



RESEARCH ARTICLE

REVISED Philympics 2021: Prophage Predictions Perplex

Programs

[version 2; peer review: 1 approved, 1 approved with reservations]

Michael J. Roach ¹, Katelyn McNair², Maciej Michalczyk ³, Sarah K Giles¹,
Laura K Inglis¹, Evan Pargin¹, Jakub Barylski³, Simon Roux⁴,
Przemysław Decewicz⁵, Robert A. Edwards¹

¹Flinders Accelerator for Microbiome Exploration, Flinders University, Adelaide, SA, 5042, Australia²Computational Sciences Research Center, San Diego State University, San Diego, CA, 92182, USA³Department of Molecular Virology, Institute of Experimental Biology, Faculty of Biology, Adam Mickiewicz University, 61-64 Poznan, Poland⁴DOE Joint Genome Institute, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA⁵Department of Environmental Microbiology and Biotechnology, Institute of Microbiology, Faculty of Biology, University of Warsaw, Warsaw, 02-096, Poland

V2 First published: 05 Aug 2021, 10:758
<https://doi.org/10.12688/f1000research.54449.1>
Latest published: 08 Apr 2022, 10:758
<https://doi.org/10.12688/f1000research.54449.2>

Abstract

Background

Most bacterial genomes contain integrated bacteriophages—prophages—in various states of decay. Many are active and able to excise from the genome and replicate, while others are cryptic prophages, remnants of their former selves. Over the last two decades, many computational tools have been developed to identify the prophage components of bacterial genomes, and it is a particularly active area for the application of machine learning approaches. However, progress is hindered and comparisons thwarted because there are no manually curated bacterial genomes that can be used to test new prophage prediction algorithms.

Methods

We present a library of gold-standard bacterial genomes with manually curated prophage annotations, and a computational framework to compare the predictions from different algorithms. We use this suite to compare all extant stand-alone prophage prediction algorithms and identify their strengths and weaknesses. We provide a FAIR dataset for prophage identification, and demonstrate the accuracy, precision, recall, and f_1 score from the analysis of ten different algorithms for the prediction of prophages.

Results

We identified strengths and weaknesses between the prophage prediction tools. Several tools exhibit exceptional f_1 scores, while others have better recall at the expense of more false positives. The tools vary greatly in runtime performance with few exhibiting all desirable qualities for large-scale analyses.

Open Peer Review

Approval Status

	1	2
version 2 (revision) 08 Apr 2022	 view	 view
version 1 05 Aug 2021	 view	 view

1. **Franklin Nobrega** , University of Southampton, Southampton, UK

2. **Karthik Anantharaman** , University of Wisconsin-Madison, Madison, USA

Kristopher Kieft, University of Wisconsin-Madison, Madison, USA

Any reports and responses or comments on the article can be found at the end of the article.

Conclusions

Our library of gold-standard prophage annotations and benchmarking framework provide a valuable resource for exploring strengths and weaknesses of current and future prophage annotation tools. We discuss caveats and concerns in this analysis, how those concerns may be mitigated, and avenues for future improvements. This framework will help developers identify opportunities for improvement and test updates. It will also help users in determining the tools that are best suited for their analysis.

Keywords

software comparison, bioinformatics tool, lysogen genome, temperate phage, prokaryotic virus



This article is included in the **Bioinformatics** gateway.

Corresponding author: Michael J. Roach (michael.roach@flinders.edu.au)

Author roles: **Roach MJ:** Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **McNair K:** Data Curation, Investigation, Methodology, Writing – Review & Editing; **Michalczyk M:** Investigation, Software, Writing – Review & Editing; **Giles SK:** Data Curation, Investigation, Writing – Review & Editing; **Inglis LK:** Data Curation, Investigation, Writing – Review & Editing; **Pargin E:** Data Curation, Investigation, Writing – Review & Editing; **Barylski J:** Investigation, Software, Writing – Review & Editing; **Roux S:** Investigation, Software, Writing – Review & Editing; **Decewicz P:** Data Curation, Investigation, Methodology, Resources, Software, Supervision, Writing – Review & Editing; **Edwards RA:** Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Methodology, Project Administration, Resources, Software, Supervision, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work supported by the National Institute Of Diabetes And Digestive And Kidney Diseases of the National Institutes of Health under Award Number RC2DK116713 to RAE. The support provided by Flinders University for HPC research resources is acknowledged.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2022 Roach MJ *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Roach MJ, McNair K, Michalczyk M *et al.* **Philympics 2021: Prophage Predictions Perplex Programs [version 2; peer review: 1 approved, 1 approved with reservations]** F1000Research 2022, 10:758 <https://doi.org/10.12688/f1000research.54449.2>

First published: 05 Aug 2021, 10:758 <https://doi.org/10.12688/f1000research.54449.1>

REVISED Amendments from Version 1

In this version we address the comments of the reviewers. The comparison now includes two additional programs (ProphET and Seeker), and an additional 29 manually-curated genomes to the gold-standard library; we have updated the results accordingly. We compare f_1 scores of the different genera that are present in the gold-standard library to assess how the performance of the programs may be influenced by taxonomy. The underlying data now includes our guidelines for manually curating genomes with prophage annotations.

Any further responses from the reviewers can be found at the end of the article

Introduction

Bacteriophages (phages), viruses that infect bacteria, can be either temperate or virulent. Temperate phages may integrate into their bacterial host genome. Such integrated phage genomes are referred to as prophages and may constitute as much as 20 percent of bacterial DNA (Casjens, 2003). They replicate as part of the host genomes until external conditions trigger a transition into the virulent lytic cycle, resulting in replication and packaging of phages and typically the death of the host bacteria. Prophages generally contain a set of core genes with a conserved gene order that facilitate integration into the host genome, assembly of phage structural components, replication, and lysis of the host cell (Kang et al., 2017; Canchaya et al., 2003). As well as these core genes, phages can contain an array of accessory metabolic genes that can effect significant phenotypic changes in the host bacteria (Breitbart, 2012). For instance, many prophages encode virulence factors such as toxins, or fitness factors such as nutrient uptake systems (Brüssow et al., 2004). Lastly, many prophages encode a variety of super-infection exclusion mechanisms to prevent concurrent phage infections, including restriction/modification systems, toxin/antitoxin genes, repressors, etc. reviewed in Abedon (2015, 2019). The function of most prophage accessory genes remains unknown.

Core (pro) phage genes have long been used for identifying prophage regions. However, there are other unique characteristics that can distinguish prophages from their host genomes: bacterial genomes have a GC skew that correlates with the direction of replication, and the insertion of prophages will generally disrupt this GC bias (Grigoriev, 1998). Transcript direction (Campbell, 2002) and length of prophage proteins have also proven to be useful metrics in predicting prophages (Akhter et al., 2012; Song et al., 2019), where phage genes are generally smaller and are oriented in the same direction (Dutilh et al., 2014). Likewise, gene density tends to be higher in phage genomes and intergenic space shorter (Amgarten et al., 2018; McNair et al., 2019).

Over the last two decades many prophage prediction tools have been developed, and they fall into two broad classes: (1) web-based tools where users upload a bacterial genome and retrieve prophage annotations including PHASTER (Arndt et al., 2016), Prophage Hunter (Song et al., 2019), Prophinder (Lima-Mendez et al., 2008), PhageWeb (Sousa et al., 2018), and RAST (Aziz et al., 2008); and (2) command-line tools where users download a program and database to run the predictions locally (although some of these also provide a web interface for remote execution). In this work we focus on this latter set of tools (Table 1) because web-based tools typically do not handle the large numbers of simultaneous requests required to run comparisons across many genomes.

Despite the abundance of prophage prediction algorithms, there has never been either a set of reference genomes against which all tools can be compared, nor a unified framework for comparing those tools to identify their relative strengths and weaknesses or to identify opportunities for improvement. We generated a set of manually annotated bacterial genomes released under the FAIR principles (Findable, Accessible, Interoperable, and Reusable), and developed an openly available and accessible framework to compare prophage prediction tools.

Methods

Running the tools

To assess the accuracy of the different prophage prediction tools, a set of 57 gold-standard publicly available bacterial genomes with manually curated prophage annotations was generated. We combined this with the 21 manually annotated genomes described in Casjens (2003) that were not already included for a total of 78 genomes for evaluating the bioinformatics tools. The genomes and prophage annotations currently included are available in Table S1. The genomes are in GenBank format and file conversion scripts are included in the framework to convert those files to formats used by the different software. The tools that are currently included in the framework are outlined in Table 1. Snakemake (Köster & Rahmann, 2012) pipelines utilising conda (Anaconda Software Distribution. Conda. v4.10.1, April 2021 (Conda, RRID:SCR_018317)) package manager environments were created for each tool to handle the installation of the tool and its dependencies, running of the analyses, output file conversion to a standardized format, and benchmarking of the run stage. Where possible, gene annotations from the GenBank files were used in the analysis to promote consistency between comparisons. DBSCAN-SWA was not able to consistently finish when using GenBank files as input, and

Table 1. Prophage identification tools currently included in benchmarking framework.

Tool (year)	Version	Package manager	Dependencies	Database size	Approach	Citation
Phage Finder (2006)	2.1		Aragorn, BLAST-legacy, HMMer, Infernal, MUMmer, tRNAscan-SE	93 MB	Legacy-BLAST, HMMs	(Fouts, 2006)
PhiSpy (2012)	4.2.6	conda, pip	Python3, BioPython, NumPy, SciPy	47 MB required, 733 MB optional (pVOGs)	Gene and nucleotide metrics, AT/CG skew, kmer comparison, machine learning, HMMs, annotation keywords	(Akhter et al., 2012)
VirSorter (2015)	1.0.6	conda	MCL, Muscle, BLAST+, BioPerl, HMMer, Diamond, Metagene_annotator	13 GB	Alignments, HMMs	(Roux et al., 2015)
ProphET (2019)	0.5.1		BLAST-legacy, EMBOSS, BedTools, Perl, BioPerl	41 Mb	Legacy-BLAST searches	(Reis-Cunha et al., 2019)
Phigaro (2020)	2.3.0	conda, pip	Python3, BeautifulSoup4, BioPython, bs4, HMMer, lxml, NumPy, Pandas, Plotly, Prodigal, PyYAML, shsix	1.6 GB	HMMs	(Starikova et al., 2020)
DBSCAN-SWA (2020)	2e61b95		Numpy, BioPython, Scikit-learn, Prokka	2.2 GB	Gene metrics, alignments	(Gan et al., 2020)
VIBRANT (2020)	1.2.1	conda	Python3, Prodigal, HMMer, BioPython, Pandas, Matplotlib, Seaborn, Numpy, Scikit-learn, Pickle	11 GB	HMMs (KEGG, Pfam, VOG), machine learning	(Kieft et al., 2020)
Seeker (2020)	1.0.3	pip	Python3, TensorFlow	64 kb	Machine learning (LSTM)	(Auslander et al., 2020)
PhageBoost (2021)	0.1.7	pip	Python3	13 MB	Gene and nucleotide metrics, machine learning	(Sirén et al., 2021)
VirSorter2 (2021)	2.2.1	conda	Python3, Snakemake, Scikit-learn, imbalanced-learn, Pandas, Seaborn, HMMer, Prodigal, screed	12 GB	Alignments, HMMs	(Guo et al., 2021)

instead the genome files in fasta format were used. Another pipeline was created to pool the results from each tool and some comparisons are illustrated in the included Jupyter notebook. Testing and development of the pipelines were conducted on Flinders University's DeepThought HPC infrastructure. The final benchmarking analysis was performed on a stand-alone node consisting of dual Intel® Xeon® Gold 6242R processors (40 cores, 80 threads), 768 GB of RAM, and 58 TB of disk space. Each tool was executed on all genomes in parallel (one thread per job), with no other jobs running. The only exception to this was Seeker which was run one at a time on a single core due to high memory requirements (see below).

Box 1. Benchmark metrics used in this analysis.

Accuracy was calculated as the ratio of correctly labelled genes to all CDS features from the GenBank file	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision was calculated as the ratio of correctly labelled phage CDS features to all predicted prophage CDS features	$\frac{TP}{TP+FP}$
Recall was calculated as the ratio of correctly labelled prophage CDS features to all known prophage CDS features	$\frac{TP}{TP+FN}$
The f_1 Score was calculated as the harmonic mean of Precision and Recall	$2 \times \frac{(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})}$

Accuracy provides an overall impression of correctness but is distorted by the vast difference in the numbers of prophage and non-prophage CDS features present in the genomes. The current gold-standard set includes 7,729 prophage proteins and 177,649 non-prophage proteins. Therefore, predicting everything as not coming from a prophage will result in an accuracy of 0.96. Similarly, identifying everything as coming from a prophage will result in high *Recall*, since that favours minimising false negatives. In contrast, *Precision* favours minimising false-positives and so only predicting very confident regions will result in high precision. The f_1 Score is the most suitable for comparing predictions as it gives equal weighting to both precision and recall, and thus balances the unevenness inherent in this data.

Benchmark metrics

The runtime and CPU time in seconds, peak memory usage and file write operations were captured by Snakemake (Snakemake, RRID:SCR_003475) for the steps running the prophage tools only (not for any file conversion steps before or after running each tool). The predictions were compared to the gold standard prophage annotations and the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) gene labels were used to calculate the performance metrics. Each application marks prophages slightly differently, and therefore we used the designation of coding sequence (CDS) features as phage or not to assess prophage predictions.

Adding new genomes

We developed the framework to simplify the addition of new genomes to the benchmarks. Each genome is provided in the standard GenBank format, and the prophages are marked by the inclusion of a non-standard flag for each genomic feature that indicates that it is part of a prophage. We use the qualifier *is_phage* = "1" to indicate prophage regions. Our guidelines for manually annotating prophages in bacterial genomes are available in the GitHub repository at github.com/linsalrob/ProphagePredictionComparisons/blob/master/Supplementary/prophageAnnotation.md.

Results and discussion**Software compared**

We compared the availability, installation, and results from ten different prophage prediction algorithms (Table 1). LysoPhD (Niu et al., 2019) could not be successfully installed and was not included in the current framework (see below). The remaining ten—PhiSpy (Akhter et al., 2012), Phage Finder (Fouts, 2006), VIBRANT (Kieft et al., 2020), VirSorter (Roux et al., 2015), Virsorter2 (Guo et al., 2021), Phigaro (Starikova et al., 2020), PhageBoost (Sirén et al., 2021), DBSCAN-SWA (Gan et al., 2020), ProphET (Reis-Cunha et al., 2019), and Seeker (Auslander et al., 2020)—were each used to predict the prophages in 78 different manually curated microbial genomes.

Most of these programs utilize protein sequence similarity and HMM searches of core prophage genes to identify prophage regions. PhageBoost leverages a large range of protein features (such as dipeptide and tripeptide combinations) with a trained prediction model. PhiSpy was originally designed to identify prophage regions based upon seven distinct characteristics: protein length, transcript directionality, AT and GC skew, unique phage words, phage insertion points, optionally phage protein similarity and sequence similarity. DBSCAN-SWA likewise uses a range of gene metrics and trained prediction models to identify prophages. Seeker uses a new neural network model applied to bacterial and phage reference genome sequences to classify sequences as phage or bacteria. It was intended for use with sort sequences but can be used to identify prophages.

Regardless of whether gene annotations are available, Virsorter2, Phigaro, PhageBoost, and ProphET all perform *de novo* gene prediction with Prodigal (Hyatt et al., 2010) and VirSorter uses MetaGeneAnnotator (Noguchi et al., 2008) for the same purpose. VIBRANT can take proteins if they have 'Prodigal format definition lines' but otherwise performs predictions with Prodigal. PhageBoost can take existing gene annotations but this requires additional coding by the user. DBSCAN-SWA can take gene annotations or can perform gene predictions with Prokka (Seemann, 2014). PhiSpy takes an annotated genome in GenBank format and uses the gene annotations provided.

Ease of installation

The prophage prediction packages Phigaro, PhiSpy, VIBRANT, VirSorter, and VirSorter2 are all able to be installed with conda from the Bioconda channel (Grüning et al., 2018), while Phispy, Phigaro, PhageBoost, and Seeker can be installed with pip—the Python package installer. Phigaro, VIBRANT, VirSorter, and VirSorter2 require a manual one-time setup to download their respective databases. Phigaro uses hard-coded file paths for its database installation, either to the user's home directory or to a system directory requiring root permissions. Neither option is ideal as it is impossible to have isolated versions or installations of the program, and it prevents updating the installation paths of its dependencies. For PhageBoost to be able to take existing gene annotations, a custom script was created to skip the gene prediction stage and run the program. Basic PhiSpy functionality is provided without requiring third-party databases. However, if the HMM search option is invoked, a database of phage-like proteins—e.g. pVOG (Grazziotin et al., 2017), VOGdb (<https://vogdb.org>), or PHROGS (Terzian P et al., 2021)—must be manually downloaded before it can be included in PhiSpy predictions. DBSCAN-SWA is not currently available on any package manager and must be pulled from GitHub, however all its dependencies are available via conda and it could easily be added in the future. All the above “manual” installation and setup steps are uncomplicated and are automatically executed by the Snakemake pipelines provided in the framework.

Phage Finder was last updated in 2006 and is not available on any package manager that we are aware of. The installation process is dated with the package scripts liberally utilising hard-coded file paths. The Snakemake pipeline for this package resolves this with soft links between the framework's directory to the user's home directory (where the package expects to be installed). The dependencies are available via conda allowing the complete installation and setup to be handled automatically by Snakemake.

Installing and running ProphET is a non-trivial task. It requires the unsupported BLAST legacy and EMBOSS packages and a set of Perl libraries, including a custom library for preparing the necessary GFF files for running the program. The dependencies are mostly available via Conda and the remaining required files are included in this repository. LysoPhD does not appear to be available to download anywhere and was dropped from the comparison.

Prophage prediction performance

There are many potential ways to compare prophage predictions. For instance, is it more important to capture all prophage regions or minimise false positives? Is it more important to identify all the phage-encoded genes, or the exact locations of the attachment site core duplications (*attL* and *attR*)? We explore several metrics to highlight the different strengths of each prophage prediction tool. PhiSpy, VIBRANT, Phigaro, and ProphET performed best for mean accuracy (Figure 1a; Table 2) while Seeker and DBSCAN-SWA performed the worst. PhiSpy, Phigaro, Phage Finder, VIBRANT, and ProphET performed best for mean precision (Figure 1b; Table 2). Seeker, DBSCAN-SWA, PhageBoost, VirSorter, and VirSorter2 all performed poorly for mean precision. This was mostly driven by a high false-positive rate compared to the other tools (Figure S1). VirSorter, VirSorter2, VIBRANT, PhiSpy, DBSCAN-SWA, and PhageBoost all had high mean recall scores.

Each tool balances between recall and precision. For example, the more conservative Phage Finder performed relatively well in terms of precision, making very confident predictions, but had one of the lower mean recall ratios and was not predicting prophages based on limited information. In contrast, the more speculative DBSCAN-SWA and PhageBoost both exhibited the opposite trend.

The f_1 Score is a more nuanced metric, as it requires high performance in both precision and recall. PhiSpy, VIBRANT, Phigaro, ProphET, VirSorter, and VirSorter2 all averaged above 0.5, while the remaining tools suffered from many false predictions (FP or FN) (Figure 1d; Table 2).

Lastly, we visualised f_1 scores for each genus across the tools to elucidate selection biases in the database (Figure S2a). *Escherichia* appeared to perform either very well or very poorly depending on the tool used but most genera appeared to be less variable between the different tools. We performed Mann-Whitney tests to compare the f_1 scores of each genus against the other genera (Figure S2b). The f_1 scores for *Streptococcus*, *Staphylococcus*, *Listeria*, and *Burkholderia* were significantly higher than the population average, and the f_1 scores for *Ralstonia*, *Photobacterium*, *Mycobacterium*, *Geobacter*, *Deinococcus*, *Cyanobacterium*, *Brucella*, and *Bifidobacterium* were significantly lower than the average. An explanation for some of this variation would be that *Streptococcus* prophages for instance are well studied and highly conserved (Rezaei Javan et al., 2019), whilst other genera have been studied less, or their (pro) phages are highly diverse.

Runtime performance

Many users will not be too concerned about runtime performance, for instance if they are performing a one-off analysis on a genome of interest all the tools will finish in a reasonable time. However, efficient resource utilization is an important consideration for large-scale analyses. Provisioning computing resources costs money and a well optimised tool that runs

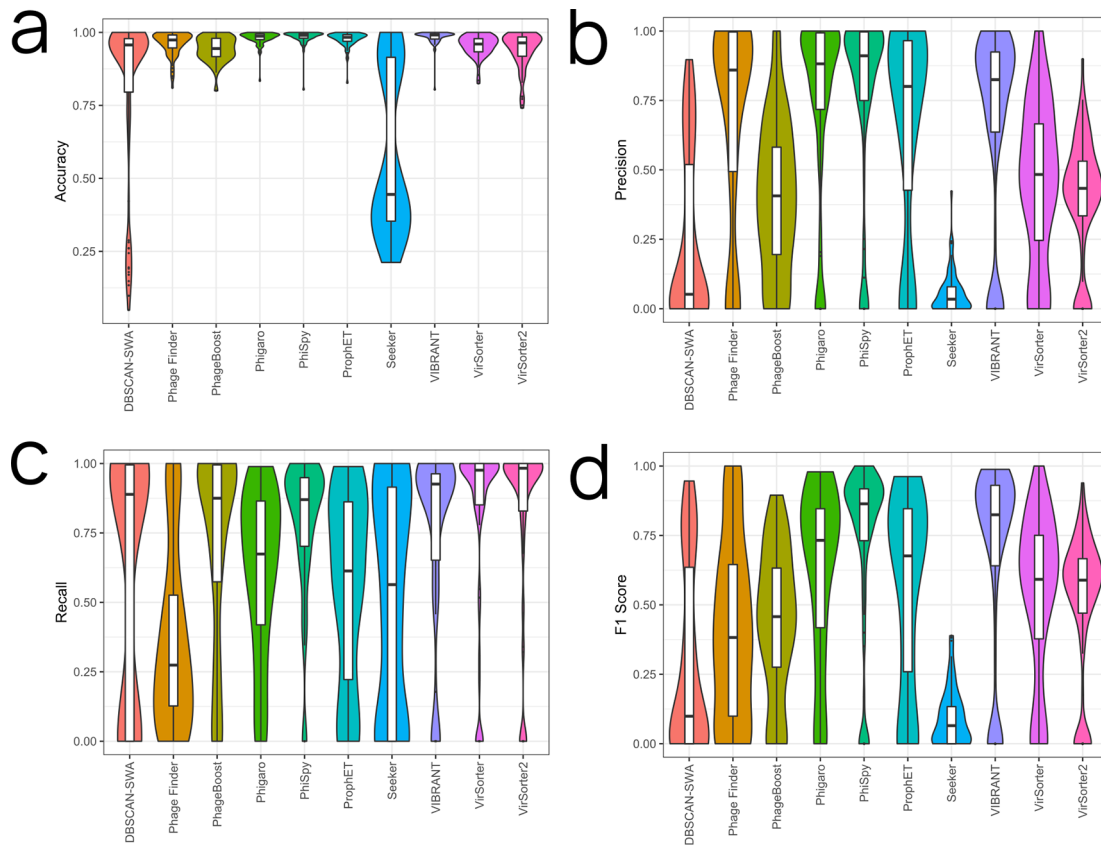


Figure 1. Prediction performance metrics for prophage callers. Violin plots for each tool are shown with individual points for each genome indicated. The graphs show: 'Accuracy' (a) as the ratio of correctly labelled genes to all genes, 'Precision' (b) as the ratio of correctly labelled phage genes to all predicted phage genes, 'Recall' (c) as the ratio of correctly labelled phage genes to all known phage genes, and 'f₁ Score' (d) as defined in the methods. For all graphs, more is generally better.

Table 2. Mean metrics for each tool as measured from our gold-standard set of genomes.

Tool	Accuracy		Precision		Recall		f ₁ score	
	Mean	sd	Mean	sd	Mean	sd	Mean	sd
DBSCAN-SWA	0.805	0.293	0.244	0.303	0.558	0.452	0.287	0.332
Phage Finder	0.962	0.0407	0.674	0.386	0.343	0.328	0.404	0.320
PhageBoost	0.942	0.0423	0.399	0.277	0.667	0.388	0.434	0.265
Phigaro	0.980	0.0232	0.748	0.337	0.566	0.323	0.611	0.320
PhiSpy	0.984	0.0246	0.772	0.330	0.731	0.309	0.733	0.306
ProphET	0.976	0.0258	0.646	0.385	0.517	0.357	0.542	0.350
Seeker	0.582	0.273	0.0584	0.0746	0.477	0.406	0.093	0.101
VIBRANT	0.983	0.0246	0.675	0.356	0.702	0.377	0.677	0.355
VirSorter	0.955	0.0365	0.451	0.288	0.762	0.373	0.532	0.299
VirSorter2	0.939	0.0623	0.399	0.219	0.766	0.372	0.508	0.257

fast translates to real-world savings. The runtime distributions across the genomes are shown for each tool in [Figure 2a](#) and [Table 3](#). The slowest prophage predictors were generally VirSorter and VirSorter2 with mean runtimes of 1,255 and 1,900 seconds respectively, except for a single DBSCAN-SWA run taking 5,718 seconds. Seeker was the fastest tool (6.65 seconds mean runtime), although this may not be a fair comparison given that multiple instances of this tool were

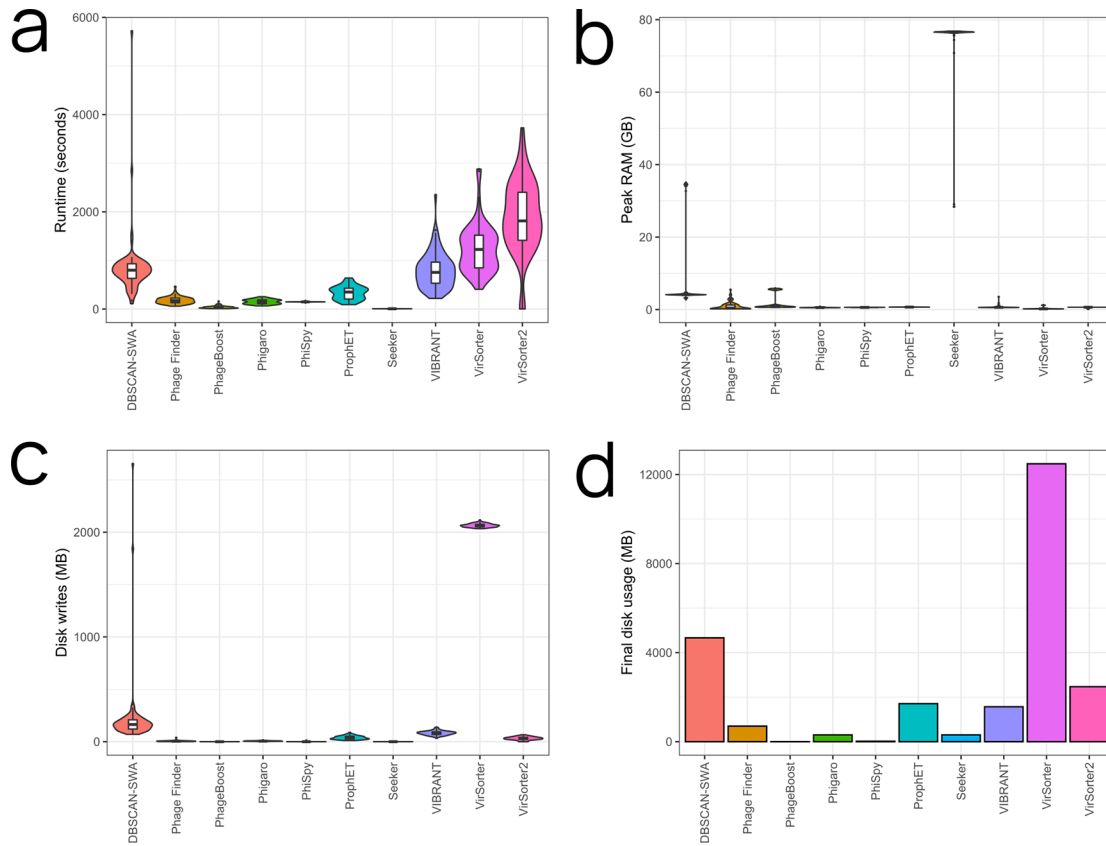


Figure 2. Runtime and peak memory usage comparison. Violin plots for each tool are shown with individual points for each genome indicated. The graphs show total runtime in seconds (a), peak memory usage in MB (b), total file writes in MB (c) and the final total disk usage (all genomes) in MB (d). For all graphs, less is better.

Table 3. Summary runtime performance metrics for each tool

Tool	Runtime (s)		Peak RAM (GB)		Disk writes (MB)		Final disk usage (MB)
	Mean	sd	Mean	sd	Mean	sd	
DBSCAN-SWA	848	653	8.76	11.0	222	342	4664
Phage Finder	180	71.6	0.932	1.07	5.39	5.21	699
PhageBoost	39.8	26.4	1.99	2.09	0.000769	0.00268	0.149
Phigaro	151	48	0.572	0.0684	6.15	2.45	305
PhiSpy	147	4.56	0.596	0.0302	0.427	2.02	20.8
ProphET	336	140	0.690	0.0105	38.3	17.1	1707
Seeker	6.65	1.45	75.2	7.63	0.019	0.00305	305
VIBRANT	798	369	0.719	0.381	82.4	20.1	1567
VirSorter	1255	500	0.287	0.229	2065	16.5	12482
VirSorter2	1900	749	0.621	0.915	31.4	15.2	2469

not able to be run at the same time (see below). PhageBoost was the next fastest (37.8 seconds mean runtime) and Phage Finder, Phigaro, PhiSpy, and ProphET all performed similarly well in terms of runtime.

Memory requirements also remain an important consideration for provisioning resources for large-scale analyses. For instance, inefficiency is encountered where the memory required by single-threaded processes exceeds the available

memory per CPU. Peak memory usage for each tool is shown in [Figure 2b](#) and [Table 3](#). Seeker had by far the highest mean peak memory usage at 75.2 GB. Our approach of running an instance of a tool on each system CPU failed for Seeker due to its extremely high peak RAM usage. We instead ran Seeker on a single CPU, one genome at a time. DBSCAN-SWA had the next highest mean peak memory of 8.76 GB. However, several runs required nearly 35 GB peak memory. Memory requirements were lowest for VirSorter with 287 MB mean peak memory, and mean peak memory usage for Phage Finder, Phigaro, PhiSpy, ProphET, VIBRANT, and VirSorter2 were all under 1 GB. Apart from Seeker and the DBSCAN-SWA outliers, there were no situations where the peak memory usage would prevent the analysis from completing on a modest personal computer. At larger-scales, Phage Finder, Phigaro, PhiSpy, ProphET, VIBRANT, VirSorter, and VirSorter2 have an advantage in terms of peak memory usage.

Another important consideration for large-scale analyses are the file sizes that are generated by the different tools. Large output file sizes can place considerable strain on storage capacities, and large numbers of read and write operations can severely impact the performance of a system or HPC cluster for all users. Total file writes for the default files (in MB, including temporary files) are shown in [Figure 2c](#) and the final disk usage for all genomes for each tool is shown in [Figure 2d](#); these are also summarised in [Table 3](#). VirSorter, DBSCAN-SWA, VIBRANT, ProphET, and VirSorter2 performed the most write operations. The other tools performed similarly well and have a clear advantage at scale as they perform far fewer disk writes. VirSorter and DBSCAN-SWA removed most of their generated files, however, the final disk usage for these tools were still the highest at 5.36 and 2.96 GB respectively. Disk usage for PhageBoost and PhiSpy was by far the lowest at 0.14 and 15 MB respectively.

Caveats

Every bioinformatics comparison involves many biases. In this comparison, PhiSpy performs well, but we developed PhiSpy and many of the gold-standard genomes were extensively used during its development to optimize the algorithm. VirSorter and VirSorter2 were primarily developed to identify viral regions in metagenomes rather than prophages in bacterial genomes—although they have been used for that e.g. in [Glickman et al. \(2020\)](#)—and filtering VirSorter and VirSorter2 hits with CheckV ([Nayfach et al., 2021](#)) is recommended. Likewise, Seeker was designed for classifying short sequences as either phage or non-phage. It was trained on phages, not prophages, and was not originally intended for classifying prophage regions in complete genomes. Furthermore, the current database of prophage annotations is heavily skewed towards more heavily studied phyla, and we show that there can be significant differences in performance depending on taxonomy. By openly providing the Prophage Prediction Comparison framework, creating a framework to install and test different software, and defining a straightforward approach to labelling prophages in GenBank files, we hope to expand our gold-standard set of genomes and mitigate many of our biases. We welcome the addition of other genomes (especially from beyond the Proteobacteria/Bacteroidetes/Firmicutes that are overrepresented in our gold-standard database).

Recent developments in alternative approaches to predict prophages, including mining phage-like genes from metagenomes and then mapping them to complete genomes ([Nayfach et al., 2021](#)) and using short-read mapping to predict prophage regions from complete bacterial genomes ([Kieft & Anantharaman, 2021](#)) have the potential to generate many more ground-truth prophage observations. However, both approaches are limited, as they identify prophages that are active, but not quiescent prophage regions. Thus, they will provide useful true positive datasets for prophage prediction algorithms but may not provide accurate true negative datasets.

Conclusions

Establishing a gold-standard dataset of prophage and associated metrics is critical to enable a robust comparison of prophage prediction tools. In particular, the current comparison suggests that most tools perform reasonably well by themselves to detect phage-like regions in complete bacterial genomes, and most of the differences between tools stem from different trade-offs between precision and recall with default parameters and different compute resource requirements. Specifically, using the gold-standard dataset and the metrics defined here, PhiSpy, VIBRANT, and Phigaro were the best performing prophage prediction tools for f_1 score. Phage Finder performs well in terms of precision at the expense of false-negatives, whereas VirSorter, VirSorter2, DBSCAN-SWA and PhageBoost perform better for recall at the expense of false-positives. Currently, Seeker, DBSCAN-SWA, VirSorter, and to a lesser extent VirSorter2 are not as well suited for large-scale identification of prophages from complete bacterial genomes when compared to the other tools, and would require using custom cutoffs and/or post-processing predictions with another tool such as CheckV. In terms of runtime performance, PhiSpy and Phigaro were among the best in every metric, and ProphET and VIBRANT performed well in most metrics. These comparisons are, however, relying on expert curation of a limited number of genomes. More genomes with manually curated prophage annotations are thus needed as well as a larger number and diversity of experimentally-verified prophages, and we anticipate that these benchmarks will change with the addition of new genomes, the addition of new tools, and as the tools are updated over time. We intentionally designed the current framework to be easily amendable and expanded, and developers are strongly encouraged to contribute by adding or

updating their tool and adding their manually curated and/or experimentally verified genomes to be included in the benchmarking. Users are strongly encouraged to check the GitHub repository for the latest results before making any decisions on which prophage prediction tool would best suit their needs.

Author contributions

RAE conceived of the study; KM and PD generated the initial gold-standard set and MJR, SKG, LI, and EP contributed to the current gold-standard set; RAE and MJR created the framework; RAE, MJR, SR, MM, and JB performed the analysis. All authors contributed to the manuscript writing.

Funding information

This work supported by the National Institute Of Diabetes And Digestive And Kidney Diseases of the National Institutes of Health under Award Number RC2DK116713 to RAE. The support provided by Flinders University for HPC research resources is acknowledged. The work conducted by the U.S. Department of Energy Joint Genome Institute (SR), a DOE Office of Science User Facility, is supported by the Office of Science of the U.S. Department of Energy under contract no. DE-AC02-05CH11231.

Competing interests

No competing interests were disclosed.

Data availability

Underlying data

Zenodo: linsalrob/ProphagePredictionComparisons: Review release. <https://doi.org/10.5281/zenodo.4739878>. (Roach & Edwards, 2021b).

This project contains the following underlying data;

- snakefiles/
 - Snakemake pipeline files for running each of the prophage prediction tools against the gold-standard prophage-annotated genomes
- rules/
 - Snakemake files with generic rules used by one or more of the Snakemake pipelines
- conda_environments/
 - Configuration files for creating conda environments for use in the Snakemake pipelines
- data/
 - Any custom small datasets required by the prophage prediction tools
- scripts/
 - Perl and Python scripts that are used in the Snakemake pipelines for performing various tasks
- ProphagePredictionsLib/
 - Library files required by the Perl and Python scripts
- jupyter_notebooks/
 - Summary metric tables for all of the tools, and example Jupyter notebook for producing the comparison figures

- `img/`
 - Example figures generated by the Jupyter notebook
- LICENCE
 - Licence file for the github repository
- Supplementary/
 - SupplementaryTables.xlsx
 - (Sheet 1) Table S1. Genomes provided in the gold-standard library with manually curated prophages
 - FigureS1.png
 - Figure S1. False positive comparison
 - FigureS2.png
 - Figure S2a. F1 scores by genus for each tool
 - Figure S2b. F1 score distribution for each genus for all tools
 - prophageAnnotation.md
 - The guidelines for manually curating prophage annotations

Underlying data and the prophage annotated GenBank files are also available at GitHub: Comparisons of multiple different prophage predictions <https://github.com/linsalrob/ProphagePredictionComparisons/tree/v0.1-beta> (Roach & Edwards, 2021a). Please note that you will need Git (git-scm.com) and Git LFS (git-lfs.github.com) to retrieve the GenBank files from the GitHub repository. Support for these files and framework pipelines are available via GitHub *issues*.

Extended data

Zenodo: Extended data for 'Philympics 2021: Prophage Predictions Perplex Programs': <https://doi.org/10.5281/zenodo.4739878>.

This project contains the following extended data:

SupplementaryTables.xlsx:

- **Table S1. Genomes provided in the gold-standard library with manually curated prophages**

FigureS1.png:

- **Figure S1. False positive comparison.** Violin plots for each tool show 'False Positives' as the number of genes incorrectly labelled prophage genes in each genome. Less is better.

FigureS2.png:

- **Figure S2a. F1 scores by genus for each tool.** F1 scores are shown for each genome grouped and coloured by genus and separated by prophage caller.

- **Figure S2b. F1 score distribution for each genus for all tools.** Boxplots of F1 scores for each genome over all tools, grouped and coloured by genus. P-values and significance are indicated for Mann-Whitney tests that were performed to determine if a genus had significantly higher or lower F1 scores when compared to the other genera.

prophageAnnotation.md

- The guidelines for manually curating prophage annotations.

References

- Abedon ST: **Bacteriophage secondary infection.** *Virologica Sinica*. 2015; **30**: 3–10.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Abedon ST: **Look Who's Talking: T-Even Phage Lysis Inhibition, the Granddaddy of Virus-Virus Intercellular Communication Research.** *Viruses*. 2019; **11**: 951.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Akhter S, Aziz RK, Edwards RA: **PhiSpy: a novel algorithm for finding prophages in bacterial genomes that combines similarity- and composition-based strategies.** *Nucleic Acids Res.* 2012; **40**: e126–e126.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Amgarten D, Braga LPP, Da Silva AM, et al.: **MARVEL, a Tool for Prediction of Bacteriophage Sequences in Metagenomic Bins.** *Front Genet.* 2018; **9**.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Arndt D, Grant JR, Marcu A, et al.: **PHASTER: a better, faster version of the PHAST phage search tool.** *Nucleic Acids Res.* 2016; **44**: W16–W21.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Aziz RK, Bartels D, Best AA, et al.: **The RAST Server: Rapid Annotations using Subsystems Technology.** *BMC Genomics*. 2008; **9**: 75.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Breitbart M: **Marine Viruses: Truth or Dare.** *Ann Rev Mar Sci.* 2012; **4**: 425–448.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Brüssow H, Canchaya C, Hardt W-D: **Phages and the Evolution of Bacterial Pathogens: from Genomic Rearrangements to Lysogenic Conversion.** *Microbiol Mol Biol Rev.* 2004; **68**: 560–602.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Campbell AM: **Preferential Orientation Preferential Orientation of Natural Lambdaoid Prophages and Bacterial Chromosome Organization.** *Theor Popul Biol.* 2002; **61**: 503–507.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Canchaya C, Proux C, Fournoux G, et al.: **Prophage Genomics.** *Microbiol Mol Biol Rev.* 2003; **67**: 238–276.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Casjens S: **Prophages and bacterial genomics: what have we learned so far?** *Mol Microbiol.* 2003; **49**: 277–300.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Dutilh BE, Cassman N, Mcnair K, et al.: **A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes.** *Nat Commun.* 2014; **5**: 4498.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Fouts DE: **Phage_Finder: Automated identification and classification of prophage regions in complete bacterial genome sequences.** *Nucleic Acids Res.* 2006; **34**: 5839–5851.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Gan R, Zhou F, Si Y, et al.: **DBSCAN-SWA: an integrated tool for rapid prophage detection and annotation.** *bioRxiv.* 2020; 2020.07.12.199018.
[Publisher Full Text](#)
- Glickman C, Kammlade SM, Hasan NA, et al.: **Characterization of integrated prophages within diverse species of clinical nontuberculous mycobacteria.** *Viol J.* 2020; **17**: 124.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Grazziotin AL, Koonin EV, Kristensen DM: **Prokaryotic Virus Orthologous Groups (pVOGs): a resource for comparative genomics and protein family annotation.** *Nucleic acids res.* 2017; **45**: D491–D498.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Grigoriev A: **Analyzing genomes with cumulative skew diagrams.** *Nucleic Acids Res.* 1998; **26**: 2286–2290.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Grüning B, Dale R, Sjödin A, et al.: **Bioconda: sustainable and comprehensive software distribution for the life sciences.** *Nat Methods.* 2018; **15**: 475–476.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Guo J, Bolduc B, Zayed AA, et al.: **VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses.** *Microbiome.* 2021; **9**: 37.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Hyatt D, Chen G-L, Locascio PF, et al.: **Prodigal: prokaryotic gene recognition and translation initiation site identification.** *BMC bioinformatics.* 2010; **11**: 119–119.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Kang HS, Mcnair K, Cuevas DA, et al.: **Prophage genomics reveals patterns in phage genome organization and replication.** *bioRxiv.* 2017: 114819.
[Publisher Full Text](#)
- Kieft K, Anantharaman K: **Deciphering active prophages from metagenomes.** *bioRxiv.* 2021: 2021.01.29.428894.
[Publisher Full Text](#)
- Kieft K, Zhou Z, Anantharaman K: **VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences.** *Microbiome.* 2020; **8**: 90.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Köster J, Rahmann S: **Snakemake—a scalable bioinformatics workflow engine.** *Bioinformatics.* 2012; **28**: 2520–2522.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Lima-Mendez G, Van Helden J, Toussaint A, et al.: **Prophinder: a computational tool for prophage prediction in prokaryotic genomes.** *Bioinformatics.* 2008; **24**: 863–865.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Mcnair K, Zhou C, Dinsdale EA, et al.: **PHANOTATE: a novel approach to gene identification in phage genomes.** *Bioinformatics.* 2019; **35**: 4537–4542.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Nayfach S, Camargo AP, Schulz F, et al.: **CheckV assesses the quality and completeness of metagenome-assembled viral genomes.** *Nat Biotechnol.* 2021; **39**: 578–585.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Niu Q, Peng S, Zhang X, et al.: **LysoPhD: predicting functional prophages in bacterial genomes from high-throughput sequencing.** 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 18–21 Nov. 2019. 2019; 1–5.
[Publisher Full Text](#)
- Noguchi H, Taniguchi T, Itoh T: **MetaGeneAnnotator: detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes.** *DNA res.* 2008; **15**: 387–396.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Reis-Cunha JL, Bartholomeu DC, Manson AL, et al.: **ProphET, prophage estimation tool: A stand-alone prophage sequence prediction tool with self-updating reference database.** *PLOS ONE.* 2019; **14**: e0223364.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Rezaei Javan R, Ramos-Sevillano E, Akter A, et al.: **Prophages and satellite prophages are widespread in Streptococcus and may play a role in pneumococcal pathogenesis.** *Nat Commun.* 2019; **10**: 4852.
- Roach MJ, Edwards RA: **linsalrob/ProphagePredictionComparisons [Online].** *GitHub.* 2021a. [Accessed].
[Reference Source](#)
- Roach MJ, Edwards RA: **linsalrob/ProphagePredictionComparisons: Review release (Version v0.1).** *Zenodo.* 2021b.

Roux S, Enault F, Hurwitz BL, *et al.*: **VirSorter: mining viral signal from microbial genomic data.** *PeerJ*. 2015; **3**: e985.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Seemann T: **Prokka: rapid prokaryotic genome annotation.** *Bioinformatics*. 2014; **30**: 2068–2069.

[PubMed Abstract](#) | [Publisher Full Text](#)

Sirén K, Millard A, Petersen B, *et al.*: **Rapid discovery of novel prophages using biological feature engineering and machine learning.** *NAR Genom Bioinform*. 2021; **3**.

[Publisher Full Text](#)

Song W, Sun H-X, Zhang C, *et al.*: **Prophage Hunter: an integrative hunting tool for active prophages.** *Nucleic Acids Res*. 2019; **47**: W74–W80.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Sousa ALD, Maués D, Lobato A, *et al.*: **PhageWeb – Web Interface for Rapid Identification and Characterization of Prophages in Bacterial Genomes.** *Fron Genet*. 2018; **9**.

[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Starikova EV, Tikhonova PO, Prianichnikov NA, *et al.*: **Phigaro: high-throughput prophage sequence annotation.** *Bioinformatics*. 2020; **36**: 3882–3884.

[PubMed Abstract](#) | [Publisher Full Text](#)

Terzian P, Olo Ndela E, Galiez C, *et al.*: **PHROG: families of prokaryotic virus proteins clustered using remote homology.** [Online]. 2021. [Accessed June 2021].

[Reference Source](#)

Open Peer Review

Current Peer Review Status:  

Version 2

Reviewer Report 23 May 2022

<https://doi.org/10.5256/f1000research.124379.r130220>

© 2022 Anantharaman K et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Karthik Anantharaman 

Department of Bacteriology, University of Wisconsin-Madison, Madison, WI, USA

Kristopher Kieft

Department of Bacteriology, University of Wisconsin-Madison, Madison, WI, USA

Our only remaining concern is still with the manual validation and curation of the gold standard dataset (i.e., Methods). We appreciate that efforts have been made to provide a GitHub repo that outlines the methods completed by the authors, but it is still lacking in detail. The guidelines provided are highly generalized guidelines and read as notes rather than methods. What we look for in methods are an ability to replicate the results or at least details of how the study was performed. Focusing on the GitHub repo, there is an abundance of ambiguous information that makes the curation of a gold standard dataset, and the cutoffs used by the authors in this publication, impossible to follow. Please provide more details of the actual methods used.

Here are some examples, though they are not limited to these:

- "Start by identifying proteins with sequence homology to known viral (phage) proteins". What ORF prediction tool was used? What annotation tool and database was used? What cutoffs were used?
- "Tracks of hypothetical proteins" How long is a 'track'? 10 proteins, 20 proteins? a specific number or example would be ideal here. The number of hypothetical annotations is also entirely dependent on the database or number of databases used to annotate.
- "Local changes in GC% content over the potential prophage region" What constitutes a change? 2%, 5%, 10% GC change? Over what size of windows?
- Once a candidate region is observed manual inspection includes searching for the presence of 'the most conserved proteins' a prophage should have or we expect to find" Is there any guidelines on this?
- "Presence of direct repeats surrounding potential prophage region (indicating potential

attachment sites)". Is there any particular upper and lower bound lengths for the repeats?

- "an additional search with the HHpred is performed". Methods require specific commands used and/or workflows, plus tool versions. Some specific examples would be good to provide.
- "Try to find nearly identical prophage region in a different bacterial genome". What is considered as nearly identical? >80%, >90% nucleotide identity?
- "Take just the parts surrounding the candidate prophage region, join them into a single continuous sequence and search NT database". What does 'just the parts' mean? Is this a nucleotide region? How long of a region?

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Metagenomics, Microbial and Phage ecology

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Reviewer Report 25 April 2022

<https://doi.org/10.5256/f1000research.124379.r130221>

© 2022 Nobrega F. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Franklin Nobrega 

School of Biological Sciences, Faculty of Environmental and Life Sciences, University of Southampton, Southampton, UK

I have no further comments to make.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Phage biology, phage-host interactions

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 31 August 2021

<https://doi.org/10.5256/f1000research.57937.r91781>

© 2021 Anantharaman K et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Karthik Anantharaman 

¹ Department of Bacteriology, University of Wisconsin-Madison, Madison, WI, USA

² Department of Bacteriology, University of Wisconsin-Madison, Madison, WI, USA

Kristopher Kieft

¹ Department of Bacteriology, University of Wisconsin-Madison, Madison, WI, USA

² Department of Bacteriology, University of Wisconsin-Madison, Madison, WI, USA

This manuscript by Roach *et al.* describes the creation and benchmarking of a gold standard bacterial dataset that can be used for prediction of prophages from bacterial genomes. Additionally, they also benchmark seven currently available software for prediction of prophages. Overall, this manuscript and the datasets provided represent a valuable resource to the community that can be used widely. We do have some comments that in our opinion can improve the manuscript, the benchmarking and the utility of the gold standard dataset that the authors aim to provide to the community.

Introduction:

- *Last paragraph:* the reasoning for this study is sound and accurate. Though the usage of the term “annotated” is slightly ambiguous here, and throughout the paper, because it is meant to designate either prophage locations or protein/gene annotations.

Methods:

First sentence: the actual method of “manually curated prophage annotations was generated” is never explained.

- What method was used? How were prophages manually identified, annotated, and validated? To me, gold standard would imply each prophage has been experimentally validated.
- Was there is specific method by which certain hosts were selected? I can see that there is “*Escherichia coli*_O157-H7” and “*Escherichia coli*_O157-H7_EDL933”. Why were so similar hosts chosen? Do they have significantly different prophages? There are multiple examples of this.
- Many also appear to be common model organisms. How was diversity ensured? Is there a possibility to use organisms more widely distributed across the tree of life?
- If this component is at all incorrect then the performance metrics, especially false positives

if the prophage boundaries are wrong, will be biased.

- How was Supplemental Table 2 generated?

Overall, there is no indication as to how the gold standard dataset, the centerpiece of the paper, was actually generated. This does not significantly affect the study's results, but this information needs to be included before publication. For example, were the host sequences chosen at random from a database? Represent variable phylogenetic backgrounds? Source environments? Was prophage phylogeny taken into account? There are certainly questions left unanswered.

Results:

- *Prophage prediction performance*: rather than only providing subjective "best" and "worst" designations, the numerical results should be provided too (e.g., accuracy of 0.9 +/- 0.1). Furthermore, what statistical metrics were used to designate "best" and "worst"?
- *Caveats*: The VIBRANT manuscript says that it also was developed to identify both temperate and virulent viral regions in metagenomes. FYI - we were not able to access the Supplementary datasets through the paper (but were able to retrieve them from bioRxiv).

Suggestions:

Overall, this comparison framework would benefit from simple, minor additions.

- Does the specific host (e.g., taxonomy) affect the results of prophage prediction for some tools? For tools that utilize HMM searches, the HMM databases may be biased towards certain groups (e.g., *E. coli* phages). We suggest that a comparison of precision/recall be compared to the source host.
- Do the performance comparisons only take into account total prophage CDS predictions or also the completeness of predicted prophages? For example, identifying all prophages but only 50% of each of those prophages is different than identifying half of all prophages but 100% of each of those.
- Is there any effect on hosts with multiple prophages? Are some tools affected by this?
- Some tools will have predicted prophages that were not in the gold standard set (false positives). What measures were taken to ensure that none of these are real prophages that were missed within the manual curation?

Is the work clearly and accurately presented and does it cite the current literature?

Yes

Is the study design appropriate and is the work technically sound?

Partly

Are sufficient details of methods and analysis provided to allow replication by others?

Partly

If applicable, is the statistical analysis and its interpretation appropriate?

Yes

Are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions drawn adequately supported by the results?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Metagenomics, Microbial and Phage ecology

We confirm that we have read this submission and believe that we have an appropriate level of expertise to state that we do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 05 Apr 2022

Michael Roach

Thank you so much for reviewing our manuscript. We hope that version 2 address the concerns with the manuscript which we outline below:

- *Last paragraph: the reasoning for this study is sound and accurate. Though the usage of the term "annotated" is slightly ambiguous here, and throughout the paper, because it is meant to designate either prophage locations or protein/gene annotations.*
- **Response:** We agree that this could be clearer. We have updated all instances of 'annotations' in the manuscript to explicitly refer to either prophage or gene annotations.
- *Methods: First sentence: the actual method of "manually curated prophage annotations was generated" is never explained. What method was used? How were prophages manually identified, annotated, and validated? To me, gold standard would imply each prophage has been experimentally validated.*
- **Response:** Experimentally validating all prophages by inducing them is not possible currently, as we do not know the signal for triggering the lytic cycles for all the prophages we know about. It is also not possible for elements such as cryptic phages, which may still offer a strong prophage signal and represent true phage genome sequence, as they are not able to be induced. For cryptic prophages we believe it is still important to include these predictions as prophage annotations. The next best approach to a gold-standard library is to manually inspect and annotate the genomes. We have included the guidelines we use as supplementary material.
- *Was there is specific method by which certain hosts were selected? I can see that there is "Escherichia_coli_O157-H7" and "Escherichia_coli_O157-H7_EDL933". Why were so similar hosts chosen? Do they have significantly different prophages? There are multiple examples of this. Many also appear to be common model organisms. How was diversity ensured? Is there a possibility to use organisms more widely distributed across the tree of life?*
- **Response:** For PhiSpy's original development it was important to include multiple

similar genomes with dissimilar numbers and positions of phages for training the algorithm. This selection bias is something that we have more recently been trying to address. We have earmarked many new diverse genomes for manual curation, and this update marks the inclusion of 10 new prophage annotated genomes from under-represented phyla. There is still a long way to go. Annotating prophages remains an extremely challenging task for underrepresented bacterial phyla, and it will remain so until our knowledgebase of known phages and phage proteins for these phyla improves.

- *If this component is at all incorrect then the performance metrics, especially false positives if the prophage boundaries are wrong, will be biased.*
- **Response:** The current state is not perfect, rather, it is the best we can do right now. We agree with the sentiment, and it is why we have designed the repository around making it easy to add and refine tools, genomes and prophage annotations over time. However, we don't believe there are enough errors to significantly affect the outcome of the evaluation.
- *How was Supplemental Table 2 generated?*
- **Response:** It was originally compiled during manual curation. We now include a script for generating this table from the prophage-annotated GenBank files, as well as an updated table to include the new genomes.
- *Overall, there is no indication as to how the gold standard dataset, the centerpiece of the paper, was actually generated. This does not significantly affect the study's results, but this information needs to be included before publication. For example, were the host sequences chosen at random from a database? Represent variable phylogenetic backgrounds? Source environments? Was prophage phylogeny taken into account? There are certainly questions left unanswered.*
- **Response:** We hope the inclusion of our guidelines for manual curation and our recent progress has alleviated these concerns. There is still a long way to go to achieve a more diverse representation of prophages and this framework is intended to support this journey.
- *Results: Prophage prediction performance: rather than only providing subjective "best" and "worst" designations, the numerical results should be provided too (e.g., accuracy of 0.9 +/- 0.1). Furthermore, what statistical metrics were used to designate "best" and "worst"?*
- **Response:** This was partially available in Supp table 3, but given its importance, we have move this to a new table (Table 2) and include a new table (Table 3) for the benchmarking results. We have also calculated and report standard deviations where applicable. "Best" and "Worst" are simply designated based on the rankings of each prophage caller based on their scores for a given metric.
- *Caveats: The VIBRANT manuscript says that it also was developed to identify both temperate and virulent viral regions in metagenomes. FYI - we were not able to access the Supplementary datasets through the paper (but were able to retrieve them from bioRxiv).*

- **Response:** Our apologies. The Genbank files are stored and can be retrieved using git-lfs. The Zenodo repo is a verbatim copy of the GitHub repo and does not resolve the git-lfs links to the GenBank files. We have added a section in underlying data to clarify this. We have also made sure there are instructions in the GitHub readme and genbank subfolder readme for syncing the GenBank files with git-lfs.
- *Suggestions: Overall, this comparison framework would benefit from simple, minor additions. Does the specific host (e.g., taxonomy) affect the results of prophage prediction for some tools? For tools that utilize HMM searches, the HMM databases may be biased towards certain groups (e.g., E. coli phages). We suggest that a comparison of precision/recall be compared to the source host.*
- **Response:** This is absolutely a consideration for every program and comes back to the problem of selection bias in the database. It's fundamentally not something we can solve in this update of the framework. We compared f1 scores for each tool against genus and the f1 scores of each genus across the population averages (see results and Figure S2a/b).
- *Do the performance comparisons only take into account total prophage CDS predictions or also the completeness of predicted prophages? For example, identifying all prophages but only 50% of each of those prophages is different than identifying half of all prophages but 100% of each of those.*
- **Response:** The performance comparisons currently only compare the numbers of correctly labelled genes. If a user were manually curating their predictions, then it could be argued that partially capturing all prophages would be better than completely capturing half of the prophages. This is something we might be able to add in the future if there is enough interest.
- *Is there any effect on hosts with multiple prophages? Are some tools affected by this?*
- **Response:** The number of prophages shouldn't impact the performance of any of the tools but this is not something we've looked at specifically. Most genomes only have 1 or a few prophages and it would be difficult to draw any conclusions without more examples at the fringes (0 prophages or say more than 6 or so). It is certainly something to consider as the dataset grows.
- *Some tools will have predicted prophages that were not in the gold standard set (false positives). What measures were taken to ensure that none of these are real prophages that were missed within the manual curation?*
- **Response:** The genomes are small enough that the entire genomes are thoroughly examined during manual curation. We don't anticipate enough errors to significantly affect the outcome of the evaluation. Nevertheless, we do anticipate that some corrections will need to be made over time, be that from missed prophages or incorrect prophage boundaries and we welcome feedback from the community about the prophage annotations.

Competing Interests: The authors declare that there are no conflicts of interest.

Reviewer Report 23 August 2021

<https://doi.org/10.5256/f1000research.57937.r91324>

© 2021 Nobrega F. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Franklin Nobrega

¹ School of Biological Sciences, Faculty of Environmental and Life Sciences, University of Southampton, Southampton, UK

² School of Biological Sciences, Faculty of Environmental and Life Sciences, University of Southampton, Southampton, UK

The manuscript by Roach and co-authors focuses on a crucial question on how to accurately determine prophage regions. With the increasing accessibility of HPC infrastructure by scholars, and the increasing number of open datasets available for mining, there is an urgent need to establish FAIR datasets to set a ground-truth for data analysis. This work sets the tone so the community can move away from using a favourite tool, to using the most accurate and reliable to address the question at hand. In particular, the tool performance part is excellently achieved, providing details for both the novice and the advanced user when considering a new tool for their pipeline. In sum, it was a pleasure to read this well-structured and well-balanced work, which I fully endorse. I would like to leave the authors with a few recommendations and suggestions.

Recommendations:

1. I believe the authors could use a more recent reference for Calendar, 1988 (Introduction).
2. Please provide the methods used for manual curation of prophage annotations, as this will contribute to increase the gold-standard genomes available.
3. What was the rationale for running PhiSpy using all modes, and not doing the same for other tools that also have different run modes? This would make for a more fair comparison.
4. Could the authors clarify which prophage prediction categories were considered for VirSorter and Virsorter2, as these will certainly affect the accuracy and recall results obtained. Similar comment for VIBRANT.
5. Data would benefit from statistical analysis.

Suggestions:

1. The first three lines of "Benchmark metrics" (Methods) seem to be more adequate to the Results section.
2. I would suggest that the authors summarize the information provided in "Software compared" (Results and discussion) as a supplementary table, since this will certainly be useful to the community.

Is the work clearly and accurately presented and does it cite the current literature?

Yes

Is the study design appropriate and is the work technically sound?

Yes

Are sufficient details of methods and analysis provided to allow replication by others?

Yes

If applicable, is the statistical analysis and its interpretation appropriate?

No

Are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions drawn adequately supported by the results?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Phage biology, phage-host interactions

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 05 Apr 2022

Michael Roach

Thank you so much for reviewing our manuscript. We hope that version 2 addresses the concerns with the manuscript which we outline below:

- *I believe the authors could use a more recent reference for Calendar, 1988 (Introduction).*
- **Response:** We have updated this section with more recent references.
- *Please provide the methods used for manual curation of prophage annotations, as this will contribute to increase the gold-standard genomes available.*
- **Response:** Guidelines are now included in additional data.
- *What was the rationale for running PhiSpy using all modes, and not doing the same for other tools that also have different run modes? This would make for a more fair comparison.*
- **Response:** We developed PhiSpy and are well familiar with the different ways of running the pipeline. To keep the comparison fair, we now only present running PhiSpy with default parameters.

- *Could the authors clarify which prophage prediction categories were considered for VirSorter and Virsorter2, as these will certainly affect the accuracy and recall results obtained. Similar comment for VIBRANT.*
- **Response:** Following valuable contributions and input from SR—now a coauthor on this paper—the Virsorter categories 1-5 are taken as prophage predictions. Virsorter2 is run with --high-confidence-only --exclude-lt2gene and we accept both predicted whole phages and integrated phage genomes as prophage predictions. VIBRANT is run with default parameters and we use all predictions from the integrated_prophage_coordinates output file.
- *Data would benefit from statistical analysis.*
- **Response:** We now include some statistical analysis as part of our database bias evaluation.
- *Suggestions: The first three lines of “Benchmark metrics” (Methods) seem to be more adequate to the Results section.*
- **Response:** We agree, the sentence has been moved to the results section.
- *I would suggest that the authors summarize the information provided in “Software compared” (Results and discussion) as a supplementary table, since this will certainly be useful to the community.*
- **Response:** We considered adding a more simplified summary table, such as ticks and crosses or 5-star ratings for various features, however these can be very subjective and would simply be our interpretation complete with our personal biases of the results.

Competing Interests: The authors declare that there are no conflicts of interest.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research