RESEARCH

4CAC: 4-class classifier of metagenome contigs using machine learning and assembly graphs

Lianrong Pu and Ron Shamir*

Abstract

Background: Microbial communities usually harbor a mix of bacteria, archaea, phages, plasmids, and microeukaryotes. Phages, plasmids, and microeukaryotes, which are present in low abundance in microbial communities, have complex interactions with bacteria and play important roles in horizontal gene transfer and antibiotic resistance. However, due to the difficulty of identifying phages, plasmids, and microeukaryotes in microbial communities, our understanding of these minor classes lags behind that of bacteria and archaea. Recently, several classifiers have been developed to separate one or two minor classes from bacteria and archaea in metagenome assemblies, but none can classify all of the four classes simultaneously. Moreover, existing classifiers have low precision on minor classes.

Results: We developed for the first time a classifier called 4CAC that is able to identify phages, plasmids, microeukaryotes, and prokaryotes simultaneously from metagenome assemblies. 4CAC generates an initial four-way classification using several sequence length-adjusted XGBoost algorithms and further improves the classification using the assembly graph. Evaluation on simulated and real metagenome datasets demonstrates that 4CAC substantially outperforms existing classifiers and combinations thereof on short reads. On long reads, it also shows an advantage unless the abundance of the minor classes is very low. 4CAC runs 1-2 orders of magnitude faster than the other classifiers.

Conclusions: 4CAC is the first four-class classifier that is able to identify phages, plasmids, microeukaryotes, and prokaryotes simultaneously from metagenome assemblies. Evaluation of 4CAC on simulated and real metagenome assemblies demonstrated that it outperforms the existing classifiers in most cases. 4CAC is an easy-to-use software and is also by far the fastest. The 4CAC software is available at https://github.com/Shamir-Lab/4CAC.

Keywords: Metagenome classification; Machine learning; Assembly graph; Phages; Plasmids; Microbial eukaryotes

Background

Microbial communities in natural and host-associated environments are often dominated by bacteria and coinhabited by fungi, protozoa, archaea, plasmids, and phages [1]. Changes in microbiome diversity, function, and density have been linked to a variety of disorders in many organisms [2, 3]. As the dominant group of species in microbial communities, bacteria have been widely studied. Taxonomic classification tools [4, 5] and metagenome binning tools [6, 7, 8, 9] were proposed to detect bacterial species present in a microbial community directly from reads or after assembling reads into contigs. It is known that the specific composition and abundance of certain bacterial species affect their host's health and fitness [10, 11, 12]. In contrast, our understanding of plasmids, phages, and microbial eukaryotes largely lags behind, due to their lower abundance and the difficulty of detecting them in microbial communities [13, 14]. Recent studies revealed that phages and plasmids play important roles in horizontal gene transfer events and antibiotic resistance [15, 16, 17, 18], and microbial eukaryotes have complex interaction with their hosts in both plant- and animal-associated microbiomes [14, 19]. To better understand the species composition and the function of each species in microbial communities, classifiers that can identify not only bacteria but also the other members of a microbial community are needed.

Many binary and three-class classifiers have been developed in recent years for separating phages and plas-

*Correspondence: rshamir@tau.ac.il
Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel
Full list of author information is available at the end of the article

mids from prokaryotes (bacteria and archaea) in microbial communities. VirSorter [20], DeepVirFinder [21], VIBRANT [22], and many other classifiers [23, 24] were designed to separate phages from prokaryotes. Plasmid classifiers, such as cBar [25], PlasFlow [26], PlasClass [27], and Deeplasmid [28], were developed to separate plasmids from prokaryotes. As both phages and plasmids are commonly found in microbial communities, three-class classifiers, such as PPR-Meta [29], viralVerify [30], and 3CAC [31], were proposed to simultaneously identify phages, plasmids, and prokaryotes from metagenome assemblies. In contrast, microbial eukaryotes, such as fungi and protozoa, are integral components of natural microbial communities but were commonly ignored or misclassified as prokaryotes in most metagenome analyses. More recently, EukRep [32], Tiara [33], and Whokaryote [34] were proposed to distinguish microeukaryotes from prokaryotes. However, even though prokaryotes, microeukaryotes, phages, and plasmids are present in most microbial communities, to the best of our knowledge, there are still no four-class classifiers that can simultaneously identify and distinguish all of them. Moreover, most classifiers ignore the fact that microbial communities are dominated by bacteria, and have low precision on the minor classes, such as phages, plasmids, and microeukaryotes [21, 31].

In this work, we present 4CAC (4-Class Adjacency-based Classifier), a fast algorithm to identify phages, plasmids, microeukaryotes, and prokaryotes simultaneously from metagenome assemblies. 4CAC generates an initial classification using a set of XGBoost algorithms trained on known reference genomes. The XGBoost classifier outputs four scores for each contig to indicate its confidence of being classified as phage, plasmid, prokaryote, or microeukaryote. To assure high precision in the classification of minor classes, we set higher score thresholds for classifying minor classes compared to prokaryotes. Subsequently, inspired by 3CAC, 4CAC utilizes the adjacency information in the assembly graph to improve the classification of short contigs and of contigs with lower confidence by the initial classification. Evaluation of 4CAC against combinations of existing classifiers on simulated and real metagenome datasets demonstrates that 4CAC has substantially better performance on short reads. On long reads, it also shows an advantage unless the abundance of the minor classes is very low.

Results

The 4CAC algorithm

To understand the species present in a microbial community, the common practice is to first assemble the sequencing reads into longer sequences called *contigs*, and then classify these contigs into classes. Broadly used metagenome assemblers, such as metaSPAdes [35] and metaFlye [36], use assembly graphs to combine overlapped reads (or k-mers) into contigs. Nodes in an assembly graph represent contigs and edges represent sequence overlaps between the corresponding contigs. Most of the existing classifiers take contigs as input and classify each of them independently based on their sequence. Our recent work on three-class classification demonstrated that neighboring contigs in an assembly graph are more likely to come from the same class and thus the adjacency information in the graph can assist the classification [31]. Therefore, here we introduce 4CAC, a four-class classifier that combines machine learning methods with assembly graph neighborhood information to classify each contig as phage, plasmid, prokaryote, microeukaryote, or uncertain.

Inspired by previous studies [27, 29, 37], to assure good classification of sequences with different lengths, we constructed five XGBoost models trained on sequence fragments of length 0.5k, 1kb, 5kb, 10kb, and 50kb, respectively. The k-mer composition of each fragment was used as the feature vector to train the XGBoost models. Given a sequence, we calculate its k-mer composition and classify it with the model that matches its length. More details on the design and implementation of the XGBoost classifiers can be found under Methods.

The XGBoost classifier outputs four scores between 0 and 1 for each sequence indicating its confidence of being classified as phage, plasmid, prokaryote, or microeukaryote. Existing algorithms [29, 27, 37] usually classify a sequence into the class with the highest score by default. To improve precision, a threshold can be specified, and sequences whose highest score is lower than the threshold will be classified as "uncertain". However, due to the overwhelming abundance of prokaryotes in the metagenome assemblies (usually $\geq 70\%$), a high threshold results in low recall in the classification of prokaryotes, while a low threshold results in low precision in the classification of the minor classes (phage, plasmid, and microeukaryote). Taking into consideration the class imbalance, we chose to set different thresholds for different classes. By default, a score threshold of 0.95 was set for phages and plasmids, and no score threshold was set for prokaryotes and eukaryotes. See Section "Length-specific classification" in Methods for more explanation on the choice of the score thresholds. This results in high precision for the classification of each class while maintaining high recall for the classification of prokaryotes.

Next, we exploit the assembly graph to improve the initial classification by the following two steps. (1) Correction of classified contigs. For a classified contig c, if

it has at least two classified neighbors and all of them belong to the same class while c belongs to a different class, 4CAC corrects the classification of c to be the same as its classified neighbors. (2) Propagation to unclassified contigs. For an unclassified contig c, if all of its classified neighbors belong to the same class, 4CAC assigns c to that class. See Section "Refining the classification using the assembly graph" in Methods for more details.

Simulated metagenomes

To evaluate the performance of 4CAC and existing classifiers, we simulated two short-read and two long-read metagenome datasets as follows. prokaryotes, their co-existing plasmids, phages, and microeukaryote genomes were selected from the NCBI GenBank Database to mimic species in a microbial community. All the genomes selected were released after December 2021, and thus they were not included in the training set of the classifier. As a *generic metagenome* scenario, we simulated reads in proportions mimicking regular metagenomic environments. As a *filtered metagenome* scenario, where reads from large genomes are filtered, the proportions were adjusted so that plasmids and viruses are enriched. The relative abundance of genomes within each class was set as in [27]. Short reads were simulated from the genome sequences using InSilicoSeq [38] and assembled by metaSPAdes. Long reads were simulated from the genome sequences using NanoSim [39] and assembled by metaFlye. Full details on the simulation and the assembly are provided in Methods. We denote by **Sim_AN** the simulation with A=S for short reads and A=L for long reads, N=G for the generic scenario and N=F for the filtered scenario. For example, Sim_SF is the short read filtered scenario. Table 1 summarizes the properties of the datasets and the assemblies.

Tested classifiers and evaluation metrics

As there are currently no four-class classifiers that can be compared with 4CAC, we combined existing classifiers to generate a four-way classification as follows. (1) The most straightforward idea is using DeepVirFinder, PlasClass, and Tiara to identify phages, plasmids, and eukaryotes from the metagenome assemblies, respectively. The remaining contigs were classified as prokaryotes. We tried all six orders of combining the classifiers and selected the best result to represent it. (Here and throughout, results were evaluated by their F1 score. See Methods for details.) This result is denoted by ***Dvf+PlasC+Tiara***. Here DeepVirFinder and PlasClass were selected because they are among the most popular phage classifiers and plasmid classifiers. For identifying eukaryotes, Tiara outperformed Whokaryote and EukRep in all our tests (See Supplementary Figure S1). Thus, in all subsequent combinations, we used Tiara to identify eukaryotes. (2) Our previous study showed that three-class classifiers outperform binary classifiers in classifying phages and plasmids simultaneously from metagenome assemblies [31]. Therefore, we further combined three-class classifiers viralVerify, PPR-Meta, and 3CAC with Tiara in the following way. For each combination, we tried classifying contigs either first by a three-class classifier or first by Tiara, and the better result was used to represent the combination. When classifying contigs by a three-class classifier first, contigs classified as chromosomes or uncertain were further classified by Tiara. When Tiara was run first, contigs classified as prokaryotes and uncertain by Tiara were further classified by the three-class classifier. For classifiers using 3CAC, we ran 3CAC based on the solution of viralVerify and PPR-Meta and selected the better result. We denote the ***combined classifier*** **A+Tiara** where A is the three-class classifier. In our benchmark, viralVerify was run with the '-p' option to enable three-class classification. PPR-Meta was run with a score threshold of 0.5 to assure reliable prediction.

XGBoost outperforms existing classifiers on simulated metagenomes

We first benchmarked our XGBoost four-way classifier (without using the graph information) against Dvf+PlasC+Tiara, PPR-Meta+Tiara, and viralVerify+Tiara on the simulated datasets. 3CAC+Tiara was not included in this comparison because 3CAC utilizes graph information to assist the classification. It was included in later comparisons (see next section). Figure 1 summarizes the results. The XGBoost classifier performed better than the combined classifiers in both precision and recall on short-read assemblies. On the long-read assemblies, viralVerify+Tiara and the XGBoost classifier had similar F1 scores, with XGBoost achieving better precision and viralVerify+Tiara achieving better recall. The separate precision, recall, and F1 scores for phage, plasmid, prokaryote, and eukaryote classification can be found in Supplementary Table S1. Consistent with our expectations, the XGBoost classifier achieved better precision and F1 scores on all the minor classes.

Performace on simulated metagenomes

Next, we tested the full 4CAC algorithm, including the correction and propagation steps. Since the XGBoost four-class classifier outperformed Dvf+PlasC+Tiara, PPR-Meta+Tiara, and viralVerify+Tiara, here we only compared 4CAC to its initial XGBoost classification and 3CAC+Tiara. Figure 2 summarizes the results. Consistent with the 3CAC algorithm, we observed that the graph refinement step improved the

recall with little or no loss of precision in all the tests. 4CAC outperformed 3CAC+Tiara in both precision and recall in almost all the simulated assemblies. The improvement of 4CAC was more substantial in short-read assemblies, perhaps because 3CAC already performs well in long-read assemblies. 4CAC improved the recall remarkably in Sim_SF, due to a larger proportion of short contigs in it (58% in Sim_SF vs. 19% in Sim_SG. See Table 1). Lower sequencing depth in Sim_SF results in a much more fragmented assembly. Surprisingly, the XGBoost classifier itself, without using the graph information, performed better than 3CAC+Tiara on the short-read assemblies. In the long-read assemblies, they had similar F1 scores, with the XGBoost classifier achieving better precision and 3CAC+Tiara achieving better recall.

Figure 3 presents the performance of the tested classifiers for each class on Sim_SG. In the classification of prokaryotes and eukaryotes, the graph refinement step of 4CAC improved both the precision and recall of the XGBoost classification. For phages and plasmids, 4CAC significantly improved recall, while sacrificing some precision over XGBoost, and achieved a higher F1 score. Compared to 3CAC+Tiara, 4CAC achieved better precision and recall for phages, prokaryotes, and eukaryotes. For plasmids, 4CAC had better precision and F1 score while 3CAC+Tiara had slightly higher recall. Overall, 4CAC had higher F1 scores than 3CAC+Tiara and the XGBoost classifier for every class. Surprisingly, the XGBoost classifier achieved higher F1 scores than 3CAC+Tiara in most cases except the classification of plasmids. Similar results were observed on the other simulated assemblies (See Supplementary Figures S2, S3, and S4).

Performance on real metagenome samples

We additionally tested the performance of classifiers on four real complex metagenomic datasets: (1) Short-read sequencing of 18 preborn infant fecal microbiome samples (NCBI accession number SRA052203), referred to as **Sharon** [40]. (2) Short-read sequencing of a microbiome sample from the Tara Oceans (NCBI accession number ERR868402), referred to as **Tara** [33]. Currently, there is no study exploring microeukaryotes in long-read sequencing of microbiome samples. To test our method on long-read sequencing metagenomic datasets, we selected two publicly available datasets: (3) Oxford Nanopore sequencing of two human saliva microbiome samples (NCBI accession number DRR214963 and DRR214965), referred to as **Oral_Nano** [41]. (4) Pacbio HiFi sequencing of a human gut microbiome sample (NCBI accession number SRR15275211), referred to as **Gut_HiFi** [42]. Datasets with short reads and long reads were assembled by metaSPAdes and metaFlye, respectively. In Sharon and Oral_Nano, the multiple samples in each dataset were co-assembled. To identify the class of contigs in these real metagenome assemblies, we used all the complete assemblies of bacteria, archaea, phages, plasmids, and microeukaryotes from the NCBI GenBank database as reference genomes and mapped contigs to these reference genomes using minimap2 [43]. A contig was considered matched to a reference sequence if it had $\geq 80\%$ mapping identity along $\geq 80\%$ of the contig length. Contigs that matched to reference genomes of two or more classes were excluded to avoid ambiguity. In all assemblies, contigs shorter than 500bp were not classified and excluded from the performance evaluation. Table 1 summarizes the properties of the datasets and the assemblies.

Similar to the result in simulated assemblies, Figure 4 shows that the graph refinement step improved both the precision and recall of the XGBoost classification and led to significant improvement in the F1 score in all the tests. On the short read datasets Sharon and Tara, in which microeukaryotes were previously identified [32, 33], 4CAC achieved moderately better precision than 3CAC+Tiara but dramatically improved the recall. For example, 4CAC improved the recall from 0.62 to 0.87 in the Tara dataset. As a result, 4CAC had substantially higher F1 score than 3CAC+Tiara.

Further analysis of the performance on the Sharon dataset reveals that the graph refinement step of 4CAC improved both the precision and recall of the XGBoost in each class classification (Figure 5). The improvement is more significant in the classification of plasmids and prokaryotes, while previous studies showed that it is more challenging to distinguish these two classes [29]. Compared to 3CAC+Tiara, 4CAC had higher F1 scores in the classification of prokaryotes and eukaryotes, but a lower F1 score on phages (Figure 5). A possible reason is that the proportion of phage contigs in the Sharon dataset is very small (0.6% vs. $\geq$ 1.3% in simulated assemblies. See Table 1). In this extreme case, viralVerify, which is used in 3CAC and classifies contigs based on their gene content, achieved higher precision than the machine learning-based methods, such as PPR-Meta and the XGBoost classifier.

On the two long-read datasets of human saliva and gut microbiome, 3CAC+Tiara outperformed 4CAC (Figures 4 (c) and (d)). Here too this is likely because each of the minor classes accounts for less than 0.6% of the contigs (Table 1). In the Gut_HiFi dataset with only three classes of contigs, it is interesting that all the four-class classifiers outperformed three-class classifiers (Supplementary Table S2).

Software and resource usage

Table 2 presents the runtime and memory usage of the classifiers. Note that all classifiers were run on

contigs at least 500 bp in each dataset since contigs shorter than 500 bp were excluded from our evaluation. To run DeepVirFinder, we also excluded contigs longer than 2 Mb because DeepVirFinder failed on these long contigs. For 3CAC we report the runtime of viralVerify and PPR-Meta, since they required the lion's share of the time, with the rest of 3CAC always taking less than 3 minutes. Due to the large runtime of viralVerify, PPR-Meta, and DeepVirFinder, 4CAC is much faster than the combined classifiers, which often require 1-2 orders of magnitudes more time. In all the tests, the peak memory usage of all tested classifiers was $\leq$ 27GB. Memory usage was the highest for DeepVirFinder in the long read assemblies and for 4CAC and PlasClass in most short read assemblies. All runs were performed on a 44-core, 2.2GHz server with 792GB of RAM. 4CAC is freely available via https://github.com/Shamir-Lab/4CAC.

Discussion and Conclusion

We presented 4CAC, the first classification algorithm for simultaneously identifying phages, plasmids, prokaryotes, and microeukaryotes in metagenome assemblies. Evaluation on simulated and real metagenomic datasets demonstrated that 4CAC substantially outperformed the combination of state-of-the-art three-class and eukaryote classifiers on short-read assemblies. 4CAC also has a large speed advantage over the combined classifiers, running usually 1-2 orders of magnitude faster. Unlike 3CAC, which requires running viralVerify or PPR-meta, 4CAC is a stand-alone algorithm and thus is also easier to use.

Taking into consideration the class imbalance of metagenome assemblies, we tried training XGBoost classifiers on imbalanced datasets with a default 0.5 threshold for all classes but did not achieve significant improvement. In contrast, the strategy of setting different probability thresholds for different classes led to a good trade-off between precision and recall. Applying subsequently the correction and propagation steps on the assembly graph significantly improved the classification of short contigs. As expected, since 3CAC uses the same refinement steps, 3CAC+Tiara achieved the second-best performance in all tests.

On simulated long read assemblies 4CAC again performed best. However, on two real datasets, the classifier combining 3CAC and Tiara was better, likely since the proportion of phages, plasmids, and eukaryotes in these samples was extremely small ($< 0.6\%$ vs. $\geq 1.3\%$ in other assemblies). Note that results may be biased by the underrepresentation of these classes in genomic databases. Given the current knowledge of species in metagenomes, we recommend using 4CAC on short reads and on host-filtered long read samples, while using 3CAC and Tiara on generic long read samples, where prokaryotes constitute the overwhelming majority.

Methods

Training and testing datasets

To train and test the XGBoost classifier, we downloaded all complete assemblies of phages, plasmids, prokaryotes (bacteria and archaea), and microeukaryotes (fungi and protozoa) from the National Center for Biotechnology Information (NCBI) GenBank database (download date April 22, 2022). After filtering out duplicate sequences, this database contained 31,129 prokaryotes, 69,882 phages, 28,702 plasmids, and 2,486 microeukaryotes. To evaluate the ability of 4CAC to identify novel species, 24,734 prokaryotes, 65,475 phages, 21,304 plasmids, and 2,315 microeukaryotes released before December 2021 were used to build the training set, while the remainder was used to build the testing set.

Training the XGBoost classifier

Inspired by previous studies [29, 27, 37], we trained several XGBoost models for different sequence lengths to assure the best performance. Specifically, five groups of fragments with lengths 0.5kb, 1kb, 5kb, 10kb, and 50kb were sampled from the reference genomes as artificial contigs. The composition information of each fragment is summarized by concatenating the canonical k-mer frequencies for k from 3 to 7, which results in a feature vector of length 10,952. We sampled 180k, 180k, 90k, 90k, and 50k fragments from each class to train the XGBoost models for sequence lengths 0.5kb, 1kb, 5kb, 10kb, and 50kb, respectively.

Length-specific classification

To assure the best classification for sequences of different lengths, we classify a sequence using the XGBoost model that is trained on fragments with the most similar length. Namely, the five XGBoost models we trained above are used to classify sequences in the respective length ranges (0, 0.75kb], (0.75kb, 3kb], (3kb, 7.5kb], (7.5kb, 30kb], and (30kb, ∞]. Given a sequence, we calculate its canonical k-mer frequency vector and use it as the feature vector to classify the sequence with the model that matches its length. The calculation of k-mer frequency vector can be performed in parallel for different sequences to achieve faster runtime.

For each sequence in the input, the XGBoost classifier outputs four scores between 0 and 1 indicating its confidence of being classified as phage, plasmid, prokaryote, or microeukaryote. Existing algorithms [29, 27, 37] usually classify a sequence into

the class with the highest score by default. To improve precision, a threshold can be specified, and sequences whose highest score is lower than the threshold will be classified as "uncertain". However, due to the overwhelming abundance of prokaryotes in the metagenome assemblies (usually $\geq 70\%$), a high threshold results in low recall in the classification of prokaryotes, while a low threshold results in low precision in the classification of the minor classes (phage, plasmid, and microeukaryote). Taking into consideration the class imbalance, we chose to set different thresholds for different classes. Tests on simulated metagenomes show that increasing score thresholds for prokaryotes and eukaryotes had little effect on the precision but decreased the recall a lot (Supplementary Figure S5). Thus we did not set specific score thresholds for prokaryotes and eukaryotes. In other words, a sequence was classified as prokaryote or eukaryote if that class had the highest score, irrespective of its value. For phages and plasmids, we tested several score thresholds (0.8, 0.85, 0.9, 0.95) and similar results were observed, while increasing the score threshold slightly improved the result in both precision and recall (see Supplementary Table S3). Note that increasing the score threshold did not decrease the recall of 4CAC, because the graph refinement step can significantly improve the recall over the initial classification. Therefore, in the 4CAC algorithm, we set the default score threshold of 0.95 for classifying contigs as phages and plasmids.

Refining the classification using the assembly graph

Nodes in an assembly graph represent contigs and edges represent sequence overlaps between the corresponding contigs. In our description below, the neighbors of a contig are its adjacent nodes in the undirected assembly graph. In the 4CAC algorithm, we exploit the assembly graph to improve the initial classification by the following two steps. The description here follows [31].

(1) Correction of classified contigs. All classified contigs are scanned in decreasing order of the number of their classified neighbors. For a classified contig c, if it has at least two classified neighbors and all of them belong to the same class while c belongs to a different class, 4CAC corrects the classification of c to be the same as its classified neighbors. Note that once a contig was corrected, the class of this contig and its classified neighbors will not be corrected anymore.

(2) Propagation to unclassified contigs. For an unclassified contig c, if all of its classified neighbors belong to the same class, 4CAC assigns c to that class. Unclassified contigs are scanned and classified in decreasing order of the number of their classified neighbors. We repeat this step until no propagation is possible.

Simulated datasets

We randomly selected 100 prokaryotes, 461 co-existing plasmids, 500 phages, and 6 microeukaryotes from the NCBI GenBank Database to mimic species in a microbial community. All the genomes selected were released after December 2021, and thus they were not included in the training set of the classifier. Two short-read and two long-read metagenome assemblies were generated from this microbial community as follows. As a *generic metagenome* scenario, we simulated reads from prokaryotes, eukaryotes, phages, and plasmids in a ratio of 56:24:10:10. As a *filtered metagenome* scenario, where reads from large genomes are filtered and thus plasmids and viruses are enriched, we simulated reads from prokaryotes, eukaryotes, phages, and plasmids in a ratio of 14:6:40:40. The abundance profiles of prokaryotes, eukaryotes, and phages were modeled by the log-normal distribution as in [27]. The copy numbers of co-existing plasmids were simulated by the geometric distribution with parameter $p = min(1, log_{10}(L)/7)$ where L is the plasmid length as in [27]. The abundance profile of plasmid genomes was calculated from their host abundance profile and the copy numbers of plasmids. 150bp short reads were simulated from the genome sequences using InSilicoSeq and assembled by metaSPAdes. Long reads were simulated from the genome sequences using NanoSim and assembled by metaFlye. The error rate of long reads was 9.8% and their average length was 14.9kb. For each assembly, contigs were mapped to the reference genomes by metaQUAST to define the ground truth. To assure confident assignment of contigs, metaQUAST excludes contigs shorter than 500bp by default.

Evaluation criteria

All the classifiers were evaluated based on precision, recall, and F1 scores calculated as follows.

- **Precision:** the fraction of correctly classified contigs among all classified contigs. Note that uncertain contigs were not included in this calculation.
- **Recall:** the fraction of correctly classified contigs among all contigs.
- **F1 score:** the harmonic mean of the precision and recall, or equivalently: $F1\ score = (2 * precision * recall)/(precision + recall)$.

Following [27, 29], the precision, recall, and F1 scores here were calculated by counting the number of contigs and did not take into account their length. The precision and recall were also calculated separately for phage, plasmid, prokaryote, and eukaryote classification. For example, the precision of phage classification was the fraction of correctly classified phage contigs among all contigs classified as phages, and the recall of

phage classification was the fraction of correctly classified phage contigs among all phage contigs.

Acknowledgements
We thank members of the Shamir Lab for their help and comments - David Pellow, Ron Saad, and Hagai Levi.

Funding
This study was supported in part by the Israel Science Foundation (grants 1339/18 and 2206/22). L.P. was supported in part by postdoctoral fellowships from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University, and from the Planning & Budgeting Committee (PBC) of the Council for Higher Education (CHE) in Israel.

Availability of data and materials
4CAC is implemented in Python and is available on GitHub (https://github.com/Shamir-Lab/4CAC). All sequencing datasets analyzed in this study are available in National Center for Biotechnology Information (NCBI), accession numbers: SRA052203 for the Sharon dataset, ERR868402 for the Tara Ocean dataset, DRR214963 and DRR214965 for the Oral_Nano dataset, and SRR15275211 for the Gut_HiFi dataset.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Both authors consented.

Authors' contributions
LP and RS designed the study. LP developed and implemented the 4CAC algorithm, analyzed the datasets, and wrote the manuscript. RS oversaw the development of the project and edited the manuscript. Both authors read and approved the final manuscript.

Author details
Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel.

References

1. Marcelino, V.R., Clausen, P.T., Buchmann, J.P., Wille, M., Iredell, J.R., Meyer, W., Lund, O., Sorrell, T.C., Holmes, E.C.: CCMetagen: comprehensive and accurate identification of eukaryotes and prokaryotes in metagenomic data. Genome Biology **21**(1), 1–15 (2020)
2. McKenney, P.T., Pamer, E.G.: From hype to hope: the gut microbiota in enteric infectious disease. Cell **163**(6), 1326–1332 (2015)
3. Gilbert, J.A., Blaser, M.J., Caporaso, J.G., Jansson, J.K., Lynch, S.V., Knight, R.: Current understanding of the human microbiome. Nature Medicine **24**(4), 392–400 (2018)
4. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biology **15**(3), 1–12 (2014)
5. Wood, D.E., Lu, J., Langmead, B.: Improved metagenomic analysis with kraken 2. Genome Biology **20**(1), 1–13 (2019)
6. Mallawaarachchi, V., Wickramarachchi, A., Lin, Y.: GraphBin: refined binning of metagenomic contigs using assembly graphs. Bioinformatics **36**(11), 3307–3313 (2020)
7. Mallawaarachchi, V., Lin, Y.: Accurate binning of metagenomic contigs using composition, coverage, and assembly graphs. Journal of Computational Biology (2022)
8. Kang, D.D., Li, F., Kirton, E., Thomas, A., Egan, R., An, H., Wang, Z.: MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. PeerJ **7**, 7359 (2019)
9. Wu, Y.-W., Simmons, B.A., Singer, S.W.: MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics **32**(4), 605–607 (2016)
10. Brooks, B., Olm, M.R., Firek, B.A., Baker, R., Thomas, B.C., Morowitz, M.J., Banfield, J.F.: Strain-resolved analysis of hospital rooms and infants reveals overlap between the human and room microbiome. Nature Communications **8**(1), 1–7 (2017)
11. Liang, Z., Dong, C., Liang, H., Zhen, Y., Zhou, R., Han, Y., Liang, Z.: A microbiome study reveals the potential relationship between the bacterial diversity of a gymnastics hall and human health. Scientific Reports **12**(1), 1–9 (2022)
12. Moss, E.L., Maghini, D.G., Bhatt, A.S.: Complete, closed bacterial genomes from microbiomes using nanopore sequencing. Nature Biotechnology **38**(6), 701–707 (2020)
13. Benler, S., Yutin, N., Antipov, D., Rayko, M., Shmakov, S., Gussow, A.B., Pevzner, P., Koonin, E.V.: Thousands of previously unknown phages discovered in whole-community human gut metagenomes. Microbiome **9**(1), 1–17 (2021)
14. Lind, A.L., Pollard, K.S.: Accurate and sensitive detection of microbial eukaryotes from whole metagenome shotgun sequencing. Microbiome **9**(1), 1–18 (2021)
15. Calero-Cáceres, W., Ye, M., Balcázar, J.L.: Bacteriophages as environmental reservoirs of antibiotic resistance. Trends in Microbiology **27**(7), 570–577 (2019)
16. Wein, T., Hülter, N., Mizrahi, I., Dagan, T.: Emergence of plasmid stability under non-selective conditions maintains antibiotic resistance. Nature Communications 10: 2595 (2019)
17. Lopatkin, A., Meredith, H., Srimani, J., Pfeiffer, C., Durrett, R., You, L.: Persistence and reversal of plasmid-mediated antibiotic resistance. Nature Communications 8: 1689 (2017)
18. Sitaraman, R.: Prokaryotic horizontal gene transfer within the human holobiont: ecological-evolutionary inferences, implications and possibilities. Microbiome **6**(1), 1–14 (2018)
19. Olm, M.R., West, P.T., Brooks, B., Firek, B.A., Baker, R., Morowitz, M.J., Banfield, J.F.: Genome-resolved metagenomics of eukaryotic populations during early colonization of premature infants and in hospital rooms. Microbiome **7**(1), 1–16 (2019)
20. Roux, S., Enault, F., Hurwitz, B.L., Sullivan, M.B.: VirSorter: mining viral signal from microbial genomic data. PeerJ **3**, 985 (2015)
21. Ren, J., Song, K., Deng, C., Ahlgren, N.A., Fuhrman, J.A., Li, Y., Xie, X., Poplin, R., Sun, F.: Identifying viruses from metagenomic data using deep learning. Quantitative Biology, 1–14 (2020)
22. Kieft, K., Zhou, Z., Anantharaman, K.: VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences. Microbiome **8**(1), 1–23 (2020)
23. Guo, J., Bolduc, B., Zayed, A.A., Varsani, A., Dominguez-Huerta, G., Delmont, T.O., Pratama, A.A., Gazitúa, M.C., Vik, D., Sullivan, M.B., *et al.*: VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. Microbiome **9**(1), 1–13 (2021)
24. Auslander, N., Gussow, A.B., Benler, S., Wolf, Y.I., Koonin, E.V.: Seeker: alignment-free identification of bacteriophage genomes by deep learning. Nucleic Acids Research **48**(21), 121–121 (2020)
25. Zhou, F., Xu, Y.: cBar: a computer program to distinguish plasmid-derived from chromosome-derived sequence fragments in metagenomics data. Bioinformatics **26**(16), 2051–2052 (2010)
26. Krawczyk, P.S., Lipinski, L., Dziembowski, A.: PlasFlow: predicting plasmid sequences in metagenomic data using genome signatures. Nucleic Acids Research **46**(6), 35–35 (2018)
27. Pellow, D., Mizrahi, I., Shamir, R.: PlasClass improves plasmid sequence classification. PLoS Computational Biology **16**(4), 1007781 (2020)
28. Andreopoulos, W.B., Geller, A.M., Lucke, M., Balewski, J., Clum, A., Ivanova, N.N., Levy, A.: Deeplasmid: deep learning accurately separates plasmids from bacterial chromosomes. Nucleic Acids Research **50**(3), 17–17 (2022)
29. Fang, Z., Tan, J., Wu, S., Li, M., Xu, C., Xie, Z., Zhu, H.: PPR-Meta: a tool for identifying phages and plasmids from metagenomic fragments using deep learning. GigaScience **8**(6), 066 (2019)
30. Antipov, D., Raiko, M., Lapidus, A., Pevzner, P.A.: Metaviral SPAdes: assembly of viruses from metagenomic data. Bioinformatics **36**(14), 4126–4129 (2020)
31. Pu, L., Shamir, R.: 3CAC: improving the classification of phages and plasmids in metagenomic assemblies using assembly graphs. Bioinformatics **38**(Supplement_2), 56–61 (2022)
32. West, P.T., Probst, A.J., Grigoriev, I.V., Thomas, B.C., Banfield, J.F.: Genome-reconstruction for eukaryotes from complex natural microbial communities. Genome Research **28**(4), 569–580 (2018)

33. Karlicki, M., Antonowicz, S., Karnkowska, A.: Tiara: deep learning-based classification system for eukaryotic sequences. Bioinformatics **38**(2), 344–350 (2022)
34. Pronk, L.J., Medema, M.H.: Whokaryote: distinguishing eukaryotic and prokaryotic contigs in metagenomes based on gene structure. BioRxiv (2021)
35. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A.: metaSPAdes: a new versatile metagenomic assembler. Genome Research **27**(5), 824–834 (2017)
36. Kolmogorov, M., Bickhart, D.M., Behsaz, B., Gurevich, A., Rayko, M., Shin, S.B., Kuhn, K., Yuan, J., Polevikov, E., Smith, T.P., *et al.*: metaFlye: scalable long-read metagenome assembly using repeat graphs. Nature Methods **17**(11), 1103–1110 (2020)
37. Ren, J., Ahlgren, N.A., Lu, Y.Y., Fuhrman, J.A., Sun, F.: VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. Microbiome **5**(1), 1–20 (2017)
38. Gourlé, H., Karlsson-Lindsjö, O., Hayer, J., Bongcam-Rudloff, E.: Simulating illumina metagenomic data with InSilicoSeq. Bioinformatics **35**(3), 521–522 (2019)
39. Yang, C., Chu, J., Warren, R.L., Birol, I.: NanoSim: nanopore sequence read simulator based on statistical characterization. GigaScience **6**(4), 010 (2017)
40. Sharon, I., Morowitz, M.J., Thomas, B.C., Costello, E.K., Relman, D.A., Banfield, J.F.: Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. Genome Research **23**(1), 111–120 (2013)
41. Yahara, K., Suzuki, M., Hirabayashi, A., Suda, W., Hattori, M., Suzuki, Y., Okazaki, Y.: Long-read metagenomics using promethion uncovers oral bacteriophages and their interaction with host bacteria. Nature Communications **12**(1), 1–12 (2021)
42. Portik, D.: Website of hifi reads: https://github.com/pacificbiosciences/pb-metagenomics-tools/blob/master/docs/pacbio-data.md
43. Li, H.: Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics **34**(18), 3094–3100 (2018)

Figures

Figure 1 Performance of four-class classifiers on simulated metagenomes without using graph information. Sim_SG and Sim_SF are datasets of short reads while Sim_LG and Sim_LF are datasets of long reads.

Figure 2 Performance of four-class classifiers on simulated metagenomes. Sim_SG and Sim_SF are datasets of short reads while Sim_LG and Sim_LF are datasets of long reads.

Figure 3 Performance on each class for simulated short read dataset Sim_SG.

Figure 4 Performance of four-class classifiers on the real datasets. (a) Sharon and (b) Tara were assembled from short reads, (c) Oral_Nano and (d) Gut_HiFi were assembled from long reads.

Figure 5 Performance on each class for real short read dataset Sharon.

Tables

Additional Files

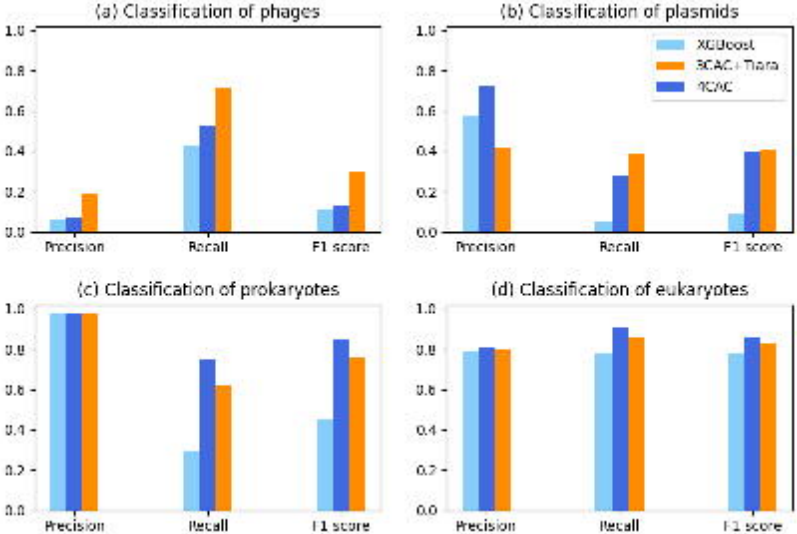
Additional file 1 — Supplementary Information
Supplementary Table S1-S3 and Figures S1-S5.

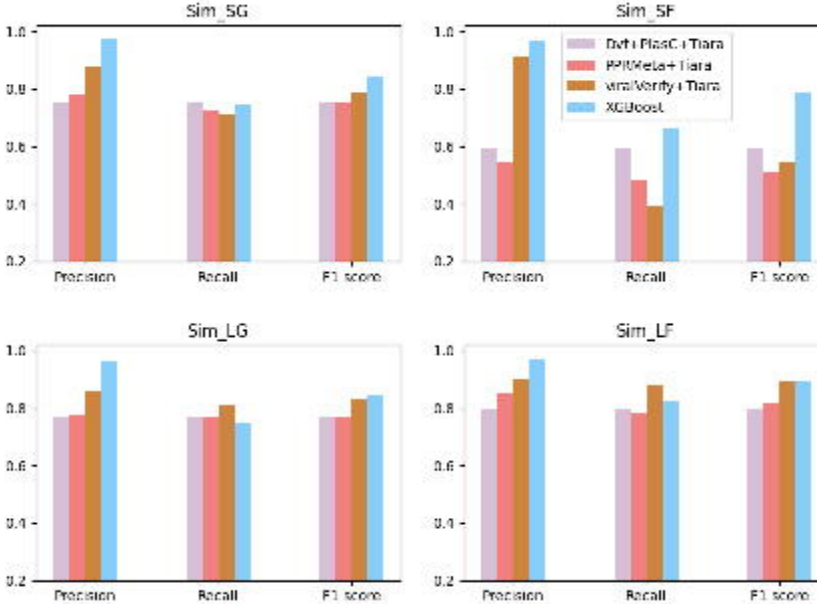
Table 1 Properties of the simulated and the real metagenomic datasets.

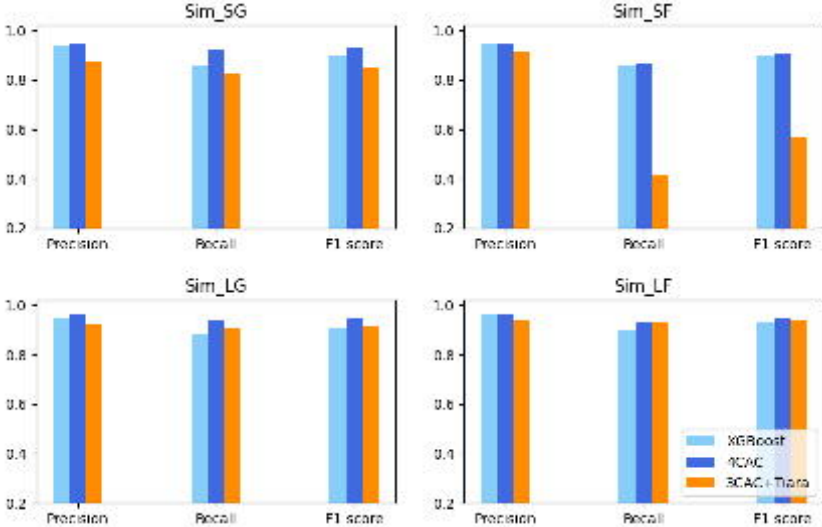
Dataset	Read type	Number of read(M)				Number of assembled contigs				Short contigs
		prokaryote	eukaryote	plasmid	phage	prokaryote	eukaryote	plasmid	phage	($<$ 1kb)
Sim_SG	MiSeq	56	24	10	10	15,460	8,112	1,725	1,275	5,095
Sim_SF	MiSeq	3.5	1.5	10	10	50,546	44,378	1,650	1,256	56,735
Sim_LG	Nanopore	0.56	0.24	0.1	0.1	1,575	148	193	202	125
Sim_LF	Nanopore	0.035	0.015	0.1	0.1	922	343	207	193	33
Sharon	HiSeq	106.3 in total				3,097	533	87	21	1,169
Tara	HiSeq	190.7 in total				16,156	31	153	1,270	11,643
Oral_Nano	Nanopore	5.6 in total				9,112	50	11	23	1,888
Gut_HiFi	Pacbio HiFi	1.9 in total				4,958	0	27	30	203

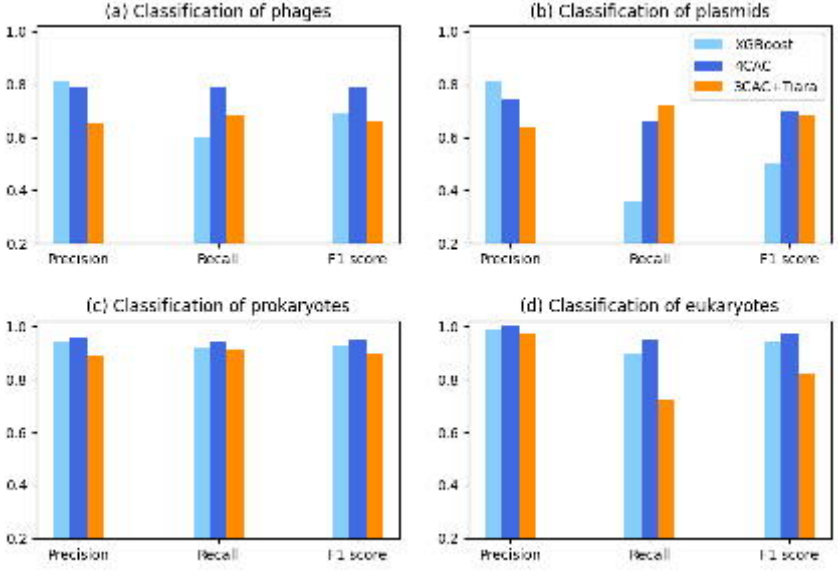
Table 2 Resource usage of the tested classifiers. Runtime is measured by wall clock time. vV, PM, PlasC, and DeepVF represent classifiers viralVerify, PPR-Meta, PlasClass, and DeepVirFinder, respectively.

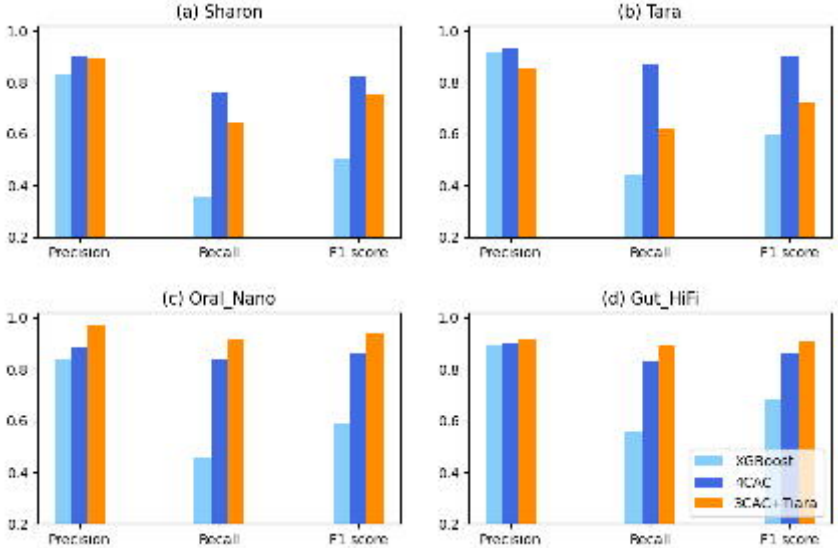
Datasets	Runtime (minutes)						RAM (GB)					
	4CAC	vV	PM	Tiara	PlasC	DeepVF	4CAC	vV	PM	Tiara	PlasC	DeepVF
Sim_SG	6.9	**322**	60.4	2.9	4.2	91.2	3.9	1.3	6.5	2.2	5.7	**21.3**
Sim_SF	4.8	**175**	25.1	2.8	3.9	62.1	15.5	0.7	6.4	1.6	**16.9**	10.2
Sim_LG	3.7	**185.4**	33.4	1.3	2.3	33.2	0.7	0.2	6.3	1.4	0.5	**13.8**
Sim_LF	1.4	**77.5**	14.9	0.7	1.2	14.7	0.6	0.3	6.3	1.4	0.4	**7.9**
Sharon	1.4	**29.9**	7.3	0.5	1.2	16.3	**9.9**	0.1	6.3	1.4	7.3	9.3
Tara	14.3	**221.1**	94.7	4.4	11.2	92.3	**26.2**	9.8	12.3	2.8	25.2	2.8
Oral_Nano	8.2	**452.5**	84.4	3.5	5.6	88.6	2.2	1.5	6.2	2.8	2.9	**10**
Gut_HiFi	9.3	**677.8**	124	4.8	8.1	109.8	2.7	2.4	9.3	3.7	3.3	**9.4**


(a) Classification of phages
(b) Classification of plasmids
(c) Classification of prokaryotes
(d) Classification of eukaryotes
XGBoost
SCAC + Tiara
SCAC
1.0
0.8
0.6
0.4
0.2
0.0
Precision
Recall
F1 score


Sim_SG
Sim_SF
Sim_LG
Sim_LF
1.0
0.8
0.6
0.4
0.2
Precision
Recall
F1 score
DeepVirFinder + Tiara
PPR-Meta + Tiara
viralVerify + Tiara
XGBoost


Sim_SG
Sim_SF
Sim_LG
Sim_LF
1.0
0.8
0.6
0.4
0.2
Precision
Recall
F1 score
XGBoost
4CAC
3CAC+Theta


(a) Classification of phages
(b) Classification of plasmids
(c) Classification of prokaryotes
(d) Classification of eukaryotes
1.0
0.8
0.6
0.4
0.2
Precision
Recall
F1 score
XGBoost
4CAC
3CAC + Tiara


(a) Sharon
(b) Tara
(c) Oral_Nano
(d) Gut_HiFi
1.0
0.8
0.6
0.4
0.2
Precision
Recall
F1 score
XGBoost
SCAC
SCAC+Tiara