Benchmarking combined informatics approaches for virus discovery: Caution is needed when combining viral identification methods

Bridget Hegarty[1*+], James Riddell[2*], Eric Bastien[3], Kathryn Langenfeld[4], Morgan Lindback[3], Anthony Wing[3], Jaspreet Saini[5], Melissa Duhaime[3]

[1] Department of Civil and Environmental Engineering, Case Western Reserve University, Cleveland, OH
[2] Department of Microbiology, The Ohio State University, Columbus, OH
[3] Department of Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, MI
[4] Department of Civil and Environmental Engineering, Stanford University, Palo Alto, CA
[5] Department of Genetic Medicine and Development, University of Geneva, Geneva, Switzerland
[*] BH and JR contributed equally to this manuscript
[+] corresponding author

Abstract

Background

Identifying viruses from mixed environmental metagenomic samples has become an essential component of microbiome studies. Many informatics tools have been developed to recover viral sequences from mixed metagenomic datasets. As each possesses different biases and behaviors, it is difficult to know which tool(s) are best suited for a particular study, and can lead to identifying many false positives and/or capturing only a small fraction of total viruses in the sample.

Results

We benchmarked 63 viral identification tool combinations using a mock environmental metagenomic dataset composed of publically available viral, bacterial, archaeal, fungal, and protist sequences (NCBI RefSeq, Ocean Virome Contig Database). We then applied these tool combinations to different aquatic metagenomes (fresh and saltwater, drinking water, wastewater) to evaluate the impact of habitat on tool and rule set performance. We evaluated 27 published viral identification tools and benchmarked six tools best suited for viral sequence recovery from mixed metagenomic datasets. We also provide broad recommendations for which tool combinations to use based on sampling conditions.

Conclusion

Combining viral ID tools captures a greater proportion of all viruses, but also captures a significantly larger proportion of false positives. Using CheckV and Kaiju as quality filtering steps can help decrease cellular contamination, but even the most optimal tool combinations do not exceed ~75% accuracy. This accuracy plateau may be because viral ID tools rely on accurately labeled data for training and validation, but many sequences in NCBI are incorrectly labeled as non-viral entities when they are actually viral, or genes are annotated as viral when they may also be naturally present in non-viral entities. While improved algorithms may lead to more accurate viral ID tools, this should be done in tandem with curating accurately labeled viral gene and sequence databases.

Background

Viruses are an essential component of microbial ecosystems: they influence nutrient cycling and microbial community dynamics[1], account for 20-40% of microbial mortality per day[2], reprogram their hosts metabolisms,[3,4] and horizontally transfer genes across taxa.[5,6] The primary approach used to discover and describe viral diversity is culture-independent metagenomic sequencing. However, viral sequences remain challenging to differentiate from non-viral ones because viruses have no universal marker gene,[7] high mutation rates,[8,9] and relatively small reference databases relative to their diversity.[10] Additionally, current environmental sample collection and sequencing methods recover many short contigs, which are challenging to classify correctly because they often do not contain enough information to leverage our knowledge of what makes viral sequences distinct.[11,12]

The challenge of identifying viral sequences in metagenomic datasets has driven the development of many viral identification tools over the past decade that aim to differentiate viral sequences from non-viral sequences. With so many tools available, it can be difficult to choose the most appropriate one for a given study. Tools differ in the types of viruses they identify, what sequence lengths they are more accurate for, and the training data and algorithms underlying them. To be confidently applied to environmental data, viral identification tools must be trained on representative sequences. For example, Virsorter2[12] included plasmid and eukaryotic genome fragments (protist and fungi), which are found in mixed metagenomic samples, in their model and validation. Virsorter2 and another tool, VIBRANT,[13] used non-refseq sequences to validate their tool and compare it to other tools, expanding the diversity of sequences each was challenged against. VIBRANT additionally compared the sets of viruses recovered by different viral identification tools across 13 environmental samples, demonstrating clear differences in the number of sequences called viral between environments. However, tools using validation strategies not truly representative of the datasets they are to be used on continue to be published, frequently claiming to have higher or comparable accuracy to these tools.

While many viral identifications tools have comparable accuracy, their underlying algorithms are quite different and tend to capture different sets of viruses from the same sample. Some have attempted to leverage these differences to capture a greater portion of the viral signal by using the outputs of multiple tools to classify viral sequences.[14–16] This approach assumes that combining multiple tools will improve overall accuracy, but this assumption has not been validated in the literature.

Multi-tool approaches pool all putative viral sequences identified by each viral identification tool and exploit the combined strengths and weaknesses of each tool to distill a set of higher confidence viral sequences. **Therefore, we hypothesized that a multi-tool approach will discover more viruses (higher recall) without greatly increasing contamination (comparable precision) in metagenomic samples** To test this, we benchmarked 63 multi-tool approaches that leverage the different strengths of existing viral identification tools and suggest pipelines best suited for short and long-read sequences. These pipelines, by returning more viral sequences with less non-viral contamination, will enable new and more accurate insights in microbial ecology.

Methods

In Figure 1, we describe the overall process of our benchmarking of 63 combinations of viral identification tools. We pulled viral, bacterial, fungal, plasmid, protist, and archaeal sequences from NCBI RefSeq, as well as supplemented the viral sequences from the Ocean Virome Contig Database. We then assessed 29 available viral identification tools and selected a subset for comparative analyses. Using these tools we defined score cutoffs for viral sequences and compared the accuracy (MCC), precision, and recall of each combination.

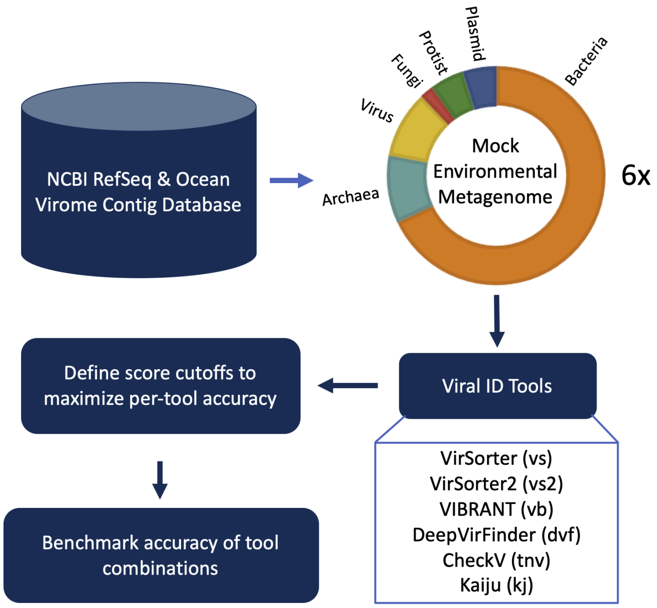

NCBI RefSeq & Ocean Virome Contig Database
Plasmid
Protist
Fungi
Virus
Archaea
Bacteria
Mock Environmental Metagenome
6x
Viral ID Tools
VirSorter (vs)
VirSorter2 (vs2)
VIBRANT (vb)
DeepVirFinder (dvf)
CheckV (tnv)
Kaiju (kj)
Define score cutoffs to maximize per-tool accuracy
Benchmark accuracy of tool combinations

Figure 1. Workflow Overview. Sequences > 3 kb were randomly pulled from NCBI and the Ocean Virome Contig Database to generate six mock environmental metagenomes. They were run through six viral identification tools, where score cutoffs were defined based on each tool's outputs to maximize their accuracy. Accuracy was then assessed for each tool combination.

Selection of viral identification tools

29 viral identification tools[12,13,17,17–42] were found through literature search and assessed to determine their suitability for inclusion in this study (Table S1). As new viral identification tools are actively being developed, only tools published before June 2022 were included in this study. Tools were included if they met the following criteria: the tool 1) identifies viruses that infect prokaryotes (i.e., bacteria and archaea), 2) can be applied to multiple environments, 3) is designed to target viral contigs of lengths greater than 3 kilobases, 4) can run millions of contigs within a few days on high performance computing clusters (i.e., not only available on a web server), 5) actively responds to user issues, 7) performs well in previous comparative studies of viral identification tools,[12,13,43] and 8) is not specific to prophages.

Four of the tools met the above criteria: DeepVirFinder,[19] VIBRANT,[13] VirSorter,[39] and VirSorter2.[12] While not strictly viral identification tools, two other applications were used to tune and improve the quality of the viral predictions in our test sets, Kaiju[44] and CheckV.[18] All six tools will be referred to as "viral identification tools" or simply "tools" in this manuscript. More information about the chosen tools can be found in Table 1.

Table 1 - Overview of viral identification tools selected for inclusion in this study.

Tool name (version)	Tool Description	Algorithmic Approach	why?
CheckV[18]	CheckV is an automated pipeline that identifies closed viral genomes, estimates the completeness of genome fragments, and removes host regions from proviruses.	HMM virus and host marker genes, virus-host boundary prediction, AAI based estimation of genome completeness	Not a viral identification tool, but provides useful benchmarking information for refining predictions from other tools.
DeepVirFinder[19]	DeepVirFinder uses a multi-layered deep learning algorithm trained on a positive set of viral sequences from viral RefSeq data and a negative set of prokaryotic ones.	K-mer based deep learning convolutional neural network	Recent and increasingly commonly used tool based on a neural net.
VIBRANT[13]	VIBRANT is a hybrid tool that uses both machine learning and protein similarity to classify viruses as either high, medium, low quality, or non-viral.	Neural network of protein annotations of HMMs	Recent and commonly used tool, provides useful gene annotation information
VirSorter[39]	VirSorter uses probabilistic models with reference and non-reference dependencies, as well as detecting hallmark viral genes.	Probabilistic modeling using HMMs	Commonly used, high quality predictions
VirSorter2[12]	VirSorter2 uses a neural network classifier built on top of the existing VirSorter infrastructure of reference based viral identification.	A multi-classifier combining a Random Forest model and expert knowledge of viral features	Recent and increasingly commonly used tool that better covers viral diversity than most other tools.
Kaiju[44]	Kaiju is a taxonomic classifier that compares metagenomic sequences to NCBI reference databases at the protein level and assigns a near or exact taxon match if one is found.	A taxonomic classifier that uses protein-level classifications to assign reads to taxon from NCBI databases.	Not a viral identification tool, but extremely fast method of taxonomic identification based on NCBI releases

Testing sets were run through each of the viral identification tools (CheckV (v0.9.0), DeepVirFinder (v1.0), Kaiju (v1.9.0), VIBRANT (v1.2.1), VirSorter (v1.0.6), and VirSorter2 (v2.2.3)) using the University of Michigan Great Lakes Supercomputing Cluster. The default settings were used except in virus-enriched samples (filtered < 0.22 µm, Table 2), the –virome flag was used for tools that had it. The non-redundant and eukaryotic NCBI database (updated 05-23-2022) was used for Kaiju taxonomic classification. All tools were run with default parameters except for specifying a 3000-base contig length cutoff.

Creation of sequence test set

To capture the variability due to test data selection, eight testing sets were created by randomly sampling genomes with replacement from NCBI and Virsorter2 curated databases to create datasets that mimicked metagenomic environmental data. As the variability between testing sets was captured with five (Figure S1), that many were used for subsequent analyses. The testing sets were composed of approximately 68% bacteria, 10% archaea, 10% virus (not proviruses), 5% plasmid, 5% protist, and 2% fungi sequences, totalling ~8k sequences. The proportion of sequences was chosen to be representative of metagenomic data, which are dominated by bacterial sequences.[11,45] The non-viral portion was randomly sampled from 5.3M bacteria, 55.1k archaea, 6.6k plasmid, 69.6k fungi, and 216.5k protist sequences from the NCBI database (accessed Nov 2019 for bacteria and archaea, April 2022 for others). The viral portion of the testing set was generated by random sampling from 13.8k viral sequences from the NCBI database and 370.153k viral sequences from the Virsorter2 curated database. As DeepVirFinder requires sequences to be less than 2.1 Mb, a custom python script was written to trim the testing set sequences to meet this length cutoff. Further, only sequences longer than 3000 bp were used.

Design of viral identification rule sets

Six rule sets were designed to predict viral contigs using outputs from at least one of the six selected tools. These rule sets were designed through three processes: 1) *Evaluating existing recommendations for tool cutoffs and application*: the recommended cutoffs for distinguishing viral and non-viral sequences in each tool's protocol were used as an initial set of rules.[12,14,15,46] 2) *Curation and evaluation of biological features:* each viral identification tool outputs biological features for each contig, e.g., VirSorter2 reports the number of viral hallmark genes identified, CheckV reports the completeness of a sequence and relative percentage of viral versus cellular genes. These biological features were used to create classification criteria (described below in Figure 2) to distinguish viral and non-viral sequences. 3) *Machine learning classifier:* a decision tree (max-depth=3) was made using the features from each viral identification tool to classify sequences as virus or non-virus (Figure S2 and Table S2). From the ML classifier we added the number of viral genes and host genes identified by CheckV to our initial set of features included in the tuning rules. The developers' recommendations for calling a sequence viral were used as a starting point; these cutoffs were then adjusted and expanded to maximize the number of true viral contigs being classified as viral and minimize the number of cellular contigs being classified as viral in the mock environmental microbial communities. An important part of this process was defining two sets of "tuning rules: (1) "tuning removal" rules that decrease the viral keep score based on distinctly non-viral sequence features and (2) "tuning addition" rules that increase the viral keep score based on distinctly viral sequence features.

Ultimately, sixty-three viral identification rule combinations were evaluated by comparing the keep score of each sequence to the classification assigned by the database. From these values, precision (the number of true viruses in our test set called viral divided by all contigs called viral), recall (the number of true viruses in our test set called viral divided by all true viruses), and Matthews Correlation Coefficient (MCC, considers relative proportion of false positives, false negatives, true positives, and true negatives)[47] were calculated (supplemental equations).

Viral Identification Rule Set Combinations

This hybrid pipeline leverages the behavior that each tool may recover different viral sequences from the testing set. For each putative viral contig or genome, the outputs of each viral identification tool are collated to generate a "keep score" as a final classification metric. Each contig is scored with the following general strategy: If a tool is confident about a contig being viral (e.g., vibrant high quality or Virsorter2 ≥ 0.95), increase the contig's "*keep_score*" by one. If a tool is only somewhat confident (e.g., vibrant low quality, VirSorter2 50-95, VirSorter category 2 or 5), increase the *keep_score* by 0.5, such that at least one other tool needs to be somewhat confident to gain a score high enough to be classified as viral (0.5 + 0.5 = 1). After the primary viral identification tools, checkV, VirSorter2, and Kaiju are used to further screen the contigs by looking at viral genes, host genes, and taxonomy. Contigs with a final *keep_score* greater than or equal to 1 were considered viral, and scores less than 1 non-viral. For a visualization of the rules see Figure 2.

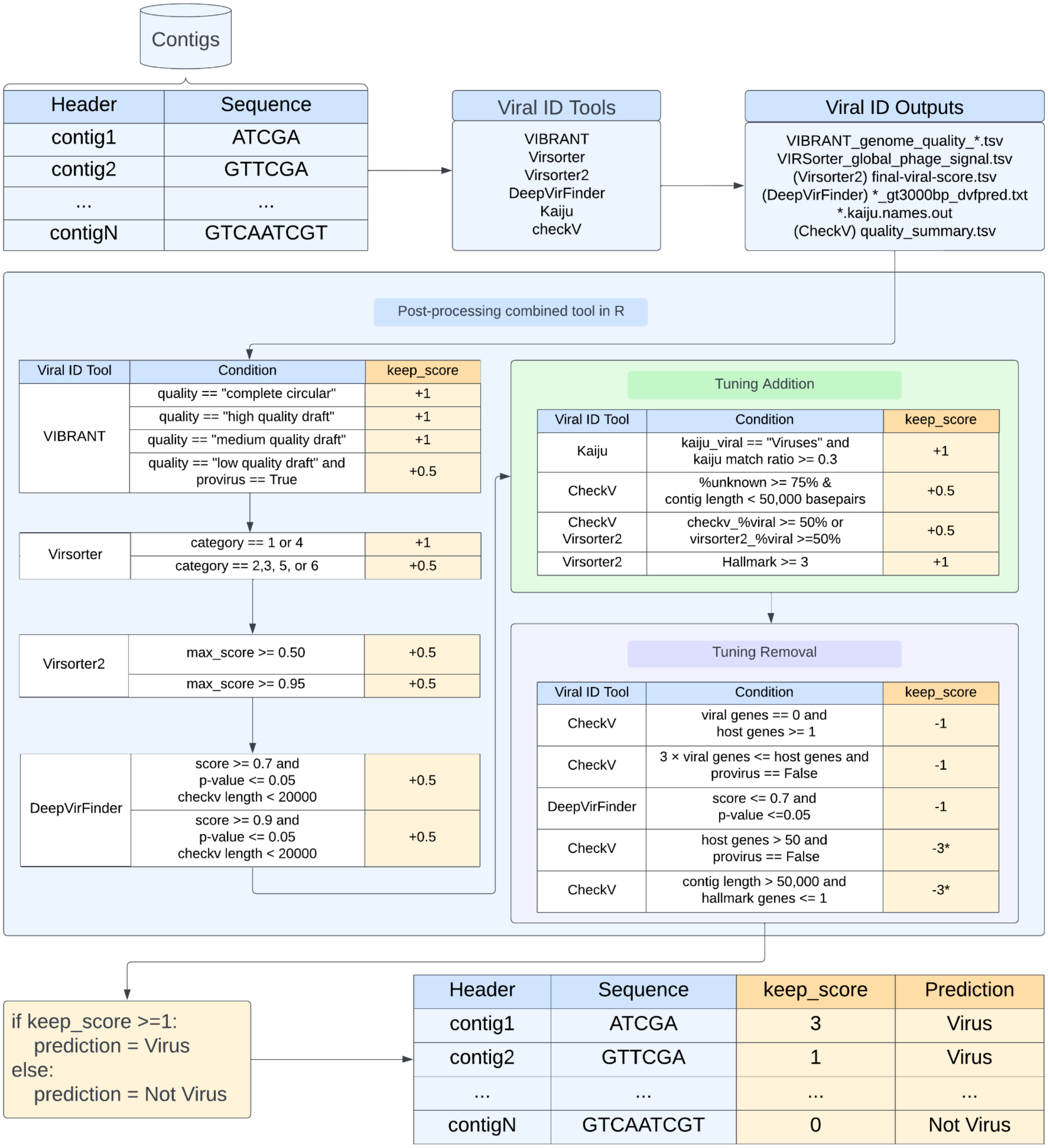
The script to identify viral contigs from these tool outputs is freely available at https://github.com/DuhaimeLab/VSTE, along with scripts to run the full testing pipeline and example runs and outputs on the environmental samples presented in this study.

Application of new pipeline to environmental metagenomes

All tool combinations were used to identify viruses from five previously published environmental datasets representing different aquatic environments and size fractions: drinking water (≥ 0.22 µm), wastewater (< 0.45 µm), eutrophic lake water (> 0.22 µm, < 0.22 µm, > 100 µm, 53-100 µm, and 3-53 µm), oligotrophic lake water (<0.22 µm), and global ocean water (0.2–3 µm). Three environments (drinking water, global ocean water, and eutrophic lake water) contained metagenomic assemblies (> 2 µm), and three environments (wastewater, eutrophic lake water, oligotrophic lake water) contained virome assemblies (< 0.2 and < 0.45µm), meaning samples were enriched for viruses by filtering through a small pore to remove most cellular organisms before DNA extraction.

Wastewater and Lake Michigan virome environmental samples were run using VIBRANT's -virome flag. All tools were run with a 3000 base pair cutoff to remove small contigs.

Sequences that (1) classified as viral by the highest precision rule set (tuning removal and VirSorter2), (2) had more than 75% of their genes not matching to any database used by CheckV (percent unknown in Figure 2), and (3) had less than 1% of the sequence aligning to the NCBI non redundant + eukaryotic database via Kaiju (kaiju match ratio in Figure 2) were considered novel. The 1% cutoff was chosen by assessing the distribution of kaiju match ratios for viral sequences of the VirSorter2 curated viral genomes.



Header	Sequence
contig1	ATCGA
contig2	GTTCGA
...	...
contigN	GTCAATCGT

Viral ID Tools

VIBRANT
Virsorter
Virsorter2
DeepVirFinder
Kaiju
checkV

Viral ID Outputs

VIBRANT_genome_quality_*.tsv
VIRSorter_global_phage_signal.tsv
(Virsorter2) final-viral-score.tsv
(DeepVirFinder) *_gt3000bp_dvfpred.txt
*.kaiju.names.out
(CheckV) quality_summary.tsv

Viral ID Tool	Condition	keep_score
VIBRANT	quality == "complete circular"	+1
	quality == "high quality draft"	+1
	quality == "medium quality draft"	+1
	quality == "low quality draft" and provirus == True	+0.5

Virsorter	category == 1 or 4	+1
	category == 2,3, 5, or 6	+0.5

Virsorter2	max_score >= 0.50	+0.5
	max_score >= 0.95	+0.5

DeepVirFinder	score >= 0.7 and p-value <= 0.05 checkv length < 20000	+0.5
	score >= 0.9 and p-value <= 0.05 checkv length < 20000	+0.5

Viral ID Tool	Condition	keep_score
Kaiju	kaiju_viral == "Viruses" and kaiju match ratio >= 0.3	+1
CheckV	%unknown >= 75% & contig length < 50,000 basepairs	+0.5
CheckV Virsorter2	checkv_%viral >= 50% or virsorter2_%viral >=50%	+0.5
Virsorter2	Hallmark >= 3	+1

Viral ID Tool	Condition	keep_score
CheckV	viral genes == 0 and host genes >= 1	-1
CheckV	3 × viral genes <= host genes and provirus == False	-1
DeepVirFinder	score <= 0.7 and p-value <=0.05	-1
CheckV	host genes > 50 and provirus == False	-3*
CheckV	contig length > 50,000 and hallmark genes <= 1	-3*

```
if keep_score >=1:
    prediction = Virus
else:
    prediction = Not Virus
```

Header	Sequence	keep_score	Prediction
contig1	ATCGA	3	Virus
contig2	GTTCGA	1	Virus
...	...	...	...
contigN	GTCAATCGT	0	Not Virus

Figure 2. Combined classifier workflow diagram. Contigs are first processed by each viral identification tool. Next, the tool outputs are loaded into an R script that manually curates a keep_score by serially analyzing each viral ID output in the order of the diagram. The last tool in the pipeline, CheckV, serves as a quality control step; it is used to remove contigs that present significantly more host-like than virus-like biometrics. If a subset of the tools are run, steps using tools left out are skipped in the R script classifier.

Results and Discussion

Viral identification tools rely on a combination of manual rules based on knowledge of sequences and machine learning. In this manuscript, we test the hypothesis that combining viral identification tools will improve accuracy. Performance of 63 combinations of six rule sets were evaluated (Figure 3). The six rule sets are as follows: four rule sets based on the four viral identification tools tested (VIBRANT, Virsorter, Virsorter2, DeepVirFinder) and two additional tuning rule sets: tuning removal (Kaiju, CheckV, VirSorter2, VirSorter, and VIBRANT) and tuning addition (Kaiju, CheckV, and Virsorter2) (Figure 2). Our tuning rules build on those manually implemented elsewhere and are critical for reducing false positives in our predictions. Through automating tuning rules, we developed an improved automated pipeline for identifying viral sequences. In addition to testing on mock environmental metagenomes, we tested selected combined pipelines on environmental datasets and investigated how many novel viruses were discovered from each environment. From these efforts we recommend the use of Virsorter2 with the tuning addition and removal rule sets for metagenomic data. In the rest of this section, we will compare the performance of our rule combinations and elaborate on their strengths and weaknesses.

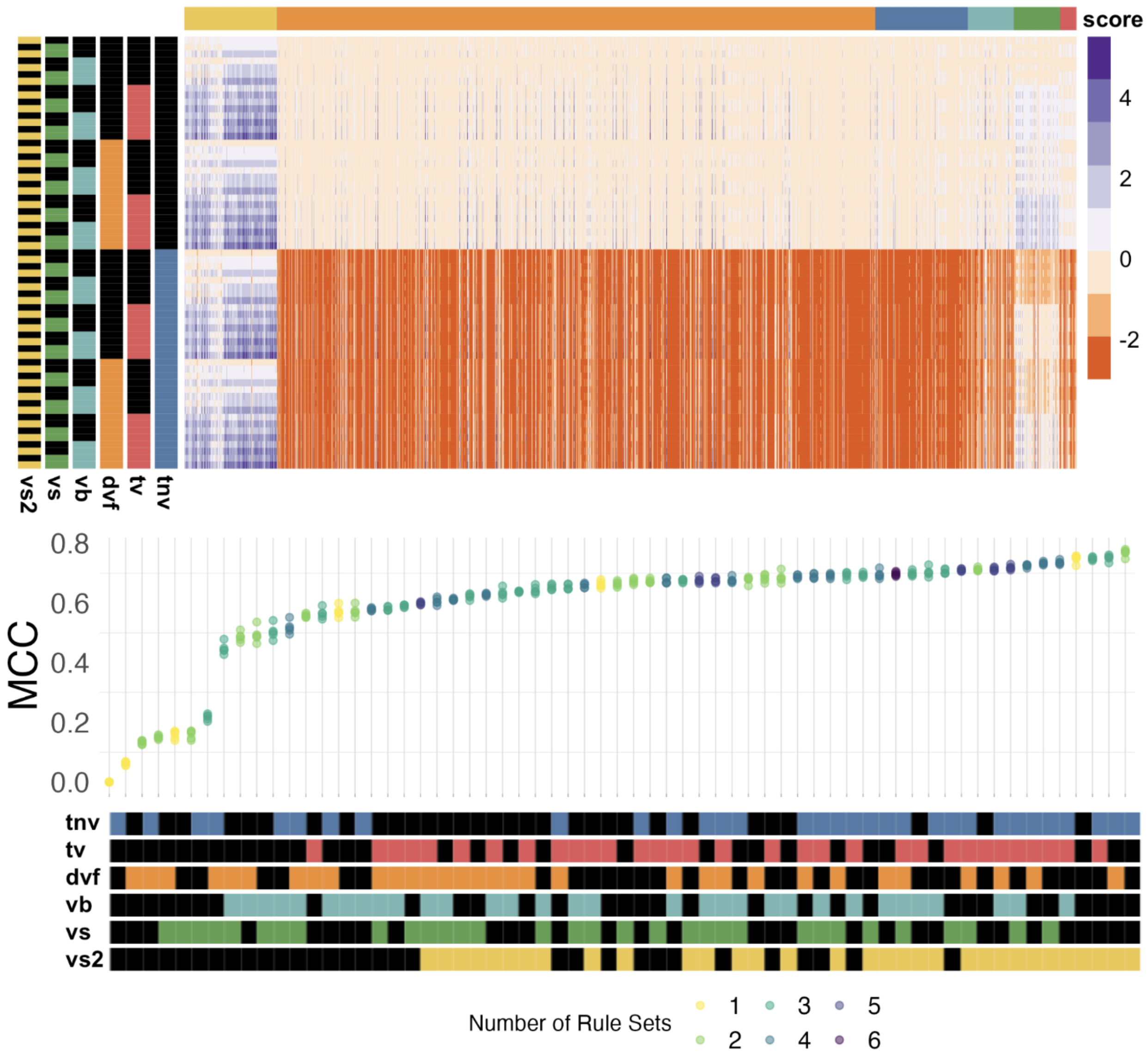


Figure 3. Comparison of different tool combinations. (Top) Viral score for each sequence in one of the sets of training data. Colored based on the score from the model (purple-viral; orange-not viral). Columns are colored based on the sequence type (yellow: viral, orange: bacterial, blue: archaea, light blue: plasmids, green: protists, and pink: fungi) and rows based on the use of each rule set. **(Bottom)** - Tool combinations ordered by increasing MCC.

More tools better… to a point

Across the 63 combinations, MCC, our metric for overall performance, ranges from 0 to 0.77 and generally increases when combining tools and tool combinations (Figure 3). Importantly, while some combinations of rules sacrifice precision as recall is improved, others greatly improve recall without compromising precision (Figure 4A). Both precision and recall increase with two or more rule sets compared to the individual rule sets (Table S3 and S4). Besides VirSorter2 (MCC = 0.75), viral identification tools on their own either miss most of the viruses or classify so many non-viruses as viruses that the viral signal is heavily contaminated (Figure 4A, Figure 5). The VIBRANT rule set comes the closest to VirSorter2 (MCC = 0.55). VirSorter (MCC = 0.16) and DeepVirFinder (MCC = 0.05) on their own had both low precision and recall.

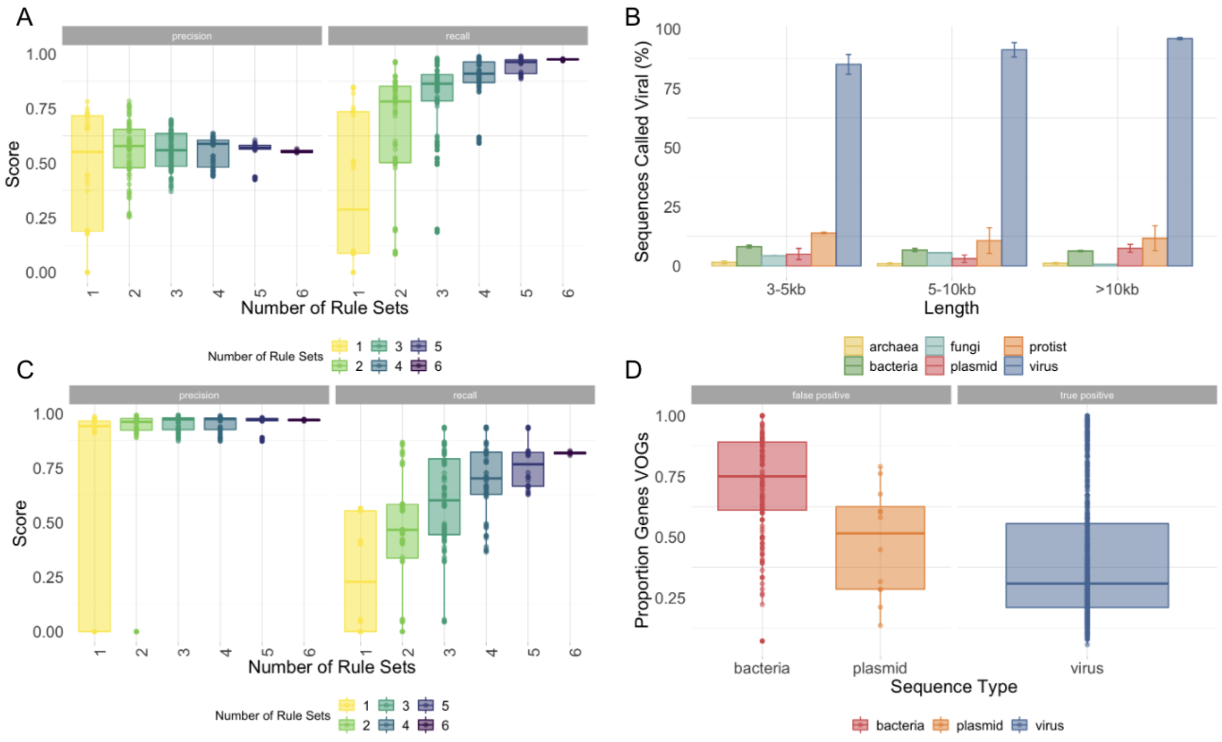


Figure 4. Performance of the various pipelines. (A) - Boxplots of the variation in precision and recall of different tool combinations based on the number of tools used in prediction (all size ranges). **(B)** - Percentage of sequences called viral averaged across the 10 testing sets. Sequences are binned by sequence type and length. Error bars represent the standard deviation across the 10 testing sets. Bars are colored by sequence type. **(C)** - Boxplots of the variation in precision and recall of different tool combinations based on the number of tools used in prediction (3-5 kb viral fragments). **(D)** - Plotting the proportion of genes on a contig with a VOG annotation based on VIBRANT broken down by sequence type (for the high MCC rules).

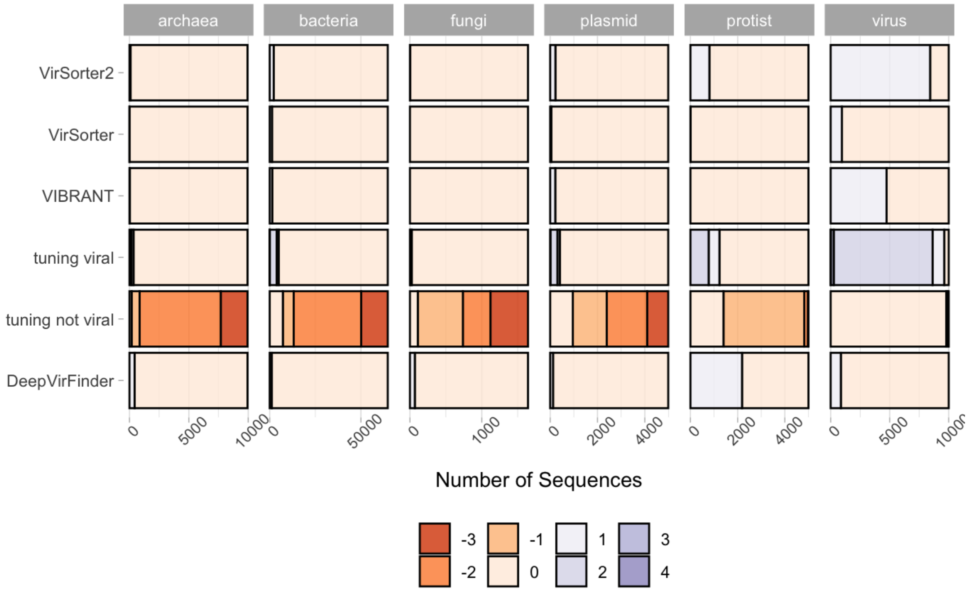


Figure 5. Performance of Individual rule sets. Number of sequences with a given viral score for each rule set faceted by the domain of the sequence.

There is no statistically significant difference between the precision of two or more rules (Table S3) or the recall of four or more rules (Table S4). Past this point, using more rules does not improve accuracy. While rule sets with VirSorter, DeepVirFinder, and VIBRANT all have more sequences in common as the number of rules in the compared sets increases, this trend is much less pronounced for VirSorter2 (Figure S3), suggesting that VirSorter, DeepVirFinder, and VIBRANT have more distinct viral predictions compared to VirSorter2. The two tools that perform best in isolation, VIBRANT and VirSorter2, have many sequences in common, but very few are gained by their union in isolation (Figure S4).

The comparable accuracy for many rule sets of just a subset of the rules was also observed from the hierarchical trees produced (Figure S3). Interestingly, similar accuracy was achieved using many different rules based on various combinations of only a few of our features. Overall, we were able to achieve similar accuracy with the decision tree approach compared to our manually-derived rules. We chose to use a decision tree in this study to have a simple and transparent model that we could explicitly compare the decisions being made by a naive model to our manual rules. Nearly all of the 100 hierarchical trees built had one feature that dominated in importance for feature splitting. However which feature was most important varied greatly; of the most important features, the three most common were as follows: percent host was the most important feature in 14, percent viral in 12, and VirSorter2's max score in 10. All features were used for splitting in at least one of the 100 trees (DeepVirFinder's p-value was only used in one).

Comparably high MCCs (~ 0.77) can be achieved using (1) VirSorter2 and tuning removal rule sets (Kaiju, CheckV, DeepVirFinder), (2) VirSorter2 with DeepVirFinder and the tuning removal rule sets, (3) VirSorter2 and both the tuning removal and tuning addition rule sets, (4) VirSorter2 in isolation, (5) VirSorter2 with VIBRANT and both the tuning removal and addition rule sets, and (6) VirSorter2 with VirSorter and both the tuning removal and addition rule sets (Figure S6). Further, VirSorter2 was in all of the top ten highest MCC rules and the tuning removal rule set was in eight. For comparison, none of the other tools (VirSorter, DeepVirFinder, and VIBRANT) were in more than three of the top ten highest MCC rules independently of the other three. All of this clearly indicates the importance of VirSorter2 and the tuning removal rules.

As the high MCC examples illustrate, many of the rule combinations have a high degree of overlap with others (Figure 6, Figure S9). While only 4% of the combinations identified more than 90% of the same viruses as another combination, 68% of the combinations were more than 50% identical to another combination. Of the six rule sets, VirSorter2's returns the most viruses. Each rule set (other than the tuning removal rule) combined with VirSorter2 adds more viruses with the most being contributed by VirSorter (Table S5). Our rule combinations use the consensus of lower quality viral predictions to improve total viral recall. For example, if VirSorter and VirSorter2 both give a viral score of +0.5, then it is called viral by the combined prediction of the two tools (combined viral score = 1).

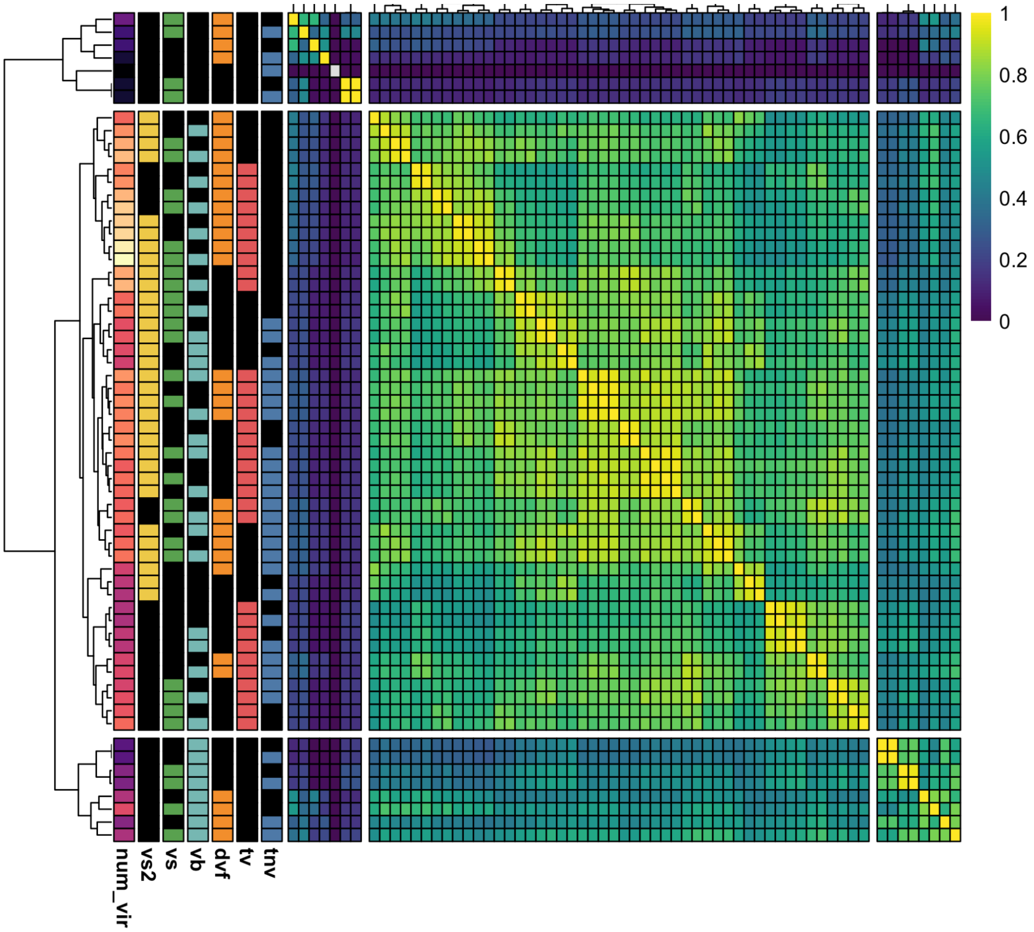


Figure 6 - Proportion of viruses in common between rules combinations. Heatmap values calculated by dividing the intersection by the union of the viruses found by both rule combinations. Bars to the left of each heatmap represent the total number of viruses identified by each tool combination (num_vir), as well as whether a tool combination used VirSorter2 (VirSorter2, yellow), VirSorter (vs, green), VIBRANT (vb, light blue), DeepVirFinder (dvf, orange), tuning addition (tv, pink), or tuning removal (tnv, indigo).

As metagenomes frequently have a high proportion of short contig fragments, we also tested the rule sets on 3-5 kb fragments from our original testing sets. Correctly identifying short fragments with only a few genes is particularly challenging for fragments that contain genes shared between viruses and hosts, as well as for those with many genes of unknown function and origin.[18] For these fragments, accuracy (MCC) improved with the number of tools with the best performance with all 6 rule sets (Figure 4C). Adding the tuning rules reduces the level of contamination significantly compared to the longer fragments. Unlike the genomes pulled (both complete and fragments), most of the false positives were not called by VIBRANT, though there was still a similar pattern where the false positives had a higher percent viral (VirSorter2 and CheckV) than the true negatives. Conversely the false negatives had a lower percent viral (VirSorter2 and CheckV) than the true positives. This illustrates the difficulty of identification of short fragments. In part due to the viral tuning rules, the accuracy for these short fragments is greater than previously published viral identification SOPs (Figure S8) (add a citation).

Distinguishing between viral and non-viral features

We were unable to simultaneously improve both precision and recall beyond ~ 75% (Figure S6). Improvements in recall past this point led to sacrifices in precision. This is because, to recover more viruses, it becomes necessary to rely more on unknown features that may also overlap with other taxa (particularly eukaryotes, which were not represented in DeepVirFinder's, VirSorter's, or CheckV's training data) (Figure S5). Many true viral features overlap with nonviral features (Figure S5) due to our imperfect knowledge of what distinguishes viruses and non-viruses (and the overlap between viral and host sequences); in the effort to capture all the true viruses, nonviruses with these features are misclassified as viral. This challenge is particularly acute when trying to accurately classify short sequences.[12,19]

To attempt to address this challenge, we designed tuning addition and removal rules based on our observations of features of viral genomes, and the VirSorter2 SOP. The contributions of each tuning removal set component are visualized in Figure 7. Table S5 illustrates how the viral removal tuning rule refines the viral predictions, ranging from 76% removal for DeepVirFinder to 10% removal for VirSorter. The proportion of predictions removed reflects the lower precision for DeepVirFinder's predictions compared to VirSorter's. Reflecting the importance of the tuning rule sets on accuracy, seven of the rule combinations of the highest MCC rule sets have both (Figure 3). Very few viral sequences are flagged by multiple of the tuning removal rules (only 0.6% of the true viral sequences). Conversely, few non-viral contigs are flagged as viral by multiple of the tuning addition rules (2% of the true not viral sequences). Further, the majority of the not viral sequences (bacterial, protist, archaeal, fungal, or plasmid) were removed by a multiple of tuning removal rules (76% of the true not viral sequences).

These rules increase our confidence that false positives will be removed, even if some of the viral identification rules mislabel them as viral. Based on our benchmarking in this paper, we would urge caution when using some of the new automated snakemakes for sequence identification that combine the output of multiple tools. We found that having rules to remove sequences that were called viral by individual (or combinations of) viral identification tools was an essential component of the accuracy of our model.

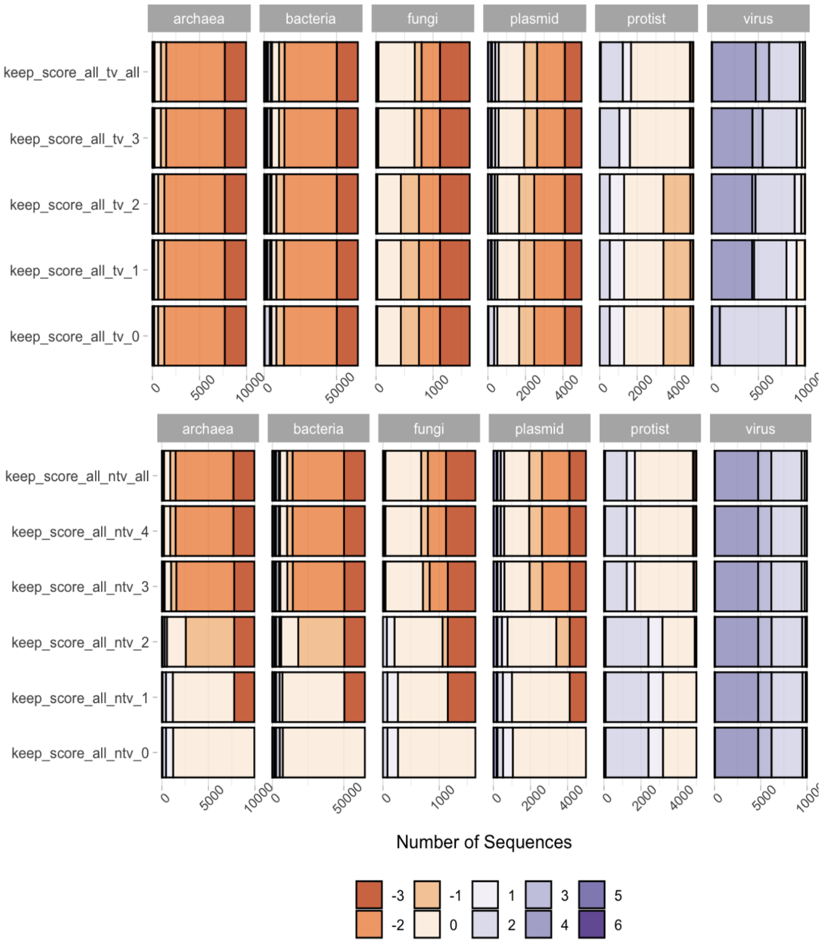


Figure 7. Individual performance of each tuning rule. Number of sequences with a given viral score for the **(Top)** tuning addition and **(Bottom)** tuning removal rules faceted by the sequence type. The tuning addition rules are (1) VirSorter2 viral hallmark genes > 2, (2) called viral by Kaiju and Kaiju match ration ≥ 0.3, (3) percent unknown ≤ 75% and sequence length ≤ 50 kb, and (4) percent viral ≥ 50% by VirSorter2 or CheckV. The tuning removal rules are (1) CheckV host genes > 50 and not called a provirus, (2) CheckV viral genes = 0 and host genes ≥ 1, (3) 3 times the number of CheckV viral genes*3 ≤ the number of CheckV host genes and not a provirus, (4) longer than 500 kb and one or fewer VirSorter2 hallmark viral genes, and (5) DeepVirFinder score ≤ 0.7 and p-value ≤ 0.05.

Environmental comparisons

We applied all 63 of the rule sets refined with the mock metagenomes to to publicly available mixed metagenomic assemblies from Lake Erie water, Lake Michigan water, drinking water from the United Kingdom and Netherlands, wastewater from a Michigan wastewater treatment plant, and TARA global oceans (Table 2, Figure 8). To compare the performance of each rule set, we analyzed the proportion of viruses recovered from each environment by different rule sets. Since we cannot know the true identity of these sequences without laborious manual inspection this allowed us to benchmark between the rule sets and to the proportions expected for similar environments.

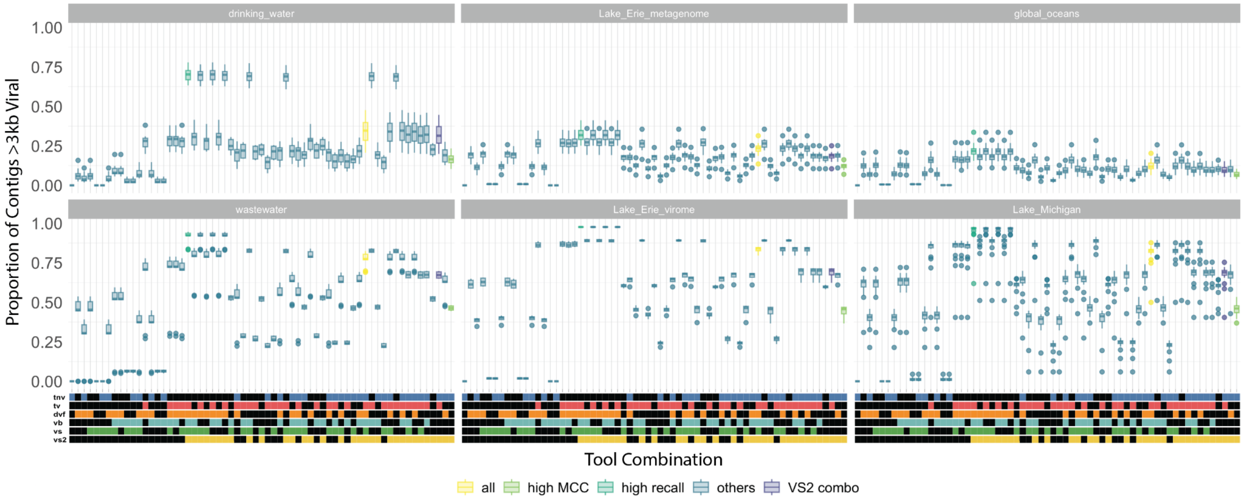


Figure 8 - Proportion of viruses predicted by each tool combination across five environmental datasets

The difference in proportion of viruses recovered between the viromes and the metagenomes was the most apparent difference between our samples. A greater proportion of sequences were identified as virus from virus-enriched samples than non-enriched. Virus-enriched samples had ~45% sequenced identified as virus whereas non-enriched samples identified anywhere from 7-19% viral. Across the rule sets, the trends in proportion of viruses recovered mimics the expected behavior from the testing data. The "high recall" rules return the most viruses, while the "high precision" rules returned many fewer. The "high MCC" and "all" rules returned a proportion in between these extremes. For the virus-enriched samples, nearly all contigs were called viral by the "high recall" tool.

The tuning removal rule greatly reduces the number of contigs identified as viral. For metagenomes, the inclusion of the tuning removal rule, on average, decreased the proportion of viruses identified by nearly half, from 21% to 12% of contigs > 3 kb. For viromes, there was a smaller decrease from 63% to 48%. Conversely, the tuning addition rule increased the proportion of viruses identified in metagenomic assemblies from 12% to 21%, on average, and in virome assemblies from 45% to 66%. Together, including both tuning addition and removal rules slightly increased the proportion of viruses identified compared to no tuning rules from 15% to 16% in metagenome assemblies and 54% to 58% in virome assemblies. While we cannot know what the true classification of our environmental sequences are, these statistics are reassuring in that the tuning rules are behaving similarly on the environmental samples as on the testing sets.

Depending on the specific experimental measures taken to reduce bacterial contamination in the viral-enriched samples, the "tuning removal" rules may be unnecessary and remove more potential viruses than hosts (Figure S7). In virome samples, the tuning removal rule set leads to ~15% fewer sequences being called viral than without it across samples. The tuning removal rule set may not be applicable to virome samples because its primary purpose is to remove cellular contamination. If most of the cellular contents were removed in the filtering process, then the reads may assemble into longer, more high quality viral contigs that are punished by the length cutoffs in the tuning removal step. Conversely, the tuning removal rule set is very important for removing false positives from metagenomic samples, where ~50% fewer sequences are called viral when including the tuning removal rule set with the rest of the rules for the environmental metagenomes tested in this study ("high recall" versus "all" rule sets, Figure 8).

Furthermore, the tuning addition rule may disproportionately contaminate the viral signal in samples where eukaryotes are not filtered out, such as the drinking water samples in this study. The drinking water samples were not pre-filtered to remove eukaryotes (Table 2), and rule combinations that included the tuning addition rule called 2 to 3 times more sequences viral than combinations without it (Figure 8). Even with the tuning removal rule, the number of sequences called viral is nearly double that of the Lake Erie and Global Oceans metagenomes. Given that eukaryotic sequences are often misclassified as viral by our rule sets, it's probable there is significant eukaryotic contamination in the viral signal. Testing the rules on further environmental data types may reveal whether the difference between drinking water and the other metagenomes was due to specific differences based on environment (for instance, that drinking water viruses were not well-represented in the databases) or related to the particular experimental design of that study.

Biases in testing data diversity limit viral discovery

The need for further experimental discovery of new viruses is well established.[12,48] Discovering novel viruses is important because existing databases only represent a small fraction of viral diversity. Virus sequence libraries are still relatively small compared to bacteria and eukarya sequence libraries despite viruses being at least two magnitudes more numerous and possessing greater sequence diversity.[49,50] Virus sequence libraries are also biased towards a few highly researched domains (e.g., oceans, human pathogens)[51] and methods to extract and identify viruses have led to taxonomic biases.[52]

To this end, we assessed the number of novel viruses our pipeline could identify by using gene annotations. Novel contigs ranged from 1-10% of all viral contigs recovered, the highest proportions being from engineered environments (Table 2). This is likely due to engineered environment samples being underrepresented in metagenomic sequencing compared to lakewater and marine environmental samples.

We broke down the number of novel viruses by virus type according to VirSorter2 (Table 2). The vast majority of novel viruses identified were classified as dsDNAphage. This is not surprising given that dsDNAphages are the most well represented in viral databases (due to biases in sample collection methods) and should be considered when doing ecological analyses of viral diversity from environmental samples.

Table 2. Environmental Data Sets. The proportion of viral sequences represents the number of viruses predicted by the high MCC rule set out of the total number of sequences greater than 3 kb.

Environment	Location	Size Fractions	Proportion Viral	Proportion of “Novel” Viruses	Proportion dsDNAphage	Proportion Lavidaviridae	Proportion NCLDV	Proportion ssDNA
oligotrophic lake water	Lake Michigan	<0.22µm	0.45	0.01	0.85	0.05	0.11	0.001
wastewater	Michigan, USA	<0.45µm	0.47	0.10	0.99	0.00	0.01	0
eutrophic lake water	Lake Erie	>0.22µm, <0.22µm, >100µm, 53-100µm, and 3-53µm	0.16	0.02	0.89	0.04	0.07	0
drinking water	United Kingdom and Netherlands	≥0.2µm	0.19	0.04	0.92	0.01	0.07	0
marine surface water	TARA Global Oceans	0.2–3 µm	0.07	0.01	0.64	0.01	0.35	0

To further investigate the biases in our data, we compared the accuracy across viral types, as well as for Refseq versus NonRefSeq sequences. The accuracy of four representative rule sets across viral types demonstrates that the accuracy of our pipelines varies based on viral type (Figure 9). The highest accuracy is for dsDNAphages across all four representative rule sets. Interestingly, the high precision rule misses more of the non-dsDNAphage viruses compared to the other rule sets. These differences likely result because dsDNAphages are overrepresented in our testing data. Promisingly, the proportion of viruses returned using the “high MCC” rule combination was similar for both the RefSeq (0.93) and NonRefSeq (0.95) viral sequences. As additional curated viral sets are published,[53] our rules can be tested against those. This will provide important information about the limitations and scope of our rules. While not directly comparable, other tools performed more poorly on NonRefSeq viral sequences compared to RefSeq viral sequences demonstrating a clear feature bias to the RefSeq database in many viral identification tools and missing potentially other key features represented in true viral diversity.

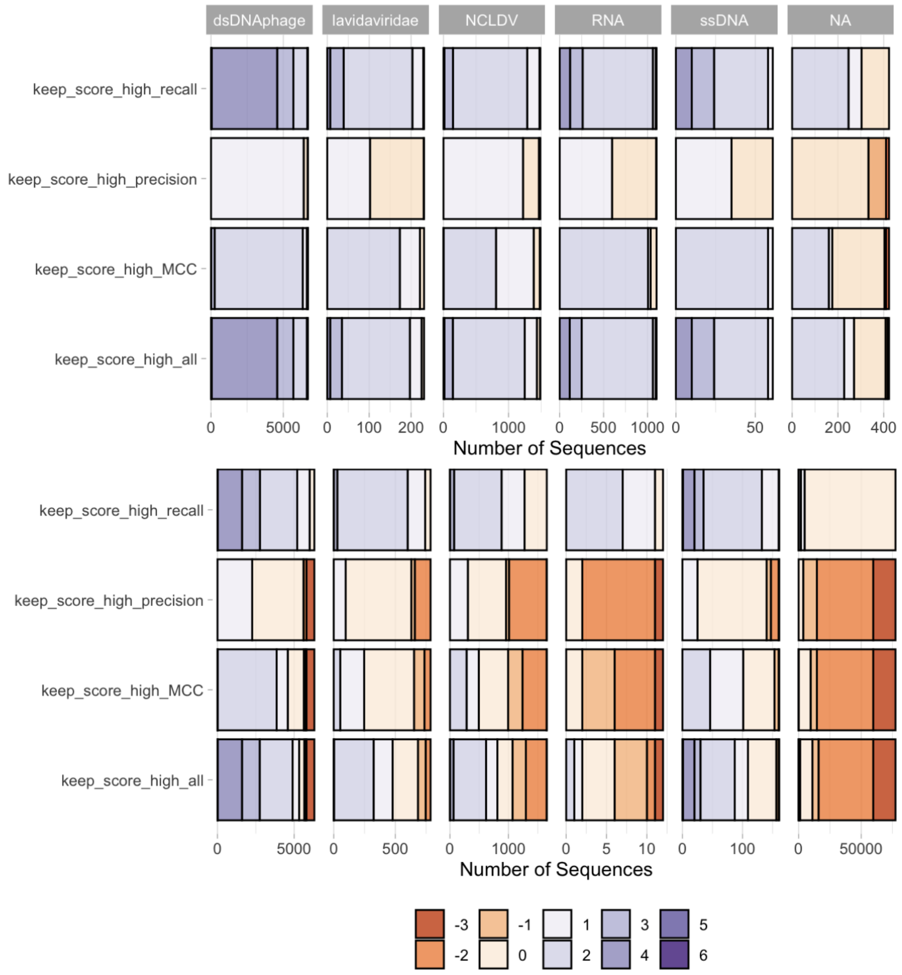


Figure 9. Performance of four representative rule sets across viral types. Number of sequences with a given viral score for four representative rule sets (high recall, high precision, high MCC, and all) faceted by the VirSorter2 viral group (NA represents sequences not called viral by VirSorter2) and separated based on **(A)** virus and **(B)** non-virus sequences. Note that the x-axes vary between panels and facets.

Mislabeled sequences inflate false positive rate

Viral identification tool classification accuracy varies with sequence length and sequence type. Viruses are more accurately classified in the 5-10 kb and > 10 kb size ranges than in the 3-5 kb size range (Figure 4D). This makes sense because, generally, longer sequences have more features such as viral hallmark genes, leading to a higher confidence classification. Fungal sequences are more frequently misclassified as viruses in the 3-5 kb and 5-10 kb size ranges compared to the > 10 kb size range. A similar proportion of bacteria, archaea, plasmids, and protists are misclassified as viruses across all of the size ranges. Looking at those sequences called viral by the high MCC rule combination, the bacteria look more "viral" than the viruses themselves; that is, the proportion of the sequence labeled as viral orthologous gene (VOG) by VIBRANT is higher in the misclassified bacteria than the viruses (p-value $< 2.2*10^{-16}$) (Figure 4D). The plasmids have a similar proportion of viral genes to the viruses (p-value = 0.63). For all three sequence types, the proportion of the sequence that is VOG increases with the number of VOGs in that sequence. For bacteria, there is a small set of short sequences with a highly viral signature (sequence predominantly VOGs).

Given the high proportion of viral genes within the misclassified sequences, we wondered if these sequences were mislabeled in NCBI. Previous studies have found widespread mislabeling of prophage as bacteria, and sequences thought to be plasmids or satellite chromosomes were actually phage genomes.[54] We first checked to determine whether the bacterial false positive sequences were ΦX 174 or *E. coli*, as ΦX contamination is a known problem across NCBI database sequences due to its use in illumina libraries.[55] Only approximately 20 ΦX sequences were taxonomically identified by Kaiju. However, approximately 1% of the viruses, 9% of the bacteria, and 5% of the plasmids were identified as *E. coli* by Kaiju. The majority of these were called "not viral" by our "high MCC" rule set; and likely represent true *E. coli* bacteria sequences and thus do not explain our high degree of false positives.

Instead, these false positives likely represent two types of sequences: (1) free viruses infecting the pure culture and (2) prophages found in their genomes. There was evidence of both types in our testing set. For instance, all four genes on the NZ_ABHO01000085.1 contig, which is labeled as a *E. coli* whole genome shotgun sequence in NCBI, have a VOG annotation (sp|A9CRB8|TAIL_BPMR1 Putative tail protein, sp|O64328|G_BPN15 Tail assembly protein G, sp|P03735|GT_LAMBD Tail assembly protein GT, REFSEQ putative prophage tail length tape measure protein). Another contig, NZ_LLFE01000196_1, which is labeled as an *Acinetobacter baumannii* strain ABBL059, has 56 genes, of which 39 have a VOG annotation and 11 were labeled as hallmark genes by VirSorter2. Similarly, many of the false positive plasmid sequences looked similar to annotated virus sequences upon manual inspection. NZ_AP017612_1, labeled as the *E. coli* strain 20Ec-P-124 plasmid pMRY16-002_2, for instance, has 127 genes, of which 86 have a VOG annotation and 12 VirSorter2 viral hallmark genes. Altogether, this further supports the known problem of phage sequences not having been removed before being deposited on NCBI. This is likely because there is currently no step to screen out viruses when doing traditional microbial omics work. Without specifically using tools to look for viral sequences, bacterial identification tools on environmental samples will classify viral sequences as mini chromosomes, plasmids, or find bacterial homology of the viral genes. Simply put, if researchers are not looking for viruses, they will not find any. This affects our precision because true viruses are incorrectly classified as bacteria or plasmid, and may have led us to be more conservative in building our rule sets since we cannot check every contig in the false positive set. Overall, we found that a significant proportion of non-viral sequences are being called viral due to inaccurate labeling of the testing data.

For the "high MCC" rule combination, 94% of the protist false positives are predicted to be Lavidaviridae and NCLDV (nucleocytoplasmic large DNA viruses). Lavidaviridae is a family of viruses that infect protists, and NCLDV is a phylum of viruses that infect protists, invertebrates, and algae. However, we cannot confidently say whether these are mislabeled sequences, as only 5% of the protist false positives were greater than 3.5 kb. Challenges in correctly identifying eukaryotic fragments have been described elsewhere.[12] Other studies have found that k-mer based approaches struggle with correctly identifying short eukaryotic fragments;[11] however, this is also a problem for tools like VirSorter2. Through our tuning rules, we were able to reduce, but not eliminate this problem.

Mislabeled sequences pose a serious challenge for viral identification tools, which all rely at least partially on accurately labeled sequences to develop their models. Viral sequences mislabeled as bacteria leads to a more inaccurate model because it generates artificially higher overlap between viral and non-viral features. To overcome this challenge, we propose implementing a viral screening step for NCBI sequence uploading (especially of whole genome sequences, where many prophage and viruses may be residing).

Implications and Future Work

This paper lends further support to the observation of papers such as VirSorter2 and VIBRANT that a combination of rules based on manual curation and machine learning provides the highest accuracy viral predictions. With the rapid rate of new tools being introduced, we hope that this paper will offer a blueprint for considering which sequence features best delineate viruses from each non-virus sequence type respectively. We further hope that our analyses demonstrate that different viral identification tools should only be combined cautiously.

Our recommendations based on this study vary depending on research question and experimental design. For a typical study investigating viral diversity and functional potential from a mixed metagenome, we recommend our "high MCC" rule set (VirSorter2 with tuning removal rule sets). If eukaryotes were filtered out of the sample (< 3 µm), the tuning additional rule can also be included. Lastly, if bacteria were also filtered out, the tuning removal rule may be eliminated. We urge caution in this case, as there will be many more false positives. For researchers seeking to only use one tool for viral identification, we strongly recommend Virsorter2 or VIBRANT. VirSorter2 has a comparable MCC to multitool rules with the caveat of more false positives. Although VIBRANT recovers fewer viruses, the high and medium confidence categories are very accurate. We do not recommend using Virsorter or DeepVirFinder in isolation, or including sequences identified as "low confidence" by VIBRANT as these have very poor accuracy on environmental metagenomes (Figure 8).

In silico prediction of viral sequences is a critical first step to any viral metagenomic study. There is a rapidly increasing number of i*n silico* viral prediction tools available, many of which have been benchmarked against each other.[12,13,19] However, previous studies have not systematically compared the effect of using a combination of these tools. As downstream analyses and conclusions of viral ecology are predicated on accurate viral prediction, choosing the best tool or combination of tools is paramount. We demonstrate that a combined tool approach is superior for viral recall to using tools in isolation: the combined approach recovers many more true viruses while maintaining a comparable amount of contamination to single tools. Increasing the proportion of high-confidence viruses identified from mixed metagenomic datasets will enable more accurate ecological analyses by decreasing contamination of the viral signal, particularly from eukaryotic sequences.

The effects of a combined viral identification approach on a virome's community composition has not been systematically tested across tools and habitats. While the proportion of viruses recovered for the aquatic environments are comparable to previous studies, testing against soil, human, and metagenomic datasets from more diverse environments could further validate if the high MCC rule set can recover the expected proportion of viruses from these environments. Further, we focused on tools that were developed primarily for bacteriophage identification. We did not evaluate tools that were specifically for human pathogens or eukaryotic viruses more broadly. For those applications, other tools may be preferable, but assessing this was beyond the scope of our analyses here.

Our testing herein demonstrated the value of automating the refinement of predictions from existing tools. Based on this, we recommend more viral identification pipelines to explicitly consider features of both viral and not-viral sequences. Utilizing new tools for plasmid and eukaryotic sequence identification[56,57] may be of particular use for this. Additionally, as knowledge of known viral versus host genes was an essential feature for distinguishing between sequences in our training data (and the most important feature for the machine learning classifier), expanding our knowledge of known viral genes is one of the best ways we can improve viral identification tools. In this way, improved viral identification and more accurate ecological conclusions will become possible.

Conclusion

Accurately predicting novel viral sequences is challenging due to similarity in features between viruses and their hosts. In this paper, we tested the benefits of combining multiple tools for viral identification. We found that recall increased with number of tools, while precision remained constant (e.g., 2 vs 6 tools: $p_{adj} \leq 10^{7}$ and $p_{adj}=0.78$, respectively). For most applications, we recommend a combination of VirSorter2 and tuning rules based on features of viral and not viral sequences. In summary, we were able to improve viral identification through a combination of rules based on multiple viral identification tools and classifiers.

References

1. Guidi, L. *et al.* Plankton networks driving carbon export in the oligotrophic ocean. *Nature* **532**, 465–470 (2016).
2. Wilhelm, S. W. & Suttle, C. A. Viruses and Nutrient Cycles in the Sea: Viruses play critical roles in the structure and function of aquatic food webs. *BioScience* **49**, 781–788 (1999).
3. Howard-Varona, C. *et al.* Phage-specific metabolic reprogramming of virocells. *ISME J.* **14**, 881–895 (2020).
4. Hurwitz, B. L. & U'Ren, J. M. Viral metabolic reprogramming in marine ecosystems. *Curr. Opin. Microbiol.* **31**, 161–168 (2016).
5. Beumer, A. & Robinson, J. B. A Broad-Host-Range, Generalized Transducing Phage (SN-T) Acquires 16S rRNA Genes from Different Genera of Bacteria. *Appl. Environ. Microbiol.* **71**, 8301–8304 (2005).
6. Göller, P. C. *et al.* Multi-species host range of staphylococcal phages isolated from wastewater. *Nat. Commun.* **12**, 6965 (2021).
7. Sullivan, M. B. Viromes, Not Gene Markers, for Studying Double-Stranded DNA Virus Communities. *J. Virol.* **89**, 2459–2461 (2015).
8. Drake, J. W. The Distribution of Rates of Spontaneous Mutation over Viruses, Prokaryotes, and Eukaryotes. *Ann. N. Y. Acad. Sci.* **870**, 100–107 (1999).
9. Peck, K. M. & Lauring, A. S. Complexities of Viral Mutation Rates. *J. Virol.* **92**, e01031-17 (2018).
10. O'Leary, N. A. *et al.* Reference sequence (RefSeq) database at NCBI: current status, taxonomic

expansion, and functional annotation. *Nucleic Acids Res.* **44**, D733–D745 (2016).

11.Ponsero, A. J. & Hurwitz, B. L. The Promises and Pitfalls of Machine Learning for Detecting Viruses in Aquatic Metagenomes. *Front. Microbiol.* **10**, (2019).

12. Guo, J. *et al.* VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. *Microbiome* **9**, 37 (2021).

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**, 90 (2020).

14. Hegarty, B. *et al.* A snapshot of the global drinking water virome: Diversity and metabolic potential vary with residual disinfectant use. *Water Res.* **218**, 118484 (2022).

15. Gregory, A. C. *et al.* Marine DNA Viral Macro- and Microdiversity from Pole to Pole. *Cell* **177**, 1109-1123.e14 (2019).

16. Rocha, U. N. da *et al.* MuDoGeR: Multi-Domain Genome Recovery from metagenomes made easy. 2022.06.21.496983 Preprint at https://doi.org/10.1101/2022.06.21.496983 (2022).

17. Tisza, M. J., Belford, A. K., Domínguez-Huerta, G., Bolduc, B. & Buck, C. B. Cenote-Taker 2 democratizes virus discovery and sequence annotation. *Virus Evol.* **7**, veaa100 (2021).

18. Nayfach, S. *et al.* CheckV assesses the quality and completeness of metagenome-assembled viral genomes. *Nat. Biotechnol.* **39**, 578–585 (2021).

19. Ren, J. *et al.* Identifying viruses from metagenomic data using deep learning. *Quant. Biol.* **8**, 64–77 (2020).

20. Czeczko, P., Greenway, S. C. & de Koning, A. P. J. EzMap: a simple pipeline for reproducible analysis of the human virome. *Bioinforma. Oxf. Engl.* **33**, 2573–2574 (2017).

21. Amgarten, D., Braga, L. P. P., da Silva, A. M. & Setubal, J. C. MARVEL, a Tool for Prediction of Bacteriophage Sequences in Metagenomic Bins. *Front. Genet.* **9**, (2018).

22. Antipov, D., Raiko, M., Lapidus, A. & Pevzner, P. A. MetaviralSPAdes: assembly of viruses from metagenomic data. *Bioinformatics* **36**, 4126–4129 (2020).

23. MetaPhinder—Identifying Bacteriophage Sequences in Metagenomic Data Sets | PLOS ONE. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0163111.

24. Deaton, J., Yu, F. B. & Quake, S. R. Mini-Metagenomics and Nucleotide Composition Aid the Identification and Host Association of Novel Bacteriophage Sequences. *Adv. Biosyst.* **3**, 1900108 (2019).

25. Arndt, D. *et al.* PHASTER: a better, faster version of the PHAST phage search tool. *Nucleic Acids Res.* **44**, W16–W21 (2016).

26. Starikova, E. V. *et al.* Phigaro: high-throughput prophage sequence annotation. *Bioinformatics* **36**, 3882–3884 (2020).

27. Fang, Z. *et al.* PPR-Meta: a tool for identifying phages and plasmids from metagenomic fragments using deep learning. *GigaScience* **8**, giz066 (2019).

28. Liu, F., Miao, Y., Liu, Y. & Hou, T. RNN-VirSeeker: A Deep Learning Method for Identification of Short Viral Sequences From Metagenomes. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **19**, 1840–1849 (2022).

29. 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 Res.* **48**, e121 (2020).

30. Li, Y. *et al.* VIP: an integrated pipeline for metagenomics of virus identification and discovery. *Sci. Rep.* **6**, 23774 (2016).

31. Tampuu, A., Bzhalava, Z., Dillner, J. & Vicente, R. ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples. *PLOS ONE* **14**, e0222271 (2019).

32. 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**, 69 (2017).

33. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biol.* **20**, 257 (2019).

34. Garretto, A., Hatzopoulos, T. & Putonti, C. virMine: automated detection of viral sequences from complex metagenomic samples. *PeerJ* **7**, e6695 (2019).

35. Zheng, T. *et al.* Mining, analyzing, and integrating viral signals from metagenomic data. *Microbiome* **7**, 42 (2019).

36. Abdelkareem, A. O., Khalil, M. I., Elaraby, M., Abbas, H. & Elbehery, A. H. A. VirNet: Deep attention model for viral reads identification. in *2018 13th International Conference on Computer Engineering and Systems (ICCES)* 623–626 (2018). doi:10.1109/ICCES.2018.8639400.

37. Wommack, K. E. *et al.* VIROME: a standard operating procedure for analysis of viral metagenome sequences. *Stand. Genomic Sci.* **6**, 421–433 (2012).

38. Rampelli, S. *et al.* ViromeScan: a new tool for metagenomic viral community profiling. *BMC Genomics* **17**, 165 (2016).

39. Roux, S., Enault, F., Hurwitz, B. L. & Sullivan, M. B. VirSorter: mining viral signal from microbial genomic data. *PeerJ* **3**, e985 (2015).

40. Miao, Y., Liu, F., Hou, T. & Liu, Y. Virtifier: a deep learning-based identifier for viral sequences from metagenomes. *Bioinformatics* **38**, 1216–1222 (2022).

41. Zhao, G. *et al.* VirusSeeker, a computational pipeline for virus discovery and virome composition analysis. *Virology* **503**, 21–30 (2017).

42. Glickman, C., Hendrix, J. & Strong, M. Simulation study and comparative evaluation of viral contiguous sequence identification tools. *BMC Bioinformatics* **22**, 329 (2021).

43. Ho, S. F. S., Wheeler, N. E., Millard, A. D. & van Schaik, W. Gauge your phage: benchmarking of bacteriophage identification tools in metagenomic sequencing data. *Microbiome* **11**, 84 (2023).

44. Menzel, P., Ng, K. L. & Krogh, A. Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nat. Commun.* **7**, 11257 (2016).

45. Edwards, R. A. & Rohwer, F. Viral metagenomics. *Nat. Rev. Microbiol.* **3**, 504–510 (2005).

46. Guo, J., Vik, D., Pratama, A. A., Roux, S. & Sullivan, M. Viral sequence identification SOP with VirSorter2. (2021).

47. Chicco, D. & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020).

48. Rose, R., Constantinides, B., Tapinos, A., Robertson, D. L. & Prosperi, M. Challenges in the analysis of viral metagenomes. *Virus Evol.* **2**, vew022 (2016).

49. Jansson, J. K. & Wu, R. Soil viral diversity, ecology and climate change. *Nat. Rev. Microbiol.* **21**, 296–311 (2023).

50. Suttle, C. A. Marine viruses — major players in the global ecosystem. *Nat. Rev. Microbiol.* **5**, 801–812 (2007).

51. Paez-Espino, D. *et al.* Uncovering Earth’s virome. *Nature* **536**, 425–430 (2016).

52. Langenfeld, K., Chin, K., Roy, A., Wigginton, K. & Duhaime, M. B. Comparison of ultrafiltration and iron chloride flocculation in the preparation of aquatic viromes from contrasting sample types. *PeerJ* **9**, e11111 (2021).

53. Elbehery, A. H. A. & Deng, L. Insights into the global freshwater virome. *Front. Microbiol.* **13**, (2022).

54. Moon, K. *et al.* Freshwater viral metagenome reveals novel and functional phage-borne antibiotic resistance genes. *Microbiome* **8**, 75 (2020).

55. Mukherjee, S., Huntemann, M., Ivanova, N., Kyrpides, N. C. & Pati, A. Large-scale contamination of microbial isolate genomes by Illumina PhiX control. *Stand. Genomic Sci.* **10**, 18 (2015).

56. Camargo, A. P. *et al.* You can move, but you can't hide: identification of mobile genetic elements with geNomad. 2023.03.05.531206 Preprint at https://doi.org/10.1101/2023.03.05.531206 (2023).

57. Gabrielli, M. *et al.* Identifying Eukaryotes and Factors Influencing Their Biogeography in Drinking Water Metagenomes. *Environ. Sci. Technol.* **57**, 3645–3660 (2023).