Published online 11 May 2022

Nucleic Acids Research, 2022, Vol. 50, No. 14 ***e83***
https://doi.org/10.1093/nar/gkac341

vRhyme enables binning of viral genomes from metagenomes

Kristopher Kieft [1,2], Alyssa Adams[1,3], Rauf Salamzade[2,4], Lindsay Kalan[4,5] and Karthik Anantharaman [1,*]

[1]Department of Bacteriology, University of Wisconsin–Madison, Madison, WI, USA, [2]Microbiology Doctoral Training Program, University of Wisconsin–Madison, Madison, WI, USA, [3]Computation and Informatics in Biology and Medicine, University of Wisconsin–Madison, Madison, WI, USA, [4]Department of Medical Microbiology and Immunology, University of Wisconsin–Madison, Madison, WI, USA and [5]Department of Medicine, University of Wisconsin–Madison, Madison, WI, USA

Received January 05, 2022; Revised April 17, 2022; Editorial Decision April 21, 2022; Accepted April 22, 2022

ABSTRACT

Genome binning has been essential for characterization of bacteria, archaea, and even eukaryotes from metagenomes. Yet, few approaches exist for viruses. We developed vRhyme, a fast and precise software for construction of viral metagenome-assembled genomes (vMAGs). vRhyme utilizes single- or multi-sample coverage effect size comparisons between scaffolds and employs supervised machine learning to identify nucleotide feature similarities, which are compiled into iterations of weighted networks and refined bins. To refine bins, vRhyme utilizes unique features of viral genomes, namely a protein redundancy scoring mechanism based on the observation that viruses seldom encode redundant genes. Using simulated viromes, we displayed superior performance of vRhyme compared to available binning tools in constructing more complete and uncontaminated vMAGs. When applied to 10,601 viral scaffolds from human skin, vRhyme advanced our understanding of resident viruses, highlighted by identification of a Herelleviridae vMAG comprised of 22 scaffolds, and another vMAG encoding a nitrate reductase metabolic gene, representing near-complete genomes post-binning. vRhyme will enable a convention of binning uncultivated viral genomes and has the potential to transform metagenome-based viral ecology.

INTRODUCTION

Viruses and bacteriophages (collectively termed viruses) are pervasive members of essentially all ecosystems. Viruses form a continuum of symbiotic interactions with their hosts, from lethal parasitism to essential mutualism (1–3). These interactions are known to impact biogeochemical and nutrient cycling processes, human health, infrastructure and industries and ecosystem community dynamics (4–7). As a result of the rising interest in viromics, the previously unknown members of the virosphere, the range in the encoded genetic potential of viruses, known viral diversity, and limits of viral genome sizes have been continuously expanding (8–12).

Metagenomic sequencing can be a mechanism to identify, recognize, understand, and even harness the information encoded on viral genomes. Most metagenomes will assemble into many short fragments (scaffolds or contigs) representing partial genome sequences. The process of binning is employed to group scaffolds into a putative genome, termed a metagenome-assembled genome (MAG). With the information encoded by a MAG, rather than individual scaffolds, stronger inferences of metabolic potential, phylogenies, taxonomy, and community interactions can be generated (13).

Conversely, viral scaffolds are typically not binned. Handling complex and often enigmatic viral scaffolds in metagenomes often poses computational challenges unique from microbes. One justification to not bin viruses is that their genomes are small relative to cellular organisms and the assumption that most scaffolds represent the majority, or the entirety, of an identifiable genome. For dsDNA viruses, the target of most viral metagenomes, genome sizes will have a general range of 20–200 kb, with the largest of viruses being 500–2,000 kb. Since the majority of scaffolds in most assembled metagenomes are <20 kb in length, it can be estimated that a single scaffold likely will not represent an entire viral genome. In fact, benchmarks have shown that viruses often do not assemble into a single scaffold (14,15). Further difficulties with binning viral genomes arise due to viruses not encoding universal single copy or marker genes, making a standardized approach for all viruses difficult to create. Additionally, studies incorporating many samples

*To whom correspondence should be addressed. Tel: +1 608 265 4537; Email: karthik@bact.wisc.edu

for co-abundance comparisons have traditionally been uncommon, and that viral populations are often comprised of highly heterogeneous genomes that result in fragmented assemblies.

Many software tools have been developed for binning bacterial, archaeal, and eukaryotic metagenomic scaffolds into MAGs (16–25). These tools employ a wide range of methodologies, mainly focusing on tetranucleotide frequencies and read coverage abundance variance comparisons between scaffolds. A significant portion of the tools tailored to bacteria and archaea also rely on identifying microbial single copy genes to inform the construction of bins along with completeness and contamination estimates. Some tools for binning microbes are suitable for binning viruses due to their independence from microbial single copy gene analysis, namely MetaBat2, VAMB, CONCOCT, and BinSanity. MetaBat2 uses a composite scoring system based on the geometric mean of tetranucleotide frequencies and coverage abundance of individual scaffolds to generate bins according to a weighted graph clustering algorithm (17). VAMB implements unsupervised deep learning variational autoencoders based on individual scaffold tetranucleotide frequencies and coverage abundance to generate bins by iterative medoid clustering (18,26). CONCOCT uses tetranucleotide frequencies and coverage abundance, reduced by multidimensional reduction, to cluster scaffolds into bins with Gaussian mixture models (27). BinSanity uses affinity propagation clustering based on coverage abundances to bin scaffolds, followed by bin refinement using tetranucleotide frequencies and GC content (24). Despite the abundance of tools for binning bacteria and archaea, there is a conspicuous dearth of tools available for binning viruses. Only one tool, CoCoNet (28), has thus far been developed for binning viral genomes from metagenomes (viral MAGs or vMAGs). CoCoNet implements an unsupervised deep learning neural network to identify shared tetranucleotide and coverage abundance patterns between scaffold pairs, followed by graph clustering of potential pairs into bins (28).

Here, we present vRhyme, a software tool that incorporates supervised machine learning based classification of diverse sequence feature compositions as well as read coverage abundance effect size comparisons to generate weighted networks of bins. vRhyme leverages unique features of viral genomes to optimize and refine the binning of vMAGs, including overcoming the lack of single copy genes by scoring protein redundancy based on the observation that viruses seldom encode redundant genes. vRhyme is capable of binning viruses from diverse families, host and source environment affiliations, varying states of genome fragmentation, and wide ranges of genome lengths. In benchmarking vRhyme, we show that it is fast, inclusive, and accurate in binning viral scaffolds, with low computational demands, in synthetic and natural metagenomes compared to other binning software. When applied to human skin metagenomes, we show that vRhyme enabled a more comprehensive analysis of shared viruses and viral features across a cohort of individuals, and likely better recapitulated natural systems. vRhyme is implemented in Python and is freely available for download at https://github.com/AnantharamanLab/vRhyme.

MATERIALS AND METHODS

Coverage processing

The input for read coverage information is variable: paired or unpaired short reads, SAM alignment file, BAM alignment file, or a pre-calculated coverage table. For short reads input, reads will be aligned to input scaffolds using either Bowtie2 (29) or BWA (30); Bowtie2 is run with the parameters –no-unal –no-discordant, the latter being for paired reads only, and BWA is run with the mem algorithm. All reads should be quality filtered before being used as input. The resulting SAM alignment file, or an input SAM alignment file, will be converted into BAM format using Samtools (31). BAM alignment files, either generated by the vRhyme pipeline or as user input, will then be processed. As such, any input combinations of short reads, SAM or BAM alignment files are compatible. BAM alignment files, if not already provided as input, are sorted and indexed using Samtools.

The Python package Pysam (https://github.com/pysam-developers/pysam) is then used to fetch aligned records within sorted and indexed BAM alignment files for processing and coverage calculations. First, aligned reads are filtered according to the percent identity alignment, as calculated by the sum of the number of gaps g and the number of mismatches m in the alignment divided by the length of the alignment l. The default is a 97% identity alignment.

$$\text{percent identity alignment} = \frac{l - g - m}{l} \cdot 100$$

Aligned reads passing the set threshold are used to calculate the total coverage of each nucleotide base per scaffold, inclusive of bases with a coverage of zero. Finally, the coverage values at the terminal ends of scaffolds are masked to increase coverage fidelity by considering erroneous read alignment at partial scaffold ends. The default is to ignore all coverage values within the first and last 150 bp of the scaffold. The average and standard deviation of coverage per scaffold is calculated according to respective, individual base coverages. All alignment filtering and coverage calculations are handled natively within vRhyme. This final step yields a coverage table comprised of the average and standard deviation of coverage per scaffold per input sample. This coverage table, or a user-generated table of the same format, can be used as input for vRhyme in place of reads or SAM/BAM alignment files.

Next, scaffold coverages across all k samples are pairwise compared using the effect size of coverage differences. First, all average coverages are increased by a pseudo-count of 0.1 to avoid coverages of zero (pseudo-counts are excluded from coverage table). Effect size is calculated by the Cohen's d effect size metric equation (32). Cohen's d is calculated as follows, where $\bar{X}_i$ and $\bar{X}_j$ are average read coverages and σ_i and σ_j are standard deviations of the coverages for a scaffold pair i and j:

$$d_{k,i,j} = \frac{\bar{X}_i - \bar{X}_j}{\sqrt{\frac{\sigma_i^2 + \sigma_j^2}{2}}}$$

For each pairwise comparison, an effect size value d_k is generated per sample k across all samples n. Values exceed-

ing the effect size threshold, set by vRhyme presets, generate an additive penalty weight p. The average effect size across all samples $\bar{X}_d$, with any added penalties, is normalized to the number of input samples, yielding a normalized effect size d', which considers higher statistical power to more sample comparisons:

$$\bar{X}_{d,i,j} = \frac{\sum_{k=1}^{n} d_{k,i,j}}{n} + p_{i,j}$$

$$d'_{i,j} = \frac{\bar{X}_{d,i,j}}{\log_{10}(n) + 1}$$

The normalized and penalized d' values are compared to a normalized preset effect size threshold and all pairwise comparisons passing the set criteria are considered as co-occurring by coverage. Any scaffold not found to co-occur with another is discarded. For computational efficiency, a pre-filter is applied where only the best (i.e. lowest d') n pairs per individual scaffold are retained, where n is '–max_edges' multiplied by 3.

Nucleotide processing

All co-occurring scaffolds by read coverage are compared by seven nucleotide content metrics. The pairwise distance calculations per metric are used as inputs to supervised machine learning models for classification. All nucleotide features and distances are calculated natively within vRhyme.

The first feature, codon usage (CU), is calculated from nucleotide open reading frames (i.e. genes). Predicted genes can be used as input, otherwise vRhyme will automate prediction using Prodigal (33) (-m -p meta). In-frame trinucleotide counts c for each of the 64 codons k (step of 3 bases) along a scaffold are divided by the total count of observed codons. The final codon, if representing a stop, is ignored. Counts are inclusive of zero counts but exclusive of ambiguous (e.g. N) bases. The following yields a CU frequency vector F_i for each codon k in scaffold i.

$$F_k = \frac{c_k}{\sum_{k=1}^{64} c_k}$$

$$F_i = (F_1,\ F_2,\ F_3 \ldots\ F_k)$$

The next three features (GC content, CpG content, and GC-skew) are calculated per scaffold from individual scaffold bases. GC content N_{gc} is calculated by the sum of all G and C bases, divided by the sum of all bases (A, T, C and G). CpG content N_{cpg} is calculated by the sum of all CG dinucleotides per scaffold (step of 1 base) divided by the sum of all bases. GC-skew N_{skew} is calculated by subtracting the total of C bases from the total G bases, divided by the sum of G and C bases.

$$N_{gc} = \frac{C+G}{C+G+A+T}$$

$$N_{cpg} = \frac{CG}{C+G+A+T}$$

$$N_{skew} = \frac{G-C}{G+C}$$

The last three features—relative tetranucleotide frequency (RTF), tetranucleotide usage deviation (TUD), and tetranucleotide zero'th order Markov method (ZOM) – are calculated from whole scaffold tetranucleotide frequencies (step of 1 base) of the forward and reverse strands (34). A total of 136 possible tetranucleotides are considered after combining identical, reverse complement, and palindromic sequences. Counts are inclusive of zero counts but exclusive of ambiguous (i.e. N) bases.

For RTF, all counts t for each of the 136 tetranucleotides k along a scaffold are divided by the total count of observed tetranucleotides. The following yields a tetranucleotide frequency vector T_i for each tetranucleotide k in scaffold i.

$$T_k = \frac{t_k}{\sum_{k=1}^{136} t_k}$$

$$T_i = (T_1,\ T_2,\ T_3 \ldots\ T_k)$$

For TUD, expected nucleotide frequencies E are first calculated by dividing the count of each base b by the sum of all bases in the scaffold. Next, observed counts per base O_b per tetranucleotide k are calculated by the sum of each base inclusive of zero counts. For each unique tetranucleotide, expected frequencies per base are raised to the power of observed frequencies multiplied by two to yield a deviation value D_b per base. The deviation values for all four bases are multiplied the count of total observed tetranucleotides and the count of the given tetranucleotide to yield a TUD value per tetranucleotide. The following yields a TUD frequency vector TUD_i for each tetranucleotide k in scaffold i.

$$E_b = \frac{b}{C+G+A+T} \text{ for } b = \text{A},\ \text{T},\ \text{C},\ \text{G}$$

$$O_b = \sum_{k=1}^{4} b \text{ for } b = \text{A},\ \text{T},\ \text{C},\ \text{G}$$

$$D_b = E_b^{(2 \cdot O_b)} \text{ for } b = \text{A},\ \text{T},\ \text{C},\ \text{G}$$

$$TUD_k = D_A \cdot D_T \cdot D_C \cdot D_G \cdot \sum_{k=1}^{136} t_k \cdot t_k$$

$$TUD_i = (TUD_1,\ TUD_2,\ TUD_3 \ldots\ TUD_k)$$

For ZOM, the same expected E_b nucleotide frequencies per base b are used. For each tetranucleotide k, the count t of the given tetranucleotide is divided by the product of each of the present tetranucleotide's bases' expected frequencies to yield a ZOM frequency vector ZOM_i for each tetranucleotide k in scaffold i.

$$ZOM_k = \frac{t_k}{E_{b_1} \cdot E_{b_2} \cdot E_{b_3} \cdot E_{b_4}}$$

$$ZOM_i = (ZOM_1,\ ZOM_2,\ ZOM_3 \ldots ZOM_k)$$

Pairwise distance calculations for GC, CpG and GC-skew are made by the absolute value difference in the respective metric's content between two scaffolds. For exam-

ple, the following is the pairwise distance P_{GC} in GC content between scaffolds i and j.

$$P_{i,j} = \left| GC_i - GC_j \right|$$

Pairwise distance calculations for CU, RTF, TUD and ZOM are made by cosine distances. For each value v_i and v_j, corresponding to the same tetranucleotide k, in frequency vectors of scaffolds i and j, with vector averages of $\bar{V}_i$ and $\bar{V}_j$, cosine similarity $S_{i,j}$ is calculated. Cosine distances between two scaffolds are calculated for CU, RTF, TUD and ZOM individually.

$$S_{i,j} = \frac{\sum_{k=1}^{n} \left(v_{i_k} \cdot v_{j_k} \right)}{\sqrt{\left(\sum_{k=1}^{n} \left(v_{i_k} \cdot \bar{V}_i \right) \right)^2 \cdot \left(\sum_{k=1}^{n} \left(v_{j_k} \cdot \bar{V}_j \right) \right)^2}}$$

The result of distance calculations is a vector $M_{i,j}$ of length seven for each pairwise comparison between scaffolds i and j.

$$M_{i,j} = \left(N_{GC},\ N_{CpG},\ N_{skew},\ S_{CU},\ S_{RTF},\ S_{TUD},\ S_{ZOM} \right)$$

Machine learning model training and testing

NCBI databases (RefSeq (35) and Genbank (36), release July 2019) were queried for 'prokaryotic virus' and genomes >10 kb in length were retained. In addition, the IMG/VR database (release July 2018) (37) was downloaded, and sequences were limited to a minimum length of 10 kb. For the IMG/VR dataset, VIBRANT (38) (v1.2.1, -virome) and CheckV (39) (v0.6.0) were used to obtain circular and/or complete sequences. The resulting NCBI and IMG/VR datasets were dereplicated by 95% identity using the method described here (–derep_only –derep_id 0.95 –frac 0.70 –method longest) and combined, resulting in a total of 11,881 putatively complete genomes. The sequences representing complete genomes in the combined dataset were split into non-overlapping fragments of 15 kb with a minimum length of 10 kb. A total of 39,105 fragments were generated for training and testing machine learning models, with 38,732 represented in the training and 30,618 represented in the testing datasets (Supplementary Figure S1a).

The machine learning models were generated based on the $M_{i,j}$ vectors described above using the generated 39,105 genome fragments. Filtering of pairwise comparisons before training and testing was made according to vRhyme default parameters (–max_gc 0.20 –min_kmer 0.60). The pairwise comparison matrix was split 75:25 for training and testing, respectively. Fragment pairs were labeled as 'same' or 'different' for supervised machine learning according to if the paired fragments originated from the same or different source genomes. An equal number (69,632) of 'same' and 'different' pairs were used for training by randomly dropping excess 'different' comparisons. For testing, a set of 38,685 'different' and 7,736 'same' pairs were used. There were no redundant pairs between the training and testing datasets.

Scikit-Learn (v0.24.2) (40) was used to generate machine learning models using a grid search approach to optimize parameters. Several models and algorithms were considered, including MLPClassifier, ExtraTrees, KNeighbors, SVC, Gradient Boost, Decision Tree and Random Forest classifiers. Iterative training and testing yielded MLPClassifier (alpha = 0.001, beta_1 = 0.7, beta_2 = 0.8, hidden_layer_sizes = (5,25,50,75,100,100,75,50,25,5), learning_rate_init = 0.0001, max_iter = 1250, n_iter_no_change = 15, tol = 1e-08) and ExtraTreesClassifier (max_depth = 10, max_features = 7, n_estimators = 1500) as the most robust.

Machine learning and network processing

Each scaffold pair is classified by the two machine learning models separately to yield two probability values of 'same', one per model. The probability values are averaged to yield $\bar{p}$. Any pair with $\bar{p}$ below the preset threshold is discarded. Then, d' calculated previously for the pair is divided by $\bar{p}$ to yield a network edge weight w.

$$w = \frac{d'}{\bar{p}}$$

Any pair with w below the preset threshold is retained for network clustering. As before, for computational efficiency, only the best (i.e. lowest w) n pairs per individual scaffold are retained, where n is '–max_edges'. Weighted networks, representing unrefined bins, are created where each node is a scaffold and each edge is a weighted connection between paired scaffolds. Networks are refined using MiniBatchK-Means implemented in Scikit-Learn with the following parameters: n_clusters = s+1, batch_size = h, max_iter = 100, max_no_improvement = 5, n_init = 5. Batch size h is 25% of the number of nodes with a minimum of 2 and maximum of 100. The number of clusters s is defined by the number of nodes with a clustering coefficient value below the preset constant 0.36 but not 0. For each node i, the clustering coefficient U_i is calculated as follows, where L_i is the degree of the node and R_i is the number of edges between the neighbors of i:

$$U_i = \frac{2 \cdot L_i}{R_i \cdot (R_i - 1)}$$

Refined networks are split into distinct, separate networks according to s. Here, each connected network represents a putative bin.

Score processing

Each binning iteration is given a score I according to protein redundancy, total bins, and the number of scaffolds binned. To calculate protein redundancy, all proteins within a bin are clustered using Mmseqs2 (41) (linclust –min-seq-id 0.5 -c 0.8 -e 0.01 –min-aln-len 50 –cluster-mode 0 –seq-id-mode 0 –alignment-mode 3 –cov-mode 5 –kmer-per-seq 75). Any proteins clustered within a bin, excluding those along the same scaffold, are considered redundant. The iteration with the maximum score is selected as the final representative. I is calculated as follows:

$$I_r = \sum_{bin=1}^{n} \frac{proteins\ clustered_{bin} - number\ of\ clusters_{bin}}{total\ proteins_{bin}}$$

$$I_s = \frac{scaffolds\ binned}{input\ scaffolds}$$

$$I_b = \frac{number\ of\ bins}{scaffolds\ binned}$$

$$I = I_s - {I_b}^2 - \left(3 \cdot \sqrt{2 \cdot I_r}\right)$$

Dereplication

vRhyme implements Nucmer (42) and MASH (43) for the dereplication of scaffolds. First, scaffolds are roughly grouped using MASH (sketch -k 31 -s 1000; dist) to reduce the pairwise comparison space. Next, all possible pairs of scaffolds within each resulting group are aligned using Nucmer (-c 1000 -b 1000 -g 1000). Regardless of the comparison method ('–method'), any pair of scaffolds with 100% identity over 100% coverage are first reduced to the longest representative. For all percent coverage calculations in dereplication, coverage is of the shortest scaffold. For '–method longest' the longest scaffold in pairs meeting the set percent identity (e.g. 97%) and percent coverage (e.g. 60%) thresholds is taken as the representative. For '–method composite', scaffold pairs meeting the percent identity and percent coverage thresholds are joined over the region of sequence overlap to yield artificially chimeric scaffolds. Any alignments exceeding the sensitivity values for merging over complex alignments, such as low identity scaffold ends without overlap, are not joined. After scaffold pairs are joined, identical cycles of MASH, Nucmer, and composite joining are completed until no further alignments are detected. For all methods, reverse complement sequence alignments are considered and adjusted accordingly.

Performance validation datasets and metrics

Scaffolds used to benchmark performance were acquired from nine separate publicly available datasets derived from eight unique metagenomes (one metagenome was split into two separate datasets). The metagenomes were acquired from marine (44,45), freshwater (46–48), human gut (49), and soil environments (50,51). Details on the studies, scaffolds, reads, and accession numbers can be found in Supplementary Table S1. Each dataset was processed separately. First, VIBRANT (v1.2.1) was used to predict viruses. From these viruses, VIBRANT and CheckV were used to identify circular scaffolds representing complete genomes. Next, scaffolds were dereplicated by 97% identity using the method described here (–derep_only –derep_id 0.97 –frac 0.70 –method longest). The non-redundant scaffolds were randomly fragmented into sequences ranging from 2 kb to 20 kb in length. A total of 999 scaffolds (i.e. putatively complete genomes) were used to generate 4,324 fragments of at least 2 kb in length. Full benchmarking was performed on the 4,324 fragments and validation of complete genome binning was performed on the 999 scaffolds representing complete genomes (Supplementary Figure S1b). Only 255 of the performance benchmarking fragments had significant sequence similarity to fragments used to train the machine learning models (Supplementary Figure S1c).

Since the circular scaffolds (sources) were estimated to be complete genomes, any of the fragments originating from the same source were expected to create a single bin, bins containing fragments from multiple sources were considered as contaminated, fragments from the same source in different bins were considered as split genomes, and fragments representing an entire source (singletons) were not expected to bin. The following equations are for genome- (source) and bin-based performance metrics, where B_e is the expected number of bins (i.e. sources with at least two fragments), B_g is the number of bins generated, G_e is the expected number of binned fragments (i.e. fragments representing B_e sources), B_o is the total number of bins containing a single source, G_t is the total number of fragments binned, G_b is the number of unique sources binned, G_o is the number of sources contained in a single bin, G_s is the total number of singletons, and G_p the number of binned singletons.

$$binned\ singletons = \frac{G_p}{G_s}$$

$$genome\ recall = \frac{G_t - G_s}{G_e}$$

$$genome\ precision = \frac{G_o - G_p}{G_b}$$

$$genome\ splitting = \frac{G_b - G_o}{G_b}$$

$$bin\ precision = \frac{G_o}{B_g}$$

$$bin\ contamination = \frac{B_g - B_o}{B_g}$$

$$genomes\ total = \frac{G_b}{B_e}$$

$$bins\ total = \frac{B_g}{B_e}$$

$$bins : genomes = \frac{B_g}{G_b - G_s}$$

$$genomes : bins = \frac{G_b - G_s}{B_g}$$

$$genomes\ score = \frac{2 \cdot \left(G_o - G_p\right)}{\left(2 \cdot \left(G_o - G_p\right)\right) + G_p + G_b - G_o}$$

$$bins\ score = \frac{2 \cdot G_o}{\left(2 \cdot G_o\right) + \left(B_g - B_o\right)}$$

To validate binning further, each pairwise connection between fragments within a bin was evaluated according to each fragment's nucleotide length. These standard performance metrics were evaluated per bin using true positive TP, true negative TN, false positive FP, and false negative

FN connections. The following equations are for pairwise nucleotide-based performance metrics:

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$specificity = \frac{TN}{TN + FP}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Performance benchmarking

The performance of vRhyme (v1.0.0) was compared to MetaBat2 (17) (v2.12.1, -s 4000 -m 2000), CONCOCT (27) (v1.0.0, -l 2000), VAMB (18) (v3.0.2, -i 2 -m 2000 -t 40), CoCoNet (28) (v1.0.0, –min-ctg-len 1000 –min-prevalence 1), and BinSanity (24) (v0.5.4, -x 2000). Additional binning tools, namely MaxBin2 (16), MyCC (19), SolidBin (20) and DASTool (22), perform microbial single copy gene analysis and were not applicable, or did not function, for viruses. For VAMB, the starting batch size had to be adjusted to accommodate the relatively small size of the input datasets, and all but three datasets failed to run. The coverage tables for each of the tools were generated from sorted BAM files using each tool's respective method, except for VAMB for which the same coverage table as MetaBat2 was used. The sorted BAM files were generated using Samtools (v1.13) with reads quality filtered by Sickle (v1.33) aligned by Bowtie2 (v2.3.5.1, –no-unal –no-discordant).

Metagenomic datasets and analyses

Publicly available metagenomes from marine (52), agricultural soil (53), and human skin (54) environments were used. Details on the studies, reads used, and accession numbers can be found in Supplementary Table S1. Viruses were predicted from each metagenome using VIBRANT and only the identified virus scaffolds were binned using vRhyme. For the human skin datasets, 270 metagenomes from a cohort of 34 individuals with eight body sites per individual were used (antecubital fossa (Af), alar crease (Al), back (Ba), nare (Na), occiput (Oc), toe-web space (Tw), umbilicus (Um) and volar forearm (Vf)). Reads were filtered for quality, adapters, and host-contamination as described previously (54) using fastp (55) (v0.21.0, –detect_adapter_for_pe) and KneadData (v0.8.0). MegaHit (56) (v1.2.9) was used to generate individual metagenomic assemblies for each sample, corresponding to the microbiome of a particular body site for a specific participant at a given timepoint. After predicting viruses, all viruses per body site were combined and dereplicated (–method longest) before binning.

It is important to note that for bins, scaffolds had to be linked with Ns in order to run CheckV analysis since there is no mode to input bins. For all benchmarking using CheckV, the tool was modified to run Prodigal with the -m flag to accommodate linking vMAGs and not predicting open reading frames across the appended strings of Ns connecting scaffolds. For taxonomy of the validation dataset, a publicly available custom reference database of NCBI viruses was used as previously described (57). In brief, DIAMOND (58) (v0.9.14) BLASTp (59) (v2.6.0) was used to identify the most likely taxonomic affiliation of a sequence.

Additional datasets and benchmarking

Additional publicly available datasets were used to assess the performance of vRhyme under different scenarios and conditions. To assess binning of related types of viruses within the same sample, a total of 101 publicly available crAssphage sequences (60) were dereplicated using vRhyme (–derep_id 0.97 –frac 0.70 –method longest) to 86 non-redundant scaffolds. The non-redundant scaffolds were randomly fragmented as described previously into 791 fragments. To assess binning of megaphages and eukaryotic viruses with large genomes, the 540 kb Prevotella phage Lak C1 (61) was randomly fragmented into 51 fragments, and four different eukaryotic viruses (62,63) with genome lengths ranging from 154 kb to 201 kb were each randomly fragmented into 11 to 19 fragments. To assess binning of active and dormant prophages, VIBRANT was used to predict prophage regions for 10 active prophages from three different hosts and 24 dormant prophages from five different hosts. Activity or dormancy was determined according to respective studies described elsewhere (64–66) and validated using PropagAtE (67) (v1.1.0). Whole prophage scaffolds from the same host genome were binned together. Details on the studies, reads used, scaffolds, and accession numbers can be found in Supplementary Table S1.

To validate protein redundancy, NCBI databases (RefSeq and Genbank, release July 2019) were queried for 'prokaryotic virus' as before and genomes greater than 3 kb in length were retained. Likewise, NCBI databases (RefSeq and Genbank, release September 2021) were queried for 'eukaryotic virus' and genomes greater than 20 kb in length were retained. Proteins were predicted using Prodigal (-p meta) for 15,238 prokaryotic and 557 eukaryotic viruses. Protein redundancy was calculated per genome based on the method described for vRhyme, with the exception that proteins could be redundant if encoded along the same scaffold.

Effect of number of samples

The effect of the number of input samples on vRhyme performance was done by stepwise increasing the number of BAM files used to calculate coverage from one to the maximum number of samples for a given dataset. To do this, samples were arranged in descending order, starting at the sample with the greatest total coverage across all scaffolds and were stepwise combined, ending with the sample with the lowest coverage.

Visualizations

All plots and visualizations were done using Matplotlib (68) (v3.2.0) and Seaborn (69) (v0.11.0). Genome alignment visualizations were made using EasyFig (70) (v2.2.2) and Geneious Prime 2019.0.3. Genome alignments to identify percent sequence identity were made using progressiveMauve (71) (development snapshot 2015–02-25). vConTACT2 (72) (v0.9.19, –rel-mode Diamond –db 'None' –pcs-mode MCL –vcs-mode ClusterONE, ClusterONE (73) v1.0) was used to construct protein clustering networks and visualized using Cytoscape (74) (v3.7.2).

RESULTS

vRhyme overview and workflow

The vRhyme workflow is done in five steps: read coverage processing, sequence feature extraction, supervised machine learning, iterative network clustering, and bin scoring (Figure 1). The base input to vRhyme are the assembled scaffolds or contigs to be binned (hereafter scaffolds) with a set minimum size of 2 kb. For optimal results, only virome scaffolds or predicted virus scaffolds should be used as input, though vRhyme can function with the input of an entire metagenome. An initial dereplication step to remove redundant input scaffolds is optional. Next, scaffolds are compared pairwise by read coverage composition per sample, which is a proxy for relative abundance. vRhyme performs optimally with an input of multiple samples (i.e. coverage files) for more robust coverage co-occurrence estimations, but it will function with a single sample input with a minor decrease in performance. Statistically dissimilar scaffolds by coverage composition are screened out and the remaining potential pairs are compared by nucleotide feature similarity. Seven total nucleotide and gene features are used to classify pairs as similar versus dissimilar using two supervised machine learning models (decision trees and neural network). Following this step, potential connections are made between scaffolds based on similarity in read coverage and nucleotide features. These connections are used to create weighted networks that are further refined into genome bins using KMeans clustering. The entire process of read coverage comparison, nucleotide feature machine learning, and weighted network refinement is performed over several *binning iterations* in parallel. vRhyme has 15 built-in presets of thresholds for Cohen's *d*, machine learning model probabilities, and network edge weights. The number of presets used is equivalent to the number of binning iterations completed. A list of all presets and their hierarchy can be found in Supplementary Table S2. Each bin within all binning iterations is scored according to protein redundancy, a proxy for contamination, and the best binning iteration by sequences binned, bins generated, and redundancy metrics is selected. The bins within this best binning iteration are reported along with relevant metadata, including number of members and total protein redundancy. Alternative binning iterations are likewise saved if manual inspection and selection of a different iteration is desired.

Assessment of binning quality

To evaluate vRhyme, we first benchmarked vRhyme against reference datasets and compared the performance to several available binning tools, all of which are built for microbes. Many binning tools and wrapper software were not suitable for viral binning due to reliance on microbial single copy genes. We were able to successfully compare vRhyme to MetaBat2 (17), VAMB (18), CoCoNet (28), CONCOCT (27) and BinSanity (24) on nine datasets curated from metagenomic data (see Methods). The nine datasets were comprised of 999 non-redundant and putatively complete viral genomes that were split into 4,324 sequence fragments of varying lengths between 2 kb and 20 kb. Of these, 1,118 fragments were less than 5 kb, 1,361 were greater than 5 kb and <10 kb, and the remaining 1,854 were greater than 10 kb. The average length was 9.4 kb. Although these fragments were derived from datasets not represented in the machine learning training dataset, we first verified that the fragments were distinct and would not result in a bias associated with an overfitted machine learning model. Based on BLASTn similarity at 70% identity and 60% overlap, only 255 (∼6%) of the 4,324 fragments were represented in the machine learning model training dataset, with all but two of the represented fragments being from the same human gut dataset.

A total of 17 different evaluation metrics were used, including five traditional metrics for recall, precision, accuracy, specificity, and F1 score (Figure 2). The five traditional metrics were calculated according to the true positive, true negative, false positive, and false negative rates of binning fragments together from the same or different source genomes (Supplementary Table S3a). Note that the machine learning models were not benchmarked individually since performance is measured based on the entire pipeline. vRhyme yielded the highest F1 score, the harmonic average of precision and recall, with an average of 0.87 across all nine datasets. MetaBat2 and VAMB performed equally with *F*1 scores of 0.81 and 0.82, respectively, but importantly VAMB only successfully binned three of the nine datasets due to input size requirements. vRhyme likewise yielded the highest, or equal to highest, average precision (0.94), accuracy (0.90) and specificity (0.96) compared to all benchmarked tools. Compared to MetaBat2, VAMB and CoCoNet, vRhyme likewise yielded the greatest average recall (0.80). CONCOCT and BinSanity yielded the greatest average recall values (0.96 and 0.91, respectively) but at the expense of precision (0.45 and 0.44, respectively). At least for viral genomes, CONCOCT and BinSanity were found to not be suitable binning options. VAMB had suitable performance on the three datasets with enough input sequences, but VAMB is likely not an option for many applications of binning viral genomes due to requiring many input sequences (e.g. tens of thousands (18)) for optimal performance. Based on these metrics, vRhyme performed exceptionally in binning viral genomes but did not considerably improve on the performance of MetaBat2.

The remaining 12 evaluation metrics were calculated according to complete genomes and individual bins. These included evaluating if genomes were placed into a single

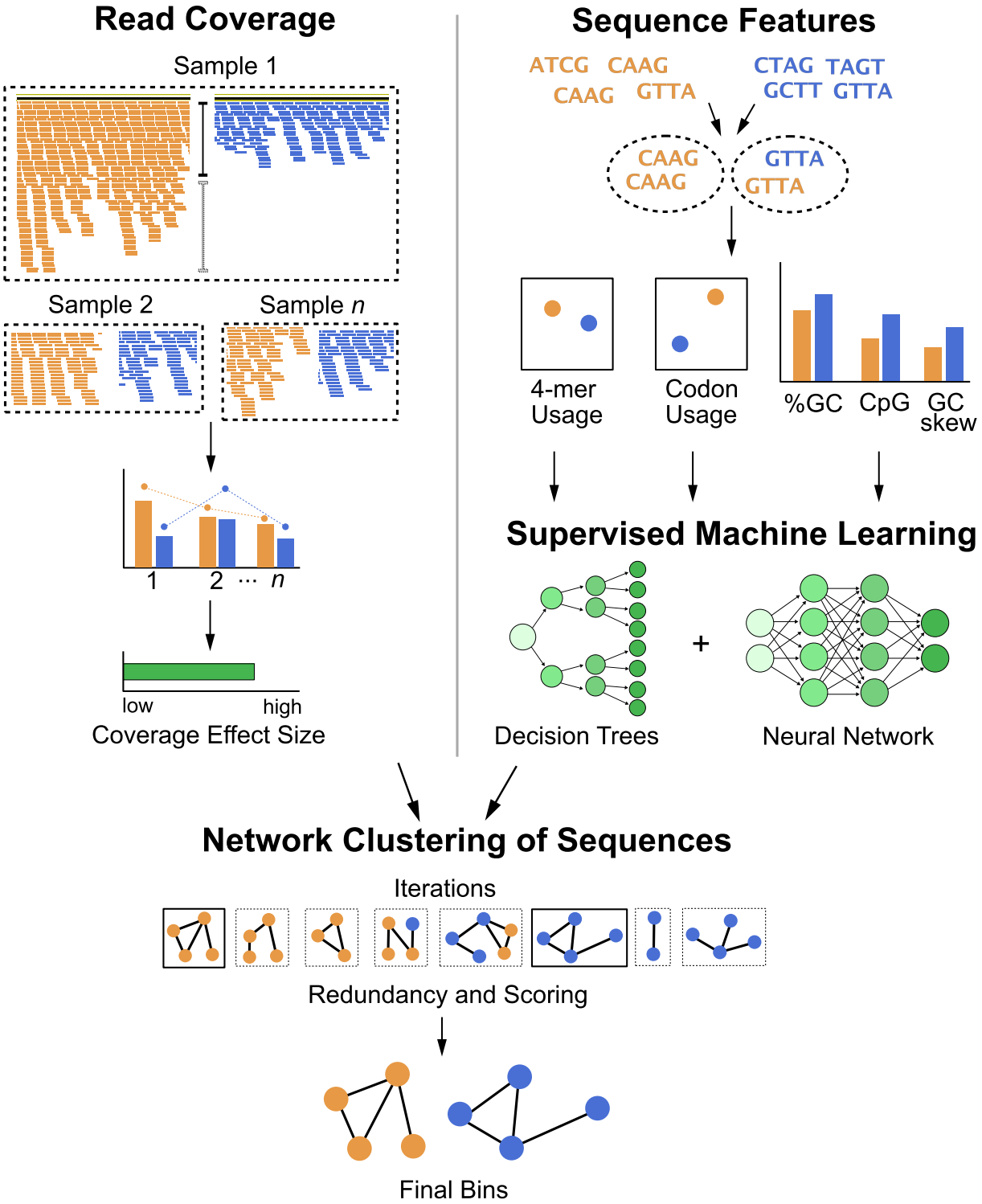

Figure 1. Flowchart of vRhyme workflow and methodology. Scaffolds are compared pairwise by read coverage effect size differences using single or multiple samples (top-left), followed by sequence feature distance comparisons (top-right). Multiple iterations of network clustering of putative bins are generated with edge weights representing normalized coverage effect size and supervised machine learning probabilities of sequence feature similarity (center). The bins are refined by KMeans clustering, and the best set of bins from a single iteration are identified after identifying protein redundancy and scoring (bottom).

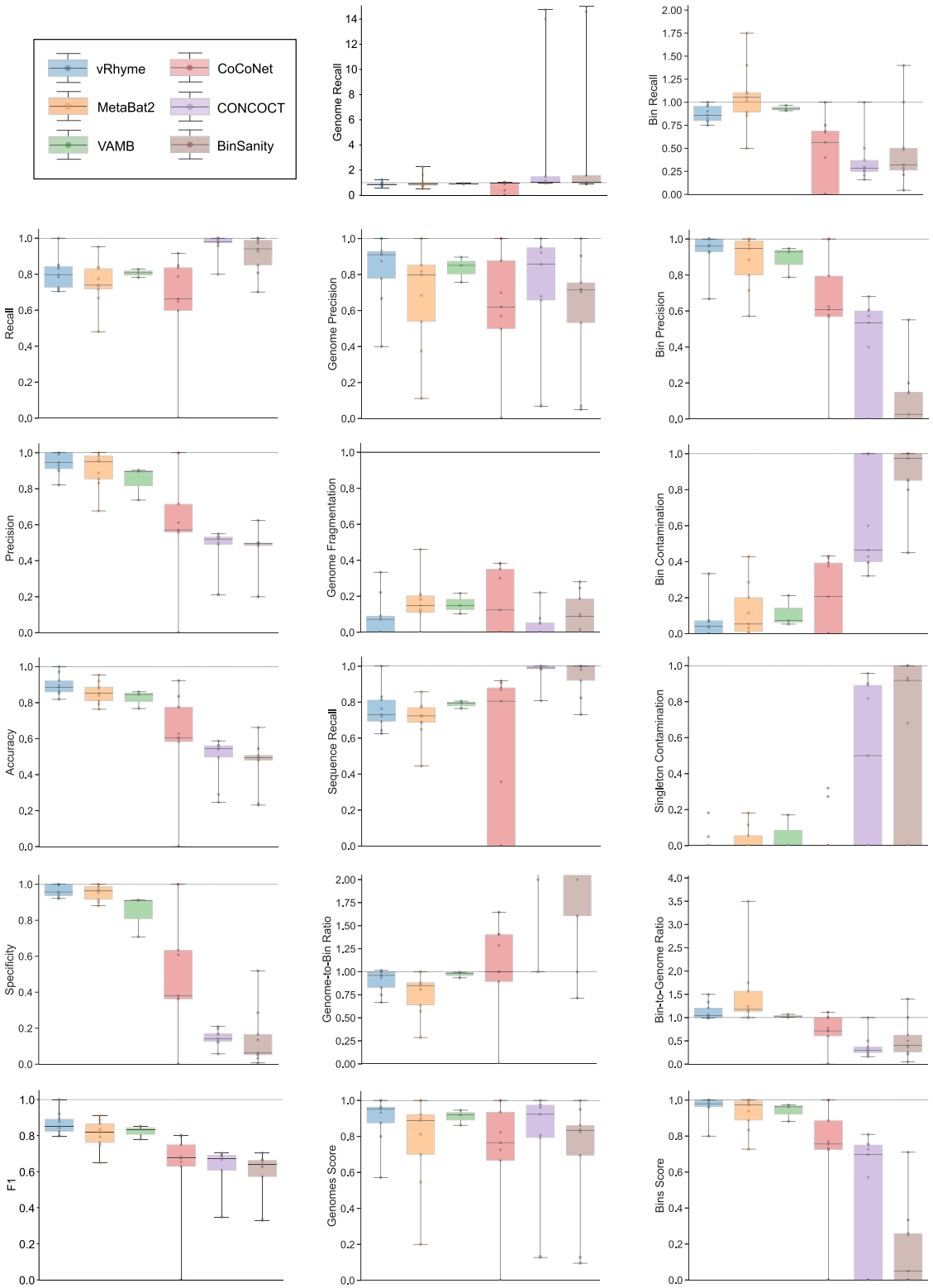

Figure 2. Benchmarking performance metrics of vRhyme compared to MetaBat2, VAMB, CoCoNet, CONCOCT and BinSanity. Each boxplot represents the results of nine different datasets, except for VAMB in which three datasets are shown. In total, 999 non-redundant genomes artificially split into 4,324 sequence fragments are shown. For some plots, a dotted line is shown at 1.0 to indicate optimal performance. CONCOCT and BinSanity are partially shown on the Genome-to-Bin Ratio plot for better visualization; each yielded an average ratio >2.0.

or separate bins, and if bins contained fragments from a single or multiple source genomes. These metrics were better able to show the distinct performance of vRhyme compared to the other tools (Supplementary Table S3b). Namely, vRhyme was better able to reduce the following: placement of genomes into separate bins, placement of fragments from multiple source genomes into a single bin, and binning circular scaffolds representing entire genomes. Importantly, this was not at the cost of reduced fragment recall by vRhyme. To combine these metrics, we created a genome score and bin score that considered recall and precision as a substitution for F1 score. For genome scores and bin scores, respectively, vRhyme (0.89 and 0.96) outperformed, or was equivalent to, MetaBat2 (0.77 and 0.93) and VAMB (0.90 and 0.93). Again, it is important to note that VAMB only successfully binned three of the nine datasets. For CoCoNet, CONCOCT, BinSanity, genome scores (0.66, 0.74 and 0.70, respectively) and bin scores (0.65, 0.48 and 0.18, respectively) reflected the propensity to 'over bin' distinct genomes together into one bin. CoCoNet did not bin any sequence in two of the datasets, and after removal of these zero-values, the average genome score and bin score for CoCoNet both increased to 0.84.

Furthermore, we evaluated how well vRhyme bins compare to the input, unfragmented genomes. First, using CheckV (39) we show a distinct change in genome completeness estimation in the binned versus unbinned sequence fragments. vRhyme was able to recapitulate the completeness of the input genomes (Figure 3A). This is supported by a similar observation in the length of the input genomes versus the bins (Figure 3B). Moreover, we estimated the taxonomy of the input genomes, fragments, and binned vMAGs. We identified a distinct decrease in the ability to identify taxonomy of the fragments, which were rescued by binning (Figure 3C). The identifiable difference in the vMAGs is a lack of Microviridae. Yet, this is to be expected since the small genome size of Microviridae (<10 kb) typically results in near-complete scaffolds that appropriately remain unbinned. Finally, we evaluated whether vRhyme could distinguish the source scaffolds. To do this, each of the nine datasets were binned, but the scaffolds were not fragmented. The expected result is that none of the circular scaffolds should bin together. Although vRhyme did bin ∼11% of the whole scaffolds, it was a marked improvement on VAMB and CoCoNet (Figure 3D).

Benchmarking vRhyme on marine viromes

We next applied vRhyme to the Global Ocean Virome 2 (GOV2) database (52) and compared the results to MetaBat2 and CoCoNet. For metagenomic datasets such as GOV2 the expected number of scaffolds to bin and the number of bins is unknown. First, all scaffolds from the GOV2 database were limited to scaffolds at least 5 kb in length and dereplicated by 98% identity. Of the 108,947 input scaffolds, vRhyme binned 56,642 scaffolds into 13,175 bins, MetaBat2 binned 57,800 scaffolds into 11,826 bins, and CoCoNet binned 91,842 scaffolds into 9,914 bins. Despite the number of bins generated being relatively similar, the number of scaffolds binned was quite different. However, vRhyme yielded 15,106 redundant proteins whereas MetaBat2 (29,334) and CoCoNet (71,364) yielded more, indicating that vRhyme was likely more precise and generated fewer contaminated bins (Figure 4A). In support of this, vRhyme generated 1,266 bins with two or more redundant proteins whereas MetaBat2 (1,648) and CoCoNet (2,743) generated more. When these likely contaminated bins were removed, vRhyme binned 48.251 scaffolds into 11.909 bins, MetaBat2 binned 33,351 scaffolds into 10,178 bins, and CoCoNet binned 35,380 scaffolds into 7,171 bins (Figure 4B). Based on protein redundancy, vRhyme was capable of binning far more viral scaffolds and generating more bins of low contamination compared to MetaBat2 and CoCoNet. Note, we identified bins with 'low contamination' to be 0–1 redundant proteins based on a benchmark of prokaryotic and eukaryotic viral genomes from NCBI databases (Supplementary Figure S2). Contamination was not estimated using CheckV as that metric does not consider contamination of multiple viral genomes, but rather contamination of non-viral sequences.

We also estimated the completeness of the 11,909 low contamination vRhyme bins and the individual 48,251 scaffolds that generated those bins using CheckV. The binned scaffolds individually yielded 25,969 (53.8%) completeness values with an average of 14% complete, 79 estimated to be 100% complete, 22,282 (46.2%) with 'NA' completeness, and 27,295 (56.6%) with 'no viral genes detected'. The scaffolds within each bin, after being linked into vMAGs, yielded 8,393 (70.5%) completeness values with an average of 48% complete, 775 estimated to be 100% complete, 3,516 (29.5%) with 'NA' completeness, and 4,039 (33.9%) with 'no viral genes detected' (Figure 4C–E). There was an increase in the number of vMAGs (195, 1.6%) versus individual scaffolds (16, 0.03%) that were estimated to be 'longer than expected', potentially due to a marginal rate of multiple genomes being binned into a single vMAG (Figure 4F). Overall, vRhyme generated vMAGs with greater average completeness to aid in downstream analyses and interpretations, even with high complexity or large datasets such as GOV2.

Discovery of vMAGs in human skin metagenomes

To demonstrate the ability of vRhyme to aid metagenome analyses and discovery, we applied vRhyme to 270 human skin metagenomes (54). Viruses were predicted from a cohort of 34 individuals with eight body sites (*Af, Al, Ba, Na, Oc, Tw, Um* and *Vf*) sampled per individual (see Methods). From all individuals, 10,601 viral scaffolds were identified and binned, across eight different body sites individually, into a total of 849 vMAGs representing 2,794 viral scaffolds. Although bins with redundant proteins may in fact be a single genome or partially redundant copies of a single genome, we ignored all vMAGs with greater than one redundant protein for analysis to yield 762 vMAGs representing 2,413 viral scaffolds, leaving the remaining 8,188 as discrete viral scaffolds (Supplementary Table S4) (Figure 5A). The taxonomic classification of UViGs pre-binning, UViGs and low redundancy vMAGs post-binning, and vMAGs-only displayed that most bins were constructed of genomes from the class Caudoviricetes, similar to the observed tax-

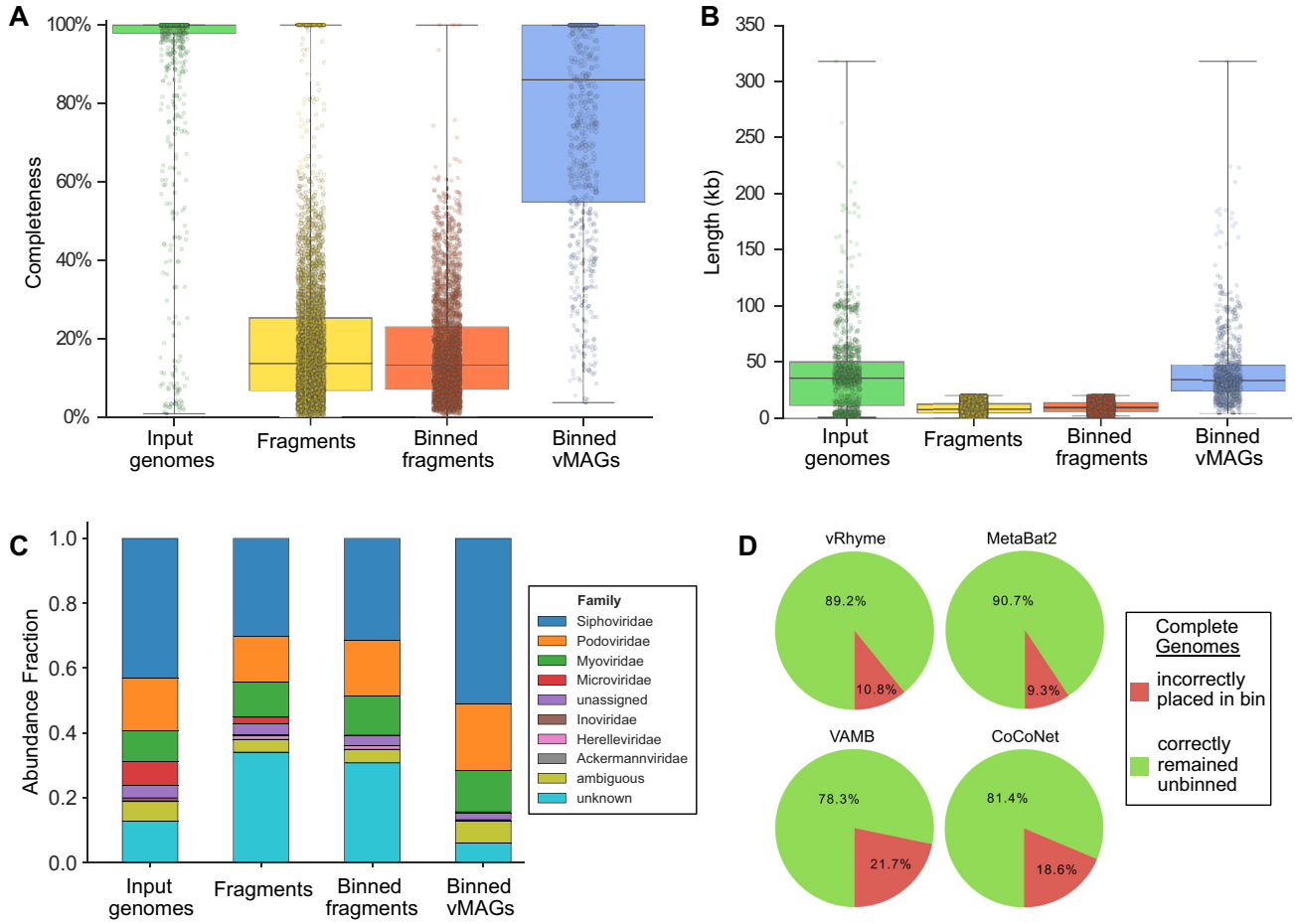

Figure 3. Impact of binning with vRhyme on the benchmarking datasets. For (A–C), the putatively complete unsplit input genomes, generated sequence fragments, binning sequence fragments, and vRhyme bins (vMAGs) are compared. (**A**) Estimation of genome completeness using CheckV. (**B**) Sequence or vMAG nucleotide length. For (A, B) each dot represents a single sequence or vMAG. (**C**) Estimation of taxonomy at the family level using a custom analysis script. 'unassigned' represents a taxonomic classification to a group with an unassigned family, 'ambiguous' represents equal assignment to multiple families (typically Caudoviricetes), and 'unknown' represents the inability to make a prediction. (**D**) Evaluation of vRhyme, MetaBat2, VAMB, and CoCoNet for the binning of complete genomes. The expectation is that complete genomes should remain unbinned as uncultivated virus genomes (UViGs).

onomy pre-binning (Supplementary Figure S3). The bins were comprised of an average of 3.2 scaffolds each. In total we identified seven bins, representing separate body sites, that were present across at least 30 individuals (Figure 5B). In addition, two bins of unique characteristics were identified and examined in detail.

The first such bin contained 22 members (Tw bin 8), more than what would be expected for a viral bin, and aligned to a reference Herelleviridae phage (Staphylococcus phage phiSA_BS2) (Figure 5c). Herelleviridae infecting abundant *Staphylococcus* on the skin are likely to be highly relevant to skin ecology and disease (75). Before binning, each of the 22 members were identified by CheckV as low-quality genome fragments with individual completeness estimations ranging from 1.8% to 7.1%. The fragments averaged 5.2 kb in length and ranged from 2.6 kb to 10.0 kb. After binning, the final bin was 115 kb in length and identified as a high-quality genome with 100% completeness by CheckV. The reference phage genome is 143 kb, suggesting the true completeness of the bin is likely 80% to 100%. All CheckV results for the skin metagenomes can be found in Supplementary Table S5.

The second bin of interest contained four members (Vf bin 113), with one encoding a nitrate reductase (*narG*) auxiliary metabolic gene (AMG) (Figure 5D). The *narG* was positioned as the last gene on a scaffold, and conventional approaches for AMG validation would suggest discarding the AMG as likely bacterial contamination. However, binning aided in the validation of the AMG as likely to be correct. The first line of evidence was the lack of any integrase or lysogenic viral signatures on any of the four binned scaffolds, suggesting the AMG is not from bacterial contamination resulting from host integration. Second, alignment of all four scaffolds to the nearest reference genome (Siphoviridae isolate ctiXA4) displayed that the AMG was situated at the intersection of two scaffolds within the genome rather than at a genome end. CheckV identified each member as low-quality with completeness values of 11.6% to 28.0% for the respective 7.4 kb to 16.8 kb scaffolds. The bin was estimated to be of medium-quality with a completeness of 74.9%, or 92% based on the length of the closest reference genome. Moreover, one of the four scaffolds lacked characteristic viral annotations to aid with manual inspection or analyses such as phylogeny, yet binning with the other

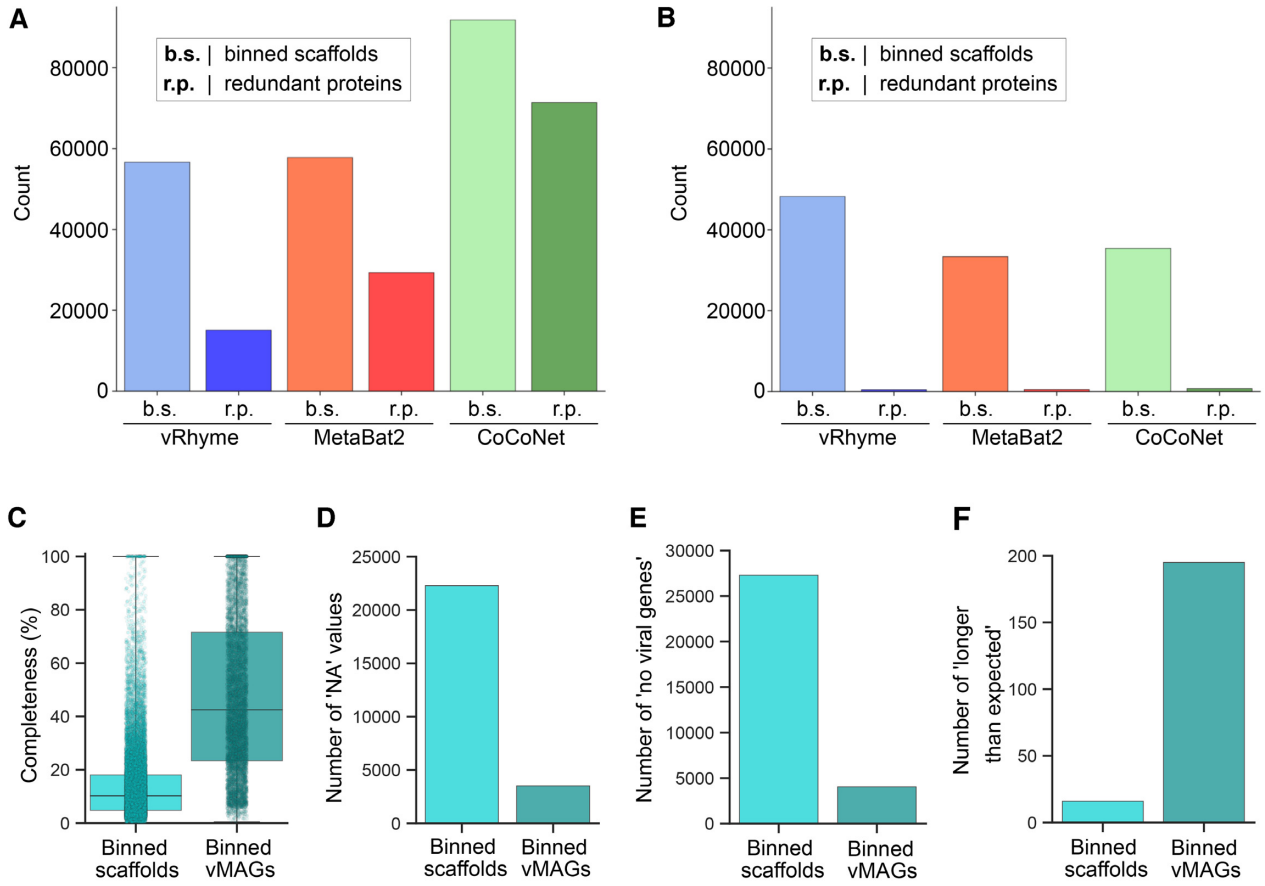

Figure 4. Benchmark binning and genome completeness evaluation of GOV2. Comparison of vRhyme, MetaBat2, and CoCoNet (**A**) raw results and (**B**) low contamination filtering results by the number of scaffolds binned and identified redundancy. For vRhyme only, CheckV was used to identify (**C**) the estimated completeness values, (**D**) number of 'NA' completeness values, (**E**) number of 'no viral genes' scaffolds/vMAGs and (**F**) number of 'longer than expected' scaffolds/vMAGs for the low contamination results of individual binned scaffolds as well as vMAGs.

scaffolds containing viral hallmark and nucleotide replication annotations was able to validate the scaffold as viral and place it in better genomic context for analysis. Therefore, binning was able to not only generate a more complete sequence, but also validate the presence of an understudied and ecologically important AMG. Using vConTACT2 (72), we clustered all of the individual, unprocessed viral scaffolds (Figure 5E) in addition to the bin with the complete binning results (low-contamination bins plus unbinned scaffolds) (Figure 5F). Clustering of the individual scaffolds placed all four scaffolds of the bin into a single cluster distinct from other groups, yet as anticipated none of the scaffolds of the bin were connected. Clustering of the binning results yielded more connections between scaffolds and vMAGs and better placed the bin within evolutionary and community relationship contexts. Complete vConTACT2 networks can be found in Supplementary Figures S4 and S5.

DISCUSSION

Binning viral scaffolds into vMAGs is uncommon, with most or all remaining as discrete virus operational taxonomic units (vOTUs) or uncultivated virus genomes (UViGs) (76). We believe adopting a more genome-centric approach for UViGs will enable innovative discoveries, such as the construction of large or highly heterogenous viral genomes that often assemble into dissimilar fragments. Here, we have presented vRhyme and demonstrate that the 'one scaffold, one virus' convention can skew interpretations of a virosphere and the interactions of its viral community members. To address this, vRhyme enables the binning of viral genomes into vMAGs using a virus-centric approach, unique from existing binning software, in an easy to use and reproducible command line tool.

In addition to performance benchmarks on artificial and real metagenomes, we evaluated the robustness of vRhyme by binning artificially fragmented NCLDV, megaphage, large eukaryotic viruses, crAssphage, active and inactive integrated prophages, and microbial genomes (Supplementary Information, Supplementary Table S6). vRhyme was largely capable of precisely binning these unique and complex viral datasets. However, notable exceptions were difficulties with separating multiple inactive (non-replicating) prophages from the same host genome as well as binning non-viral genomes, though the latter was an anticipated limitation. Moreover, we displayed that vRhyme is efficient and likely precise in binning large and complex datasets using GOV2 and agricultural soil viromes (53) (Supplementary Information, Supplementary Table S7). In total, we

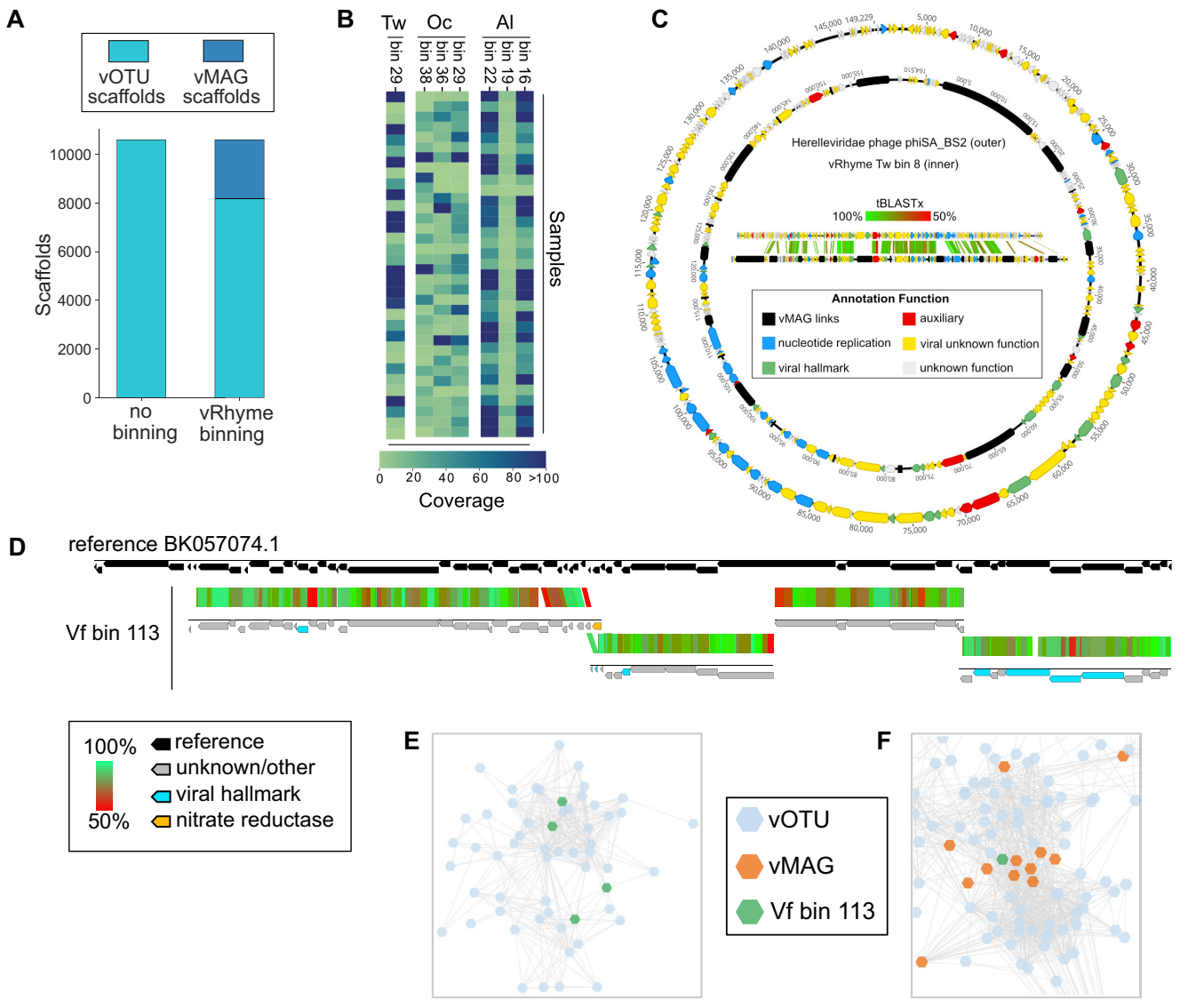

Figure 5. Binning improves and expands the analysis of viruses from human skin. (**A**) Comparison of the number of original viral scaffolds identified across all individuals before and after binning. (**B**) Heatmap of coverage for the seven common bins per individual. (**C**) Genome visualization and alignment of Herelleviridae reference phiSA_BS2 (outer) and Tw bin 8 (inner). Each arrow represents a predicted open reading frame and black bars are artificial connections between vMAG scaffolds. (**D**) Alignment of vRhyme Vf bin 113 to the closest reference virus Siphoviridae isolate ctiXA4 (BK057074.1). Each of the four scaffolds were independently aligned by tBLASTx similarity. The *narG* AMG is labeled in yellow and viral hallmark annotations are labeled in light blue. (**E**) Representative cluster from all input viral scaffolds generated by vConTACT2, with the four Vf bin 113 scaffolds labeled in green. There are no connections between any of the four green scaffolds. Each dot represents a single scaffold. (**F**) Partial network from all vRhyme binned and unbinned viral scaffolds generated by vConTACT2, with vMAG bins labeled in orange and Vf bin 113 in green. For (E, F), Complete network diagrams can be found in Supplementary Figures S4 and S5.

hope that with the availability of vRhyme as a reliable binning tool, vMAG construction will become a common practice and adopted into existing frameworks of studying viral ecology, host associations, community interactions, evolution, and biogeochemical cycling.

To further evaluate the computational capabilities of vRhyme or potential restraints, we assessed the effect of the coverage calculation methods, the number of input coverage samples and the effect of user-modifiable parameters on performance, as well as the runtime, memory usage and reproducibility of binning (Supplementary Information). We found that vRhyme performs optimally with multiple input samples for more robust coverage variance comparisons, though the optimal value depends on how the dataset or metagenome was constructed (Supplementary Table S8, Figure S6). For example, a metagenome assembled from a single, standalone sample may perform suitably. As for modifying parameters, vRhyme likely will yield optimal results with the default settings due to the coverage calculation method employed and built-in binning iterations (Supplementary Table S9, Figure S7). Furthermore, the runtime of vRhyme for average sized viral datasets was on the scale of seconds. The GOV2 dataset, the largest dataset evaluated, finished in 93 min with 2.3 GB of memory using 15

CPU threads (Supplementary Table S10, Figure S8). Lastly, the methods employed by vRhyme allow it to be fully reproducible. Overall, we found the necessary requirements to be relatively low and even possible on personal laptop systems.

There are several important considerations in the binning of vMAGs that are unique from microbial MAGs. First, any viral scaffold not contained within a bin (vMAG) should be considered as a vOTU or UViG. This aligns with the 'one scaffold, one virus' convention which is likely true for many viral genomes, especially circular and complete genomes. In the skin datasets presented here, ∼23% of the viral scaffolds were binned into low contamination vMAGs and the remaining ∼77% should still be utilized in analyses as discrete scaffolds. Second, an entire metagenome can be used as input to vRhyme, or viral binning in general, with the caveat that contamination of bins with non-viral sequences may be higher with the added advantage that fewer viral scaffolds may be missed. For example, many phage genomes are arranged in cassettes such that structural, nucleotide replication, lysis and auxiliary genes form distinct regions. If these regions were to assemble into separate scaffolds, virus identification may only identify a portion of the scaffolds, such as missing an auxiliary region, whereas binning may place them all together into a single vMAG. When applied to a synthetic dataset of predominately non-viral sequences, MetaBat2 performed better than vRhyme (Supplementary Information, Supplementary Table S11). Third, accurate read coverage profiles are crucial for accurate binning. This is true for all binning software that depend on differential coverage and is especially true for distinguishing bins of integrated prophages from a single host population. vMAGs representing prophages generated by vRhyme will likely represent the greatest fraction of redundant, contaminated bins.

DATA AVAILABILITY

vRhyme and all auxiliary scripts are freely available as open-source Python code at https://github.com/AnantharamanLab/vRhyme.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank members of the Anantharaman laboratory at the University of Wisconsin-Madison for helpful feedback and discussions.

Author contributions: K.K. and K.A. designed the study. K.K., A.A. and R.S. developed code and conducted bioinformatic analyses. K.K. and K.A drafted the manuscript. All authors (K.K., A.A., R.S., L.K. and K.A.) reviewed the results and approved the manuscript.

FUNDING

National Institute of General Medical Sciences of the National Institutes of Health [R35GM143024 to K.A., R35GM137828, U19AI142720 to L.K.]; A.A. was funded by a University of Wisconsin-Madison CIBM postdoctoral traineeship from the National Library of Medicine [T15LM007359]; K.K. was supported by a Wisconsin Distinguished Graduate Fellowship Award from the University of Wisconsin-Madison; William H. Peterson Fellowship Award from the Department of Bacteriology, University of Wisconsin-Madison. Funding for open access charge: NIH Grant funds (to K.A.).

Conflict of interest statement. None declared.

REFERENCES

1. Drew,G.C., Stevens,E.J. and King,K.C. (2021) Microbial evolution and transitions along the parasite–mutualist continuum. *Nat. Rev. Microbiol.*, **19**, 623–638.
2. Roossinck,M.J. (2015) Move over, bacteria! Viruses make their mark as mutualistic microbial symbionts. *J. Virol.*, **89**, 6532–6535.
3. Barr,J.J. (2019) Missing a phage: unraveling tripartite symbioses within the human gut. *Msystems*, **4**, e00105-19.
4. Hurwitz,B.L. and U'Ren,J.M. (2016) Viral metabolic reprogramming in marine ecosystems. *Curr. Opin. Microbiol.*, **31**, 161–168.
5. Howard-Varona,C., Lindback,M.M., Bastien,G.E., Solonenko,N., Zayed,A.A., Jang,H., Andreopoulos,B., Brewer,H.M., Rio,T.G.del, Adkins,J.N. *et al.* (2020) Phage-specific metabolic reprogramming of virocells. *ISME J.*, **14**, 881–895.
6. Kieft,K., Breister,A.M., Huss,P., Linz,A.M., Zanetakos,E., Zhou,Z., Rahlff,J., Esser,S.P., Probst,A.J., Raman,S. *et al.* (2021) Virus-associated organosulfur metabolism in human and environmental systems. *Cell Rep.*, **36**, 109471.
7. Barr,J.J., Auro,R., Furlan,M., Whiteson,K.L., Erb,M.L., Pogliano,J., Stotland,A., Wolkowicz,R., Cutting,A.S., Doran,K.S. *et al.* (2013) Bacteriophage adhering to mucus provide a non-host-derived immunity. *Proc. Natl. Acad. Sci. U.S.A.*, **110**, 10771–10776.
8. Al-Shayeb,B., Sachdeva,R., Chen,L.-X., Ward,F., Munk,P., Devoto,A., Castelle,C.J., Olm,M.R., Bouma-Gregson,K., Amano,Y. *et al.* (2020) Clades of huge phages from across Earth's ecosystems. *Nature*, **578**, 425–431.
9. Paez-Espino,D., Eloe-Fadrosh,E.A., Pavlopoulos,G.A., Thomas,A.D., Huntemann,M., Mikhailova,N., Rubin,E., Ivanova,N.N. and Kyrpides,N.C. (2016) Uncovering Earth's virome. *Nature*, **536**, 425–430.
10. Tisza,M.J. and Buck,C.B. (2021) A catalog of tens of thousands of viruses from human metagenomes reveals hidden associations with chronic diseases. *Proc. Natl. Acad. Sci. U.S.A.*, **118**, e2023202118.
11. Roux,S., Krupovic,M., Daly,R.A., Borges,A.L., Nayfach,S., Schulz,F., Sharrar,A., Carnevali,P.B.M., Cheng,J.-F., Ivanova,N.N. *et al.* (2019) Cryptic inoviruses revealed as pervasive in bacteria and archaea across Earth's biomes. *Nat. Microbiol.*, **4**, 1895–1906.
12. Dutilh,B.E., Cassman,N., McNair,K., Sanchez,S.E., Silva,G.G.Z., Boling,L., Barr,J.J., Speth,D.R., Seguritan,V., Aziz,R.K. *et al.* (2014) A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes. *Nat. Commun.*, **5**, 4498.
13. Bowers,R.M., Kyrpides,N.C., Stepanauskas,R., Harmon-Smith,M., Doud,D., Reddy,T.B.K., Schulz,F., Jarett,J., Rivers,A.R., Eloe-Fadrosh,E.A. *et al.* (2017) Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. *Nat. Biotechnol.*, **35**, 725–731.
14. Roux,S., Emerson,J.B., Eloe-Fadrosh,E.A. and Sullivan,M.B. (2017) Benchmarking viromics: an in silico evaluation of metagenome-enabled estimates of viral community composition and diversity. *PeerJ*, **5**, e3817.
15. Schulz,F., Andreani,J., Francis,R., Boudjemaa,H., Khalil,J.Y.B., Lee,J., Scola,B.L. and Woyke,T. (2020) Advantages and limits of metagenomic assembly and binning of a giant virus. *Msystems*, **5**, e00048-20.
16. Wu,Y.-W., Tang,Y.-H., Tringe,S.G., Simmons,B.A. and Singer,S.W. (2014) MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. *Microbiome*, **2**, 26.
17. Kang,D.D., Li,F., Kirton,E., Thomas,A., Egan,R., An,H. and Wang,Z. (2019) MetaBAT 2: an adaptive binning algorithm for

robust and efficient genome reconstruction from metagenome assemblies. *PeerJ*, **7**, e7359.
18. Nissen,J.N., Johansen,J., Allesøe,R.L., Sønderby,C.K., Armenteros,J.J.A., Grønbech,C.H., Jensen,L.J., Nielsen,H.B., Petersen,T.N., Winther,O. *et al.* (2021) Improved metagenome binning and assembly using deep variational autoencoders. *Nat. Biotechnol.*, **39**, 555–560.
19. Lin,H.-H. and Liao,Y.-C. (2016) Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. *Sci. Rep.*, **6**, 24175.
20. Wang,Z., Wang,Z., Lu,Y.Y., Sun,F. and Zhu,S. (2019) SolidBin: improving metagenome binning with semi-supervised normalized cut. *Bioinformatics*, **35**, 4229–4238.
21. Mallawaarachchi,V., Wickramarachchi,A. and Lin,Y. (2020) GraphBin: refined binning of metagenomic contigs using assembly graphs. *Bioinformatics*, **36**, 3307–3313.
22. Sieber,C.M.K., Probst,A.J., Sharrar,A., Thomas,B.C., Hess,M., Tringe,S.G. and Banfield,J.F. (2018) Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. *Nat. Microbiol.*, **3**, 836–843.
23. Uritskiy,G.V., DiRuggiero,J. and Taylor,J. (2018) MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome*, **6**, 158.
24. Graham,E.D., Heidelberg,J.F. and Tully,B.J. (2017) BinSanity: unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation. *PeerJ*, **5**, e3035.
25. West,P.T., Probst,A.J., Grigoriev,I.V., Thomas,B.C. and Banfield,J.F. (2018) Genome-reconstruction for eukaryotes from complex natural microbial communities. *Genome Res.*, **28**, 569–580.
26. Johansen,J., Plichta,D.R., Nissen,J.N., Jespersen,M.L., Shah,S.A., Deng,L., Stokholm,J., Bisgaard,H., Nielsen,D.S., Sørensen,S.J. *et al.* (2022) Genome binning of viral entities from bulk metagenomics data. *Nat. Commun.*, **13**, 965.
27. Alneberg,J., Bjarnason,B.S., de Bruijn,I., Schirmer,M., Quick,J., Ijaz,U.Z., Lahti,L., Loman,N.J., Andersson,A.F. and Quince,C. (2014) Binning metagenomic contigs by coverage and composition. *Nat. Methods*, **11**, 1144–1146.
28. Arisdakessian,C.G., Nigro,O.D., Steward,G.F., Poisson,G. and Belcaid,M. (2021) CoCoNet: an efficient deep learning tool for viral metagenome binning. *Bioinformatics*, **37**, 2803–2810.
29. Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–359.
30. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, **25**, 1754–1760.
31. Li,H., Handsaker,B., Wysoker,A., Fennell,T., Ruan,J., Homer,N., Marth,G., Abecasis,G., Durbin,R. and 1000 Genome Project Data Processing Subgroup1000 Genome Project Data Processing Subgroup (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.
32. Cohen,J. (2013) In: *Statistical Power Analysis for the Behavioral Sciences*. Academic Press, NY.
33. Hyatt,D., Chen,G.-L., LoCascio,P.F., Land,M.L., Larimer,F.W. and Hauser,L.J. (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinf.*, **11**, 119.
34. Siranosian,B., Perera,S., Williams,E., Ye,C., de Graffenried,C. and Shank,P. (2015) Tetranucleotide usage highlights genomic heterogeneity among mycobacteriophages. *F1000Res*, **4**, 36.
35. O'Leary,N.A., Wright,M.W., Brister,J.R., Ciufo,S., Haddad,D., McVeigh,R., Rajput,B., Robbertse,B., Smith-White,B., Ako-Adjei,D. *et al.* (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.*, **44**, D733–D745.
36. Clark,K., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J. and Sayers,E.W. (2016) GenBank. *Nucleic Acids Res.*, **44**, D67–D72.
37. Paez-Espino,D., Chen,I.-M.A., Palaniappan,K., Ratner,A., Chu,K., Szeto,E., Pillay,M., Huang,J., Markowitz,V.M., Nielsen,T. *et al.* (2017) IMG/VR: a database of cultured and uncultured DNA viruses and retroviruses. *Nucleic Acids Res.*, **45**, D457–D465.
38. Kieft,K., Zhou,Z. and Anantharaman,K. (2020) VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences. *Microbiome*, **8**, 90.
39. Nayfach,S., Camargo,A.P., Schulz,F., Eloe-Fadrosh,E., Roux,S. and Kyrpides,N.C. (2021) CheckV assesses the quality and completeness of metagenome-assembled viral genomes. *Nat. Biotechnol.*, **39**, 578–585.
40. Pedregosa,F., Varoquaux,G., Gramfort,A., Michel,V., Thirion,B., Grisel,O., Blondel,M., Prettenhofer,P., Weiss,R., Dubourg,V. *et al.* (2011) Scikit-learn: machine learning in python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
41. Steinegger,M. and Söding,J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, **35**, 1026–1028.
42. Marçais,G., Delcher,A.L., Phillippy,A.M., Coston,R., Salzberg,S.L. and Zimin,A. (2018) MUMmer4: a fast and versatile genome alignment system. *PLoS Comput. Biol.*, **14**, e1005944.
43. Ondov,B.D., Treangen,T.J., Melsted,P., Mallonee,A.B., Bergman,N.H., Koren,S. and Phillippy,A.M. (2016) Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.*, **17**, 132.
44. Anantharaman,K., Duhaime,M.B., Breier,J.A., Wendt,K.A., Toner,B.M. and Dick,G.J. (2014) Sulfur oxidation genes in diverse deep-sea viruses. *Science*, **344**, 757–760.
45. Li,M., Baker,B.J., Anantharaman,K., Jain,S., Breier,J.A. and Dick,G.J. (2015) Genomic and transcriptomic evidence for scavenging of diverse organic compounds by widespread deep-sea archaea. *Nat. Commun.*, **6**, 8933.
46. Tran,P.Q., Bachand,S.C., McIntyre,P.B., Kraemer,B.M., Vadeboncoeur,Y., Kimirei,I.A., Tamatamah,R., McMahon,K.D. and Anantharaman,K. (2021) Depth-discrete metagenomics reveals the roles of microbes in biogeochemical cycling in the tropical freshwater Lake Tanganyika. *ISME J.*, **15**, 1971–1986.
47. Okazaki,Y., Nishimura,Y., Yoshida,T., Ogata,H. and Nakano,S. (2019) Genome-resolved viral and cellular metagenomes revealed potential key virus-host interactions in a deep freshwater lake. *Environ. Microbiol.*, **21**, 4740–4754.
48. Coutinho,F.H., Cabello-Yeves,P.J., Gonzalez-Serrano,R., Rosselli,R., López-Pérez,M., Zemskaya,T.I., Zakharenko,A.S., Ivanov,V.G. and Rodriguez-Valera,F. (2020) New viral biogeochemical roles revealed through metagenomic analysis of Lake Baikal. *Microbiome*, **8**, 163.
49. He,Q., Gao,Y., Jie,Z., Yu,X., Laursen,J.M., Xiao,L., Li,Y., Li,L., Zhang,F., Feng,Q. *et al.* (2017) Two distinct metacommunities characterize the gut microbiota in Crohn's disease patients. *Gigascience*, **6**, 1–11.
50. Trubl,G., Roux,S., Solonenko,N., Li,Y.-F., Bolduc,B., Rodríguez-Ramos,J., Eloe-Fadrosh,E.A., Rich,V.I. and Sullivan,M.B. (2019) Towards optimized viral metagenomes for double-stranded and single-stranded DNA viruses from challenging soils. *PeerJ*, **7**, e7265.
51. Woodcroft,B.J., Singleton,C.M., Boyd,J.A., Evans,P.N., Emerson,J.B., Zayed,A.A.F., Hoelzle,R.D., Lamberton,T.O., McCalley,C.K., Hodgkins,S.B. *et al.* (2018) Genome-centric view of carbon processing in thawing permafrost. *Nature*, **560**, 49–54.
52. Gregory,A.C., Zayed,A.A., Conceição-Neto,N., Temperton,B., Bolduc,B., Alberti,A., Ardyna,M., Arkhipova,K., Carmichael,M., Cruaud,C. *et al.* (2019) Marine DNA viral Macro- and Microdiversity from pole to pole. *Cell*, **177**, 1109–1123.
53. Santos-Medellin,C., Zinke,L.A., Horst,A.M., Gelardi,D.L., Parikh,S.J. and Emerson,J.B. (2021) Viromes outperform total metagenomes in revealing the spatiotemporal patterns of agricultural soil viral communities. *ISME J.*, **15**, 1956–1970.
54. Swaney,M.H., Sandstrom,S. and Kalan,L.R. (2021) Cobamide sharing drives skin microbiome dynamics. bioRxiv doi: https://doi.org/10.1101/2020.12.02.407395, 10 November 2021, preprint: not peer reviewed.
55. Chen,S., Zhou,Y., Chen,Y. and Gu,J. (2018) fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*, **34**, i884–i890.
56. Li,D., Liu,C.-M., Luo,R., Sadakane,K. and Lam,T.-W. (2015) MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, **31**, 1674–1676.
57. Kieft,K., Zhou,Z., Anderson,R.E., Buchan,A., Campbell,B.J., Hallam,S.J., Hess,M., Sullivan,M.B., Walsh,D.A., Roux,S. *et al.* (2021) Ecology of inorganic sulfur auxiliary metabolism in widespread bacteriophages. *Nat. Commun.*, **12**, 3503.
58. Buchfink,B., Xie,C. and Huson,D.H. (2015) Fast and sensitive protein alignment using DIAMOND. *Nat. Methods*, **12**, 59–60.

59. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
60. Norman,J.M., Handley,S.A., Baldridge,M.T., Droit,L., Liu,C.Y., Keller,B.C., Kambal,A., Monaco,C.L., Zhao,G., Fleshner,P. *et al.* (2015) Disease-Specific alterations in the enteric virome in inflammatory bowel disease. *Cell*, **160**, 447–460.
61. Devoto,A.E., Santini,J.M., Olm,M.R., Anantharaman,K., Munk,P., Tung,J., Archie,E.A., Turnbaugh,P.J., Seed,K.D., Blekhman,R. *et al.* (2019) Megaphages infect Prevotella and variants are widespread in gut microbiomes. *Nat. Microbiol.*, **4**, 693–700.
62. Israeli,O., Cohen-Gihon,I., Zvi,A., Shifman,O., Melamed,S., Paran,N., Laskar-Levy,O. and Beth-Din,A. (2019) Complete genome sequence of the first camelpox virus case diagnosed in Israel. *Microbiol. Resour. Announc.*, **8**, e00671-19.
63. Caro-Vegas,C., Sellers,S., Host,K.M., Seltzer,J., Landis,J., Fischer,W.A., Damania,B. and Dittmer,D.P. (2020) Runaway Kaposi Sarcoma-associated Herpesvirus Replication correlates with systemic IL-10 levels. *Virology*, **539**, 18–25.
64. Hertel,R., Rodríguez,D.P., Hollensteiner,J., Dietrich,S., Leimbach,A., Hoppert,M., Liesegang,H. and Volland,S. (2015) Genome-Based identification of active prophage regions by next generation sequencing in Bacillus licheniformis DSM13. *PLoS One*, **10**, e0120759.
65. Gutiérrez,R., Markus,B., Carstens Marques de Sousa,K., Marcos-Hadad,E., Mugasimangalam,R.C., Nachum-Biala,Y., Hawlena,H., Covo,S. and Harrus,S. (2018) Prophage-Driven genomic structural changes promote bartonella vertical evolution. *Genome Biol. Evol.*, **10**, 3089–3103.
66. Ho,C.-H., Stanton-Cook,M., Beatson,S.A., Bansal,N. and Turner,M.S. (2016) Stability of active prophages in industrial Lactococcus lactis strains in the presence of heat, acid, osmotic, oxidative and antibiotic stressors. *Int. J. Food Microbiol.*, **220**, 26–32.
67. Kieft,K. and Anantharaman,K. (2022) Deciphering active prophages from metagenomes. *mSystems*, **7**, e00084-22.
68. Hunter,J.D. (2007) Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90–95.
69. Waskom,M.L. (2021) seaborn: statistical data visualization. *J. Open Source Software*, **6**, 3021.
70. Sullivan,M.J., Petty,N.K. and Beatson,S.A. (2011) Easyfig: a genome comparison visualizer. *Bioinformatics*, **27**, 1009–1010.
71. Darling,A.E., Mau,B. and Perna,N.T. (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. *PLoS One*, **5**, e11147.
72. Jang,H.B., Bolduc,B., Zablocki,O., Kuhn,J.H., Roux,S., Adriaenssens,E.M., Brister,J.R., Kropinski,A.M., Krupovic,M., Lavigne,R. *et al.* (2019) Taxonomic assignment of uncultivated prokaryotic virus genomes is enabled by gene-sharing networks. *Nat. Biotechnol.*, **37**, 632–639.
73. Nepusz,T., Yu,H. and Paccanaro,A. (2012) Detecting overlapping protein complexes in protein-protein interaction networks. *Nat. Methods*, **9**, 471–472.
74. Shannon,P. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.*, **13**, 2498–2504.
75. Byrd,A.L., Belkaid,Y. and Segre,J.A. (2018) The human skin microbiome. *Nat. Rev. Microbiol.*, **16**, 143–155.
76. Roux,S., Adriaenssens,E.M., Dutilh,B.E., Koonin,E.V., Kropinski,A.M., Krupovic,M., Kuhn,J.H., Lavigne,R., Brister,J.R., Varsani,A. *et al.* (2019) Minimum information about an uncultivated virus genome (MIUViG). *Nat. Biotechnol.*, **37**, 29–37.