# Benchmarking metagenomic classification tools for long-read sequencing data

Josip Marić[1,*], Krešimir Križanović[1,*], Sylvain Riondet[2,3], Niranjan Nagarajan[2,3,#] and Mile Šikić[1,2,#]

[1]Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia, [2]A*STAR Genome Institute of Singapore, [3] National University of Singapore

[*]Authors contributed equally

[#]To whom correspondence should be addressed.

## Abstract

We performed a comprehensive assessment of metagenomics classification tools on long sequenced reads. In addition to well defined mock communities, we prepared various synthetic datasets to simulate real-life scenarios. The results show that off-the-shelf mappers such as Minimap2 or Ram are at least comparable with mapping-based classification tools in most accuracy measures while not being much slower than kmer based tools and requiring equal or less RAM. Majority of tested tools are prone to report organisms not present in datasets and underperform in the case of high presence of host's genetic material. Furthermore, longer read lengths make classification easier, but due to the difference in read length distributions among species, the usage of only longest reads reduces the accuracy. Finally, evaluation on a mock community shows the importance of careful isolation of genetic material and sequencing preparation.

**Availability and implementation:** Python scripts used to generate all figures and tables in this study, and all supplementary texts and figures are available via the Github repository https://github.com/lbcb-sci/MetagenomicsBenchmark. Datasets, supporting files, analysis results and reports are available via Zenodo repository https://doi.org/10.5281/zenodo.5203182.

## Introduction

Imagine that one is interested in the analysis of a sequenced metagenomics sample. The study aims to provide information on present organisms and their quantity. However, the accuracy of the final result depends on many factors such as contamination with other genetic material (i.e. host's DNA), material isolation, sequencing preparation, used sequencing technology and classification tools. The recent improvement in both the length and accuracy of long-read sequencing technologies promises a more precise analysis. In this manuscript, we evaluated several tools for metagenomic sample analysis based on long-read whole metagenome de novo sequencing. In addition, we investigated the performance of tools for classifying present organisms using datasets that mimic routine experiments.

The advent of high-throughput sequencing has enabled a detailed analysis of microbial communities and their hosts through metagenomics[1,2]. Together with genetic material isolation, an essential component of metagenomic sequencing workflows is a computational method for recognizing organisms present in a sample. The majority of current methods are tailored to work with short, accurate reads from second-generation sequencing technologies. However, due to an increase in accuracy and throughput, long-read sequencing technologies are gaining popularity. Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT) are the most popular long-read sequencing technologies. Metagenomic sequencing approaches can be divided into marker gene (typically 16S rRNA) sequencing[3] and whole-metagenome shotgun sequencing. Since the 16S rRNA gene consists of both conserved and variable regions, it is suitable for cost-effective bacteria and archaea profiling. On the other hand, whole-metagenome shotgun sequencing covers all genomic information in a sample, enabling additional analyses such as binning, antibiotic resistance gene profiling, and metabolic function profiling. Metagenomic analysis pipelines often begin by detecting and quantifying the taxa in a sample. When most of the genomes present in the sample are unknown, metagenomic de novo assembly methods (i.e.[4]) are used. Otherwise, the sequenced data can be compared to a reference database that stores genomic information related to various taxa. This work aims to analyze the performance of methods based on the comparison of long-read sequencing data with a reference database. Although there are several benchmarking studies on long reads[5–7], our analysis includes both PacBio (including HiFi reads) and ONT sequencing technologies, incorporates an evaluation of the influence of the database on the results and assesses tradeoffs between running time and memory requirements in typical use cases with real sequencing data.

## Results

We tested eight metagenomic classification tools, which could be roughly divided into (1) kmer-based (Kraken2[8], Centrifuge[9], CLARK[10], CLARK-S[11]) and (2) mapping-based (MetaMaps[12], MEGAN-LR[13]; Minimap2[14], Ram[15]). We also evaluated Bracken[16], a statistical method that computes the abundance of species using taxonomy labels assigned by Kraken/Kraken2. Minimap2 and Ram are off the shelf mappers whose outputs we adapted for metagenomics classifications. Minimap2 was tested in two modes: full alignment mode (calculating alignment path) and mapping mode (calculating approximate alignments), giving us 10 tools in total.

We created datasets to highlight some common use cases in microbiology analysis using reads sequenced by Oxford Nanopore Technologies or Pacific Biosystem devices.

There are two main goals for classification algorithms: to identify species and to evaluate their abundances. Reaching these objectives highly depends on the community's content and the actual number of reads for each species. Therefore, using existing reads, we synthesised several simple to complex communities containing 3 to 50 species, with highly abundant to very sparse species.

- Datasets ONT1, PB1, PB4 reflect a community of bacteria without eukaryotic species.

- Datasets ONT2 and PB2 reflect metagenomics datasets with one or more eukaryotic species and many bacterial species.

- Dataset PB3 reflects a community with predominantly human reads (99 %) and two low abundance bacterial species, reflecting what one might see in an infection setting.

- Datasets PB1+NEG and PB2 represent a situation where a significant portion of the reads comes from an organism that is not present in the database and which has no similar organisms in the database. For the PB1+NEG dataset, those reads were obtained by generating "shuffled" reads using the human genome, while for the PB2 dataset, those are the reads belonging to *D. melanogaster* and human.

We also used three well defined mock community datasets PB Zymo, ONT Zymo and PB PB ATCC. It is important to notice that for synthesized communities, we used reads sequenced with older PacBio technologies, mock communities are sequenced using Sequel 2 hifi technology.

The tools were tested in four different areas:

1. Read level classification – how accurately can they classify each read.
2. Abundance estimation – how well can they be used to estimate the abundance of organisms in the sample.
3. Organism detection – how accurately can they detect organisms in a sample.
4. Computational resource usage - running time and consumption of RAM memory.

We focus our analysis on microbial species. Therefore, accuracy and abundance errors are calculated only for the microbial species, ignoring reads assigned to the human.

### Read level classification

In the first analysis, we assess the tools' read level classification accuracy on seven synthesized datasets. We analysed both species and genus levels. Figure 1 shows that mapping-based tools dominate on almost all datasets and on both levels. Differences between mapping-based and kmer-based tools vary up to 10 % at species levels. The only exception is MEGAN which performs similarly to kmer-based tools. Minimap2 with alignment overperforms other tools, followed by Minimap2 without alignment, Ram and MetaMaps. An interesting case is the ONT1 dataset which contains reads of two species of the Vibrio genus that are not in the database. Since there are other similar species of the Vibrio genus in the database, some tools, such as MEGAN and Minimap2, tend to assign those reads to other similar Vibrio species, while other tools, such as Clark-S and Ram, tend to leave those reads unassigned. Therefore, the results on the ONT1 dataset are almost reversed when analysing genus and species level of classification. Clark-S and Ram have the highest accuracy when inspecting the ONT1 dataset at the species level and lowest when inspecting the dataset at the genus level, while Minimap2 and MEGAN have the highest accuracy at the genus level but perform worse at the species level.

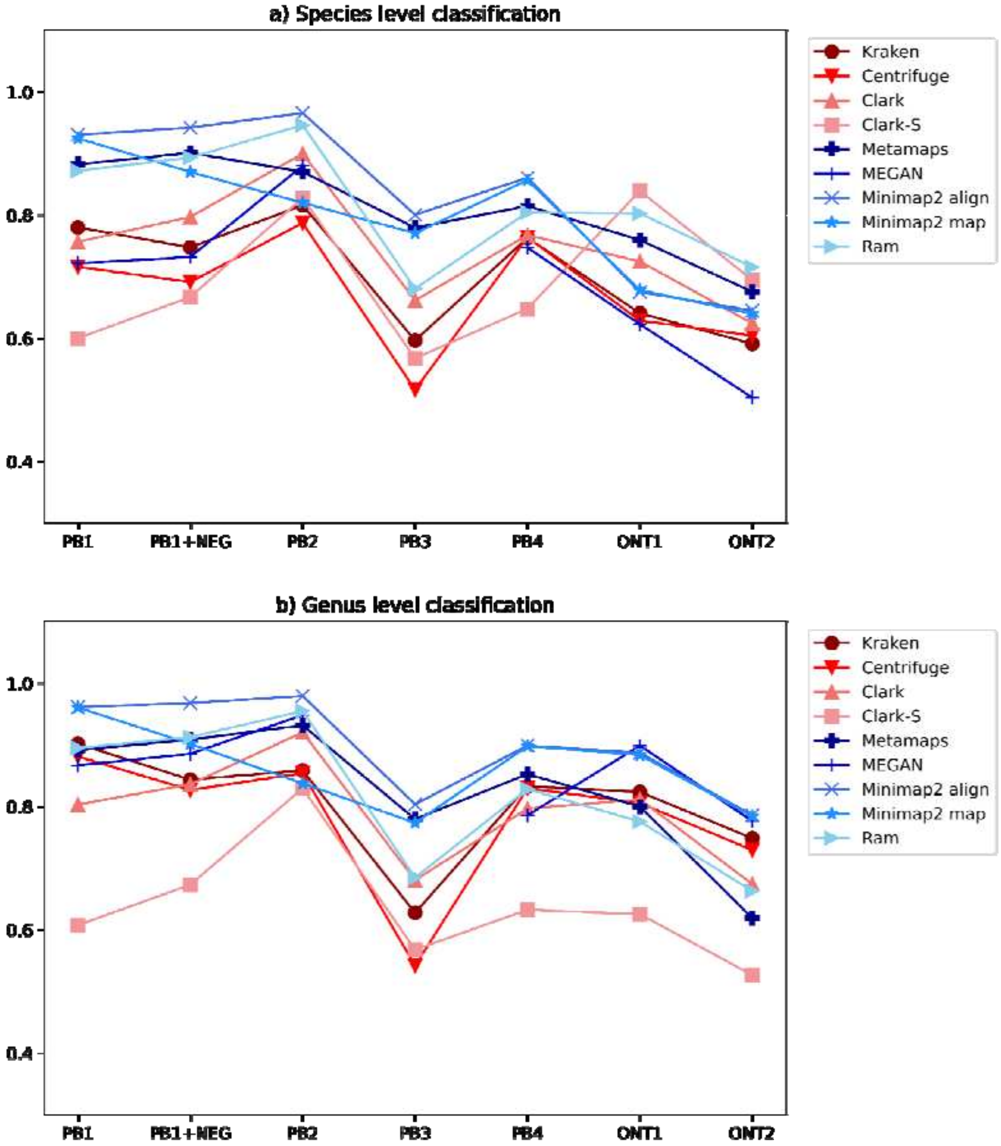


**Figure 1. Read level classification accuracy, comparison between species and genus level classification.** Kmer-based and mapping-based tools are represented in red and blue, respectively. Plot a) shows species level classification for which reads are considered correctly classified if classified to a correct species. Plot b) shows genus-level classification for which reads are considered correctly classified if classified to a correct genus. Results for MEGAN are unavailable for the PB3 dataset.

Since there is an imbalance in the number of reads per species, we also calculated the F1 score for each class (organism in the sample) separately and averaged them (F1 macro average). Using F1 macro average instead of accuracy shows a similar pattern for most datasets with a clear domination of Minimap2 and a smaller distance between mapping-based and kmer-based tools (Supplementary Figure 1).

We further investigated the influence of the read length on classification. We present only analysis for Minimap2 with alignment, the most accurate tool at the read level. As it is evident from Figure 2, increasing the read length increases the level of classification. However, due to different read length distributions per organism, we could not select only the longest reads. Detailed analysis on how the read length impacts the results is provided in Supplementary Table 2.

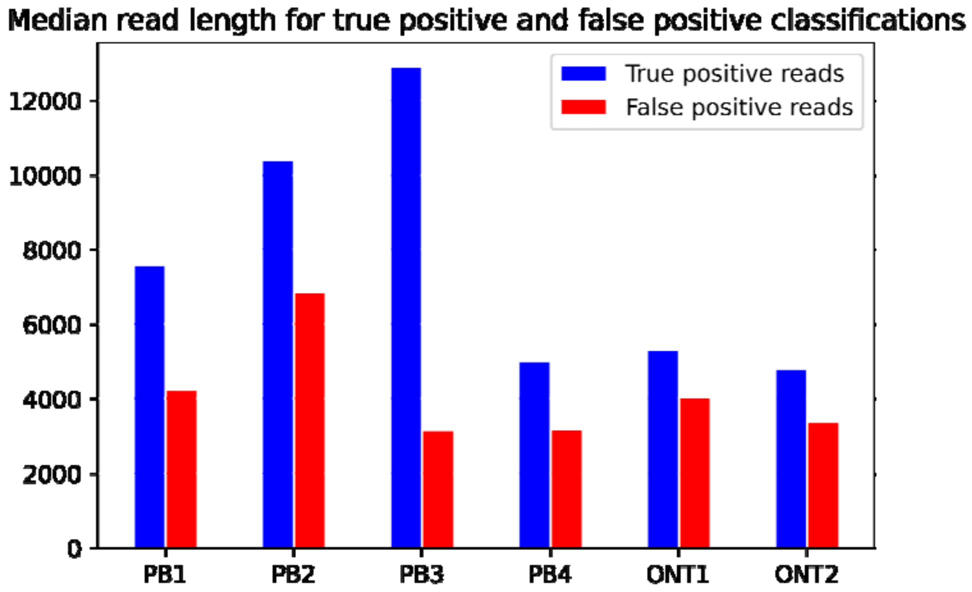


**Figure 2. Comparison between classification accuracy and read length.** The figure shows median read length for true positive and false positive read classifications for each dataset. The results shown in the figure were obtained using Minimap2 with alignment.

### The abundance estimation

The abundance estimation is arguably the most important assessment. In microbiology, the abundance of a species is defined as the ratio of cells in the community. However, most assessed tools report read counts instead, which does not take into account that larger genomes will yield more reads for the same number of cells. Supplementary Table 3 shows on real datasets that a measure that includes genome sizes performs similar or better than read counts. Therefore, we used read-level classification output from each tool to calculate the abundances, which are then compared between tools. How the abundance measure is calculated and compared is described in detail in the Methods section.

We analysed the abundances for seven synthesized and three real datasets on species level. Table 1 shows the results. For species present in datasets, we calculate the mean and std of the absolute difference between calculated and real abundance in percentages. Furthermore, we present calculated cumulative abundances of species not present in the datasets. Minimap2 outperforms other tools in absolute differences between abundances of present organisms. In most of the datasets, its mean difference is below 2%. However, other tools are not far away. PB3 dataset is specific due to the high percentage of human reads (99%). For this dataset, MetaMaps achieves the best results.

Regarding species not present in the dataset, CLARK-S surpasses others, followed by MetaMaps, Ram and MEGAN. Minimap2 is more prone to reporting organisms not present in the sample, and we deem there is space for improvement in the postprocessing analysis or by changing its parameters such as kmer length or the percentage of filtered kmers.

Supplementary Figure 2 shows a more detailed analysis of abundance errors for each tool and dataset.

Results on mock communities are similar among tools. Kraken2 and Bracken slightly overperform others in the abundance of species present in the database but usually reports more unexisting species in the sample. It is important to note that results for the PB_Zymo dataset are significantly worse than for the other two real datasets. Since all tools report similar results, we think that the problem for this dataset might have been in isolation of genetic material and preparation of the sample for sequencing.

**Table 1. Abundance estimation error in percentages on species level.** The abundance estimation error is calculated by comparing the abundances calculated for each tool to the ground truth. Errors are calculated separately within the dataset and outside the dataset. For organisms within the sample, the mean and standard deviation of the abundance error are displayed. For organisms outside the sample, the absolute value of the abundance error is summed up and displayed in the table. Each dataset name is followed by the number of species in that dataset in parentheses. Results for MEGAN are unavailable for datasets PB3, PB_atcc and PB_zymo. The best (lowest) values are printed in bold.

| Dataset (no of species) | In/Out of dataset | Kraken2 | Bracken | Centrifuge | CLARK | CLARK-S | Metamaps | MEGAN | Minimap2 align | Minimap2 map | Ram |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **ONT1 (18)** | In | 1.77 ± 3.07 | 2.62 ± 3.04 | 1.82 ± 3.05 | 1.79 ± 3.09 | 2.51 ± 3.16 | 2.34 ± 3.04 | 2.29 ± 3.04 | **1.67 ± 3.08** | **1.67 ± 3.08** | 1.90 ± 3.02 |
| | Out | 21.1 | 20.5 | 21.8 | 18.2 | **4.93** | 16.7 | 20.3 | 21.9 | 21.6 | 17.57 |
| **ONT2 (8)** | In | 4.96 ± 5.67 | 6.98 ± 4.52 | 5.24 ± 4.93 | 4.99 ± 5.55 | 7.52 ± 4.84 | 5.19 ± 4.53 | 5.70 ± 5.30 | **4.29 ± 5.30** | 4.53 ± 5.42 | 4.68 ± 5.21 |
| | Out | 39.9 | 32.5 | 42.0 | 34.5 | **14.5** | 23.1 | 36.2 | 33.9 | 36.3 | 31.3 |
| **PB1 (8)** | In | 1.27 ± 0.95 | 4.24 ± 2.04 | 2.44 ± 2.54 | 1.83 ± 1.66 | 4.09 ± 2.57 | 0.55 ± 0.25 | 2.81 ± 1.81 | **0.43 ± 0.31** | **0.43 ± 0.36** | 0.60 ± 0.41 |
| | Out | 6.16 | 6.35 | 6.68 | 5.46 | **0.45** | 1.17 | 1.66 | 2.28 | 2.74 | 1.92 |
| **PB1+NEG (8)** | In | 2.25 ± 0.94 | 4.31 ± 2.00 | 2.60 ± 2.74 | 1.83 ± 1.66 | 4.09 ± 2.57 | 0.55 ± 0.26 | 3.82 ± 2.13 | **0.43 ± 0.31** | 1.76 ± 0.55 | 0.60 ± 0.41 |
| | Out | 18.0 | 8.15 | 20.8 | 5.50 | **0.45** | 1.20 | 2.01 | 2.28 | 14.0 | 1.92 |
| **PB2 (13)** | In | 1.68 ± 1.54 | 1.33 ± 0.71 | 2.37 ± 2.30 | 0.82 ± 1.44 | 1.77 ± 1.70 | 1.10 ± 1.79 | 1.17 ± 0.98 | **0.30 ± 0.49** | 3.13 ± 3.66 | **0.30 ± 0.48** |
| | Out | 21.8 | 5.22 | 30.7 | 5.59 | **0.18** | 11.5 | 2.30 | 3.90 | 40.7 | 2.60 |
| **PB3 (3)** | In | 13.9 ± 9.74 | 7.37 ± 6.12 | **4.16 ± 0.67** | 12.1 ± 5.49 | 10.1 ± 5.52 | 9.06 ± 3.47 | - | 9.89 ± 3.27 | 23.1 ± 19.2 | 23.4 ± 2.94 |
| | Out | 22.4 | 6.57 | 21.6 | 11.3 | **0.65** | 5.55 | - | 8.05 | 71.4 | 14.8 |
| **PB4 (46)** | In | 0.27 ± 0.46 | 0.75 ± 1.12 | 0.30 ± 0.59 | 0.27 ± 0.43 | 0.49 ± 0.83 | 0.26 ± 0.46 | 0.50 ± 0.98 | **0.21 ± 0.40** | 0.23 ± 0.42 | 0.24 ± 0.42 |
| | Out | 10.38 | 12.2 | 11.1 | 8.66 | 4.87 | 8.44 | **4.42** | 8.28 | 9.15 | 7.80 |
| **ONT Zymo (10)** | In | **1.17 ± 0.84** | 2.20 ± 1.68 | 2.04 ± 2.68 | 1.49 ± 1.31 | 1.90 ± 2.55 | 1.72 ± 1.68 | 2.59 ± 2.04 | 1.52 ± 1.27 | 1.57 ± 1.33 | 1.70 ± 1.46 |
| | Out | 7.11 | 5.05 | 6.11 | 4.56 | **0.16** | 0.34 | 1.31 | 1.32 | 1.22 | 1.59 |
| **PB ATCC (20)** | In | 1.20 ± 1.97 | **0.94 ± 1.38** | 1.28 ± 2.12 | 1.06 ± 1.88 | 1.05 ± 1.93 | 1.05 ± 1.88 | - | 1.05 ± 1.87 | 1.06 ± 1.87 | 1.05± 1.8 7 |
| | Out | 1.69 | 1.98 | 0.51 | 0.18 | **0.04** | 0.12 | - | 0.31 | 0.45 | 0.36 |
| **PB Zymo (17)** | In | 3.84 ± 5.10 | **3.79 ± 4.98** | 3.86 ± 5.13 | 3.88 ± 5.08 | 4.54 ± 5.46 | 4.00 ± 5.07 | - | 3.96 ± 5.07 | 3.90 ± 5.08 | 4.07 ± 5.09 |
| | Out | 44.8 | 41.5 | 45.4 | 43.7 | **16.0** | 40.6 | - | 41.7 | 43.6 | 40.4 |

Additionally, we analysed a cumulative abundance estimation error. We calculate it as a total sum of absolute values of differences between true and calculated abundance for each reported species independently, present or not present in the original sample. The main part of the tests was performed on a database constructed for each tool from the same set of sequences: NCBI-NR database with all bacterial and archaeal genomes, plus the human genome. We also tested the tools on a database without a human genome (containing only bacterial and archaeal genomes). The comparison of the abundance estimation error for both databases is given in Table 2.

Table 2 shows that MetaMaps and Minimap2 with alignment outperform other tools, followed by Ram. Kmer-based tools Kraken2, Centrifuge, CLARK and CLARK-S perform similarly, and their results are near to those achieved by Ram. Datasets PB2 and PB3 have a higher percentage of human reads (20% and 99%, respectively). Ram achieves the best results on the PB2 dataset and MetaMaps on PB3. Results show that having a host genetic material in a dataset significantly increases the abundance levels of taxa not present in the sample. When the high proportion of reads belongs to the host, most tools struggled even when the human genome was present in the database.

Comparing data from well-defined, accurately characterized mock communities ONT Zymo and PB ATCC (hifi reads) difference in abundance estimation accuracy of tools between datasets is not high. Unfortunately, we could not find ONT and PacBio data for the same mock community, so we cannot conclude about the influence of the sequencing technology on tools' performances. Results on PacBio Zymo Gut Microbiome Standard dataset (PB Zymo) were again worse for all tools.

**Table 2. Comparing abundance estimation error for the database with human genome and database without human genome.**

The table shows the total abundance estimation error for each dataset and tool and for two databases: database with the human genome and database without the human genome. The error is calculated by calculating the absolute value of the difference between abundance calculated for each tool and true abundance and summing it up across all organisms (in and out of sample). Each dataset name is followed by the percentage of human reads in that dataset in parentheses. Results for MEGAN are unavailable for datasets PB3, PB_atcc and PB_zymo.

| Dataset (% human) | Database | Kraken2 | Bracken | Centrifuge | CLARK | CLARK-S | Metamaps | MEGAN | Minimap2 align | Minimap2 map | Ram |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **ONT1 (0%)** | **human** | 52.9 | 67.6 | 54.6 | 50.3 | **50.0** | 58.9 | 61.5 | 52.0 | 51.7 | 51.7 |
| | **no human** | 53.1 | 68.0 | 54.7 | 50.3 | **50.1** | 58.9 | 61.5 | 52.0 | 51.7 | 51.7 |
| **ONT2 (5.78%)** | **human** | 79.8 | 88.3 | 84.0 | 74.5 | 74.7 | **64.6** | 81.8 | 68.2 | 72.5 | 68.7 |
| | **no human** | 83.7 | 88.2 | 90.6 | 76.0 | 74.3 | **64.7** | 84.2 | 72.3 | 75.9 | 69.6 |
| **PB1 (0%)** | **human** | 16.3 | 40.2 | 27.2 | 20.1 | 33.1 | **5.5** | 24.2 | 5.8 | 6.1 | 6.7 |
| | **no human** | 16.6 | 40.3 | 27.3 | 20.1 | 33.6 | **5.5** | 24.2 | 5.8 | 6.2 | 6.6 |
| **PB1+ NEG (0%)** | **human** | 36.0 | 42.6 | 41.6 | 20.1 | 33.1 | **5.6** | 32.6 | 5.8 | 28.1 | 6.7 |
| | **no human** | 39.4 | 43.0 | 45.7 | 20.1 | 33.6 | **5.6** | 32.6 | 5.8 | 31.1 | 6.6 |
| **PB2 (20%)** | **human** | 43.7 | 22.5 | 61.5 | 16.3 | 23.2 | 25.8 | 17.6 | 7.8 | 81.4 | **6.4** |
| | **no human** | 89.2 | 54.2 | 98.3 | 63.8 | 23.3 | 26.0 | 38.9 | 49.0 | 106.8 | **8.8** |
| **PB3 (99%)** | **human** | 50.1 | 21.3 | 104.8 | 35.5 | **20.8** | 23.7 | 63.7 | 37.8 | 117.7 | 61.6 |
| | **no human** | 145.7 | 145.1 | 145.7 | 145.7 | 140.1 | **96.0** | 145.7 | 145.5 | 145.5 | 140.3 |
| **PB4 (0%)** | **human** | 23.0 | 46.7 | 24.9 | 21.0 | 27.3 | 20.3 | 27.2 | **18.0** | 19.6 | 18.9 |
| | **no human** | 23.3 | 46.6 | 25.1 | 21.1 | 27.6 | 20.3 | 27.2 | **18.0** | 19.7 | 18.8 |
| **ONT zymo (0%)** | **human** | 18.8 | 27.1 | 26.5 | 19.5 | 19.2 | 17.5 | 27.2 | **16.5** | 16.9 | 18.5 |
| | **no human** | 18.8 | 27.1 | 26.6 | 19.5 | 19.2 | 17.5 | 27.2 | **16.5** | 16.9 | 17.4 |
| **PB atcc (0%)** | **human** | 25.8 | **20.8** | 26.1 | 21.4 | 21.1 | 21.1 | - | 21.4 | 21.6 | 21.5 |
| | **no human** | 25.8 | **20.8** | 26.1 | 21.4 | 21.1 | 21.1 | - | 21.4 | 21.6 | 21.5 |
| **PB zymo (0%)** | **human** | 110.1 | 106.0 | 111.0 | 109.7 | **93.1** | 108.6 | - | 109.1 | 109.9 | 109.6 |
| | **no human** | 110.2 | 106.0 | 111.0 | 109.8 | **93.1** | 108.6 | - | 109.1 | 109.9 | 109.6 |

## Organism detection

We also assessed how well tools identify organisms present in a sample. Table 3 shows how the number of correctly and incorrectly recognised organisms is related to a threshold - minimal number of assigned reads for reporting an organism as present in the sample. For most datasets, the number of incorrectly recognized species decreases while keeping the recognition of present organisms. However, if there are species with a very low number of reads, such as in datasets PB4 (lowest proportion of reads - 0.005 %) and ONT1 (lowest proportion of reads - 0.01 %), thresholds may also influence recognition of present organisms. In accordance with the results in previous sections, CLARK-S surpasses other tools for all datasets, followed by Ram. MetaMaps is the second best at PB3, confirming it as a good choice in the case of the large presence of host genetic material. However, Minimap2 and Ram are also close.

**Table 3. True positive and false positive organism detection.** The table shows true and false positive organism detections for three different thresholds: 1, 10 and 50. A threshold represents a number of reads that need to be assigned to that organism for it to be considered present in the sample. The data is presented as the number of false-positive detections (organisms incorrectly reported as present), followed by the number of true positive detections in parentheses (organisms correctly reported as present). Each dataset name is followed by the number of species in that dataset in parentheses. Results for MEGAN are unavailable for dataset PB3.

| **Dataset (TP)** | **Thres hold** | **Kraken2** | **Bracken** | **Centrifuge** | **CLARK** | **CLARK-S** | **Metamaps** | **MEGAN** | **Minimap2 align** | **Minimap2 map** | **Ram** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **ONT1 (18)** | **1** | 2162(15) | 135 (14) | 2365 (15) | 905 (15) | **51** (15) | 596 (15) | 547 (15) | 428 (15) | 592 (15) | 127 (15) |
| | **5** | 375 (14) | 135 (14) | 502 (14) | 223 (15) | **20** (15) | 78 (15) | 160 (15) | 132 (14) | 154 (15) | 54 (15) |
| | **50** | 33 (14) | 42 (14) | 42 (14) | 26 (15) | **3** (14) | 10 (15) | 18 (14) | 21 (14) | 21 (15) | 13 (15) |
| **ONT2 (8)** | **1** | 2033 (6) | 87 (6) | 2318 (6) | 596 (6) | **62** (6) | 323 (6) | 216 (6) | 170 (6) | 247 (6) | 102 (6) |
| | **5** | 172 (6) | 87 (6) | 215 (6) | 125 (6) | **17** (6) | 45 (6) | 69 (6) | 69 (6) | 79 (6) | 43 (6) |
| | **50** | 19 (6) | 26 (6) | 30 (6) | 15 (6) | **6** (6) | 8 (6) | 12 (6) | 12 (6) | 13 (6) | 9 (6) |
| **PB1 (8)** | **1** | 942 (8) | 73 (8) | 999 (8) | 590 (8) | **63** (8) | 111 (8) | 101 (8) | 91 (8) | 427 (8) | 83 (8) |
| | **5** | 157 (8) | 73 (8) | 116 (8) | 124 (8) | **13** (8) | 22 (8) | 34 (8) | 38 (8) | 50 (8) | 39 (8) |
| | **50** | 28 (8) | 31 (8) | 22 (8) | 23 (8) | **2** (8) | 4 (8) | 8 (8) | 10 (8) | 11 (8) | 13 (8) |
| **PB1+ NEG (8)** | **1** | 3035 (8) | 127 (8) | 2877 (8) | 594 (8) | **62** (8) | 177 (7) | 116 (8) | 91 (8) | 2467 (8) | 83 (8) |
| | **5** | 494 (8) | 127 (8) | 476 (8) | 124 (8) | **13** (8) | 23 (7) | 40 (8) | 38 (8) | 287 (8) | 39 (8) |
| | **50** | 28 (8) | 34 (8) | 26 (8) | 23 (8) | **2** (8) | 4 (7) | 10 (8) | 10 (8) | 13 (8) | 13 (8) |
| **PB2 (13)** | **1** | 3005(12) | 83 (12) | 3337 (12) | 448 (12) | **42** (12) | 119 (12) | 218 (12) | 108 (12) | 3556 (12) | 77 (12) |
| | **5** | 299 (12) | 83 (12) | 390 (12) | 68 (12) | **9** (12) | 16 (12) | 42 (12) | 32 (12) | 617 (12) | 30 (12) |
| | **50** | 15 (12) | 18 (12) | 19 (12) | 9 (12) | **1** (12) | 4 (11) | 5 (12) | 8 (12) | 25 (12) | 5 (12) |
| **PB3 (3)** | **1** | 72 (3) | 4 (3) | 107 (3) | 29 (3) | **2** (3) | 10 (3) | - | 15 (3) | 165 (3) | 19 (3) |
| | **5** | 10 (3) | 4 (3) | 10 (3) | 5 (3) | **0** (3) | 2 (3) | - | 5 (3) | 25 (3) | 5 (3) |
| | **50** | **0** (3) | **0** (3) | **0** (3) | **0** (3) | **0** (3) | 1 (3) | - | **0** (3) | 6 (3) | 1 (3) |
| **PB4 (46)** | **1** | 1603 (42) | 67 (40) | 1544 (41) | 516 (42) | **50** (40) | 227 (40) | 171 (41) | 163 (41) | 831 (42) | 146 (41) |
| | **5** | 128 (41) | 67 (40) | 105 (41) | 101 (40) | **15** (39) | 39 (39) | 54 (40) | 57 (41) | 73 (40) | 57 (40) |
| | **50** | 23 (35) | 27 (35) | 23 (34) | 14 (35) | **5** (33) | 13 (33) | 13 (34) | 23 (34) | 21 (34) | 19 (34) |

### Computational resource usage

Finally, we analysed running time and memory usage for evaluated tools. Results are presented in Table 4. As expected, kmer-based tools, apart from CLARK-S, dominate in the running time. For our test datasets, Centrifuge has the lowest running time for most datasets. However, in comparison with mappers such as Minimap2 and especially Ram, the difference between best kmer-based tools and mappers is below one order of magnitude. MetaMaps and MEGAN are much slower. Ram uses the least amount of RAM memory. Kraken2, Centrifuge, Minimap2 and MEGAN, for most datasets, use 2-3 times more memory. CLARK, CLARK-S and MetaMaps use between 10-15 times more.

Ram and Minimap2 execution times were additionally tested by mapping only one sequence to the whole database file. The execution time for both was around 1000 seconds, which suggests that the database parsing and indexing take about that much time. Both tools could have their execution time improved by storing and loading preprocessed database indexes to the disk.

For Bracken, we analysed the running time and memory consumption for the database building procedure because that procedure needs to be executed for every dataset independently since datasets have a different average read lengths, a parameter required by this procedure. The abundance estimation script executes almost instantaneously.

Additionally, we analysed the scalability of used tools on several different dataset sizes. The results are presented in Supplementary Table 1. Even for the largest datasets, Ram is still at most around 10x slower than Kraken2, the fastest kmer-based tool. Although Centrifuge is the fastest tool when analysing execution times presented in Table 4, Kraken2 is the tool that has the lowest execution times when tested on larger datasets. This happens because, for smaller datasets, index loading takes a great part of the execution time and Centrifuge has the smallest database index. On larger datasets, where the actual sequence classification takes a greater part of the execution time, Kraken2 outperforms other tools.

All resource usage measurements were performed on a machine with sufficient disk space, 775 GB RAM and 256 virtual CPUs. Measurements were performed using 12 threads. Between runs, we cleared RAM Memory Cache, file system buffer and swap space.

**Table 4. Resource usage.** The table shows running time (in seconds) and memory usage (in GB) for all tools and datasets.

**Execution time / s**

| Dataset / tool | Kraken2 | Bracken | Centrifuge | CLARK | CLARK-S | Metamaps | MEGAN | Minimap2 align | Minimap2 map | Ram |
|---|---|---|---|---|---|---|---|---|---|---|
| **ONT1** | 314 | 126166 | **275** | 974 | 3913 | 37839 | 67090 | 3145 | 1710 | 1482 |
| **ONT2** | 315 | 147207 | **291** | 954 | 3917 | 39844 | 75214 | 2874 | 1808 | 1509 |
| **PB1** | 312 | 137017 | **284** | 972 | 3942 | 54829 | 84852 | 3797 | 1890 | 1541 |
| **PB1+NEG** | 321 | 145769 | **296** | 968 | 4100 | 68941 | 157526 | 3778 | 1799 | 1600 |
| **PB2** | 326 | 191661 | **296** | 993 | 4114 | 58815 | 119849 | 2641 | 1647 | 1607 |
| **PB3** | 320 | 224048 | **309** | 979 | 4075 | 145416 | - | 1904 | 1541 | 1591 |
| **PB4** | 308 | 98826 | **267** | 963 | 3862 | 50180 | 58238 | 2843 | 1597 | 1511 |
| **ONT zymo** | 327 | 144387 | **305** | 979 | 4084 | 70622 | 160225 | 4220 | 2283 | 1697 |
| **PB atcc** | **317** | 68971 | 329 | 975 | 3957 | 76897 | - | 3044 | 1928 | 1303 |
| **PB zymo** | 317 | 179206 | **292** | 953 | 3996 | 63142 | - | 2364 | 1604 | 1272 |

**Memory / GB**

| Dataset / tool | Kraken2 | Bracken | Centrifuge | CLARK | CLARK-S | Metamaps | MEGAN | Minimap 2 align | Minimap2 map | Ram |
|---|---|---|---|---|---|---|---|---|---|---|
| **ONT1** | 43.04 | 45.31 | 37.00 | 119.56 | 271.24 | 205.67 | 26.87 | 39.82 | 34.18 | **14.05** |
| **ONT2** | 43.12 | 25.38 | 36.91 | 119.39 | 271.46 | 208.46 | 78.91 | 47.40 | 31.34 | **14.06** |
| **PB1** | 43.03 | 25.47 | 37.08 | 118.91 | 271.16 | 208.46 | 30.43 | 28.60 | 19.98 | **14.10** |
| **PB1+NEG** | 43.01 | 25.39 | 37.02 | 119.15 | 271.17 | 208.46 | 30.94 | 27.52 | 21.04 | **14.20** |
| **PB2** | 43.04 | 24.39 | 36.56 | 120.29 | 271.42 | 146.29 | 108.22 | 31.30 | 22.35 | **13.97** |
| **PB3** | 42.99 | 25.38 | 36.09 | 120.61 | 271.32 | 208.45 | - | 24.02 | 21.47 | **14.26** |
| **PB4** | 43.04 | 25.42 | 36.67 | 119.33 | 271.25 | 208.46 | 29.90 | 27.32 | 22.71 | **14.13** |
| **ONT zymo** | 43.02 | 25.41 | 37.06 | 120.08 | 271.14 | 208.46 | 42.57 | 41.56 | 38.23 | **14.27** |
| **PB atcc** | 43.01 | 24.37 | 36.00 | 120.05 | 271.25 | 208.46 | - | 28.77 | 19.77 | **9.07** |
| **PB zymo** | 42.98 | 24.42 | 35.94 | 120.06 | 271.22 | 208.41 | - | 26.19 | 21.00 | **9.15** |

## Discussion

The results show that long-read mapper Minimap2 (with enabled alignment) overperforms other tools in read accuracy on most datasets at both species and genus levels.

When comparing read accuracy on genus and species level, for some datasets (i.e. ONT1 and ONT2), the order of best-performing tools significantly differ. The reason is species in the sample missing from the database, but there are similar species in the database. While tools such as CLARK-S, Ram and MetaMaps tend to assign reads specifically to the original species, others, such as Minimap2 and kmer based tools, tend to assign reads to similar species if the original ones are not present in the database. Therefore, the former tools perform better at the species level, and the latter tools yield better results at the genus level. A useful upgrade to classification tools would be to provide some information about the confidence of whether the read belongs to a similar species or it doesn’t belong to any species in the database.

Instead of read counts for the calculation of abundances we used a measure which involves lengths of reads and genomes. The results on real datasets show that this measure achieves similar or more precise abundance calculations. Therefore, we recommend using this measure for abundance estimation.

Together with MetaMaps, Minimap2 with alignment exceeds other tools on abundance estimation, too. However, Minimap2 reports more false-positive organisms than some other tools, especially CLARK-S.

Ram mapper, which uses just a portion of minimizers used by original Minimap, performs slightly worse or like MetaMaps and better than MEGAN on both read accuracy and abundance estimation while having fewer falsely detected species. In addition, it is usually two orders of magnitude faster than these two tools and up to three times faster than Minimap2 with alignment. Finally, it and uses the least amount of memory among all tools. We deem that Ram might be a good compromise solution, especially since it can be run on a laptop even for the largest tested datasets (111 GB). It required less than 16 GB of RAM and finished in less than 4 hours using 12 threads. In addition, it shows in which direction new methods might be developed.

MetaMaps achieves very good results in abundance calculation when we consider all reported species. It is less prone to error than other tools in the case of a high presence of host reads. Its major drawback is its long execution time.

Kmer based tools such as Kraken2, Centrifuge and CLARK perform worse on read accuracy than mapping based tools, worse on abundance estimation for synthetic datasets, and report more false-positive species. On two real datasets, Kraken2 slightly surpasses other tools on abundance estimation of organisms present in the sample but still reports more false-positive organisms. On larger datasets, Kraken2 was the fastest. Using Bracken for the abundance calculation based on Kraken2 output achieved mixed results. On some datasets, such as those with the present human genome, it significantly improves Kraken2 results. On others, especially those sequenced by ONT, it performs worse. Bracken calculation of average read lengths substantially increases running time.

Kmer-based tools, apart from CLARK-S, are faster than mapping-based. Yet, modern mappers Minimap2 and especially Ram are only up to 10x slower on most of datasets. On many datasets kmer based tools were only up to three times faster than Ram while using more memory. However, we argue that due to their speed, kmer based tool can still be used in many applications, especially when the precision on genus level is good enough.

CLARK-S is an outlier among kmer based tools. It is worse than other tools in the accuracy estimation on both read and abundance levels for present organisms in the sample, not faster than modern mappers and uses more RAM than any other tool. However, it stands out in the organism's detection and performs well in abundance estimation when we include all reported organisms.

Comparing reads' length for correctly and incorrectly classified reads, we found that median read length for true positives is significantly higher than for false positives. Unfortunately, distributions of reads and the sparseness of particular species do not allow usage of only longer reads because it has a negative impact on species abundance calculation.

It is important to emphasize that our analysis of the PB Zymo dataset shows how the results are sensitive to all wet lab steps which precede sequencing.

Finally, this assessment shows that with long sequencing technologies, the boundary blurs between kmer-based and mapping based tools. Modern mappers use fewer kmers in the calculation of mapping candidate positions which makes them faster. We believe that with the further improvement in long-read sequencing technology, most methods will move to the detection of smaller numbers of kmers in combination with chaining matches. To reduce the number of false positives, they will probably need an additional postprocessing step using methods such as the EM algorithm. Finally, we believe there is probably space for the improvement in careful curation of existing databases with reference genomes.

## Methods

This chapter gives a description of used tools, how to test datasets were constructed, how the testing was performed, and testing metrics were calculated.

### Tools

Tested tools can be classified into:

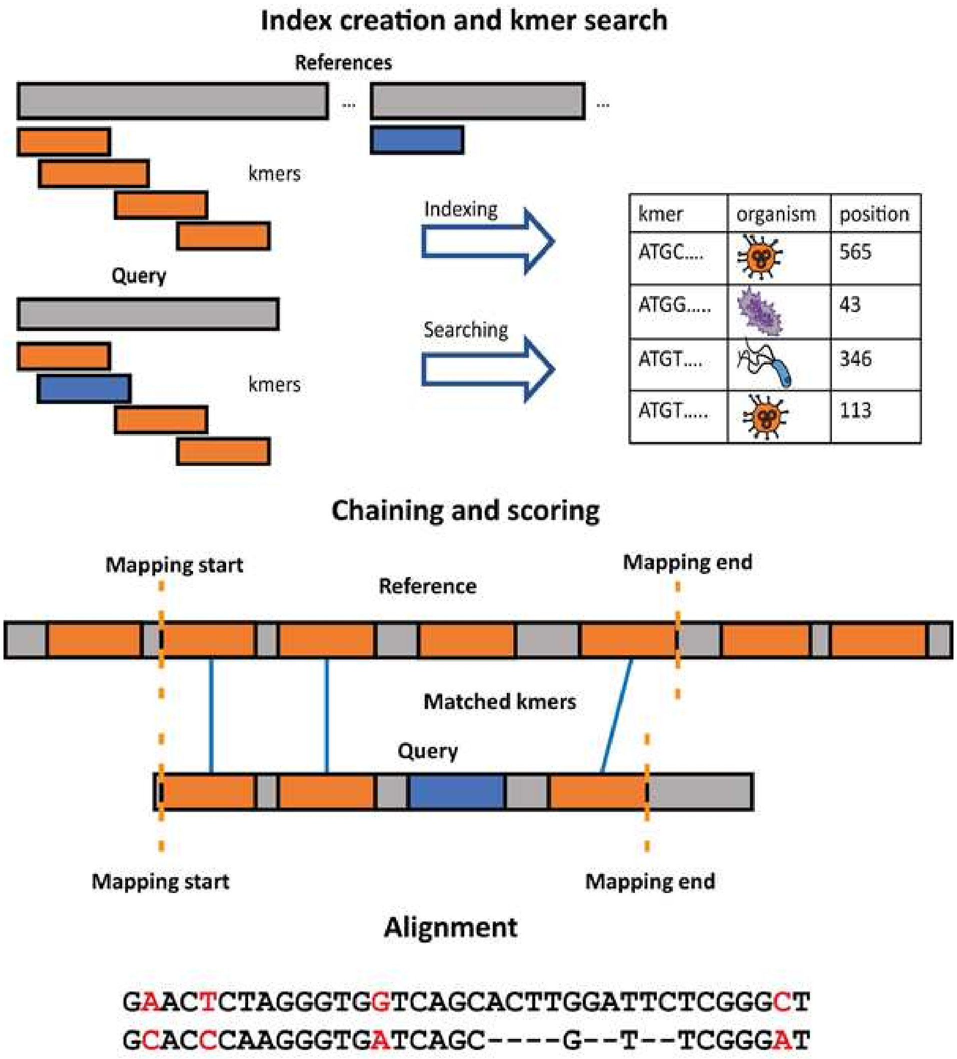
- k-mer based: Kraken2, Centrifuge, CLARK and CLARK-S
- mapping-based: MetaMaps, MEGAN-LR, Minimap2 and Ram

Since Kraken2 usually uses Bracken[16] for the calculation of abundances, we included it in the analysis.

Tools start with the initial assignment of reads to genomes using in advance prepared databases of known organisms. Once when all reads are assigned, various methods are used to fine-tune the classification using information from assigned reads and taxonomy trees. The most popular post-processing approaches are Expectation-Maximization (EM) estimation (MetaMaps, Centrifuge), Bayesian estimation (Bracken) and read assignment using the least common ancestor approach (MEGAN-LR, Kraken2).

The initial assignment of reads is based on aligning reads to a database of determined genomes. Aligning (Figure 3) might be divided into three steps: (1) Searching for exact or approximate matches of short substrings of length k (kmers) or longer in a previously prepared index which contains a list of kmers from genomes (2) Chaining kmer matches into a sequence, scoring the sequence, finding approximate positions of read in a genome (mapping), and choosing the best genome candidates (3) Alignment of a read and candidate genomes using exact dynamic programming algorithm. While kmer-based tools use only the first step, mapping-based tools use first and second or all three of them. Each additional step adds to accuracy but significantly increases the running time.

Usually, kmers are of a fixed size. The initial approach was to use all sliding windows of size k in a sequence. This might lead to high accuracy, but it is too slow. Therefore, modern tools usually use just a few discriminative kmers per genome, or they choose a lexicographically smallest kmer in a window of w consecutive kmers - minimizer[17].



| kmer | organism | position |
|---|---|---|
| ATGC.... | | 565 |
| ATGG..... | | 43 |
| ATGT.... | | 346 |
| ATGT..... | | 113 |

**Figure 3. Read alignment.** Read alignment consists of three steps (1) Indexing and kmer search, (2) Chaining and scoring (3) Alignment. Kmer-based tools use only the first step, and usually, they do not care about the position in the genome. Mapping based tools use the first and second steps, which increase accuracy but last much longer. The alignment step provides the exact alignment and the score but additionally increases the running time.

The output of various tools was processed to obtain read-level classifications and abundance of various species in a sample. We evaluated and analyzed the performance of all tools. Short descriptions and versions of each tool are available in Supplementary Materials 1. Specific parameters and scripts used to run each tool are given in Supplementary Materials 2.

While the results for MEGAN are not as good as for other mapping-based tools, it should be noted that they might be better when used with a protein database for which the MEGAN-LR pipeline was designed. Furthermore, we were unable to successfully run our version of the MEGAN-LR pipeline on the PB3 synthetic dataset and on PB_zymo and PB_atcc real datasets. In the case of the PB3 dataset, the mapping phase using the LAST aligner would go on for several days, and after that, the CPU and memory usage would drop down to almost zero, but the process would not complete. Output produced in that way was corrupted and could not be used for testing. After three trials, we decided to drop the results. In the case of PB_zymo and PB_atcc datasets, the LAST aligner produced a very large MAF file with correct alignments, which we could not convert to an alignment out file (.DAA). This resulted in no classified reads.

Since Minimap2 are Ram are not intended for metagenomic classifications and often prints several mapping results for a single sequence, the best classification for each sequence, for the *paf* output files, without the alignment, was determined with the following expression:

**2*(mapping_length * number_of_matches) / (mapping_length + number_of_mathces)**, where the *mapping_length* and *number_of_matches* are found in each row of the *paf* file. For the *sam* output files, with alignment, the best classification for each sequence was determined by the highest alignment score.

## Database

We assessed six metagenomic classification tools that were either newly developed or modified to work with long reads. In addition, we added two mappers for long reads. Each classification tool comes with a prebuilt default database and with instructions on how to build and use a custom database. To remove bias related to default databases, we built a database for each tool based on the same set of organisms. We used the NCBI-NR database with all bacterial and archaeal genomes, plus the human genome. Genome sequences were downloaded (April 5th 2020) along with the taxonomy files nodes.dmp and names.dmp. This allows the tools to be tested independent of the content of their default database. The details of how each database index was created for every tool is presented in Supplementary Materials 3.

### Test datasets

To have realistic sequencing datasets while retaining control on our mock communities' exact content and building the ground truth, we constructed in silico datasets by mixing real reads from isolated, sequenced species. Data was downloaded from multiple sources (details in Supplementary Table 5), including the European Nucleotide Archive (ENA[18]) and the National Center for Biotechnology Information (NCBI[19]). This in-silico approach provides a ground truth and great flexibility to create diverse datasets while offering real reads with their natural errors and length variance. Most of the datasets contain around 100,000 reads to allow all tools to classify them within a few days. We varied the proportion of species, some with even distributions, some with decreasing ratios with as little as five reads for one species (PB4 dataset). Seven test datasets were synthesized with the following composition: two ONT, four PacBio and one negative dataset containing PacBio and randomized reads.

- **ONT1**: 18 bacterial species with a percentage of reads varying from 18% down to 0.01%.
- **ONT2**: Human (about 4000 reads) + 7 bacteria, 10,000 reads each.
- **PB1**: 10 bacteria, 10% each (including two strains of *E. coli*).

- **PB2**: 20% human reads, 20% fruit fly (*D. melanogaster*), 10% archaea (*M. labreanum* Z), and ten bacteria, varying from 10% to ~1%.
- **PB3**: 99% human reads, plus two bacteria: 0.9% E. coli and 0.1% *S. aureus*.
- **PB4**: 46 bacterial species with the percentage of reads varying from 10% to 0.005%.
- **PB1+NEG**: It contains all the reads from PB1 datasets with additional 20000 “randomized” reads that should not be assigned to any organism. Randomized reads were obtained by shuffling the human genome (GRCH38.p7) using *esl-shuffle* script from the hmmer3 [20] package (version 3.3.2) as described by Lindgreen et al.[21].

All datasets that do not contain human reads are mapped to the human reference with minimap2 to check if there are contaminations with human reads in any of the datasets. No sequences that belong to non-human species mapped to the human genome with a significant quality.

In addition to synthetic datasets, the tools were also tested on three real datasets obtained by sequencing mock metagenomic communities. The results reported by the tested tools were used to calculate abundances and compared to standard specifications obtained from manufacturer pages.

- **ONT_zymo**: obtained by GridION sequencing of a Zymo Community Standard, consists of 8 bacteria and 2 yeasts with the expected abundance varying from 0.37% to 21.6% (downloaded from LomanLabs https://lomanlab.github.io/mockcommunity/).
- **PB_atcc**: obtained by PacBio HiFi sequencing of an ATCC MSA-1003 standard, consists of 20 different bacterial species with the expected abundance varying from 0.02% to 18% (download from NCBI archive, SRA run identifier: SRR11606871).
- **PB_zymo**: obtained py PacBio HiFi sequencing of a Zymo D6331 Gut Microbiome Standard, consists of 16 bacteria and one yeast, with the expected abundance varying

from 0.0001% to about 20% (download from NCBI archive, SRA run identifier: SRR13128014). However, for this dataset, the results obtained by all tools differed significantly from the specification.

## Testing procedures

The tools' output was processed to obtain percentages of DNA reads and species' abundances in the sample. We evaluated the correctness of DNA read classification at species and genus level, i.e., only classifications that were assigned to a tax id which belongs to the species or lower-level were used in the species-level analysis; and only classifications assigned to the genus or lower levels were used in the genus-level analysis. Outputs of the tools, which contain classification of reads to taxons, were processed. Taxonomic ids and ranks were extracted from the nodes.dmp file downloaded from the NCBI website.

### Read-level classification

To evaluate the quality of read level classification, we calculate four basic values first:

- True positives (TP): the number of reads that were classified to a correct species.
- False positives (FP): the number of reads that were classified as an incorrect species.
- True negatives (TN): the number of reads that remained unclassified and belonged to an organism not present in the database.
- False negatives (FN): the number of reads that remained unclassified but belonged to an organism present in the database.

These four values are then used to calculate more complex and useful evaluation metrics. The first metric used is classification accuracy – the percentage of reads that were correctly classified.

**Accuracy = (TP+TN) / (TP+FP+TN+FN)**

Since accuracy does not consider the proportion of each species in the dataset, we also used the F1 score. F1 score is calculated from precision and recall values:

**Precision (PR) = TP / (TP+FP)**

**Recall (RC) = TP / (TP+FN)**

**F1 = 2* PR*RC / (PR+RC)**

To make the F1 measure less biased towards larger classes, we calculate the F1 score for each class (organism in the sample) separately and average them (F1 macro average). Because the F1 score is zero for classes not in the database (as the number of true positives is zero), those classes are omitted from the calculation.

### Abundance

Abundance represents the percentage of genomes of a specific taxon in the sample. Abundances calculated by benchmarked tools significantly differ due to differences in definitions and calculations. Furthermore, most of them does not consider the genome sizes and differences in the distributions of reads lengths among species. Therefore, we calculate the abundance of a species as the sum of the lengths of assigned reads divided by its average genome length from the database. Obtained values are normalized in a manner that the total sum of abundances is 1. Genome lengths that were used were obtained from the NCBI database. For abundance estimation, we tested species or lower-level classification only and considered all classifications to higher-level taxa as incorrect. It is important to note that Bracken produces only read counts assigned to species. To compare it with other tools, abundances - the percentage of genomes of species in the sample, were calculated by normalising read counts with the average genome length of the species to which corresponding read counts were assigned.

## References

1. Quince, C., Walker, A. W., Simpson, J. T. & Loman, N. J. Shotgun metagenomics, from sampling to analysis. *Nature* (2017).
2. McFall-Ngai, M. *et al.* Animals in a bacterial world, a new imperative for the life sciences. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 3229–3236 (2013).
3. Hamady, M. & Knight, R. Microbial community profiling for human microbiome projects: Tools, techniques, and challenges. *Genome Res.* **19**, 1141–1152 (2009).
4. Bertrand, D. *et al.* Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes. *Nat. Biotechnol.* **37**, 937–944 (2019).
5. Nicholls, S. M., Quick, J. C., Tang, S. & Loman, N. J. Ultra-deep, long-read nanopore sequencing of mock microbial community standards. *Gigascience* **8**, (2019).
6. Pearman, W. S., Freed, N. E. & Silander, O. K. Testing the advantages and disadvantages of short- and long- read eukaryotic metagenomics using simulated reads. *BMC Bioinformatics* **21**, 220 (2020).
7. Leidenfrost, R. M., Pöther, D.-C., Jäckel, U. & Wünschiers, R. Benchmarking the MinION: Evaluating long reads for microbial profiling. *Sci. Rep.* **10**, 5125 (2020).
8. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biol.* **20**, 257 (2019).
9. Kim, D., Song, L., Breitwieser, F. P. & Salzberg, S. L. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res.* **26**, 1721–1729 (2016).
10. Ounit, R., Wanamaker, S., Close, T. J. & Lonardi, S. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics* **16**, 236 (2015).
11. Ounit, R. & Lonardi, S. Higher classification sensitivity of short metagenomic reads with CLARK-S. *Bioinformatics* **32**, 3823–3825 (2016).
12. Dilthey, A. T., Jain, C., Koren, S. & Phillippy, A. M. Strain-level metagenomic

assignment and compositional estimation for long reads with MetaMaps. *Nat. Commun.* **10**, 3066 (2019).

13. Huson, D. H. *et al.* MEGAN-LR: new algorithms allow accurate binning and easy interactive exploration of metagenomic long reads and contigs. *Biol. Direct* **13**, 6 (2018).
14. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
15. Vaser, R. & Šikić, M. Time- and memory-efficient genome assembly with Raven. *Nature Computational Science* **1**, 332–336 (2021).
16. Lu, J., Breitwieser, F. P., Thielen, P. & Salzberg, S. L. Bracken: estimating species abundance in metagenomics data. *PeerJ Comput. Sci.* **3**, e104 (2017).
17. Roberts, M., Hayes, W., Hunt, B. R., Mount, S. M. & Yorke, J. A. Reducing storage requirements for biological sequence comparison. *Bioinformatics* **20**, 3363–3369 (2004).
18. Leinonen, R. *et al.* The European Nucleotide Archive. *Nucleic Acids Res.* **39**, D28–31 (2011).
19. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **39**, D38–51 (2011).
20. Eddy, S. R. Accelerated Profile HMM Searches. *PLoS Comput. Biol.* **7**, e1002195 (2011).
21. Lindgreen, S., Adair, K. L. & Gardner, P. P. An evaluation of the accuracy and speed of metagenome analysis tools. *Sci. Rep.* **6**, 19233 (2016).

## Acknowledgements

This work has been supported by the National Medical Research Council, Singapore, under the grant MOH-000649-01 (Rapid diagnostic of infectious diseases based on nanopore sequencing and AI methods), by the European Union through the European Regional

Development Fund, under the grant KK.01.1.1.01.0009 (DATACROSS) and by the Croatian Science Foundation, under the grant HRZZ-IP-2018-01-5886 (SIGMA)

Conflict of Interest: none declared.