TITLE: Benchmarking metagenomic classification tools for long-read sequencing data

AUTHORS:

1. Josip Marić[1*]
2. Krešimir Križanović[1*]
3. Sylvain Riondet[2,3]
4. Niranjan Nagarajan[2,3#]
5. 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 this work
[#] Corresponding authors

ABSTRACT

In recent years, both long-read sequencing and metagenomic analysis have been significantly advanced. Although long-read sequencing technologies have been primarily used for de novo genome assembly, they are rapidly maturing for widespread use in other applications. In particular, long reads could potentially lead to more precise taxonomic identification, which has sparked an interest in using them for metagenomic analysis.

Here we present a benchmark of several state-of-the-art tools for metagenomic taxonomic classification, tested on in-silico datasets constructed using real long reads from isolate sequencing. We compare tools that were either newly developed or modified to work with long reads, including k-mer based tools Kraken2, Centrifuge and CLARK, and mapping-based tools MetaMaps and MEGAN-LR. The test datasets were constructed with varying numbers of bacterial and eukaryotic genomes to simulate different real-life metagenomic applications. The tools were tested to detect species accurately and precisely estimate species abundances in the samples.

Our analysis shows that all tested classifiers provide useful results, and the composition of the used database strongly influences the performance. Using the same database, tested tools achieve comparable results except for MetaMaps, which slightly outperform others in most metrics, but it is significantly slower than k-mer based tools.

We deem there is significant room for improvement for all tested tools, especially in lowering the number of false-positive detections.

1. INTRODUCTION

The advent of high-throughput sequencing has enabled a detailed analysis of microbial communities and their hosts through metagenomics [1], [2]. An essential component of metagenomic sequencing workflows is computational methods for recognising organisms present in a sample. Most current methods are tailored to work with short, accurate reads from second-generation sequencing technologies. However, due to an increase in their accuracy and throughput, long-read sequencing technologies are gaining popularity. Currently, the most popular long-read sequencing technologies are those by Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT).

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. However, whole-metagenome shotgun sequencing covers all genetic 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 sequenced from a sample. If most of the present genomes 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 do a performance analysis of methods based on the comparison of long-read sequencing data with reference databases. Although there are several benchmarking studies on long reads [5]–[7], our analysis includes both PacBio and ONT sequencing technologies incorporates a detailed evaluation of the influence of the database on analysis results, and assesses tradeoffs between running time and memory requirements in typical use cases with real sequencing data.

2. METHODS

We constructed several test datasets based on reads from PacBio and ONT sequencing for this benchmark.

The tools were tested using two types of databases:

- The default database for each tool is downloaded from the tool's webpage or described in the corresponding publication.

- Manually prepared database with bacteria, archaea, and the human genome sequences from RefSeq database.

2.1. TESTED TOOLS AND SETTINGS

For this benchmark, we use Kraken2, Centrifuge, CLARK, MetaMaps, and MEGAN-LR. The tools can be classified into those that use detailed read-level mappings and those that approximate read mapping using shorter k-mers.

- k-mer based: Kraken2, Centrifuge, CLARK
- mapping-based: MetaMaps, MEGAN-LR

The output of various tools was processed to obtain the presence/absence and relative abundance of various species in a sample. These two measures were used to evaluate and analyse the performances of all tools.

Specific parameters and scripts used to run each tool are given in the Supplementary materials.

2.1.1. Kraken2

Kraken is a tool that assigns taxonomic labels to DNA sequences using exact alignments of k-mers [8]. In this study, we used Kraken2 [9], an improved version of Kraken, which reduces memory usage. The Kraken2 report file contains summarised hierarchical percentages of taxa abundances in the dataset, while the classification file contains the unique classification of each read to an assigned taxa id. We used the former to obtain Kraken2 calculated abundances and the latter to calculate the percentages of reads assigned to the corresponding taxa id.

2.1.2. Centrifuge

Centrifuge is a metagenomic classification tool that uses an indexing scheme based on the Burrows-Wheeler transform (BWT) and the FM index [10].

2.1.3. CLARK

CLARK [11] builds an extensive index of discriminative k-mers for all target sequences, and an object (sequencing read) is assigned to a target sequence with which it shares the highest number of k-mers. CLARK does not provide a default database. To be comparable to default databases for other tools containing bacterial and archaeal genomes, we built a bacteria and archaea database from the Refseq database.

2.1.4. MetaMaps

MetaMaps [12] performs classification in two steps. In the first step, possible mapping locations are generated using an approximate mapping algorithm based on minimizers. In the second step, mapping locations are scored probabilistically, and a total sample composition is estimated

using an EM algorithm. MetaMaps was run with a minimum read length set to 500 (option -m) instead of the default 1000, due to shorter reads in the test datasets.
MetaMaps provides a default database (miniSeq+H) which contains microbial genomes and the human genome.

2.1.5. MEGAN-LR

MEGAN6 is a comprehensive toolbox for interactively analysing microbiome data [13]. It uses an LCA-based algorithm for taxonomic binning and an interval-tree based algorithm for functional binning. In 2018, it was upgraded to work with long reads - Megan LR [14].

2.2. TEST DATASETS

To have realistic sequencing datasets, while retaining control on our mock communities' exact content and building a ground truth, we constructed *in silico* datasets by mixing real reads from isolated sequenced species. Data was downloaded from multiple sources (details in Supplementary materials), including the European Nucleotide Archive (ENA (Harrison et al. 2020)) and the National Center for Biotechnology Information (NCBI [15]). This *in-silico* method 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 ratio, with as little as five reads for one species (PB4).

As current classifiers emphasise microbial communities and human DNA, these account for most of our datasets, with only one set containing other eukaryotes. We have seen an increase in eukaryotic species in genomic databases, but most classifiers do not scale yet with the amount of data to index, which results in either very high RAM needs and/or slow processing of reads. Speed and memory improvements will be a trend for future developments.

ONT datasets:

- 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.

PacBio datasets:

- 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: 50 bacterial species with the percentage of reads varying from 10% to 0.005%.

Use cases

We created datasets to highlight some common use cases in microbiology analysis. There are two main goals for classification algorithms: identify species and evaluate their abundances. Reaching these objectives highly depends on the community's content and the actual number of reads for each species. Therefore, we designed 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 is one with predominantly human reads and two low abundance bacterial species, reflecting what one might see in an infection setting.

2.3. TESTING PROCEDURES

The tools' output was processed to obtain percentages of DNA reads and species' abundances in the sample.

We evaluated the correctness of the classification of DNA reads at the species level, i.e., only the classifications of the reads that were assigned to the tax id, which belongs to the species level, were used in the analysis. Outputs of the tools, which contain the classification of reads to taxons, were processed. Taxonomic ids and ranks were extracted from the nodes.dmp file downloaded from the NCBI website. The percentage of classified reads was compared with the actual percentage of reads in each dataset.

The other measure which we calculated is abundance. The abundance represents the percentages of genomes of a specific taxon in the sample. Abundances calculated by benchmarked tools significantly differ due to differences in definitions and calculations. Therefore, we standardised these for calculating the abundance of a species in the sample. A comparison of the abundances calculated by the tools can be found in the Supplementary materials.

We define the abundance of one species in the sample as the sum of the lengths of reads classified to that species tax id divided by the average size of genomes for that species. These calculated values are then weighted to sum to 1 to represent the relative abundance of a species in the sample. The average genome length of the species was calculated from genomes in the Refseq database. Relative abundance is calculated by normalising all absolute abundances to one. This approach does not tackle the issue with the variance of classification depending on the read's location but improves a raw read count.

To compare the tools' ability to correctly determine sample composition, precisions-recall plots, based on read-level classifications, were drawn. Each point in the plot is drawn for a certain

threshold represented by a number of reads that need to be assigned to an organism for it to be considered detected. The thresholds chosen were the ones for which precision and recall values change, namely the numbers of reads assigned to a certain organism by a classifier. Classification results are sorted descending according to the number of reads, and each read count is used as a threshold. For each threshold, precision and recall for organism detection are calculated. Organisms with the number of reads higher or equal to the threshold are considered detected, while all others are considered as not detected. This is used to calculate true and false, positive and negative organism detections and to plot a point on the precision-recall curve.

3. RESULTS AND DISCUSSION

The classification results are influenced by the dataset, the algorithm applied, and the database. Five metagenomic classification tools (Kraken2, Centrifuge, CLARK, MetaMaps, and MEGAN-LR) were tested using six datasets (2 ONT and 4 PacBio) and two different databases. Each tool comes with a prebuilt default database and with instructions on how to build and use a custom database. First, the tools were tested using their default databases, as this is how many users would use them. Since the databases were built at different times before their publication, with various purposes in mind (some tools emphasise their small memory needs, some highlight their extensiveness), organisms present in one database might not be present in the others. The results on default databases differ significantly among tested tools.

To remove the bias related to default databases, we built an additional 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. This allows the tools to be tested independent of the content of their default database. For easier reference in the following text, the databases will be referred to as:

- defaultDB – default database of each tool
- customDB - NCBI NR database containing bacteria, archaea, and human genomes

3.1. The choice of database

Results using a default database (defaultDB) for each tool show a high variance (both as a generic term and as a well-defined mathematical concept). This is visible in Figures 1 and 2. The best performing tool changes across datasets, and the overall winner is not evident. This is expected since each tool comes with its default database containing different organisms. If an organism present in a tested dataset is not present in a database, a tool will not classify its reads correctly.

Figure 1 shows classification accuracy (calculated on the read-by-read basis) for defaultDB and customDB. We can see that on the default database, the results vary much more than on the custom database. For each tool, there is a dataset that the tool performs best (or close to best)

on. However, on the custom database, the results are much more uniform with MetaMaps having the highest accuracy on the majority of datasets.

It should be noted here that for the MEGAN-LR pipeline, the default database contained protein sequences (NCBI NR database), while the custom database contained nucleotide sequences (NCBI NT database). In both cases, reads were mapped to the database using the LAST aligner, and the results were processed in the same way. However, using a nucleotide database does produce more alignments and consequently requires more resources in the first part of the pipeline. This was done to avoid excessive customisation of each tool, and each tool was used with default parameters as much as possible.

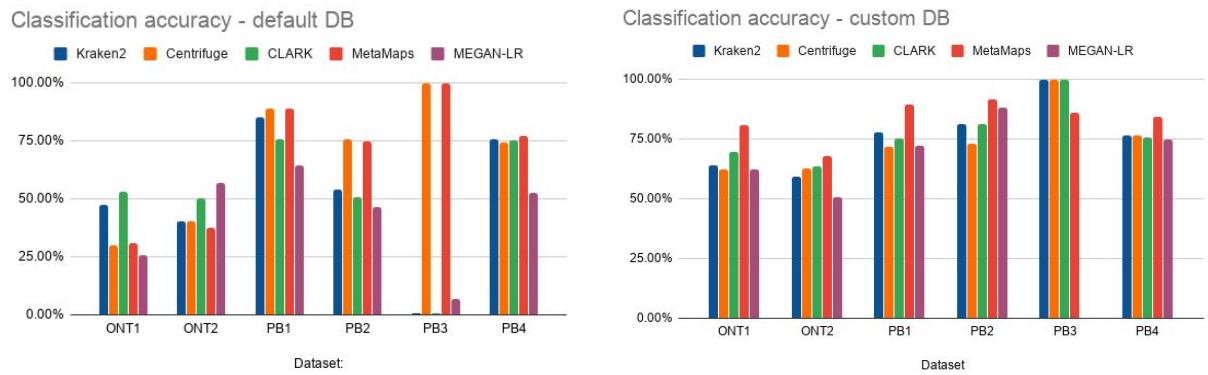

Figure 1. Classification accuracy comparison: defaultDB Vs. customDB

Higher variability on the default database is more apparent in Figure 2, which compares classification accuracy variance for the default and the custom database. One can see that the variance is much higher with the default database on all datasets, even up to several times. The difference is extreme on the PB3 dataset because it contains many human reads, and some default databases contain the human genome, while others do not, resulting in very high variance. Because the results of MEGAN-LR are unavailable for the PB3 dataset with the custom database, they were not used for variance calculation.

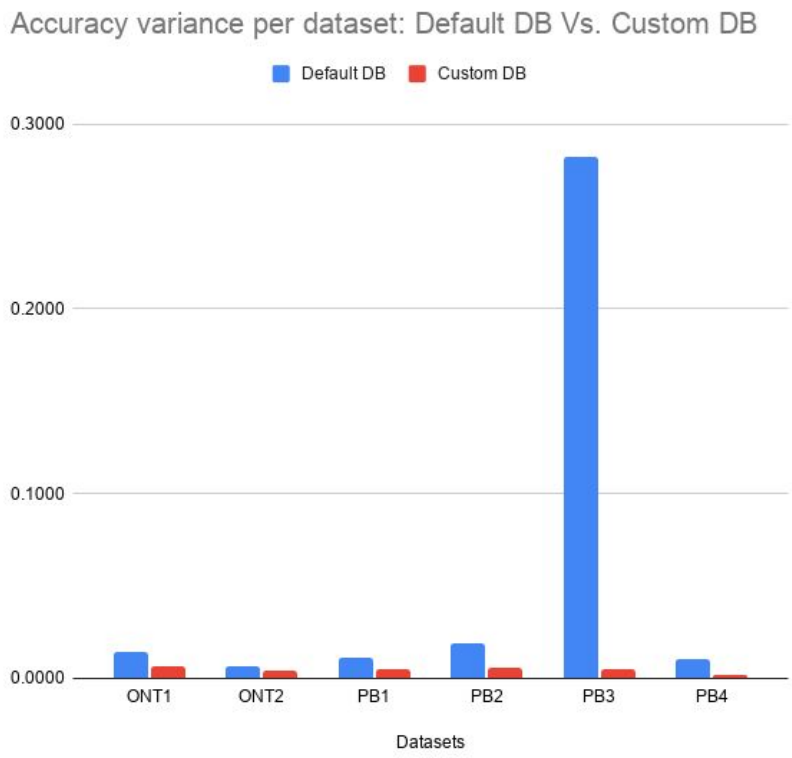

Figure 2. Classification accuracy variance per dataset

These results, while expected, still emphasise the importance of choosing the right database. It is advisable to build a custom database, regardless of the tool used unless you are confident that the default database contains all necessary organisms for the analysis.
Following the obtained results, we argue that the composition of the database has a more significant impact on classification results than the choice of classification tool.

3.2. Classifier comparison

For a more detailed comparison, we use only the custom build database, containing the same set of organisms for all tools.

The tools were tested in four different areas:

- read level classification – how accurately can they classify each read
- organism detection – how accurately can they detect organisms in a sample
- abundance estimation – how well can they be used to estimate the abundance of each organism in the sample
- resource usage – how much memory and computing power do they require

3.2.1 Read level classification

Read level classification obtained from tested tools were compared to ground truth for species-level classification only. Assignments to higher-level taxa were not considered correct.

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 belong 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)

The results for classification accuracy are shown in Figure 3. It should be noted that we were unable to successfully run MEGAN-LR pipeline on PB3 dataset. The mapping phase using 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.

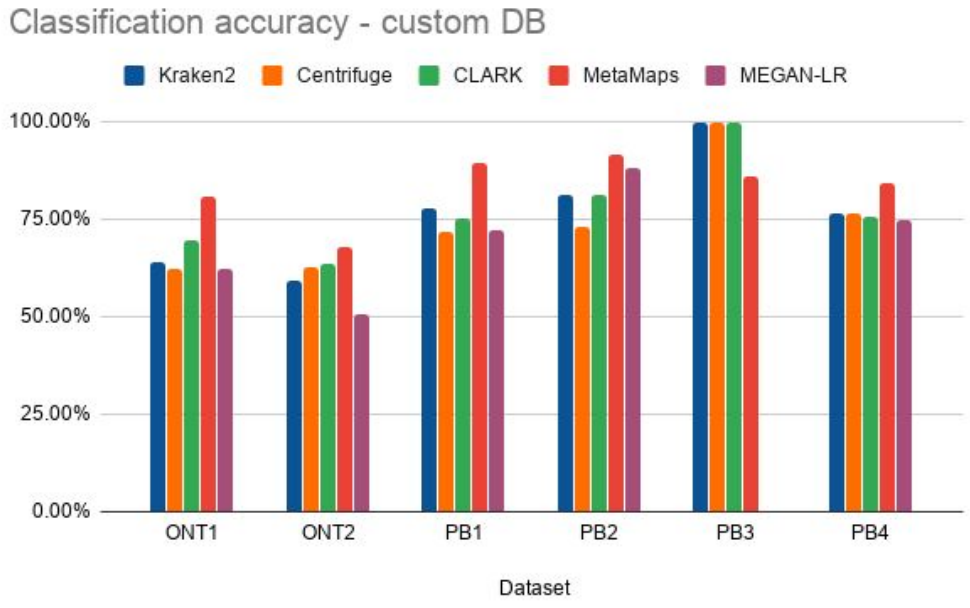

Figure 3. Classification accuracy when using the same database (MEGAN-LR did complete on PB3)

Classification accuracy is not entirely appropriate for evaluating imbalanced datasets (as is the case for most of our datasets). Therefore, as a second metric, we 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. The results for the F1 macro average are shown in Figure 4.

We can see that for both accuracy and F1 score, MetaMaps gives the best results on the majority of datasets. It has the highest accuracy and largest F1 macro average on four datasets (ONT1, PB1, PB2, and PB4). Although MetaMaps also has the highest accuracy on the ONT2 dataset, it does not have the highest F1 macro avg. Additionally, on ONT2 and PB3 datasets, MetaMaps' results are very close to the best ones.

It seems that MetaMaps, while having the best read level classification accuracy overall, has some trouble with the human reads, resulting in lower accuracy on datasets containing human reads.

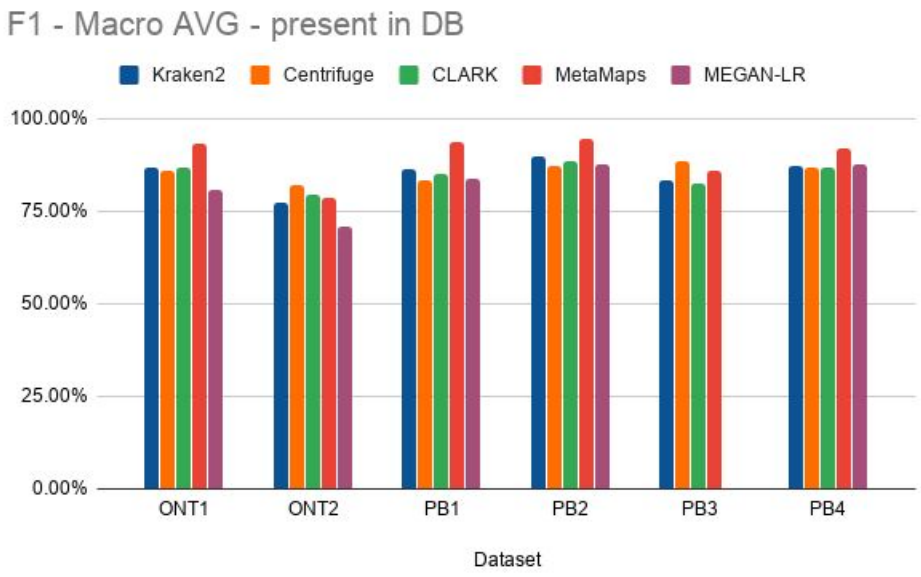

Figure 4. The macro average for F1 score when using the same database

The analysis thus far examined species level classification only. However, two datasets (ONT1 and ONT2) contain several species from the *Vibrio* genus, some of which are present in the database and some are not. Although evaluating classification results spanning multiple taxonomic levels is outside the scope of this paper, we decided to investigate this specific case in more detail, generating a second set of results with genus-level evaluation for the *Vibrio* genus and species level evaluation for other species.

Figure 5 shows classification accuracy and F1 macro average comparison between full species level classification and a specific case with the genus level for the *Vibrio* genus for the ONT1 dataset. We can see that the F1 macro average increases for all classifiers for a roughly equal amount. Classification accuracy also increases for all classifiers except for MetaMaps, for which it even decreases very slightly. This is due to the reads that MetaMaps reports as unclassified, which were previously considered as true negatives. Still, in this analysis, they are regarded as false negatives, since in this case, we consider it correct for them to be classified within the *Vibrio* genus. The new results show that MetaMaps is still clearly the best classifier looking at the F1 macro average. Kraken2 came off second best.

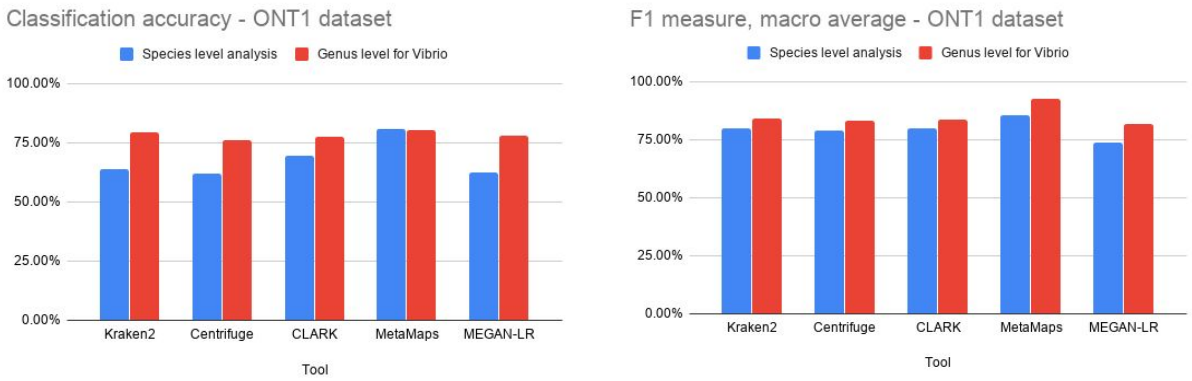

Figure 5. Comparing classification results on ONT1 dataset for full species level classification and the special case with genus level for Vibrio genus

3.2.2. Organism detection

In the second part, tools are compared to correctly identified organisms present in a sample . When using the same database for all tools, tools are generally able to detect all microorganisms in the sample present in the database. The exception is the PB4 dataset, which contains the largest number of organisms, some of which are represented with a tiny number of reads. However, classifiers differ significantly in the number of false positives, both false-positive read classifications, and false-positive organism detection. The number of falsely positive read classifications is shown in Figure 6.

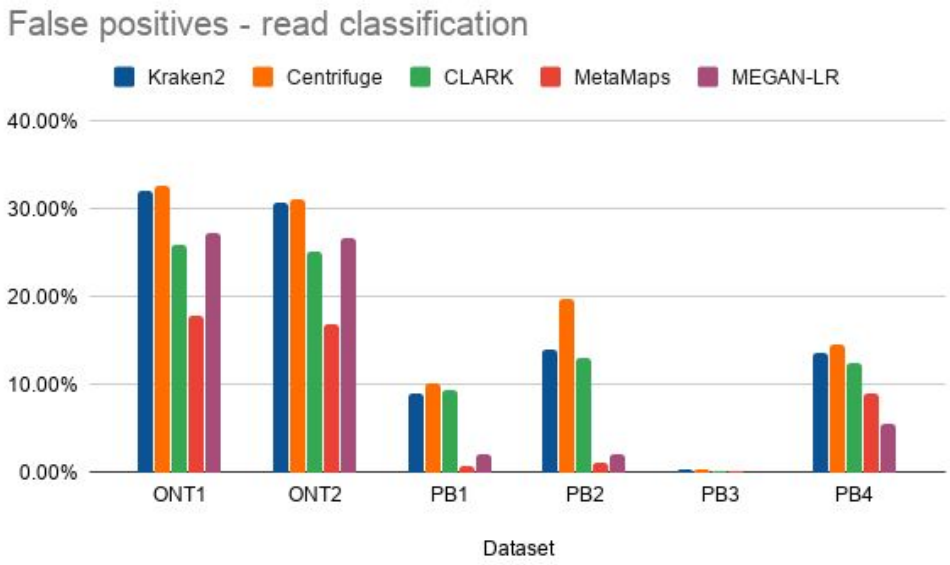

Figure 6. Percentage of false-positive read classifications

As expected, mapping-based tools (MetaMaps and MEGAN-LR) have fewer falsely positive read classifications than k-mer based tools. MetaMaps, in general, has the fewest number of false positives except on the PB4 dataset, where MEGAN-LR has the fewest. However, MetaMaps classifies more reads than MEGAN-LR on all datasets. Looking over all datasets, the number of false positives is higher for ONT datasets than for PacBio datasets, possibly due to a higher error rate.

To further investigate tools performance, precision-recall plots for organism detection are drawn. Figure 7 shows the precision-recall curves for all datasets. On PB1, PB2, and PB3 datasets, all tools can achieve completely accurate organism detection or close to it (if the right threshold is chosen). MetaMaps and Kraken2 have the best results on those datasets. On ONT datasets, the results are significantly worse due to a higher error rate and several species from the *Vibrio* genus not present in the database. However, the best tool is not clear. PB4 dataset is the most complex one, with 46 different species and per-species read count ranging from 5 to 15000. Looking at the Area Under the Precision-Recall Curve (AUPRC), the best results are probably obtained by MetaMaps and MEGAN-LR. However, regardless of the threshold chosen, many less abundant organisms remain undetected with a significant number of falsely positive detections. The AUPRC provides an excellent overview of the sensitivity (recall) and precision of each classifier for each dataset.

Since some of our datasets explicitly contain species not found in the database on those datasets (both ONT datasets), classifiers *can not* find all species which results in recall values below one. It reflects the actual and typical case of mock communities with species whose reference genome hasn't been determined yet. Nevertheless, this often leads to higher false-positive rates, as we can see in the ONT1, for example. Since two most abundant species

are not in the database they get classified under a wrong taxonomy for all classifiers (except MetaMaps), which sets the precision to zero on the left-hand size of the graph before correctly predicted species at lower abundance increase the precision.

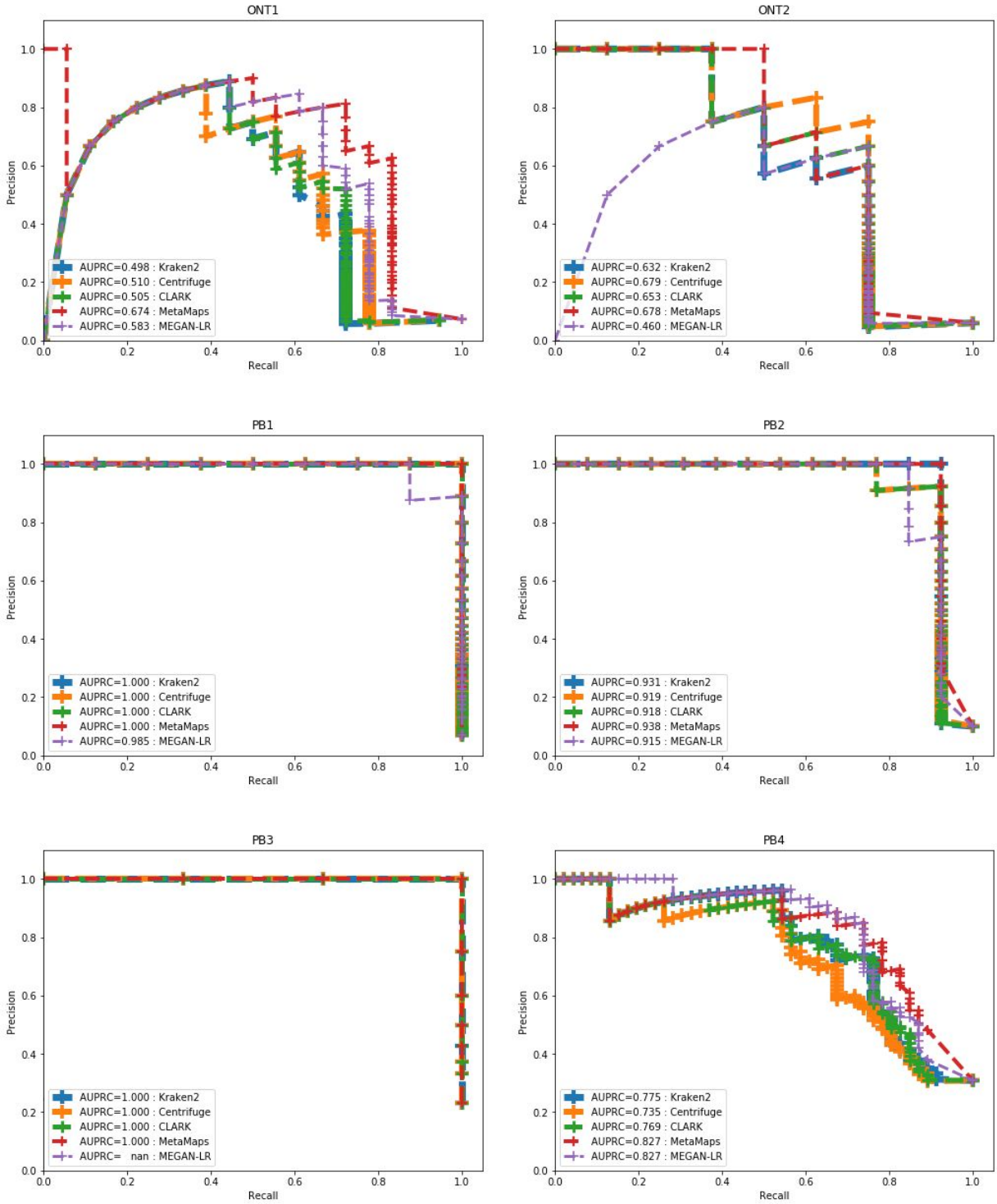

Figure 7. Precision-recall curves for identifying organisms in a sample. The precision and the recall are evaluated for each result of the classifiers, from the highest abundance species to the lowest.

AUPRC curves for PB1, PB2 and PB3 datasets look almost perfect. For these datasets, classifiers manage to identify the majority of species without flaws (100% precision), then output many false positives, which drops the precision. Sometimes the classifier fails to identify the few last species as for PB2.

This behaviour is common and often patched by cutting the results at a particular threshold (usually 1% or 0.1%). It, therefore, produces an AUPRC of 1.0, hiding the false positives. This solution works when all species are around equal abundance, or at least when the lowest abundant species to find is still above the “noise”, making it easy to filter it out. Nevertheless, this method is empiric, and the threshold between true positives and false positives is not obvious without ground truth, making it hard to determine.

Results become more interesting when species are at varying abundance (ONT1 and PB4), with curves telling more about the actual performance of the classifiers. MetaMaps stands out on both graphs, managing to keep the precision high while detecting all the species in the sample before failing with the species not in its index. The other methods yield more false positives before finding all species of the dataset. On ONT1, all methods except MetaMaps predict false positives before finding true positives, resulting in a curve starting at 0 precision, before catching up with the next species.

Looking at precision-recall plots, the overall effectiveness of a classifier can be determined by measuring the area under the curve (AUPRC). We can say that overall, Metamaps has the largest AUPRC, which is best visible on datasets ONT1 and PB4.

None of the tested tools can reliably detect organisms present in a sample. However, due to the lower number of false-positive read classifications and somewhat more accurate organism detection, alignment-based tools perform better than k-mer based ones, particularly MetaMaps.

3.2.3. Abundance estimation

The main problem with abundance estimation is that MEGAN-LR does not report abundances. All other tools except MetaMaps report read percentages assigned to a specific organism as its abundance (with an occasional difference less than 0.01%, possibly due to rounding the results). MetaMaps reports values that are close to read percentages but not exactly equal. A more detailed analysis of abundances reported by tools is given in the Supplementary materials.

In microbiology, the abundance of a species is defined as the ratio of cells in the community. However, most tools take the raw read count, which does not take into account that larger genomes will yield more reads for the same number of cells, or that reads from widely shared genomic segments lead to misclassification or no classification at all. There is currently a gap

between the biology definition and the output of the classifiers, which may lead to erroneous conclusions.

To make a meaningful comparison, we used read-level classification produced by each tested tool and calculated abundances for each reported species using the same algorithm.

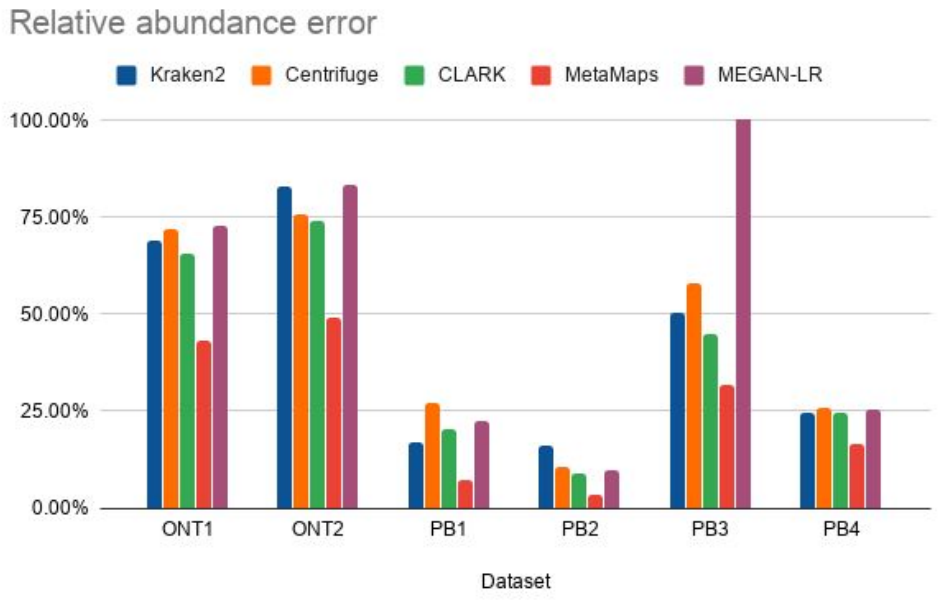

Figure 8. Relative abundance error

The abundance calculated for each classifier is then compared to the ground truth. Abundance error is calculated by summing up absolute errors across all species (both in the dataset and those that a classifier reported as present). Figure 8 shows an abundance's error comparison for all classifiers. It should be noted that the abundance error can go over 1 in case if a large number of reads belonging to a large genome are incorrectly classified to one or smaller genomes. This can reduce the large genome abundance by a small amount but increase (incorrect) abundance of small genomes by a more considerable amount.

The results show that Metamaps has the lowest abundance estimation error. On the PB3 dataset, MEGAN-LR has a very high abundance estimation error due to assigning a lot of human reads to other organisms.

3.2.4. Some additional testing

As described in the paper [14], the MEGAN-LR pipeline uses a protein database for classification. Nucleotide sequencing is mapped to a protein database using the LAST aligner. Our approach to the tests was to use all of the tools in the same conditions, out of the box, as much as possible. Therefore, in our test using the custom database, instead of protein, we used the nucleotide database even for the MEGAN-LR pipeline.

Since we tried to use the tools with as little customisation as possible, building a specific protein database for MEGAN-LR that would be the equivalent of the used custom database was outside the scope of our research. Although MEGAN-LR default mode assumes a protein database such as NR, the results (Figure 9) show that using the custom nucleotide database improved MEGAN-LR results on all datasets apart from ONT2.

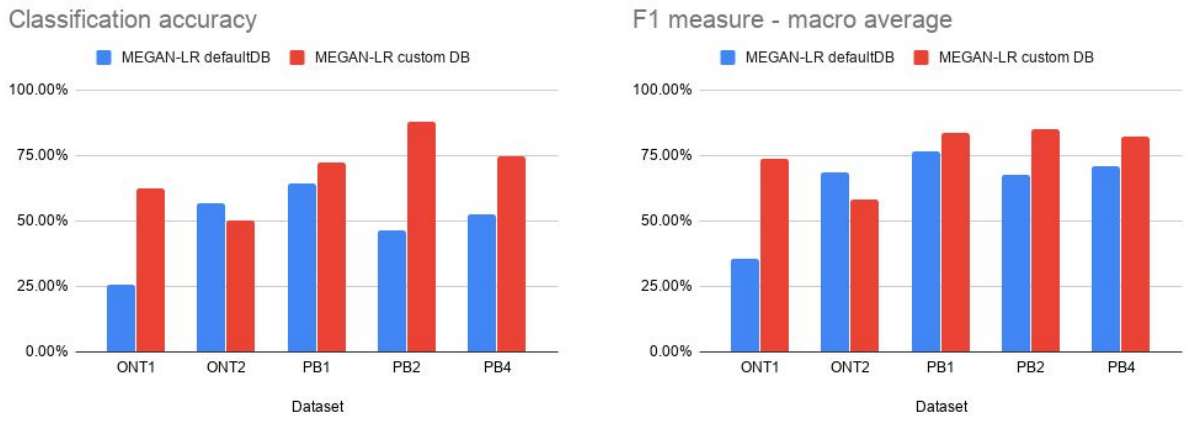

Figure 9. Comparison of MEGAN-LR results on default and custom database

Aside from the default database and the custom database used for the benchmark, we also tested all of the tools on another custom-built database containing only archaea and bacteria genome (without the human genome). Comparing the results on custom-built databases with and without the human genome can give us some additional insight into the influence of the database composition on the classification results. The differences between the results are shown below in tables 1 and 2. The values in the tables represent the difference in results on two databases, with the positive values meaning that the database with the human genome is providing better results. Negative values are highlighted in green. We can see that adding the human genome to the database can, in some cases, result in poorer classification.

If we consider classification accuracy shown in Table 1, even on datasets containing human reads, in some cases adding the human genome to the database can result in lower accuracy. This is somewhat misleading since if the database does not contain the human genome, human reads are considered correct if unclassified and counted as true negatives. If the database includes the human genome, human reads are considered correct only if they are classified as human and counted as true positives. If a classification tool is not very good at mapping human reads to the human genome, many will remain unclassified when the human genome is added to the database, resulting in lower classification accuracy. This is most apparent on the ONT1 dataset, for which adding the human genome to the database lowers classification accuracy for all tools except Centrifuge. We can speculate that due to the higher error rate of ONT reads (compared to PacBio), all tools except Centrifuge are unable to correctly classify a large enough number of human reads, thus lowering the classification accuracy.

Table 1 - Accuracy difference between database with the human genome and database without human genome

Dataset:	Contains human reads	Kraken2	Centrifuge	CLARK	MetaMaps	MEGAN-LR
ONT1	No	-0.50%	3.98%	-0.56%	-0.07%	0.01%
ONT2	Yes	-2.41%	11.34%	-2.00%	-4.82%	-4.48%
PB1	No	-0.59%	12.83%	-0.75%	0.00%	-0.01%
PB2	Yes	15.20%	13.04%	12.84%	-2.95%	6.92%
PB3	Yes	86.51%	86.05%	80.12%	-13.59%	-
PB4	No	-0.08%	7.88%	-0.42%	0.01%	-0.02%

Instead of looking directly at classification accuracy, somebody might better observe the effects of adding the human genome to the database through the number of true positive classifications. Table 2 displays the difference in the percentage of true positive classifications between the databases with and without the human genome. Here we can see that if a dataset contains human reads, adding the human genome to the database will increase the true positives count. However, if a dataset does not have human reads, adding the human genome to the database will in some cases slightly decrease the number of true positives for Kraken2, CLARK, and MEGAN-LR, meaning that some reads belonging to other organisms that were previously correctly classified, are now incorrectly classified as human reads. This leads us to conclude that having too many organisms in a database will negatively impact classification results.

Table 2 - Percentage of true positive classifications for each read, the difference between database with the human genome and database without human genome

Dataset:	Contains human reads	Kraken2	Centrifuge	CLARK	MetaMaps	MEGAN-LR
ONT1	No	0.01%	5.69%	-0.18%	0.00%	0.00%
ONT2	Yes	1.16%	15.43%	2.36%	0.80%	0.91%
PB1	No	-0.58%	12.82%	-0.75%	0.00%	-0.02%
PB2	Yes	19.74%	23.40%	19.79%	17.08%	19.95%
PB3	Yes	98.90%	99.04%	98.91%	85.09%	-
PB4	No	-0.05%	7.98%	-0.41%	0.01%	-0.03%

It is interesting to note that Centrifuge does not decrease its classification quality (neither accuracy nor true positives count) with the addition of the human genome, indicating that its algorithm is very precise when mapping human reads.

3.2. Resource usage

All resource usage measurements were performed on an AWS machine with sufficient disk space, with 256GB RAM and 32 virtual CPUs. R5 type memory-optimized machines were used. Measurements were performed using 12 threads. The memory and running times are given in Tables 3 and 4. Since we were unable to run MEGAN-LR on the PB3 dataset, the results are not present.

Table 3 - Running time in seconds

	ONT1	ONT2	PB1	PB2	PB3	PB4
Kraken2	61.481	61.322	62.47	73.065	70.56	58.836
Centrifuge	83.933	103.267	85.383	91.963	92.06	72.122
CLARK	834.377	823.544	708.854	631.863	575.539	525.574
MetaMaps	32446.051	37884.055	42326.089	42055.800	77472.582	37955.443
MEGAN-LR	67089.676	75213.996	84851.512	119848.799	-	58237.704

Table 4 - Memory usage in MB

	ONT1	ONT2	PB1	PB2	PB3	PB4
Kraken2	43650	43750	43612	43632	43600	43622
Centrifuge	33101	33068	33232	32762	32258	32810
CLARK	118662	118489	118769	119399	119716	118440
MetaMaps	146424	146596	146185	146469	146421	146159
MEGAN-LR	26899	78931	30443	108194	-	30027

As expected, we can see that mapping-based tools require much more time than k-mer based ones, by 2-3 orders of magnitude, with the MEGAN-LR being slower of the two. Also, among the k–mer based tools, CLARK is significantly slower than Centrifuge and Kraken2, with Kraken2 being the fastest.

Regarding memory, MetaMaps requires the most RAM, while Centrifuge, on average, requires the least. All of the tools except MEGAN-LR use roughly the same amount of memory for each dataset: MEGAN-LR varies from 27 GB to 108 GB.

Aside from RAM and CPU usage, MEGAN-LR protocol also requires quite large amounts of disk space. Disk usage for the MEGAN-LR pipeline is given in the supplementary materials.

CONCLUSION

This benchmark represents a comprehensive test of currently available metagenomic classification tools with support for long-read sequencing technologies. The tools were tested on several datasets of varying complexity, constructed from isolate sequencing with both PacBio and ONT technologies. The tools were tested on their ability to classify reads, detect organisms present in a sample, and on how usable they are for abundance estimation.

The results lead us to draw several conclusions.

Firstly, it is crucial to carefully prepare the database. Genomes not present in the database will not be detected. Also, usage of a very large database might lead to worse results, as was shown in the comparison of results on databases with and without the human genome. All tools, apart from Centrifuge, tend to worsen their performance in the case when the human genome was present in the database but not in the dataset

Secondly, all tools are usable and can detect all or most of the organisms present in a sample. However, all tools produce a significant number of false positives, which significantly diminishes their usefulness. Among the tested tools, MetaMaps showed the best results, having the best classification accuracy and F1 score and having the lowest abundance estimation error. However, MetaMaps performance decreased when the human genome was present in datasets.

As expected, k-mer based tools are much faster and require fewer resources than alignment-based tools. MetaMaps required the most memory and was second slowest (after MEGAN-LR). Both MetaMaps and MEGAN-LR were two to three orders of magnitude slower than Kraken2 and Centrifuge, making them very impractical to use in many situations.

We argue that there is room for improvement of metagenomic long-read classification tools. MetaMaps shows the best overall results, but requires a lot of computational resources. Kraken2 and Centrifuge require the least resources, but also produce somewhat worse results with a lot of false positives. The choice of the right tool will depend on the size and complexity of the testing sample and available resources.

Acknowledgements

This research was funded by A*STAR Singapore, 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).

REFERENCES

[1] C. Quince, A. W. Walker, J. T. Simpson, and N. J. Loman, “Shotgun metagenomics, from sampling to analysis,” *Nature*, 2017, [Online]. Available: https://www.nature.com/nbt/journal/v35/n9/abs/nbt.3935.html.

[2] M. McFall-Ngai *et al.*, “Animals in a bacterial world, a new imperative for the life sciences,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 9, pp. 3229–3236, Feb. 2013.

[3] M. Hamady and R. Knight, “Microbial community profiling for human microbiome projects: Tools, techniques, and challenges,” *Genome Res.*, vol. 19, no. 7, pp. 1141–1152, Jul. 2009.

[4] D. Bertrand *et al.*, “Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes,” *Nat. Biotechnol.*, vol. 37, no. 8, pp. 937–944, Aug. 2019.

[5] S. M. Nicholls, J. C. Quick, S. Tang, and N. J. Loman, “Ultra-deep, long-read nanopore sequencing of mock microbial community standards,” *Gigascience*, vol. 8, no. 5, May 2019, doi: 10.1093/gigascience/giz043.

[6] W. S. Pearman, N. E. Freed, and O. K. Silander, “Testing the advantages and disadvantages of short- and long- read eukaryotic metagenomics using simulated reads,” *BMC Bioinformatics*, vol. 21, no. 1, p. 220, May 2020.

[7] R. M. Leidenfrost, D.-C. Pöther, U. Jäckel, and R. Wünschiers, “Benchmarking the MinION: Evaluating long reads for microbial profiling,” *Sci. Rep.*, vol. 10, no. 1, p. 5125, Mar. 2020.

[8] D. E. Wood and S. L. Salzberg, “Kraken: ultrafast metagenomic sequence classification using exact alignments,” *Genome Biol.*, vol. 15, no. 3, p. R46, 2014.

[9] D. E. Wood, J. Lu, and B. Langmead, “Improved metagenomic analysis with Kraken 2,” *Genome Biol.*, vol. 20, no. 1, p. 257, Nov. 2019.

[10] D. Kim, L. Song, F. P. Breitwieser, and S. L. Salzberg, “Centrifuge: rapid and sensitive classification of metagenomic sequences,” *Genome Res.*, vol. 26, no. 12, pp. 1721–1729, Dec. 2016.

[11] R. Ounit, S. Wanamaker, T. J. Close, and S. Lonardi, “CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers,” *BMC Genomics*, vol. 16, p. 236, Mar. 2015.

[12] A. T. Dilthey, C. Jain, S. Koren, and A. M. Phillippy, “Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps,” *Nat. Commun.*, vol. 10, no. 1, p. 3066, Jul. 2019.

[13] D. H. Huson *et al.*, “MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data,” *PLoS Comput. Biol.*, vol. 12, no. 6, p.

e1004957, Jun. 2016.
[14] D. H. Huson *et al.*, “MEGAN-LR: new algorithms allow accurate binning and easy interactive exploration of metagenomic long reads and contigs,” *Biol. Direct*, vol. 13, no. 1, p. 6, Apr. 2018.
[15] E. W. Sayers *et al.*, “Database resources of the National Center for Biotechnology Information,” *Nucleic Acids Res.*, Oct. 2020, doi: 10.1093/nar/gkaa892.

SUPPLEMENTARY MATERIALS

Tool usage

The commands used to classify the datasets for each tool are listed here.

Kraken

Kraken was run with the default settings.

```
> kraken2 --report <report_file> --db <database_path> <dataset_path> >
<classification_file>
```

Previously Kraken's database index was built with the following commands:

```
> kraken2-build --download-taxonomy --db <database_path>
> kraken2-build --download-library bacteria --db <database_path>
> kraken2-build --download-library archaea --db <database_path>
> kraken2-build --add-to-library human_reference.fa --db <database_path>
> kraken2-build --build --db <database_path>
```

The resulting Kraken's database index takes 42 GB of disk space.

For the default Kraken's database a standard database was used, built with the following command:

```
> kraken2-build --standard --db <database_path>
```

Centrifuge

Centrifuge was run with default settings, but with the parameter k=1 to set the tool to classify each read to only one taxon. The command used was:

```
> centrifuge -x <path_to_database> -k 1 -f <path_to_dataset> -S <classification_file>
--report-file <report_file>
```

By running the additional command:

```
> centrifuge-kreport -x <path_to_database> <classification_file> > <report_file>
```

where the *classification_file* is the output file from the previous command, we get the *report_file* that contains abundances of taxa in the dataset in the Kraken2 format.

Centrifuge's database index was built with the previous commands:

```
> centrifuge-download -o taxonomy taxonomy
> centrifuge-download -o library -m -d "archaea,bacteria" refseq > seqid2taxid.map
> centrifuge-download -o library -d "vertebrate_mammalian" -a "Chromosome" -t 9606 -c
'reference genome' >> seqid2taxid.map
> cat library/*/*.fna > input-sequences.fna
> centrifuge-build -p 4 --conversion-table seqid2taxid.map --taxonomy-tree
taxonomy/nodes.dmp --name-table taxonomy/names.dmp input-sequences.fna <database_path>
```

Centrifuge's index takes around 35GB of space. The default Centrifuge's database index was built with the following command:

```
> make p_compressed+h+v
```

MetaMaps

MetaMaps was run with default settings. The custom database with bacteria, archaea and the human genome was built with the following commands:

```
> metamaps mapDirectly --all -r <path_to_database> -q <path_to_dataset> -o
<classification_file_prefix>
> metamaps classify --mappings <classification_file_prefix> --DB <path_to_database>
```

First command produces mappings of reads to the reference genomes from the database. The second command then calculates the final classification of reads to taxa id's.

The following commands were used to build the database index from the RefSeq database of archaea, bacteria and human genome.

```
> perl downloadRefSeq.pl --seqencesOutDirectory download/refseq --taxonomyOutDirectory
download/taxonomy
> perl annotateRefSeqSequencesWithUniqueTaxonIDs.pl --refSeqDirectory download/refseq
--taxonomyInDirectory download/taxonomy --taxonomyOutDirectory
download/taxonomy_uniqueIDs
> perl util/addTaxonIDToFasta.pl --inputFA hg38.primary.fna --outputFA
hg38.primary.fna.with9606 --taxonID 9606
> perl buildDB.pl --DB <path_to_database> --FASTAs
download/refseq,hg38.primary.fna.with9606 --taxonomy download/taxonomy_uniqueIDs
```

The result of database creation, among some smaller files is the fasta file named DB.fa which for the archaea, bacteria and human genome takes 66 GB of disc space.
MetaMaps' default database was found on the tool's Github page and was downloaded from this link: https://www.dropbox.com/s/6p9o42yufx2vkae/miniSeq%2BH.tar.gz?dl=0

CLARK

CLARK was run with the default settings. The databases for classification are chosen with the following command:

```
> set_targets.sh <path_to_database> bacteria human
```

With this command, complete genomes from RefSeq of both bacteria and archaea as well as the human genome are integrated into the database index. The classification is done with the following command:

```
> classify_metagenome.sh -O <path_to_dataset> -R <classification_file>
```

If it has not been previously created, this command will build the database index which for archaea, bacteria and human genome takes around 125 GB of disk space.

MEGAN-LR

The MEGAN-LR pipeline uses the LAST aligner to align long reads to a protein database, a script to convert alignments to the .DAA format and the MEGAN application to analyze the data. Megan6 community edition version 6_18_5 with Last aligner version 2.27.1 was used.
MEGAN was used with the NR protein database, but also with the NT nucleotide database.
Reads were first mapped to the reference database using the LAST aligner (using option -F15 for mapping to proteins). The .MAF file produced was converted to .DAA format using the maf2daa tool. A .DAA file was prepared for import to MEGAN, using the daa-meganizer tool with the option *-lg* for long reads. Finally, read counts per species were extracted in CSV format using MEGAN command line mode with the following script:

```
> open file='/mnt/data/megan-test/test_dataset_NR.daa';
> update;
> select nodes=all;
> export what=CSV format=readName_to_taxonId separator=tab counts=assigned
file='/home/kkrizanovic/metabenchmark/results/Mock_100000-bacteria-l1000-q10_2BacArHum
-RN2TAX.csv';
> quit;
```

Regarding the custom database creation, the MEGAN-LR pipeline uses the LAST aligner for mapping, and constructing a custom database is actually creating a LAST index on the corresponding set of reference genomes. The instructions used were obtained from the LAST tutorial page: http://last.cbrc.jp/doc/last-tutorial.html.

For default database containing protein sequences LAST index was created using the following instruction:

```
> lastdb -p -cR01 lastnt NT.fa
```

For custom database containing nucleotide sequences, LAST index was created using the following instruction:

```
> lastdb -cR01 lastcustom NR.fa
```

It should be noted that creating a last index took several 1-2 days on the server we used. Also, when indexing nucleotide sequences (NCBI bacteria, archaea, and human genomes), the index created was quite large, taking around 500GB of disk space.

Abundance estimation (reported by tools)

In microbiology, the abundance of a species is defined as the ratio of cells in the community. However, none of the tested tools calculates the abundance by this definition. Most tools simply

take the raw read count, MetaMaps reports somewhat modified read counts with the results still closer to read counts than to correct abundances, while MEGAN-LR doesn't report abundances at all.

We can see an example of the results in Figure 10. The figure shows the results on the PB1 dataset, displaying first our ground truth for abundance estimation (named "True abundance"), and then, side by side, abundance reported by a tool and read count percentages reported by that tool (e.g. Kraken2 displays abundances reported by Kraken2 tool, while Kraken2-RC represents read count percentages reported by also Kraken2). MEGAN-LR was left out since it does not report abundances.

We can see that all abundances reported by tools are very different from our abundance calculation. Also, we can see that for Kraken2, Centrifuge, and CLARK, abundances and read count percentages are entirely equal, while for MetaMaps, they are close with slight differences.

Similar results are obtained for all test datasets.

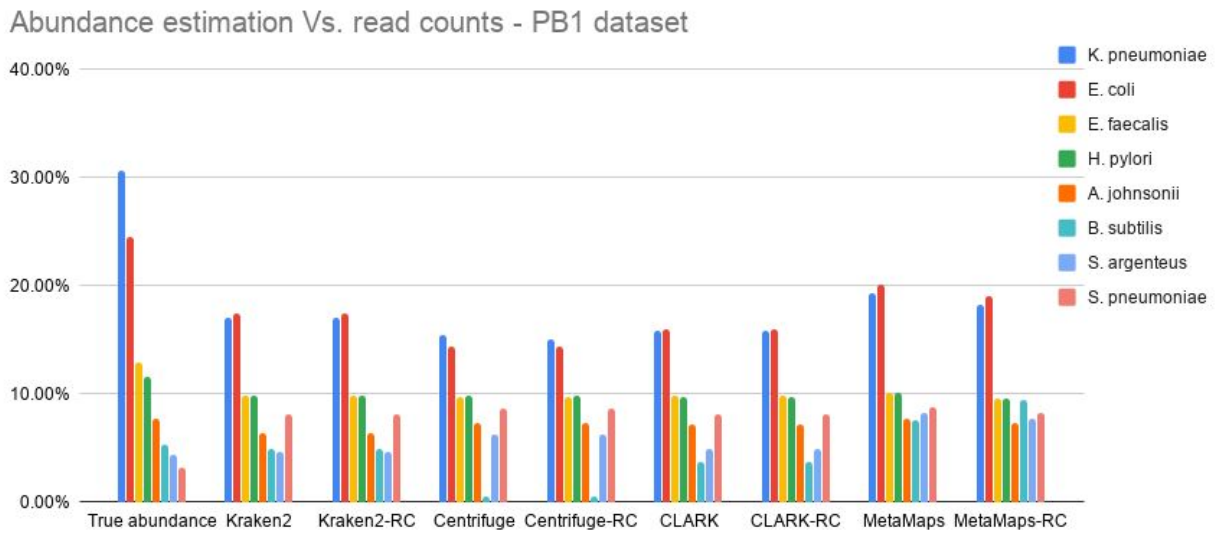

Figure 10. Comparing abundance estimation reported by tools to read counts assigned to each organism on PB1 dataset

MEGAN-LR disk usage

Table 5 shows disk usage for MEGAN-LR. The first step in the pipeline is using the LAST aligner to align reads to a reference database. The LAST aligner produces output in MAF format, which takes up a lot of space. MAF output is then converted into a much smaller DAA format using a MEGAN-LR conversion script. The actual size of the intermediate MAF file represents the maximum disk usage for the MEGAN-LR pipeline. While the output of the LAST aligner can be redirected (piped) to the MEGAN conversion script, the script also creates

temporary files on the disk which are, in our experience, not much smaller than the MAF file itself (more the 50% of the MAF file on all test datasets).

Using the default reference database, which is the NCBI-NR protein database, generates much smaller MAF files, which might fit into available memory and thus allow using the conversion script without any temporary disk files. However, when using a nucleotide database, the MAF files are much larger, as can be seen in the table. While single entries in the table might seem large, but manageable if multiple datasets are analysed, they can quickly add up, and the total disk space used needs to be taken into account.

Table 5 - MEGAN-LR - disk usage (MAF size)

	ONT1	ONT2	PB1	PB2	PB3	PB4
human DB	335 GB	524 GB	781 GB	618 GB	-	400 GB
default DB	118 GB	59 GB	97 GB	100 GB	158 GB	68 GB

Data used for dataset construction

Data was downloaded from NCBI SRA (Sequence Read Archive), Sanger Institute pages (NCTC 3000) and PacBio GitHub DevNet pages (Microbial multiplexing dataset and Human Genome in a Bottle dataset). Isolates are identified by their identifier in the corresponding database, given below:

Identifier	Source	Species
Nanopore data		
SRR5891470	NCBI	Acinetobacter baumannii
SRR6255797	NCBI	Amphiprion ocellaris
DRR095709	NCBI	Apis cerana japonica
ERR2173373	ENA	Arabidopsis thaliana
SRR5817726	NCBI	Bacteroides caccae
SRR5817721	NCBI	Blautia hansenii DSM 20583
SRR5434252	NCBI	Borreliella burgdorferi
SRR5457531	NCBI	Clostridioides difficile
SRR5817725	NCBI	Enterocloster bolteae
DRR048282	NCBI	Dengue virus
DRR129651	NCBI	Vibrio litoralis DSM 17657

DRR129656	NCBI	Vibrio aphrogenes
DRR129658	NCBI	Vibrio algivorus
DRR129660	NCBI	Vibrio casei
DRR138509	NCBI	Homo sapiens
DRR138514	NCBI	Homo sapiens
DRR154114	NCBI	Streptococcus pneumoniae
DRR155409	NCBI	Staphylococcus aureus
DRR161061	NCBI	Staphylococcus argenteus
DRR164908	NCBI	Leptotrichia trevisanii
ERR1993378	ENA	Nippostrongylus brasiliensis
ERR2014660	EN	Solanum americanum
SRR6118140	NCBI	Escherichia coli
SRR5890147	NCBI	Helcococcus kunzii
SRR5902346	NCBI	Homo sapiens
SRR6348587	NCBI	Klebsiella pneumoniae
SRR5812844	NCBI	Lomentospora prolificans
SRR5319868	NCBI	Maccullochella peelii
SRR6192842	NCBI	Melanaphis sacchari
SRR5997379	NCBI	Mesoplasma chauliocola
SRR6129216	NCBI	Mesoplasma florum
SRR6082028	NCBI	Mesoplasma lactucae ATCC 49193
SRR5889392	NCBI	Oryza coarctata
SRR5851902	NCBI	Rhodotorula mucilaginosa
SRR6059708	ENA	Saccharomyces cerevisiae
ERR2177070	ENA	Solanum pennellii
Accession id / File Name	**Database**	**Species**
PacBio		
NCTC10116	NCTC	Acholeplasma laidlawii
NCTC12156	NCTC	Acinetobacter baumannii
NCTC10308	NCTC	Acinetobacter johnsonii

NCTC10221	NCTC	Actinobacillus ureae
NCTC8251	NCTC	Aerococcus viridans
NCTC8049	NCTC	Aeromonas hydrophila
NCTC5906	NCTC	Aggregatibacter aphrophilus ATCC 33389
Arabidopsis P5C3	PacBio GitHub DevNet	Arabidopsis thaliana
ACTC 14579	PacBio GitHub DevNet	Bacilus cereus 971
Bacillus subtilis W23	PacBio GitHub DevNet	Bacillus subtilis
NCTC12860	PacBio	Bartonella grahamii
NCTC11829	PacBio	Blautia producta
NCTC2611	PacBio	Brevibacillus brevis
ATCC 25416	PacBio GitHub DevNet	Burkholderia cepacia
NCTC11168	NCTC	Campylobacter jejuni
NCTC12120	NCTC	Cedecea neteri
NCTC13790	NCTC	Citrobacter braakii
NCTC11811	NCTC	clostridium indolis
SRX499318	ENA	drosophila melanogaster
NCTC11813	NCTC	Eggerthella lenta
NCTC418	NCTC	enterobacter aerogenes
OG1RF ATCC 47077D-5	PacBio GitHub DevNet	enterococcus faecalis
escherichia coli K12	PacBio GitHub DevNet	escherichia coli
NCTC4450	NCTC	escherichia coli
ATCC 9637	PacBio GitHub DevNet	escherichia coli W
NCTC11401	NCTC	fluoribacter gormanii
NCTC10562	NCTC	fusobacterium nucleatum
NCTC11413	NCTC	gallibacterium anatis
NCTC10667	NCTC	gordonia bronchialis

NCTC11483	NCTC	haemophilus ducreyi
NCTC11931	NCTC	haemophilus influenzae
ATCC 700824	PacBio GitHub DevNet	helicobacter pylori J99
HG002	PacBio GitHub DevNet	Homo sapiens
NCTC10816	NCTC	jonesia denitrificans
ATCC BAA-2146	PacBio GitHub DevNet	klebsiella pneumoniae
INF274 rs 2	NCTC	klebsiella pneumoniae
NCTC11040	NCTC	kytococcus sedentarius
NCTC13720	NCTC	lactobacillus acidophillus
NCTC13768	NCTC	lactobacillus brevis
NCTC11417	NCTC	legionella pneumophila
NCTC11994	NCTC	listeria monocytogenes
ATCC 43576	PacBio GitHub DevNet	methanocorpusculum labreanum Z
NCTC2665	NCTC	micrococcus luteus
NCTC11656	NCTC	mobiluncus curtisii
NCTC10437	NCTC	mycobacterium aurum
ATCC 700532	PacBio GitHub DevNet	neisseria meningitidis FAM18
NCTC11134	NCTC	nocardia farcinica
NCTC12168	NCTC	ochrobactrum anthropi
NCTC10343	NCTC	paenibacillus polymyxa
NCTC9381	NCTC	pantoea agglomerans
NCTC10204	NCTC	pasteurella multocida
NCTC11647	NCTC	photobacterium damselae
SRX3923117	NCBI	pseudomonas koreensis P19E3
rhodopseudomonas palustris	PacBio GitHub DevNet	rhodopseudomonas palustris
NCTC10917	NCTC	rothia dentocariosa
NCTC12419	NCTC	salmonella bongori

NCTC10384	NCTC	salmonella enterica
NCTC12961	NCTC	serratia plymuthica
NCTC13711	NCTC	staphylococcus argenteus
NCTC13277	NCTC	staphylococcus aureus
ATCC 25923	PacBio GitHub DevNet	staphylococcus aureus subsp aureus
NCTC7466	NCTC	streptococcus pneumoniae
ATCC 35405	PacBio GitHub DevNet	treponema denticola A
NCTC11317	NCTC	vibrio campbellii
NCTC10938	NCTC	yersinia enterocolitica
NCTC12986	NCTC	yersinia ruckeri