Nucleic Acids Research, 2024, 1–12
https://doi.org/10.1093/nar/gkae799
Methods

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

Lianrong Pu [1,2,*] **and Ron Shamir** [1,*]

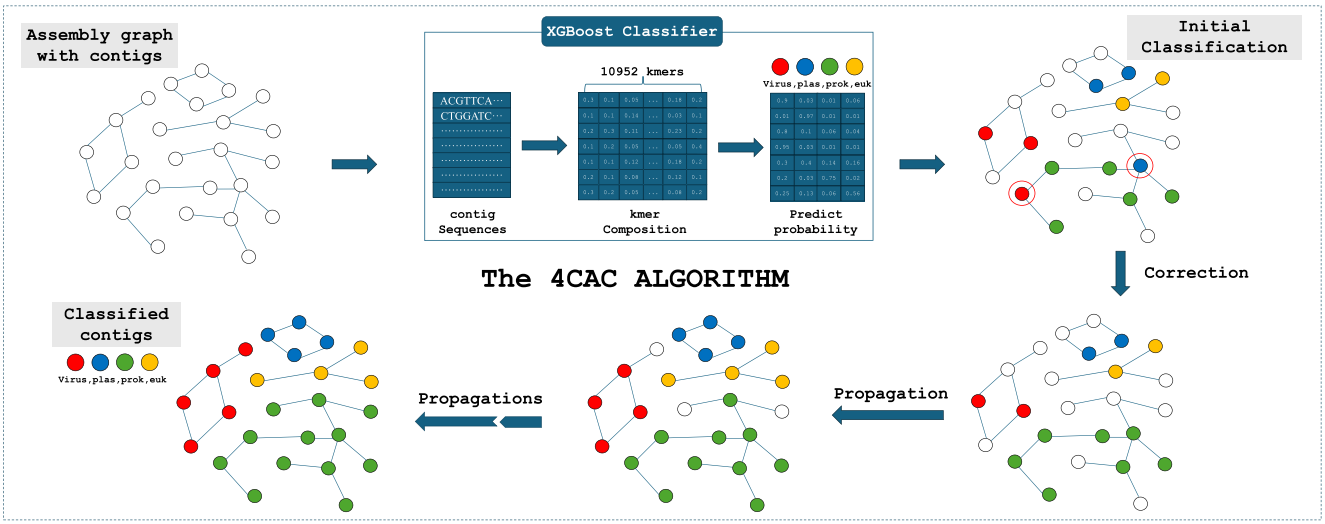
[1]The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel
[2]School of Computer Science and Technology, Shandong University, Qingdao, China

[*]To whom correspondence should be addressed. Tel: +972 3 640 5383; Email: rshamir@tau.ac.il
Correspondence may also be addressed to Lianrong Pu. Email: lianrong.pu@gmail.com

Abstract

Microbial communities usually harbor a mix of bacteria, archaea, plasmids, viruses and microeukaryotes. Within these communities, viruses, plasmids, and microeukaryotes coexist in relatively low abundance, yet they engage in intricate interactions with bacteria. Moreover, viruses and plasmids, as mobile genetic elements, play important roles in horizontal gene transfer and the development of antibiotic resistance within microbial populations. However, due to the difficulty of identifying viruses, 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 more minor classes from bacteria and archaea in metagenome assemblies. However, these classifiers often overlook the issue of class imbalance, leading to low precision in identifying the minor classes. Here, we developed a classifier called 4CAC that is able to identify viruses, plasmids, microeukaryotes, and prokaryotes simultaneously from metagenome assemblies. 4CAC generates an initial four-way classification using several sequence length-adjusted XGBoost models 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. The 4CAC software is available at https://github.com/Shamir-Lab/4CAC.

Graphical abstract



Introduction

Microbial communities in natural and host-associated environments are often dominated by bacteria and coinhabited by archaea, fungi, protozoa, plasmids and viruses (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 (6–9) were proposed to detect bacterial species present in a microbial community directly from

Received: December 21, 2023. **Revised:** July 13, 2024. **Editorial Decision:** August 26, 2024. **Accepted:** September 2, 2024

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 (10–12). In contrast, our understanding of plasmids, viruses, 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 viruses and plasmids play important roles in horizontal gene transfer events and antibiotic resistance (15–18), 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 viruses and plasmids from prokaryotes (bacteria and archaea) in microbial communities. VirSorter2 (20), DeepVirFinder (21), VIBRANT (22) and many other classifiers (23,24) were designed to separate viruses from prokaryotes. Plasmid classifiers, such as PlasFlow (25), PlasClass (26), Deeplasmid (27), PLASMe (28) and Platon (29) were developed to separate plasmids from prokaryotes. As both viruses and plasmids are commonly found in microbial communities, three-class classifiers, such as PPR-Meta (30), viralVerify (31), 3CAC (32) and geNomad (33) were proposed to simultaneously identify viruses, 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 (34), Tiara (35) and Whokaryote (36) were proposed to distinguish microeukaryotes from prokaryotes. However, even though prokaryotes, microeukaryotes, viruses and plasmids are present in most microbial communities, there is a lack of four-class classifiers that can simultaneously identify and distinguish all of them. (DeepMicroClass (37), a five-way classifier, was published while this article was under review and is included in our analysis.) Moreover, most classifiers overlook the fact that microbial communities are dominated by bacteria and thus classes are imbalanced in metagenome assemblies. Therefore, they have low precision in classifying minor classes such as viruses, plasmids, and microeukaryotes (21,32). Additionally, although short contigs usually account for a large proportion of short-read assemblies, existing classifiers exhibit poor performance on short contigs by either misclassifying them or designating them as uncertain.

In this work, we present 4CAC (4-class adjacency-based classifier), a fast algorithm to identify viruses, plasmids, microeukaryotes, and prokaryotes simultaneously from metagenome assemblies. 4CAC generates an initial classification using a set of XGBoost models trained on known reference genomes. The XGBoost classifier outputs four scores for each contig to indicate its confidence of being classified as a virus, 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 in 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.

Materials and methods

4CAC accepts as input a set of contigs and the associated assembly graph, and aims to classify each contig in the input as virus, plasmid, prokaryote, microeukaryote, or uncertain. 4CAC generates four-class classifications with high precision by combining machine learning methods with graph information. The details of the algorithm are explained below.

Design and implementation of the XGBoost classifier

Training, validation and testing datasets

To train and test our classifier, we downloaded all complete assemblies of viruses, plasmids, prokaryotes (bacteria and archaea), and microeukaryotes (fungi and protozoa) from the National Center for Biotechnology Information (NCBI) GenBank database (download date 22 April 2023). After filtering out duplicate sequences, this database contained 31 129 prokaryotes, 69 882 viruses, 28 702 plasmids and 2486 microeukaryotes, which were further divided into three parts based on the release dates of the genomes. Genomes released before December 2021 were used for **training**, those released between December 2021 and April 2022 were used for **validation**, and those released after April 2022 were used for **testing**. In this way, the sets of genomes for training, validation, and testing are disjoint. Statistics from the training set were used to construct the XGBoost classifier. The validation set was used to tune the threshold of each class. The test genomes were used to construct simulated metagenomes that were used to benchmark all algorithms.

Note that splitting the data based on release date does not completely prevent high similarity between the datasets, as the identification of novel species relies heavily on known reference genomes. Supplementary Figure S1 summarizes the similarity distribution between the testing and training datasets. Reassuringly, for prokaryotes, eukaryotes, viruses, and plasmids, only 3.7%, 9.5%, 15.4% and 18.2% of the testing genomes, respectively, had similarity >85% to the training genomes. We believe this constitutes a realistic scenario, as some sequences encountered in new samples are likely to have highly similar counterparts in the database.

Training the XGBoost classifier

Inspired by previous studies (26,30,38), we trained several XGBoost models for different sequence lengths to assure the best performance. Specifically, five groups of fragments with lengths 0.5 kb, 1 kb, 5 kb, 10 kb and 50 kb were sampled from the training 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 0.5 kb, 1 kb, 5 kb, 10 kb and 50 kb, 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, 0.75 kb], (0.75 kb, 3 kb], (3 kb, 7.5 kb], (7.5 kb, 30 kb], and (30 kb, ∞]. 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 a virus, plasmid, prokaryote or microeukaryote. Existing algorithms (26,30,38) 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 ≥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 (virus, plasmid, and microeukaryote). Taking into consideration the class imbalance in metagenome assemblies, we chose to set different thresholds for classifying different classes. Tests on simulated metagenomes assembled from the validation dataset show that increasing score thresholds for prokaryotes and eukaryotes had little effect on the precision but decreased the recall a lot (Supplementary Figures S2 and S3). 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 viruses 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 S1 and Figure S4). Note that increasing the score threshold did not decrease the recall of 4CAC, because the graph refinement step implemented later 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 viruses and plasmids.

Refining the classification using the assembly graph

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 (39) and metaFlye (40), 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. In our description below, the neighbors of a contig are its adjacent nodes in the undirected assembly graph. Most 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 (32). Therefore, here too we exploit the assembly graph to improve the initial classification by the following two steps.

(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.

Since each contig can be corrected at most once, the correction step is run only once. After the correction step, the propagation step is applied iteratively until no more uncertain contigs can be classified.

Simulated metagenomes

To evaluate the performance of 4CAC and existing classifiers, we simulated two short-read and two long-read metagenome test datasets as follows. 100 prokaryotes, 461 co-existing plasmids, 500 viruses and 6 microeukaryotes were randomly selected from the NCBI GenBank Database to mimic species in a microbial community. All the genomes were selected from the test set, and thus they were not included in the training and validation sets of the classifier. As a *generic metagenome* scenario, we simulated reads in proportions that mimic typical metagenomic environments. Specifically, reads from prokaryotes, eukaryotes, viruses and plasmids were simulated in a ratio of 56:24:10:10. As a *filtered metagenome* scenario, where reads from host genomes are filtered out and thus plasmids and viruses are enriched, we simulated reads from prokaryotes, eukaryotes, viruses and plasmids in a ratio of 14:6:40:40. To achieve the desired proportions after the filtering step, we randomly removed 93.8% of host reads from the generic metagenome scenario while keeping the reads from viruses and plasmids unchanged. The relative abundance of genomes within each class was set as in (26). The abundance profiles of prokaryotes, eukaryotes, and viruses were modeled by the log-normal distribution. 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 (26). The abundance profile of plasmid genomes was calculated from their host abundance profile and the copy numbers of plasmids. 150 bp short reads were simulated from the genome sequences using InSilicoSeq (41) and assembled by metaSPAdes. Long reads were simulated from the genome sequences using NanoSim (42) and assembled by metaFlye. The error rate of long reads was 9.8% and their average length was 14.9 kb. For each assembly, contigs were mapped to the reference genomes by metaQUAST (43) to define the ground truth. To ensure confident assignment of contigs, contigs with ambiguous alignment results by metaQUAST and contigs shorter than 500 bp were excluded from our analysis. 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.

To determine the best score thresholds of classifying contigs by XGBoost classifier, we generated four additional validation datasets analogously to those described above, by selecting 100 prokaryotes, 424 co-existing plasmids, 500 viruses and 6 microeukaryotes from the validation genomes. We call them Sim_SG_valid, Sim_SF_valid, Sim_LG_valid and Sim_LF_valid, respectively.

Finally, since the newly released species used in the test and validation sets may contain species similar to those used for training the classifier, we simulated additional datasets wherein all species in the test set are guaranteed to be sufficiently different from the training species. We selected 100 prokaryotes, 419 co-existing plasmids, 500 viruses and 6 microeukaryotes from the test set that have similarities less than 85% to any of the species in the training and validation set. The other steps of generating reads and assembling reads into contigs were the same as above. We call these sets Sim_SG_lowANI, Sim_SF_lowANI, Sim_LG_lowANI and Sim_LF_lowANI, respectively. Table 1 summarizes the properties of the datasets and the assemblies.

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 (26,30), 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 virus, plasmid, prokaryote and eukaryote classification. For example, the precision of virus classification was the fraction of correctly classified virus contigs among all contigs classified as viruses, and the recall of virus classification was the fraction of correctly classified virus contigs among all virus contigs.

Results

We benchmarked the performance of 4CAC against existing classifiers using both simulated and real metagenome assemblies of long and short reads. For comparison, we selected eight binary classifiers and four three-way classifiers. Our analysis revealed that 4CAC outperforms existing classifiers across almost all the tested datasets. Furthermore, we combined existing binary and three-way classifiers to generate four-way classifications and evaluated their effectiveness. Additionally, a five-way classifier, DeepMicroClass (37), was published while this article was under review and is included in our analysis.

4CAC outperforms existing classifiers on simulated metagenomes

To evaluate 4CAC in classifying viruses, plasmids, and eukaryotes from metagenome assemblies, we conducted a comprehensive comparison against the start-of-the-art binary classifiers, including the viral classifiers DeepVirFinder and VIBRANT, the plasmid classifiers PlasClass, Platon and PLASMe, and the eukaryote classifiers EukRep, Whokaryote, and Tiara. Figure 1 summarizes the results. 4CAC outperforms almost all binary classifiers in each class classification, except in the classification of eukaryotes, where Tiara achieves a slightly higher F1 score on the Sim_LF dataset (here and throughout, results were evaluated by their F1 score. See Methods for details). In classifying viruses, the XGBoost classifier designed in this study, without using the graph information, outperforms the start-of-the-art viral classifiers. In plasmid classification, the XGBoost classifier achieves the second-best performance in long-read assemblies, while Platon and PLASMe are the second-best in short-read assemblies. In classifying eukaryotes, all classifiers have good performance in long-read assemblies with Tiara and 4CAC achieving the best result. However, in short-read assemblies, 4CAC and the XGBoost classifier maintain consistently high F1 scores while the performance of the other eukaryote classifiers is markedly lower. Not surprisingly, by utilizing the graph information, 4CAC improved the XGBoost classification results in 11 out of 12 classifications across all datasets, and the improvement is dramatic in classifying plasmids from short-read assemblies.

Supplementary Figure S5 summarizes the precision and recall of 4CAC and the binary classifiers. In classifying viruses and eukaryotes, 4CAC achieves the best precision and comparable recall in all the simulated datasets. In the classification of plasmids, Platon achieves the highest precision but low recall, while PLASMe achieves the highest recall but low precision across all four simulated datasets. 4CAC attains the second-best precision and recall, resulting in the best F1 scores in all the simulated datasets. Compared to the initial classification generated by XGBoost classifier, the graph refinement step of 4CAC dramatically improves the recall in classifying viruses and plasmids from short-read assemblies.

Note that these binary classifiers were developed to identify viruses, plasmids, and microeukaryotes from metagenome assemblies respectively, and classify the remaining contigs as prokaryotes. There are currently no tools specifically designed for prokaryote classification. Supplementary Figure S6 illustrates the performance of binary classifiers and 4CAC in classifying prokaryotes. As anticipated, 4CAC achieves the best performance.

In addition, we conducted a comprehensive comparison between 4CAC and a set of three-way classifiers specifically designed to classify viruses and plasmids simultaneously from metagenome assemblies. The evaluated classifiers included PPR-Meta, viralVerify, geNomad, and 3CAC. Figure 2 summarizes the results. Note that 3CAC utilizes either PPR-Meta or viralVerify to generate its classification. Therefore, we refer to the execution of 3CAC using viralVerify as 3CAC(vV) and using PPR-Meta as 3CAC(PM). Across the various datasets, 4CAC consistently achieved the highest F1 scores in classifying both viruses and plasmids, outperforming the other classifiers. The only exception was observed in the Sim_LF dataset, where 3CAC(vV)) exhibited a slightly higher F1 score than

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	Virus	Prokaryote	Eukaryote	Plasmid	Virus	(<1 kb)
Sim_SG	MiSeq	56	24	10	10	15 460	8112	1725	1275	5095
Sim_SF	MiSeq	3.5	1.5	10	10	50 546	44 378	1650	1256	56 735
Sim_LG	Nanopore	0.56	0.24	0.1	0.1	1575	148	193	202	125
Sim_LF	Nanopore	0.035	0.015	0.1	0.1	922	343	207	193	33
Sim_SG_lowANI	MiSeq	56	24	10	10	16 157	608	2476	870	3691
Sim_SF_lowANI	MiSeq	3.5	1.5	10	10	49 700	14 369	2422	853	36 658
Sim_LG_lowANI	Nanopore	0.56	0.24	0.1	0.1	879	32	200	118	36
Sim_LF_lowANI	Nanopore	0.035	0.015	0.1	0.1	640	158	178	125	4
Sim_SG_valid	MiSeq	56	24	10	10	12 378	880	2369	1480	3630
Sim_SF_valid	MiSeq	3.5	1.5	10	10	50 809	16 488	2316	1449	42 736
Sim_LG_valid	Nanopore	0.56	0.24	0.1	0.1	1392	74	256	170	104
Sim_LF_valid	Nanopore	0.035	0.015	0.1	0.1	1120	173	270	161	47
Sharon	HiSeq		106.3 in total			3097	533	87	21	1169
Tara	HiSeq		190.7 in total			16 156	31	153	1270	11 643
Oral_Nano	Nanopore		5.6 in total			9112	50	11	23	1888
Gut_HiFi	Pacbio HiFi		1.9 in total			4958	0	27	30	203

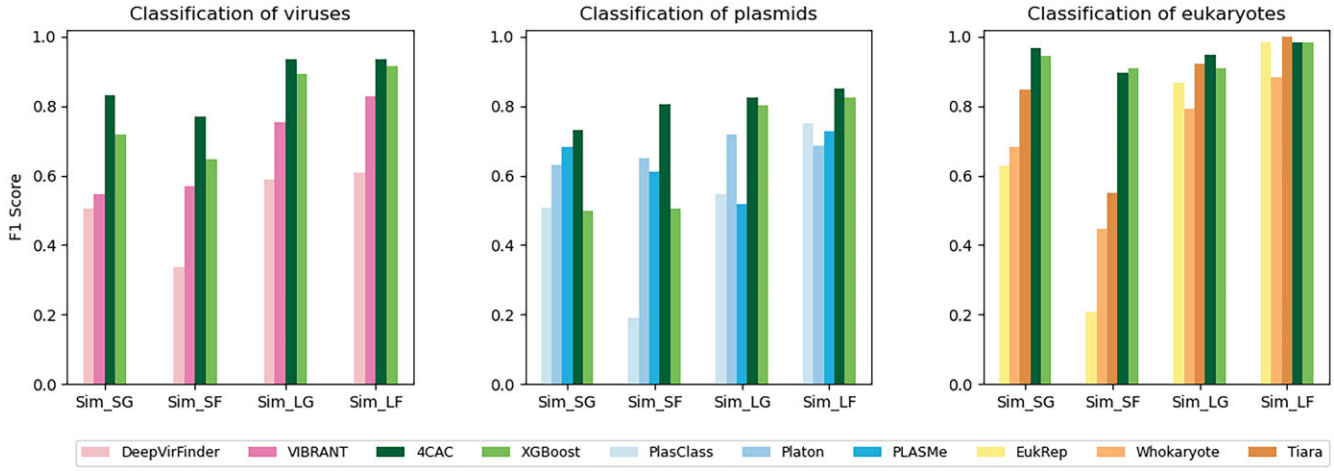


Figure 1. Performance of binary classifiers and 4CAC on simulated metagenomes. XGBoost represents the XGBoost classifier designed in this study without using graph information.

4CAC. 3CAC(vV) has better performance than 3CAC(PM) and performed as the second-best classifier in most tests as it also utilizes graph information to improve its classification. Among the stand-alone three-way classifiers, geNomad and PPR-meta had the highest F1 score in classifying viruses while viralVerify was the best in classifying plasmids in most tests. Supplementary Figure S7 summarizes the precision and recall of 4CAC and the comparing three-way classifiers. 4CAC achieves the best precision and recall comparable to the best across all the simulated datasets.

It is important to note that, in order to ensure a fair comparison, eukaryotic contigs were excluded from our benchmark of three-way classifiers. Similarly, only two classes of contigs were considered when benchmarking binary classifiers. Supplementary Figures S8 and S9 provide a comprehensive overview of the results when all contigs were included. As expected, the inclusion of all contigs led to a decline in the performance of both binary and three-way classifiers, as they tend to misclassify contigs that are not modeled. For example, eukaryotic contigs and plasmid contigs can be misclassified as viruses by viral classifiers, and eukaryotic contigs can be misclassified as plasmids or viruses by three-way classifiers, etc. This highlights the need for a four-way classifier that is able to identify viruses, plasmids, eukaryotes, and prokaryotes simultaneously from metagenome assemblies.

The tools benchmarked in this paper were run with their default training models. Each tool was trained using different features and potentially different training datasets. It is challenging, if not impossible, to ensure the same training dataset for tools trained on different features. To assess whether the use of different training datasets affects the performance of the benchmarked tools, we retrained PlasClass and DeepVirFinder using the same training dataset as 4CAC, as all three use *k*-mer composition as training features. Supplementary Figure S10 demonstrates that the retrained classifiers exhibit similar or slightly poorer performance than their default models, which were used throughout this study.

Utilizing existing algorithms to generate four-way classifications

We tested how effective existing binary and three-way classifiers are for four-way classification. Toward that goal, we combined existing classifiers to generate a four-way classification as follows. (i) The most straightforward idea is using VIBRANT and Platon to identify viruses and plasmids from

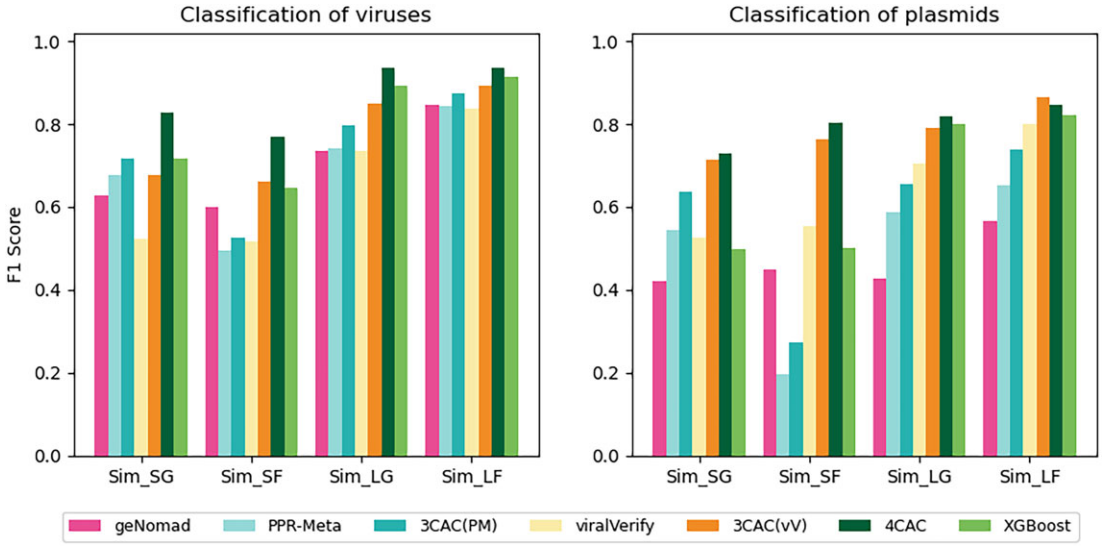


Figure 2. Performance of three-way classifiers and 4CAC on simulated metagenomes. XGBoost represents the XGBoost classifier designed in this study without using graph information.

metagenome assemblies. The remaining contigs are further classified as eukaryotes, prokaryotes, or uncertain by Tiara. We ran either VIBRANT or Platon first and selected the solution with a higher F1 score. This result is denoted by **Binary+**. Here, VIBRANT, Platon, and Tiara were selected because they performed best in binary classifications of viruses, plasmids, and eukaryotes from metagenome assemblies (shown in Figure 1). (ii) Comparing three-way classifiers to binary classifiers demonstrated that 3CAC(vV) outperformed all binary classifiers in classifying viruses and plasmids from metagenome assemblies (Figures 1 and 2). Therefore, we further combined 3CAC(vV) with Tiara in the following way. We first classified contigs by 3CAC(vV) and set aside these classified as plasmids and viruses, then used Tiara to classify the rest into eukaryotes, prokaryotes, or uncertain. We also repeated the process in the reverse order, running first Tiara and then 3CAC(vV). We then selected the solution with a higher F1 score. This result is denoted by **3CAC(vV)+Tiara**. DeepMicroClass, a five-way classifier, was published while this article was under review, and we included it in our benchmark as well. DeepMicroClass classifies contigs into prokaryotes, eukaryotes, plasmids, viruses infecting prokaryotic hosts and viruses infecting eukaryotic hosts. To facilitate comparison with 4CAC, the two latter classes were mapped to the single class of viruses.

We also wished to evaluate alignment-based methods, which match the contigs to a database of known genomes. For this purpose, we included Minimap2 (44) in our benchmark. Minimap2 aligns contigs to all genomes in the training dataset and classifies them by the best match. Contigs with good matches (>80% match along >80% of the contig) to multiple classes were classified as uncertain. To ensure fairness, reference genomes used for simulation were kept blind to Minimap2.

Figure 3 demonstrates that 4CAC outperformed DeepMicroClass and the combined classifiers in each classification across almost all datasets. In the long-read assembly Sim_LF, 3CAC(vV)+Tiara had a slightly higher F1 score than 4CAC in classifying plasmids and eukaryotes. Compared to the initial XGBoost classification, 4CAC consistently improved the F1 score across all tests, and the improvement was more substantial in classifying viruses and plasmids from short-read assemblies. Further analysis revealed that 19% and 58% of contigs in Sim_SG and Sim_SF respectively are shorter than 1kb (Table 1). These short contigs are often classified as uncertain in the initial classification. However, when considering the classification in the assembly graph, neighboring contigs that are long and confidently classified help in classifying these short contigs. Supplementary Figure S11 demonstrates that the graph refinement step of 4CAC dramatically improved recall while sacrificing a little bit of precision in classifying viruses and plasmids from short-read assemblies. Furthermore, DeepMicroClass achieved the highest recall but the lowest precision, resulting in very low F1 Scores. A possible reason is that DeepMicroClass tends to classify all contigs, whereas other classifiers designate a contig as uncertain when there is insufficient evidence for classification.

The performance of combined classifiers exhibits greater variability across diverse datasets. Not surprisingly, 3CAC(vV)+Tiara outperformed Binary+ in almost all the tests. Compared to combined classifiers, 4CAC improved the F1 score remarkably in classifying eukaryotes and prokaryotes from short-read assembly Sim_SF. This may be caused by a larger proportion of short contigs in Sim_SF (58% in Sim_SF versus 19% in Sim_SG. See Table 1). Short contigs are commonly unclassified by existing classifiers while 4CAC is able to classify most of them according to their neighboring long contigs in the assembly graph.

Figure 4 summarizes the total precision, recall, and F1 score of four-class classifiers. 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 DeepMicroClass and the combined classifiers in both precision and recall in all the simulated assemblies, while XGBoost was the second-best. 4CAC improved the recall remarkably in Sim_SF, due to a larger proportion of short contigs in it. Surprisingly, the XGBoost classifier itself, without using the graph information, had comparable or even better precision and recall than combined classifiers. Note that DeepMicroClass achieves the same precision and recall because it classifies all contigs.

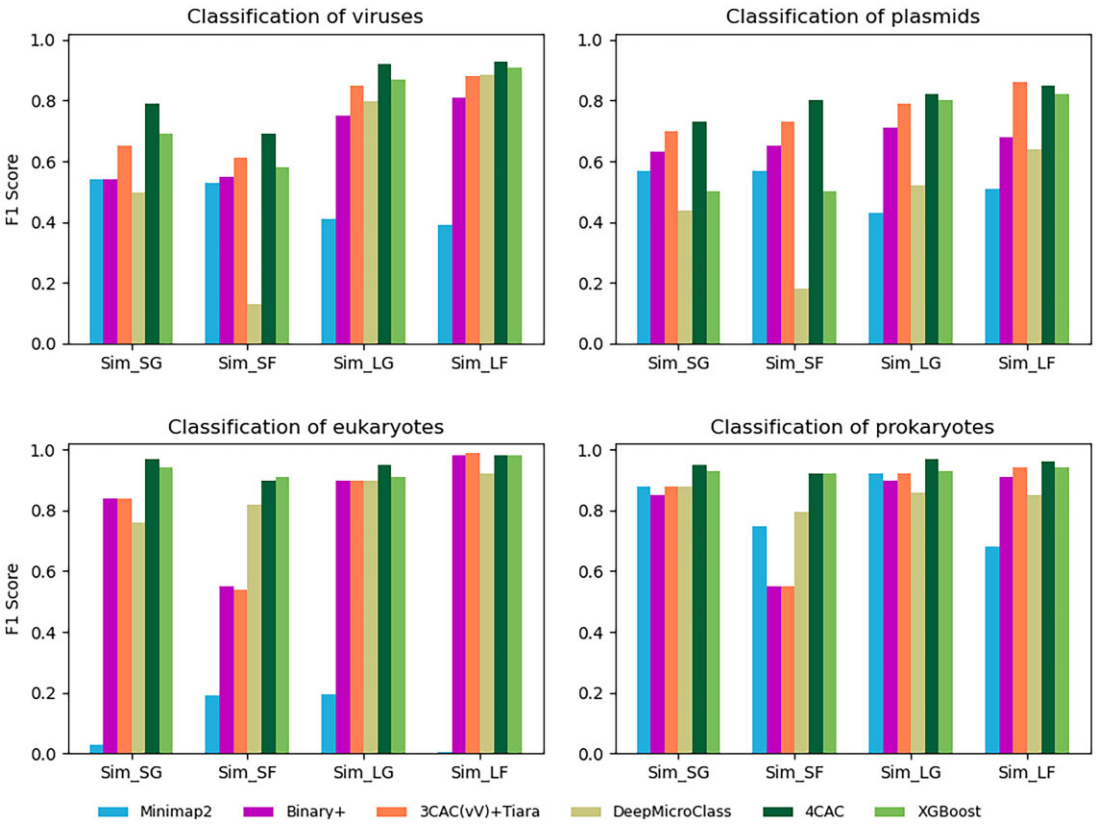


Figure 3. Performance of four-class classifiers on each class of simulated metagenomes. XGBoost represents the XGBoost classifier designed in this study without using graph information.

The alignment-based method Minimap2 had lower F1 scores than the machine learning-based methods, especially in classifying eukaryotes (Figure 3). A possible reason is that >50% of the test eukaryotes have little similarity to the training dataset (Supplementary Figure S1). Figure 4 and Supplementary Figure S11 reveal that while Minimap2 achieves the highest precision, it suffers from very low recall, resulting in poor F1 scores across all the simulated datasets.

To test the classifiers in identifying novel species, we used the four simulated datasets Sim_*_lowANI, where only genomes in the test set with low similarity to the training set of genomes were included. Here too, 4CAC consistently outperformed the combined classifiers (Supplementary Figure S12).

Performance on real metagenome samples

We additionally tested the performance of classifiers on four real complex metagenomic datasets: (i) short-read sequencing of 18 preterm infant fecal microbiome samples (NCBI accession number SRA052203), referred to as **Sharon** (45). (ii) Short-read sequencing of a microbiome sample from the Tara Oceans (NCBI accession number ERR868402), referred to as **Tara** (35). 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: (iii) Oxford Nanopore sequencing of two human saliva microbiome samples (NCBI accession number DRR214963 and DRR214965), referred to as **Oral_Nano** (46). (iv) Pacbio HiFi sequencing of a human gut microbiome sample (NCBI accession number SRR15275211), referred to as **Gut_HiFi**. 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, viruses, plasmids, and microeukaryotes from the NCBI GenBank database as reference genomes and mapped contigs to these reference genomes using Minimap2 (44). A contig was considered matched to a reference sequence if it had ≥80% mapping identity along ≥ 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 500 bp were not classified and excluded from the performance evaluation. Table 1 summarizes the properties of the datasets and the assemblies.

Since 3CAC(vV)+Tiara consistently outperformed the combination of binary classifiers (Figure 4), here we only compared 4CAC and its initial XGBoost classification to 3CAC(vV)+Tiara and DeepMicroClass. Similar to the result in simulated assemblies, Figure 5 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 most tests. In the Gut_HiFi dataset, 4CAC slightly improved the recall of XGBoost classification while sacrificing a little bit of precision, and resulted in a similar F1 score. On the short read datasets Sharon and Tara, in which microeukaryotes were previously identified (34,35), 4CAC achieved moderately better precision than 3CAC(vV)+Tiara but dramatically improved the recall. For example, 4CAC improved the recall from 0.54 to 0.87 in the Tara dataset. As a result, 4CAC had a substantially higher F1 score than

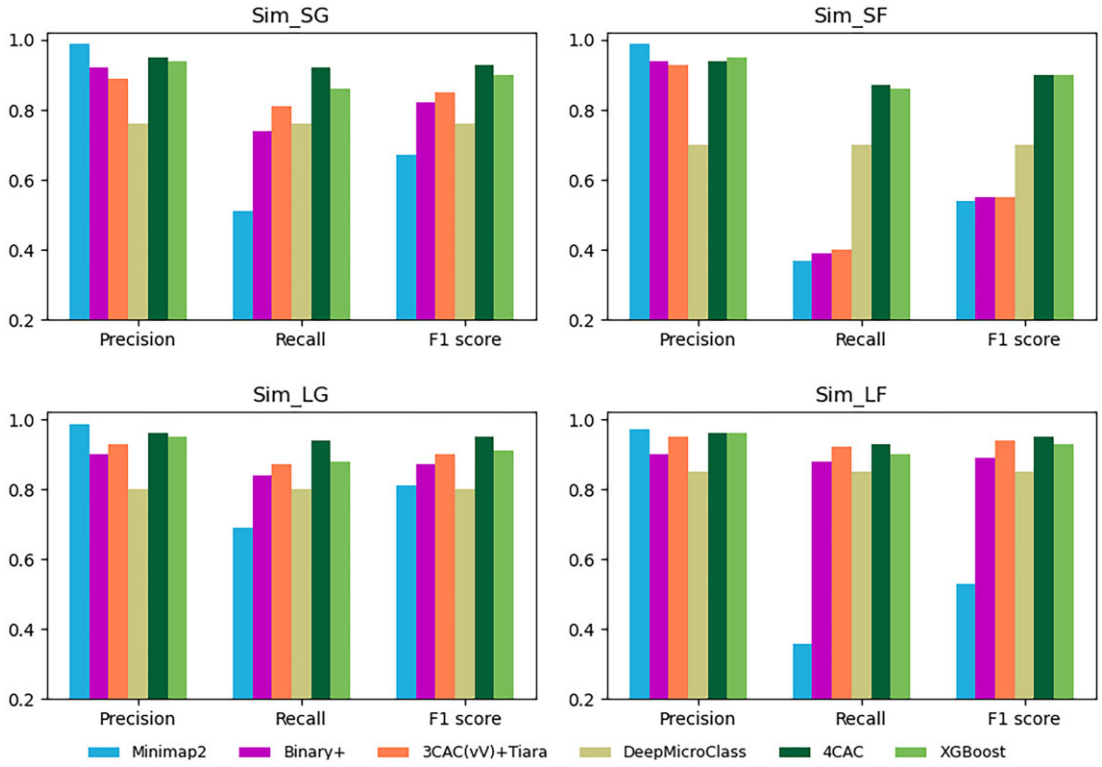


Figure 4. Performance of four-class classifiers on simulated metagenomes.

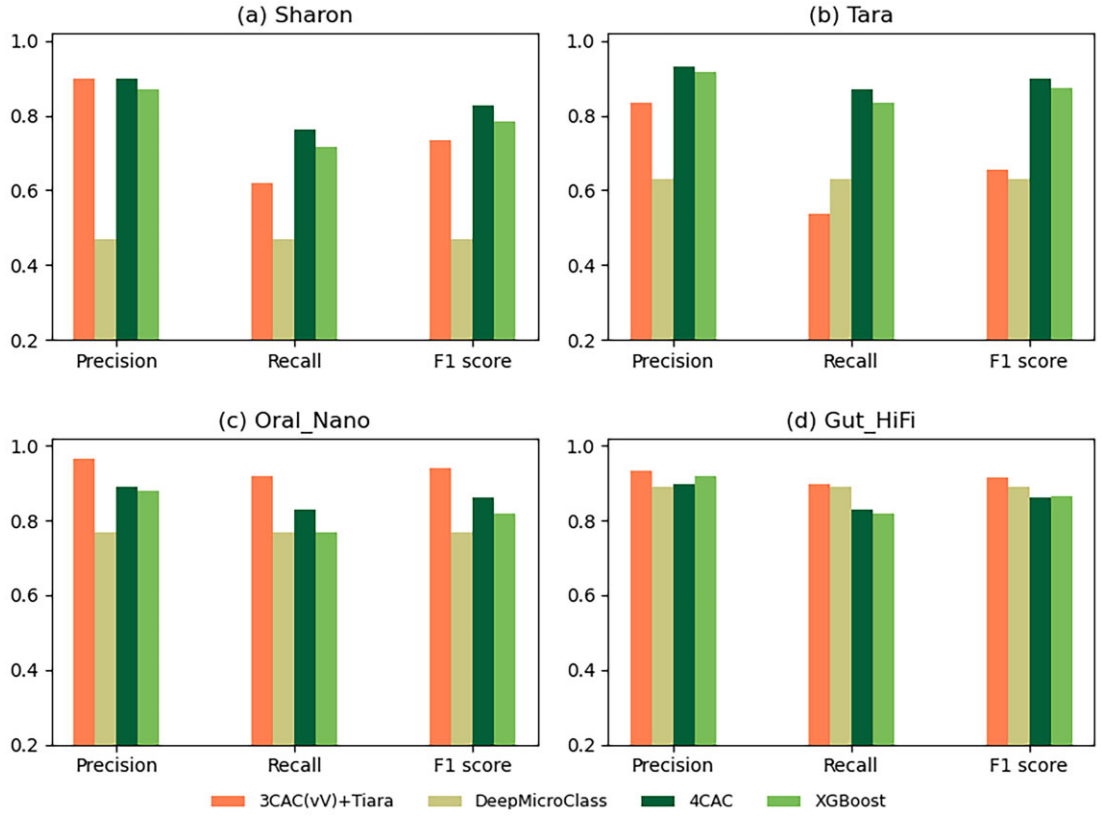


Figure 5. Performance of four-class classifiers on the real datasets. (**a**) Sharon and (**b**) Tara were assembled from short reads, (**c**) Oral_Nano and (**d**) Gut_HiFi were assembled from long reads.

3CAC(vV)+Tiara. DeepMicroClass performed the worst on these two datasets due to its very low precision.

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 6). The improvement is more significant in the classification of plasmids, which is consistent with the observation on simulated assemblies (30). Compared to 3CAC(vV)+Tiara, 4CAC had higher F1 scores in the classification of prokaryotes and eukaryotes, but a lower F1 score on viruses (Figure 6). A possible reason is that the proportion of viral contigs in the Sharon dataset is very small (0.6% versus ≥1.3% in simulated assemblies. See Table 1). In this extreme case, viralVerify, which is used in 3CAC(vV) and classifies contigs based on their gene content, achieved higher precision than the machine learning-based methods trained on composition information, such as PPR-Meta and the XGBoost classifier. DeepMicroClass attained comparable precision but lower recall in the classification of prokaryotes, leading to a lower F1 score. For viruses and eukaryotes, DeepMicroClass showed comparable recall, and for plasmids it had the highest recall but very low precision. Overall, it had lower F1 scores than the other classifiers.

On the two long-read datasets of human saliva and gut microbiome, 3CAC(vV)+Tiara outperformed 4CAC (Figures 5 (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).

We checked the real human metagenome datasets for possible host contamination. Although DNA was expected to be filtered out by sample preprocessing before sequencing these samples, it is possible that some human DNA remained. To address this concern, we mapped all reads in the three datasets originating from humans to the latest human reference genome (T2T-CHM13). 1.7% of reads in the Oral_Nano dataset had matches (>0.8 identities and >0.8 coverage) to the human genome, while no matching reads were found in Sharon and Gut_HiFi datasets. Furthermore, we mapped contigs classified as eukaryotes in the Oral_Nano dataset to the human genome. Four contigs classified as eukaryotes by 4CAC and 17 contigs classified as eukaryotes by Tiara had matches in the human genome. Hence, 4CAC appears more robust to human genome contamination.

Software and resource usage

Table 2 presents the runtime of the classifiers. 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 min. Due to the large runtime of viralVerify, geNomad, Platon, and VIBRANT, 4CAC is much faster than those other classifiers, which often require 1–2 orders of magnitudes more time. DeepMicroClass was on average a bit faster than 4CAC. Supplementary Table S2 summarizes the memory usage of the classifiers. Memory usage was the highest for geNomad in all the tests. All runs were performed on a 44-core, 2.2 GHz server with 792 GB of RAM. 4CAC is freely available via https://github.com/Shamir-Lab/4CAC.

Discussion and conclusion

We presented 4CAC, a classification algorithm for simultaneously identifying viruses, plasmids, prokaryotes, and microeukaryotes in metagenome assemblies. Evaluation on simulated and real metagenomic datasets demonstrated that 4CAC substantially outperformed existing classifiers in most tests. 4CAC also has a large speed advantage over the combined classifiers, running usually 1–2 orders of magnitude faster. DeepMicroClass had a slightly faster runtime than 4CAC but poorer performance on both simulated and real datasets.

In contrast to 3CAC, which necessitates the execution of viralVerify, PPR-meta, DeepVirFinder and PlasClass, 4CAC is a stand-alone algorithm, making it more user-friendly. To generate an initial classification with high precision, 3CAC reduces false positives from viralVerify and PPR-Meta by running DeepVirFinder and PlasClass on contigs classified as viruses and plasmids. In contrast, 4CAC generates an initial classification using its own XGBoost classifier and reduces false positives in minor classes by setting higher score thresholds.

Sequence classification methods fall into two main categories: reference-based and reference-free. Reference-based methods classify contigs by mapping them to a database of known genomes. This approach has several limitations: (a) short contigs are harder to classify as they may map to multiple references, leading to ambiguous or uncertain classifications; (b) reliance on a database of known genomes hinders the identification of novel species; (c) this strategy becomes resource-intensive as the reference database grows. 4CAC is a reference-free method, and it uses information on neighboring contigs in the assembly graph to assist the classification of short contigs. Evaluation on simulated datasets demonstrates that while the reference-based method Minimap2 has slightly better precision than 4CAC, 4CAC nearly doubles the recall of Minimap2, resulting in a much higher F1 score than Minimap2 (Figure 4 and Supplementary Figure S11). Moreover, Minimap2 is much slower than 4CAC. For the simulated dataset Sim_SG, 4CAC took 6.9 min, whereas Minimap2 took 310 min.

Machine learning-based classifiers often assign scores to predictions, indicating their confidence. However, these scores do not reflect the true probabilities of the predictions. Indeed, when we attempted training XGBoost classifiers on class-imbalanced datasets using a default score threshold of 0.5 for all classes, results were unsatisfactory. By setting different probability thresholds for different classes, we obtained a good trade-off between precision and recall. We set the threshold at 0.95 for viral and plasmid classification based on the observation of class imbalance in metagenome assemblies. Note, however, that when applying the same classifier to samples with varying class compositions, the results may exhibit significantly different false positive rates, and this is true for 4CAC as well. 4CAC is specifically designed for metagenome assemblies, where the proportion of viral and plasmid contigs is typically low compared to prokaryotic contigs.

On two real datasets assembled from long reads, where the relative abundance of viruses, plasmids and eukaryotes was extremely low (<0.6% compared to over 1.3% in other assemblies), the combined classifier 3CAC(vV)+Tiara outperformed 4CAC. This difference in performance could potentially be attributed to the tendency of classifiers trained on *k*-mer compositions to yield a higher false positive rate com-

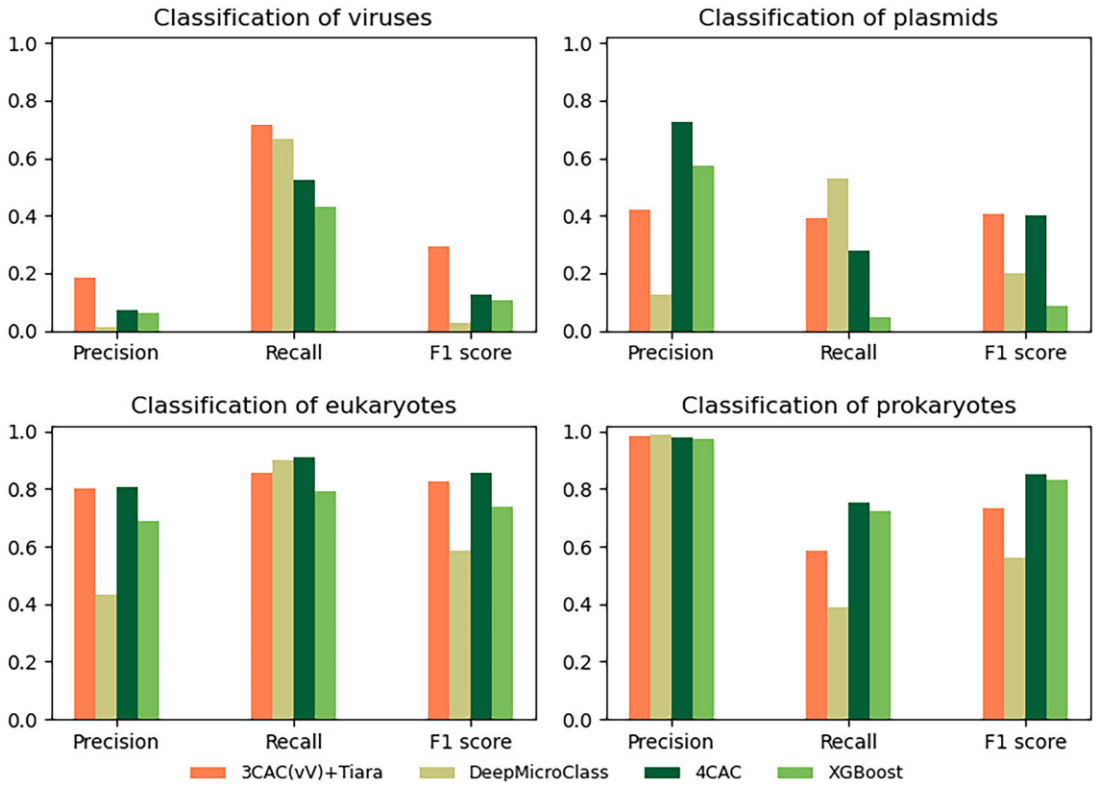


Figure 6. Performance on each class for real short read dataset Sharon.

Table 2. Runtime of the tested classifiers

	4CAC	DeepMC	viralV	PPR-M	geNomad	Tiara	PlasClass	Platon	DeepVF	VIBRANT
Sim_SG	6.9	4.2	322	60.4	106.9	2.9	4.2	241.5	91.2	185.2
Sim_SF	4.8	5.1	175	25.1	154.8	2.8	3.9	643.1	62.1	82.1
Sim_LG	3.7	1.3	185.4	33.4	92.8	1.3	2.3	22	33.2	139.8
Sim_LF	1.4	0.8	77.5	14.9	27.1	0.7	1.2	22	14.7	53.1
Sharon	1.4	3	29.9	7.3	16.4	0.5	1.2	49.4	16.3	25.3
Tara	14.3	11.8	221.1	94.7	155.8	4.4	11.2	638.9	92.3	50.6
Oral_Nano	8.2	3.1	452.5	84.4	140.1	3.5	5.6	301.1	88.6	201.1
Gut_HiFi	9.3	4.9	677.8	124	252.7	4.8	8.1	444.6	109.8	426.1

Runtime is measured by wall clock time in minutes. DeepMC, ViralV, PPR-M, and DeepVF represent classifiers DeepMicroClass, viralVerify, PPR-Meta, and DeepVirFinder respectively. The binary classifier PLASMe was only run on the four simulated datasets, with an average runtime of 2.6 min for short-read assemblies and 15 min for long-read assemblies.

pared to classifiers trained on the gene content of contigs. It is important to mention that these results may be biased by the underrepresentation of these classes in genomic databases. Given the current knowledge about species in metagenomes, we recommend using 4CAC on short reads and on host-filtered long read samples. For generic long read samples, where prokaryotes constitute the majority, we suggest utilizing 3CAC(vV) followed by Tiara for optimal results.

The implementation of the correction and propagation steps on the assembly graph yielded substantial improvements in the classification of short contigs. As anticipated, the combined classifier 3CAC(vV)+Tiara demonstrated the second-best performance across all tests since 3CAC utilizes similar refinement procedures.

Our study has several limitations. First, as mentioned above, performance is affected by the relative abundance of the different classes in the input data. Second, the refinement step in 4CAC may misclassify some sequences, especially those that underwent horizontal gene transfer across classes, e.g. proviruses and integrated plasmids. However, as we have shown, that step improves overall performance. Furthermore, to detect prophages from metagenome assemblies, tools designed specifically for this purpose, such as Prophage Hunter (47), and PHASTEST (48), would be better choices. In future work, we aim to incorporate factors such as contig coverage and length to enhance the identification of proviruses. Third, 4CAC does not categorize contigs at various taxonomic levels such as genus and species. Taxonomic classification requires different tools and approaches that are specifically designed for that goal, such as Kraken2 (5) and MetaPhlAn4 (49).

Finally, there is a chance of leakage between the training and test sets in case very similar sequences reside in both. Our additional tests on simulated datasets that included only species with low similarity to the training set species confirmed the advantage of 4CAC. The partition into training

and test sets by GenBank release date is a common practice, which was also adopted in most of the classifiers that we evaluated [e.g. (21,26,30)]. Furthermore, it also gives a realistic performance estimate, since when a method is applied to a new sample, some of the sequences encountered are likely to have highly similar counterparts in the database. Still, the evaluation of classifiers on real metagenome assemblies remains challenging due to the lack of ground truth. The database of known reference genomes for bacteria is much larger than that of phages, plasmids, and microeukaryotes, which may lead to bias in evaluating the results. The performance of 4CAC and other classifiers may be underestimated due to the presence of novel species and of contigs that are too short to match confidently to the reference genomes.

Data availability

4CAC is implemented in Python and is available on GitHub (https://github.com/Shamir-Lab/4CAC) and Zenodo. DOI: 10.5281/zenodo.13383932. 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.

Supplementary data

Supplementary Data are available at NAR Online.

Acknowledgements

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

Author contributions: L.P. and R.S. designed the study and evaluated the results. L.P. developed and implemented the 4CAC algorithm, analyzed the datasets and wrote the manuscript. R.S. oversaw the development of the project and reviewed the manuscript. Both authors read and approved the final manuscript.

Funding

Israel Science Foundation [1339/18 and 2206/22, in part]; 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. Funding for open access charge: Israel Science Foundation [1339/18 and 2206/22].

Conflict of interest statement

None declared.

References

1. Marcelino,V.R., Clausen,P.T., Buchmann,J.P., Wille,M., Iredell,J.R., Meyer,W., Lund,O., Sorrell,T.C. and Holmes,E.C. (2020) CCMetagen: comprehensive and accurate identification of eukaryotes and prokaryotes in metagenomic data. *Genome Biol.*, **21**, 103.
2. McKenney,P.T. and Pamer,E.G. (2015) From hype to hope: the gut microbiota in enteric infectious disease. *Cell*, **163**, 1326–1332.
3. Gilbert,J.A., Blaser,M.J., Caporaso,J.G., Jansson,J.K., Lynch,S.V. and Knight,R. (2018) Current understanding of the human microbiome. *Nat. Med.*, **24**, 392–400.
4. Wood,D.E. and Salzberg,S.L. (2014) Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.*, **15**, R46.
5. Wood,D.E., Lu,J. and Langmead,B. (2019) Improved metagenomic analysis with Kraken 2. *Genome Biol.*, **20**, 257.
6. Mallawaarachchi,V., Wickramarachchi,A. and Lin,Y. (2020) GraphBin: refined binning of metagenomic contigs using assembly graphs. *Bioinformatics*, **36**, 3307–3313.
7. Mallawaarachchi,V. and Lin,Y. (2022) Accurate binning of metagenomic contigs using composition, coverage, and assembly graphs. *J. Comput. Biol.*, **29**, 1357–1376.
8. Kang,D.D., Li,F., Kirton,E., Thomas,A., Egan,R., An,H. and Wang,Z. (2019) MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ*, **7**, e7359.
9. Wu,Y.-W., Simmons,B.A. and Singer,S.W. (2016) MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics*, **32**, 605–607.
10. Brooks,B., Olm,M.R., Firek,B.A., Baker,R., Thomas,B.C., Morowitz,M.J. and Banfield,J.F. (2017) Strain-resolved analysis of hospital rooms and infants reveals overlap between the human and room microbiome. *Nat. Commun.*, **8**, 1814.
11. Liang,Z., Dong,C., Liang,H., Zhen,Y., Zhou,R., Han,Y. and Liang,Z. (2022) A microbiome study reveals the potential relationship between the bacterial diversity of a gymnastics hall and human health. *Sci. Rep.*, **12**, 5663.
12. Moss,E.L., Maghini,D.G. and Bhatt,A.S. (2020) Complete, closed bacterial genomes from microbiomes using nanopore sequencing. *Nat. Biotechnol.*, **38**, 701–707.
13. Benler,S., Yutin,N., Antipov,D., Rayko,M., Shmakov,S., Gussow,A.B., Pevzner,P. and Koonin,E.V. (2021) Thousands of previously unknown phages discovered in whole-community human gut metagenomes. *Microbiome*, **9**, 78.
14. Lind,A.L. and Pollard,K.S. (2021) Accurate and sensitive detection of microbial eukaryotes from whole metagenome shotgun sequencing. *Microbiome*, **9**, 58.
15. Calero-Cáceres,W., Ye,M. and Balcázar,J.L. (2019) Bacteriophages as environmental reservoirs of antibiotic resistance. *Trends Microbiol.*, **27**, 570–577.
16. Wein,T., Hülter,N., Mizrahi,I. and Dagan,T. (2019) Emergence of plasmid stability under non-selective conditions maintains antibiotic resistance. *Nat. Commun.*, **10**, 2595.
17. Lopatkin,A., Meredith,H., Srimani,J., Pfeiffer,C., Durrett,R. and You,L. (2017) Persistence and reversal of plasmid-mediated antibiotic resistance. *Nat. Commun.*, **8**, 1689.
18. Sitaraman,R. (2018) Prokaryotic horizontal gene transfer within the human holobiont: ecological-evolutionary inferences, implications and possibilities. *Microbiome*, **6**, 163.
19. Olm,M.R., West,P.T., Brooks,B., Firek,B.A., Baker,R., Morowitz,M.J. and Banfield,J.F. (2019) Genome-resolved metagenomics of eukaryotic populations during early colonization of premature infants and in hospital rooms. *Microbiome*, **7**, 26.
20. 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.* (2021) VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. *Microbiome*, **9**, 37.
21. Ren,J., Song,K., Deng,C., Ahlgren,N.A., Fuhrman,J.A., Li,Y., Xie,X., Poplin,R. and Sun,F. (2020) Identifying viruses from metagenomic data using deep learning. *Quant. Biol.*, **8**, 64–77.
22. Kieft,K., Zhou,Z. and Anantharaman,K. (2020) VIBRANT: automated recovery, annotation and curation of microbial viruses,

and evaluation of viral community function from genomic sequences. *Microbiome*, **8**, 90.
23. Roux,S., Enault,F., Hurwitz,B.L. and Sullivan,M.B. (2015) VirSorter: mining viral signal from microbial genomic data. *PeerJ*, **3**, e985.
24. Auslander,N., Gussow,A.B., Benler,S., Wolf,Y.I. and Koonin,E.V. (2020) Seeker: alignment-free identification of bacteriophage genomes by deep learning. *Nucleic Acids Res.*, **48**, e121.
25. Krawczyk,P.S., Lipinski,L. and Dziembowski,A. (2018) PlasFlow: predicting plasmid sequences in metagenomic data using genome signatures. *Nucleic Acids Res.*, **46**, e35.
26. Pellow,D., Mizrahi,I. and Shamir,R. (2020) PlasClass improves plasmid sequence classification. *PLoS Comput. Biol.*, **16**, e1007781.
27. Andreopoulos,W.B., Geller,A.M., Lucke,M., Balewski,J., Clum,A., Ivanova,N.N. and Levy,A. (2022) Deeplasmid: deep learning accurately separates plasmids from bacterial chromosomes. *Nucleic Acids Res.*, **50**, e17.
28. Tang,X., Shang,J., Ji,Y. and Sun,Y. (2023) PLASMe: a tool to identify PLASMid contigs from short-read assemblies using transformer. *Nucleic Acids Res.*, **51**, e83.
29. Schwengers,O., Barth,P., Falgenhauer,L., Hain,T., Chakraborty,T. and Goesmann,A. (2020) Platon: identification and characterization of bacterial plasmid contigs in short-read draft assemblies exploiting protein sequence-based replicon distribution scores. *Microbial. Genom.*, **6**, e000398.
30. Fang,Z., Tan,J., Wu,S., Li,M., Xu,C., Xie,Z. and Zhu,H. (2019) PPR-Meta: a tool for identifying phages and plasmids from metagenomic fragments using deep learning. *GigaScience*, **8**, giz066.
31. Antipov,D., Raiko,M., Lapidus,A. and Pevzner,P.A. (2020) Metaviral SPAdes: assembly of viruses from metagenomic data. *Bioinformatics*, **36**, 4126–4129.
32. Pu,L. and Shamir,R. (2022) 3CAC: improving the classification of phages and plasmids in metagenomic assemblies using assembly graphs. *Bioinformatics*, **38**, ii56–ii61.
33. Camargo,A.P., Roux,S., Schulz,F., Babinski,M., Xu,Y., Hu,B., Chain,P.S., Nayfach,S. and Kyrpides,N.C. (2023) Identification of mobile genetic elements with geNomad. *Nat. Biotechnol.*, **42**, 1303–1312.
34. West,P.T., Probst,A.J., Grigoriev,I.V., Thomas,B.C. and Banfield,J.F. (2018) Genome-reconstruction for eukaryotes from complex natural microbial communities. *Genome Res.*, **28**, 569–580.
35. Karlicki,M., Antonowicz,S. and Karnkowska,A. (2022) Tiara: deep learning-based classification system for eukaryotic sequences. *Bioinformatics*, **38**, 344–350.
36. Pronk,L.J. and Medema,M.H. (2022) Whokaryote: distinguishing eukaryotic and prokaryotic contigs in metagenomes based on gene structure. *Microbial. Genom.*, **8**, 000823.
37. Hou,S., Tang,T., Cheng,S., Liu,Y., Xia,T., Chen,T., Fuhrman,J.A. and Sun,F. (2024) DeepMicroClass sorts metagenomic contigs into prokaryotes, eukaryotes and viruses. *NAR Genom. Bioinform.*, **6**, lqae044.
38. Ren,J., Ahlgren,N.A., Lu,Y.Y., Fuhrman,J.A. and Sun,F. (2017) VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, **5**, 69.
39. Nurk,S., Meleshko,D., Korobeynikov,A. and Pevzner,P.A. (2017) metaSPAdes: a new versatile metagenomic assembler. *Genome Res.*, **27**, 824–834.
40. 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.* (2020) metaFlye: scalable long-read metagenome assembly using repeat graphs. *Nat. Methods*, **17**, 1103–1110.
41. Gourlé,H., Karlsson-Lindsjö,O., Hayer,J. and Bongcam-Rudloff,E. (2019) Simulating Illumina metagenomic data with InSilicoSeq. *Bioinformatics*, **35**, 521–522.
42. Yang,C., Chu,J., Warren,R.L. and Birol,I. (2017) NanoSim: nanopore sequence read simulator based on statistical characterization. *GigaScience*, **6**, gix010.
43. Mikheenko,A., Saveliev,V. and Gurevich,A. (2016) MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*, **32**, 1088–1090.
44. Li,H. (2018) Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**, 3094–3100.
45. Sharon,I., Morowitz,M.J., Thomas,B.C., Costello,E.K., Relman,D.A. and Banfield,J.F. (2013) Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization. *Genome Res.*, **23**, 111–120.
46. Yahara,K., Suzuki,M., Hirabayashi,A., Suda,W., Hattori,M., Suzuki,Y. and Okazaki,Y. (2021) Long-read metagenomics using PromethION uncovers oral bacteriophages and their interaction with host bacteria. *Nat. Commun.*, **12**, 27.
47. Song,W., Sun,H.-X., Zhang,C., Cheng,L., Peng,Y., Deng,Z., Wang,D., Wang,Y., Hu,M., Liu,W., *et al.* (2019) Prophage Hunter: an integrative hunting tool for active prophages. *Nucleic Acids Res.*, **47**, W74–W80.
48. Wishart,D.S., Han,S., Saha,S., Oler,E., Peters,H., Grant,J.R., Stothard,P. and Gautam,V. (2023) PHASTEST: faster than PHASTER, better than PHAST. *Nucleic Acids Res.*, **51**, W443–W450.
49. Blanco-Míguez,A., Beghini,F., Cumbo,F., McIver,L.J., Thompson,K.N., Zolfo,M., Manghi,P., Dubois,L., Huang,K.D., Thomas,A.M., *et al.* (2023) Extending and improving metagenomic taxonomic profiling with uncharacterized species using MetaPhlAn 4. *Nat. Biotechnol.*, **41**, 1633–1644.

Received: December 21, 2023. **Revised:** July 13, 2024. **Editorial Decision:** August 26, 2024. **Accepted:** September 2, 2024