













Genome analysis

Phables: from fragmented assemblies to high-quality bacteriophage genomes

Vijini Mallawaarachchi ^{1,*}, Michael J. Roach ¹, Przemyslaw Decewicz ^{1,2},
Bhavya Papudeshi ¹, Sarah K. Giles ¹, Susanna R. Grigson ¹, George Bouras ^{3,4},
Ryan D. Hesse ¹, Laura K. Inglis ¹, Abbey L.K. Hutton ¹, Elizabeth A. Dinsdale ¹,
Robert A. Edwards ¹

¹Flinders Accelerator for Microbiome Exploration, College of Science and Engineering, Flinders University, Adelaide, South Australia 5042, Australia

²Department of Environmental Microbiology and Biotechnology, Institute of Microbiology, Faculty of Biology, University of Warsaw, Warsaw 02-096, Poland;

³Adelaide Medical School, Faculty of Health and Medical Sciences, The University of Adelaide, Adelaide, South Australia 5005, Australia

⁴The Department of Surgery—Otolaryngology Head and Neck Surgery, Central Adelaide Local Health Network, Adelaide, South Australia 5000, Australia

*Corresponding author. Flinders Accelerator for Microbiome Exploration, College of Science and Engineering, Flinders University, Bedford Park, Adelaide, South Australia 5042, Australia. E-mail: vijini.mallawaarachchi@flinders.edu.au

Associate Editor: Peter Robinson

Abstract

Motivation: Microbial communities have a profound impact on both human health and various environments. Viruses infecting bacteria, known as bacteriophages or phages, play a key role in modulating bacterial communities within environments. High-quality phage genome sequences are essential for advancing our understanding of phage biology, enabling comparative genomics studies and developing phage-based diagnostic tools. Most available viral identification tools consider individual sequences to determine whether they are of viral origin. As a result of challenges in viral assembly, fragmentation of genomes can occur, and existing tools may recover incomplete genome fragments. Therefore, the identification and characterization of novel phage genomes remain a challenge, leading to the need of improved approaches for phage genome recovery.

Results: We introduce Phables, a new computational method to resolve phage genomes from fragmented viral metagenome assemblies. Phables identifies phage-like components in the assembly graph, models each component as a flow network, and uses graph algorithms and flow decomposition techniques to identify genomic paths. Experimental results of viral metagenomic samples obtained from different environments show that Phables recovers on average over 49% more high-quality phage genomes compared to existing viral identification tools. Furthermore, Phables can resolve variant phage genomes with over 99% average nucleotide identity, a distinction that existing tools are unable to make.

Availability and implementation: Phables is available on GitHub at <https://github.com/Vini2/phables>.

1 Introduction

Bacteriophages (hereafter ‘phages’) are viruses that infect bacteria, which influence microbial ecology and help modulate microbial communities (Edwards and Rohwer 2005, Rodriguez-Valera *et al.* 2009). Phages are considered the most abundant biological entity on earth, totalling an estimated 10^{31} particles (Comeau *et al.* 2008). Since their discovery by Frederick Twort in 1915 (Twort 1915), phages have been isolated from many diverse environments (Keen 2015). When sequencing technologies were first developed, phage genomes were the first to be sequenced due to their relatively small genome size (Sanger *et al.* 1977). With the advent of second-generation sequencing technologies, the first metagenomic samples to be sequenced were phages (Breitbart *et al.* 2002). The availability of advanced sequencing technologies has facilitated the investigation of the effects of phages on the functions of microbial communities,

especially in the human body’s niche areas. For example, phages residing in the human gut have a strong influence on human health (Łusiak-Szelachowska *et al.* 2017) and impact gastrointestinal diseases such as inflammatory bowel disease (IBD) (Norman *et al.* 2015). To date, our understanding of the diversity of phages is limited, as most have not been cultured due to the inherent difficulty of recovering phages from their natural environments. Although countless millions of phage species are thought to exist, only 26 048 complete phage genomes have been sequenced according to the INfrastructure for a PHAGE REference Database (INPHARED) (Cook *et al.* 2021) (as of the September 2023 update).

Metagenomics has enabled the application of modern sequencing techniques for the culture-independent study of microbial communities (Hugenholtz *et al.* 1998). Metagenomic sequencing provides a multitude of sequencing reads from the genetic material in environmental samples that are composed

of a mixture of prokaryotic, eukaryotic, and viral species. Metagenomic analysis pipelines start by assembling sequencing reads from metagenomic samples into longer contiguous sequences that are used in downstream analyses. Most metagenome assemblers (Peng *et al.* 2011, Namiki *et al.* 2012, Li *et al.* 2015, Nurk *et al.* 2017) use ‘de Bruijn graphs’ (Pevzner *et al.* 2001) as the primary data structure where they break sequencing reads into smaller pieces of length k , known as k -mers, and represent $(k - 1)$ -mers as vertices and k -mers as edges. After performing several simplification steps, the final ‘assembly graph’ represents sequences as vertices and connection information between these sequences as edges (Nurk *et al.* 2017, Mallawaarachchi *et al.* 2020a). Non-branching paths in the assembly graph (paths where all vertices have an in-degree and out-degree of one, except for the first and last vertices) are referred to as ‘unitigs’ (Kecelioglu and Myers 1995). Unitigs are entirely consistent with the read data and belong to the final genome(s). Assemblers condense unitigs into individual vertices and resolve longer optimized paths through the branches into contiguous sequences known as ‘contigs’ (Bankevich *et al.* 2012). As the contextual and contiguity information of reads is lost in de Bruijn graphs, mutations in metagenomes with high strain diversity appear as ‘bubbles’ in the assembly graph where a vertex has multiple outgoing edges (branches) which eventually converge as incoming edges into another vertex (Pevzner *et al.* 2001, 2004). Assemblers consider these bubbles as errors and consider one path of the bubble corresponding to the dominant strain (Bankevich *et al.* 2012) or terminate contigs prematurely (Li *et al.* 2015). Moreover, most metagenome assemblers are designed and optimized for bacterial genomes and fail to recover viral populations with low coverage and genomic repeats (Roux *et al.* 2017, Sutton *et al.* 2019). However, previous studies have shown that contigs connected to each other are more likely to belong to the same genome (Mallawaarachchi *et al.* 2020a,b, 2021). Hence, the assembly graph retains important connectivity and neighbourhood information within fragmented assemblies. This concept has been successfully applied to develop tools such as GraphMB (Lamurias *et al.* 2022), MetaCoAG (Mallawaarachchi and Lin 2022a,b), and RepBin (Xue *et al.* 2022), where the assembly graphs are utilized in conjunction with taxonomy-independent metagenomic binning methods to recover high-quality metagenome-assembled genomes (hereafter MAGs) of bacterial genomes. Moreover, assembly graphs have been used for bacterial strain resolution in metagenomic data (Quince *et al.* 2021). However, limited studies have been conducted to resolve phage genomes in metagenomic data, particularly in viral-enriched metagenomes.

Computational tools have enabled large-scale studies to recover novel phages entirely from metagenomic sequencing data (Simmonds *et al.* 2017) and gain insights into interactions with their hosts (Nayfach *et al.* 2021b, Roach *et al.* 2022b, Hesse *et al.* 2022). While exciting progress has been made towards identifying new phages, viral dark matter remains vast. Current methods are either too slow or result in inaccurate or incomplete phage genomes. Generating high-quality phage genomes via de novo metagenome assembly is challenging due to the modular and mosaic nature of phage genomes (Hatfull 2008, Belcaid *et al.* 2010, Lima-Mendez *et al.* 2011). Repeat regions can result in fragmented assemblies and chimeric contigs (Casjens and Gilcrease 2009, Merrill *et al.* 2016). Hence, current state-of-the-art computational tools rely on the combination of either more

conservative tools based on sequence- and profile-based screening [e.g. MetaPhinder (Jurtz *et al.* 2016)] or machine learning approaches based on nucleotide signatures [e.g. Seeker (Auslander *et al.* 2020), refer to Supplementary Table S1 in Section S1]. Resulting predictions are then evaluated using tools such as CheckV (Nayfach *et al.* 2021a) and VIBRANT (Kieft *et al.* 2020) to categorize the predicted phages based on their completeness, contamination levels, and possible lifestyle (virulent or temperate) (McNair *et al.* 2012). Due to the supervised nature of the underlying approaches, most of these tools cannot characterize novel viruses that are significantly different from known viruses. Moreover, the approach used by these tools can be problematic with fragmented assemblies where contigs do not always represent complete genomes. In an attempt to address this limitation, tools such as MARVEL (Amgarten *et al.* 2018) and PHAMB (Johansen *et al.* 2022) were developed to identify viral metagenome-assembled genomes (vMAGs) of phages from metagenomic data. These programs rely on existing taxonomy-independent metagenomic binning tools such as MetaBAT2 (Kang *et al.* 2019) or VAMB (Nissen *et al.* 2021) and attempt to predict viral genome bins from this output using machine learning techniques.

Metagenomic binning tools are designed to capture nucleotide and sequence coverage-specific patterns of different taxonomic groups; therefore, sequences from viruses with low and uneven sequence coverage are often inaccurately binned. Many metagenomic binning tools filter out short sequences [e.g. shorter than 1500 bp (Kang *et al.* 2019)], which further result in the loss of essential regions in phage genomes that are often present as short fragments in the assembly (Casjens and Gilcrease 2009). Moreover, most metagenomic binning tools struggle to distinguish viruses from genetically diverse populations with high strain diversity and quasispecies dynamics. These tools do not resolve the clustered sequences into contiguous genomes and the bins produced often contain a mixture of multiple strains resulting in poor-quality MAGs (Meyer *et al.* 2022). Existing solutions developed for viral quasispecies assembly only consider one species at a time (Baaijens *et al.* 2020, Freire *et al.* 2021, 2023) and cannot be applied to complex metagenomes. Despite the recent progress, it is challenging for currently available tools to recover complete high-quality phage genomes from metagenomic data, and a novel approach is required to address this issue. The use of connectivity information from assembly graphs could overcome these challenges [as shown in previous studies on bacterial metagenomes (Lamurias *et al.* 2022, Mallawaarachchi and Lin 2022a, 2022b)] to enable the recovery of high-quality phage genomes.

In this article, we introduce Phables, a software tool that can resolve phage genomes from viral metagenome assemblies. First, Phables identifies phage-like components in the assembly graph using conserved genes. Second, using read mapping information, graph algorithms and flow decomposition techniques, Phables identifies the most probable combinations of varying phage genome segments within a component, leading to the recovery of accurate phage genome assemblies (Fig. 1). We evaluated the quality of the resolved genomes using different assessment techniques and demonstrate that Phables produces complete and high-quality phage genomes.

2 Materials and methods

Figure 1 presents the overall workflow of Phables. Reads from single or multiple viral metagenomic samples are

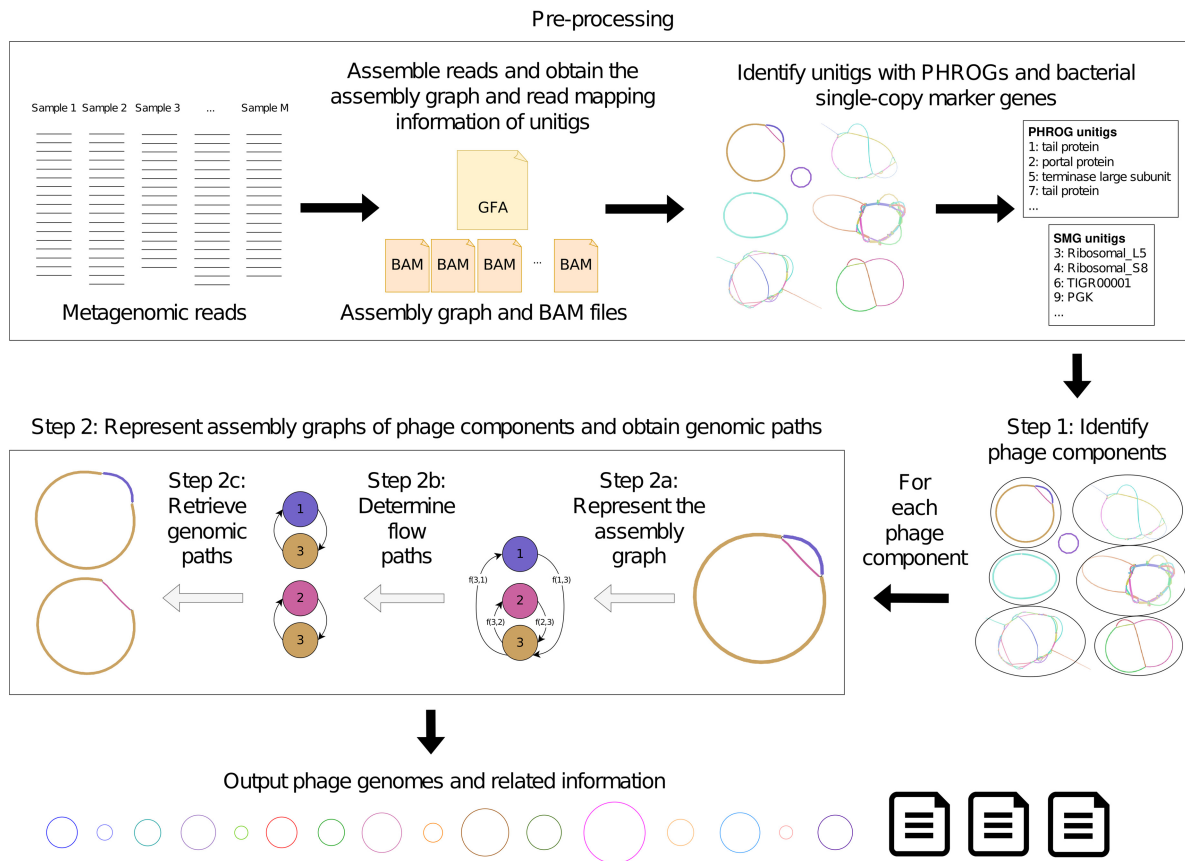


Figure 1. Phables workflow. Preprocessing: assemble reads, obtain the assembly graph and read mapping information, and identify unitigs with PHROGs and bacterial single-copy marker genes. Step 1: Identify phage components from the initial assembly. Step 2: For each phage component, represent the assembly graph, determine the flow paths and retrieve the genomic paths. Finally, output phage genomes and related information.

assembled, and the assembly graph and read mapping information are obtained. The unitig sequences from the assembly graph are extracted and screened for Prokaryotic Virus Remote Homologous Groups (PHROGs) (Terzian *et al.* 2021) and bacterial single-copy marker genes. Phables identifies sub-graphs known as ‘phage components’ and resolves separate phage genomes from each phage component. Finally, Phables outputs the resolved phage genomes and related information. Each step of Phables is explained in detail in the following sections.

2.1 Preprocessing

The preprocessing step performed by Phables uses an assembly graph and generates the read mapping information and the gene annotations required for Step 1 in the workflow. We recommend Hecatomb (Roach *et al.* 2022a) to assemble the reads into contigs and obtain the assembly graph. However, Phables can work with any assembly graph in the Graphical Fragment Assembly (GFA) format.

The unitig sequences are extracted from the assembly graph, and the raw sequencing reads are mapped to the unitigs using Minimap2 (Li 2018) and Samtools (Li *et al.* 2009). Phables uses CoverM (<https://github.com/wwood/CoverM>) [available from Coverage (<https://github.com/beardymcjohnface/Coverage>)] to calculate the read coverage of unitigs, using the reads from all samples, and records the mean coverage (the average number of reads that map to each base of the unitig).

Phables identifies unitigs containing PHROGs (Terzian *et al.* 2021). PHROGs are viral protein families commonly used to annotate prokaryotic viral sequences. MMSeqs2 (Steinberger and Söding 2017) is used to identify PHROGs in unitigs using an identity cutoff of 30% and an e-value of less than 10^{-10} (by default).

Next, Phables identifies unitigs containing bacterial single-copy marker genes. Most bacterial genomes have conserved genes known as single-copy marker genes (SMGs) that appear only once in a genome (Dupont *et al.* 2012, Albertsen *et al.* 2013). FragGeneScan (Rho *et al.* 2010) and HMMER (Eddy 2011) are used to identify sequences containing SMGs. SMGs are considered to be present if more than 50% (by default) of the gene length is aligned to the unitig. The list of SMGs is provided in [Supplementary Table S2 in Section S2](#).

2.2 Step 1: Identify phage components

Phables identifies components from the final assembly graph where all of its unitigs do not have any bacterial SMGs (identified from the preprocessing step) and at least one unitig contains one or more genes belonging to a PHROG for at least one of the PHROG categories: ‘head and packaging’, ‘connector’, ‘tail’ and ‘lysis’, which contain known phage structural proteins and are highly conserved in tailed phages (Auslander *et al.* 2020) (refer to [Supplementary Fig. S1 in Section S3](#) for an analysis of the PHROG hits to all known phage genomes). The presence of selected PHROGs ensures the components are phage-like and represent potential phage genomes. The absence of bacterial SMGs further ensures that the

components are not prophages. These identified components are referred to as ‘phage components’. Components that are comprised of a single circular unitig (the two ends of the unitig overlap) or a single linear unitig and that satisfy the above conditions for genes are considered phage components only if the unitig is longer than the predefined threshold ‘minlength’ that is set to 2000 bp by default, as this is the approximate lower bound of genome length for tailed phages (Luque *et al.* 2020).

2.3 Step 2: Represent assembly graphs of phage components and obtain genomic paths

2.3.1 Step 2a: Represent the assembly graph

Following the definitions from STRONG (Quince *et al.* 2021), we define the assembly graph $G = (V, E)$ for a phage component where $V = 1, 2, 3, \dots, |V|$ is a collection of vertices corresponding to unitig sequences that make up a phage component and directed edges $E \in V \times V$ represent connections between unitigs. Each directed edge $(u^{d_1} \rightarrow v^{d_2})$ is defined by a starting vertex ‘minlength’ and an ending vertex v (the arrow denotes the direction of the overlap), where $d_1, d_2 \in \{+, -\}$ indicates whether the overlap occurs between the original sequence, indicated by a + sign or its reverse complement, indicated by a - sign.

The weight of each edge $(u^{d_1} \rightarrow v^{d_2})$ irrespective of the orientation of the edge, termed $w_e(u \rightarrow v)$ is set to the minimum of the read coverage values of the two unitigs u and v . We also define the confidence of each edge $(u^{d_1} \rightarrow v^{d_2})$ irrespective of whether the overlap occurs between the original sequence or its reverse complement, termed $c_e(u \rightarrow v)$ as the number of paired-end reads spanning across $(u \rightarrow v)$. Here, the forward read maps to unitig u and the reverse read maps to unitig v . We also define the confidence of paths $(t \rightarrow u \rightarrow v)$ termed $c_p(t \rightarrow u \rightarrow v)$ as the number of paired-end reads spanning across $(t \rightarrow u \rightarrow v)$. Here, the forward read maps to unitig t and the reverse read maps to unitig v . Paired-end information has been used in previous studies for assembling viral quasispecies to untangle assembly graphs (Freire *et al.* 2023). Moreover, paired-end reads are widely used in manual curation steps to join contigs from metagenome assemblies and extend them to longer sequences (Chen *et al.* 2020). The more paired-end reads map to the pair of unitigs, the more confident we are about the overlap represented by the edge (refer to Supplementary Fig. S2 in Section S4 for histograms of edge confidence).

2.3.2 Step 2b: Determine flow paths

Phables models the graph of the phage component as a flow decomposition problem and obtains the genomic paths with their corresponding coverage values. We define three cases

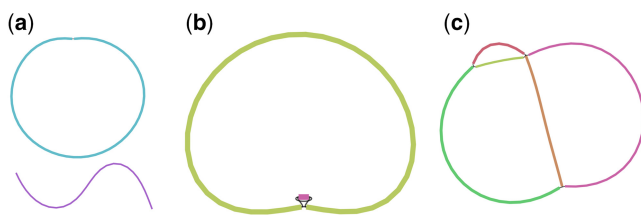


Figure 2. Cases of phage components. (a) Case 1 represents a phage component with one circular or linear unitig. (b) Case 2 represents a phage component with two circular unitigs connected to each other. (c) Case 3 represents a phage component that is more complex with multiple unitigs and multiple paths.

based on the number and arrangement of unitigs present in the phage component as shown in Fig. 2.

Case 1: When the phage component has only one circular or linear unitig longer than the predefined threshold ‘minlength’, Phables considers this unitig as one genome. The genomic path is defined as the unitig sequence itself.

Case 2: The phage components in Case 2 have two circular unitigs connected together where at least one is longer than the predefined threshold ‘minlength’. This is an interesting case as the shorter unitig corresponds to the ‘terminal repeats’ of phages. Some phages have double-stranded repeats at their termini which are a few hundred base pairs in length and are exactly the same in every virion chromosome (i.e. they are not permuted) (Casjens and Gilcrease 2009). The terminal repeats are generated by a duplication of the repeat region in concert with packaging (Chung *et al.* 1990, Zhang and Studier 2004) (refer to Supplementary Fig. S3 in Section S5). This type of end structure could be overlooked when a phage genome sequence is determined by shotgun methods because sequence assembly can merge the two ends to give a circular sequence. Phables successfully resolves these terminal repeats to form complete genomes. To resolve the phage component in Case 2, we consider the shorter unitig as the terminal repeat. Now we combine the original sequence of the terminal repeat to the beginning and end of the longer unitig (refer to Supplementary Fig. S3 in Section S5). The coverage of the path will be set to the coverage of the longer unitig.

Case 3: In Case 3, we have more complex phage components where there are more than two unitigs forming branching paths, and we model them as a minimum flow decomposition (MFD) problem. The MFD problem decomposes a directed acyclic graph (DAG) into a minimum number of source-to-sink ($s - t$) paths that explain the flow values of the edges of the graph (Vatnlen *et al.* 2008, Dias *et al.* 2022). The most prominent applications of the MFD problem in bioinformatics include reconstructing RNA transcripts (Tomescu *et al.* 2013, Shao and Kingsford 2017, Gatter and Stadler 2019) and viral quasispecies assembly (Baaijens *et al.* 2020). The MFD problem can be solved using integer linear programming (ILP) (Schrijver 1998).

In the viral metagenomes, we have identified structures containing several phage variant genomes, that are similar to viral quasispecies often seen in RNA viruses (Domingo and Perales 2019). Hence, Phables models each of the remaining phage components as an MFD problem and uses the minimum flow decomposition using integer linear programming (MFD-ILP) implementation from Dias *et al.* (2022). MFD-ILP finds a flow decomposition $FD(\mathcal{P}, \mathbf{w})$ with a set of $s - t$ flow paths \mathcal{P} and associated weights \mathbf{w} such that the number of flow paths is minimized. These flow paths represent possible genomic paths. An example of a phage component with possible paths is shown in Fig. 3.

First, we convert the assembly graph of the phage component into a DAG. We start by removing ‘dead-ends’ from G . We consider a vertex to be a dead-end if it has either no incoming edges or no outgoing edges, which arise due to errors at the start or end of reads that can create protruding chains of deviated edges (Bankevich *et al.* 2012). Dead-ends are particularly problematic in later steps of Phables as they can affect the continuity of genomic paths. Hence, their removal ensures that all the possible paths in the graph form closed cycles. We eliminate dead-ends by recursively removing vertices with either no incoming edges or no outgoing edges. Note

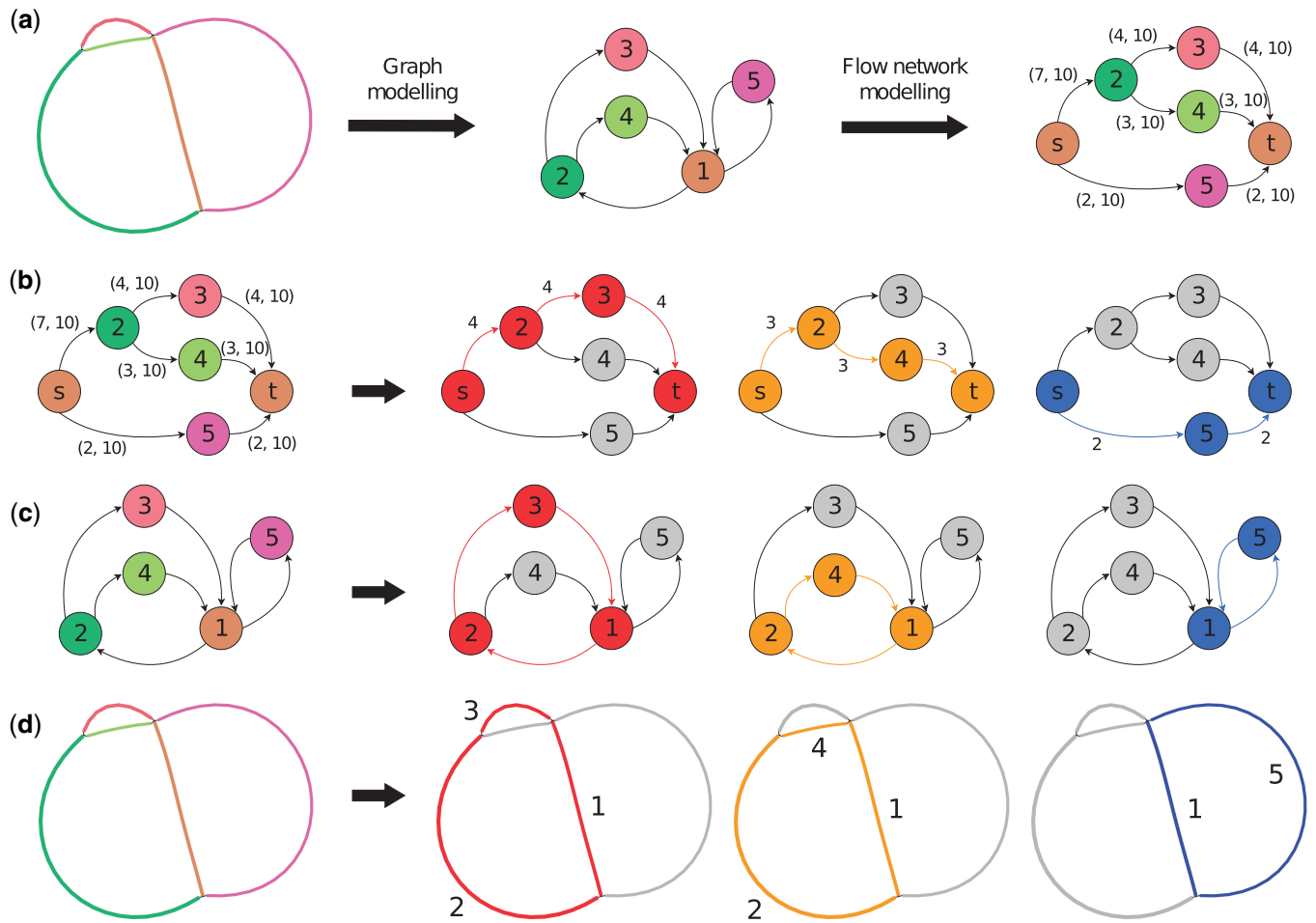


Figure 3. Modelling of an example Case 3 phage component. Example of a Case 3 phage component (a) being modelled as a graph, a flow network and resolved into paths denoted using (b) flow network visualization with flow values, (c) graph visualization with directed edges, and (d) Bandage (Wick *et al.* 2015) visualization (with corresponding unitig numbers). Here, three $s - t$ flow paths ($1 \rightarrow 2 \rightarrow 3$, $1 \rightarrow 2 \rightarrow 4$, and $1 \rightarrow 5$) can be obtained corresponding to three phage genomes. The thick black arrows in B to D denote the resolution into three paths.

that removing one dead-end can cause another vertex that is linked only to the removed one to become a dead-end, hence the removal process is done recursively.

Since a Case 3 phage component forms a cyclic graph as shown in Fig. 3a, we have to identify a vertex to represent the source/sink (referred to as st) in order to convert the graph to a DAG and model it as a flow network. Starting from every vertex (‘source’), we conduct a breadth-first-search and identify an iterator, ($level, vertices$), where ‘vertices’ is the non-empty list of vertices at the same distance ‘level’ from the ‘source’. The method that generates this iterator is known as *bfs_layers* and we use the NetworkX implementation (<https://networkx.org/documentation/stable/reference/algorithms/>). We extract the vertices in the final layer and check if their successors are equal to ‘source’. If this condition holds for some vertex in G , we consider this vertex to be the st vertex. If more than one vertex satisfies the condition to be a st vertex, then we pick the vertex corresponding to the longest unitig as the st vertex. This process is carried out to find a vertex common to the flow paths (refer to Supplementary Algorithm S1 in Section S6). As an example, consider vertex 1 in Fig. 3a. When we do a breadth-first-search starting from vertex 1, the vertices in the last layer in our iterator will be 3 and 4. The

successor of both 3 and 4 is vertex 1. Since the successors of the vertices in the last layer are the same as the starting vertex, we consider vertex 1 as the st vertex.

The edges of G that are weighted according to unitig coverage, may not always satisfy the conservation of flow property because of uneven sequencing depths at different regions of the genomes (Peng *et al.* 2011). Hence, we use inexact flow networks that allow the edge weights to belong to an interval. Once we have identified a st vertex, we separate that vertex into two vertices for the source s and sink t . We create an inexact flow network $G_f = (V, E, f, \bar{f})$ from s to t and model the rest of the vertices and edges in G . For example, in Fig. 3b vertex 1 is broken into two vertices s and t , and the network flows from s to t . For every edge $(u \rightarrow v) \in E$, we have associated two positive integer values $f_{uv} \in f$ and $\bar{f}_{uv} \in \bar{f}$, satisfying $f_{uv} \leq \bar{f}_{uv}$, where $f_{uv} = w_e(u \rightarrow v)$, $\bar{f}_{uv} = \lfloor \alpha \times cov_{max} \rfloor$, $\alpha \geq 1$ is the coverage multiplier parameter (1.2 by default) and cov_{max} is the maximum coverage of a unitig in the phage component. In Fig. 3b, each edge has two values (f_{uv}, \bar{f}_{uv}) that define the flow interval for the inexact flow network G_f . This modelling ensures that the flow through each edge is bounded by a relaxed interval between the edge weight and the maximum coverage within the component. For example, in Fig. 3b,

the edge ($2 \rightarrow 3$) has a weight of 4 (which is the minimum of the read coverage values of the two unitigs 2 and 3 obtained from Step 2a). Here, $\alpha = 1.2$ and $cov_{max} = 9$ for the component, so we set $f_{uv} = 4$ and $\bar{f}_{uv} = 10$.

Next, we define a set of simple paths $\mathcal{R} = \{R_1, R_2, R_3, \dots, R_j\}$, where the edges that form each path have paired-end reads spanning across them, i.e. $c_e(u \rightarrow v) \geq mincov$ ($mincov = 10$ by default). Enforcing these paths to contain paired-end reads ensures that genuine connections are identified and reflected in at least one decomposed path. For example, in Fig. 3b, the edge ($2 \rightarrow 3$) has 10 paired-end reads spanning across the edge. Hence, we add the path $R_1 = (2, 3)$ to \mathcal{R} . Moreover, for a path ($t \rightarrow u \rightarrow v$) passing through the junction u (where the in-degree and out-degree are non-zero), we add the path $R_j = (t, u, v)$ to \mathcal{R} , if $c_p(t \rightarrow u \rightarrow v) \geq mincov$ or if $|w_e(t \rightarrow u) - w_e(u \rightarrow v)|$ is less than a predefined threshold $cov_{tolerance}$ (100 by default). This allows Phables to specify longer subpaths across complex junctions.

Now we model our inexact flow network G_f as a minimum inexact flow decomposition (MIFD) problem and determine a minimum-sized set of $s - t$ paths $\mathcal{P} = (P_1, P_2, P_3, \dots, P_k)$ and associated weights $\mathbf{w} = (w_1, w_2, w_3, \dots, w_k)$ with each $w_i \in \mathbb{Z}^+$ where the following conditions hold.

- 1) $f_{uv} \leq \sum_{i \in \{1, \dots, k\} s.t. (u,v) \in P_i} w_i \leq \bar{f}_{uv} \forall (u, v) \in E$
- 2) $\forall R_j \in \mathcal{R}, \exists P_i \in \mathcal{P}$ such that R_j is a subpath of P_i

A path P_i will consist of unitigs with orientation information. The weight w_i will be the coverage of the genome represented by the path P_i .

2.3.3 Step 2c: Retrieve genomic paths

The flow paths obtained from Cases 1 and 2 described in the previous section are directly translated to genomic paths based on the unitig sequences. In Case 3, we get $s - t$ paths from the flow decomposition step (as shown in Fig. 3b). The paths longer than the predefined threshold ‘minlength’ and have a predefined coverage threshold of ‘mincov’ (10 by default) or above are retained. For each remaining path, we remove t from the path as s and t are the same vertex and combine the nucleotide sequences of the unitigs corresponding to the vertices and the orientation of edges in the flow path (refer to Fig. 3c and d). Once the genomic paths of phage components are obtained, we record the constituent unitigs, path length (in bp), coverage (i.e. the flow value of the path), and the guanine-cytosine (GC) content of each genomic path.

3 Experimental design

3.1 Simulated phage dataset

We simulated reads from the following four phages with the respective read coverage values and created a simulated phage dataset (referred to as ‘simPhage’) to evaluate Phables.

- 1) Enterobacteria phage P22 (AB426868) - $100 \times$
- 2) Enterobacteria phage T7 (NC_001604) - $150 \times$
- 3) Staphylococcus phage SAP13 TA-2022 (ON911718) - $200 \times$
- 4) Staphylococcus phage SAP2 TA-2022 (ON911715) - $400 \times$

The Staphylococcus phage genomes have an average nucleotide identity (ANI) of 96.89%. Paired-end reads were simulated

using InSilicoSeq (Gourlé *et al.* 2019) with the predefined MiSeq error model. We used metaSPAdes (Nurk *et al.* 2017) from SPAdes version 3.15.5 to assemble the reads into contigs and obtain the assembly graph for the simPhage dataset. Supplementary Tables S3 and S5 in Section S7 summarize the details of the simulations and assemblies.

3.2 Real datasets

We tested Phables on the following real viral metagenomic datasets available from the National Center for Biotechnology Information (NCBI).

- 1) Water samples from Nansi Lake and Dongping Lake in Shandong Province, China (NCBI BioProject number PRJNA756429), referred to as ‘Lake Water’
- 2) Soil samples from flooded paddy fields from Hunan Province, China (NCBI BioProject number PRJNA866269), referred to as ‘Paddy Soil’
- 3) Wastewater virome (NCBI BioProject number PRJNA434744), referred to as ‘Wastewater’
- 4) Stool samples from patients with IBD and their healthy household controls (NCBI BioProject number PRJEB7772) (Norman *et al.* 2015), referred to as ‘IBD’

All the real datasets were processed using Hecatomb version 1.0.1 to obtain a single assembly graph for each dataset (Roach *et al.* 2022a). Supplementary Tables S3–S5 in Section S7 summarize the information about the datasets and their assemblies.

3.3 Tools benchmarked

We benchmarked Phables with PHAMB (Johansen *et al.* 2022) (version 1.0.1), a viral identification tool that predicts whether MAGs represent phages and outputs genome sequences. PHAMB takes binning results from a metagenomic binning tool and predicts bins that contain bacteriophage sequences. The MAGs for PHAMB were obtained by running VAMB (version 3.0.8), a binning tool that does not rely on bacterial marker genes, in co-assembly mode on the original contigs with the author-recommended parameter `--minfasta 2000` and the `--cuda` flag. The commands used to run all the tools can be found in Supplementary Section S8.

3.4 Evaluation criteria

3.4.1 Evaluation criteria for binning tools

The resolved genomes from Phables and identified MAGs from PHAMB were evaluated using CheckV version 1.0.1 (Nayfach *et al.* 2021a) (with reference database version 1.5) which compares bins/genomes against a large database of complete viral genomes. We compare the following metrics from the CheckV results.

- 1) CheckV viral quality
- 2) Completeness of sequences—number of sequences with $>90\%$ completeness
- 3) Contamination of sequences—number of sequences with $<10\%$ contamination
- 4) The number and length distribution of sequences with the following warnings
 - a) Contig $>1.5 \times$ longer than expected genome length
 - b) High *kmer.freq* may indicate a large duplication

Since PHAMB predicts all viral bins, we only consider the bins from PHAMB that contain the contigs corresponding to the unitigs recovered by Phables for a fair comparison.

3.4.2 Evaluation criteria for resolved genomes

The number of components resolved by Phables for each case was recorded. The viral quality of the resolved genomes and the unitigs and contigs contained in the corresponding genomic paths were evaluated using CheckV (Nayfach *et al.* 2021a). Since the reference genomes for the simPhage dataset were available, we evaluated the resolved genomes using metaQUAST (Mikheenko *et al.* 2016).

4 Results

4.1 Benchmarking results on the simulated phage dataset

We first benchmarked Phables using the simPhage dataset. We evaluated the resolved phage genomes using metaQUAST (Mikheenko *et al.* 2016). We analysed the genome coverage reported from metaQUAST and the average coverage values reported by Phables. Figure 4 denotes the Bandage (Wick *et al.* 2015) visualization of the assembly graph from the simPhage dataset and how Phables resolved the complex Case 3 component containing the two Staphylococcus phages.

Phables recovered the two Staphylococcus phage genomes with over 92% genome coverage (refer to Table 1). The slightly low genome coverage for Staphylococcus phage SAP2 TA-2022 may have been due to the omission of the dead-end which was not properly assembled. Moreover, Phables has recovered the circular genome of Enterobacteria phage P22 and the linear genome of Enterobacteria phage T7 as well. According to Table 1, the coverage values reported from Phables are similar or close to the actual simulated coverage values of the genomes. VAMB failed to run on this dataset as there were fewer contigs than the minimum possible batch size and hence PHAMB could not be run.

4.2 Benchmarking results on the real datasets

Phables resolves unitigs within phage components to produce multiple complete and high-quality genomes from the viral metagenomes (Fig. 5). The genome quality of Phables results was compared with the vMAG prediction tool PHAMB (Johansen *et al.* 2022) and evaluated using CheckV (Nayfach *et al.* 2021a). Figure 5 denotes the comparison of length distributions and bin/genome counts of different CheckV quality categories for Phables and PHAMB results. Unlike Phables, PHAMB has produced genomes with longer sequences as shown in Fig. 5a, c, e, and g, because PHAMB combines all the contigs in a bin to form one long sequence. As denoted in Fig. 5b, d, f, and h, Phables has recovered the greatest number of complete and high-quality genomes combined for all the datasets; 165 in Lake Water, 389 in Paddy Soil, 55 in Wastewater, and 205 in IBD, with 49.54% more genomes recovered than PHAMB on average.

Phables accurately recovers short sequences such as terminal repeats that are challenging for metagenomic binning tools to recover using the assembly graph and produces high-quality genomes. We observed that VAMB incorrectly binned the majority of the short sequences, which reduced the quality of PHAMB results. For example, the repeat sequences in the

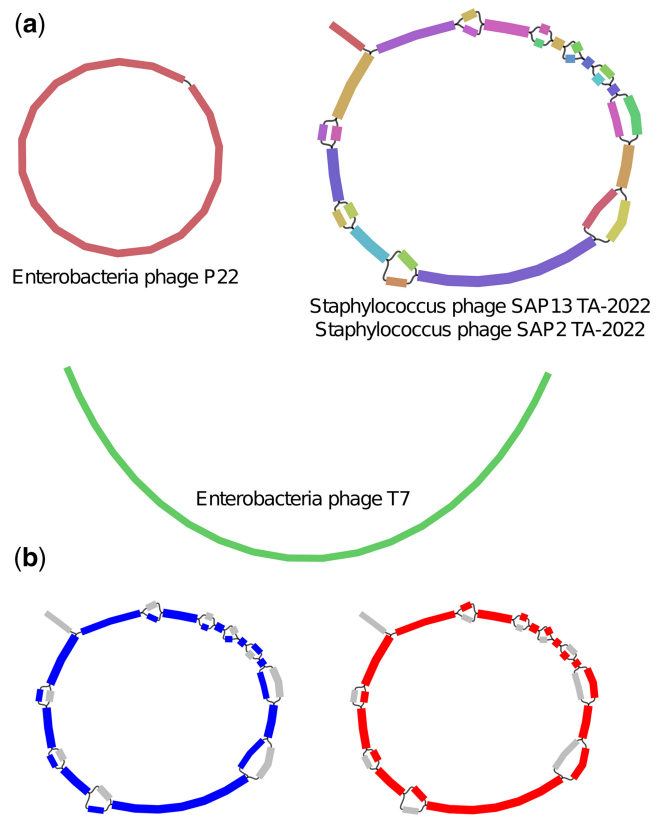


Figure 4. simPhage assembly graph. Visualization of the (a) assembly graph from the simPhage dataset with phage components and (b) resolution of two paths (red and blue) from the Staphylococcus phage component.

Table 1. Evaluation results for the genomes resolved from Phables for the simPhage dataset.

Genome	Simulated coverage	Phables predicted coverage	Genome coverage (%)
P22	100	100	99.947
T7	150	150	99.599
SAP13 TA-2022	200	206	100.00
SAP2 TA-2022	400	401	92.406

Case 2 phage components identified by Phables had a mean length of 600 bp in Lake Water, 649 bp in Paddy Soil, 511 bp in Wastewater, and 638 bp in IBD datasets (refer to Supplementary Table S6 in Section S9 for exact lengths of the sequences). All of these short sequences, except for those from the IBD dataset were found in a different bin than the bin of their connected longer sequence in the PHAMB results (8 out of 8 in Lake Water, 2 out of 2 in Paddy Soil, and 1 out of 1 in Wastewater). Phables recovered these short repeat sequences along with their connected longer sequences within a phage component using the connectivity information of the assembly graph.

Phables resulted in a high number of low-quality genomes as determined by CheckV in the Wastewater dataset compared to the other datasets (Fig. 5f). A possible reason for this is that these may be novel phages (as they contain conserved phage markers even though CheckV categorizes them as ‘low-quality’ or ‘not-determined’), and so they are not yet present in the databases that CheckV relies on.

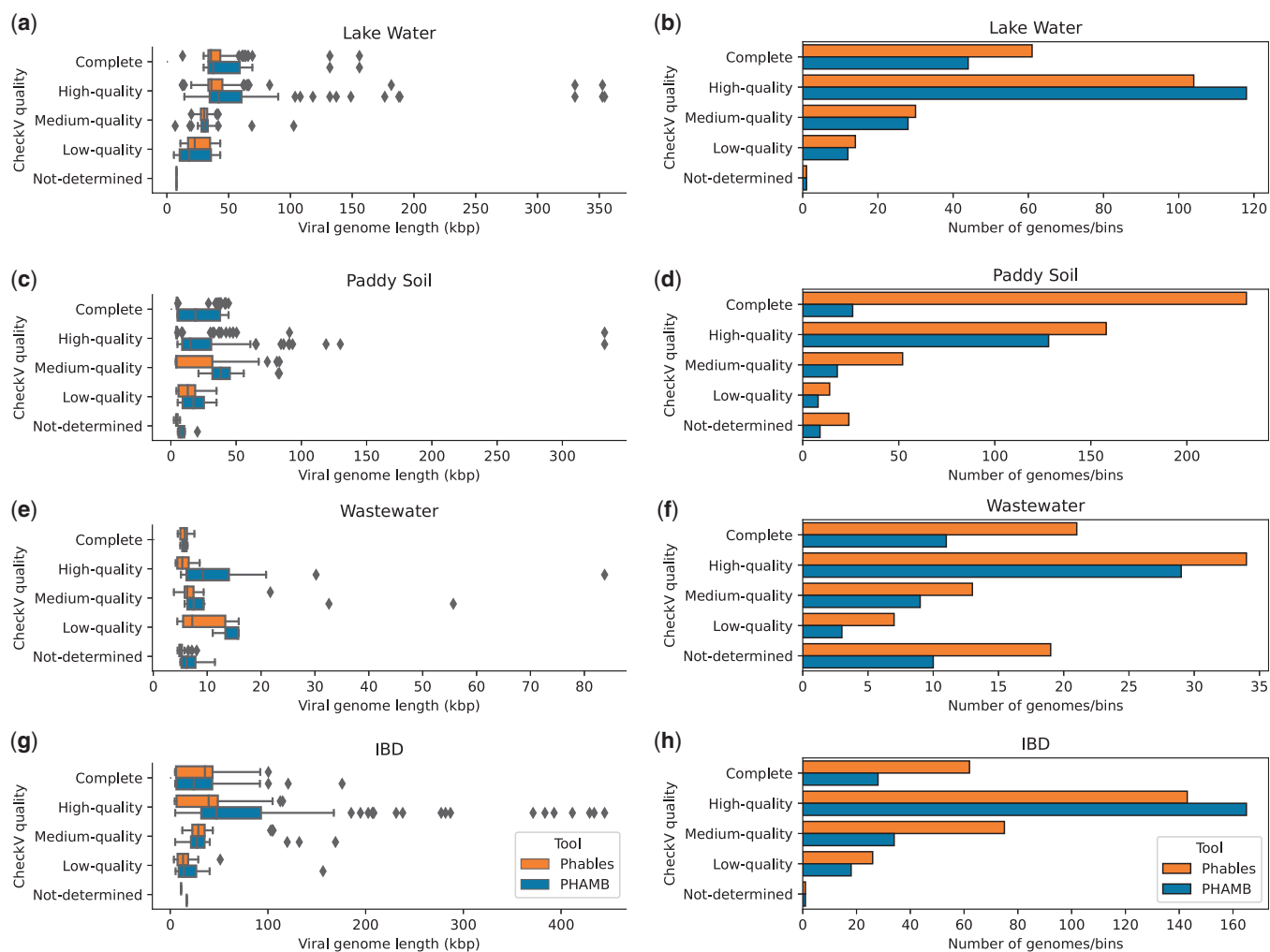


Figure 5. Comparison of CheckV quality. Genome length distribution (first column of figures) and abundance of genomes (second column of figures) belonging to different CheckV quality categories identified by Phables (denoted in orange) and PHAMB (Johansen *et al.* 2022) (denoted in blue) for the viral metagenomic datasets Lake Water, Paddy soil, Wastewater, and IBD.

PHAMB does not carry out any resolution steps when combining the contigs of identified MAGs, which results in erroneous genome structures, high levels of contamination and duplications within genomes because of the presence of multiple variant genomes. Such duplications are identified from the warnings reported by CheckV. Hence, we evaluated the number and length distribution of sequences having CheckV warnings and the results are shown in Fig. 6. PHAMB has produced the highest number of genomes with CheckV warnings and produced some very long genomes (≈ 355 – 485 kb as shown in Fig. 6a and g), suggesting the combination of two or more variant genomes together in a bin. Only a few genomes produced from Phables (five or less) contain CheckV warnings (refer to Supplementary Table S8 in Section S10 for the exact number of genomes with warnings). These results show that Phables accurately recovers variant genomes including regions like terminal repeats from viral metagenomic samples and produces more high-quality/complete genomes compared to existing state-of-the-art tools.

4.3 Components resolved and comparison of resolved genomes

The number of phage components resolved by Phables under each case was recorded for all the datasets (refer to

Supplementary Table S7 in Section S9 for the exact counts). Most of the resolved components belong to either Case 1 with a single circular unitig or Case 2 with the terminal repeat. When resolving Case 2 components, Phables provides information regarding terminal repeats such as the length of the repeat region, that will be overlooked by other tools. Except for the IBD dataset, Phables was able to resolve all the Case 3 phage components from the rest of the datasets. In a few cases, the Case 3 phage components could not be resolved because Phables was unable to find a *st* vertex for these very complex phage components (refer to Supplementary Fig. S8 in Section S11 for examples of unresolved phage components).

Assemblers attempt to resolve longer paths in the assembly graph by connecting unitigs to form contigs (Bankevich *et al.* 2012, Kolmogorov *et al.* 2019). However, they are still unable to resolve complete genomes for complex datasets due to the mosaic nature of phage genomes and produce fragmented assemblies. Phables can be used to resolve these problematic contigs (or unitigs) and obtain high-quality genomes. Figure 7 denotes the comparison of CheckV quality of the genomes resolved in Phables and the unitigs and contigs included in the phage components of Cases 2 and 3. The most complete and high-quality sequences can be found as genomes (61 and 104

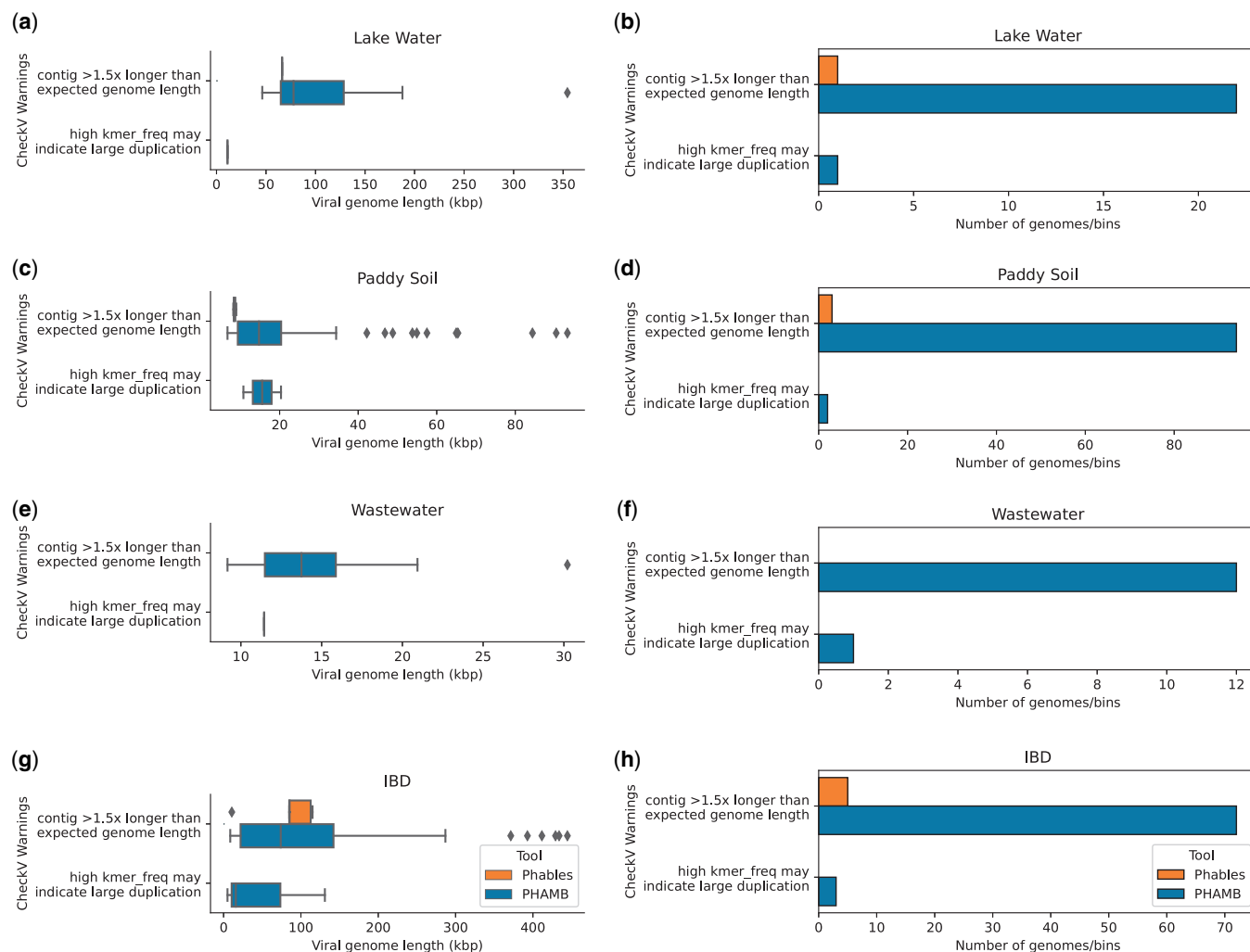


Figure 6. Comparison of CheckV warnings. Genome length distribution (first column of figures) and abundance of genomes (second column of figures) having the selected CheckV warnings from Phables (denoted in orange) and PHAMB (Johansen *et al.* 2022) (denoted in blue) results for the viral metagenomic datasets Lake Water, Paddy soil, Wastewater, and IBD.

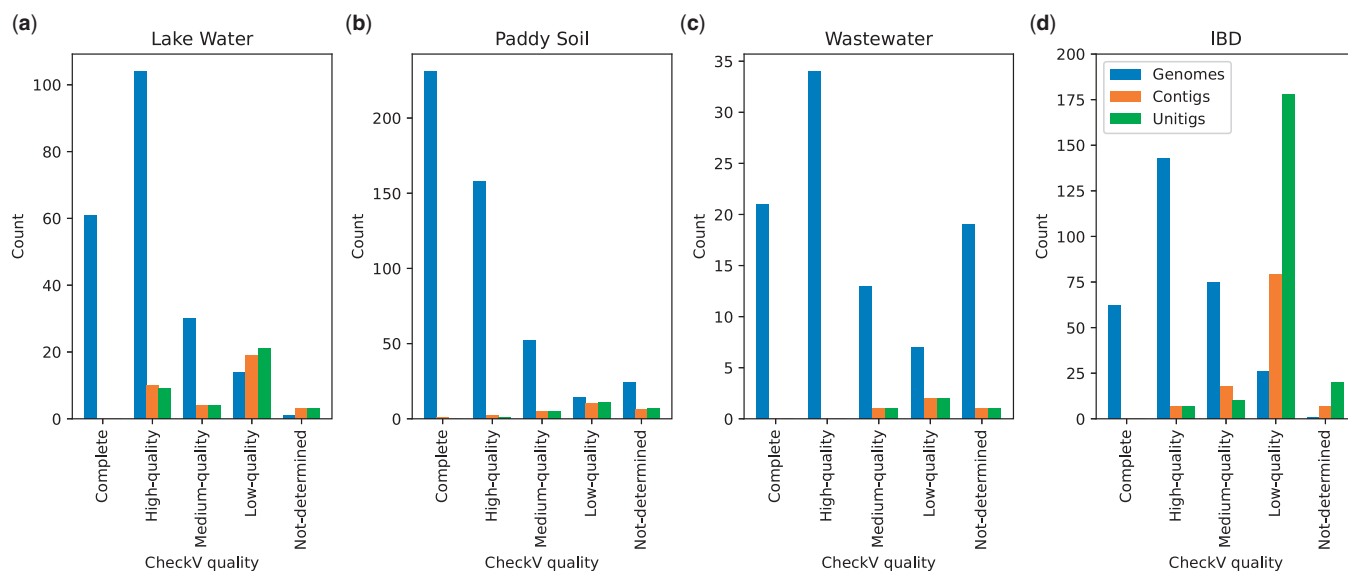


Figure 7. Comparison of Phables genomes, contigs and unitigs. Counts of resolved genomes of Phables, unitigs and contigs included in the phage components of Cases 2 and 3 with different CheckV qualities in the viral metagenomic datasets Lake Water, Paddy soil, Wastewater, and IBD.

for Lake Water, 231 and 158 for Paddy Soil, 21 and 34 for Wastewater, and 62 and 143 for IBD, respectively). In contrast, most medium- and low-quality genomes can be found from contigs and unitigs. Hence, genomes resolved using Phables have higher quality and will be better candidates for downstream analysis than contigs.

We compared the similarity between the genomes recovered within each Case 3 phage component for the IBD dataset using *pyani* (Pritchard *et al.* 2016), *pyGenomeViz* (<https://moshi4.github.io/pyGenomeViz/>), and *MUMmer* (Marçais *et al.* 2018) (refer to [Supplementary Section S12](#) for the detailed results). The ANI analysis revealed that the genomes resolved had over 95% ANI with some genomes having over 99% ANI and over 85% alignment coverage. Moreover, as shown in [Supplementary Fig. S10](#), the mosaic genome structure can be clearly seen where some unitigs are shared between genomes and some genomes have unique unitigs. Depending on the size and location within a specific genome, these unitigs potentially correspond to functional modules. Hence, Phables can resolve highly similar variant genomes with mosaic genome structures that the assemblers and binning tools are unable to distinguish.

4.4 Phage components from other assembly methods

We extended our testing of Phables with co-assemblies obtained from other metagenome assemblers including *metaSPAdes* (Nurk *et al.* 2017) and *MEGAHIT* (Li *et al.* 2015) to show that the components with bubbles observed in the assembly graph are not an artefact of the assembly approach used in Hecatomb. Co-assembly is conducted by combining reads from multiple metagenomes and assembling them together, which increases the sequencing depth and provides sufficient coverage for low-abundance genomes to be recovered (Delgado and Andersson 2022). However, this becomes a computationally intensive approach as the number of samples increases, and hence we have limited the results to just the Lake Water dataset. The results are provided in [Supplementary Section S13](#) and show that the phage component structures are still present in the assemblies and were correctly resolved by Phables, producing more high-quality genomes than PHAMB.

4.5 Implementation and resource usage

The source code of Phables was implemented using Python 3.10.12 and is available as a pipeline (including all the preprocessing steps) developed using *Snakemake* (Roach *et al.* 2022c). The commands used to run all the software can be found in [Supplementary Section S8](#). The running times of Phables core methods and running times including the preprocessing steps were recorded for all the datasets and can be found in [Supplementary Tables S10 and S11 in Section S14](#). The core methods of Phables can be run in under 2 min with less than 4 gigabytes of memory for all the datasets.

Phables uses a modified version of the MFD-ILP implementation from Dias *et al.* (2022) which supports inexact flow decomposition with subpath constraints. Gurobi version 10.0.2 (<https://www.gurobi.com/>) was used as the ILP solver. To reduce the complexity of the ILP solver, the maximum number of unitigs in a phage component to be solved was limited to 200.

5 Discussion

The majority of the existing viral identification tools rely on precomputed databases and models, only identify whether assembled sequences are of viral origin, and cannot produce complete and high-quality phage genomes. Viral binning tools have been able to overcome these shortcomings up to a certain extent by producing vMAGs, but they are fragmented and do not represent continuous genomes. Generally, the assembly process produces many short contigs where some represent regions which while important are challenging to resolve in phages, such as terminal repeat regions. These short contigs are discarded or binned incorrectly by binning tools, producing incomplete MAGs. Moreover, the mosaic genome structures of phage populations are a widely-documented phenomenon (Hatfull 2008, Belcaid *et al.* 2010, Lima-Mendez *et al.* 2011), and cannot be resolved by existing assemblers and binning tools. The resulting MAGs may contain multiple variant genomes assembled together and hence have high contamination.

Here, we introduce Phables, a new tool to resolve complete and high-quality phage genomes from viral metagenome assemblies using assembly graphs and flow decomposition techniques. We studied the assembly graphs constructed from different assembly approaches and different assembly software and consistently observed phage-like components with variation ('phage components'). Phables models the assembly graphs of these components as a minimum flow decomposition problem using read coverage and paired-end mapping information and recovers the genomic paths of different variant genomes. Experimental results confirmed that Phables recovers complete and high-quality phage genomes with mosaic genome structures, including important regions such as terminal repeats. However, Phables can identify certain plasmids as phages [e.g. 'phage-plasmids' (Ravin *et al.* 1999, Pfeifer *et al.* 2021, 2022)] because they can encode proteins homologous to phage sequences (refer to [Supplementary Section S15](#)). Hence, if users run mixed-microbial communities through Phables, further downstream analysis is required to ensure that the predicted genomes do not include plasmids.

Decomposing assembly graphs has become a popular method to untangle genomes and recover variant genomes from assemblies and while we have successfully used it to obtain mostly circular phage genomes, further work needs to be conducted to handle metagenomes of mixed-microbial communities and recover the range of phage genomes. In the future, we intend to add support for long-read assemblies from dedicated metagenome assemblers that will enable Phables to enforce longer subpaths that will span across more sequences during the flow decomposition modelling. We also intend to extend the capabilities of Phables to recover linear phage genomes from complex components and explore the avenues for recovering high-quality eukaryotic viral genomes from metagenomes.

Acknowledgements

The authors thank Prof Christopher Quince for the insights and suggestions regarding the development of methods. This research was undertaken with the resources and services from Flinders University's High-Performance Computing platform (<https://doi.org/10.25957/FLINDERS.HPC.DEEPThought>) and the National Computational Infrastructure (NCI), Australia.

Author contributions

V.M. designed the methods, developed the software, performed all analyses and wrote the paper. M.J.R. preprocessed the datasets. M.J.R. and P.D. assisted with developing the pipeline, optimizing the steps and writing the paper. M.J.R., B.P., S.R.G., P.D., G.B., and L.K.I. tested the software and assisted with the data analysis. S.K.G., R.D.H., and A.L.K.H. curated data. All the authors reviewed the manuscript and provided detailed feedback. P.D., E.A.D., and R.A.E. conceived the project and wrote the paper with input from all authors.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

This work was supported by the National Institutes of Health (NIH) National Institute of Diabetes and Digestive and Kidney Diseases [RC2DK116713], the Australian Research Council [DP220102915], and the Polish National Agency for Academic Exchange (NAWA) Bekker Programme [BPN/BEK/2021/1/00416 to P.D.].

Data and code availability

All the real datasets containing raw sequencing data used for this work are publicly available from their respective studies. The Lake Water dataset was downloaded from NCBI with BioProject number PRJNA756429, the Paddy Soil dataset from BioProject number PRJNA756429, the Wastewater dataset from BioProject number PRJNA434744, and the whole genome sequencing runs of the IBD data from BioProject number PRJEB7772. The sequencing reads for the simPhage dataset, all the assembled data and results from all the tools are available on Zenodo at <https://zenodo.org/record/8137197>.

The code of Phables is freely available on GitHub under the MIT license and can be found at <https://github.com/Vini2/Phables>. All analyses in this study were performed using Phables v.1.1.0 with default parameters.

References

- Albertsen M, Hugenholz P, Skarshewski A *et al.* Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nat Biotechnol* 2013;31:533–8.
- Amgarten D, Braga LPP, da Silva AM *et al.* MARVEL, a tool for prediction of bacteriophage sequences in metagenomic bins. *Front Genet* 2018;9:304.
- Auslander N, Gussow AB, Benler S *et al.* Seeker: alignment-free identification of bacteriophage genomes by deep learning. *Nucleic Acids Res* 2020;48:e121.
- Baaijens JA, Stougie L, Schönhuth A. Strain-aware assembly of genomes from mixed samples using flow variation graphs. In: Schwartz R (ed.), *Research in Computational Molecular Biology*. Cham: Springer International Publishing, 2020, 221–2. ISBN 978-3-030-45257-5.
- Bankevich A, Nurk S, Antipov D *et al.* SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J Comput Biol* 2012;19:455–77.
- Belcaid M, Bergeron A, Poisson G. Mosaic graphs and comparative genomics in phage communities. *J Comput Biol* 2010;17:1315–26.
- Breitbart M, Salamon P, Andresen B *et al.* Genomic analysis of uncultured marine viral communities. *Proc Natl Acad Sci U S A* 2002;99:14250–5.
- Casjens SR, Gilcrease EB. Determining DNA packaging strategy by analysis of the termini of the chromosomes in tailed-bacteriophage virions. *Methods Mol Biol* 2009;502:91–111.
- Chen L-X, Anantharaman K, Shaiber A *et al.* Accurate and complete genomes from metagenomes. *Genome Res* 2020;30:315–33.
- Chung Y-B, Nardone C, Hinkle DC. Bacteriophage T7 DNA packaging: III. A “hairpin” end formed on T7 concatemers may be an intermediate in the processing reaction. *J Mol Biol* 1990;216:939–48.
- Comeau AM, Hatfull GF, Krisch HM *et al.* Exploring the prokaryotic virosphere. *Res Microbiol* 2008;159:306–13. Exploring the prokaryotic virosphere.
- Cook R, Brown N, Redgwell T *et al.* INfrastructure for a PHAge REference Database: identification of large-scale biases in the current collection of cultured phage genomes. *Phage (New Rochelle)* 2021; 2:214–23.
- Delgado LF, Andersson AF. Evaluating metagenomic assembly approaches for biome-specific gene catalogues. *Microbiome* 2022; 10:72.
- Dias FH, Williams L, Mumei B *et al.* Efficient minimum flow decomposition via integer linear programming. *J Comput Biol* 2022;29:1252–67.
- Domingo E, Perales C. Viral quasispecies. *PLoS Genet* 2019;15:e1008271.
- Dupont CL, Rusch DB, Yooseph S *et al.* Genomic insights to SAR86, an abundant and uncultivated marine bacterial lineage. *ISME J* 2012;6:1186–99.
- Eddy SR. Accelerated profile hmm searches. *PLoS Comput Biol* 2011;7:e1002195.
- Edwards RA, Rohwer F. Viral metagenomics. *Nat Rev Microbiol* 2005; 3:504–10.
- Freire B, Ladra S, Paramá JR *et al.* Inference of viral quasispecies with a paired de Bruijn graph. *Bioinformatics* 2021;37:473–81.
- Freire B, Ladra S, Paramá JR *et al.* ViQUF: de novo viral quasispecies reconstruction using unitig-based flow networks. *IEEE/ACM Trans Comput Biol Bioinform* 2023;20:1550–62.
- Gatter T, Stadler PF. Ryūtō: network-flow based transcriptome reconstruction. *BMC Bioinformatics* 2019;20:190.
- Gourlé H, Karlsson-Lindsjö O, Hayer J *et al.* Simulating illumina metagenomic data with InSilicoSeq. *Bioinformatics* 2019;35:521–2.
- Hatfull GF. Bacteriophage genomics. *Curr Opin Microbiol* 2008;11:447–53. Antimicrobials/Genomics.
- Hesse RD, Roach M, Kerr EN *et al.* Phage diving: an exploration of the carcharhinid shark epidermal virome. *Viruses* 2022;14:1969.
- Hugenholz P, Goebel BM, Pace NR. Impact of culture-independent studies on the emerging phylogenetic view of bacterial diversity. *J Bacteriol* 1998;180:4765–74.
- Johansen J, Plichta DR, Nissen JN *et al.* Genome binning of viral entities from bulk metagenomics data. *Nat Commun* 2022;13:965.
- Jurtz VI, Villarroel J, Lund O *et al.* MetaPhinder—identifying bacteriophage sequences in metagenomic data sets. *PLoS One* 2016;11:e0163111. 09
- Kang DD, Li F, Kirton E *et al.* MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* 2019;7:e7359.
- Kececioglu JD, Myers EW. Combinatorial algorithms for DNA sequence assembly. *Algorithmica* 1995;13:7–51.
- Keen EC. A century of phage research: bacteriophages and the shaping of modern biology. *Bioessays* 2015;37:6–9.
- 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* 2020;8:90.

- Kolmogorov M, Yuan J, Lin Y *et al.* Assembly of long, error-prone reads using repeat graphs. *Nat Biotechnol* 2019;**37**:540–6.
- Lamurias A, Sereika M, Albertsen M *et al.* Metagenomic binning with assembly graph embeddings. *Bioinformatics* 2022;**38**:4481–7.
- Li D, Liu C-M, Luo R *et al.* MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* 2015;**31**:1674–6.
- Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 2018;**34**:3094–100.
- Li H, Handsaker B, Wysoker A, *et al.*; 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics* 2009;**25**:2078–9.
- Lima-Mendez G, Toussaint A, Leplae R. A modular view of the bacteriophage genomic space: identification of host and lifestyle marker modules. *Res Microbiol* 2011;**162**:737–46.
- Luque A, Benler S, Lee DY *et al.* The missing tailed phages: prediction of small capsid candidates. *Microorganisms* 2020;**8**:1944.
- Eusiak-Szelachowska M, Weber-Dąbrowska B, Jończyk-Matysiak E *et al.* Bacteriophages in the gastrointestinal tract and their implications. *Gut Pathog* 2017;**9**:44.
- Mallawaarachchi V, Lin Y. MetaCoAG: binning metagenomic contigs via composition, coverage and assembly graphs. In: Pe'er I (ed.), *Research in Computational Molecular Biology*. Cham: Springer International Publishing, 2022a, 70–85. ISBN 978-3-031-04749-7.
- Mallawaarachchi V, Lin Y. Accurate binning of metagenomic contigs using composition, coverage, and assembly graphs. *J Comput Biol* 2022b;**29**:1357–76. PMID: 36367700.
- Mallawaarachchi V, Wickramarachchi A, Lin Y. GraphBin: refined binning of metagenomic contigs using assembly graphs. *Bioinformatics* 2020a;**36**:3307–13.
- Mallawaarachchi VG, Wickramarachchi AS, Lin Y. GraphBin2: Refined and Overlapped Binning of Metagenomic Contigs Using Assembly Graphs. In: Kingsford C and Pisanti N (eds), *20th International Workshop on Algorithms in Bioinformatics (WABI 2020)*, volume 172 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020b, 8:1–8:21, ISBN 978-3-95977-161-0.
- Mallawaarachchi VG, Wickramarachchi AS, Lin Y. Improving metagenomic binning results with overlapped bins using assembly graphs. *Algorithms Mol Biol* 2021;**16**:3.
- Marçais G, Delcher AL, Phillippy AM *et al.* MUMmer4: a fast and versatile genome alignment system. *PLoS Comput Biol* 2018;**14**:e1005944.
- McNair K, Bailey BA, Edwards RA. PHACTS, a computational approach to classifying the lifestyle of phages. *Bioinformatics* 2012;**28**:614–8.
- Merrill BD, Ward AT, Grose JH *et al.* Software-based analysis of bacteriophage genomes, physical ends, and packaging strategies. *BMC Genomics* 2016;**17**:679.
- Meyer F, Fritz A, Deng Z-L *et al.* Critical assessment of metagenome interpretation: the second round of challenges. *Nat Methods* 2022;**19**:429–40.
- Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics* 2016;**32**:1088–90.
- Namiki T, Hachiya T, Tanaka H *et al.* MetaVelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Res* 2012;**40**:e155.
- Nayfach S, Camargo AP, Schulz F *et al.* CheckV assesses the quality and completeness of metagenome-assembled viral genomes. *Nat Biotechnol* 2021a;**39**:578–85.
- Nayfach S, Páez-Espino D, Call L *et al.* Metagenomic compendium of 189,680 DNA viruses from the human gut microbiome. *Nat Microbiol* 2021b;**6**:960–70.
- Nissen JN, Johansen J, Allesøe RL *et al.* Improved metagenome binning and assembly using deep variational autoencoders. *Nat Biotechnol* 2021;**39**:555–60.
- Norman JM, Handley SA, Baldrige MT *et al.* Disease-specific alterations in the enteric virome in inflammatory bowel disease. *Cell* 2015;**160**:447–60.
- Nurk S, Meleshko D, Korobeynikov A *et al.* Metaspades: a new versatile metagenomic assembler. *Genome Res* 2017;**27**:824–34.
- Peng Y, Leung HCM, Yiu SM *et al.* Meta-IDBA: a de novo assembler for metagenomic data. *Bioinformatics* 2011;**27**:94–101.
- Pevzner PA, Tang H, Tesler G. De novo repeat classification and fragment assembly. *Genome Res* 2004;**14**:1786–96.
- Pevzner PA, Tang H, Waterman MS. An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci USA* 2001;**98**:9748–53.
- Pfeifer E, Bonnin RA, Rocha EPC. Phage-plasmids spread antibiotic resistance genes through infection and lysogenic conversion. *mBio* 2022;**13**:e01851–22.
- Pfeifer E, Moura de Sousa JA, Touchon M *et al.* Bacteria have numerous distinctive groups of phage-plasmids with conserved phage and variable plasmid gene repertoires. *Nucleic Acids Res* 2021;**49**:2655–73.
- Pritchard L, Glover RH, Humphris S *et al.* Genomics and taxonomy in diagnostics for food security: soft-rotting enterobacterial plant pathogens. *Anal Methods* 2016;**8**:12–24.
- Quince C, Nurk S, Raguideau S *et al.* STRONG: metagenomics strain resolution on assembly graphs. *Genome Biol* 2021;**22**:214.
- Ravin NV, Svarchevsky AN, Dehò G. The anti-immunity system of phage-plasmid N15: identification of the antirepressor gene and its control by a small processed RNA. *Mol Microbiol* 1999;**34**:980–94.
- Rho M, Tang H, Ye Y. FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Res* 2010;**38**:e191. 08
- Roach MJ, Beecroft SJ, Mihindukulasuriya KA *et al.* Hecatomb: an end-to-end research platform for viral metagenomics. bioRxiv, <https://doi.org/10.1101/2022.05.15.492003>, 2022a, preprint: not peer reviewed.
- Roach MJ, McNair K, Michalczyk M *et al.* Philympics 2021: prophage predictions perplex programs. *F1000Res* 2022b;**10**:758.
- Roach MJ, Pierce-Ward NT, Suchecki R *et al.* Ten simple rules and a template for creating workflows-as-applications. *PLoS Comput Biol* 2022c;**18**:e1010705.
- Rodriguez-Valera F, Martin-Cuadrado A-B, Rodriguez-Brito B *et al.* Explaining microbial population genomics through phage predation. *Nat Rev Microbiol* 2009;**7**:828–36.
- Roux S, Emerson JB, Eloë-Fadrosch EA *et al.* Benchmarking viromics: an *in silico* evaluation of metagenome-enabled estimates of viral community composition and diversity. *PeerJ* 2017;**5**:e3817.
- Sanger F, Air GM, Barrell BG *et al.* Nucleotide sequence of bacteriophage ϕ X174 DNA. *Nature* 1977;**265**:687–95.
- Schrijver A. *Theory of Linear and Integer Programming*. Chichester: John Wiley & Sons 1998.
- Shao M, Kingsford C. Accurate assembly of transcripts through phase-preserving graph decomposition. *Nat Biotechnol* 2017;**35**:1167–9.
- Simmonds P, Adams MJ, Benkó M *et al.* Virus taxonomy in the age of metagenomics. *Nat Rev Microbiol* 2017;**15**:161–8.
- Steinberger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol* 2017;**35**:1026–8.
- Sutton TDS, Clooney AG, Ryan FJ *et al.* Choice of assembly software has a critical impact on virome characterisation. *Microbiome* 2019;**7**:12.
- Terzian P, Olo Ndela E, Galiez C *et al.* PHROG: families of prokaryotic virus proteins clustered using remote homology. *NAR Genom Bioinform* 2021;**3**:lqab067.
- Tomescu AI, Kuosmanen A, Rizzi R *et al.* A novel min-cost flow method for estimating transcript expression with RNA-Seq. *BMC Bioinformatics* 2013;**14** Suppl 5:S15.
- Twort F. An investigation on the nature of ultra-microscopic viruses. *Lancet* 1915;**186**:1241–3.
- Vatinlin B, Chauvet F, Chrétienne P *et al.* Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths. *Eur J Oper Res* 2008;**185**:1390–401.
- Wick RR, Schultz MB, Zobel J *et al.* Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics* 2015;**31**:3350–2.
- Xue H, Mallawaarachchi V, Zhang Y *et al.* RepBin: constraint-based graph presentation learning for metagenomic binning. *AAAI* 2022;**36**:4637–45.
- Zhang X, Studier F. Multiple roles of T7 RNA polymerase and T7 lysozyme during bacteriophage T7 infection. *J Mol Biol* 2004;**340**:707–30.