4CAC: 4-class classification of metagenome assemblies using machine learning and assembly graphs

Lianrong Pu and Ron Shamir

Blavatnik School of Computer Science, Tel Aviv University
lianrongpu@mail.tau.ac.il, rshamir@tau.ac.il

Abstract. 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 from 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.

Here, 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 of 4CAC against existing classifiers on simulated and real metagenome datasets demonstrates that 4CAC substantially outperforms existing classifiers on short reads. On long reads, it shows an advantage unless the abundance of the minor classes is very low. It is also by far the fastest. The 4CAC software is available at https://github.com/Shamir-Lab/4CAC.

1 Introduction

Microbial communities in natural and host-associated environments are often dominated by bacteria and coinhabited by fungi, protozoa, archaea, plasmids, and phages [22]. Changes in microbiome diversity, function, and density have been linked to a variety of disorders in many organisms [23, 8]. As the dominant group of species in microbial communities, bacteria have been widely studied. Taxonomic classification tools [39, 38] and metagenome binning tools [21, 20, 11, 40] 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 [5, 17, 24]. In contrast, our understanding of plasmids, phages, and microbial eukaryotes largely lags behind, due to their lower abundance in microbial communities [4, 18]. Recent studies revealed that phages and plasmids in microbial communities play important roles in horizontal gene transfer events and antibiotic resistance [6, 36, 19, 35], and microbial eukaryotes have complex interaction with their hosts in both plant- and animal-associated microbiomes [18, 26]. 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 plasmids from prokaryotes (bacteria and archaea) in microbial communities. VirSorter [33], DeepVirFinder [32], VIBRANT [13], and many other phage classifiers [10, 3] were designed to separate phages from prokaryotes in metagenome assemblies. Plasmid classifiers, such as cBar [43], PlasFlow [15], PlasClass [27], and Deeplasmid [1], 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 [7], viralVerify [2], and 3CAC [30], were proposed to simultaneously identify phages, plasmids, and prokaryotes from metagenome assemblies. On the other hand, 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 [37], Tiara [12], and Whokaryote [29] were proposed to distinguish microeukaryotes from prokaryotes in metagenome assemblies. However, even though prokaryotes, microeukaryotes, phages, and plasmids are present in most microbial communities, 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 [32, 30].

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 for different sequence lengths. The XGBoost classifier outputs four scores for each contig to indicate its probability 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 classified with lower confidence by the initial XGBoost classifier. Evaluation of 4CAC against existing classifiers on simulated and real metagenome datasets demonstrates that 4CAC substantially outperforms existing classifiers on short reads. On long reads, it also shows an advantage unless the abundance of the minor classes is very low.

2 Methods

4CAC accepts as input a set of contigs and the associated assembly graph, and aims to classify each contig in the input as phage, 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.

2.1 Design and implementation of the XGBoost classifier

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 [7, 27, 31], 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 of sequences with 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 to indicate its probabilities of being classified as phage, plasmid, prokaryote, or microeukaryote.

Existing algorithms [7, 27, 31] 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 minor classes (phage, plasmid, and microeukaryote). Taking into consideration the class imbalance in metagenome assemblies, here we chose to set different thresholds for different classes (0.95 for phage and plasmid, 0.5 for microeukaryote by default). This results in high precision for the classification of each class while maintaining high recall for the classification of prokaryotes.

2.2 Refining the classification using the assembly graph

To understand the species presented in a microbial community, the common practice is to first assemble the sequencing reads into longer sequences called *contigs*, and then classify these assembled contigs into classes. Broadly used metagenome assemblers, such as metaSPAdes [25] and metaFlye [14], use assembly graphs to combine overlapped reads (or k-mers) into contigs. Nodes in an assembly graph represent contigs and edges represent subsequence overlaps between the corresponding contigs. In our description below, the neighbors of a contig are its adjacent nodes in the assembly graph. Most of the existing classifiers take contigs as input and classify each of them independently based on its 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 be used to improve the classification [30]. Therefore, here too we exploit the assembly graph to improve the initial classification by the following two steps.

(1) Correction of classified contigs. A classified contig c is called *incongruous* if it has at least two classified neighbors and all of them belong to the same class, while c belongs to a different class. We reason that an incongruous contig was likely wrongly classified and its classification needs to be corrected to be consistent with its classified neighbors. Therefore, 4CAC scans all the incongruous contigs in decreasing order of the number of their classified neighbors and corrects the classification of each incongruous contig to match its classified neighbors. Note that once an incongruous contig is corrected, this contig and all its classified neighbors are not incongruous and will not be corrected again.

(2) Propagation to unclassified contigs. An unclassified contig is called *implied* if it has one or more classified neighbors and all of them belong to the same class. 4CAC dynamically maintains a list of implied contigs sorted in decreasing order of the number of their classified neighbors. At each iteration, 4CAC classifies the first implied contig c in the list according to its classified neighbors, and then removes c and updates the sorted list. Note that only the unclassified neighbors of c need to be updated at that iteration. We repeat this step until the list is empty.

3 Experimental Setup

3.1 Datasets and Tools

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 with 80% host (prokaryotes and microeukaryotes) proportion. As a *filtered metagenome* scenario, where reads from large host genomes are filtered and thus plasmids and viruses are enriched, we simulated reads with 20% host proportion. For both scenarios, the proportions of phages and plasmids were set to be the same, 10% each in the generic scenario and 40% each in the filtered scenario. The relative abundance of genomes within each class was done as in [27]. Short reads were simulated from the genome sequences using InSilicoSeq [9] and assembled by metaSPAdes. Long reads were simulated from the genome sequences using NanoSim [42] and assembled by metaFlye. Full details on the simulation and the assembly are provided in the Supplementary file. We denote by **Sim_AN** the simulation with A=S for short reads and A=L for long reads, and N is the proportion of host reads. For example, Sim_S20 is the short read filtered scenario. Table 1 presents a summary of the simulated metagenome assemblies.

Real Datasets. We 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** [34]. (2) Short-read sequencing of a microbiome sample from the Tara Oceans (NCBI accession number ERR868402), referred to as **Tara** [12]. 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** [28]. 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 [16]. 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.

Tools Used. As there are currently no four-class classifiers that can be compared with 4CAC, we combined 3-class classifiers viralVerify, PPR-Meta, and 3CAC with eukaryote classifiers Tiara and Whokaryote as follows. We first classified the input contigs into phages, plasmids, chromosomes, and uncertain by the three-class classifier and then classified contigs in the chromosome or uncertain class into prokaryotes or microeukaryotes by the eukaryote classifier. We also tried first classifying contigs into eukaryotes, prokaryotes, and uncertain and then classifying contigs in the prokaryote or uncertain class into three classes. The better result was used to represent each combination. 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+B** when A is the 3-class classifier and B is the eukaryote classifier. In our benchmark, viralVerify was run with '-p' option to enable three-class classification. PPR-Meta was run with a score threshold of 0.5 to assure reliable prediction.

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

Dataset	Read type	Number of reads(M)				Number of assembled contigs				Short contigs
		prokaryote	eukaryote	plasmid	phage	prokaryote	eukaryote	plasmid	phage	(< 1kb)
Sim_S80	MiSeq	56	24	10	10	15,460	8,112	1,725	1,275	5,095
Sim_S20	MiSeq	3.5	1.5	10	10	50,546	44,378	1,650	1,256	56,735
Sim_L80	Nanopore	0.56	0.24	0.1	0.1	1,575	148	193	202	125
Sim_L20	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

3.2 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, which can be calculated as: $F1\ score = (2 * precision * recall)/(precision + recall)$.

Following [27, 7], 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 specifically for phage, plasmid, prokaryote, and microeukaryote 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.

4 Results

We benchmarked the performance of 4CAC against combined classifiers using both simulated and real metagenome assemblies of long and short reads. Tests on eukaryote classifiers showed that Tiara always outperforms Whokaryote (Supplementary Table S1), thus we report here only on combined classifiers using Tiara.

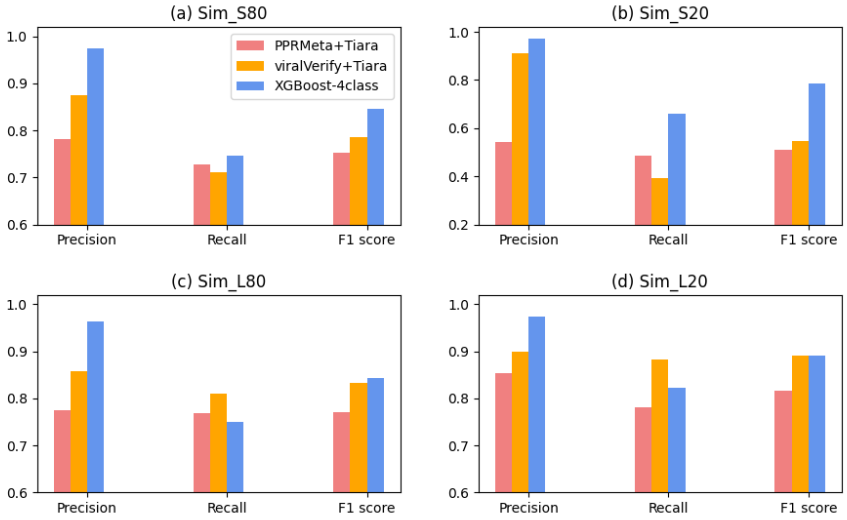

Fig. 1. Performance of four-class classifiers on simulated metagenome assemblies without using graph information. Sim_S80 and Sim_S20 are assembled from short reads while Sim_L80 and Sim_L20 are assembled from long reads.

4.1 Performance on simulated metagenome assemblies

We first benchmarked our XGBoost four-class classifier (without using the graph information) against viralVerify+Tiara and PPR-Meta+Tiara on the simulated datasets. Figure 1 shows that our XGBoost classifier performs better than the combined classifiers in both precision and recall in short read assemblies. In the long read assemblies, viralVerify+Tiara and the XGBoost classifier have similar F1 scores with the XGBoost classifier achieving better precision and viralVerify+Tiara achieving better recall. The specific precision, recall, and F1

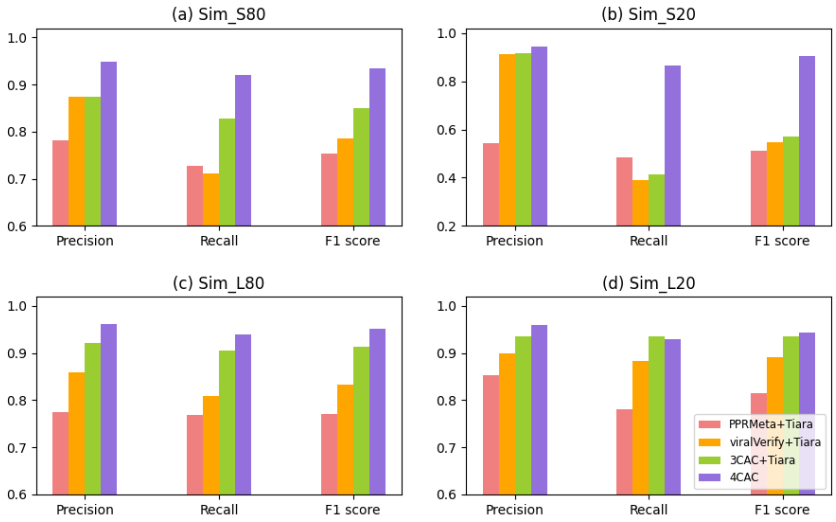

Fig. 2. Performance of four-class classifiers on simulated metagenome assemblies. Sim_S80 and Sim_S20 are assembled from short reads while Sim_L80 and Sim_L20 are assembled from long reads.

score for phage, plasmid, prokaryote, and eukaryote classification can be found in Supplementary Table S2. Consistent with our expectation, the XGBoost classifier always achieves better precision and F1 score in minor classes, such as phages, plasmids, and eukaryotes.

We then tested the full 4CAC algorithm, including the correction and propagation steps. Figure 2 demonstrates that 4CAC outperforms existing classifiers in both precision and recall in all the simulated assemblies. Not surprisingly, 3CAC+Tiara achieves the second-best performance. The improvement of 4CAC is more substantial in short-read assemblies, which may be because 3CAC already performs well in long-read assemblies. 4CAC improves the recall remarkably in Sim_S20 due to a larger proportion of short contigs in it (58% in Sim_S20 vs. 19% in Sim_S80. See Table 1). Lower sequencing depth in Sim_S20 results in a much more fragmented assembly. Figure 3 presents the performance of the tested classifiers for each class in Sim_S80. 4CAC outperforms existing classifiers in the F1 score in the classification of each class. 4CAC always achieves better precision in the classification of minor classes and slightly lower precision but higher recall in the classification of prokaryotes. Supplementary Figures S1, S2, and S3 present similar results in the other simulated assemblies.

4.2 Performance on real microbiome samples

We tested 4CAC and the combined classifiers on the short read datasets Sharon and Tara, in which microeukaryotes were previously identified [37, 12]. Figures 4 (a) and (b) show that 4CAC achieves moderately better precision than the best combined classifier and dramatically improves the recall. For example, 4CAC improves the recall from 0.62 to 0.87 in the Tara dataset. As a result, 4CAC substantially outperforms existing classifiers in the F1 score.

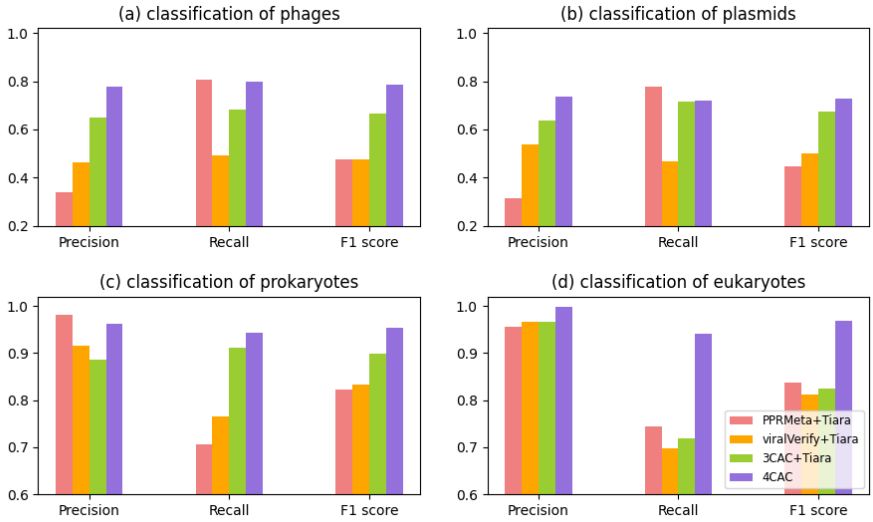

Fig. 3. Performance on each class for simulated short read dataset Sim_S80.

Further analysis reveals that 4CAC has higher F1 scores in the classification of prokaryotes and eukaryotes, but lower F1 scores than 3CAC+Tiara and viralVerify+Tiara 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 classifies contigs based on their gene content, achieves higher precision than the machine learning-based methods, such as PPR-Meta and the XGBoost classifier.

We also tested the methods on two long read datasets of human saliva and gut microbiome. Figures 4 (c) and (d) show that 3CAC+Tiara outperforms 4CAC. 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 outperform three-class classifiers (Supplementary Table S3).

4.3 Software and resource usage

Table 2 presents the runtime and memory usage of the classifiers. For 3CAC we report the runtimes 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 runtimes of viralVerify and PPR-Meta, 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 needed was largest for 4CAC in the short read assemblies and for PPR-Meta in the long read assemblies. All runs were done on a 44-core, 2.2GHz server with 792GB of RAM. 4CAC is freely available via https://github.com/Shamir-Lab/4CAC.

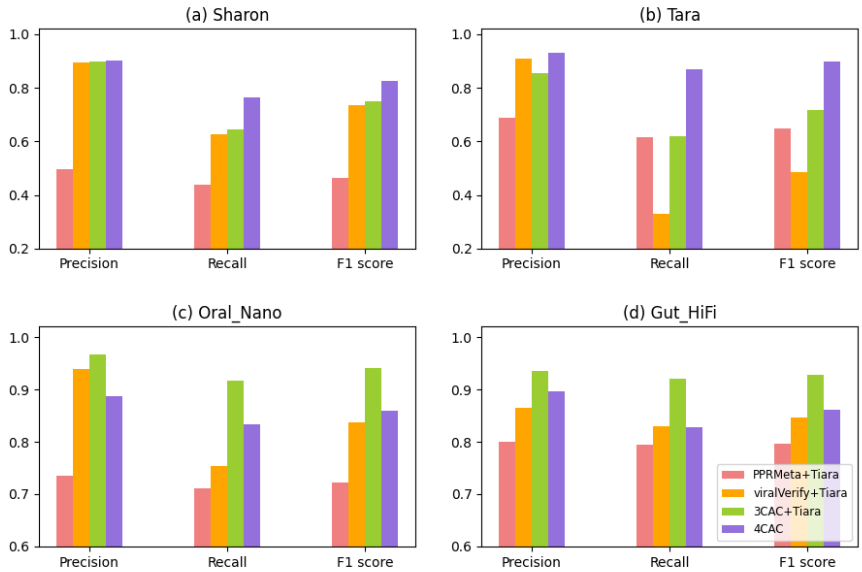

Fig. 4. Performance of four-class classifiers on the real datasets. (a) Sharon and (b) Tara were assembled from short reads, (c) Oral_Nano and (d) Gut_HiFi were assembled from long reads.

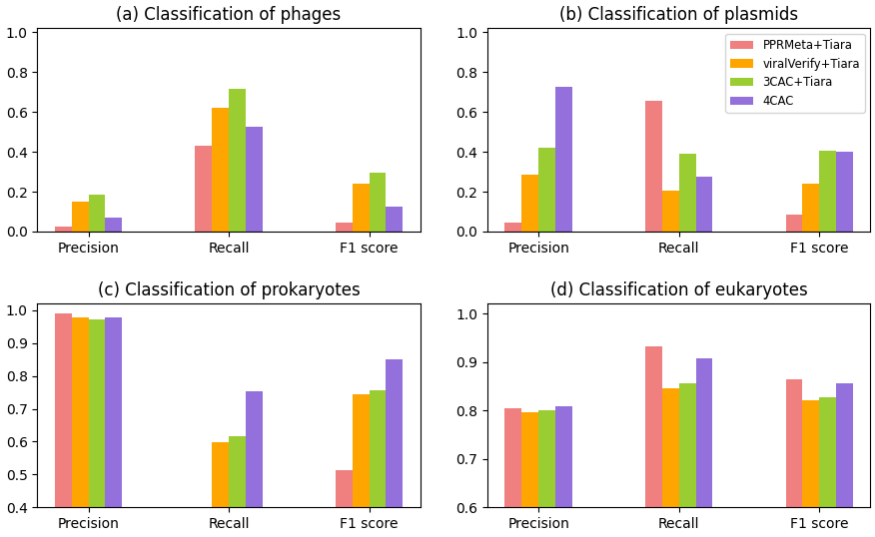

Fig. 5. Performance on each class for real short read dataset Sharon.

Table 2. Resource usage of the tested classifiers. Runtime is measured by wall clock time.

Datasets	Runtime (minutes)				RAM (GB)			
	4CAC	**viralVerify**	**PPR-Meta**	**Tiara**	**4CAC**	**viralVerify**	**PPR-Meta**	**Tiara**
Sim_S80	12.3	**322**	70.2	2.9	**24.4**	1.3	6.5	2.2
Sim_S20	7.4	**175**	41.1	2.8	**26**	0.7	6.4	1.6
Sim_L80	3.7	**185.4**	33.4	1.3	0.7	0.2	**6.3**	1.4
Sim_L20	1.4	**77.5**	14.9	0.7	0.6	0.3	**6.3**	1.4
Sharon	1.4	**29.9**	7.3	0.5	**9.9**	0.1	6.3	1.4
Tara	155.7	384.2	**530.5**	4.4	**26.2**	9.8	12.3	2.2
Oral_Nano	8.2	**452.5**	84.4	3.5	2.2	1.5	**6.2**	2.8
Gut_HiFi	9.3	**677.8**	124	4.8	2.7	2.4	**9.3**	3.7

5 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 outperforms 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. As a stand-alone algorithm, it is also easier to use, unlike 3CAC, which requires running viralVerify or PPR-meta.

Taking into consideration that prokaryotes are usually dominant in metagenome assemblies, we first tried training XGBoost classifiers on imbalanced datasets 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.

In 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, and using 3CAC and Tiara on generic long read samples, where prokaryotes constitute the overwhelming majority.

Acknowledgments

We thank Ron Saad for his helpful comments. 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.

References

1. 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), e17–e17 (2022)
2. Antipov, D., Raiko, M., Lapidus, A., Pevzner, P.A.: Metaviral SPAdes: assembly of viruses from metagenomic data. Bioinformatics **36**(14), 4126–4129 (2020)
3. 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), e121–e121 (2020)
4. 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)
5. 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)
6. Calero-Cáceres, W., Ye, M., Balcázar, J.L.: Bacteriophages as environmental reservoirs of antibiotic resistance. Trends in Microbiology **27**(7), 570–577 (2019)
7. 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), giz066 (2019)
8. 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)
9. Gourlé, H., Karlsson-Lindsjö, O., Hayer, J., Bongcam-Rudloff, E.: Simulating illumina metagenomic data with insilicoseq. Bioinformatics **35**(3), 521–522 (2019)
10. 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)
11. 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**, e7359 (2019)
12. Karlicki, M., Antonowicz, S., Karnkowska, A.: Tiara: deep learning-based classification system for eukaryotic sequences. Bioinformatics **38**(2), 344–350 (2022)
13. 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)
14. 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)
15. Krawczyk, P.S., Lipinski, L., Dziembowski, A.: Plasflow: predicting plasmid sequences in metagenomic data using genome signatures. Nucleic Acids Research **46**(6), e35–e35 (2018)
16. Li, H.: Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics **34**(18), 3094–3100 (2018)
17. 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)
18. Lind, A.L., Pollard, K.S.: Accurate and sensitive detection of microbial eukaryotes from whole metagenome shotgun sequencing. Microbiome **9**(1), 1–18 (2021)
19. 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)
20. Mallawaarachchi, V., Lin, Y.: Accurate binning of metagenomic contigs using composition, coverage, and assembly graphs. Journal of Computational Biology (2022)

21. Mallawaarachchi, V., Wickramarachchi, A., Lin, Y.: Graphbin: refined binning of metagenomic contigs using assembly graphs. Bioinformatics **36**(11), 3307–3313 (2020)
22. 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)
23. McKenney, P.T., Pamer, E.G.: From hype to hope: the gut microbiota in enteric infectious disease. Cell **163**(6), 1326–1332 (2015)
24. 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)
25. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.: metaSPAdes: a new versatile de novo metagenomics assembler. arXiv preprint arXiv:1604.03071 (2016)
26. 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)
27. Pellow, D., Mizrahi, I., Shamir, R.: Plasclass improves plasmid sequence classification. PLoS Computational Biology **16**(4), e1007781 (2020)
28. Portik, D.: Website of hifi reads: https://github.com/pacificbiosciences/pb-metagenomics-tools/blob/master/docs/pacbio-data.md
29. Pronk, L.J., Medema, M.H.: Whokaryote: distinguishing eukaryotic and prokaryotic contigs in metagenomes based on gene structure. bioRxiv (2021)
30. Pu, L., Shamir, R.: 3cac: improving the classification of phages and plasmids in metagenomic assemblies using assembly graphs. Bioinformatics **38**(Supplement_2), ii56–ii61 (2022)
31. 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)
32. 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 pp. 1–14 (2020)
33. Roux, S., Enault, F., Hurwitz, B.L., Sullivan, M.B.: Virsorter: mining viral signal from microbial genomic data. PeerJ **3**, e985 (2015)
34. 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)
35. Sitaraman, R.: Prokaryotic horizontal gene transfer within the human holobiont: ecological-evolutionary inferences, implications and possibilities. Microbiome **6**(1), 1–14 (2018)
36. Wein, T., Hülter, N., Mizrahi, I., Dagan, T.: Emergence of plasmid stability under non-selective conditions maintains antibiotic resistance. Nature Communications 10: 2595 (2019)
37. 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)
38. Wood, D.E., Lu, J., Langmead, B.: Improved metagenomic analysis with kraken 2. Genome biology **20**(1), 1–13 (2019)
39. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome biology **15**(3), 1–12 (2014)
40. 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)
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. Yang, C., Chu, J., Warren, R.L., Birol, I.: Nanosim: nanopore sequence read simulator based on statistical characterization. GigaScience **6**(4), gix010 (2017)
43. 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)