}{TP + FN}.$$

The precision was computed as the proportion of all reads classified as positive that were true positives, according to

$$precision = \frac{TP}{TP + FP}.$$

The F1-score is defined as the harmonic mean of sensitivity and precision and was computed as

$$F1 = \frac{2TP}{2TP + FP + FN}.$$

For the ONT Clinical data, the sensitivity, precision and F1 metrics were computed per biological DNA sample and aggregated according to the virus expected to be found in the samples. For each virus, the distribution of F1 scores was tested for differences between taxonomic classifiers using the non-parametric Kruskal-Wallis test at a significance level of 5%.

2.9 Classifier computational resource requirements

The wall time and peak memory usage for building the *PlusPF*-derived custom databases for the different taxonomic classifiers was measured using the *GNU find* utility. All computations were performed on cluster nodes with 16 CPU cores and either 128GB or 256GB of memory depending on classifier database size. All reference sequences used to build the databases were in uncompressed FASTA format.

For *DIAMOND* the time needed to merge and deduplicate the input FASTA files was included in the wall time. Similarly, for *Metabuli* the wall time includes the time needed to merge the *nucl_gb.accession2taxid* and *nucl_wgs.accession2taxid files*.

GNU find was also used to measure the wall time and peak memory usage when classifying one of the host-filtered ONT Clinical samples, consisting of 2.1 million reads and 1.6 Gbp. To investigate the impact of database caching and loading times, three different types of runs were measured:

- cold run, defined as a classification of a single sample without any pre-caching of the database.
- Warm run, an initial untimed cold run was performed with a different warmup sample, immediately followed by a timed classification of the target sample. The purpose of the warmup run was to measure the potential effect of database caching between sample runs.
- Batch run, measured only for *Sylph*, *Kraken2* and *MetaCache* and defined as the processing of samples in batches without unloading of the database from memory in between samples. For each batch run the target sample was processed together with the warmup sample, and the additional time required for the target sample was measured.

For *Kraken2* and *MetaCache*, the batch runs were executed using memory mapping and interactive mode respectively, as described previously. For *MetaCache,* the target sample processing time was extracted from the classifier's standard output during execution and for *Kraken2* the target processing time could be measured directly using GNU time. For *Sylph*, the time added by the target sample was measured by first classifying the warmup sample by itself, followed by classifying the warmup and target samples together. The additional target

sample time was then calculated by subtracting the warmup sample time from the two-sample total time.

3. Results

3.1 Taxonomic classifier performance on mock data

The performance of *Kraken2*, *DIAMOND*, *Metabuli*, *MetaCache* and *Sylph* was first compared on ONT data from a simulated, mock viral community dataset. The dataset contained a known mix of constituent viruses at viral loads between 60-60 000 gc/mL in a background of human DNA or RNA with internal spike-in controls of *Lambdavirus* DNA and *Bacteriophage MS2* RNA. Following subsampling to 5Gb and removal of host-reads, the ability of the classifiers to detect reads at different viral loads was assessed in the form of per-species abundances (Figure 2A).

Among the tested classifiers, *DIAMOND*, *Metabuli* and *MetaCache* detected reads from 7 of the 8 expected species at 60 000 gc/mL, while *Kraken2* and *Sylph* detected 6 and 5 species, respectively. The *Lambdavirus* DNA was not detected by any classifier across all samples, although positive *BLAST*-aligned reads were present.

While both Kraken2 and Sylph failed to detect *Influenza B*, it should be noted that this virus was detected in only one read by the other classifiers and *BLAST*. Interestingly, Sylph also failed to detect *Human betaherpesvirus 5* at the default minimum average nucleotide identity (ANI) threshold of 95%, despite it being the largest of the investigated genomes and all other classifiers detecting it at viral loads down to 600 gc/mL.

Overall, the abundance estimates from the classifiers were similar to those obtained from *BLAST*, with a tendency for *Kraken2* to report lower abundances and the estimated *Sylph* abundances to be inflated up to a hundredfold for some species compared to *BLAST*. *Sylph* genome profiling followed by the mapping of reads to the detected genomes with *Minimap2* (*Sylph + Minimap2* method) resulted in abundances largely in line with those of *BLAST*.

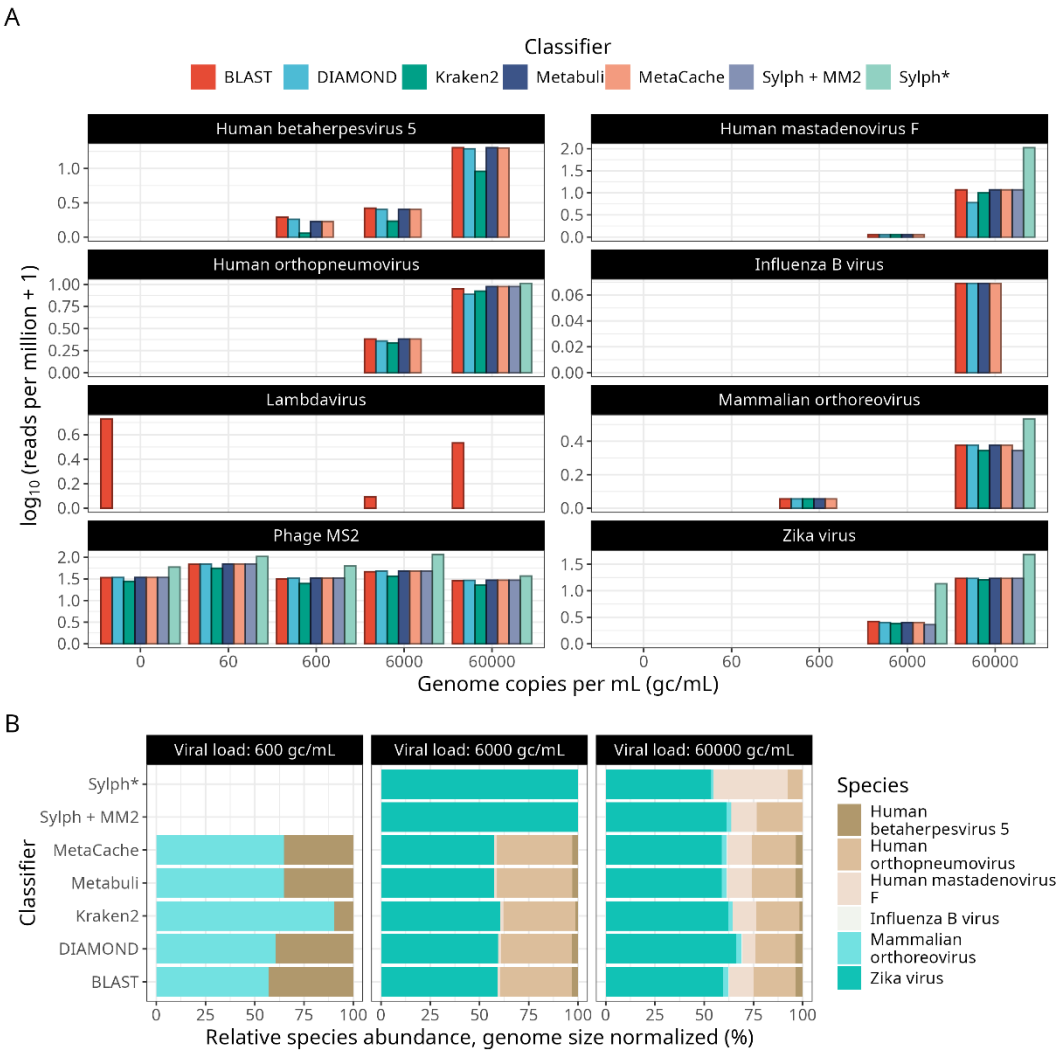


Figure 2. Abundances of detected species in mock viral community dataset for different classifiers and viral loads. *Each viral load corresponds to reads from two technical replicates each of one DNA and one RNA sample (except for 0 gc/mL with only one RNA replicate). Before classification, all samples were subsampled to 5Gb and host-reads were removed by mapping to the T2T-CHM13 human reference genome with Minimap2. (A) Normalized read counts in RPM for the classifiers, shown as base 10 log(x+1)-transformed values. RPM counts of BLAST-positive reads are included as a baseline. Phage MS2 and Lambdavirus represent the RNA and DNA internal controls respectively and should be present at fixed concentrations. (B) The distribution of classified reads per species, relative to the total number of classified reads for different viral loads and classifiers. All read counts were normalized to the species genome size to compensate for sequencing bias of longer genomes. The internal controls are not included. Sylph + MM2 represent abundances based on reads mapped to a Sylph-identified genome using Minimap2.*
**: Sylph read counts without Minimap2 are estimates calculated as described in the text.*

Excluding the internal sequencing controls, the theoretical viral composition of the mock community was expected to be made up of viruses from the virome mix in equal concentrations. To evaluate the ability of the classifiers to accurately reflect the composition of the viral community, relative abundances normalized to

the genome size of each species were computed (Figure 2B). The purpose of using relative abundances normalized to genome size was to compensate for biased read counts between different species due to the higher probability of sequencing longer genomes. The results show that the theoretical composition is not reflected by the data, with Zika virus making up more than half of the normalized *BLAST* distribution at all viral loads with detectable viruses. Overall, the classifier distributions follow *BLAST* closely for detected species, with the exception of *Sylph* diverging significantly by inflating the proportion of *adenovirus*.

The accuracy of the taxonomic classification for the mock community was evaluated on the read level by computing sensitivity, precision and F1 detection metrics for each classifier across the different viral loads (Figure 3), using *BLAST* read alignments as a baseline to verify classifications. As shown in Figure 3, *Kraken2* displays the lowest sensitivity, correctly identifying around 70-80% of positive reads at different viral loads, while *Metabuli* and *MetaCache* held stable at over 95% sensitivity for all non-zero viral loads. The sensitivity of *DIAMOND* drops off sharply at the highest viral load, likely reflecting lower read counts assigned to the highly abundant *adenovirus* (see also Figure 1A for comparison). The sensitivity of *Sylph* meanwhile suffers at the higher viral loads, mainly because it was able to detect fewer species. While *Kraken2* shows the highest precision, reaching 100% at 0, 6000 and 60 000 gc/mL, the other classifiers are close behind with values between 96% and 99%, such that the F1 scores closely resemble the sensitivity curve.

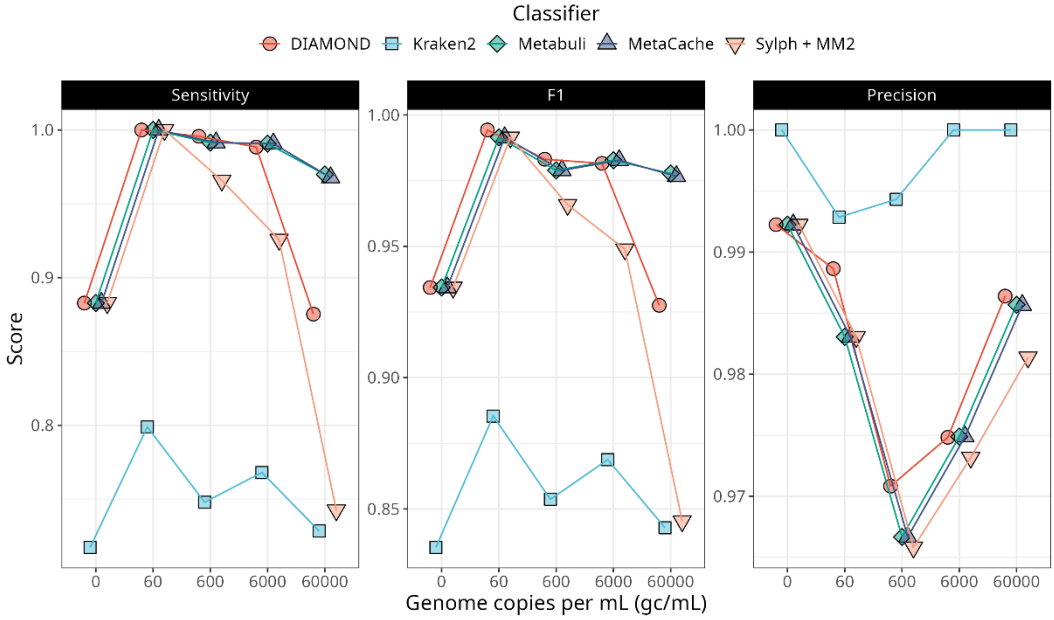


Figure 3. Detection metrics for different classifiers on mock viral community dataset. *Sensitivity, precision and F1 scores were calculated from the number of true positive, false positive and false negative reads, using a pairwise comparison between the read*

taxonomies assigned by each classifier andto per-read BLAST alignments. All species expected to be found in the samples, including internal controls, were included in the calculations, . Since Sylph does not report per-read classifications, the reported metrics for Sylph are instead based on counts of reads that were first mapped to a Sylph-identified genome using Minimap2 (MM2)

3.2 Taxonomic classifier performance on clinical data

Having evaluated ONT-ready classifiers on a mock viral community, the classifiers were next benchmarked on an ONT dataset from clinical DNA samples containing known PCR-verified DNA viruses. Following read preprocessing and host-removal, the read counts of the target viruses were normalized using reads per million (RPM). The normalized per-virus read counts from different samples were then aggregated to compute abundances for both HL-SAN and non-HL-SAN DNA extraction protocols (Figure 4).

As a first observation, the samples treated with HL-SAN had substantially higher abundances in terms of RPM compared to the standard protocol samples, likely reflecting a higher proportion of viral genomic sequence being sequenced in the host-depleted samples. Further, similar to the mock community data *Metabuli, MetaCache* and the *Sylph + Minimap2* method consistently reported slightly higher abundances than those from *BLAST* alignments. While *Sylph* was able to detect all expected viruses, the abundances calculated from estimated *Sylph* read counts showed opposite trends between the +/- HL-SAN samples. For the non-HL-SAN samples, the estimates were about 2-3 times higher than the *BLAST* abundances, similar to the behaviour seen for the mock viral community data. In contrast, the estimated *Sylph* abundances for the HL-SAN+ samples were substantially lower than the *BLAST* values for 3 out of 4 viruses. *Kraken2* and *DIAMOND* abundances were lower than *BLAST* across all viruses, and this was especially pronounced for *Herpes Simplex Virus 1*.

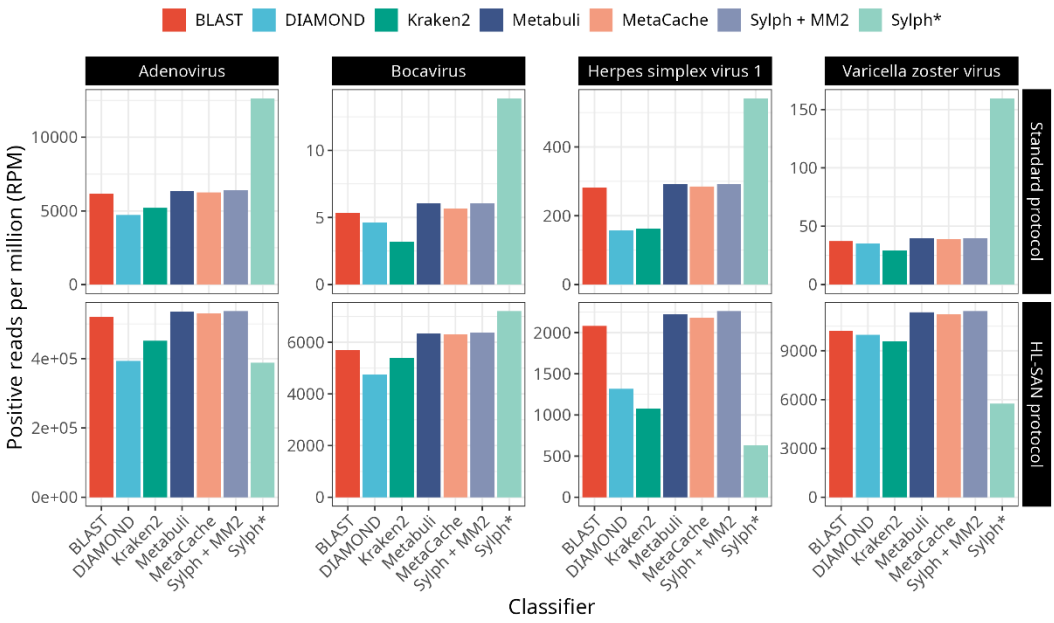


Figure 4. Normalized (RPM) abundances of detected species in ONT patient data for different classifiers and species. *The read counts include all classified reads at or below the species level, including any false positives, and are aggregated across PCR-positive DNA samples. The "HL-SAN protocol" abundances are derived from samples prepared using a HL-SAN host genomic depletion protocol, while the "Standard protocol" abundances are from samples that underwent DNA extraction without the HL-SAN step. Counts of BLAST-positive reads are included as a baseline for comparison.*

To investigate the relationship between classifier accuracy and viral species, read-level detection metrics were computed for the patient DNA samples and grouped by the known PCR-verified viral pathogen (Figure 5). A clear difference can be seen between the classifiers in terms of sensitivity, with *MetaCache*, *Metabuli* and *Sylph + Minimap2* consistently outperforming *Kraken2* and *DIAMOND*. *Metabuli* performed particularly well, detecting reads with near 100% sensitivity with the exception of one HSV-1-positive sample, for which none of the classifiers were able to detect any reads. The *Sylph + Minimap2* method also posted near-perfect sensitivity, but was unable to classify any reads from an additional HSV-1 sample. *MetaCache* meanwhile classified this and most other samples with sensitivity exceeding 95%.

For *Kraken2* and *DIAMOND* the sensitivity was around 70%-75% for *adenovirus* and *bocavirus*, increasing to 80%-85% for VZV. For both classifiers the sensitivity dropped considerably for the HSV-1 samples however, going down to 25% for *Kraken2* and around 40% for *DIAMOND*.

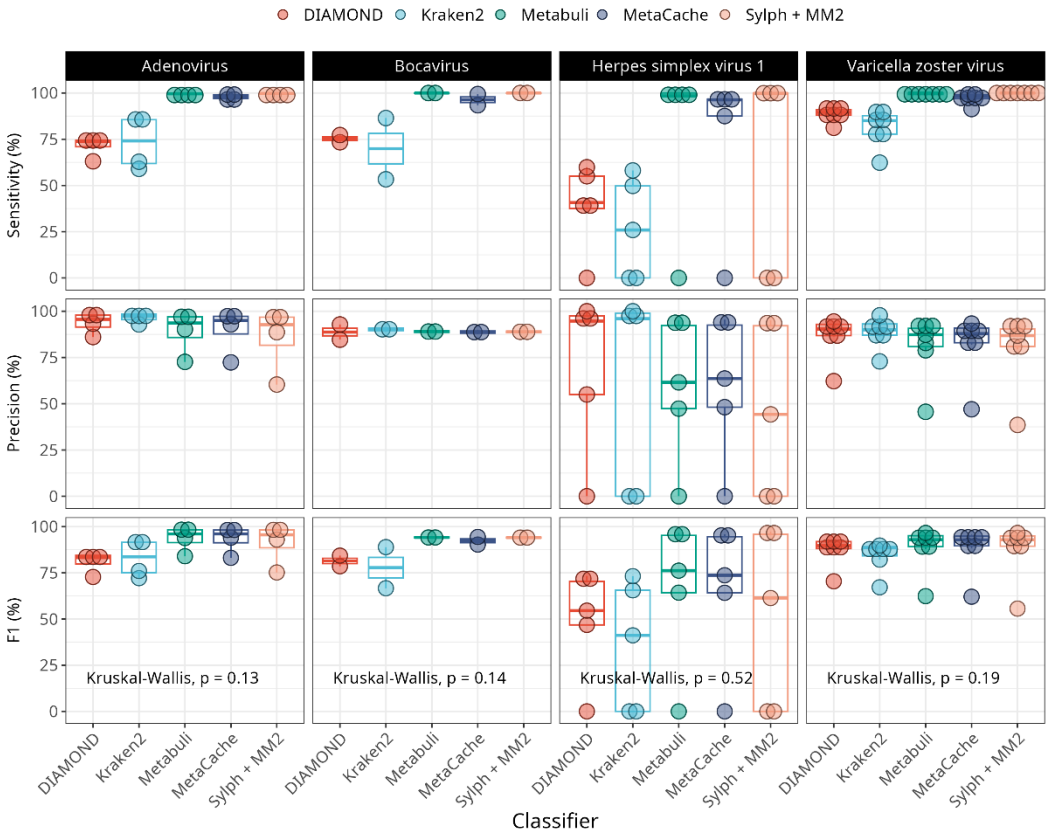


Figure 5. Detection metrics for different classifiers on patient ONT data. *Each dot represents a distinct DNA sample, with reads aggregated from one or two sequencing replicates using the same extraction protocol. The sensitivity, precision and F1 scores were calculated only for species that were previously PCR-confirmed to be present in the sample. Comparisons of classifier-assigned read taxonomies to per-read BLAST alignments were used to categorize reads as true positives, false positives or false negatives. The boxplots show the median as well as the first and third quartiles. The whiskers of the boxplot include all values within 1.5 x IQR from the nearest hinge. The p-values for Kruskal-Wallis tests of the F1 distributions are shown.*

All classifiers were able to detect true positive reads with high precision for *adenovirus*, *bocavirus* and VZV samples, with median values between 87% and 97%. The precision was overall lower for HSV-1-positive reads, with the more sensitive *Metabuli*, *Metacache* and *Sylph + Minimap2* methods reporting median precisions of 62%, 64% and 44%, respectively. *Kraken2* and *DIAMOND* had higher median precisions of around 95%.

When considering sensitivity and precision together, the distribution of F1 scores overall favored the *Metabuli*, *MetaCache* and *Sylph + Minimap2* methods, although no statistically significant differences were found for any of the viruses at a 5% significance level. For VZV, all classifiers performed similarly, while *Metabuli* and *MetaCache* were the strongest performers for classifying HSV-1 samples, with *Kraken2* bringing up the rear.

3.3 Benchmarking classifier computational requirements

When considiering the implementation of a taxonomic classifier within a routine clinical metagenomics workflow, it is important to take into account not only the classification performance but also the computational requirements of putting the tool into use. When comparing the wall time and memory needed to build the custom *PlusPF*-derived databases (Figure 6A), *Sylph* was the fastest tool by far, finishing in 9 minutes compared to the second fastest *DIAMOND* at 2 hours. *Kraken2* was the slowest at 21 hours, taking almost twice as long as *Metabuli*. In terms of memory usage, *DIAMOND* was the least taxing with a peak use of 6 GB of RAM, followed by *Sylph* at 23 GB. *MetaCache* required the most memory with a peak use of 191 GB, about twice as much as *Metabuli* or *Kraken2*.

Using the respective custom databases, the classifiers were further benchmarked in terms of the wall time needed to process a sample of 2.1 million reads (Figure 6B). For the purpose of classifying a single sample without pre-caching of the database, *Metabuli* took the longest with 8 hours, 14 times slower than *DIAMOND*. *Sylph* meanwhile finished in 1 min, while *Kraken2* and *MetaCache* followed closely at around 3 min. When processing the same sample after an initial untimed run with a different sample in order to take advantage of potential database caching, only *Sylph*, *Kraken2* and *MetaCache* showed small improvements in speed. Running the sample as part of a batch run resulted in greater speed gains, with *Sylph* and *MetaCache* reducing the processing time by around 85%, while *Kraken2* finished almost 75% faster. This indicates that *Sylph* and *MetaCache* provide competitive processing times compared to *Kraken2*, whereas *DIAMOND* and especially *Metabuli* may slow down the analysis workflow. It should be noted that since *Sylph* reports classifications only at the genome level, additional tools such as *Minimap2* must be used to map sample reads to an assigned genome if reads-level data is desired. This may considerably extend the sample processing time, depending on the number, size and complexity of the *Sylph*-classified genomes to be mapped.

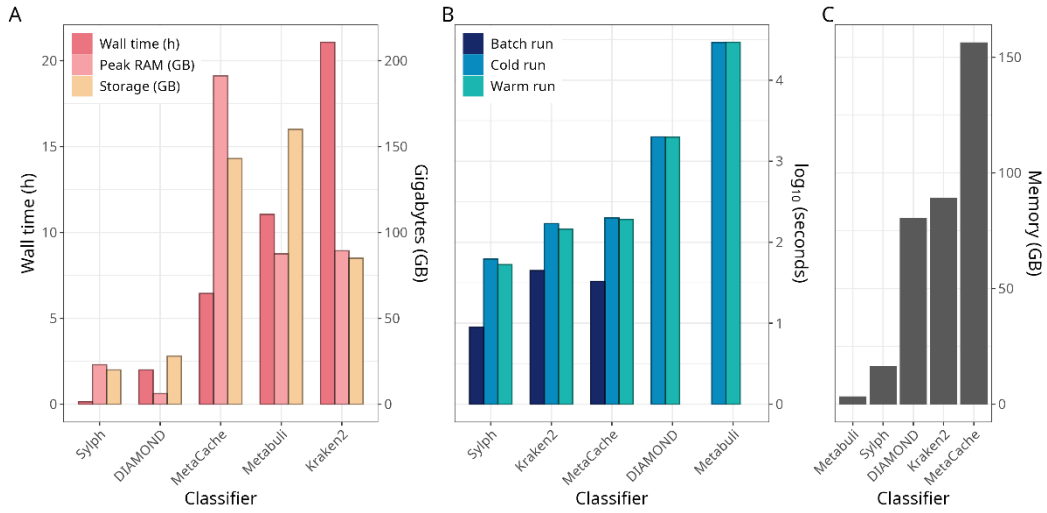


Figure 6. Benchmarking classifier computational resource use. *(A) The wall time and memory used to build a PlusPF-derived custom database for different classifiers, sorted by wall time in ascending order. (B) The wall time required by different classifiers to profile a 1.6 Gb ONT sequenced patient sample of 2.1 million reads. Base 10 log-transformed times are shown in ascending order. Cold run times show the classification time for a single sample, without any pre-caching of the database. Warm run times show the classification time after an initial (untimed) classification of a different sample, showing potential speedups due to database caching. Batch run times are shown for classifiers that support batch processing of samples (Sylph, MetaCache) or database memory mapping (Kraken2), and represent the processing time for the second sample in a batch run. (C) The peak memory usage by the classifiers when processing the sample as in (B).*

4. Discussion

ONT sequencing has gained popularity for metagenomic analyses both in research and clinical diagnostics due to its long read length, short turnaround time, and lower cost. While taxonomic classification tools for ONT data are continuously being developed, comprehensive benchmarking of those remains limited. In this study, we evaluated five taxonomic classifiers for ONT metagenomics data using both a mock viral community and clinical samples.

4.1 Classifier performance on mock viral community

4.1.1 Lambdavirus may be misclassified as *E. coli*

In this study we failed to detect the internal *lambdavirus* DNA control. This was also observed in the original paper of the dataset (Buddle *et al.*, 2024), where *lambdavirus* reads were being classified as *Escherischia coli* (*E. coli*) due to the presence of an integrated lambda phage in *E. coli* reference genomes. To check if similar results were observed across the classifiers in this study, the classifications of the 33 reads with positive *BLAST* hits for *lambdavirus* were cross-checked for *Kraken2, DIAMOND, Metabuli* and *MetaCache*. All reads were classified as *E. coli* by MetaCache compared to none by *Kraken2*, 1 by *DIAMOND* and 7 by *Metabuli*. Among the reads not classified as *E. coli*, all were unclassified by *DIAMOND*, while *Metabuli* classified 4 as bacteria above the species level. *Kraken2* classified 20 reads as Eukaryota (10 at the species level and 10 at higher levels) and 10 as bacteria at the species level.

Since the *BLAST* verification was based on the virus-only *nt_viruses* database, integrated *lambdavirus* and *E.coli* sequences couldn't be accurately predicted. To address this, the reads with *lambdavirus BLAST* hits were also aligned to the *nt_core* database using the NCBI *BLAST* web tool. All reads gave strong hits of over 90% identity to both *E. coli* and *lambdavirus*. These results are compatible with the conclusion drawn by Buddle *et al.*, that *lambdavirus* DNA tends to be misclassified as *E. coli* and that *lambda* DNA may therefore not be suitable for use as an internal sequencing control for metagenomics.

4.1.2 *Sylph* may struggle to detect viruses at low coverage

Sylph was notably the only benchmarked taxonomic classifier that failed to detect *Human betaherpesvirus 5* at any viral load. This was surprising given the previously reported high sensitivity of *Sylph* on nanopore data ('Genomic and epigenomic insights into microbial biology with nanopore metagenomic and isolate sequencing', 2024), combined with the detection of the virus by all other classifiers at lower viral loads. To further investigate this unexpected result, the

ONT MSA-1008 samples were reprofiled using *Sylph* with lower thresholds for the minimum containment ANI (the `--minimum-ani` parameter) required for a reference genome to be included in the *Sylph* profile. At a minimum ANI of around 85% *Sylph* was able to detect not only *Human betaherpesvirus 5* at viral loads down to 600 gc/mL but also *Influenza B* virus at the highest load, bringing *Sylph*'s detection rate up to the level of *Metabuli*, *MetaCache* and *DIAMOND*.

A possible explanation for *Sylph* not detecting *Influenza B virus* at a minimum ANI of 95% is that the *Influenza* viruses are known to have variable genomes (Tsai and Chen, 2011). To explore this the single *Influenza B BLAST*-positive read was aligned to the reference genome used to build the *Sylph* database, with a best match of 92% identity. This could indicate that the default minimum ANI of 95% may be too stringent for profiling highly variable viruses with *Sylph*.

For the less variable *Human betaherpesvirus 5* DNA virus a different explanation is likely needed. While it is known that *Sylph* detects viruses with lower sensitivity than bacteria due to the smaller genome size (Shaw and Yu, 2024), this does not apply for *Human betaherpesvirus 5* given that it has the largest genome of all the viruses in the ONT MSA-1008 data. We therefore speculated that the failure to detect the virus may also have been due to low coverage of the reference genome in the sequencing data.

Sylph estimates the containment ANI using an effective coverage-adjusted measure, which can compensate for low genome coverage. For bacteria *Sylph* has been shown to correct ANI estimates to >95% for effective coverages as low as 0.008x (Shaw and Yu, 2024) when the true ANI was >99%, although the estimates became lower and less accurate with decreasing coverages. We therefore hypothesized that the ANI estimate may break down below a certain coverage level, and that this detection limit would depend on both the true genome ANI as well as the genome size.

To investigate this, the viral genome coverage in the host-filtered metagenomic samples was checked by mapping the sequenced reads to the reference genomes identified by *Sylph* with *Minimap2* and then calculating the breadth of coverage at a minimum of 1x coverage depth. The viral genomes detected at a minimum ANI of 95% all had coverages above 36%, whereas the coverage of the *Influenza B* genome was at 11%. For the *Human betaherpesvirus 5* genome coverages ranged from 2% to 25% between samples. This indicates that while the 95% ANI threshold may be sufficient for low-abundance bacterial genomes, it may not apply in the same way to viral genomes. Due to their smaller genome sizes, we would expect *Sylph* to require a higher minimum coverage of a reference genome to reach the 95% containment ANI estimate for viruses compared to bacteria. Given that the viruses that *Sylph* failed to detect at the 95% ANI threshold were also those with the lowest coverage, it seems likely that low viral genome coverage may explain the limited detection rate of *Sylph* on this dataset. This also

suggests that the containment ANI approach of *Sylph* may put it at a disadvantage for the detection of low-abundance viruses compared to classifiers that operate on the level of single reads. To achieve higher sensitivity for such samples, an adjustment of the minimum ANI level appears to be necessary.

4.2 Classifier performance on clinical data

4.2.1 DIAMOND's lower sensitivity on clinical samples

On the mock viral community dataset the performance of *DIAMOND* was equivalent to that of *Metabuli* and *MetaCache* for most viral loads, showing high precision and clearly outperforming *Kraken2* in terms of sensitivity. When classifying reads from real patient samples however *DIAMOND* showed consistently lower sensitivity than *Metabuli* and *MetaCache*, performing at a similar level as *Kraken2*. The sensitivity of *DIAMOND* as well as *Kraken2* was especially low for HSV-1 reads compared to the other viral species. We investigated this by checking the HSV-1 *BLAST*-positive reads that *KRAKEN2* and *DIAMOND* failed to detect. For *DIAMOND* there were 10 944 such reads, out of which 83% were classified as *Simplexvirus* at the genus level, meaning that *DIAMOND* was failing to distinguish between the reference sequences of HSV-1 and closely related viruses. This is in line with protein-based classifiers being known to have lower specificity due to higher sequence conservation at the protein level (Kim and Steinegger, 2024), resulting here in classifications that fail to resolve below the genus level. This then manifests as lower sensitivity when classifications above the species level are discarded.

For *Kraken2* only around 10% of the false negative reads were correctly classified at the genus level, with 90% being unclassified. This indicates the opposite problem of *DIAMOND*, with *Kraken2* being unable to detect conserved homology between divergent nucleotide sequences. Of note, 67% of *Kraken2* and 59% of *DIAMOND* false negatives reads were not shared between the two classifiers, indicating that the classifiers were indeed struggling to classify the reads for different reasons.

A noticeable difference between the mock and clinical datasets is that the clinical samples had many more short reads compared to the mock data (Figure 1). Longer read lengths have previously been associated with higher specificity (Buddle *et al.,* 2024). Since *DIAMOND* is a protein-based classifier, it may be that its performance suffers more from the loss of long-read information, given that there is an additional reduction in sequence length from the translation of DNA to protein. Additionally, as genes are more conserved on the amino acid-level compared to the nucleotide-level, it is likely that longer reads would be especially helpful for protein-based classifiers in resolving closely related sequences. In a previous benchmarking study on ONT data by Portik *et al*. (Portik, Brown and

Pierce-Ward, 2022), a *DIAMOND+MEGAN* protein-based method saw a sharp drop in performance when the dataset was altered to include a higher proportion of short reads, while the performance of *Kraken2* was mostly unaffected. This is largely in line with our results, although a direct comparison is difficult to make due to the use of metrics at the level of species detection in Portik *et al*.'s study, instead of the read-level metrics used here. Taken together, this suggests that the shorter read length in the clinical dataset likely explains some of the observed loss of sensitivity for *DIAMOND*.

4.2.2 *Metabuli*, *MetaCache* and *Sylph + Minimap2*

The superior performance of *MetaCache* over *Kraken2* seen in this study is in agreement with the observations of *MetaCache's* authors (Müller *et al.*, 2017). This may reflect the use of shorter default k-mer lengths in *MetaCache*, which is supposed to increase sensitivity without reducing specificity due to *MetaCache's* context aware minhashing method. Meanwhile, *Metabuli* is designed to achieve both high specificity and homology sensitivity by simultaneously encoding sequence information at the nucleotide and the amino acid-level (Kim and Steinegger, 2024), which could help explain its strong performance also on the seemingly challenging HSV-1 data. Unlike the mock data, *Sylph* detected the viruses in all clinical samples except two HSV-1 samples that had 1 and 8 *BLAST*-positive reads respectively. Given the previous discussion on the likely impact of genome coverage on *Sylph's* sensitivity, the improved performance may be explained by the higher viral abundances observed overall for the clinical data, which is likely to correlate with higher genome coverage. Finally, the fact that *Minimap2* classified reads in strong agreement with *BLAST* is not surprising, as the two alignment-based tools have previously been found to report similar abundances on metagenomic data (Bahk and Sung, 2024).

4.3 Computational performance

The computational benchmarks performed in this study showed significant differences between the compared taxonomic classifiers in terms of time, memory and storage requirements. All classifiers support building custom databases, allowing for a uniform set of reference sequence data to be used. The database build time varied considerably, with *Sylph* taking only minutes to complete a database from over 200GB of sequence data. While the database construction step may typically not be the most time critical in a metagenomics workflow, it can still be advantageous to be able to quickly modify or update the database with minimal delay. The *Sylph* database was also highly memory efficient, with the memory and storage requirements being adjustable through the use of different k-mer subsampling rates. These results are in line with previous benchmarks of *Sylph* (Shaw and Yu, 2024). Notably, the *MetaCache* database needed more than

twice the memory of *Kraken2*, indicating that the previously reported (Müller *et al.*, 2017) advantage of *MetaCache* over the original *Kraken* in terms of memory efficiency no longer holds after
the improvements made in *Kraken2* (Wood, Lu and Langmead, 2019). It should be noted that *MetaCache* provides an option to build a database in several parts for reduced memory use, although this was not explored in this study.

In processing sample reads there was a clear division of performance in terms of speed between the classifiers. Importantly, through processing samples in batches *Sylph, Kraken2* and *MetaCache* could process samples up to 100 and 1000 times faster than *DIAMOND* and *Metabuli* respectively. In terms of memory use *Metabuli* stood out by using only a few GB, which could explain its slow long processing times. Without the tweaks made to *DIAMOND* as reported in the methods it used very little memory at the cost of substantially longer processing times. Attempts were made to similarly tweak *Metabuli's* memory use through the `–max-ram` parameter to achieve shorter processing times, but this appeared to have no effect.

It is important to note that processing times for the combined *Sylph + Minimap2* method used to get classifications at the level of individual reads are not presented. This is because the time added will depend on the number of detected genomes, which of these the user is interested in having mapped to reads as well as the size and complexity of the genomes.

4.4 Taxonomic classifier pros and cons

From an analysis of the benchmarking results it is clear that the optimal choice of classifier will depend on the specific use case and computational resources available to the user. In making a recommendation we will approach this question from the perspective of a clinical use case, where the goal of performing the metagenomic analysis is typically to detect a single viral pathogen in a patient sample. This is unlike other perhaps more common use cases for these tools, such as characterising the overall composition of a complex microbiome. In clinical use, a metagenomic assay is often performed for immunocompromised patients, or in situations where traditional methods have failed to identify an infectious agent. Important criteria for a suitable classifier for this use case therefore include high sensitivity under different viral loads and for different viral families, reflecting the diversity of sample types and pathogens found in the clinical setting.

In this study we have benchmarked the DNA-based methods *Kraken2, Metabuli, MetaCache* and *Sylph,* as well as the protein-based *DIAMOND*. The overall picture produced by the sensitivity, precision and F1 scores indicates that the *Metabuli, Metacache* and *Sylph + Minimap2* methods present substantially higher sensitivity in classifying viral reads from patient samples compared to Kraken2 and DIAMOND at the species level, at a comparatively small loss in

precision. Further, the improvement in sensitivity may be particularly pronounced for certain viral species such as HSV-1, potentially indicating that the performance of *Kraken2* and *DIAMOND* could vary across different viral species.

While *Kraken2* offered competitive computational performance, its limited sensitivity and poor ability to detect HSV-1 makes it difficult to recommend. *DIAMOND* performed very well overall on the synthetic data but saw a sharp drop in species-level read sensitivity when moving to the clinical samples, likely reflecting the cost in specificity of its greater ability to detect homology through conserved amino-acid sequence. As this feature could potentially be used to detect pathogens with poor representation in the reference database, such as novel viral strains, a potential use case for *DIAMOND* could be as a secondary classifier in combination with a highly sensitive DNA-based tool. *DIAMOND* would then be used on reads that the first tool did not classify correctly. Here *MetaCache* could be a suitable choice for primary classifier if its greater memory requirements can be satisfied, having performed well across datasets and all investigated species, while offering efficient processing times.

Metabuli meanwhile showed best-in-class sensitivity and potentially offers the strengths of both DNA- and protein-based classifiers in one tool, which could remove the need to combine DNA- and protein based classifiers. A serious drawback of *Metabuli* found in this study however was its highly inefficient sample processing times. Further research would also be needed to compare the performance of *Metabuli* and protein-based classifiers in terms of detecting underrepresented genomes through related species on clinical data, which was outside the scope of this study.

While *Sylph* performed best-in-class on the computational side, its poor performance on low-coverage viruses and inaccurate abundance estimates are serious drawbacks. *Sylph's* lack of read-level classifications are also a problem, as it makes results difficult to verify with a secondary method. We worked around this by mapping reads to *Sylph*-detected genomes with *Minimap2*, with overall strong results. A potential problem with this approach though is that individual reads could be assigned to different taxa, complicating the interpretation of the results and potentially adding further complexity to the workflow.

4.5 Limitations

There were several limitations of this study. First, the taxonomic classifiers were only evaluated on viruses, but from a clinical perspective their performance on bacterial pathogens is also highly relevant, and in particular for bacterial species that are challenging to culture. Second, although we only looked at a few viruses in this study, we could see differences in the performance of some taxonomic classifiers between species. This is an indication that the performances reported here may not generalize to other viral pathogens, and that further evaluations on

additional species is needed. It would also be valuable to evaluate different sample types such as feces, as the microbial load and diversity can vary greatly between different human sites.

We used mostly the default parameters of the evaluated classifiers. While outside the scope of this study, it is possible that careful tuning of classifier parameters could significantly alter some of the performances reported here.

4.6 Summary and conclusion

We evaluated the *Kraken2, DIAMOND, Metabuli, MetCache* and *Sylph* taxonomic classifiers on viral ONT datasets. Overall, *Metabuli* and *MetaCache* classified reads with the highest sensitivity and F1 scores across both mock and clinical datasets. *DIAMOND* performed well on the mock data but saw reduced species-level sensitivity on the clinical data, likely due to a shorter reads distribution. For use in a clinical diagnostic workflow, the DNA-based MetaCache is an attractive option due to its strong performance and fast processing times. *DIAMOND* could be used as a complement to leverage amino-acid level conservation to detect species with low representation in the reference database. *Metabuli* might combine the strengths of DNA- and protein-based classifiers in one tool, but suffered from slow processing times. Before adapting a classifier for clinical use, more research is needed using additional viral and bacterial species as well as different sample types.

References

Bahk, K. and Sung, J. (2024) ‘SigAlign: an alignment algorithm guided by explicit similarity criteria’, *Nucleic Acids Research*, 52(15), pp. 8717–8733. Available at: https://doi.org/10.1093/nar/gkae607.

Benoit, P. *et al.* (2024) ‘Seven-year performance of a clinical metagenomic next-generation sequencing test for diagnosis of central nervous system infections’, *Nature Medicine*, 30(12), pp. 3522–3533. Available at: https://doi.org/10.1038/s41591-024-03275-1.

Buchfink, B., Reuter, K. and Drost, H.-G. (2021) ‘Sensitive protein alignments at tree-of-life scale using DIAMOND’, *Nature Methods*, 18(4), pp. 366–368. Available at: https://doi.org/10.1038/s41592-021-01101-x.

Buchfink, B., Xie, C. and Huson, D.H. (2015) ‘Fast and sensitive protein alignment using DIAMOND’, *Nature Methods*, 12(1), pp. 59–60. Available at: https://doi.org/10.1038/nmeth.3176.

Buddle, S. *et al.* (2024) ‘Evaluating metagenomics and targeted approaches for diagnosis and surveillance of viruses’, *Genome Medicine*, 16(1), p. 111. Available at: https://doi.org/10.1186/s13073-024-01380-x.

Charalampous, T. *et al.* (2019) ‘Nanopore metagenomics enables rapid clinical diagnosis of bacterial lower respiratory infection’, *Nature Biotechnology*, 37(7), pp. 783–792. Available at: https://doi.org/10.1038/s41587-019-0156-5.

Danecek, P. *et al.* (2021) ‘Twelve years of SAMtools and BCFtools’, *GigaScience*, 10(2), p. giab008. Available at: https://doi.org/10.1093/gigascience/giab008.

Dilthey, A.T. *et al.* (2019) ‘Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps’, *Nature Communications*, 10(1), p. 3066. Available at: https://doi.org/10.1038/s41467-019-10934-2.

Fourgeaud, J. *et al.* (2024) ‘Performance of clinical metagenomics in France: a prospective observational study’, *The Lancet Microbe*, 5(1), pp. e52–e61. Available at: https://doi.org/10.1016/S2666-5247(23)00244-6.

Gan, M. *et al.* (2024) ‘Antimicrobial resistance prediction by clinical metagenomics in pediatric severe pneumonia patients’, *Annals of Clinical Microbiology and Antimicrobials*, 23(1), p. 33. Available at: https://doi.org/10.1186/s12941-024-00690-7.

‘Genomic and epigenomic insights into microbial biology with nanopore metagenomic and isolate sequencing’ (2024), 21 May. Available at: https://nanoporetech.com/resource-centre/genomic-and-epigenomic-insights-into-microbial-biology-with-nanopore-metagenomic-and-isolate-sequencing (Accessed: 15 May 2025).

Greninger, A.L. *et al.* (2015) ‘Rapid metagenomic identification of viral pathogens in clinical samples by real-time nanopore sequencing analysis’, *Genome Medicine*, 7(1), p. 99. Available at: https://doi.org/10.1186/s13073-015-0220-9.

Grüning, B. *et al.* (2018) ‘Bioconda: sustainable and comprehensive software distribution for the life sciences’, *Nature Methods*, 15(7), pp. 475–476. Available at: https://doi.org/10.1038/s41592-018-0046-7.

Hall, M.B. (2022) ‘Rasusa: Randomly subsample sequencing reads to a specified coverage’, *Journal of Open Source Software*, 7(69), p. 3941. Available at: https://doi.org/10.21105/joss.03941.

Kim, J. and Steinegger, M. (2024) ‘Metabuli: sensitive and specific metagenomic classification via joint analysis of amino acid and DNA’, *Nature Methods*, 21(6), pp. 971–973. Available at: https://doi.org/10.1038/s41592-024-02273-y.

Leidenfrost, R.M. *et al.* (2020) ‘Benchmarking the MinION: Evaluating long reads for microbial profiling’, *Scientific Reports*, 10(1), p. 5125. Available at: https://doi.org/10.1038/s41598-020-61989-x.

Li, H. (2018) ‘Minimap2: pairwise alignment for nucleotide sequences’, *Bioinformatics*, 34(18), pp. 3094–3100. Available at: https://doi.org/10.1093/bioinformatics/bty191.

Menzel, P., Ng, K.L. and Krogh, A. (2016) ‘Fast and sensitive taxonomic classification for metagenomics with Kaiju’, *Nature Communications*, 7, p. 11257. Available at: https://doi.org/10.1038/ncomms11257.

Milanese, A. *et al.* (2019) ‘Microbial abundance, activity and population genomic profiling with mOTUs2’, *Nature Communications*, 10, p. 1014. Available at: https://doi.org/10.1038/s41467-019-08844-4.

Morgulis, A. *et al.* (2008) ‘Database indexing for production MegaBLAST searches’, *Bioinformatics (Oxford, England)*, 24(16), pp. 1757–1764. Available at: https://doi.org/10.1093/bioinformatics/btn322.

Müller, A. *et al.* (2017) ‘MetaCache: context-aware classification of metagenomic reads using minhashing’, *Bioinformatics*, 33(23), pp. 3740–3748. Available at: https://doi.org/10.1093/bioinformatics/btx520.

Ounit, R. *et al.* (2015) ‘CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers’, *BMC Genomics*, 16(1), p. 236. Available at: https://doi.org/10.1186/s12864-015-1419-2.

Portik, D.M., Brown, C.T. and Pierce-Ward, N.T. (2022) ‘Evaluation of taxonomic classification and profiling methods for long-read shotgun metagenomic sequencing datasets’, *BMC Bioinformatics*, 23(1), p. 541. Available at: https://doi.org/10.1186/s12859-022-05103-0.

Ratcliff, J.D. *et al.* (2024) ‘Improved resolution of avian influenza virus using Oxford Nanopore R10 sequencing chemistry’, *Microbiology Spectrum*, 12(12), p. e0188024. Available at: https://doi.org/10.1128/spectrum.01880-24.

Sanderson, N.D. *et al.* (2024) ‘Evaluation of the accuracy of bacterial genome reconstruction with Oxford Nanopore R10.4.1 long-read-only sequencing’, *Microbial Genomics*, 10(5), p. 001246. Available at: https://doi.org/10.1099/mgen.0.001246.

Schoch, C.L. *et al.* (2020) ‘NCBI Taxonomy: a comprehensive update on curation, resources and tools’, *Database: The Journal of Biological Databases and Curation*, 2020, p. baaa062. Available at: https://doi.org/10.1093/database/baaa062.

Shaw, J. and Yu, Y.W. (2024) ‘Rapid species-level metagenome profiling and containment estimation with sylph’, *Nature Biotechnology*, pp. 1–12. Available at: https://doi.org/10.1038/s41587-024-02412-y.

Shen, W. *et al.* (2016) 'SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation', *PloS One*, 11(10), p. e0163962. Available at: https://doi.org/10.1371/journal.pone.0163962.

Stamouli, S. *et al.* (2023) 'nf-core/taxprofiler: highly parallelised and flexible pipeline for metagenomic taxonomic classification and profiling'. bioRxiv, p. 2023.10.20.563221. Available at: https://doi.org/10.1101/2023.10.20.563221.

Steinegger, M. and Söding, J. (2017) 'MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets', *Nature Biotechnology*, 35(11), pp. 1026–1028. Available at: https://doi.org/10.1038/nbt.3988.

Steinig, E. and Coin, L. (2022) 'Nanoq: ultra-fast quality control for nanopore reads', *Journal of Open Source Software*, 7(69), p. 2991. Available at: https://doi.org/10.21105/joss.02991.

Truong, D.T. *et al.* (2015) 'MetaPhlAn2 for enhanced metagenomic taxonomic profiling', *Nature Methods*, 12(10), pp. 902–903. Available at: https://doi.org/10.1038/nmeth.3589.

Tsai, K.-N. and Chen, G.-W. (2011) 'Influenza genome diversity and evolution', *Microbes and Infection*, 13(5), pp. 479–488. Available at: https://doi.org/10.1016/j.micinf.2011.01.013.

Van Uffelen, A. *et al.* (2024) 'Benchmarking bacterial taxonomic classification using nanopore metagenomics data of several mock communities', *Scientific Data*, 11(1), p. 864. Available at: https://doi.org/10.1038/s41597-024-03672-8.

Wood, D.E., Lu, J. and Langmead, B. (2019) 'Improved metagenomic analysis with Kraken 2', *Genome Biology*, 20(1), p. 257. Available at: https://doi.org/10.1186/s13059-019-1891-0.

Wood, D.E. and Salzberg, S.L. (2014) 'Kraken: ultrafast metagenomic sequence classification using exact alignments', *Genome Biology*, 15(3), p. R46. Available at: https://doi.org/10.1186/gb-2014-15-3-r46.

Yang, Y., Jiang, X.-T. and Zhang, T. (2014) 'Evaluation of a hybrid approach using UBLAST and BLASTX for metagenomic sequences annotation of specific functional genes', *PloS One*, 9(10), p. e110947. Available at: https://doi.org/10.1371/journal.pone.0110947.

Data availability

Scripts written in bash, Python and R used for running the analyses described in the methods section and to generate the figures in the report are made available in a git repository at https://github.com/bewh0001/metagenomics_taxclass_master_project.

Popular science summary

With metagenomics the characteristics of all genetic material from a sample can be analyzed simultaneously. Applied to diagnostic medicine, this means that the DNA from infectious organisms can be detected in patient samples using one test. This can provide a valuable complement to traditional tests, or be used to detect previously uncharacterized pathogens, such as new strains of an evolving virus.

Metagenomics relies on next generation sequencing technologies. The technology of choice has typically been Illumina, due to its high throughput and low error rate. The alternative Nanopore platform has become increasingly viable in recent years due to major improvements in sequencing accuracy. Nanopore can sequence DNA in much larger segments compared to Illumina, which can improve downstream analysis. Nanopore is also faster, allowing for a reduction in the time to diagnosis, and often cheaper. Metagenomics with Nanopore therefore has the potential to save costs and lead to improved patient outcomes.

The identification of pathogens from metagenomic data requires specialized taxonomic classifiers. A classifier uses efficient algorithms to compare sequenced segments to a database and identify the species. There are several classifiers for use with Nanopore data, but there is a need to compare their performance on clinically relevant data before being put in use. In this study we compared the performance of five different classifiers on Nanopore data derived from samples containing different viruses. To evaluate performance we computed sensitivity and precision. Sensitivity measures the ability of a classifier to find the sequenced DNA segments that belong to a given species, or so called true positives. A high precision meanwhile means that the classifier is unlikely to generate false positives, that is to assign the species to a segment in error.

Our results showed that the *Metabuli* and *MetaCache* classifiers were the most sensitive, finding almost all true positives. The widely used *Kraken2* classifier was considerably less sensitive, while *DIAMOND* performed better on data with a higher proportion of long sequenced DNA fragments. This is likely because *DIAMOND* works by first translating the DNA to protein, which is evolutionarily more conserved. This means that longer sequences may be needed to distinguish closely related species from each other. Finally, the *Sylph* classifier seemed to have problems with detecting viruses whose genomes were poorly represented by the sequenced fragments, which could make *Sylph* unsuitable for detecting small viral quantities. The computational requirements of the classifiers varied. *Metabuli* was slow, which could be a problem for analysis times. Overall, *MetaCache* may be the strongest candidate due to a combination of high accuracy and fast run times, but its high memory requirements could be a barrier to implementation. It may be a good idea to combine *MetaCache* with *DIAMOND*, in order to also get the benefits of a protein-based classifier.

Publishing and archiving

Approved students' theses at SLU can be published online. As a student you own the copyright to your work and in such cases, you need to approve the publication. In connection with your approval of publication, SLU will process your personal data (name) to make the work searchable on the internet. You can revoke your consent at any time by contacting the library.

Even if you choose not to publish the work or if you revoke your approval, the thesis will be archived digitally according to archive legislation.

You will find links to SLU's publication agreement and SLU's processing of personal data and your rights on this page:

- https://libanswers.slu.se/en/faq/228318

☒ YES, I, Benjamin Walsh, have read and agree to the agreement for publication and the personal data processing that takes place in connection with this

☐ NO, I/we do not give my/our permission to publish the full text of this work. However, the work will be uploaded for archiving and the metadata and summary will be visible and searchable.