PHAGE: Therapy, Applications, and Research
Volume 2, Number 4, 2021
© Mary Ann Liebert, Inc.
DOI: 10.1089/phage.2021.0015

Phage Genome Annotation: Where to Begin and End

Anastasiya Shen, BSc, MSc[1] and Andrew Millard, BSc, PhD[2,i]

Abstract

With the renewed interest in phage research, coupled with the rising accessibility to affordable sequencing, ever increasing numbers of phage genomes are being sequenced. Therefore, there is an increased need to assemble and annotate phage genomes. There is a plethora of tools and platforms that allow phage genomes to be assembled and annotated. The choice of tools can often be bewildering for those new to phage genome assembly. Here we provide an overview of the assembly and annotation process from obtaining raw reads to genome submission, with worked examples, providing those new to genome assembly and annotation with a guided pathway to genome submission. We focus on the use of open access tools that can be incorporated into workflows to allow easy repetition of steps, highlighting multiple tools that can be used and common pitfalls that may occur.

Keywords: phage, genome assembly, genome annotation

Introduction

The recent renaissance in phage research and developments in sequencing technologies has resulted in increasing numbers of phage genomes being sequenced.[1] While phages are similar to bacteria in how they can be assembled and annotated, there are significant differences. The most important difference being the expectation for phages that their genomes can be completely assembled with short reads. Many tools developed for bacterial genome assembly and annotation work well with phage genomes, with some modifications.

Here we give an overview of phage genome assembly, structural and functional annotation, and submission to a member of The International Nucleotide Sequence Database Collaboration. We provide examples, highlighting the choices that can be made with a focus on open-source software via a Linux interface (Supplementary Data S1), which allows the process to be streamlined by the production of a workflow manager or simple bash script. Details of scripting and workflow managers are not discussed in detail other than to say tools such as NextFlow[2] and SnakeMake[3] allow the adaptation of pipelines written in bash and other scripting languages. There are also workflows available via web browsers that offer all in one solution such as PATRIC and the Galaxy interface[4] for phage genome assembly.

The stages of bacteriophage assembly and annotation are split into three areas: genome assembly, genome annotation, and genome submission. Each of these sections is broken down into subsections with suggestions on tools and pitfalls that might occur at each step. The process covers the questions raised by Turner et al.

While it might appear counterintuitive, we recommend those new to sequencing phage genomes to look at the requirements for genome submission before any genome sequencing. Understanding the data required for genome submission before starting the process can save considerable time when the time comes for final submission. For sequence submission to the European Nucleotide Archive (ENA), the current information that is required to gain an accession number is the following: a taxonomy ID, a unique locus tag, phage name, project number, depth of coverage, library insert size,

[1]Center for Evolutionary Hologenomics, University of Copenhagen, Copenhagen, Denmark.
[2]Department of Genetics and Genome Biology, University of Leicester, University Road, Leicester, United Kingdom.
[i]ORCID ID (https://orcid.org/0000-0002-3895-2854).

sequencing platform, and assembly software. Collating this information as the assembly and annotation process is completed can save considerable time when genomes are to be submitted.

Bacteriophage Genome Assembly

Presequencing

Before any genome assembly, it is important to determine in advance what is required from the final genome assembly. Bacteriophages can have different packaging strategies that will result in different genomic termini or circularly permuted genomes.[5] If it is important to know the exact termini of the genome, then the choice of sequencing is a critical factor and deciding this beforehand can prevent further experimental work later. There are many different library preparation methods for both short- and long-read sequencing.

The choice of library preparation method will be based on several factors, including the amount of input genomic DNA (particularly high for long-read sequencing), provision by service providers, the number of genomes to be sequenced, and cost. The use of transposon-based library preparation methods such as NexteraXT that uses tagmentation has the benefit of requiring minimal amounts of input DNA and providing easy automation for multiple library preparations simultaneously.[6] However, the insertion of a transposon requires some bases upstream of the insertion site, and thus, the terminal bases will never be sequenced.[6] This prevents easy identification of the termini based on sequencing data alone if transposon-based approaches are used as tools, such as PhageTerm, not compatible with such library preparations.[7]

The vast majority of bacteriophage genomes can be sequenced via the use of short-read sequencing if they are sequenced as individual isolates.[8] With limited advantages of assembly, it is possible to use longer reads as most phages isolated to date do not have repeats that prevent assembly, as can happen with large bacterial genomes.[8]

However, the use of long-read technologies such as the Oxford nanopore technology (ONT) and PacBio offers other advantages including identification of DNA modifications,[9] although these techniques are far more specialized and identification of base modification is covered here. ONT sequencing via the use of a minION does offer other advantages compared to Illumina sequencing, including rapid turnaround time from DNA to genome sequence, sequencing a single genome at a time and allowing users to sequencing within their own laboratory with minimal investment in infrastructure.[10] However, the initial genome assembly of ONT reads can be more challenging than that of Illumina sequencing. For the focus of this overview, we focus on the use of short-read Illumina sequencing and long-read nanopore sequencing.

Genome sequencing

To obtain an assembly, it is necessary to obtain reads from a phage genome. For dsDNA phages, it is now apparent this is not always a trivial process. It has been known for decades that phages can modify the nucleotides in their genomes.[11] However, there are now increasing reports of phages with hypermodified DNA that is recalcitrant to standard sequencing methods.[12–14]

To overcome this, innovative approaches of using RNA-Seq to reconstitute the genome from phage transcripts[14] or rolling circle amplifications to remove DNA modifications have been applied.[12,13] As the number of phages isolated increases and therefore sequenced, it is likely these approaches will have to be applied more often. For the purpose of this work, it is assumed that the genomic material is dsDNA from phages capable of producing virions and not recalcitrant to standard sequencing approaches.

Genome Assembly

Quality control of reads

The first step in the assembly of phage genomes is the quality control of read data and understanding how data have been produced and/or returned by a sequencing provider (Steps 1–3 in Supplementary protocol). If the sequence provider also provides an assembly of the data, it is desirable to understand how much data were used for the assembly. It is entirely possible to sequence a phage genome at >2,500× coverage for ∼£100. Having an indication of the depth of sequencing is essential for optimal assembly. Assuming raw reads are provided, an initial assessment of read quality is an essential first step to produce optimal assemblies.

The use of FASTQC allows an overview of quality statistics of all the reads that have been returned and the number of reads per sample.[15] Following examination of the quality statistics, sequencing adapters and low-quality bases can be trimmed off. Reads can be trimmed with various tools, including but not limited to trim_galore,[16] sickle,[17] and BBDuk. Within this example, we utilize the Seqtk package (Step 4 in Supplementary protocol). Given that phage genomes are generally sequenced to a very high coverage, it is recommended to be strict in the parameters that are used for quality control of the data.

Having removed adapters and trimmed off poor-quality bases and removed any control DNA spike-ins that are used with Illumina sequencing (e.g., PhiX control), the next step is the assessment of how much data are present (Step 3 in Supplementary protocol). While more data might intuitively be thought to produce better genome assemblies, this is often not the case with short-read data. Using 30× coverage allows the assembly of most phage genomes and using very high coverage of data can result in assembly problems[8] Therefore, it is suggested that reads are randomly subsampled to give ∼50–100× coverage of the genome. Given that genome sizes are rarely known before assembly, this can be roughly calculated using the number of reads, read length, and an ''average genome'' size.

(number reads × read length)/expected genome size

For example, MiGs (https://www.migscenter.com/) return on average 450 Mbp of data per genome, which is 15,000,000 paired-end reads of 150 bp in length, equating to 4,500 coverage of a 100 kb phage genome. Reads can be randomly subsampled with a variety of tools, including Seqtk toolkit,[18] BBnorm, or reformat.sh as part of the wider BBtools package. Within this example Seqtk was used (Step 4 in Supplementary protocol).

The process of quality control is different for ONT data, with raw output files being FAST5 rather than FASTQ sequences. Initially base calling is required to convert FAST5 to FASTQ, followed by demultiplexing (if multiple samples were combined in a library), which can be performed with Guppy. Further demultiplexing can be done if necessary with a variety of tools including Porechop[19] and Qcat.[20] Overall read metrics can be gained using various tools including Poretools[21] and MINIOQC.[22] There are extensive tutorials available through the nanopore community portal, providing guidance on nanopore tools.

Utilizing nanopore sequencing will generally require increased read depth compared with Illumina data. With limited publications to date using minION data for sequencing phage isolates, the optimal read coverage is unknown. Based on data available for bacterial genomes, it is likely a sequencing depth of 100–200 × will be required to assemble high-quality genomes,[23] with consensus base calling accuracy of >Q50 (99.999% accuracy) (https://nanoporetech.com/accuracy). While it is possible to assemble phage genomes from far lower coverage, these will likely contain numerous base calling errors, which may lead to the presence of artificial frameshifts within coding sequences. Further details on different assembly options using nanopore data can be found in Wick et al.[24]

It is also possible to assemble phage genomes using a hybrid approach, combining long and short reads. For the sequencing of complex phage communities (viromes), this approach is advantageous.[25,26] However, for individual phage isolates, the approach has minimal benefits and will add considerable costs and thus not recommended for the majority of phage genome sequencing projects.

Assembly

There are numerous genome assemblers suitable for phage genomes, including but not limited to SPAdes,[27] Megahit,[28] Ray,[29] and MetaVelvet.[30] Previous research found that most assemblers produce consistent genome assemblies.[8] We recommend the use of SPAdes with the option ''–only-assembler,'' based on previous analyses.[8] If for some reason the input DNA was amplified before sequencing (e.g., to remove DNA modifications or meet input requirements), then the ''–sc'' flag for multiple displacement amplifications is advantageous. With SPAdes, the output file ''contigs.fasta'' will contain the resultant assembly that may contain a single contig or multiple contigs (Step 5 in Supplementary protocol). Other assembly programs will produce very similar outputs but likely have a different output name.

There are numerous assemblers for ONT reads, including SPAdes,[27] FLYE,[31] wtdgb2,[32] Unicycler,[33] and Tricycler.[24] Given the relatively small number of phage genomes sequenced to date with this technology and rapidly evolving software, a consensus on the most appropriate assembler has not been reached for phages. The assembly with any of these tools may require some optimization of parameters as detailed for each tool. The following assembly may also require rounds of polishing to correct errors, which can be achieved using a variety of tools including Medaka[34] and Pilon.[35] The tool FLYE was used as an example in this work (Step 17 in Supplementary protocol). Once a genome has been assembled, the annotation steps are the same regardless if the genome was assembled from Illumina or ONT reads.

Assembly validation

At this stage, it is important to check the assembled contigs to help determine how well the assembly has worked and identify any errors. If SPAdes was used, the *contigs.fa* output file will contain the contigs of interest (Step 5 in Supplementary protocol).

Preliminary checking of contigs

It is most likely that the resulting contigs from the assembly will contain a single contig with high coverage, while the remaining contigs are very short with low coverage, this is indicative of assembly of a complete phage genome. Variations from this will require some further work to identify what they are. Contigs from contaminating host DNA need to be removed before phage genome annotation. Assemblies can be assessed by the use of Bandage[36] that allows the assembly-graphs to be visualized. The input file for Bandage is an assembly graph file rather than a FASTA file; if SPAdes is used for assembly this will be a *_.fastg_ file.

In conjunction with genome coverage, the assembly graph can be used to view the assembly and to identify common assembly-associated issues (Steps 6–8 in Supplementary protocol). When viewing the assembly graphs, it is important to note that while phage genomes may appear circular, this does not mean that the genome is circular. The circularity arises as an artifact either from a circularly permuted genome or terminal repeats making it appear circular (Fig. 1, see Q4 in Turner et al.).

Depth of sequencing coverage per contig

The absolute read coverage per contig can be obtained by mapping reads to a set of contigs. There are multiple tools to map reads, including BWA-MEM,[37] Bowtie2,[38] bbmap.sh,[39] and minimap2,[40] all of which are suitable for mapping, and coverage can be calculated using other tools. Using bbmap.sh allows read coverage of each contig to be directly calculated as an output file with the ''–covstats'' flag, without the need for another tool (Step 7 in Supplementary protocol).

The resultant output file allows coverage for each contig to be assessed, which is a useful parameter to assess how well the assembly has worked. In most cases, this consists of a single large contig with high coverage and smaller contigs with far lower coverage. This is indicative of a complete phage genome with some background host DNA. If the coverage is very high >200 ×, further subsampling of reads is recommended. Alternatively, if a low coverage (<20 ×) of the largest contig is obtained, subsampling and keeping a higher proportion of reads are recommended.

Based on the results obtained, it might be necessary to resample the reads and repeat the assembly process. This may well be an iterative process requiring multiple subsamplings and assembly runs (see Q1 in Turner et al.).

If reads were first subsampled, it is also good practice to map all the short reads that passed quality control, providing statistics of depth of sequencing coverage for the sample, which is required later for genome submission and can be used for error correction with pilon.[35] Reads that do not map to the putative phage contig can also be collected at this stage. The rationale for doing this is to check if other phages are present. For some bacterial hosts, low-level induction of prophages can occur, and this will not have been detected in the reads that have been subsampled and assembled.

The removal of reads associated with the most abundant phage contig increases the ability to identify any prophages

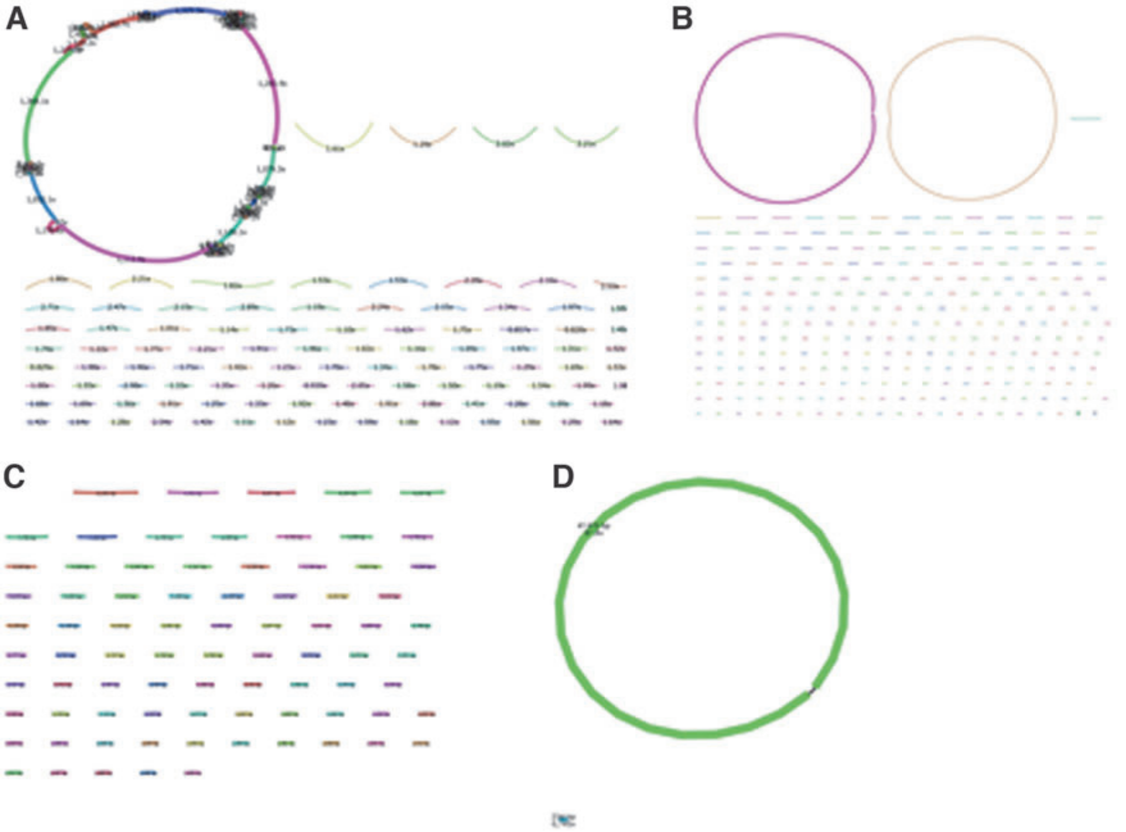


FIG. 1. **(A)** A single-phage contig is present but not assembled. The different colors represent individual contigs. The large circle suggests that a single circular contig is present, but bubbles in the assembly graph prevent assembly into a single contig, likely caused by excess genome coverage. Subsampling reads to a lower depth of coverage allows assembly in this case. **(B)** Two phages present in a single sample (based on two circular contigs), based on coverage, one is more abundant than the other. **(C)** No phage present. Large numbers of contigs over 10 kb in length, without a set of contigs. **(D)** A single contig was assembled. It is important to note that the bandage presents the complete contigs as circular; this does not mean the genomes are circular. In addition, the coverage data in Bandage are scaled and thus proportional, they do not show absolute genome coverage.

that have been induced at a low level in a subsequent assembly. To do this, all reads that were not mapped to the putative phage should be extracted and reassembled with SPAdes (or any other assembler).

Preliminary identification of closest relatives

To aid in genome annotation, it is useful to identify if the putative phage genome is similar to previously sequenced phages (Step 9 in Supplementary protocol, see Q3 in Turner et al.). There are multiple ways to identify close relatives, including the INPHARED database and associated scripts, which provide a command line interface to rapidly compare against complete phage genomes in GenBank.[1] Alternatively, BLASTn against the Virus database can be utilized to search via a web interface using the taxaID Viruses: 10239 to search only against viruses.

The similarity to other phages and the sequencing library method used determines the next steps in annotation. This also helps early on to determine if a complete phage genome has been assembled, as any closely related phage genome would be expected to have a similar genome length. The recent development of PhageClouds offers a web interface to rapidly compare against a database of >600,000 viruses with a visual representation of genome size that helps judge completeness based on similarity and size of other phages. Alternatively, a prediction of genome completeness can be assessed using the command line tool checkV, which also indicates termini type[41] (see Q2 in Turner et al.).

Genome reordering

It is important to note that assemblers do not output genomes that are ordered correctly; this does not mean the assembly is wrong, rather it may require reorientation as it is not designed to identify terminal repeats or identify circularly permuted genomes (see Q4 in Turner et al.). How reordering is approached will depend on the library preparation method and how related to other phages the new phage genome is (Fig. 2).

If a nontransposon-based library preparation was used, then PhageTerm can be used to predict the packaging strategy of the phage and automatically reorder the genome if it has defined termini.[7] Alternatively, if a transposon-based approach was used, other approaches will be required. If the preliminary phage identification resulted in it being similar to other phages (>95% ANI), it is possible to use the information from a close relative for genome reordering, if the

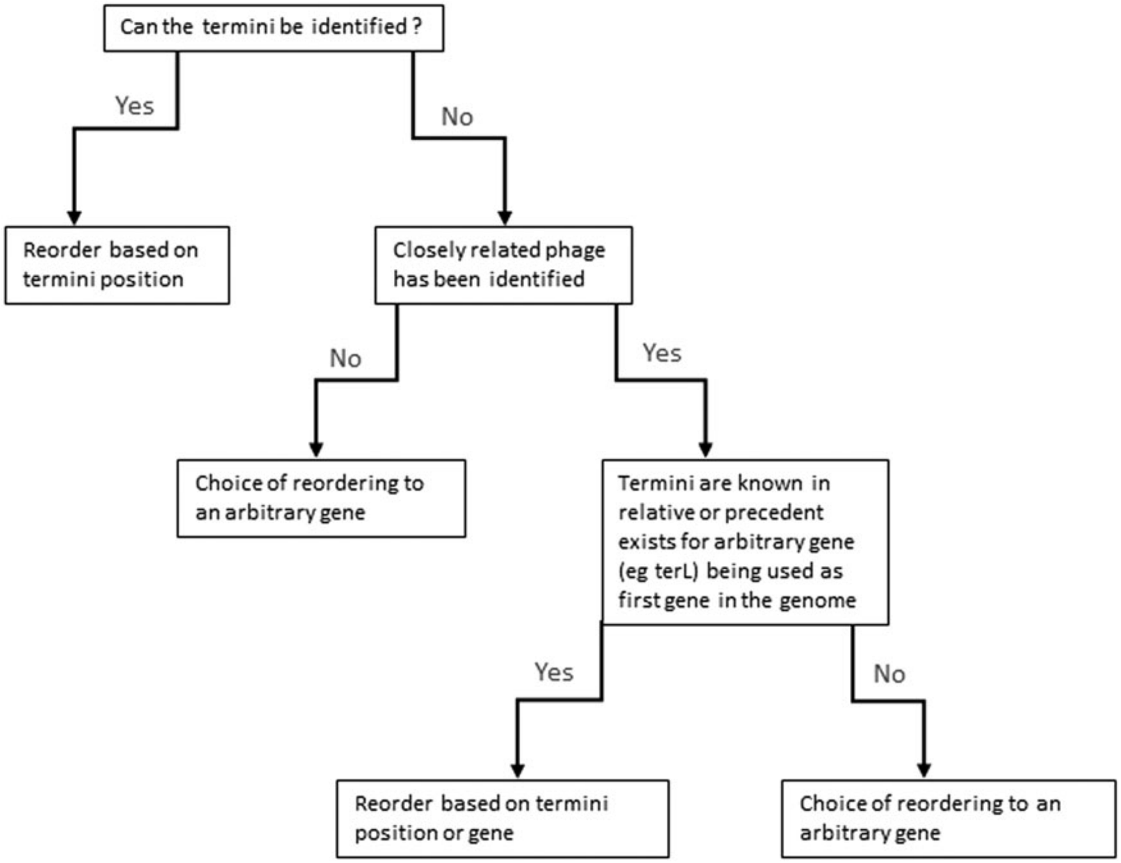

Can the termini be identified ?
Yes
No
Reorder based on termini position
Closely related phage has been identified
No
Yes
Choice of reordering to an arbitrary gene
Termini are known in relative or precedent exists for arbitrary gene (eg terL) being used as first gene in the genome
Yes
No
Reorder based on termini position or gene
Choice of reordering to an arbitrary gene

FIG. 2. Decision process for reordering of phage genomes.

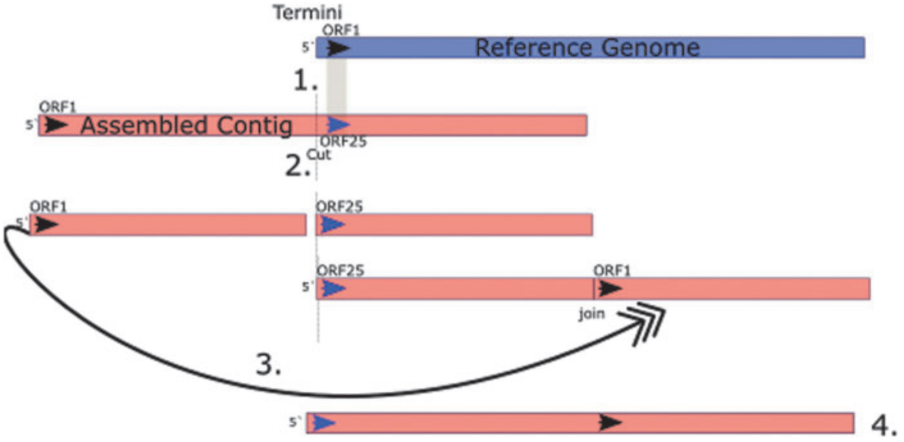
termini of a relative have been identified previously. The outputs of CheckV also identify terminal repeats within sequences (see Q2 in Turner et al.). Otherwise, a specific gene may be arbitrarily selected as the start point of close relatives (e.g., terL, split between terS and terL [see Q2 in Turner et al.]). Utilizing this gene and maintaining consistency across similar genomes have some benefits of visualizing genomic comparisons.

Previously, it has been suggested that the large terminase subunit should be set as the first gene, if termini have not been identified.[42] However, the benefits of this are limited if the phage is not closely related to any other phages. It is important to state if the termini have been identified and, if not, to ensure that the arbitrary start point is not intragenic.

Genome reordering can be achieved manually by first identifying the new first base in the genome and manually editing the sequence, so any bases upstream of the new first base are moved to the 3′ of the genome (Fig. 3). If the genomes are closely related, Mummer4 allows rapid whole-genome alignments.[43] Whole-genome alignments of very closely related phages can be achieved with other tools such as MAFFT[44] or MAUVE.[45]

Alternatively, an initial annotation can be performed to identify the gene that will be used as the arbitrary start of the

FIG. 3. Reordering phage genomes. (1) Align the genome against a reference genome and identify the gene/start point that will become the new start position. (2) Cut the genome upstream of the position that will become the new 5′ start. (3) Move the cut fragment to the left of the new start position to the 3′ end and join. (4) Check for assembly errors with Pilon and recall all genes.



genome, for example, terL. Ordering the genome at this point can save considerable time later. Whenever a genome is reordered and before final gene calling, the use of plion[35] or REAPR[46] is advised to ensure no errors have been introduced and correct any local mis-assemblies along with erroneous base calls in the original assembly.[35]

Troubleshooting genome assembly

While most phage genomes will assemble, there are always those that do not. There are several reasons for incomplete assembly, a common reason being very high or low read coverage (Table 1). Thus, it is recommended to subsample reads before assembly. In addition, there are other reasons that a single-phage contig is not assembled. Including bias in mol GC% content, with transposon-based library preparations well known to target AT-rich regions or undercut high GC regions that can lead to broken assemblies across these regions.[47] Alternatively, the presence of multiple very similar phage genomes with high microdiversity can cause assemblies to break.[48] Some of these issues can be resolved bioinformatically by viewing assembly graphs (*.fastg from SPAdes output) in Bandage and the use of polymerase chain reaction to sequence any gaps in the genome.

Genome Annotation

Gene calling (structural annotation) (see Q5 in Turner et al.)

There are multiple stand-alone tools for gene calling and the prediction of tRNAs. For gene calling, Prodigal,[49] GeneMarkS,[50] Glimmer,[51] and MetaGeneAnnotator[52] are commonly used and are available through a command line interface or web interface. Recently, a phage-specific gene calling algorithm has been developed (Phanotate) that incorporates several of the above tools to identify a consensus from multiple tools.[53] For the identification of tRNAs, Aragorn,[54] tRNAFinder,[55] or tRNAScan-SE[56] can be utilized. If these tools are run alone, the results must be collated into a single coordinate file to overlay the information onto a genome using tool such as Artemis[57] or Ugene.[58]

Artemis and Ugene also allow users to manually BLAST each sequence to provide a functional annotation or export all predicted gene sequences to allow a batch BLAST analysis (see Q6 in Turner et al.).

Alternatively, Prokka can be used that combines the process of coding sequences, annotation of genes, and formatting of files for submission to an International Nucleotide Sequence Database. Prokka has considerable advantages as it allows rapid and consistent annotation of a genome with formatting of files ready for submission with minimal reformatting. Prokka also provides files that can be directly used as input for downstream genome analysis. Gene calling in Prokka is provided by Prodigal, with tRNA identification through ARAGORN.[54]

In addition, CRISPR arrays and tRNAs/tmRNAs are searched for by default. The inbuilt databases are primarily designed for bacteria, but are easily adapted for bacteriophages. Prokka prescribes a function through a hierarchical approach, using a BLAST database, followed by more sensitive Hidden Markov models (HMMs). For the more experienced user, a series of scripts can be utilized to create a specific database from a set of genomes, for instance, all current phage genomes. It also allows addition of HMM databases, allowing the use of prebuilt databases such as the pVOG database[59] or PHROGs.[60]

The use of PHROGs in particular provides good initial functional annotation of phage proteins. Alternatively, using the ‘‘–proteins'' option allows the user to provide a GenBank file used for annotation, which is particularly useful if a closely related phage is already well annotated (Steps 13–15 in Supplementary protocol).

The output of Prokka provides a good starting point for further annotation and eventual submission to a database. The GenBank file can be read directly into graphical interfaces such as Artemis or Ugene for further annotation and manipulation or can be manipulated directly with a text editor. Often the automatic annotation will result in uninformative product descriptions, for example, ‘‘gp23,'' which is ambiguous and needs correction to describe a function such as ‘‘major capsid protein'' (see Q7 in Turner et al.). The choice of database used for annotation can have significant effects on the number of proteins that are assigned a potential function.

Annotation of virion structural genes

With increasing amounts of phage genomic data, machine learning approaches have been utilized to make predictors for specific gene types. Machine learning was utilized to distinguish between structural virion and nonvirion proteins, although successful was not easily implemented into pipelines.[61] Recently, more user-friendly approaches have been

Table 1. Common Results Obtained from Sequencing Data

Result	*Indicative of*	*Action*
Single contig (circular) at 50–200 × coverage	A single complete phage genome	Proceed to determine if it is a complete phage.
Single contig (circular) at 50–200 × coverage with multiple small contigs at low coverage	A single complete phage genome with some background host DNA	Proceed to determine if it is a complete phage and remove host contigs.
Two large contigs (>20 kb) with multiple smaller contigs at low coverage	Two complete phage genomes or single broken phage assembly	Look for differences in coverage of contigs, for example, 1000 × versus 50 × suggests they are different pieces of DNA.
Multiple large contigs (10 kb plus)	Broken phage assembly, multiple different phages present, or high levels of host DNA present	Repeat assembly with reads subsampled at different depths.

developed to classify phage structural proteins that allow access via a web server (http://edwards.sdsu.edu/phanns) or integration into pipelines via the command line.[62] For phages that have limited similarity to well-characterized phages, this can help considerably in ascribing predicted gene function.

Template-based homology searching

Further predicted protein functions can be gained using comparison of phage proteins with known protein structures, through a process of template-based modeling. This approach is available through the Phyre2 website (www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index) that allows users to submit individual sequences or batches of jobs.[63] Given the computational cost and time required, there is limited benefit of running Phyre2 analysis on well-annotated proteins such as DNA polymerase and there is more benefit for proteins that do not have any form of functional annotation initially.

Specific annotation for therapeutic phages

For phages considered for use in therapy, there is considerable interest for determining whether they are strictly lytic and if they carry antimicrobial resistance genes or genes that might alter their bacterial host virulence (see Q8 in Turner et al.). The presence of antibiotic resistance genes in lytic phages is a rare event,[1,64] while carriage of antibiotic resistance genes and virulence genes in temperate phages is more common.[1]

The software ABRicate allows users to rapidly search against a range of antibiotic resistance gene and virulence databases for such genes.[65] Alternatively AMRFinder (https://www.ncbi.nlm.nih.gov/pathogens/antimicrobial-resistance/AMRFinder/) or graphical interfaces to the CARD database (https://card.mcmaster.ca/)[66] and the Virulence factor database (www.mgc.ac.cn/cgi-bin/VFs/v5/main.cgi?func=VFanalyzer) are also available.[67] Careful use of thresholds should be used for identification of such genes[1,64] to avoid over interpretation (see Q8 in Turner et al.).

Prediction of lytic and temperate lifestyles

There are many tools available for prediction of the lifestyle of phages, and in the case of phage therapy, this is particularly relevant where the avoidance of temperate phages is preferred (see Q8 in Turner et al.). These include BACPHLIP,[68] PhageAI,[69] and PHACTS,[70] all of which offer web interfaces for prediction with stand-alone code available for PHACTS and BACPHLIP to allow integration into pipelines with application programming interface access to PhageAI.

More recently, tools that simultaneously predict the lytic nature of a phage and the presence of antibiotic resistance genes (ARGs) or virulence factors have been developed for example, PhageLeads (http://130.226.24.116/phagecompass/index.html#/PhageLeads). As with all tools, these are only predictions of the life cycle that require experimental validation. The recent characterization of the cyanophage S-TIP37 highlights that the presence of an integrase gene does not mean a phage will enter a lysogenic life cycle.[71]

Introns and Inteins

Since the discovery of an intron in the thymidylate synthase gene of phage T4,[72] there have been increasing numbers of reports of introns in a variety of genes in phages, including ribonucleotide reductase,[73] terminase subunits,[74] DNA polymerase,[75] and the core photosynthesis gene *psbA*.[76] Current tools for gene prediction in phages will not correctly predict the presence of introns and therefore will require manual inspection. A telltale sign of an intron is the annotation of two genes that are adjacent to each other with the same function, and this often results as the intron introduces a premature stop codon into the gene. By alignment of suspected intron containing genes with homologues lacking introns, it is often possible to identify the insertion site of introns through manual inspection of alignments (see Q5 in Turner et al.).

Inteins, which are autocatalytic internal proteins that are capable of excising themselves from the surrounding extein protein, are also found within phages. Similar to introns, they are often found in conserved phage genes such as those encoding ribonucleotide reductase,[77] DNA polymerase,[78,79] and terminase subunits.[12] In common with introns, these will not be identified by gene calling software and will require manual identification and curation.

Genome submission

In the interests of open data sharing, it is good practice to submit genome sequences before peer review of any publication of the work. Through this work, we have tried to identify the information that is required for submission of phage genomes through the ENA, which has a submission guide (https://ena-docs.readthedocs.io/en/latest/submit/general-guide/webin-cli.html). For submission to GenBank or DDBJ, similar information is required, although the process will differ. We have included a checklist of requirements for ENA (Table 2). For submission to the ENA, the registration of a study is required to gain a project number (https://ena-docs.readthedocs.io/en/latest/submit/study/interactive.html). At this point, unique locus_tags can also be registered. Both raw reads and an assembled genome can be submitted. If the phage is new, it may require the registration of a taxaID.

To complete the submission of raw reads and an assembled genome, a manifest file will be required, which also requires a sample accession (requires registration), depth of sequencing coverage (calculated previously), assembly program,

Table 2. Check List of Information Required That Should Be Obtained Before Genome Submission

Information	*Notes*
Taxonomy ID	If the phage represents a new taxon, this must be requested before submission.
Locus tag	Must be registered before genome submission. Can be registered before starting the sequencing project when a project number is registered.
Project no.	Can be registered before sequencing of the phage genome, required for submission.
Sequencing depth	Provided by the sequencing provider or calculated during assembly.
Assembly software	Chosen during the assembly process.
Phage name	Required for submission and useful to have decided before sequencing to avoid renaming issues through a project.

and sequencing platform (https://ena-docs.readthedocs.io/en/latest/submit/assembly/genome.html?highlight=manifest#chromosome-assembly). For submitting the annotated genome to the ENA, the General Feature Format (GFF) file from Prokka needs to be converted to EMBL format and a manifest file produced. A GFF file can be converted to EMBL with the GFF3toEMBL script[80] with a manifest file created in a text editor (see Q10 and Q11 in Turner et al.).

Comparative Genomic Analysis and Beyond

Taxonomic classification

Complete comparative genomic analysis encompasses a huge area of research that goes beyond the scope of this work. Here we provide a brief overview of software that is available for comparative genomic analysis. For taxonomic classification of phages at the level of species and genus, there are now clear guidelines from the International Committee on Taxonomy of Viruses (ICTV) based on nucleotide identity.[81] As stated previously, VIRIDIC[82] offers an automated solution for comparison of phage genomes and identification of genera and species based on the nucleotide sequence. Other tools can be used and have been detailed elsewhere.[81]

Identification of relationships beyond the genus level requires a more in-depth analysis. The development of all versus all protein analyses offers identification of subfamily- and family-level classification. The development of vContact2 places new phage genomes into viral clusters based on a network graph.[83] vContact2 is available as a stand-alone tool or via web interface through iVirus, with preformatted databases also available as input.[1]

Alternative methods are also available via online tools including VipTree, which builds on the original phage proteomic tree,[84] with a simple interface that allows uploading of phage genomes.[85] One disadvantage of this online version is that it only utilizes the RefSeq database for phages, which is not representative of total phage genomes.[1] The stand-alone version does allow custom databases, but is slightly more involved to run. Classification can also be achieved using VICTOR, which allows users to upload genomes.[86] While the underlying algorithm can be scaled for comparisons and placement of thousands of genomes, the current web implementation is severely limited by the number of genomes that can be uploaded.

Further classification can be achieved by phylogenetic analysis based on single genes or core genes. The output files from Prokka (*gbk files) have the advantage that they can be directly input into comparative genomic software. For core-gene analysis Roary[87] and/or GET_HOMOLOGUES[88] offer a relatively easy pathway into core-gene identification. In the case of Roary, the resulting core-gene alignment file can be directly used as input for IQ-Tree for further phylogenetic analysis.

The use of GET_HOMOLOGUES in combination with GET_PHYLO_MARKERS[89] offers a complete workflow for phylogenetic analysis based on core genes. The approach of aligning core genes is preferable to attempting to align the complete genomes, which is not a suitable method due to poor alignment and incorporation of intergenic regions into the underlying alignments. For the classification of phage above the genus level in particular, contacting the ICTV for guidance is recommended (see Q13 in Turner et al.).

Microdiversity

When sequencing a phage genome, although the phage will likely have been purified through several rounds of purification, the extracted DNA is from a population of phages. Obtaining very high-coverage genome sequencing provides the opportunity to identify variations (single nucleotide polymorphisms [SNPs] and indels) within the population. There are multiple tools for SNP and indel detection, including FreeBayes,[90] SNIPPY, and VarScan2.[91]

Conclusion

There are multiple methods for genome assembly, all of which have pros and cons. Here we have focused on open-source software that can be integrated into pipelines. The initial installation of software comes with a time cost. However, once installed, many of the steps explained here can then be easily repeated or automated with minimal effort. By providing a walk-through example, we hope to encourage others to utilize such tools and provide a reference point for others to begin their phage genome annotation journey.

Author Disclosure Statement

No competing financial interests exist.

Funding Information

A.M. was funded through UK Medical Research Council grant CLIMB-BIGDATA (MR/L015080/1). N.S. was funded through the FHF—Norwegian Seafood Research Fund (''Genomic epidemiology and Phage-based Prevention of salmon associated Pasteurella,'' grant No. 901707).

Supplementary Material

Supplementary Data S1
Supplementary Table S1
Supplementary Table S2

References

1. Cook R, Brown N, Rihtman B, et al. INfrastructure for a PHAge REference Database: Identification of large-scale biases in the current collection of phage genomes. bioRxiv 2021:2021.05.01.442102. doi: 10.1101/2021.05.01.442102.
2. Di Tommaso P, Chatzou M, Floden EW, et al. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017;35(4):316–319.
3. Mölder F, Jablonski KP, Letcher B, et al. Sustainable data analysis with Snakemake. F1000Res. 2021;10:33.
4. Ramsey J, Rasche H, Maughmer C. Galaxy and Apollo as a biologist-friendly interface for high-quality cooperative phage genome annotation. PLoS Comput Biol. 2020;16(11):e1008214.
5. Black LW. DNA packaging in dsDNA bacteriophages. Annu Rev Microbiol. 1989;43:267–292.
6. Picelli S, Björklund AK, Reinius B, et al. Tn5 transposase and tagmentation procedures for massively scaled sequencing projects. Genome Res. 2014;24(12):2033–2040.
7. Garneau J, Depardieu F, Fortier L-C, et al. PhageTerm: a tool for fast and accurate determination of phage termini and packaging mechanism using next-generation sequencing data. Sci Rep. 2017;7:8292.

8. Rihtman B, Meaden S, Clokie MRJ, et al. Assessing Illumina technology for the high-throughput sequencing of bacteriophage genomes. Peer J. 2016;4:e2055.
9. Kot W, Olsen NS, Nielson TK, et al. Detection of preQ0 deazaguanine modifications in bacteriophage CAjan DNA using Nanopore sequencing reveals same hypermodification at two distinct DNA motifs. Nucleic Acids Res. 2020; 48(18):10383–10396.
10. González-Escalona N, Allard MA, Brown EW, et al. Nanopore sequencing for fast determination of plasmids, phages, virulence markers, and antimicrobial resistance genes in Shiga toxin-producing *Escherichia coli*. PLoS One 2019;14(7):e0220494.
11. Thomas J, Orwenyo J, Wang L-X, et al. The odd 'RB' phage—Identification of arabinosylation as a new epigenetic modification of DNA in T4-like phage RB69. Viruses. 2018;10(6):313.
12. Rihtman B, Puxty RJ, Hapeshi A, et al. A new family of globally distributed lytic roseophages with unusual deoxythymidine to deoxyuridine substitution. Curr Biol. 2021; 31(14):3199.e4–3206.e4.
13. Korn AM, Hillhouse AE, Sun L, et al. Comparative genomics of three novel jumbo bacteriophages infecting *Staphylococcus aureus*. J Virol. 2021;95(19):e0239120.
14. Leskinen K, Pajunen MI, Vilanova MVG, et al. YerA41, a *Yersinia ruckeri* bacteriophage: Determination of a non-sequencable DNA bacteriophage genome via RNA-sequencing. Viruses 2020;12(6):620.
15. Bioinformatics B. *FastQC: A Quality Control Tool for High Throughput Sequence Data*. Cambridge, United Kingdom: Babraham Institute; 2011.
16. Krueger F. Trim galore. A wrapper tool around Cutadapt and FastQC to consistently apply quality and adapter trimming to FastQ files. [Software]. 2015. Available at https://github.com/FelixKrueger/TrimGalore (accessed November 2, 2021).
17. Joshi NA, Fass JN. Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files. (Version 1FV3) [Software]. 2011. Available at https://github.com/najoshi/sickle (accessed November 2, 2021).
18. Li H. seqtk Toolkit for processing sequences in FASTA/Q formats. GitHub. Available at https://github.com/lh3/seqtk (accessed November 2, 2021).
19. Wick RR. Porechop. Github. 2017. Available at https://githubcom/rrwick (accessed November 2, 2021).
20. qcat. Available at https://github.com/nanoporetech/qcat (accessed November 2, 2021).
21. Loman NJ, Quinlan AR. Poretools: A toolkit for analyzing nanopore sequence data. Bioinformatics. 2014;30(23): 3399–3401.
22. Lanfear R, Schalamun M, Kainer D, et al. MinIONQC: Fast and simple quality control for MinION sequencing data. Bioinformatics. 2019;35(3):523–525.
23. Neal-McKinney JM, Liu KC, Lock CM, et al. Comparison of MiSeq, MinION, and hybrid genome sequencing for analysis of *Campylobacter jejuni*. Sci Rep. 2021;11(1):5676.
24. Wick RR, Judd LM, Cerdeira LT, et al. Trycycler: Consensus long-read assemblies for bacterial genomes. Genome Biol. 2021;22(1):266.
25. Cook R, Hooton S, Trivedi U, et al. Hybrid assembly of an agricultural slurry virome reveals a diverse and stable community with the potential to alter the metabolism and virulence of veterinary pathogens. Microbiome. 2021;9(1):65.
26. Zablocki O, Michelsen M, Burris M, et al. VirION2: A short- and long-read sequencing and informatics workflow to study the genomic diversity of viruses in nature. PeerJ 9: e11088.
27. 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(5):455–477.
28. Li D, Liu CM, 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(10):1674–1676.
29. Boisvert S, Laviolette F, Corbeil J. Ray: Simultaneous assembly of reads from a mix of high-throughput sequencing technologies. J Comput Biol. 2010;17(11):1519–1533.
30. 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(20):e155.
31. Kolmogorov M, Yuan J, Lin Y, et al. Assembly of long, error-prone reads using repeat graphs. Nat Biotechnol. 2019;37(5):540–546.
32. Ruan J, Li H. Fast and accurate long-read assembly with wtdbg2. Nat Methods 2020;17(2):155–158.
33. Wick RR, Judd LM, Gorrie CL, et al. Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. PLoS Comput Biol. 2017;13(6):e1005595.
34. Medaka. Available at https://github.com/nanoporetech/medaka (accessed November 2, 2021).
35. Walker BJ, Abeel T, Shea T, et al. Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One. 2014;9(11):e112963.
36. Wick RR, Schultz MB, Zobel J, et al. Bandage: Interactive visualization of de novo genome assemblies. Bioinformatics. 2015;31(20):3350–3352.
37. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv. 2013;3.
38. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9(4):357–359.
39. Bushnell B. BBMap: A fast, accurate, splice-aware aligner. 2014. Available at https://www.osti.gov/biblio/1241166 (accessed November 2, 2021).
40. Li H. Minimap2: Pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–3100.
41. Nayfach Camargo AP, Schulz F, et al. CheckV assesses the quality and completeness of metagenome-assembled viral genomes. Nat Biotechnol. 2021;39(5):578–585.
42. Philipson CW, Voegtly LJ, Lueder MR, et al. Characterizing phage genomes for therapeutic applications. Viruses. 2018;10(4):188.
43. Kurtz S, Phillippy A, Delcher AL, et al. Versatile and open software for comparing large genomes. Genome Biol. 2004;5(2):R12.
44. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. Mol Biol Evol. 2013;30(4):772–780.
45. Rissman AI, Mau B, Biehl BS, et al. Reordering contigs of draft genomes using the Mauve Aligner. Bioinformatics. 2009;25(16):2071–2073.
46. Hunt M, Kicuchi T, Sanders M, et al. REAPR: A universal tool for genome assembly evaluation. Genome Biol. 2013; 14(5):R47.
47. Browne PD, Nielsen TK, Kot W, et al. GC bias affects genomic and metagenomic reconstructions, underrepresenting GC-poor organisms. GigaScience. 2020;9(2):giaa008.
48. Warwick-Dugdale J, Solonenko N, Moore K, et al. Long-read viral metagenomics captures abundant and micro-

diverse viral populations and their niche-defining genomic islands. Peer J. 2019. doi: 10.7717/peerj.6800.

49. Hyatt D, Chen GL, Locascio PF, et al. Prodigal: Prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics. 2010;11:119.
50. Besemer J, Lomsadze A, Borodovsky M. GeneMarkS: A self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. Nucleic Acids Res. 2001;29(12): 2607–2618.
51. Delcher AL, Harmon D, Kasif S, et al. Improved microbial gene identification with GLIMMER. Nucleic Acids Res. 1999;27(23):4636–4641.
52. Noguchi H, Taniguchi T, Itoh T. MetaGeneAnnotator: Detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. DNA Res. 2008;15(6):387–396.
53. McNair K, Zhou C, Dinsdale EA, et al. A. PHANOTATE: A novel approach to gene identification in phage genomes. Bioinformatics. 2019;35(22):4537–4542.
54. Laslett D, Canback B. ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. Nucleic Acids Res. 2004;32(1):11–16.
55. Hauth A, Fichant GA, Burks C. Identifying potential tRNA genes in genomic DNA sequences: TRNASCAN Version 2.0. J Mol Biol. 1995;220:659–671.
56. Lowe TM, Chan PP. tRNAscan-SE On-line: Integrating search and context for analysis of transfer RNA genes. Nucleic Acids Res. 2016;44(W1):W54–W57.
57. Rutherford K, Parkhill J, Crook J, et al. Artemis: Sequence visualization and annotation. Bioinformatics. 2000;16(10): 944–945.
58. Okonechnikov K, Golosova O, Fursov M, et al. Unipro UGENE: A unified bioinformatics toolkit. Bioinformatics. 2012;28(8):1166–1167.
59. Grazziotin AL, Koonin EV, Kristensen DM. Prokaryotic Virus Orthologous Groups (pVOGs): A resource for comparative genomics and protein family annotation. Nucleic Acids Res. 2017;45(D1):D491–D498.
60. Terzian P, Olo Ndela E, Galiez C, et al. PHROG: Families of prokaryotic virus proteins clustered using remote homology. NAR Genom Bioinform. 2021;3(3):lqab067.
61. Zhang L, Zhang C, Gao R, et al. An ensemble method to distinguish bacteriophage virion from non-virion proteins based on protein sequence characteristics. Int J Mol Sci. 2015;16(9):21734–21758.
62. Cantu VA, Salamon P, Seguritan V, et al. PhANNs, a fast and accurate tool and web server to classify phage structural proteins. PLoS Comput Biol. 2020;16(11): e1007845.
63. Kelley LA, Mezulis S, Yates CM, et al. The Phyre2 web portal for protein modeling, prediction and analysis. Nat Protoc. 2015;10(6):845–858.
64. Enault F, Briet A, Bouteille L, et al. Phages rarely encode antibiotic resistance genes: A cautionary tale for virome analyses. ISME J. 2017;11(1):237–247.
65. Seemann T. ABRicate: Mass screening of contigs for antimicrobial and virulence genes. Department of Microbiology and Immunology, The University of Melbourne, Melbourne, Australia. 2018. Available at https://github.com/tseemann/abricate (last accessed on February 28, 2019).
66. Alcock BP, Raphenya AR, Lau TTY, et al. CARD 2020: Antibiotic resistome surveillance with the comprehensive antibiotic resistance database. Nucleic Acids Res. 2020; 48(D1):D517–D525.
67. Liu B, Zheng D, Jin Q, et al. VFDB 2019: A comparative pathogenomic platform with an interactive web interface. Nucleic Acids Res. 2019;47(D1):D687–D692.
68. Hockenberry AJ, Wilke CO. BACPHLIP: Predicting bacteriophage lifestyle from conserved protein domains. Peer J 2021;9:e11396.
69. Tynecki P, Guziński A, Kazimierczak J, et al. PhageAI—Bacteriophage life cycle recognition with machine learning and natural language processing. bioRxiv 2020;2020.07.11.198606. doi: 10.1101/2020.07.11.198606.
70. McNair K, Bailey BA, Edwards RA. PHACTS, a computational approach to classifying the lifestyle of phages. Bioinformatics. 2012;28(5):614–618.
71. Shitrit D, Hackl T, Laurenceau R, et al. Genetic engineering of marine cyanophages reveals integration but not lysogeny in T7-like cyanophages. ISME J. 2021. [Epub ahead of print]; DOI: 10.1038/s41396-021-01085-8
72. Maley GF, Maley F, Belfort M. Intervening sequence in the thymidylate synthase gene of bacteriophage T4. Proc Natl Acad Sci U S A. 1984;81(10):3049–3053.
73. Dwivedi B, Xue B, Lundin D, et al. A bioinformatic analysis of ribonucleotide reductase genes in phage genomes and metagenomes. BMC Evol Biol. 2013; 13:33.
74. Daniel A, Bonnen PE, Fischetti V Daniel. First complete genome sequence of two Staphylococcus epidermidis bacteriophages. J Bacteriol. 2007;189(5):2086–2100.
75. Sauder AB, Quinn MR, Brouillette A, et al. Genomic characterization and comparison of seven Myoviridae bacteriophage infecting Bacillus thuringiensis. Virology. 2016;489:243–251.
76. Millard A, Clokie MRJ, Shub DA, et al. Genetic organization of the psbAD region in phages infecting marine Synechococcus strains. Proc Natl Acad Sci U S A. 2004; 101(30):11007–11012.
77. Lazarevic V, Soldo B, Düsterhöft A. Introns and intein coding sequence in the ribonucleotide reductase genes of Bacillus subtilis temperate bacteriophage SPβ. Proc Natl Acad Sci U S A. 1998;95(4):1692–1697.
78. Imam M, Alrashid B, Patel F, et al. vB_PaeM_MIJ3, a novel jumbo phage infecting *Pseudomonas aeruginosa*, possesses unusual genomic features. Front Microbiol. 2019; 10:2772.
79. Rajarajan S, Ibrahim KS, Pandian SK. AP-APSE dpol intein: A novel family A DNA polymerase intein domain. Bioinformation. 2011;6(4):149–152.
80. J. Page A, Steinbiss S, Taylor B, et al. GFF3toEMBL: Preparing annotated assemblies for submission to EMBL. J Open Source Softw. 2016;1:80.
81. Turner D, Kropinski AM, Adriaenssens EM. A roadmap for genome-based phage taxonomy. Viruses 2021;13(3): 506.
82. Moraru C, Varsani A, Kropinski AM. VIRIDIC—A novel tool to calculate the intergenomic similarities of. Viruses. 2020;12(11):1268.
83. Bolduc B, Jang HB, Doulcier G, et al. vConTACT: An iVirus tool to classify double-stranded DNA viruses that infect Archaea and Bacteria. Peer J. 2017;5:e3243.
84. Rohwer F, Edwards R. The phage proteomic tree: A genome-based taxonomy for phage. J Bacteriol. 2002;184: 4529–4535.

85. Nishimura Y, Yoshida T, Kuronishi M, et al. ViPTree: The viral proteomic tree server. Bioinformatics. 2017. doi: 10.1093/bioinformatics/btx157.
86. Meier-kolthof JP, Göker M. VICTOR: Genome-based phylogeny and classification of prokaryotic viruses. Bioinformatics. 2017;33(21):3393–3404.
87. Page AJ, Cummins CA, Hunt M, et al. Roary: Rapid large-scale prokaryote pan genome analysis. Bioinformatics. 2015;31(22):3691–3693.
88. Contreras-Moreira B, Vinuesa P. GET_HOMOLOGUES, a versatile software package for scalable and robust microbial pangenome analysis. Appl Environ Microbiol. 2013; 79(24):7696–7701.
89. Vinuesa P, Ochoa-Sánchez LE, Contreras-Moreira B. GET_PHYLOMARKERS, a software package to select optimal orthologous clusters for phylogenomics and inferring pan-genome phylogenies, used for a critical genotaxonomic revision of the genus Stenotrophomonas. Front Microbiol. 2018;9.
90. Richter F, Morton SU, Kim SW, et al. Whole genome de novo variant identification with FreeBayes and neural network approaches. bioRxiv. 2020;2020.03.24.994160. DOI: 10.1101/2020.03.24.994160
91. Koboldt DC, Larson DE, Wilson RK. Using VarScan 2 for germline variant calling and somatic mutation detection. Curr Protoc Bioinformatics. 2013;44:15.4.1–17.

Address correspondence to:
Andrew Millard, BSc, PhD
Department of Genetics and Genome Biology
University of Leicester
University Road
Leicester LE1 7RH
United Kingdom

E-mail: adm39@le.ac.uk