2023

VIBES: A Workflow for Annotating and Visualizing Viral Sequences Integrated into Bacterial Genomes

Conner J. Copeland

VIBES: A Workflow for Annotating and Visualizing Viral Sequences Integrated into Bacterial Genomes

By

Conner J. Copeland

Bachelor of Science, The University of Montana, Missoula, MT, 2020

Thesis

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Cellular, Molecular, and Microbial Biology

The University of Montana
Missoula, MT

Autumn 2023

Approved by:

Ashby Kinch Ph.D., Dean
Graduate School

Travis Wheeler Ph.D., Chair
Cellular, Molecular, and Microbial Biology
R. Ken Coit College of Pharmacy, University of Arizona, Tucson, AZ

Patrick Secor Ph.D.
Cellular, Molecular, and Microbial Biology

Brandon Cooper Ph.D.
Cellular, Molecular, and Microbial Biology

Mark Grimes Ph.D.
Cellular, Molecular, and Microbial Biology

Jason McDermott Ph.D.
Pacific Northwest National Labs, Richland, WA

Copeland, Conner J., M.S., October 2023 Cellular, Molecular, and Microbial Biology

VIBES: A Workflow for Annotating and Visualizing Viral Sequences Integrated into Bacterial Genomes

Chairperson: Travis Wheeler

Bacteriophages are viruses that infect bacteria. Many bacteriophages integrate their genomes into the bacterial chromosome and become prophages. Prophages may substantially burden or benefit host bacteria fitness, acting in some cases as parasites and in others as mutualists, and have been demonstrated to increase host virulence. The increasing ease of bacterial genome sequencing provides an opportunity to deeply explore prophage prevalence and insertion sites. Here we present VIBES, a workflow intended to automate prophage annotation in complete bacterial genome sequences. VIBES provides additional context to prophage annotations by annotating bacterial genes and viral proteins in user-provided bacterial and viral genomes. The VIBES pipeline is implemented as a Nextflow-driven workflow, providing a simple, unified interface for execution on local, cluster, and cloud computing environments. For each step of the pipeline, a container including all necessary software dependencies is provided. VIBES produces results in simple tab separated format and generates intuitive and interactive visualizations for data exploration. Despite VIBES' primary emphasis on prophage annotation, its generic alignment-based design allows it to be deployed as a general-purpose sequence similarity search manager. We demonstrate the utility of the VIBES prophage annotation workflow by searching for 178 Pf phage genomes across 1,072 *Pseudomonas* spp. genomes.

ACKNOWLEDGMENTS

I extend my deepest gratitude to Travis Wheeler, whose kindness, insightful advice, and passion for science saw me through a graduate student career that spanned a pandemic, multiple crises in my personal life, and mental health revelations – I couldn't have made it without your unwavering support. Likewise, I'm incredibly grateful to Jack Roddy for his excellent SODA visuals and choice of project name, VIBES, as well as to George Lesica for his early guidance in Nextflow scripting. I'd like to thank Pat Secor and Amelia Schmidt for their roles in the early conceptualization of VIBES and providing interesting and exciting test cases and feedback throughout the workflow's development. I'd also like to thank Elizabeth Burgener, Julie Portois, and Paul Bollyky for sharing their dataset and analysis for VIBES testing. Many thanks to the members of the Wheeler Lab for sharing their excellent ideas, software expertise, and memorable adventures during conference trips, with particular thanks to Tim Anderson for his friendship, code reviews, and thesis text reviews. Finally, I'd like to thank my friends and family for all their love and support throughout my academic adventures.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND KEY TERMS

- API: Application Programming Interface. A defined interface allowing a piece of software to be acccessed by other software.
- *Att* site: Attachment site. *AttP* is on the phage genome and *attB* is on the bacterial genome.
- E-value: A statistical measure of the quality of an alignment. For an alignment with score S produced when searching database D, the E-value represents the number of alignments with score at least S that are expected to result if D consists of only non-homologous (random) sequence.
- FASTA: A common sequence file format, in which header lines are denoted by '>' and all other lines contain sequence. A file containing multiple sequences in FASTA format is called a multi-FASTA file.
- H-NS: Histone-like nucleoid-structuring protein.
- HMM: Hidden Markov Model. A specific kind of statistical model made up of states and transition probabilities between states. An HMM can be used to generate sequential data, or to classify observed sequential data.
- HPC: High Performance Computing, a compute cluster.
- HTML: HyperText Markup Language. Standard language used to generate webpages.
- kb: Kilobase, or 1,000 nucleotides.
- mb: Megabase, or 1,000,000 nucleotides.
- *OriC*: Origin of chromosomal replication.

- *Pa*: Pseudomonas aeruginosa.
- pHMM: Profile HMM. A specialized HMM used to model a biological sequence family, capturing position-specific character probabilities and indel rates.
- Phage: Bacteriophage.
- Prophage: The genetic material of a phage, incorporated into the genome of a bacterium; able to produce phages if activated.
- Query: A well-characterized sequence or pHMM used as a reference to search for in a sequence similarity search.
- RAM: Random access memory. Used by computers to store values associated with programs currently being run.
- ssDNA: Single stranded DNA.
- Target: An unannotated sequence. The subject of a sequence similarity search.
- TSV: Tab-Separated Value file format. Each value in a row is separated by a tab character.
- URL: Uniform Resource Locator. Address of a resource on the Internet.
- VIBES: Viral Integrations in Bacterial genomES. A Nextflow-based workflow manager designed for annotation of prophage within bacterial genomes. The subject of this thesis.

CHAPTER 1 INTRODUCTION

1.1 Genomes, Genes, and Proteins

A genome is a set of genetic material containing instructions for how an organism will grow, survive, and reproduce in the context of its environment. In cells, genomes take the form of long chains (chromosomes) of two interlocking, antiparallel strands of DNA (double-stranded DNA) made up of a combination of 4 nucleotides represented by the letters A, T, C, and G. Genomes encode functional molecules as a long series of genes, some of which are functional RNAs, such as tRNAs, while others encode proteins. The creation of a functional molecule begins when its gene is transcribed from coding DNA to RNA; if the gene encodes a protein, the RNA is further translated to amino acids. Access to coding DNA is controlled by regulatory DNA sequences, which respond to the environment in and around cells, allowing organisms to activate different genes under different environmental conditions. Changing, inserting, or deleting nucleotide sequences in a genome (mutations) can therefore significantly impact the behavior of organisms by either modifying regulatory sequences that control when and in what quantity specific molecules are produced or by changing, adding, or removing gene products. Put simply, modifying an organism's genome has the potential to substantially change how it reacts to its environment.

1.1.1 Bacterial Genomes

In approximately 90% of bacterial species, most or all of the cell's genome is contained by a single chromosome [1]. Generally, bacterial genomes take the form of a circular chromosome that is sometimes accompanied by much smaller plasmids, though there are some cases of linear bacterial chromosomes [2, 3] or cases in which secondary replicons account for a substantial portion

of the genome [4]. Bacterial genomes range in size from as small as ∼133kb [5] to as large as ∼14.8mb [6], with a mean length of ∼3.65mb [1]. Unlike eukaryotic genomes, bacterial genomes exhibit a deletional bias that generally maintains genomes primarily made up of protein-coding regions [7]. Bacterial protein-coding regions are often organized into operons, or clusters of co-regulated genes with related functions transcribed as a single mRNA [8].

Circular genome replication is initiated within a single *OriC* region and proceeds bidirectionally to a site called *dif*, which is equidistant from *OriC*. Two large complexes of enzymes (composed of DNA polymerases, primases, nucleases, DNA ligases, and various accessory proteins that coordinate other enzymes and attach the complex to template strands) move down both sides of the genome, using both strands of DNA as templates simultaneously. This forms two replication forks, which are preceded by DNA helicases and topoisomerases. DNA polymerases only add nucleotides to the 3' end of a DNA molecule, complicating replication of the antiparallel strand. Called lagging-strand synthesis, this is addressed by primases, which form RNA primers that allow DNA polymerase III (which can only add nucleotides to an existing chain of nucleotides) to synthesize relatively short sections of complementary DNA known as Okazaki fragments. DNA polymerase I then removes RNA primers from the lagging strand, replacing them with DNA equivalents. This leaves a break in the DNA backbone of phosphodiester bonds, which is repaired by DNA ligase. In *Escherichia coli*, replication forks proceed until they encounter *ter* sequences, where Tus proteins bound to the *ter* sites terminate replication.

Sometimes, the two daughter DNA molecules join into a chromosome dimer, which is resolved by the Xer system during the segregation of the daughter chromosomes into daughter cells. A DNA translocase, FtsK, positions the dimer such that the two copies of the *dif* site are located in the division septum, where the Xer system detects two copies of *dif* and promote recombination at the site, resolving the dimer (but only following an interaction with FtsK). Sometimes, daughter chromosomes become interlinked in a process called catenation as a result of the torsional stresses exerted by replication. Decatenation is mediated by type II topoisomerases, which break one chromosome to disentangle the daughter molecules. For a more detailed summary of bacterial

genome replication, see [9].

1.2 Bacteriophage and Host Interactions

1.2.1 Bacteriophage

Bacteriophages (phages), viruses that infect bacteria, are as ubiquitous as their hosts. They are found everywhere that we find populations of bacteria, from forest soils and the oceans to hydrothermal springs and the human gut. Phages pose a significant threat to bacteria: in marine ecosystems, up to one third of bacteria are killed by phages every day [10]. The strong pressure exerted by the threat of phage infection has led bacteria to evolve a diverse array of active antiphage defense systems. Defense system genes are gained and lost at rates higher than any other class of gene [11] and are often carried by mobile genetic elements (notably including prophages [12, 13]), enabling rapid dissemination of defenses via horizontal gene transfer [14]. Though antiphage systems employ diverse mechanisms, defenses generally hinge on two elements: a sensor that detects phage and an effector that degrades invading phage nucleic acids, disrupts gene expression long enough for phage nucleic acids to be degraded, or destroys the infected host in a mechanism called abortive infection [14, 15]. Recent research indicates that nucleic acid degrading defenses are widespread, accounting for ∼61% of detected antiviral systems [16]. These include restriction-modification systems [17] and CRISPR-Cas systems [18]. Other relatively widespread defense mechanisms detect conserved phage proteins and trigger abortive infection [16], possibly because synthesis of phage proteins indicates an infection so advanced that recovery is likely impossible [14]. Conserved phage protein sensing systems like Avs [19] and Stk2 [20] primarily target proteins essential for replication while others similar to DSR [21] and CapRel [22] detect structural components of virions. Another class of defenses is activated by the disruption of bacterial cell machinery. For example, the PrrC, ToxIN, and AvcD systems are all triggered when phage replication interrupts the inhibition of their respective genes, setting off immune responses or abortive infection [23, 24, 25].

Phages can be purely parasitic (lytic) and replicate at the expense of their bacterial hosts. However, in addition to lytic replication, temperate phages can alternatively undergo lysogenic

replication in which the phage genome typically integrates into the host chromosome as a prophage that is replicated each time the host cell divides (vertical transmission). Integration is sometimes achieved with an integrase, a site-specific recombinase that targets an attachment site (*att* site) on the phage genome (*attP*) and an identical site on the bacterial genome (*attB*). The phage genome and bacterial chromosome are recombined such that *attP* and *attB* flank the prophage sequence, enabling the prophage to eventually excise itself from its host genome and begin lytic replication [26] (Fig 1.1). However, other forms of prophage integration have been identified in which prophages resemble plasmids and remain in the cytoplasm [27] or integrate into the host chromosome randomly [28]. Temperate phages are common: studies estimate that 50% [29] or up to 75% [30] of sequenced bacterial genomes contain at least one prophage.

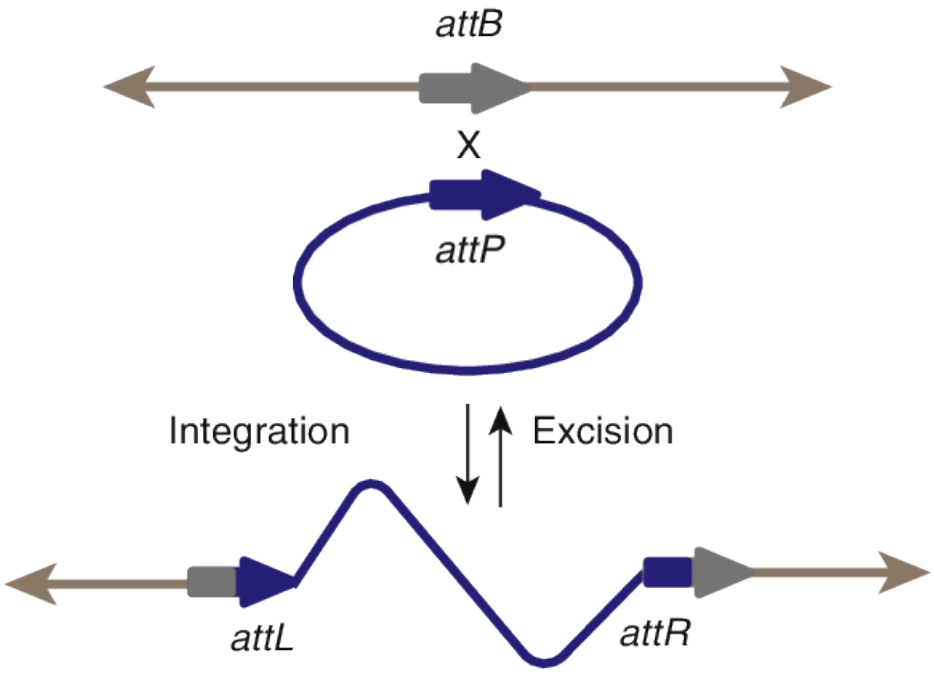


Figure 1.1: Diagram of prophage integration reproduced from Fogg et al 2014 [31].

1.2.2 Phage-Host Interactions

Phages have evolved to become adept manipulators of bacterial cellular processes, allowing them to evade host defenses or improve conditions for viral replication. For instance, viral homologues of *psbA*, a bacterial gene associated with photosystem II, are encoded as auxiliary metabolic genes in the genomes of 88% of phages that infect cyanobacteria [32], likely to minimize disruption of host photosynthesis during infection and provide better conditions for progeny virion assembly [32, 33].

Prophages benefit from thriving hosts via vertical transmission, which can incentivize the development of mutualistic phage-host relationships. A typical form of host-phage mutualism is lysogenic conversion, a phenomenon in which prophages encode factors that benefit the fitness of their hosts [34]. For example, some phages carry genes that promote resistance to infection from competing viruses [35] or encode antiphage defense systems [12, 13], while other phages encode virulence factors, aiding host pathogenicity [36].

Prophages sometimes modify host gene expression by integrating into or near to regulatory sequences. In some cases, prophages that disrupt regulatory regions in host genomes have evolved to excise themselves in response to the same cues that activate transcription of the disrupted gene, restoring functionality to disrupted genes [37].

1.2.3 Pf Phage

Many bacterial species are lysogenized by filamentous phages in the Inoviridae family [38]. The Gram-negative opportunistic pathogen *Pseudomonas aeruginosa* (*Pa*) is frequently lysogenized by an inovirus called Pf [39, 40, 41]. Pf virions are made up of an elongated, circular single-stranded DNA genome bound to several thousand copies of a major coat protein (CoaB) and are capped at both ends with a minor coat protein (CoaA) [42, 43] (Fig 1.2).

The Pf phage genome can be divided into two general regions: a conserved core genome that contains genes necessary for essential functions [39] and poorly characterized accessory genes that flank the core genome (Fig 1.3). Though not all core Pf genes are characterized, those that are encode the major (*coaB*) and minor (*coaA*) coat proteins, an excisionase (*xisF*), an integrase

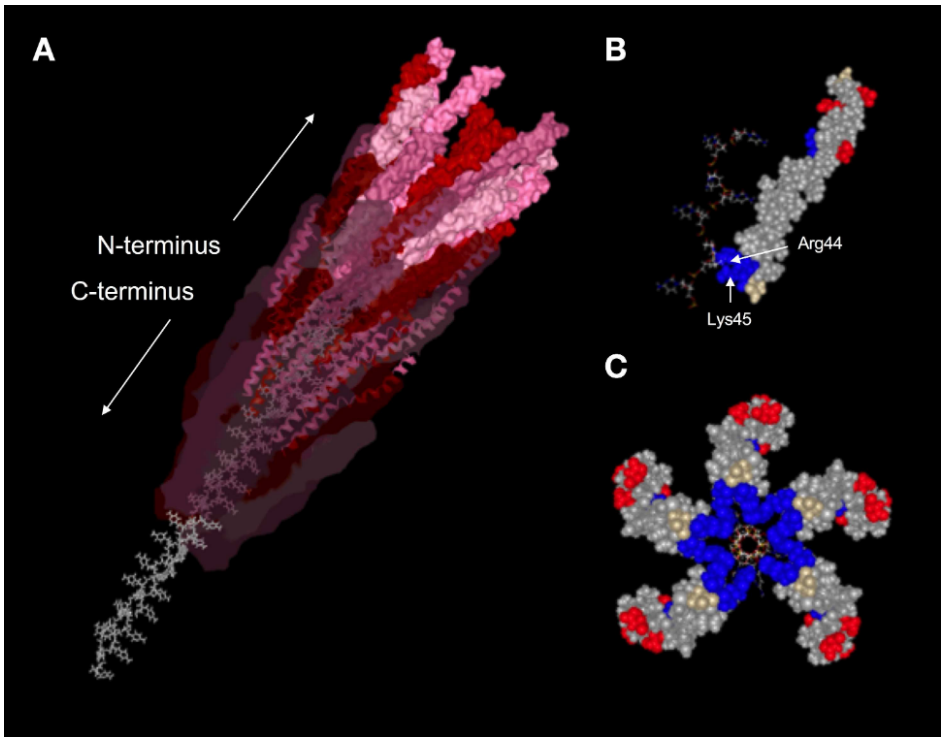


Figure 1.2: Pf1 virion structure reproduced from Secor et al 2020 [39]. **(A)** Ribbon diagram with Van der Waals surface representation of a portion of the assembled Pf1 virion (PDBID: 1PFI) showing the helical arrangement of CoaB subunits around the ssDNA viral genome. **(B)** Space-filling model of a single CoaB subunit bound to a stretch of cytosines. Arg44 and Lys45 are situated on either side of the DNA backbone and act to stabilize it through electrostatic interactions. Postively charged residues are in blue, neutral residues are in gray, and negatively-charged residues are in red. **(C)** Cross-sectional view of five CoaB subunits situated around the packaged viral genome. Amino acids are colored by charge as in **(B)**.

(*intF*), a c repressor (*pf4r*), an ssDNA binding protein (*p5*, *PA0720*), a replication initiation protein (*PA0727*) [39], and a protein that acts both as a host quorum sensing inhibitor and a superinfection exclusion factor (*pfsE*) [44].

Pf phage infection is initiated when the minor coat protein, CoaA, binds to the end of a type IV pili in a "tip-to-tip fashion" similar to other known inoviruses [42]. Type IV pili mediate twitch motility by extending and retracting from the cell [45]. When a pilus with an attached phage is retracted into the periplasm, CoaA interacts with TolA, a secondary receptor protein that is critical to the function of the Tol/Pal system [46] and thus highly conserved, ensuring its availability for infection [42]. Little is known of the exact mechanism that allows the phage genome to traverse

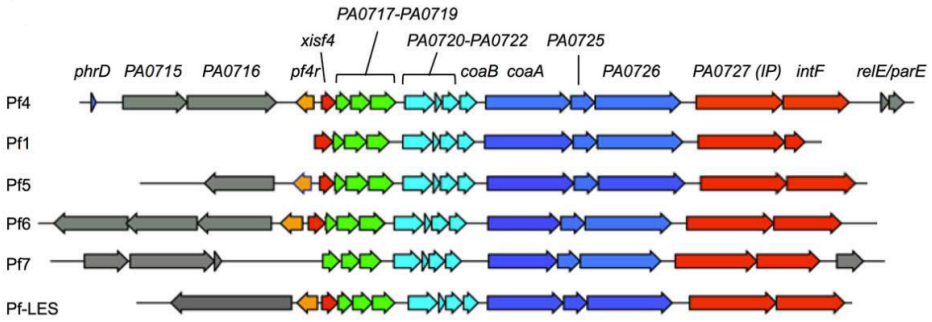


Figure 1.3: Comparison of Pf prophage sequences reproduced from Secor et al 2020 [39]. Core genome operons are color-coded relative to Pf4. Gray indicates accessory operons.

the inner cell membrane, except that the CoaB major coat proteins are shed into (and retained by) the inner membrane while the phage genome traverses the membrane and is deposited in the cytoplasm [42]. Once in the cytoplasm, the phage genome is converted into a replicative form and enters into either replication or lysogeny, depending on conditions in the host cell [39]. When replicating, PAO727 recruits host enzymes DNA polymerase III and UvrD and synthesizes copies of the replicative form and the ssDNA Pf genome via rolling circle replication [47]. Copies of the Pf genome are stabilized with a single stranded binding protein, PAO720 [48]. Meanwhile, copies of CoaB and CoaA are produced, and CoaB is inserted into the inner membrane by host enzymes Sec/YidC [49]. Like all inoviruses, progeny Pf virions are assembled at the cell envelope, where they can be secreted through the envelope without lysing the host [50] (Fig 1.4).

When conditions favor lysogeny, IntF integrates the genome into the host chromosome as a prophage. Pf prophages maintain lysogeny by suppressing transcription of *XisF*, their excisionase gene, through expression of a c repressor, Pf4r. *Pa* H-NS family proteins MvaT and MvaU also coordinate to repress *XisF* [51]. In response to oxidative stress [52], nutrient limitation [53], or other factors [54], *XisF* is desuppressed. XisF represses transcription of Pf4r, positively regulates the operon containing *IntF* and *PA0727*, and excises the prophage from the host chromosome [51]. PA0727 then recruits host enzymes to the replicate form of the prophage genome, initiating replication [47].

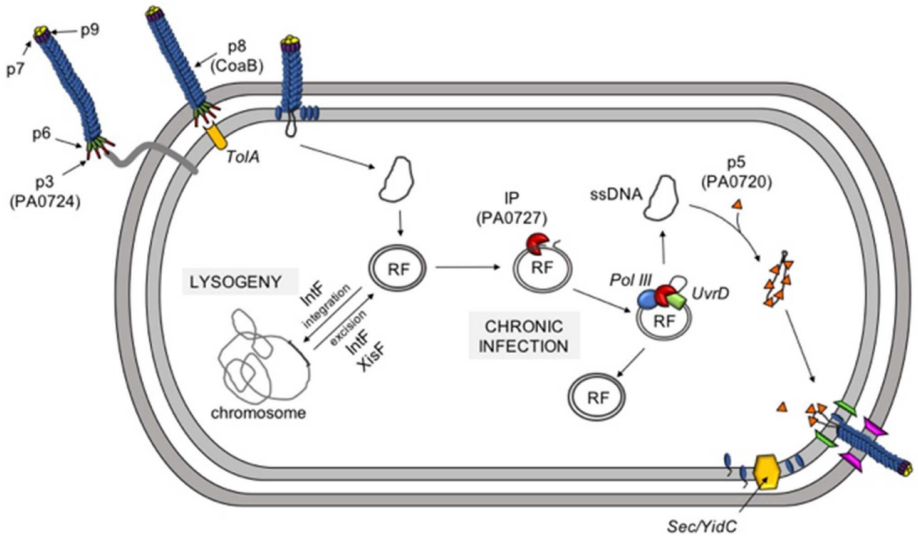


Figure 1.4: Diagram of Pf lifecycle reproduced from Secor et al 2020 [39]. Infection begins when CoaA (p3, PAO724) binds to a type IV pilus and the virion is retracted into the periplasm, where CoaA interacts with TolA. CoaB is shed from the viral genome as it moves through the inner membrane. Once in the cytoplasm, the ssDNA genome is converted into its replicative form and either integrates as a prophage or initiates viral replication. During replication, the initiator protein PAO727 recruits DNA polymerase III and UvrD to create additional ssDNA genomes and replicative forms. ssDNA genomes are stabilized with p5 (PAO720) and targeted to the inner membrane, where copies of CoaB have been inserted by host enzymes Sec/YidC. P5 is displaced by CoaB as a progeny virion is assembled and secreted from the cell envelope.

Pf virion replication plays a role in *Pa* biofilm development by lysing cells in the center of a colony, releasing DNA that adds to biofilm structural integrity [55]. Pf virions themselves also act as structural components in *Pa* biofilms, protecting bacteria from desiccation and antibiotics [56, 57]. Indeed, the presence of Pf virions in the airways of cystic fibrosis patients is associated with antibiotic resistance [58]. Additionally, Pf virions are immunomodulatory and induce maladaptive antiviral immune responses that promote infection initiation [59] and interfere with wound healing by inhibiting keratinocyte migration [60].

1.3 Sequence Annotation

1.3.1 General introduction

The advent of faster and cheaper genome sequencing has led to an explosion of genomic information that has transformed the study of biology, motivating the development of computational techniques to categorize and label the nucleotide sequences that make up genomes (sequence annotation). Since some genomes, such as those of eukaryotic organisms, can be on the order of billions of nucleotides in length, it is a formidable challenge to develop sequence annotation methods that can keep pace with the ever-increasing rate at which genomes are sequenced. Generally, sequence annotation hinges on an inversion of the observation that two sequences evolved from a common ancestral sequence will be more similar to each other than will two randomly-selected sequences. Inverted, the assumption is: if two sequences are more similar to each other than is expected under random chance, then they are likely to be descendants of a common ancestral sequence, which implies (but does not guarantee) that the sequences share similar functions. Computational approaches compare sequences in a process called alignment (Fig 1.5), in which a statistical model determines whether well-characterized sequences (queries) are evolutionarily related to unannotated sequences (targets) by inferring likely substitutions, insertions, or deletions. Models compute an alignment score based on the collection of inferred mutations and a mutation probability model. High-scoring alignments are suggestive of shared evolutionary history, but even random, unrelated sequences may produce high-scoring alignments. To provide guidance to the user, annotation tools usually generate a statistic called the "E-value" which reports the expected number of alignments producing the reported score by random chance, given the size of query and target databases used in the search. Modern, sophisticated alignment models take into account information such as per-position mutation rates and conserved regions in sequence families, enhancing their ability to detect sequences that are likely related.

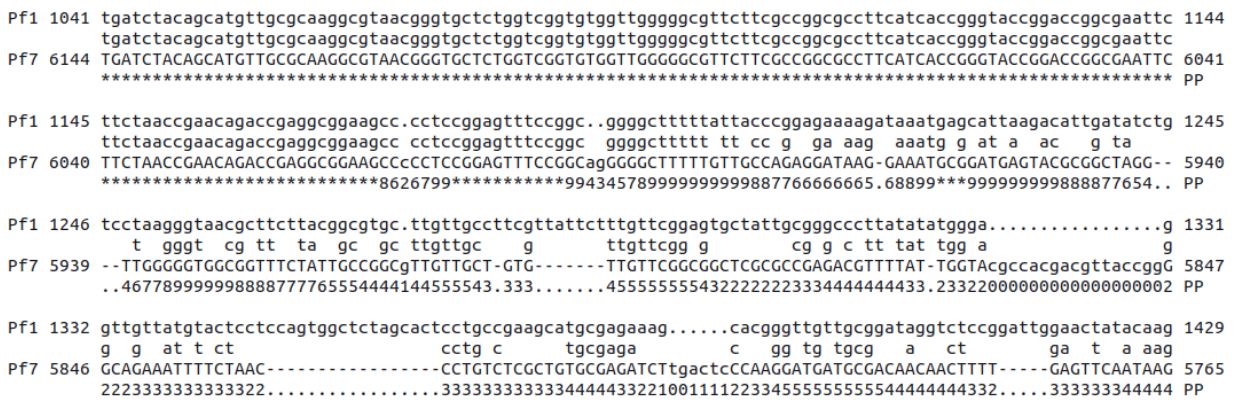


Figure 1.5: Excerpt from sequence alignment of two Pf phage genomes.

1.3.2 Viral Sequence Annotation

The important and varied roles of prophages in bacterial communities — parasites, mutualists, and sometimes pathogenicity aides — motivates the development of high-quality software methods that identify and classify prophage integrations into bacterial genomes. Generally, phage sequence annotation tools are designed to annotate either prophages in whole bacterial genome sequence or phage genome fragments in metagenomic datasets. While tools designed to annotate prophages integrated into bacterial genomes can be used to annotate metagenomic data (and vice versa), metagenomics-focused approaches generally only classify sequences as viral or nonviral, while prophage annotation techniques often include additional analyses such as *att* site annotation, bacterial gene annotation, viral gene annotation, and generation of visualizations that summarize output (Table 1.2). Most recent approaches focus on annotation in a metagenomic context, as metagenomic datasets have proven to be rich sources of previously unknown viral sequences [38, 61]. Viral annotation, particularly in metagenomic contexts, requires tools to strike a balance between sensitivity and speed. High sensitivity is necessary to overcome high mutation rates in viral proteins combined with an increased risk of sequencing error stemming from the low abundance of viral sequences in most metagenomic datasets. Meanwhile, reasonable labeling speed is required when annotating large datasets. As a result of these constraints, viral annotation software has

generally converged on a few techniques for identifying viral sequences: sequence similarity search against large databases of viral genomes [62, 63, 64], machine learning approaches based on statistical features such as k-mer frequencies, with a recent emphasis on neural networks [65, 66, 67], or some combination of both approaches [68, 69, 70]. Viral annotation software often includes further analysis of predicted prophage regions to help filter out false positives. Common approaches for further analysis of potential prophage regions include clustering regions of phage-like genes, analyzing sequences based on features thought to be indicative of phage genomes, and machine learning classification by models trained to distinguish bacterial and phage DNA sequences (Table 1.1).

1.3.3 Whole Bacterial Genome Prophage Annotation Tools

PHASTEST [71] is a recent update of PHASTER [62], a popular web-server based approach. PHASTEST identifies phage-like genes by searching submitted bacterial genomes for open reading frames with Prodigal, then searching translated amino acid sequences against a database of phages/prophages and phage genes with BLAST+. Phage-like genes are then clustered into prophage regions with the DBSCAN clustering algorithm and labeled intact, incomplete, or questionable. PHASTEST also annotates bacterial genomes with protein-coding genes using Diamond BLAST and reports the GC content percentage of phage regions, along with viral gene annotations. To run PHASTEST on a dedicated cluster, users can submit individual bacterial genomes on its website or submit batch jobs through PHASTEST's command line URL API interface. To reduce queue waiting times for individuals seeking to annotate large volumes of bacterial genomes, PHASTEST also offers a Docker [72] container that enables local execution of the tool.

DBSCAN-SWA [63] is a recent approach similar to PHASTEST. DBSCAN-SWA searches bacterial genomes against a database of 10,463 complete prophage and 684,292 phage proteins with Diamond BLASTP to identify phage-like genes, which are then clustered into prophage regions by a version of DBSCAN coupled with a sliding window algorithm. By default, clusters of at least 6 phage-like genes with a distance of no more than 3kb between each gene qualify as prophage regions. DBSCAN-SWA also provides phage gene annotations, tRNA annotation, *att* site annotation, and

Tool	Primary Domain	Phage Detection	Verification Analysis	Visual Output	Availability
DBSCAN-SWA	Whole genome	Similarity search	Clustering phage-like genes	Yes	Web server, command line software
DeepVirFinder	Metagenomic	ML classifier	N/A	No	Command line software
PHASTEST	Whole genome	Similarity search	Clustering phage-like genes	Yes	Web server
Prophage Hunter	Whole genome	Similarity search	ML classifier	Yes	Command line software
VIBRANT	Metagenomic	Similarity search	ML classifier, feature analysis	No	Web server, command line software
VirFinder	Metagenomic	ML classifier	N/A	No	Command line software
VirSorter	Metagenomic	Similarity search	Feature analysis	No	Command line software
VirSorter2	Metagenomic	Similarity search	ML classifier	No	Web server, command line software
Virtifier	Metagenomic	ML classifier	N/A	No	Command line software

Table 1.1: Overview of viral annotation software.

bacterial gene annotation through Prokka [73]. DBSCAN-SWA is available both through a web server that users can submit individual genomes to and as command line software that users can run themselves.

Prophage Hunter [70] is another whole genome prophage annotation tool that has two modes: a more sensitive mode that uses sequence similarity search and an alternative, similarity-search-free mode. In similarity search mode, Prophage Hunter uses BLASTX to search genomes for matches to libraries of phage genes that the authors found particularly indicative of prophage integrations.

Tool	*Att* site annotation	Bacterial gene annotation	Viral gene annotation	Output visuals
DBSCAN-SWA	✓	✓	✓	✓
PHASTEST	✓	✓	✓	✓
Prophage Hunter	✓	-	-	✓

Table 1.2: Features of prophage annotation software.

Regions with hits are further searched against Pfam [74] with InterProScan [75] for hits to domains that occur in the phage gene libraries. In the alternative mode, the bacterial genome is split into 10kb windows that are assigned a score based on 24 sequence summary statistics associated with prophages. 20kb windows upstream and downstream of regions with sufficient matches to phage genes or a sufficiently high feature score are searched for direct repeats indicative of *att* sites with BLASTN and then classified as host or prophage sequences by a logistic regression model trained on the aforementioned 24 sequence features. Prophage Hunter's output include prophage regions labeled as active, ambiguous, or inactive, phage protein-coding gene annotations, and *att* site annotations. The Prophage Hunter publication links to a web server, but the website seems to have been removed. However, the Prophage Hunter code is available via a GitHub repository.

All of the above methods generate visual summaries that increase the legibility of their output. Both DBSCAN-SWA and Prophage Hunter are available as stand-alone software that can be run by researchers seeking to conduct large-scale analysis in a cluster or cloud computing environment, but require installation of prerequisite software and manual creation of job submission scripts to run in HPC environments. This significantly increases the minimum computational skillset necessary to conduct large-scale analysis of prophage integrated into complete bacterial genome sequences. PHASTEST somewhat alleviates this by offering a containerized version of their approach, reducing the need to install prerequisites. However, managing widescale execution of containerized PHASTEST instances in a HPC environment through job submission scripts or other customized approaches is still left to end users, leaving intact a significant minimum computational skillset requirement to deploy the tool at scale.

1.4 VIBES

Here, we introduce and describe VIBES, an automated command line workflow for annotation of bacterial genomes that emphasizes identification of prophage integrations. VIBES supplements standard bacterial gene labeling with in-depth analysis of prophage integrations, producing machine- and human-readable text output files coupled with interactive HTML visualizations that facilitate further analysis of output data. VIBES is designed to:

- annotate prophages with high sensitivity;
- annotate bacterial genes on input genomes, using Prokka [73];
- annotate viral genes within input viral genomes, using the PHROG database [76];
- accept a potentially large number of bacterial genomes and candidate phage genomes as input;
- substantially reduce prerequisite installation and automatically manage distribution of workload to cluster/cloud/local resources; and
- create interactive HTML visuals that display the above, as well as display which regions that map to user input prophagess are most prevalent among all input bacterial genomes.

To the best of our knowledge, VIBES is the first prophage annotation software approach that includes output visualizations designed to facilitate investigation of patterns of prophage integration and shared prophage-host homology across entire target bacterial genome datasets. VIBES is also the first prophage annotation approach that is engineered to enable efficient, massively-parallelized prophage search and genome annotation by end users on a wide variety of hardware and compute environments. VIBES also has the capacity to serve as manager of massively-parallelized sequence similarity searches, even if the queries and targets of those searches are not prophages and bacteria, respectively.

CHAPTER 2 METHODS

2.1 Implementation

VIBES (**V**iral **I**ntegrations in **B**acterial genom**ES**) is an automated prophage search workflow that uses containerized components coordinated by the Nextflow workflow management software [77] to produce output tab-separated value (TSV) annotation tables accompanied by interactive HTML files that summarize matches to prophage sequences. To annotate prophage integrations, VIBES is provided with an input FASTA file containing all prophage sequences to seek and a collection of bacterial genomes in FASTA format to annotate with prophages; it performs search using the software *nhmmer* [78]. To annotate bacterial protein-coding genes, rRNA, and tRNA, VIBES uses Prokka [73]. To annotate query prophage protein-coding genes, VIBES uses the BATH protein-coding DNA annotation software [79] and the PHROG v4 prokaryotic viral protein database [76] by default. The user can optionally substitute their own viral protein database. Figure 2.1 provides a visual representation of the three independent annotation workflows, run in parallel and managed by VIBES, that produce bacterial gene annotations, viral sequence integrations, and viral gene annotations.

Before running VIBES, the user must install a software container system such as Docker [72] or Singularity/Apptainer [80] (usually the latter on HPC systems, where VIBES is likely to be utilized). These container systems enable the development and release of portable and reproducible software environments with fine-grained control over configuration and dependency conflicts while also retaining high performance. The user must also install the workflow management software Nextflow. Nextflow manages downloading and running containers, submits jobs to compute cluster job scheduling software (i.e. SLURM) or cloud computing architectures (i.e. AWS Batch),

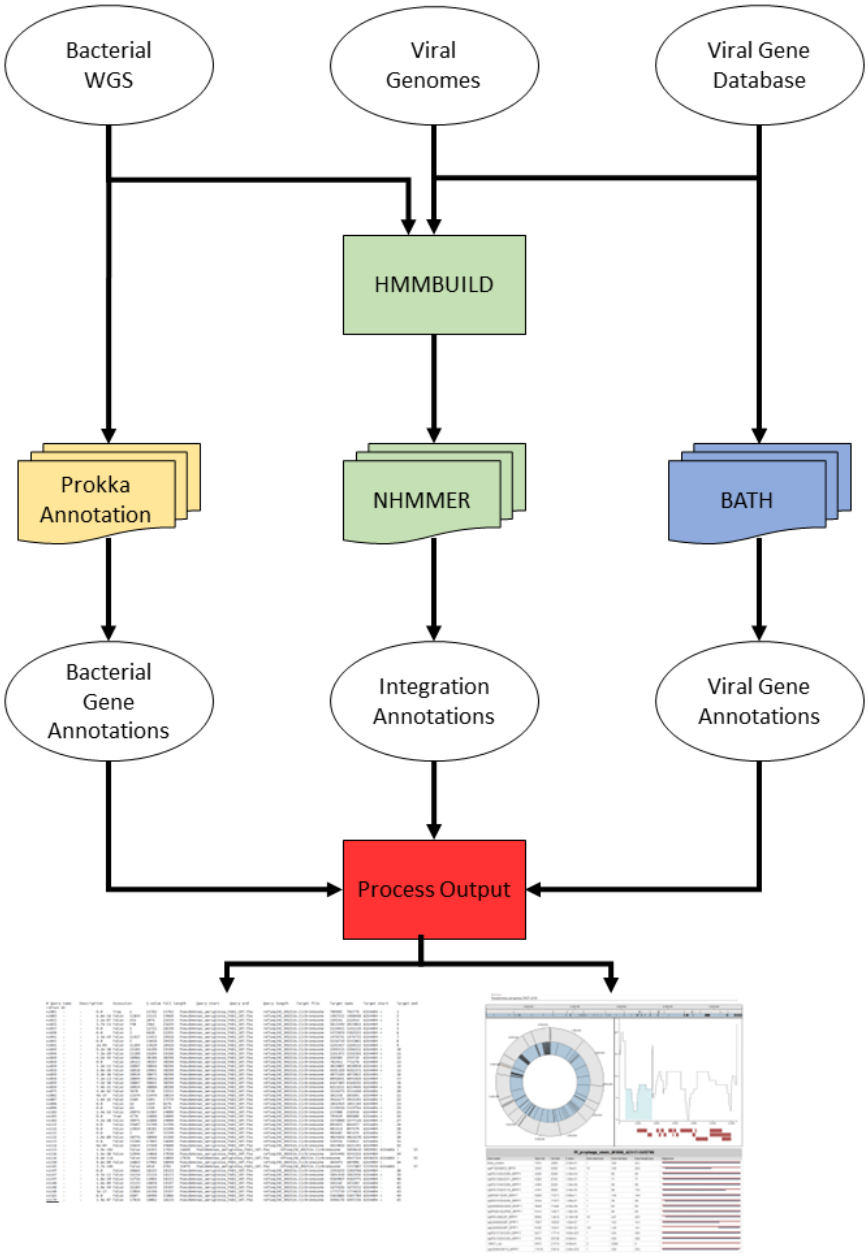


Figure 2.1: VIBES workflow schematic. Displays how input data moves through the VIBES annotation workflow. Bacterial gene annotation processes are yellow, prophage annotation processes are green, viral gene annotation processes are blue, and visualization processes are red. The annotation processes are independent of each other. Stacked icons indicate processes parallelized automatically by Nextflow.

caches and checkpoints in-progress jobs in case of a crash, provides interpretable workflow status updates, runs on a wide variety of operating systems and hardware configurations, and can run VIBES locally as needed. After a user configures the workflow to run on their system and launches it, Nextflow requires no further user interaction to identify task dependencies, automatically maximizing parallelism by running as many tasks with satisfied dependencies as available resources allow.

The VIBES release consists of a Nextflow workflow script, several helper scripts written in Python and Perl, JavaScript and HTML files that produce the visualizations, and a Docker image that manages the internal configuration and dependency map of multiple tools. VIBES software and workflow can be found at https://github.com/TravisWheelerLab/VIBES.

Tool	Purpose
BATH [79]	Annotation of protein-coding DNA on query prophage
Easel [81]	Determine length of input bacterial genome sequence
nhmmer [78]	DNA-to-DNA identification of prophages on bacterial genomes
Prokka [73]	Bacterial genome annotation

Table 2.1: Dependencies managed by VIBES Docker container.

Here, we describe each component of VIBES and its interactive visual output.

2.2 VIBES Components

2.2.1 Prophage Search Component

The primary component of the VIBES workflow is its prophage search. This component searches for user-provided query prophage sequences within bacterial genomes to identify prophage integrations. Identification of prophage within bacterial genomes is performed using a DNA sequence annotation tool, *nhmmer* [78], with default settings. Though it is slower than blastn [82], *nhmmer*'s improved sensitivity in the face of high sequence divergence and neutral mutation [78] is useful in the context of prophages, which can mutate at rates comparable to ssRNA viruses [83]

and may show substantial divergence from query sequences. In general, any matches to a query prophage that fail to meet an E-value threshold (1e-5 by default) are discarded.

Sequence annotation tools such as *nhmmer* frequently produce fragmented alignments when presented with sequences highly diverged from the query sequences, particularly when a match contains a large inserted or deleted element relative to its nearest query. As a result, single prophage integrations may be reported as several fragments that lie close to each other on a bacterial genome in *nhmmer* output. To address these potentially fragmented integrations, VIBES includes a post-processing step that examines every match detected on a single bacterial genome, looking for consecutive matches that satisfy all of the following criteria: potential fragments must match to the same query phage sequence (Fig 2.2A), occur in the same order on the bacterial genome as on the query (Fig 2.2B), be close to each other on the bacterial genome (Fig 2.2Ca), and overlap minimally on the query phage (Fig 2.2Cb). Gaps between two matches on the query prophage sequence are not penalized, as they may represent large deletions. Matches that meet these criteria are assigned a common integration ID that instructs the interactive visual component to display the fragments together rather than separately (see Interactive Visuals), effectively joining the fragments into a single integration.

2.2.2 Gene Annotation of Both Viral and Bacterial Genomes

VIBES provides supplementary context to identification and investigation of prophage integration sites by identifying protein-coding genes in both full bacterial genomes and query prophage sequences. Each bacterial genome is annotated using the annotation tool Prokka [73] via StaPH-B's Docker image [84], supporting gene annotation without requiring users to download or set up sequence databases. Like the prophage search component, each bacterial genome is annotated independently of other genomes, allowing Nextflow to fan out as many parallel Prokka annotation tasks as resources permit.

VIBES also produces gene annotations for the user-provided prophage sequences with its viral protein-coding gene annotation component. This component uses a translated search tool,

Figure 2.2: Depicts all conditions that must be met for consecutive matches to be joined (assigned the same integration ID and displayed as one integration in visual output). **A**: Join candidates must be assigned to the same query phage. **B**: Join candidates must occur in the same order on both the query phage and target bacterial genomes. **Ca**: Given a query phage genome of length n, a match that ends at position i on the bacterial genome, and a consecutive match that begins at position j, two matches are considered near enough for fragment joining if $|i - j| \leq n * k$, where k is a fragment gap threshold value set to 0.25 by default. **Cb**: Given a fragment whose match to the query viral genome ends at position s and a consecutive fragment whose match to the query viral genome begins at position t, the fragments are joined only when $|t - s| \leq \theta$, where θ is a constant set to 50 by default. Large gaps between matches on the query prophages are not penalized, as they may represent large deletions.

BATH [79], to search a viral protein database against prophage DNA sequences. Translated search tools like BATH do not penalize neutral mutations that change DNA sequences without modifying the encoded protein sequence, making them especially well-suited to annotating sequences with high mutation rates such as viral genomes. BATH's translated search is also robust to frameshift-inducing insertions or deletions, which can confound other translated search tools. By default, VIBES uses the PHROG v4 viral gene database [76] reformatted as a BATH-compatible HMM database, but users can substitute other amino acid sequence or HMM databases as desired (see

Section 4.1.1).

Although VIBES was developed with annotating prophage integrations in mind, it is primarily a framework for managing and parallelizing runs of *nhmmer*, Prokka, and BATH with some prophage-annotation-specific features (the default PHROG database is phage-specific and the VIBES-SODA visualization suite assumes query sequences are prophage). In particular, the Prophage Search Component simply searches for matches to a query database (prophages by design) in a set of target genomes (bacteria by design) and can easily be repurposed by providing the workflow with a non-phage query sequence file and a set of non-prokaryotic genomes. Likewise, the Prokka bacterial gene annotation and BATH translated amino annotation components can be used to orchestrate massively parallel protein-coding sequence annotation, even on datasets where prophage integrations are not of interest (see Section 4.1.1).

2.2.3 Interactive Visual Generation

To facilitate further analysis and improve human readability of results, VIBES produces dynamic annotation visualizations in HTML files that can be opened in a web browser. These visuals depict prophage annotations, bacterial gene annotations, and viral gene annotations. After all other workflow tasks are complete, VIBES generates a collection of HTML files, each of which contains a dynamic visualization built with the SODA sequence annotation visualization library [85], each of which contains interactive annotation visualizations for its associated genome. The HTML files may be opened locally in a web browser, or they may be hosted on a web server. The generated interactive visualizations are described in Section 3.1.3.

2.3 Pf Prophage Search

To demonstrate the utility of VIBES as a prophage identification tool, we searched 1072 *Pseudomonas* isolates from 130 species (Table 2.2) for integrations of 178 Pf phage variants. *Pseudomonas* spp. genomes were acquired from the Pseudomonas Genome Database (v21.1) [86]. Some records in the database were renamed to resolve characters that conflict with standard Bash com-

mands while three records contained no sequence information. Two of the three empty records were populated with data from GenBank while the third was determined to be redundant and deleted. Phage sequence coordinates for 179 partial or complete Pf prophages were obtained from a study examining Pf prophage lineages [40]. 126 *Pa* genomes were downloaded using accession IDs provided in the study, from which 179 Pf prophages were extracted and assigned identifiers. One phage sequence, labeled vs015, contained a substantial insertion that extended the length of the sequence to over 70kb. Such a long query sequence requires a prohibitive amount of memory to search for, so vs015 was removed from our query database, leaving a total of 178 Pf prophage query sequences.

Analysis was conducted on the University of Arizona's Puma HPC cluster on nodes that each contain 94 AMD EPYC 7642 cores and 512 GB of RAM.

Species	Count
Pseudomonas aeruginosa	494
Pseudomonas chlororaphis	55
Pseudomonas putida	49
Pseudomonas syringae	48
Pseudomonas fluorescens	39
Pseudomonas stutzeri	21
Pseudomonas protegens	20
Pseudomonas monteilii	10
Pseudomonas amygdali	9
Pseudmonas brassicacearum	8

Table 2.2: 10 most prevalent *Pseudomonas* species with complete genome sequences available in the Pseudomonas Genome Database. 120 additional species occurred at a lower frequency. Additionally, several genomes were not assigned to a species.

CHAPTER 3 RESULTS

3.1 Overview of VIBES Output Features

3.1.1 Detected Prophages in Bacterial Genomes

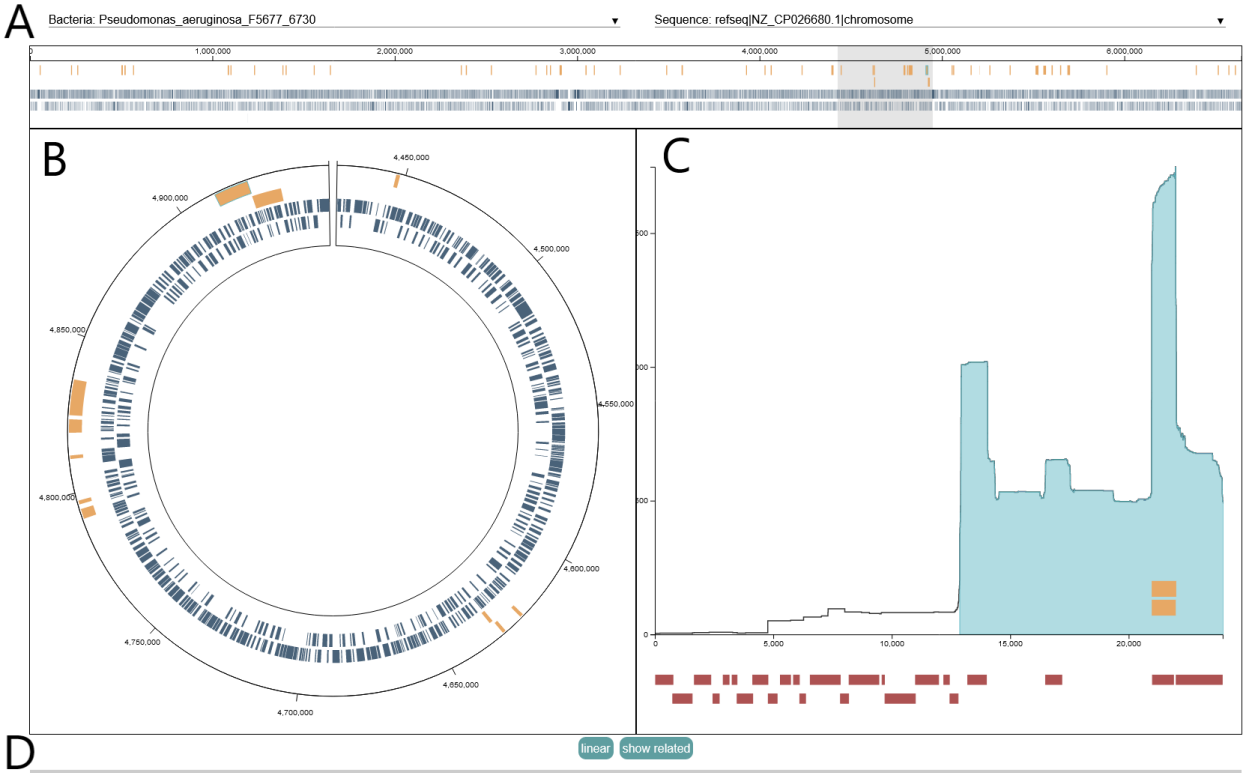
For each input bacterial genome, VIBES produces a tab-separated value (TSV) file describing each detected potential prophage sequence in the genome. The TSV fields include matching phage name, match E-value, score, match start and end positions on both query (phage) and target (bacterial) sequences, match strand, a match integration ID (see Prophage Search Component under Methods and Materials), and a full-length field populated with True (full length) or False (partial). By default, a match is called full length if it is at least 70% the length of the best-matching prophage sequence, though this parameter can be modified by the user.

3.1.2 Bacterial and Viral Gene Annotation

Annotation of genes within bacterial genomes are generated by Prokka with its default annotation databases and settings. For each bacterial genome, full Prokka output is saved and optionally compressed into a zipped tar archive. Annotations of genes within prophage genomes are output in their own TSV format files with fields identical to those produced for prophage annotations except the match ID field, which is excluded for phage gene annotations.

3.1.3 Interactive HTML Visual Output (Fig 3.1)

After each workflow process has completed, VIBES produces the SODA-based HTML visualization files. The interactive representations of the workflow's output allows users to investigate annotations in a bacterial genome and potential prophages with the following components:



vs103

Gene name	Start (nt)	End (nt)	E-value	Gene start (aa)	Gene end (aa)	Gene length (aa)	Alignment
phrog_1864	3260	6729	4.40e-37	1	72	73	
phrog_4515	3472	7606	5.40e-147	1	221	224	
phrog_757	4131	8906	3.30e-74	10	246	247	
phrog_10578	4785	9948	8.00e-73	1	126	126	
phrog_1144	5297	11025	2.70e-92	1	144	144	
phrog_1147	5853	11954	1.40e-47	1	83	84	
phrog_2145	6117	12479	8.00e-50	1	83	83	
phrog_815	6559	14386	2.80e-162	5	420	420	
phrog_306	7835	16014	4.40e-46	1	115	115	
phrog_1041	8198	17655	1.20e-191	1	419	420	
phrog_1159	9714	20711	7.10e-250	2	428	428	
phrog_1	11003	22982	3.70e-68	115	452	611	
phrog_2268	12196	24628	7.00e-35	2	83	84	
phrog_12577	12457	25258	9.60e-67	1	115	115	
phrog_8061	22006	45958	9.70e-203	15	659	659	
phrog_8999	31	796	5.40e-93	5	263	270	
phrog_13337	755	2331	1.50e-128	1	274	275	
phrog_7381	1669	4033	7.00e-131	1	231	231	
phrog_5560	2452	5170	1.50e-42	1	89	90	
phrog_147	2884	6019	5.40e-24	3	91	100	
phrog_11683	9591	19283	8.40e-7	1	38	61	
phrog_789	13206	27203	1.90e-16	341	605	637	
phrog_9266	16493	33672	4.10e-34	16	286	316	
phrog_3736	21004	42898	6.20e-50	2	294	303	

Figure 3.1: Example VIBES interactive annotation visualization page, displaying a bacterial replicon with gene and prophage annotations, where the selected integration falls on the closest-matching viral genome, and viral gene annotations. **3A**: The full interactive visualization page. **3B**: The bacterial replicon plot includes 2 modes to represent a selected bacterial replicon: linear and circular, both marked with integration and gene annotations. **3C**: The position occurrence plot displays information about a selected integration, related integrations, and prophage gene annotations. **3D**: The query phage gene annotation table contains detailed information about gene annotations on the closest matching user provided phage genome.

3.1.3.1 Bacterial Replicon Plots (Fig 3.1B)

The visualizations include both circular and linear representations of the selected replicon (users can switch the view between the chromosome and other replicons). Both representations of the replicon are marked with detected viral integrations (yellow) and bacterial genes (blue) to assist in analysis of integrations and phage landing sites. Hovering over a blue bacterial gene marker displays the name of the gene, while hovering over a yellow phage integration marker displays the name of the prophage. Users can select a viral integration to inspect it more closely (see Position Occurrence Plot below). Users can zoom in on the replicon and click and drag to pan across the genome, making gene or integration annotations larger and easier to interact with; simultaneously, the currently visible portion of the genome is highlighted in gray across the replicon along top of the page. The circular genome can be changed to a linear representation, and vice versa, by clicking the linear button below the interactive replicon.

3.1.3.2 Position Occurrence Plot (Fig 3.1C)

To assist users in investigating patterns of phage integration, a position-specific occurrence plot is displayed for a selected integration. The selected integration may be changed by clicking on a corresponding glyph in the genome annotation chart, or via the drop-down input at the top of the plot. The x-axis of the plot corresponds to each position (nucleotide) in a query phage sequence while the y-axis displays a count at each position summing every occurrence of that position in every integration in the dataset, emphasizing regions of a phage sequence that most often integrate into host genomes. The blue shaded region along the x-axis displays the extent of the currently selected integration on the query sequence it matched to. Yellow bars over the x-axis show where any other integrations matching to the same query phage on the selected bacterial genome matched to the query, indicating regions of the phage integrated repeatedly into the same genome. The yellow bars indicating where other integrations of the same phage fall on the viral genome can be hidden by clicking the hide related button located under the position occurrence plot. Under the x-axis, red bars display where viral gene annotations fall on the phage genome. Hovering over a viral gene

annotation bar shows the name of the gene, while clicking on it highlights its row on the query phage gene annotation table.

3.1.3.3 Query Phage Gene Annotation Table (Fig 3.1D)

At the bottom of the visualization is a table of viral protein-coding gene annotations on query phage sequence most closely matching the currently selected integration. The phage gene annotation table contains a row for each annotated gene displaying its name, start and end positions on the query phage genome, annotation e-value, start and end positions relative to the reference gene amino acid sequence, and an alignment figure that visually depicts the extent of the match on the query phage sequence (blue line) compared to the reference amino acid sequence (red line).

3.2 Ground Truth Data Comparison

To assess the performance of the workflow on known prophages, I ran VIBES against two reference *Pa* genomes, PAO1 and UCBPP-PA14, which contain Pf4 [87] and Pf5 [88], respectively. Both genome sequences were obtained from the Pseudomonas Genome Database [86]. Pf4 and Pf5 prophage were extracted from their hosts and provided to VIBES as queries. To indicate the performace of VIBES relative to other prophage annotation tools, I also submitted the genome sequences of PAO1 and UCBPP-PA14 to DBSCAN-SWA and PHASTEST through their web servers. These tools use *de novo* annotation databases to annotate prophages via their web servers, and so are not directly comparable to VIBES (as VIBES was given exact query sequences to search for in each respective genome). Still, I think it is useful to see whether the outputs of the three approaches are similar or dissimilar to each other. I attempted to also submit both genomes to Prophage Hunter, but its web server is no longer available and issues encountered during local installation prevented its involvement. Results of the ground truth data comparison are summarized in Table 3.1.

VIBES was run on default settings and set to filter out results shorter than 1kb, as in the Broad Spectrum Analysis 3.3.1. VIBES ran its searches in parallel, completing both in 880 seconds. Both VIBES and PHASTEST generate bacterial genome annotations, output visualizations, and viral

Tool	PAO1 Runtime (s)	PA14 Runtime (s)	PAO1 Prophage	UCBPP-PA14 Prophage
DBSCAN-SWA	387	414	None	None
PHASTEST	536	552	1 Intact YMC11, 1 Intact Pf1	1 Incomplete YMC11, 1 Intact Pf1
VIBES	880 (both)	880 (both)	1 Full (Pf4)	1 Full (Pf5)

Table 3.1: Summary of ground truth annotation of PAO1 and UCBPP-PA14 with Pf4 and Pf5.

gene annotations. As expected, VIBES recovers a full-length Pf4 integration in PAO1 spanning from PA0714 - PA0729.1 [87] and a full-length Pf5 integration in UCBPP-PA14 spanning from PA14_48870 - PA14_49040 [88]. Notably, VIBES did not identify any other sequences in PAO1 or PA14 that matched to Pf4 or Pf5 and were long enough to pass the 1kb filter. PHASTEST recovered 1 Pf1 integration, which it ranked intact, in a 15.8kb window centered on the location of Pf4 in PAO1 and 1 Pf1 integration, ranked intact, in a 18.2kb window centered on Pf5 in UCBPP-PA14. DBSCAN-SWA did not detect prophages in either genome, but this could occur if its *de novo* annotation database does not contain queries homologous to Pf4 or Pf5.

3.3 Application To *Pseudomonas* spp. Datasets

To explore the utility of the VIBES workflow for identifying (possibly fragmented) phage integrations within bacterial isolates, I applied it to two different *Pseudomonas* spp. and Pf phage datasets.

3.3.1 Broad Spectrum *Pseudomonas* Analysis

This analysis was conducted on the dataset composed of 1,072 publicly available *Pseudomonas* spp. genomes obtained from the Pseudomonas Genome Database [86] and 178 Pf phage variants published in a study on Pf phage lineages [40]. Nextflow reported that the prophage detection component of the workflow consumed 13,526.3 CPU hours across 2,099 tasks in its prophage search

component, 398.8 CPU hours across 1,072 tasks in its bacterial gene annotation component, and 49.7 CPU hours across 539 tasks in its viral gene annotation component, totaling 13,974.8 CPU hours consumed across a total of 3,710 tasks (more details on resource usage can be found in Table 3.2).

Component	Total CPU Hours	Tasks Run	Most Expensive Process	CPUs Allocated	Mean RAM	Mean Runtime (minutes)
Prophage Search	13,526.3	2,099	*nhmmer*	2	36.8 GB	412.5
Bacterial Gene Annotation	398.8	1,072	Prokka	6	901.2 MB	3.5
Viral Protein Annotation	49.7	539	BATH	2	403.5 MB	8.6

Table 3.2: Resource usage. CPUs Allocated, Mean RAM, and Mean Runtime all display values for the most expensive process in each component, as the computational cost of other elements were negligible. GB stands for gigabyte and MB stands for megabyte.

VIBES reported 51,386 partial and 517 full-length Pf phage integrations. Of the 51,903 integrations identified, 1,398 were composite integrations formed by 2 or more fragments joined together. The vast majority of reported integrations were less than 1,500 nucleotides in length (Fig 3.2). Although the workflow was set to discard matches less than 1,000 nucleotides in length, the median and average lengths of identified integrations were 1,240 and 2,419 respectively.

3.3.2 Comparison to Special-Purpose Analysis of Pf Phage in *P. aeruginosa*

To evaluate the general-purpose VIBES workflow's value for prophage discovery, I compared its results to a custom-built tool designed to identify Pf prophages. The dataset used in the comparison is composed of 91 *P. aeruginosa* clinical isolates sequenced by Elizabeth Burgener in the Bollyky lab [89] and shared with us during development of VIBES. Julie Portois in the Bollyky lab developed a custom-built pipeline that identifies full-length Pf prophages in bacterial genomes using BLAST

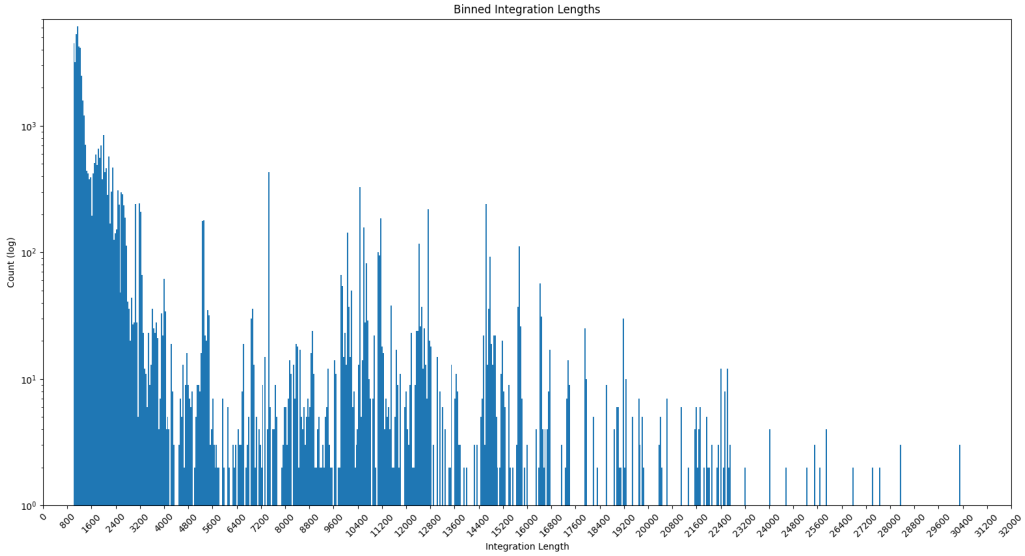


Figure 3.2: Integration Length Bin Plot. Counts of integrations binned by length, where joined integrations were summed together. The y-axis uses a log scale due to the large number of short integrations identified.

searches for conserved Pf phage genes [personal communication, manuscript in prep]. I used VIBES to search the bacterial genomes with 2 Pf phage genomes, Pf1 and Pf7. The Bollyky lab pipeline identified 41 complete Pf prophages while VIBES identified 24 full and 21 partial prophages. VIBES recovered all prohages identified by the Bollyky lab pipeline, though it called 21 of those prophages partial rather than full-length, likely a result of the prophage query database (only two phage genomes) failing to broadly represent Pf phage diversity. Of these 21 partial integrations, all but one covered at least 30% of their query phage genome, with each mapping to regions of query phage genomes containing at least 5 genes. VIBES also identified 1 full length and 3 partial integrations not called by the Bollyky lab pipeline, which discards partial integrations.

CHAPTER 4 DISCUSSION

4.1 Recommended Usage

VIBES is best-suited to conduct large-scale analysis of relatively well-characterized prophages in cluster or cloud computing environments where parallelization of search processes yields the largest gains, and where the impact of its convenience features (prerequisite management with Docker/Singularity and job orchestration and automated job management via Nextflow) are most pronounced. VIBES is also well-suited to large-scale analyses of homologous sequences shared between prophages and host genomes, given the high sensitivity of *nhmmer* search to distant homology [78] relative to other sequence similarity search software. As indicated by the comparison to the Bollyky lab pipeline, VIBES may be more effective than other tools at detecting prophage fragments, which are missing the majority of the phage genome due to either incomplete integration or post-integration mutational events [90].

VIBES may also be useful as a secondary step in *de novo* prophage annotation. Other tools, such as PHASTEST [71], are well-suited to identifying prophage integrations, but provide relatively coarse annotations (as seen in the Ground Truth analysis, Section 3.2, in which PHASTEST characterized Pf4 and Pf5 as integrations of Pf1 and identified regions containing the prophages, rather than exact prophage boundaries). Once integrations have been coarsely identified and users have a notion of what prophage sequences to supply as queries, VIBES is relatively well suited to determine exact prophage strains and more precise integration boundaries if prophage regions are provided as target sequences.

4.1.1 Alternative Use of the VIBES Workflow

To a large degree, VIBES is a workflow that coordinates sequence similarity search and bacterial genome annotation when supplied with target and query sequences. The only VIBES components that impose requirements apart from target sequences in FASTA format and queries sequences in a multi-FASTA file are Prokka [73] annotation (which requires bacterial sequence input) and visualization generation (which assumes that targets are bacterial genomes and queries are prophages). As a result, users are essentially free to use VIBES to conduct searches against nucleotide sequences as they see fit, so long as they specify query sequence type (DNA, RNA, or amino) and disable incompatible workflow components through the VIBES configuration file.

Users who wish to annotate one or more nucleotide sequences with other nucleotide sequences can do so by creating a multi-FASTA file of desired queries supplied to the phage_file argument in the VIBES configuration file. Desired targets should be in FASTA format and stored together in a folder specified by the genome_files argument. To conduct nucleotide-to-nucleotide search, the detect_integrations argument (which enables or disables *nhmmer* search) should be set to true and the seq_type arugment set to either dna or rna in lowercase. As an example of this kind of use, the VIBES workflow was used to annotate target sequences with potential primer queries during workflow development.

Users who want to annotate nucleotide sequences with proteins can do so by creating a multi-FASTA file of protein queries, supplied to the viral_protein_db argument, and a folder of FASTA format nucleotide queries, supplied to the genome_files argument. To conduct translated search, the annotate_phage_genes argument (which enables or disables BATH search) should be set to true. As an example of this kind of use, we used VIBES to search for PHROG v4 protein family members [76] in the genome of a eukaryotic organism, *Rana temporaria* (the common frog) [91].

4.1.2 Investigating VIBES Output

Following annotation steps, VIBES provides users with its interactive visual output to enable further investigation of potential prophage sequences. From the visual output, users can see the

extent of a match relative to its nearest query, annotated bacterial genes, annotated genes on the query viral genome, and other nearby matches (Fig. 3.1). Presence of a prophage on the target genome is signaled by either a relatively long match to a query phage sequence that spans several gene-coding regions or a cluster of matches to one query phage sequence. On the other hand, a short, isolated match to a query sequence that spans a single gene is ambiguous. If the gene spanned by the short match corresponds to a protein typically associated with viruses, such as an integrase or a gene known to be conserved in some viral families, it may be the case that the match is the only region of a prophage sequence that shares sufficient similarity with the query sequences submitted by the user. In this case, users could further analyze these potentially viral ambiguous sequences by extracting a ∼20-30kb window centered on the match with a tool such as seqkit subseq [92]. Extracted windows could be re-submitted to VIBES with only the viral gene annotation subworkflow enabled for annotation with PHROGs [76], or could be submitted to other annotation software with good support for *de novo* annotation such as PHASTEST [71].

Short, isolated matches may also signal a homologous region of bacterial sequence that is not of viral origin. Some phage genomes have been found to carry viral homologues of bacterial metabolic genes [33, 93]. The presence of such homologous genes in query prophage genomes may therefore lead to matches to bacterial gene sequences that are not necessarily acquired from prophage. In these cases, spikes on the Position Occurrence Plot in interactive visual output (Fig. 3.1C) over single genes with many more matches than other regions of the query genome may represent regions of homology to relatively common bacterial genes. Comparing these matches to annotated bacterial genes at the same location on the target genome may also help resolve whether the regions are of bacterial or viral origin.

4.2 Limitations and Future Directions

4.2.1 Integration Splitting

Currently, VIBES employs a relatively simple, heuristic approach to addressing integrations split into multiple matches by regions of low similarity to query prophages. While such split integrations

are apparent in interactive visual output due to the close proximity of matches belonging to the same integration, this may lead to simple programmatic analyses of VIBES output overcounting detected integrations. This is especially true in the worst case scenario of one integration being split up by matches to different query prophage sequences, which might occur if the target integration is the result of recombination between two query viruses [90, 34] or belongs to a family that is descended from a common ancestor of the two matching queries. In such a case, the heuristic approach to joining split integrations currently employed by VIBES will fail to join the constituent matches into one integration.

This limitation may be better addressed by the addition of a match clustering step, similar to the approach used by PHASTEST and DBSCAN-SWA [63] to narrow down potential prophage regions. Such an approach is discussed in depth in Section 4.2.2. Another potential solution is the adaptation of an annotation adjudication approach akin to AURORA [94] (previously named PolyA [95]), which would enable the VIBES to better decide between two queries that match one target region, handle recombination between query strains more effectively, and improve phage-host boundary detection. AURORA is designed to operate in the domain of transposable element annotation, so software engineering work to adapt it to prophage annotation is likely necessary for optimal performance.

4.2.2 Lack of Prophage Verification Analysis

A major limitation of the VIBES workflow is its omission of techniques that automatically score the likelihood of each hit to a query prophage sequence representing a complete or partial prophage sequence, rather than a homologous non-phage region of the host genome. As a result, VIBES output is noisy compared to other prophage annotation software. For example, VIBES reported tens of thousands of ∼1kb hits to query Pf phage sequences in the broad spectrum *Pseudomonas* analysis (Fig. 3.2), many of which are likely hits to proteins on *Pseudomonas spp.* genomes that are homologous to proteins carried on some Pf phage sequences in my query database. While VIBES provides some coarse-grained analysis of these sequences via the Position Occurrence Plot

(Fig. 3.1C) in interactive visual output and its complete/partial classification, determining exactly which short hits to query viruses are remnant prophages and which short hits are homologous sequences of bacterial origin is currently left to the user. While this may be desirable in some analyses, in most cases this poses a substantial analytical burden that other approaches seek to address.

An simple extension of the workflow that would enable further verification of phage sequences would be the addition of a Nextflow process employing a tool such as CheckV [96], which is designed to characterize the quality of metagenome-assembled viral genomes and predict boundaries between bacterial and viral sequences. This would require some re-tooling of the last stages of the pipeline, but would not require implementation of a new verification approach.

One potential extension that would address both resolving split integrations and verifying potential prophages would be the inclusion of a phage-like region clustering step, similar to the DBSCAN clustering approach employed by both PHASTEST and DBSCAN-SWA. Essentially, this allows users to set a minimum number of matches to phage genes and a minimum match density for a target genome region to be labeled as a prophage. This would filter out all short, isolated matches to query phage sequences in VIBES output, significantly reducing the number of matches currently reported by the workflow. Such an approach would also handle cases of single prophage integrations split into multiple matches by regions of low similarity, and would be more robust to handling cases where those matches point to different query sequences than the current match joining strategy employed by VIBES. However, this clustering technique may filter out matches to distantly related prophage regions that only share homology with query phages in a small number of highly conserved genes.

Another potential avenue for implementing prophage verification analysis draws inspiration from Virsorter [64] and Virsorter2 [68], two metagenomics-focused annotation tools that use HMMER similarity search software to initially identify regions with matches to phage protein sequence families. Both tools extend HMMER search with analysis of sequence features thought to signal phage regions, such as more heavily weighting matches to 'hallmark viral genes', density of vi-

ral gene annotations, gene size, and GC content. Virsorter2 extends this analysis by computing more sequence features and including random forest models trained on features of different viral groups and which provide a high-level taxonomic assignment of detected phage regions. This sort of prophage verification analysis would provide VIBES with a means of evaluating potentially viral sequences that does not strictly depend on detecting sizeable clusters of matches, which could maintain relatively high sensitivity to distantly homologous prophages while providing a means of filtering out ambiguous bacterial sequences. Extending VIBES with this sort of sequence feature analysis would require substantial modifications and additions to the workflow.

Another potential means of filtering viral and nonviral sequences would be training a neural network to distinguish between the two. Neural network approaches for classifying sequences as viral/nonviral are still relatively new, but tend to outperform other methods in sensitivity to distantly homologous viruses [97]. Such an addition to the VIBES workflow would require substantial work to engineer and train, including careful construction of training and test datasets that would allow rigorous evaluation of the model's performance and maximally mitigate the dataset composition bias discussed in Section 4.2.5. A potential means of achieving this is by downloading a large dataset of phage sequences, such as PhageScope [98] and clustering its constituent sequences based on nucleotide sequence similarity with a tool such as CD-HIT [99]. This would allow for the construction of testing sets containing phage sequences with little or no similarity to sequences in the training set, allowing for evaluation on 'novel' (from the perspective of the model) phages. Public release of such a dataset may itself have value, since many viral classification models use relatively course methods of splitting training and testing sets, such as splitting by sequence upload date.

4.2.3 Limited *de novo* Annotation Supoort

The default VIBES workflow does little to support *de novo* annotation of phage sequences, in which bacterial genomes are scanned for a broad range of reference prophages. Currently, to conduct such an analysis with VIBES, users would have to identify or construct a multi-FASTA

file containing a diverse set of known temperate phage genomes or multiple sequence alignments of related phage genomes. *De novo* annotation is further hampered by the lack of further analysis of potential prophages identified by VIBES, discussed above, which makes it more difficult to discern prophage regions from homologous bacterial sequences, especially when given a diverse set of temperate phage queries.

This limitation could be addressed by the generation of a default VIBES *de novo* annotation database. Such a database could be supplied to VIBES as its query file without further modification of the workflow, maintaining the ability for users to substitute their own queries. *De novo* viral annotation databases used by similarity search approaches typically contain viral protein sequences [64, 68, 69, 70], though some do use viral nucleotide sequences [62, 63]. Phage protein sequences could be sourced from databases such as IMG/VR [100], pVOGs [101], or PHROG [76]. To maximize search sensitivity to distant homologues, sequences should be clustered into protein families, and each family aligned with multiple sequence alignment software such as MAFFT [102]. Aligned protein families can then be provided to VIBES in a multi-FASTA query file, which the workflow will use to generate profile Hidden Markov Models that capture information about conserved regions and likely mutations for each protein family [103].

4.2.4 Lack of Search Tool Options

Though the nhmmer search algorithm is more sensitive to distant homology than other search algorithms [78], it is relatively computationally expensive. Indeed, viral sequence annotation tools that use HMMER search algorithms have been demonstrated to be among the slowest annotation techniques [104]. Users seeking to conduct large-scale search who are willing to trade lower sensitivity for lower program runtime or resource usage may therefore be dissuaded from using VIBES.

This limitation is relative easy to address through the addition of a broader choice of sequence similarity search tools with less expected sensitivity and resource usage such as LAST, a fast DNA to DNA search tool [105], or blastn [82]. Inclusion of other search tools would primarily require

modification of the VIBES Nextflow script, tool output parsing scripts, and configuration file, a relatively straightforward task for developers familiar with Nextflow.

4.2.5 General Caveats to Viral Sequence Annotation

It is worth spending some time discussing caveats that apply not just to the VIBES workflow, but to viral sequence annotation generally. A significant challenge facing the field is the highly biased composition of the relatively limited set of complete genomes sequenced from samples isolated in labs. A 2021 study seeking to compile the complete genomes of phages isolated from bacterial hosts identified 14,244 such genomes available at the time [106]. 75% of the phages in their dataset (INPHARED) are derived from only 30 bacterial genera, while ∼54% of temperate phage genomes were isolated from only 3 host genera. The dataset also displayed a significant bias in favor of lytic phage genomes, with only ∼30% of the sequences in their dataset predicted to contain genes necessary for the initiation of a lysogenic cycle.

When taking into account phage genomes computationally predicted from sources such as metagenomic data, a considerably larger quantity of sequences is available. One recently compiled database boasts a library of over 765,000 nonredundant putative phage sequences [98]. However, the results of computational techniques that seek to annotate phage sequences depend heavily on isolated and sequenced phage genomes that ultimately comprise our ground truth datasets. Techniques that employ sequence similarity search can only find sequences with some amount of homology to sequences in the query database, while machine learning techniques (especially neural network approaches, which have recently been popular in viral sequence analysis and annotation methods [66, 67, 69, 107]) are vulnerable to dataset composition biases that skew their predictions [108, 109]. The impact of this dataset composition bias is evident in the findings of a recent benchmarking study: all benchmarked models and tools were less sensitive to phages not in the class Caudoviricetes, whose members make up ∼93% of publicly available phage sequences on RefSeq [97]. These limitations are not intractable, and computational techniques have been successfully employed in the identification of novel phages [110, 111], but should nevertheless be kept in mind

when evaluating computationally predicted phage sequences.

Another limitation facing the field generally is the absence of standardized, widely used performance benchmarks. Currently, each viral sequence annotation tool uses its own, likely imperfect means of benchmarking performance. Some approaches, including mine, demonstrate usefulness with an analysis or a ground truth comparison. Other approaches measure performance by constructing in-house benchmarking datasets, but some only report accuracy while others additionally report precision, recall, and F1 metrics. This smear of approaches makes it difficult to comprehensively compare the performance of different tools without independent testing [104, 97], as even when detailed metrics are reported, they are generally reported on different datasets. Though independent testing of tools is obviously valuable, the viral annotation software development community, and my work in particular, would benefit from the development of a standard core of benchmarks that developers could use to assess tool performance across a variety of relevant tasks such as viral sequence identification, viral gene annotation, and *att* site annotation. This would allow for direct comparison to existing approaches and alleviate the need to devise new benchmarks to demonstrate the efficacy of each tool. Centralized benchmarks have been especially successful in driving innovation in protein structure prediction [112], genome assembly [113], protein function prediction [114], and multiple sequence alignment [115].

4.2.6 Additional Future Directions

Another potential extension of VIBES would be the inclusion of an *att* site annotation technique. This is a somewhat common feature of prophage annotation software (Table 1.2) that has use in determining the exact boundaries of prophage regions in which *att* sites remain intact. This could be accomplished by setting an existing sequence similarity search tool to look for short, exact repeats in a window around matches to query prophages [70], or by devising a search tool designed explicitly to search for direct flanking repeats.

BIBLIOGRAPHY

[1] diCenzo George C. and Finan Turlough M., "The divided bacterial genome: Structure, function, and evolution," *Microbiol. Mol. Biol. Rev.*, vol. 81, no. 3, pp. 10.1128/mmbr.00 019–17, Aug. 2017.

[2] C. Baril, C. Richaud, G. Baranton, and I. S. Saint Girons, "Linear chromosome of borrelia burgdorferi," *Res. Microbiol.*, vol. 140, no. 8, pp. 507–516, Oct. 1989.

[3] M. S. Ferdows and A. G. Barbour, "Megabase-sized linear DNA in the bacterium borrelia burgdorferi, the lyme disease agent," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 86, no. 15, pp. 5969–5973, Aug. 1989.

[4] P. S. G. Chain, V. J. Denef, K. T. Konstantinidis, L. M. Vergez, L. Agulló, V. L. Reyes, L. Hauser, M. Córdova, L. Gómez, M. González, M. Land, V. Lao, F. Larimer, J. J. LiPuma, E. Mahenthiralingam, S. A. Malfatti, C. J. Marx, J. J. Parnell, A. Ramette, P. Richardson, M. Seeger, D. Smith, T. Spilker, W. J. Sul, T. V. Tsoi, L. E. Ulrich, I. B. Zhulin, and J. M. Tiedje, "Burkholderia xenovorans LB400 harbors a multi-replicon, 9.73-mbp genome shaped for versatility," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 42, pp. 15 280–15 287, Oct. 2006.

[5] J. T. Van Leuven, R. C. Meister, C. Simon, and J. P. McCutcheon, "Sympatric speciation in a bacterial endosymbiont results in two genomes with the functionality of one," *Cell*, vol. 158, no. 6, pp. 1270–1280, Sep. 2014.

[6] K. Han, Z.-F. Li, R. Peng, L.-P. Zhu, T. Zhou, L.-G. Wang, S.-G. Li, X.-B. Zhang, W. Hu, Z.-H. Wu, N. Qin, and Y.-Z. Li, "Extraordinary expansion of a sorangium cellulosum genome from an alkaline milieu," *Sci. Rep.*, vol. 3, p. 2101, 2013.

[7] A. Mira, H. Ochman, and N. A. Moran, "Deletional bias and the evolution of bacterial genomes," *Trends Genet.*, vol. 17, no. 10, pp. 589–596, Oct. 2001.

[8] A. E. Osbourn and B. Field, "Operons," *Cell. Mol. Life Sci.*, vol. 66, no. 23, pp. 3755–3775, Dec. 2009.

[9] L. R. Snyder, J. E. Peters, T. M. Henkin, and W. Champness, *Molecular Genetics of Bacteria*. Wiley, Jan. 2014.

[10] C. A. Suttle, "Marine viruses–major players in the global ecosystem," *Nat. Rev. Microbiol.*, vol. 5, no. 10, pp. 801–812, Oct. 2007.

[11] P. Puigbò, K. S. Makarova, D. M. Kristensen, Y. I. Wolf, and E. V. Koonin, "Reconstruction of the evolution of microbial defense systems," *BMC Evol. Biol.*, vol. 17, no. 1, p. 94, Apr. 2017.

[12] F. Rousset, F. Depardieu, S. Miele, J. Dowding, A.-L. Laval, E. Lieberman, D. Garry, E. P. C. Rocha, A. Bernheim, and D. Bikard, "Phages and their satellites encode hotspots of antiviral systems," *Cell Host Microbe*, vol. 30, no. 5, pp. 740–753.e5, May 2022.

[13] A. Fillol-Salom, J. T. Rostøl, A. D. Ojiogu, J. Chen, G. Douce, S. Humphrey, and J. R. Penadés, "Bacteriophages benefit from mobilizing pathogenicity islands encoding immune systems against competitors," *Cell*, vol. 185, no. 17, pp. 3248–3262.e20, Aug. 2022.

[14] H. Georjon and A. Bernheim, "The highly diverse antiphage defence systems of bacteria," *Nat. Rev. Microbiol.*, vol. 21, no. 10, pp. 686–700, Oct. 2023.

[15] A. Lopatina, N. Tal, and R. Sorek, "Abortive infection: Bacterial suicide as an antiviral immune strategy," *Annu Rev Virol*, vol. 7, no. 1, pp. 371–384, Sep. 2020.

[16] F. Tesson, A. Hervé, E. Mordret, M. Touchon, C. d'Humières, J. Cury, and A. Bernheim, "Systematic and quantitative view of the antiviral arsenal of prokaryotes," *Nat. Commun.*, vol. 13, no. 1, p. 2561, May 2022.

[17] M. R. Tock and D. T. F. Dryden, "The biology of restriction and anti-restriction," *Curr. Opin. Microbiol.*, vol. 8, no. 4, pp. 466–472, Aug. 2005.

[18] F. Hille, H. Richter, S. P. Wong, M. Bratovič, S. Ressel, and E. Charpentier, "The biology of CRISPR-Cas: Backward and forward," *Cell*, vol. 172, no. 6, pp. 1239–1259, Mar. 2018.

[19] L. A. Gao, M. E. Wilkinson, J. Strecker, K. S. Makarova, R. K. Macrae, E. V. Koonin, and F. Zhang, "Prokaryotic innate immunity through pattern recognition of conserved viral proteins," *Science*, vol. 377, no. 6607, p. eabm4096, Aug. 2022.

[20] F. Depardieu, J.-P. Didier, A. Bernheim, A. Sherlock, H. Molina, B. Duclos, and D. Bikard, "A eukaryotic-like Serine/Threonine kinase protects staphylococci against phages," *Cell Host Microbe*, vol. 20, no. 4, pp. 471–481, Oct. 2016.

[21] J. Garb, A. Lopatina, A. Bernheim, M. Zaremba, V. Siksnys, S. Melamed, A. Leavitt, A. Millman, G. Amitai, and R. Sorek, "Multiple phage resistance systems inhibit infection via SIR2-dependent NAD+ depletion," *Nat Microbiol*, vol. 7, no. 11, pp. 1849–1856, Nov. 2022.

[22] T. Zhang, H. Tamman, K. Coppieters 't Wallant, T. Kurata, M. LeRoux, S. Srikant, T. Brodiazhenko, A. Cepauskas, A. Talavera, C. Martens, G. C. Atkinson, V. Hauryliuk, A. Garcia-Pino, and M. T. Laub, "Direct activation of a bacterial innate immune system by a viral capsid protein," *Nature*, vol. 612, no. 7938, pp. 132–140, Nov. 2022.

[23] C. K. Guegler and M. T. Laub, "Shutoff of host transcription triggers a toxin-antitoxin system to cleave phage RNA and abort infection," *Mol. Cell*, vol. 81, no. 11, pp. 2361–2373.e9, Jun. 2021.

[24] G. Kaufmann, "Anticodon nucleases," *Trends Biochem. Sci.*, vol. 25, no. 2, pp. 70–74, Feb. 2000.

[25] B. Y. Hsueh, G. B. Severin, C. A. Elg, E. J. Waldron, A. Kant, A. J. Wessel, J. A. Dover, C. R. Rhoades, B. J. Ridenhour, K. N. Parent, M. B. Neiditch, J. Ravi, E. M. Top, and C. M.

Waters, "Phage defence by deaminase-mediated depletion of deoxynucleotides in bacteria," *Nature Microbiology*, vol. 7, no. 8, pp. 1210–1220, Jul. 2022.

[26] H. A. Nash, "Integration and excision of bacteriophage lambda: the mechanism of conservation site specific recombination," *Annu. Rev. Genet.*, vol. 15, pp. 143–167, 1981.

[27] M. B. Łobocka, D. J. Rose, G. Plunkett, 3rd, M. Rusin, A. Samojedny, H. Lehnherr, M. B. Yarmolinsky, and F. R. Blattner, "Genome of bacteriophage P1," *J. Bacteriol.*, vol. 186, no. 21, pp. 7032–7068, Nov. 2004.

[28] A. I. Bukhari and D. Zipser, "Random insertion of mu-1 DNA within a single gene," *Nat. New Biol.*, vol. 236, no. 69, pp. 240–243, Apr. 1972.

[29] C. Howard-Varona, K. R. Hargreaves, S. T. Abedon, and M. B. Sullivan, "Lysogeny in nature: mechanisms, impact and ecology of temperate phages," *ISME J.*, vol. 11, no. 7, pp. 1511–1520, Jul. 2017.

[30] G. López-Leal, L. C. Camelo-Valera, J. M. Hurtado-Ramírez, J. Verleyen, S. Castillo-Ramírez, and A. Reyes-Muñoz, "Mining of thousands of prokaryotic genomes reveals high abundance of prophages with a strictly narrow host range," *mSystems*, vol. 7, no. 4, p. e0032622, Aug. 2022.

[31] P. C. M. Fogg, S. Colloms, S. Rosser, M. Stark, and M. C. M. Smith, "New applications for phage integrases," *J. Mol. Biol.*, vol. 426, no. 15, pp. 2703–2716, Jul. 2014.

[32] J. Warwick-Dugdale, H. H. Buchholz, M. J. Allen, and B. Temperton, "Host-hijacking and planktonic piracy: how phages command the microbial high seas," *Virol. J.*, vol. 16, no. 1, p. 15, Feb. 2019.

[33] R. J. Puxty, A. D. Millard, D. J. Evans, and D. J. Scanlan, "Shedding new light on viral photosynthesis," *Photosynth. Res.*, vol. 126, no. 1, pp. 71–97, Oct. 2015.

[34] H. Brüssow, C. Canchaya, and W.-D. Hardt, "Phages and the evolution of bacterial pathogens: from genomic rearrangements to lysogenic conversion," *Microbiol. Mol. Biol. Rev.*, vol. 68, no. 3, pp. 560–602, table of contents, Sep. 2004.

[35] J. Bondy-Denomy, J. Qian, E. R. Westra, A. Buckling, D. S. Guttman, A. R. Davidson, and K. L. Maxwell, "Prophages mediate defense against phage infection through diverse mechanisms," *ISME J.*, vol. 10, no. 12, pp. 2854–2866, Dec. 2016.

[36] P. L. Wagner and M. K. Waldor, "Bacteriophage control of bacterial virulence," *Infect. Immun.*, vol. 70, no. 8, pp. 3985–3993, Aug. 2002.

[37] R. Feiner, T. Argov, L. Rabinovich, N. Sigal, I. Borovok, and A. A. Herskovits, "A new perspective on lysogeny: prophages as active regulatory switches of bacteria," *Nat. Rev. Microbiol.*, vol. 13, no. 10, pp. 641–650, Oct. 2015.

[38] S. Roux, M. Krupovic, R. A. Daly, A. L. Borges, S. Nayfach, F. Schulz, A. Sharrar, P. B. Matheus Carnevali, J.-F. Cheng, N. N. Ivanova, J. Bondy-Denomy, K. C. Wrighton, T. Woyke, A. Visel, N. C. Kyrpides, and E. A. Eloe-Fadrosh, "Cryptic inoviruses revealed as pervasive in bacteria and archaea across earth's biomes," *Nat Microbiol*, vol. 4, no. 11, pp. 1895–1906, Nov. 2019.

[39] P. R. Secor, E. B. Burgener, M. Kinnersley, L. K. Jennings, V. Roman-Cruz, M. Popescu, J. D. Van Belleghem, N. Haddock, C. Copeland, L. A. Michaels, C. R. de Vries, Q. Chen, J. Pourtois, T. J. Wheeler, C. E. Milla, and P. L. Bollyky, "Pf bacteriophage and their impact on pseudomonas virulence, mammalian immunity, and chronic infections," *Front. Immunol.*, vol. 11, p. 244, Feb. 2020.

[40] K. Fiedoruk, M. Zakrzewska, T. Daniluk, E. Piktel, S. Chmielewska, and R. Bucki, "Two lineages of pseudomonas aeruginosa filamentous phages: Structural uniformity over integration preferences," *Genome Biol. Evol.*, vol. 12, no. 10, pp. 1765–1781, Oct. 2020.

[41] P. Knezevic, M. Voet, and R. Lavigne, "Prevalence of pf1-like (pro)phage genetic elements among pseudomonas aeruginosa isolates," *Virology*, vol. 483, pp. 64–71, Sep. 2015.

[42] I. D. Hay and T. Lithgow, "Filamentous phages: masters of a microbial sharing economy," *EMBO Rep.*, vol. 20, no. 6, Jun. 2019.

[43] P. Knezevic, E. M. Adriaenssens, and Ictv Report Consortium, "ICTV virus taxonomy profile: Inoviridae," *J. Gen. Virol.*, vol. 102, no. 7, Jul. 2021.

[44] C. M. Schwartzkopf, V. L. Taylor, M.-C. Groleau, D. R. Faith, A. K. Schmidt, T. L. Lamma, D. M. Brooks, E. Déziel, K. L. Maxwell, and P. R. Secor, "Inhibition of pqs signaling by the pf bacteriophage protein pfse enhances viral replication in pseudomonas aeruginosa," *Molecular Microbiology*, 2023.

[45] J. S. Mattick, "Type IV pili and twitching motility," *Annu. Rev. Microbiol.*, vol. 56, pp. 289–314, Jan. 2002.

[46] A. Walburger, C. Lazdunski, and Y. Corda, "The Tol/Pal system function requires an interaction between the c-terminal domain of TolA and the n-terminal domain of TolB," *Mol. Microbiol.*, vol. 44, no. 3, pp. 695–708, May 2002.

[47] Martínez Eriel and Campos-Gómez Javier, "Pf filamentous phage requires UvrD for replication in pseudomonas aeruginosa," *mSphere*, vol. 1, no. 1, pp. 10.1128/msphere.00 104–15, Feb. 2016.

[48] G. G. Kneale, "Dissociation of the pf1 nucleoprotein assembly complex and characterisation of the DNA binding protein," *Biochim. Biophys. Acta*, vol. 739, no. 2, pp. 216–224, Mar. 1983.

[49] M. Chen, J. C. Samuelson, F. Jiang, M. Muller, A. Kuhn, and R. E. Dalbey, "Direct interaction of YidC with the sec-independent pf3 coat protein during its membrane protein insertion," *J. Biol. Chem.*, vol. 277, no. 10, pp. 7670–7675, Mar. 2002.

[50] J. Rakonjac, N. J. Bennett, J. Spagnuolo, D. Gagic, and M. Russel, "Filamentous bacteriophage: biology, phage display and nanotechnology applications," *Curr. Issues Mol. Biol.*, vol. 13, no. 2, pp. 51–76, Apr. 2011.

[51] Y. Li, X. Liu, K. Tang, P. Wang, Z. Zeng, Y. Guo, and X. Wang, "Excisionase in pf filamentous prophage controls lysis-lysogeny decision-making in pseudomonas aeruginosa," *Mol. Microbiol.*, vol. 111, no. 2, pp. 495–513, Feb. 2019.

[52] Q. Wei, P. N. L. Minh, A. Dötsch, F. Hildebrand, W. Panmanee, A. Elfarash, S. Schulz, S. Plaisance, D. Charlier, D. Hassett, S. Häussler, and P. Cornelis, "Global regulation of gene expression by OxyR in an important human opportunistic pathogen," *Nucleic Acids Res.*, vol. 40, no. 10, pp. 4320–4333, May 2012.

[53] Y. Lee, S. Song, L. Sheng, L. Zhu, J.-S. Kim, and T. K. Wood, "Substrate binding protein DppA1 of ABC transporter DppBCDF increases biofilm formation in pseudomonas aeruginosa by inhibiting pf5 prophage lysis," *Front. Microbiol.*, vol. 9, p. 30, Jan. 2018.

[54] S. Castang and S. L. Dove, "Basis for the essentiality of H-NS family members in pseudomonas aeruginosa," *J. Bacteriol.*, vol. 194, no. 18, pp. 5101–5109, Sep. 2012.

[55] S. A. Rice, C. H. Tan, P. J. Mikkelsen, V. Kung, J. Woo, M. Tay, A. Hauser, D. McDougald, J. S. Webb, and S. Kjelleberg, "The biofilm life cycle and virulence of pseudomonas aeruginosa are dependent on a filamentous prophage," *ISME J.*, vol. 3, no. 3, pp. 271–282, Mar. 2009.

[56] A. K. Tarafder, A. von Kügelgen, A. J. Mellul, U. Schulze, D. G. A. L. Aarts, and T. A. M. Bharat, "Phage liquid crystalline droplets form occlusive sheaths that encapsulate and protect infectious rod-shaped bacteria," *Proceedings of the National Academy of Sciences*, vol. 117, no. 9, pp. 4724–4731, 2020.

[57] P. R. Secor, J. M. Sweere, L. A. Michaels, A. V. Malkovskiy, D. Lazzareschi, E. Katznelson, J. Rajadas, M. E. Birnbaum, A. Arrigoni, K. R. Braun, S. P. Evanko, D. A. Stevens, W. Kaminsky, P. K. Singh, W. C. Parks, and P. L. Bollyky, "Filamentous bacteriophage

promote biofilm assembly and function," *Cell Host Microbe*, vol. 18, no. 5, pp. 549–559, Nov. 2015.

[58] E. B. Burgener, J. M. Sweere, M. S. Bach, P. R. Secor, N. Haddock, L. K. Jennings, R. L. Marvig, H. K. Johansen, E. Rossi, X. Cao, L. Tian, L. Nedelec, S. Molin, P. L. Bollyky, and C. E. Milla, "Filamentous bacteriophages are associated with chronic pseudomonas lung infections and antibiotic resistance in cystic fibrosis," *Sci. Transl. Med.*, vol. 11, no. 488, Apr. 2019.

[59] J. M. Sweere, J. D. Van Belleghem, H. Ishak, M. S. Bach, M. Popescu, V. Sunkari, G. Kaber, R. Manasherob, G. A. Suh, X. Cao, C. R. de Vries, D. N. Lam, P. L. Marshall, M. Birukova, E. Katznelson, D. V. Lazzareschi, S. Balaji, S. G. Keswani, T. R. Hawn, P. R. Secor, and P. L. Bollyky, "Bacteriophage trigger antiviral immunity and prevent clearance of bacterial infection," *Science*, vol. 363, no. 6434, Mar. 2019.

[60] M. S. Bach, C. R. de Vries, A. Khosravi, J. M. Sweere, M. C. Popescu, Q. Chen, S. Demirdjian, A. Hargil, J. D. Van Belleghem, G. Kaber, M. Hajfathalian, E. B. Burgener, D. Liu, Q.-L. Tran, T. Dharmaraj, M. Birukova, V. Sunkari, S. Balaji, N. Ghosh, S. S. Mathew-Steiner, M. S. El Masry, S. G. Keswani, N. Banaei, L. Nedelec, C. K. Sen, V. Chandra, P. R. Secor, G. A. Suh, and P. L. Bollyky, "Filamentous bacteriophage delays healing of pseudomonas-infected wounds," *Cell Rep Med*, vol. 3, no. 6, p. 100656, Jun. 2022.

[61] D. Paez-Espino, E. A. Eloe-Fadrosh, G. A. Pavlopoulos, A. D. Thomas, M. Huntemann, N. Mikhailova, E. Rubin, N. N. Ivanova, and N. C. Kyrpides, "Uncovering earth's virome," *Nature*, vol. 536, no. 7617, pp. 425–430, Aug. 2016.

[62] D. Arndt, J. R. Grant, A. Marcu, T. Sajed, A. Pon, Y. Liang, and D. S. Wishart, "PHASTER: a better, faster version of the PHAST phage search tool," *Nucleic Acids Res.*, vol. 44, no. W1, pp. W16–21, Jul. 2016.

[63] R. Gan, F. Zhou, Y. Si, H. Yang, C. Chen, C. Ren, J. Wu, and F. Zhang, "DBSCAN-SWA:

An integrated tool for rapid prophage detection and annotation," *Front. Genet.*, vol. 13, p. 885048, Apr. 2022.

[64] S. Roux, F. Enault, B. L. Hurwitz, and M. B. Sullivan, "VirSorter: mining viral signal from microbial genomic data," *PeerJ*, vol. 3, p. e985, May 2015.

[65] J. Ren, N. A. Ahlgren, Y. Y. Lu, J. A. Fuhrman, and F. Sun, "VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data," *Microbiome*, vol. 5, no. 1, p. 69, Jul. 2017.

[66] J. Ren, K. Song, C. Deng, N. A. Ahlgren, J. A. Fuhrman, Y. Li, X. Xie, R. Poplin, and F. Sun, "Identifying viruses from metagenomic data using deep learning," *Quant Biol*, vol. 8, no. 1, pp. 64–77, Mar. 2020.

[67] Y. Miao, F. Liu, T. Hou, and Y. Liu, "Virtifier: A deep learning-based identifier for viral sequences from metagenomes," *Bioinformatics*, Dec. 2021.

[68] J. Guo, B. Bolduc, A. A. Zayed, A. Varsani, G. Dominguez-Huerta, T. O. Delmont, A. A. Pratama, M. C. Gazitúa, D. Vik, M. B. Sullivan, and S. Roux, "VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses," *Microbiome*, vol. 9, no. 1, p. 37, Feb. 2021.

[69] K. Kieft, Z. Zhou, and K. Anantharaman, "VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences," *Microbiome*, vol. 8, no. 1, p. 90, Jun. 2020.

[70] W. Song, H.-X. Sun, C. Zhang, L. Cheng, Y. Peng, Z. Deng, D. Wang, Y. Wang, M. Hu, W. Liu, H. Yang, Y. Shen, J. Li, L. You, and M. Xiao, "Prophage hunter: an integrative hunting tool for active prophages," *Nucleic Acids Res.*, vol. 47, no. W1, pp. W74–W80, Jul. 2019.

[71] D. S. Wishart, S. Han, S. Saha, E. Oler, H. Peters, J. R. Grant, P. Stothard, and V. Gautam, "PHASTEST: faster than PHASTER, better than PHAST," *Nucleic Acids Res.*, vol. 51, no. W1, pp. W443–W450, Jul. 2023.

[72] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," *Linux J.*, vol. 2014, no. 239, p. 2, Mar. 2014.

[73] T. Seemann, "Prokka: rapid prokaryotic genome annotation," *Bioinformatics*, vol. 30, no. 14, pp. 2068–2069, Jul. 2014.

[74] J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G. A. Salazar, E. L. L. Sonnhammer, S. C. E. Tosatto, L. Paladin, S. Raj, L. J. Richardson, R. D. Finn, and A. Bateman, "Pfam: The protein families database in 2021," *Nucleic Acids Res.*, vol. 49, no. D1, pp. D412–D419, Jan. 2021.

[75] P. Jones, D. Binns, H.-Y. Chang, M. Fraser, W. Li, C. McAnulla, H. McWilliam, J. Maslen, A. Mitchell, G. Nuka, S. Pesseat, A. F. Quinn, A. Sangrador-Vegas, M. Scheremetjew, S.-Y. Yong, R. Lopez, and S. Hunter, "InterProScan 5: genome-scale protein function classification," *Bioinformatics*, vol. 30, no. 9, pp. 1236–1240, May 2014.

[76] P. Terzian, E. Olo Ndela, C. Galiez, J. Lossouarn, R. E. Pérez Bucio, R. Mom, A. Toussaint, M.-A. Petit, and F. Enault, "PHROG: families of prokaryotic virus proteins clustered using remote homology," *NAR Genom Bioinform*, vol. 3, no. 3, p. lqab067, Sep. 2021.

[77] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," *Nat. Biotechnol.*, vol. 35, no. 4, pp. 316–319, Apr. 2017.

[78] T. J. Wheeler and S. R. Eddy, "nhmmer: DNA homology search with profile HMMs," *Bioinformatics*, vol. 29, no. 19, pp. 2487–2489, Oct. 2013.

[79] G. Krause and T. J. Wheeler, "BATH: Better alignments with translated HMMER."

[80] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLoS One*, vol. 12, no. 5, p. e0177459, May 2017.

[81] "Easel - a C library for biological sequence analysis."

[82] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, "BLAST+: architecture and applications," *BMC Bioinformatics*, vol. 10, p. 421, Dec. 2009.

[83] K. E. McElroy, J. G. K. Hui, J. K. K. Woo, A. W. S. Luk, J. S. Webb, S. Kjelleberg, S. A. Rice, and T. Thomas, "Strain-specific parallel evolution drives short-term diversification during *Pseudomonas aeruginosa* biofilm formation," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 14, pp. E1419–27, Apr. 2014.

[84] State Public Health Bioinformatics Community, "Prokka docker image."

[85] J. W. Roddy, G. T. Lesica, and T. J. Wheeler, "SODA: a TypeScript/JavaScript library for visualizing biological sequence annotation," *NAR Genom Bioinform*, vol. 4, no. 4, p. lqac077, Dec. 2022.

[86] G. L. Winsor, E. J. Griffiths, R. Lo, B. K. Dhillon, J. A. Shay, and F. S. L. Brinkman, "Enhanced annotations and features for comparing thousands of pseudomonas genomes in the pseudomonas genome database," *Nucleic Acids Res.*, vol. 44, no. D1, pp. D646–53, Jan. 2016.

[87] J. S. Webb, M. Lau, and S. Kjelleberg, "Bacteriophage and phenotypic variation in pseudomonas aeruginosa biofilm development," *J. Bacteriol.*, vol. 186, no. 23, pp. 8066–8073, Dec. 2004.

[88] M. J. Mooij, E. Drenkard, M. A. Llamas, C. M. J. E. Vandenbroucke-Grauls, P. H. M. Savelkoul, F. M. Ausubel, and W. Bitter, "Characterization of the integrated filamentous phage pf5 and its involvement in small-colony formation," *Microbiology*, vol. 153, no. Pt 6, pp. 1790–1798, Jun. 2007.

[89] "Bollyky lab website," https://bollykylab.com/research/, Oct. 2020, accessed: 2023-8-23.

[90] C. Canchaya, C. Proux, G. Fournous, A. Bruttin, and H. Brüssow, "Prophage genomics," *Microbiol. Mol. Biol. Rev.*, vol. 67, no. 2, pp. 238–76, table of contents, Jun. 2003.

[91] J. W. Streicher, Wellcome Sanger Institute Tree of Life programme, Wellcome Sanger Institute Scientific Operations: DNA Pipelines collective, Tree of Life Core Informatics collective, and Darwin Tree of Life Consortium, "The genome sequence of the common frog, rana temporaria linnaeus 1758," *Wellcome Open Res*, vol. 6, p. 286, Oct. 2021.

[92] W. Shen, S. Le, Y. Li, and F. Hu, "SeqKit: A Cross-Platform and ultrafast toolkit for FASTA/Q file manipulation," *PLoS One*, vol. 11, no. 10, p. e0163962, Oct. 2016.

[93] R. Wu, C. A. Smith, G. W. Buchko, I. K. Blaby, D. Paez-Espino, N. C. Kyrpides, Y. Yoshikuni, J. E. McDermott, K. S. Hofmockel, J. R. Cort, and J. K. Jansson, "Structural characterization of a soil viral auxiliary metabolic gene product - a functional chitosanase," *Nat. Commun.*, vol. 13, no. 1, p. 5485, Sep. 2022.

[94] J. W. Roddy, K. Carey, and T. J. Wheeler, "Aurora: Adjudicate uncertain regions and output reliable annotations," https://github.com/TravisWheelerLab/aurora, 2023.

[95] K. M. Carey, R. Hubley, G. T. Lesica, D. Olson, J. W. Roddy, J. Rosen, A. Shingleton, A. F. Smit, and T. J. Wheeler, "PolyA: a tool for adjudicating competing annotations of biological sequences," Feb. 2021.

[96] S. Nayfach, A. P. Camargo, F. Schulz, E. Eloe-Fadrosh, S. Roux, and N. C. Kyrpides, "CheckV assesses the quality and completeness of metagenome-assembled viral genomes," *Nat. Biotechnol.*, vol. 39, no. 5, pp. 578–585, Dec. 2020.

[97] K. E. Schackart, 3rd, J. B. Graham, A. J. Ponsero, and B. L. Hurwitz, "Evaluation of computational phage detection tools for metagenomic datasets," *Front. Microbiol.*, vol. 14, p. 1078760, Jan. 2023.

[98] R. H. Wang, S. Yang, Z. Liu, Y. Zhang, X. Wang, Z. Xu, J. Wang, and S. C. Li, "PhageScope: a well-annotated bacteriophage database with automatic analyses and visualizations," *Nucleic Acids Res.*, Oct. 2023.

[99] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, "CD-HIT: accelerated for clustering the next-generation sequencing data," *Bioinformatics*, vol. 28, no. 23, pp. 3150–3152, Dec. 2012.

[100] A. P. Camargo, S. Nayfach, I.-M. A. Chen, K. Palaniappan, A. Ratner, K. Chu, S. J. Ritter, T. B. K. Reddy, S. Mukherjee, F. Schulz, L. Call, R. Y. Neches, T. Woyke, N. N. Ivanova, E. A. Eloe-Fadrosh, N. C. Kyrpides, and S. Roux, "IMG/VR v4: an expanded database of uncultivated virus genomes within a framework of extensive functional, taxonomic, and ecological metadata," *Nucleic Acids Res.*, Nov. 2022.

[101] A. L. Grazziotin, E. V. Koonin, and D. M. Kristensen, "Prokaryotic virus orthologous groups (pVOGs): a resource for comparative genomics and protein family annotation," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D491–D498, Jan. 2017.

[102] K. Katoh, K. Misawa, K.-I. Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform," *Nucleic Acids Res.*, vol. 30, no. 14, pp. 3059–3066, Jul. 2002.

[103] S. R. Eddy, "Profile hidden markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.

[104] M. J. Roach, K. McNair, M. Michalczyk, S. K. Giles, L. K. Inglis, E. Pargin, J. Barylski, S. Roux, P. Decewicz, and R. A. Edwards, "Philympics 2021: Prophage predictions perplex programs," *F1000Res.*, vol. 10, no. 758, p. 758, Apr. 2022.

[105] S. M. Kiełbasa, R. Wan, K. Sato, P. Horton, and M. C. Frith, "Adaptive seeds tame genomic sequence comparison," *Genome Res.*, vol. 21, no. 3, pp. 487–493, Mar. 2011.

[106] R. Cook, N. Brown, T. Redgwell, B. Rihtman, M. Barnes, M. Clokie, D. J. Stekel, J. Hobman, M. A. Jones, and A. Millard, "INfrastructure for a PHAge REference database: Identifica-

tion of Large-Scale biases in the current collection of cultured phage genomes," *Phage (New Rochelle)*, vol. 2, no. 4, pp. 214–223, Dec. 2021.

[107] Z. Bai, Y.-Z. Zhang, S. Miyano, R. Yamaguchi, K. Fujimoto, S. Uematsu, and S. Imoto, "Identification of bacteriophage genome sequences with representation learning," *Bioinformatics*, Aug. 2022.

[108] L. Chen, A. Cruz, S. Ramsey, C. J. Dickson, J. S. Duca, V. Hornak, D. R. Koes, and T. Kurtzman, "Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening," *PLoS One*, vol. 14, no. 8, p. e0220113, Aug. 2019.

[109] B. Garcia Santa Cruz, M. N. Bossa, J. Sölter, and A. D. Husch, "Public covid-19 x-ray datasets and their impact on model bias - a systematic review of a significant problem," *Med. Image Anal.*, vol. 74, p. 102225, Dec. 2021.

[110] B. E. Dutilh, N. Cassman, K. McNair, S. E. Sanchez, G. G. Z. Silva, L. Boling, J. J. Barr, D. R. Speth, V. Seguritan, R. K. Aziz, B. Felts, E. A. Dinsdale, J. L. Mokili, and R. A. Edwards, "A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes," *Nat. Commun.*, vol. 5, p. 4498, Jul. 2014.

[111] M. J. Tisza, D. V. Pastrana, N. L. Welch, B. Stewart, A. Peretti, G. J. Starrett, Y.-Y. S. Pang, S. R. Krishnamurthy, P. A. Pesavento, D. H. McDermott, P. M. Murphy, J. L. Whited, B. Miller, J. Brenchley, S. P. Rosshart, B. Rehermann, J. Doorbar, B. A. Ta'ala, O. Pletnikova, J. C. Troncoso, S. M. Resnick, B. Bolduc, M. B. Sullivan, A. Varsani, A. M. Segall, and C. B. Buck, "Discovery of several thousand highly diverse circular DNA viruses," *Elife*, vol. 9, Feb. 2020.

[112] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, "Critical assessment of methods of protein structure prediction (CASP)-Round XV," *Proteins*, vol. 91, no. 12, pp. 1539–1549, Dec. 2023.

[113] K. R. Bradnam, J. N. Fass, A. Alexandrov, P. Baranay, M. Bechner, I. Birol, S. Boisvert, J. A. Chapman, G. Chapuis, R. Chikhi, H. Chitsaz, W.-C. Chou, J. Corbeil, C. Del Fabbro, T. R. Docking, R. Durbin, D. Earl, S. Emrich, P. Fedotov, N. A. Fonseca, G. Ganapathy, R. A. Gibbs, S. Gnerre, E. Godzaridis, S. Goldstein, M. Haimel, G. Hall, D. Haussler, J. B. Hiatt, I. Y. Ho, J. Howard, M. Hunt, S. D. Jackman, D. B. Jaffe, E. D. Jarvis, H. Jiang, S. Kazakov, P. J. Kersey, J. O. Kitzman, J. R. Knight, S. Koren, T.-W. Lam, D. Lavenier, F. Laviolette, Y. Li, Z. Li, B. Liu, Y. Liu, R. Luo, I. Maccallum, M. D. Macmanes, N. Maillet, S. Melnikov, D. Naquin, Z. Ning, T. D. Otto, B. Paten, O. S. Paulo, A. M. Phillippy, F. Pina-Martins, M. Place, D. Przybylski, X. Qin, C. Qu, F. J. Ribeiro, S. Richards, D. S. Rokhsar, J. G. Ruby, S. Scalabrin, M. C. Schatz, D. C. Schwartz, A. Sergushichev, T. Sharpe, T. I. Shaw, J. Shendure, Y. Shi, J. T. Simpson, H. Song, F. Tsarev, F. Vezzi, R. Vicedomini, B. M. Vieira, J. Wang, K. C. Worley, S. Yin, S.-M. Yiu, J. Yuan, G. Zhang, H. Zhang, S. Zhou, and I. F. Korf, "Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species," *Gigascience*, vol. 2, no. 1, p. 10, Jul. 2013.

[114] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, M. N. Hamid, L. Davis, T. Dogan, V. Atalay, A. S. Rifaioglu, A. Dalkıran, R. Cetin Atalay, C. Zhang, R. L. Hurto, P. L. Freddolino, Y. Zhang, P. Bhat, F. Supek, J. M. Fernández, B. Gemovic, V. R. Perovic, R. S. Davidović, N. Sumonja, N. Veljkovic, E. Asgari, M. R. K. Mofrad, G. Profiti, C. Savojardo, P. L. Martelli, R. Casadio, F. Boecker, H. Schoof, I. Kahanda, N. Thurlby, A. C. McHardy, A. Renaux, R. Saidi, J. Gough, A. A. Freitas, M. Antczak, F. Fabris, M. N. Wass, J. Hou, J. Cheng, Z. Wang, A. E. Romero, A. Paccanaro, H. Yang, T. Goldberg, C. Zhao, L. Holm, P. Törönen, A. J. Medlar, E. Zosa, I. Borukhov, I. Novikov, A. Wilkins, O. Lichtarge, P.-H. Chi, W.-C. Tseng, M. Linial, P. W. Rose, C. Dessimoz, V. Vidulin, S. Dzeroski, I. Sillitoe, S. Das, J. G. Lees, D. T. Jones, C. Wan, D. Cozzetto, R. Fa, M. Torres, A. Warwick Vesztrocy, J. M. Rodriguez, M. L. Tress, M. Frasca, M. Notaro, G. Grossi, A. Petrini, M. Re, G. Valentini, M. Mesiti, D. B. Roche, J. Reeb, D. W. Ritchie, S. Aridhi, S. Z. Alborzi, M.-D. Devignes, D. C. E. Koo,

R. Bonneau, V. Gligorijević, M. Barot, H. Fang, S. Toppo, E. Lavezzo, M. Falda, M. Berselli, S. C. E. Tosatto, M. Carraro, D. Piovesan, H. Ur Rehman, Q. Mao, S. Zhang, S. Vucetic, G. S. Black, D. Jo, E. Suh, J. B. Dayton, D. J. Larsen, A. R. Omdahl, L. J. McGuffin, D. A. Brackenridge, P. C. Babbitt, J. M. Yunes, P. Fontana, F. Zhang, S. Zhu, R. You, Z. Zhang, S. Dai, S. Yao, W. Tian, R. Cao, C. Chandler, M. Amezola, D. Johnson, J.-M. Chang, W.-H. Liao, Y.-W. Liu, S. Pascarelli, Y. Frank, R. Hoehndorf, M. Kulmanov, I. Boudellioua, G. Politano, S. Di Carlo, A. Benso, K. Hakala, F. Ginter, F. Mehryary, S. Kaewphan, J. Björne, H. Moen, M. E. E. Tolvanen, T. Salakoski, D. Kihara, A. Jain, T. Šmuc, A. Altenhoff, A. Ben-Hur, B. Rost, S. E. Brenner, C. A. Orengo, C. J. Jeffery, G. Bosco, D. A. Hogan, M. J. Martin, C. O'Donovan, S. D. Mooney, C. S. Greene, P. Radivojac, and I. Friedberg, "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens," *Genome Biol.*, vol. 20, no. 1, p. 244, Nov. 2019.

[115] J. D. Thompson, P. Koehl, R. Ripp, and O. Poch, "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark," *Proteins*, vol. 61, no. 1, pp. 127–136, Oct. 2005.