

RESEARCH ARTICLE

Galaxy and Apollo as a biologist-friendly interface for high-quality cooperative phage genome annotation

Jolene Ramsey^{1,2}, Helena Rasche^{1,2}[✉], Cory Maughmer^{1,2}, Anthony Criscione^{1,2}, Eleni Mijalis^{1,2}[✉], Mei Liu^{1,2}, James C. Hu^{1,2}[†], Ry Young^{1,2}, Jason J. Gill^{1,3}^{*}

1 Center for Phage Technology, Texas A&M University, College Station, Texas, United States of America, **2** Department of Biochemistry and Biophysics, Texas A&M University, College Station, Texas, United States of America, **3** Department of Animal Science, Texas A&M University, College Station, Texas, United States of America

 These authors contributed equally to this work.

[†] Deceased.

[✉] Current address: Erasmus Medical Center, Clinical Bioinformatics group, Department of Pathology, Rotterdam, the Netherlands.

[✉] Current address: Louisiana State University, School of Medicine, Shreveport, Louisiana, United States of America.

* jason.gill@tamu.edu


 OPEN ACCESS

Citation: Ramsey J, Rasche H, Maughmer C, Criscione A, Mijalis E, Liu M, et al. (2020) Galaxy and Apollo as a biologist-friendly interface for high-quality cooperative phage genome annotation. *PLoS Comput Biol* 16(11): e1008214. <https://doi.org/10.1371/journal.pcbi.1008214>

Editor: Mihaela Pertea, Johns Hopkins University, UNITED STATES

Received: May 15, 2020

Accepted: August 2, 2020

Published: November 2, 2020

Copyright: © 2020 Ramsey et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All new source code described here is accessible at GitHub under open source GPLv3 license (<https://github.com/TAMU-CPT/galaxy-tools>). The galaxy-tools repository readme describes intended use. Users can access the Galaxy and Apollo instances at <https://cpt.tamu.edu/galaxy-pub>, where a free user account will allow saving of data. Much of the data resulting from use of the annotation pipelines described here are continuously collated on our BioProject page (<https://www.ncbi.nlm.nih.gov/bioproject/?term=PRJNA222858>), which publicly displays

Abstract

In the modern genomic era, scientists without extensive bioinformatic training need to apply high-power computational analyses to critical tasks like phage genome annotation. At the Center for Phage Technology (CPT), we developed a suite of phage-oriented tools housed in open, user-friendly web-based interfaces. A Galaxy platform conducts computationally intensive analyses and Apollo, a collaborative genome annotation editor, visualizes the results of these analyses. The collection includes open source applications such as the BLAST+ suite, InterProScan, and several gene callers, as well as unique tools developed at the CPT that allow maximum user flexibility. We describe in detail programs for finding Shine-Dalgarno sequences, resources used for confident identification of lysis genes such as spanins, and methods used for identifying interrupted genes that contain frameshifts or introns. At the CPT, genome annotation is separated into two robust segments that are facilitated through the automated execution of many tools chained together in an operation called a workflow. First, the structural annotation workflow results in gene and other feature calls. This is followed by a functional annotation workflow that combines sequence comparisons and conserved domain searching, which is contextualized to allow integrated evidence assessment in functional prediction. Finally, we describe a workflow used for comparative genomics. Using this multi-purpose platform enables researchers to easily and accurately annotate an entire phage genome. The portal can be accessed at <https://cpt.tamu.edu/galaxy-pub> with accompanying user training material.

accessions for linked genome records, raw data, and published results.

Funding: This work was supported by National Science Foundation (<https://nsf.gov>) awards EF-0949351 and DBI-1565146 to J.C.H., R.Y., and J.J.G., and by support from Texas AgriLife Research and Texas A&M University. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

In the modern genomic era, scientists without extensive bioinformatic training need to apply advanced computational analyses to genome annotation. At the Center for Phage Technology, we use two open source, web-based platforms: Galaxy, for reproducible computational analyses, and Apollo, a collaborative genome annotation editor, to facilitate annotation of phage genomes. Unlike the genomes of most cellular life forms, phage genomes are usually a single contiguous molecule <200,000 bases in length. Their size allows high standards for complete, evidence-based annotations, and is amenable to genomics education settings. The Galaxy and Apollo system described here is used for original biological research and development of new bioinformatic tools to analyze many individual phage genomes, as well as clusters of related phages. Our robust suite of phage-oriented tools moves all aspects of control and choice into the user's court. In comparison to widely-used automated and fast command-line annotation methods, our integrated and flexible approach benefits from trained human intervention to result in high-quality final annotations. We have educated a steady stream of scientists, including both undergraduate and graduate students, informally and through formal university course offerings using Galaxy-Apollo to annotate phage genomes, deposit the data in public sequence repositories, and publish the results.

This is a *PLOS Computational Biology* Software paper.

Introduction

Bacteriophage, or phage, are the viruses of bacteria. Their study cracked open critical concepts in genetics, and allowed detailed gene mapping before genome maps could be generated with ease [1,2]. While phage genomes were the first to be sequenced in their entirety, phage research declined considerably before sequencing technologies took off. Researchers from disparate fields in the modern age have come to a new appreciation for the potential that phage have to help solve current problems, as well as the commensurate challenges facing their application [3]. Scientists around the world are collecting phages for a diverse panel of bacterial hosts that are relevant for the clinic, in industry, and as model organisms. One stated intent is to establish organized repositories, or phage banks, as a community resource for their distribution [4,5]. Coupling this with a surge in use of phage for education and research training, an incredible boom of phage sequencing has also emerged, with great promise for extending our understanding of fundamental phage biology [6].

The great sequencing explosion has resulted in many new viral [7] and phage genomes being deposited into online databases. Because these genomes are small, we can have high standards for complete, confirmed contigs, and high-quality structural and functional annotation. Unfortunately, many of the tools available to accomplish this task are command-line based. When tools available for annotation run on the command-line, they are not accessible to many biologists. Even if they are, the output is not visual, integrated, or contextualized, and requires much outside analysis. NCBI has released a suite of command-line interface tools for (eukaryotic) virus annotation recently [8], and new phage-specific annotation pipelines are command-line based making teamwork less fluid [9]. While the capability to do the needed

analyses is there, it is still out of reach for many teams. A great need in the field is therefore to have easy-to-use, web-based tools with a graphical user interface for annotation and community annotation platforms to improve quality and promote shared input on phage assessment for use in any given application. Because phages are being rushed into application in humans for phage therapy without measured and thorough safety checks put in place, many others agree that manual inspection for high-quality annotation is needed [10]. The Center for Phage Technology (CPT) at Texas A&M University has harnessed two powerful online platforms to solve this problem.

Methods

Galaxy & Apollo at the CPT

The Galaxy Project is a web platform suitable for beginner and advanced biologists to perform analyses on biological sequence data [11]. Apollo is a browser-based, evidence-driven, and community-focused genome annotation editor based on the popular JBrowse genome viewer [12,13]. The CPT created a Galaxy-Apollo bridge to link the powerful sequence analysis capabilities of Galaxy with the evidence-based community annotation platform of Apollo to provide a complete environment for the analysis and annotation of phage genomes (Fig 1) [14]. To date, more than 120 phage genomes have been annotated using this platform and publicly deposited (see NCBI BioProject PRJNA222858). Documentation on the CPT’s Galaxy-Apollo phage annotation pipeline, written at a level suitable for teaching undergraduates, accompanies the CPT resources at <https://cpt.tamu.edu/training-material/>, with new tutorials added regularly.

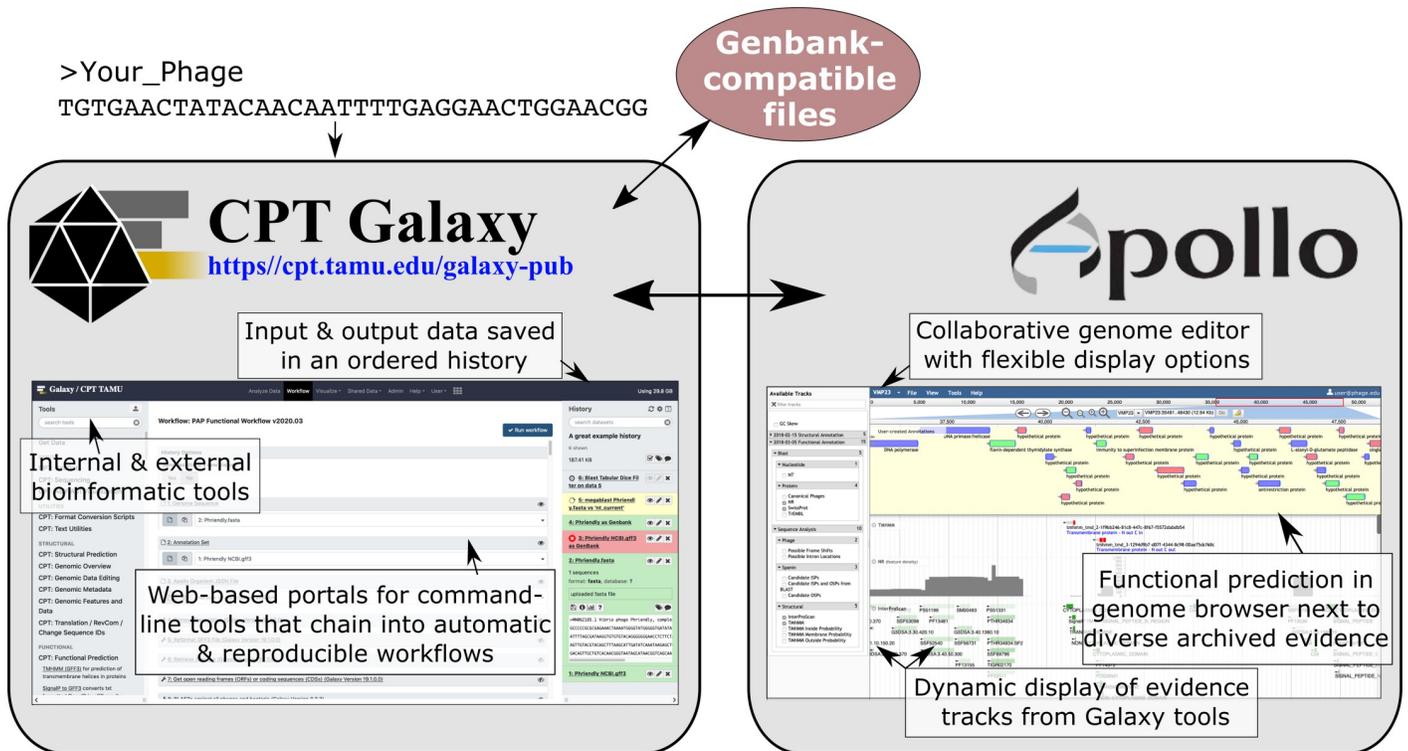


Fig 1. The Galaxy and Web Apollo interface used for analyses and annotation. By coupling the Galaxy platform for analyses with the editing capabilities in Apollo, contextualized evidence can be used to iteratively annotate genomes as a team effort.

<https://doi.org/10.1371/journal.pcbi.1008214.g001>

CPT Galaxy, a flexible web-based platform for computational analyses

The major benefit of Galaxy as a bioinformatics platform is that it allows traditionally-trained biologists to perform, record, and visualize complex analyses on their own data within a browser-based graphical user interface, rather than at the command-line. Researchers spend their time analyzing the biological implications of results, rather than troubleshooting the execution of scripts or managing files at the command-line. Crucially, the browser-based interface makes the system accessible to the common scientist. A second powerful feature of Galaxy is the ability to chain tools together in workflows that run jobs (the equivalent of a computational experiment) automatically, which makes the system convenient for programmers and bench scientists alike. Local or worldwide project administrators maintain the system operation, update tools, and provide user support.

Detailed descriptions of Galaxy functionality are provided by the developers, and here we give only a few highlights pertinent to our discussion on its use for phage annotation [11]. The user interface in Galaxy revolves around three main panels where the work is ordered (Fig 1). Tools, or bioinformatic scripts, for data analysis are cataloged and searchable in the left panel. The center Data Analysis panel provides a point-and-click atmosphere for setting job parameters and viewing data. The right-hand panel visualizes the stored history of each user analysis, which enables good documentation and promotes reproducibility. Galaxy histories, akin to a digital notebook, can be shared among users.

The standard Galaxy installation comes with tool sets contributed by the scientific community, and can be expanded through tools available in the ToolShed, by adapting open-source tools for the Galaxy interface, or by writing new scripts to suit specialized analysis needs [15–17]. The CPT Galaxy instance offers various phage-specific tools described below and a comprehensive selection of curated BLAST databases, including SwissProt, NCBI nt and nr, as well as certain custom databases [18]. A common challenge in using a large collection of tools from different sources is the interoperability of the outputs. To facilitate moving data output from one tool into the next, we have also gathered, adapted or produced a comprehensive set of scripts that convert or parse various tool outputs into standardized formats, including conversions between tab-separated value (tsv), XML, Genbank, and GFF3 formats (see S1 Table for partial list, search at <https://cpt.tamu.edu/galaxy-pub> for full list).

Apollo at the CPT for teaching and crowd-sourcing advanced phage genome annotation

JBrowse is a widely-used, embeddable genome browser, capable of displaying genome annotations and features from the level of the entire chromosome down to the DNA sequence [19]. Apollo is a web-based genome editor that builds upon JBrowse to provide the ability to do persistent, manual genome annotation and curation inside an internet browser window. Multiple users can view and/or edit the same organism, resulting in Apollo being colloquially referred to as the ‘Google Docs’ for genome annotation. Whole genome and feature analysis results conducted in Galaxy are displayed in tracks below their corresponding locations in the genome, allowing the user to integrate multiple types of evidence within genomic context when making annotations. The centralization of evidence, especially within genomic context, makes the view comprehensive and spares users from mentally integrating evidence in disparate formats from multiple sources, a key achievement of the system. A virtually unlimited number of evidence tracks can be added, dictated primarily by the user’s need for, and access to, additional analysis types. Various forms of metadata can be added in both standard and free-form entry methods, allowing rich annotation of each feature within a genome. As

needed, additional evidence from up-to-date analyses in Galaxy can be added to the Apollo organism evidence tracks through the Galaxy-Apollo bridge.

The Galaxy-Apollo bridge

To meet the needs of reproducible phage annotation and scaling these processes for the influx of new phage genomes, the CPT developed a Galaxy-Apollo bridge. The bridge consists of a number of Galaxy tools which interface directly with Apollo, enabling Galaxy users to move data into Apollo, automate certain tasks within Apollo, and finally to extract user-curated annotations back into Galaxy for further downstream analysis.

During the annotation process, a user will create a JBrowse instance within Galaxy, combining the genome and analysis results for visualization. The **Create or Update Organism** tool sends this data to Apollo either as a newly created organism in the Apollo system, or as an update to an existing one. The **Annotate** tool uses the output of the Create or Update step to provide an iframe window into Apollo, permitting users to retain the context of their Galaxy analysis history and tools, while allowing users to easily switch back to interactive analysis. When manual curation is complete, users can run **Retrieve Data**, to fetch the data from Apollo's 'User-created annotations' track into Galaxy. Tools including **GFF3 to Apollo Annotations** and **Delete all annotations from an Apollo record** enable significantly more automation between Galaxy and Apollo, and future updates are working toward entirely automated annotation pipelines. This set of tools bridges the two different worlds of Galaxy and Apollo and tells one cohesive story: from DNA, to an annotated genome, with the reproducibility of Galaxy and the freedom and real-time collaboration of Apollo.

These tools inspired the Galaxy community to start the Galaxy Genome Annotation project (GGA; <https://github.com/galaxy-genome-annotation/>), resulting in the development of a new Python library for interacting with Apollo, a command-line suite for Apollo, and a new set of Galaxy tools generated from the Python library.

Documentation available for new users

To promote widespread adoption by biologists, step-by-step tutorials with background for using Galaxy tools and the Apollo platform are freely available at the CPT website: <https://cpt.tamu.edu/training-material/>. Their format is based on resources provided by the Galaxy Training Network [20]. Tutorials are written at levels suitable for use by undergraduates learning how to annotate phage genomes. The topics covered make them useful for the scientist as well, including tutorials on phage genome assembly from Illumina reads and instructions for polishing and depositing genomes in the NCBI Genbank database.

Galaxy tools useful for phage annotation

The customization of the CPT Galaxy instance is best understood as a unified collection of tools that address the specific needs for phage genome annotation; these tools are available for ad hoc usage or as part of a workflow within Galaxy. The challenges associated with phage genome annotation are inextricably linked with general prokaryotic annotation, and intrinsically different from eukaryotic systems. Since intron splicing is rare in prokaryotes and phages, and thus is not assumed *a priori*, tools in the CPT Galaxy are tailored to take advantage of the fact that genes and protein-coding sequences can be called directly from the DNA sequence. Furthermore, many features of the pipeline described below could also be useful for annotating bacterial genomes, although the larger size of bacterial genomes would benefit from increased automation in certain procedures. Below, we describe the development of resources motivated by gaps we experienced within the available annotation tools and platforms.

Quality control of gene calling using ShineFind

All phages rely on host machinery for translation, and as such their genes typically have recognizable prokaryotic signals. The Shine-Dalgarno (S-D) sequence is a component of the ribosome binding site (RBS) situated upstream of the start codon in prokaryotic genes, and its presence can serve as a strong indicator of gene starts. However, gene callers with high-quality algorithms do not universally include the ribosomal binding site when predicting the locations of protein-coding genes. We wrote a stand-alone tool called **ShineFind** to annotate S-D sequences as part of the standard gene model in Apollo. This Python script accepts GFF3-formatted gene calls and its associated FASTA DNA sequence file as inputs. ShineFind then extracts the upstream sequence for all CDS features that do not already have an RBS in their gene model; this defaults to extraction of upstream nucleotides 3–24, but is editable by the user at runtime. Those upstream sequences are then searched for matches to the *E. coli* consensus S-D sequence AGGAGGT, or smaller subsets of this sequence (see full list in [S2 Table](#)). By default, the longest match to the consensus is returned, but all the smaller matches can also be output. The returned S-D match is added to the GFF3 gene model as a child of the gene feature and named “Shine_Dalgarno_sequence” (a sequence ontology term [21,22]), and can then be displayed within Apollo. The consensus S-D sequence and recognized subset sequences can be swapped for an alternate sequence set by editing the Python script. GFF3 annotations that already contain S-D annotations can be stripped of these features using the tool **GFF3 Feature Type Filter**, which is also part of the Structural Workflow described below.

Finding candidate spanin genes

There are special cases among phage genes, such as the lysis genes, which are difficult to detect by standard methods relying on simple sequence similarity. The spanin genes needed for outer membrane disruption during phage lysis of Gram-negative bacterial hosts are often missed or mis-annotated, largely due to their unique genetic architecture in which one gene is often embedded in an alternate reading frame within the sequence of another [23]. Spanins can function as a single or two-protein system, but the proteins share the same characteristics. The inner membrane spanin (i-spanin) contains a transmembrane domain, whereas the outer membrane spanin (o-spanin) will be cleaved and lipoylated by the bacterial lipoprotein processing system at a Cys residue at the end of a lipobox motif [24,25]. Their varied genetic architectures, particularly when the o-spanin gene is fully embedded within the i-spanin gene, and clear domain signatures are keys to the suite of tools that we have developed to find them [26].

To account for the fact that spanin genes are highly prone to be uncalled or called with incorrect start sites, spanin-finding begins with naïve ORF calling by the tool **Get open reading frames (ORFs) or coding sequences (CDSs)** [16,17] finding all possible ORFs (using the NCBI translation table 11 for Bacteria and a 30 aa minimum length cutoff), and generating an output in GFF3 format. The gene model within the GFF3 file is corrected by the addition of a gene parent feature to every CDS. The ORFs are then filtered to include only those with a common phage start codon (ATG, GTG or TTG) and ShineFind is used to add potential Shine-Dalgarno sites where they exist.

From there, the naïve gene sequences are translated and analyzed in three ways. 1) Protein similarity to spanins in a curated database [23] is determined by BLASTp with a 0.001 expectation value cutoff. 2) Potential lipobox motifs expected for outer membrane spanins are identified using the tools **LipoP** [27] and **Identify Lipoboxes**, a less stringent regex amino acid motif search for the four-residue motif [ILMFTV]-[any residue except REKD]-[GAS]-[C], or [A]-[W]-[AGS]-[C]. 3) Inner-membrane spanin candidates are identified through **TMHMM**, which predicts transmembrane regions [28]. Finally, to reduce noise, the results from Lipobox/LipoP

and TMHMM predictions are cross filtered to require putative inner- and outer-membrane spanin gene pairs occur within 50 nucleotides of each other and be located on the same strand, consistent with the architecture of known spanin genes [23]. Analysis using both explicit protein similarity by BLASTp and searching for more general spanin signatures allows for the identification of all spanin types and high-quality selection of likely start sites. This approach was able to identify a test set of experimentally verified spanins. However, the output needs to be critically assessed before genes are confidently assigned spanin functional predictions; the signature-based approach of identifying genes that pair with nearby genes containing lipoprotein or TMD signals is prone to producing multiple false-positive results in the average phage genome, as it ignores genomic context and the presence of other genes. These false-positive results can typically be excluded upon cursory examination of the results due to their presence embedded within other, obviously non-spanin genes. Additionally, drawing connections within novel phages has allowed the computational identification of novel spanin genes [23].

Identifying interrupted genes

Several types of interrupted genes are present in phages, including programmed translational frameshifts, introns or inteins. Researchers typically use an integrated approach to detect the presence and boundaries of expected interrupted genes. Because the functional analysis performs sequence similarity searches, we can take advantage of those results to identify interrupted proteins compared to their uninterrupted counterparts present in the database. Due to biological variety in this area and chain annotation that tends to propagate common annotations (over quality annotations), much work is needed here to populate the databases with accurate annotations. The **Intron Detection** tool can detect legitimate introns (often in essential genes), frameshifts or premature stops produced by sequencing errors (such as those introduced by long-read sequencers), programmed translational frameshifts (frequently found in capsid proteins or tape measure chaperones), and other split or duplicate gene phenomena.

In the CPT Galaxy pipeline, the interrupted genes detection tool parses XML-formatted BLASTp results to determine whether multiple query proteins from the phage genome have amino acid identity to a single target protein in the database. The presence of a single database protein as multiple discrete segments in the query genome is the signature of a gene disrupted by an intron, programmed translational frameshift, or sequencing artifact. The tool requires that the separated gene fragments in the query genome exist within a user-defined number of bases, with the default set at 10,000. Separation minimum evaluates overlap of queries on the subject sequence (in case gene calls with incorrect starts were input); negative values allow overlap and positive/zero values allow separation. Logical checks in the script include removing alignments with high identity over a short part of a longer segment (identities/hsp length < 0.3 are discarded), requiring at least two unique high scoring pairs for any given target, and ensuring that the common hits are encoded on the same strand. This tool can also detect separated genes that span the genome ends, for cases where a genome may circularize or if it is opened in the middle of a gene.

In the case of introns, which are self-splicing RNAs usually found in essential genes, they share common secondary RNA structure but are not commonly predicted as part of annotation pipelines unless they contain a well-known intron-encoded protein. Studies have shown that ~25% of bacteria contain genes with group II introns, while phage regularly contain group I introns [29–31]. In the case of T4 and T4-like phages, these mobile elements are seen to vary substantially [32–34]. The most reliable way to find the true boundaries is by manual inspection of high-quality alignments of the intron-disrupted phage protein segments against a non-

interrupted homolog from the database, which is identified in the tool output in Apollo. When uninterrupted analogs do not exist in the database, or the nearest uninterrupted analog has weak sequence identity, boundaries cannot be accurately determined bioinformatically.

There are several kinds of important proteins in phages that are produced via programmed translational frameshifting mechanisms, such as the capsid or tail proteins and the tail tape measure chaperones, of which λ is the best characterized example [35–40]. When the frame-shifted downstream open reading frame of a protein has been called as a separate protein-coding gene (often without a strong SD), it can be detected by the interrupted genes tool. However, in cases where no experimental data exists for the matched protein in the database, the potential for an interrupted gene should be interpreted with caution.

Finally, the interrupted genes tool can be very helpful for interpreting long reads with many errors, such as those that result from nanopore sequencing. Taking into consideration the noisy output of gene calling (reducing stringency for ORF length and starts), this will be most successful where the genome has high similarity to previously deposited genomes.

Custom BLAST databases and BLAST restriction by TaxID

The mainstay method for sequence comparison in genome annotation is BLAST [41]. While the number of sequenced phage genomes present in the NCBI database has increased rapidly since next-generation sequencing became widespread [42,43], so have sequences of cellular organisms and eukaryotic viruses [44]. Sequences outside bacteria and phages are usually not relevant to our analyses, drown out low similarity hits of interest, and significantly increase required processing time. The CPT Galaxy instance, with locally installed database copies updated about three times per year from their respective home sites, offers at least two ways to mitigate those weaknesses: custom databases and restriction by TaxID.

Custom databases. Users can generate their own local databases in Galaxy (via the **NCBI BLAST+ makeblastdb** tool), or use public databases pre-compiled by the CPT. Significantly, we offer a ‘Canonical Phages Database’, representing the proteomes of phages with significant publication history or of historical importance. This list includes paradigm phages such as Lambda, T4, and T7, as well as P22, P2, and phi29 (see [S3 Table](#) for full list). The reason for generating this custom database is to ensure that those phage proteins annotated based on experimental evidence are detectable as hits against novel phage proteins, rather than them being lost amid a slew of less relevant results. This ensures that when there is a single degree of separation between the predicted protein of a novel phage and a studied protein in a canonical phage, that prediction can have higher confidence.

When analyzing phage genomes for therapy applications, toxin/anti-toxin systems, restriction-modification systems, CRISPR-related proteins, and anti-CRISPR proteins are of particular interest to detect. Despite wet bench experiments being performed on many of these proteins, they are poorly annotated in most databases. Custom databases from users or the community can be added to Galaxy for specialized analyses.

Restriction by TaxID. In the **BLAST+ 2.9.0** suite using version 5 databases, standalone BLAST jobs can now restrict searches by TaxID, as has long been available on the NCBI BLAST webserver. On the CPT Galaxy instance, we modified the BLAST wrapper to make this functionality available, with searching using non-species-level IDs available with the TaxonKit tool to retrieve all species-level TaxIDs that are required by BLAST 2.9 [45]. Using curated lists of TaxIDs that include all phages, bacteria, and/or canonical phages ([S3 Table](#)), we are able to drastically reduce the processing time while targeting searches more effectively.

Additional tools

In CPT Galaxy, there are also assembly tools, including **SPAdes** and **FastQC**. Since phage genomes are relatively small, typically <200 kb [46], they usually assemble completely in a single contig that is opened randomly by the assembler. Biologically, phage genomes can have physical ends with or without terminal repeats, or be circular [43]. As with other genome browsers, wraparounds are not supported at this time in Apollo. Therefore, the genome is displayed linearly and users must exercise careful judgement on interpretation of all features that border or span the boundary of unannotated genomes. Relatively few phages have had the state of their physical genomic termini probed. Therefore, we rely on comparisons to phages with empirically verified ends and computational predictions, whose accessibility is limited by the specific sequencing technology used. **PhageTerm** is an end prediction program installed in CPT Galaxy [47]. These software assessments can be useful in determining the most logical place to re-open a genome, usually decided by convention in the field. **Genome Editor** is a tool that allows rearrangement of the genome, with or without associated annotations. There are also tools that extract subsets of data from various genome record file types, validators to check their format, and a **PhageQC** report generator, useful for reviewing the quality of external genome annotations. For those accustomed to or desiring an external tabular file that collates their annotation data, the **Annotation Table** tool generates a downloadable Excel-compatible table. These, and a whole suite of plain text, table, and protein/nucleic acid manipulation tools are searchable at <https://cpt.tamu.edu/galaxy-pub/>.

After all desired analyses are completed, the logical next step is to deposit the sequence with its annotations into one of the public sequence repositories. The tools to retrieve the data from Apollo into Galaxy, as well as interconvert between file types, aid in this essential step. The **Retrieve Data** tool yields a DNA FASTA file and GFF3 file. These can be converted into a Genbank file with the bacterial (default) translation table qualifier added using the **GFF3 to Genbank** tool, then into the 5-column table used with NCBI's BankIt service using the **Genbank to Five Column Format** tool. Updates and new tools are regularly added to the CPT Galaxy platform. The most current development version is available through our public Github at (<https://github.com/TAMU-CPT/galaxy-tools>).

Workflows in Galaxy facilitate efficient multi-step analyses required for phage genome annotation

While having all these tools in one place is useful, the human hours required to sit at the computer and manually move the output from one tool into the next, and converting those files into compatible formats along the way, is prohibitive. We prefer to spend our time on the things that computers cannot do as well: making judgment calls on the annotations. Galaxy workflows offer the ability to chain many tools together automatically, such that a long and tedious analysis can be reproducibly applied to multiple datasets with a few clicks. Three main workflows are the mainstay of our phage annotation pipeline: the Structural Annotation workflow, the Functional Annotation workflow, and the Comparative Genomics workflow. Using the tools described above and in [S1 Table](#), analysis is conducted in Galaxy and then transferred to Apollo via the Galaxy-Apollo bridge, where evidence is displayed for interrogation by the user, who has ultimate control over the annotation process.

Structural annotation workflow

The purpose of the structural workflow is to provide the researcher with the evidence needed to make high-quality, confident gene calls with accurate prediction of the gene start. This is

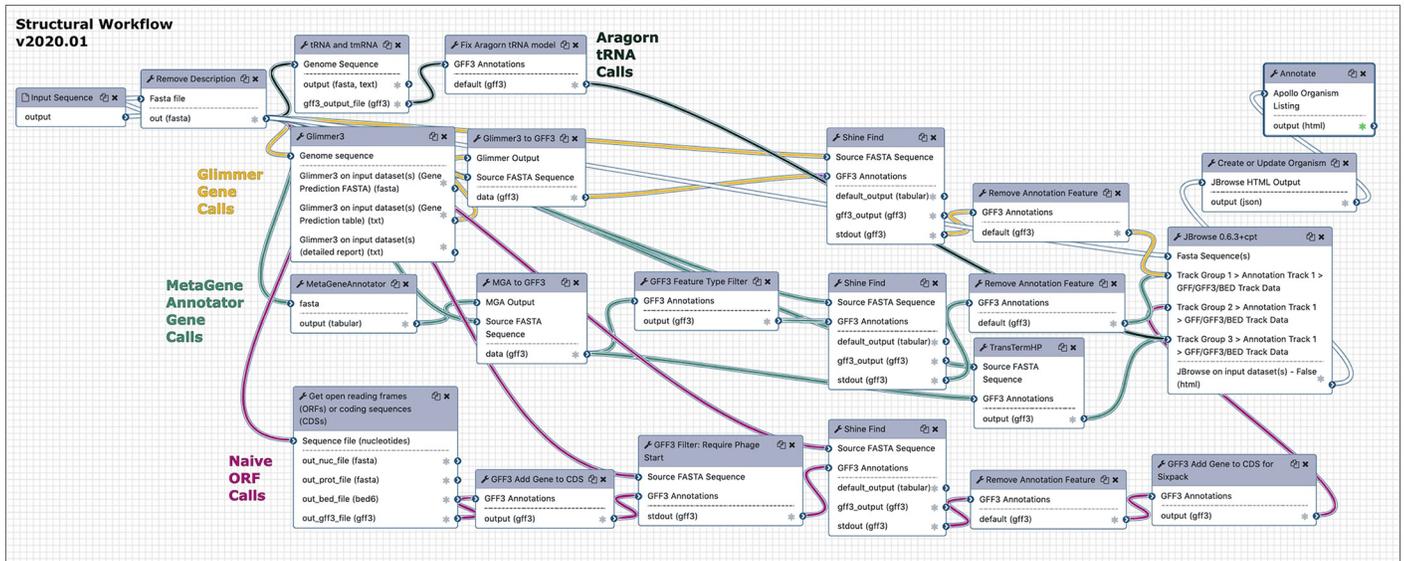


Fig 2. The structural workflow chains together tools in Galaxy for gene calling. The structural workflow accepts as input a nucleotide FASTA genome sequence, and processes it through ARAGORN for tRNAs (dark noodles), Glimmer and MetaGeneAnnotator for high-confidence gene predictions (golden and teal, respectively), and Get ORFs as a naïve ORF/CDS caller (magenta). Potential protein-coding genes are filtered to ensure the presence of a phage (ATG/GTG or TTG) start codon and a Shine-Dalgarno feature is added to all features that have a detectable match. These are interconverted between formats and the gene models are corrected for display in Apollo.

<https://doi.org/10.1371/journal.pcbi.1008214.g002>

accomplished in 23 total steps from input genome FASTA file to the updated Apollo organism with evidence tracks (Fig 2). The structural workflow is relatively quick for phage genomes, usually taking <20 minutes to complete.

From the naked input DNA sequence, five separate gene analysis tracks are initiated. Two gene callers, **GLIMMER 3.0** and **MetaGeneAnnotator v1.0 (MGA)**, both popular prokaryotic prediction programs, are used to predict the locations of protein-coding genes [48,49]. Their outputs are converted into GFF3 format and processed by ShineFind; RBS sequences assigned by MGA are discarded (feature filter step) in favor of the ShineFind algorithm. The outputs are then properly formatted with a gene model structure that follows gene-CDS and gene-Shine_Dalgarno parent-child relationships for display as a feature track for JBrowse. The MGA gene calls are used to predict rho-independent transcriptional terminators using **TransTermHP** [50]. A third gene caller, **Get open reading frames (ORFs) or coding sequences (CDSs)**, uses the Sixpack framework from EMBL, preset to use the NCBI translation table 11, for minimum 30 aa long ORFs with both a start, stop codon [16,17]. The GetORFs output is further filtered to only genes with phage start codons (ATG, GTG, and TTG) along with reformatting the gene model before formatting for display as a feature track for JBrowse. tRNA and tmRNA genes are predicted by **ARAGORN v2.36** and also formatted for display as a feature track in JBrowse [51].

Phage genomes are highly compact due to packaging constraints imposed by the amount of genomic DNA or RNA that can fit into the capsid [52,53]. With a typical >90% coding density, protein-coding genes often overlap by several bases with adjacent coding sequences. Many overlapping genes are missed by gene calling algorithms trained on bacterial genomes which are not subjected to compaction [53–55], which is why the CPT Galaxy pipeline is designed to allow comparison of the results from three programs. MGA and GLIMMER results usually agree, and predict the majority of protein-coding genes in the genome. However, when gaps are present, they can often be filled by reading frames identified in the naïve

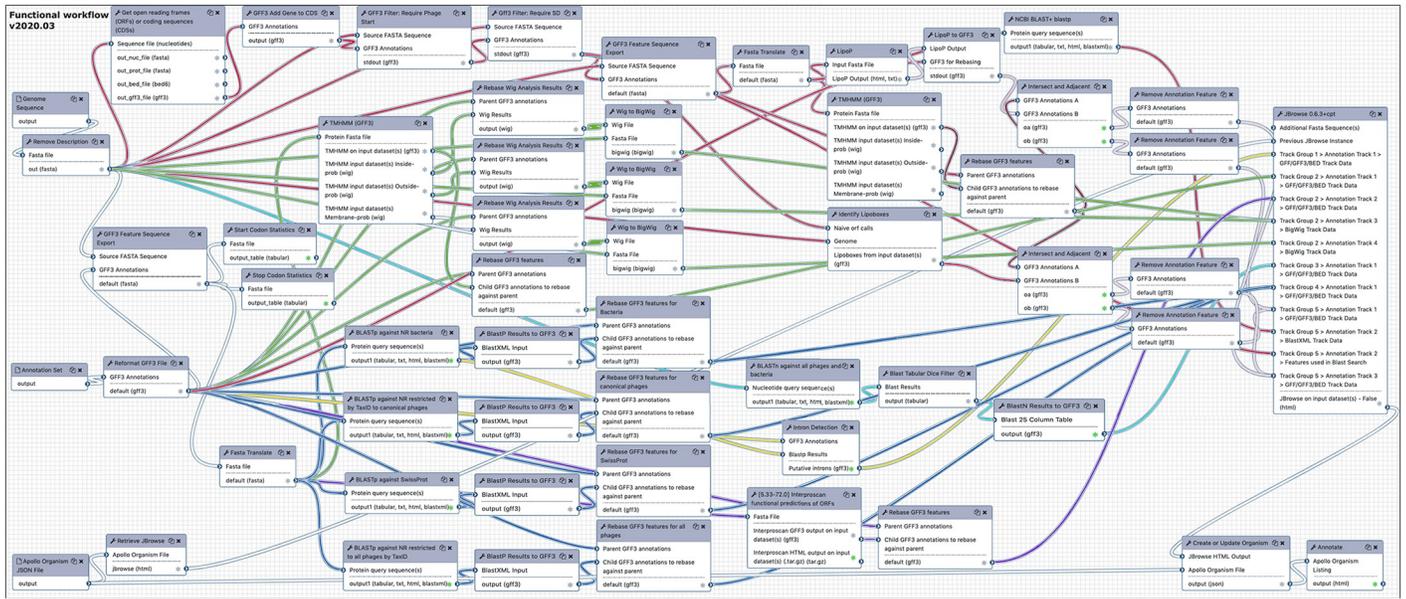


Fig 3. The functional workflow links tools in Galaxy used for functional prediction. Inputs for the functional workflow are gene calls paired with their genome. These are piped through five sub-paths within the analysis. 1) The BLASTn path uses full genomic nucleotide sequence (light blue noodles). 2) The BLASTp protein analysis against curated (UniProtKB SwissProt) and sequence-inclusive databases (NCBI nonredundant (nr)) (dark blue noodles). 3) The search for interrupted genes like introns compiles separate CDS hits to the same protein (yellow noodles). 4) A directed search for spanin proteins using TMHMM, lipobox-finding (using LipoP and a motif search), and BLASTp against a curated database (magenta and pink noodles). 5) Domain analysis plots comprehensive TMHMM outputs and InterProScan results for conserved domains and signatures (green and purple noodles, respectively).

<https://doi.org/10.1371/journal.pcbi.1008214.g003>

(and noisy) GetORFs/Sixpack track. The user may evaluate the tool outputs and determine which predicted genes are promoted to protein-coding gene features in the Apollo editor window. As ARAGORN only displays tRNA predictions of high confidence, these are typically all promoted as features. The TransTermHP results are evaluated by the user, and may be promoted as annotations based on the criteria of a score >95, a hairpin stem with at least 5 matches, a minimum four-T run in the T-tail after the stem, and genomic context. These annotations can then be imported back into Galaxy as a GFF3 file for further analysis.

Functional annotation workflow

The goal of the functional workflow is to provide the user with contextualized evidence to predict the function of the proteins encoded by their input genes. The 57 steps comprising the functional workflow are the main workhorse of the phage annotation process (Fig 3). Due to the use of BLAST and InterProScan, the functional workflow can take several hours to complete, commensurate with the size of the input dataset.

The independent segments of the workflow will be discussed in these sections: BLAST analysis, InterProScan, and transmembrane domain finding (the intron and spanin detection segments are described above). As an added value feature, start and stop codon statistics are calculated.

BLAST. First, the genome sequence is analyzed by BLASTn (dc-megablast) with a 0.001 expectation cutoff, against the NCBI nonredundant (nt) database restricted by all phages and bacterial TaxIDs. The tabular BLASTn results are parsed with the **Blast Tabular Dice Filter** tool (filters results with low Dice coefficient, and requires 50–100% identity), then mapped back onto the genome as a GFF3 for display in JBrowse. Second, the CDS features in GFF3

format are extracted from the structural annotation results, translated, and analyzed in four separate jobs by traditional BLASTp at a 0.001 expectation value cutoff and default parameters. One job searches against the UniProtKB SwissProt database, which includes only manually annotated and reviewed sequences, providing a first good clue to their function. Two BLASTp jobs search against the NCBI nr database restricted to all phages and canonical phages (S3 Table) by TaxID. The fourth BLASTp analysis searches against the entire nr database. All BLASTp outputs (in XML format) are converted to GFF3, then 'rebased', or mapped, back to the parent genome for display as an evidence track in JBrowse format.

InterProScan is a conserved domain search tool incorporating information from several other services [56–59]. In the CPT Galaxy workflow, the same translated CDS features used for BLASTp analysis are searched against conserved domain databases by InterProScan, which returns a GFF3-formatted result that is mapped (or 'rebased') back onto the parent genome for JBrowse display.

Transmembrane domains are predicted from the translated features using TMHMM 2.0 [28]. The simple output is a GFF3 file, which, when displayed in Apollo shows the location of the probable TMD, and gives its predicted orientation in the feature name. Three additional outputs for inside, outside, and membrane probability are converted from wig to bigwig format for a plot format display [60], and mapped back onto the parent GFF3 at the proper location. Additionally, transmembrane domains and signal peptides are predicted by Phobius, which is incorporated into the InterProScan software package [59,61].

The final output of the workflow is the direct Apollo link to the updated JBrowse instance for the organism. In Apollo, each evidence track will be displayed in context with the rest of the genome for evaluation in functional prediction. After predictions are completed, the genome annotations can be returned to Galaxy with the Retrieve Data tool, yielding a FASTA and paired GFF3 file. These can then be converted into the five-column table required by NCBI for deposition into Genbank, as described elsewhere.

Comparative genomics workflow

The goal of the comparative workflow is to identify the most related phages to the query organism, compared at both nucleotide and protein levels. This workflow requires the genomic DNA sequence in FASTA format and the GFF3 annotations of protein-coding genes, and returns tables listing the most related organisms by BLASTn and BLASTp analysis, and a series of pairwise DNA dot plots. There are 11 steps in the comparative workflow and it usually takes less than ten minutes to run (Fig 4).

Protein comparison is performed by a BLASTp search against NR restricted by all phage TaxIDs (S3 Table) and an expectation value cutoff of 0.001. The top hits (default is 20) are parsed from the custom tabular output by TaxID and returned as a table with the phage common name, NCBI TaxID, and number of similar unique proteins they share with the query phage.

Full-genome nucleotide comparison starts with a traditional BLASTn megablast restricted to all phages in the NCBI nr database using a 0.001 expectation value cutoff. The number of user-specified top hits (default is ten) are parsed from the custom tabular output based on percent identity (calculated for the query:subject pair according to the Dice coefficient = $(2 * \# \text{ identical matches}) / ((\text{genome 1 length}) + (\text{genome 2 length}))$). With this information, the full FASTA record for each top matching genome is retrieved via the NCBI **EFetch** utility. After concatenation, the related genomes are processed by **MIST** v3 to generate a dot plot; MIST builds upon Gepard [62] to generate a matrix of multiple pairwise dot plots for a set of input sequences. The MIST Python script calls Gepard for each pairwise comparison and builds the

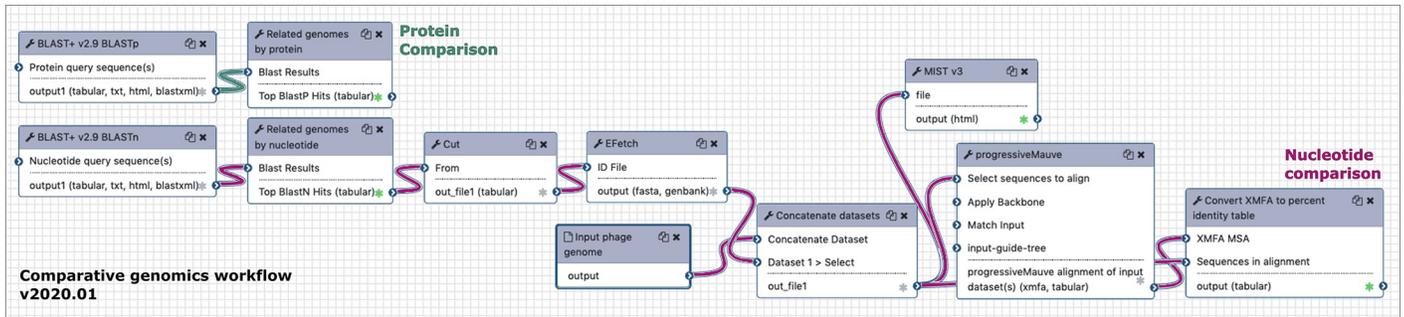


Fig 4. The comparative genomics workflow calculates nucleotide and protein similarity to other phages. The protein comparison branch starts with a BLASTp job against the nr database, restricted by all phage TaxIDs (see full list in S3 Table), and is then sorted according to the organisms with the highest number of unique protein hits (teal noodles). The nucleotide comparison branch begins with a BLASTn job against the nt database, also restricted by all phage TaxIDs. Top nucleotide hits are sorted based on dice score, which accounts for the total coverage. The top five genome sequences are fetched from NCBI, concatenated with the query genome and routed to MIST v3 for a dot plot, and progressiveMauve for calculation of pairwise percent identity (magenta noodles).

<https://doi.org/10.1371/journal.pcbi.1008214.g004>

combined image and image map. DNA sequences are also compared by **progressiveMauve** v2.4.0, with the aligned blocks extracted and output as a table with the **Convert XMFA to percent identity table** tool, which presents overall percent nucleotide identity as a Dice coefficient [62,63].

The data in these tables can be directly used for publication, as reference for finding literature on the genome type, or as the basis for additional comparative genomics including the generation of synteny plots using tools in CPT Galaxy like **X-Vis**, **Genome Mapper**, or **Easy-Fig** [64].

More on workflows

The three workhorse workflows described can be customized in the Galaxy workflow editor by addition or removal of individual tools, updating tool functionality, or simply tweaking job parameters as needed. New mainstay workflow versions we prepare regularly become publicly available to the community, and users can contribute their customizations or specialized workflows as well. The signature workflows described above are geared towards the annotation of new genomes, but we will note a few additional applications of the entire interface here. Importantly, pre-annotated genomes can be loaded into Apollo using the *Upload Genbank into Apollo* (Genbank file input) or *Upload Previously Annotated Sequence to Apollo* (FASTA and GFF3 input) workflows for updating by the user or community, as determined by whom the user gives access to the organism. This will aid in our goal of reaching at least one “gold-standard” annotated genome per representative phage type. Users can also compare directly to the original annotations by adding them as an evidence track with the *GFF3 to Apollo evidence track* workflow.

CPT Galaxy users have also built and shared a variety of generally applicable routines. There is a whole collection of workflows that will perform an analysis and push the results to an evidence track in Apollo, usually BLAST jobs. The workflows *Custom BLASTp to Apollo Evidence Track–UserDB* and its variation from a *Local DB* allow running a BLASTp analysis against a user-generated local database. A specific use case is the *BLAST antiCRISPRdb to Apollo Evidence Track* workflow, which searches against a local copy of the curated anti-CRISPRdb [65]. Another useful workflow set, *X-vis from GFF3/Fasta*, will generate a synteny map converting BLAST similarity into protein XMFA format for visualization across the entire genome. The direct links to current versions of the workflows mentioned here are listed in S4

[Table](#). A full list of the most recent published versions for the annotation pipeline and all other workflows can be accessed at https://cpt.tamu.edu/galaxy-pub/workflows/list_published.

Results & discussion

Comparison to other available annotation tools

Various groups host annotation systems publicly accessible for phage annotation. DNAMaster is arguably the most widely used program for teaching phage annotation in the undergraduate education context [66]. Thousands of students have performed phage annotation through the SEA-PHAGES program in this stand-alone Windows program written originally by Jeffrey Lawrence. Recently, an independent research group released multiPhATE, a downloadable command-line bioinformatics pipeline for functional annotation of phage isolates [9]. One unique aspect in multiPhATE is an accompanying tool designed specifically to detect phage genes called PHANOTATE [54].

Prior to these examples with adaptations tuned to phage annotation, most automated pipelines in wide use were developed around prokaryotic annotation needs. The service offered with deposit to Genbank, the Prokaryotic Genome Annotation Pipeline (PGAP), is now available as a stand-alone program for bacterial and archaeal genomes and used for all RefSeq sequences [67,68]. DFAST is a downloadable and web-based prokaryotic genome annotation pipeline that boasts a 10-minute or less processing time for full bacterial genomes [69]. The RASTtk pipeline is a downloadable and web-based prokaryotic genome annotation service, used in the microbiology community as the basis for PATRIC (PAThOSystems Resource Integration Center) [70–72]. Their use of subsystems to perform quick and automatic annotations has also been applied to phage genome annotation, recently described in manual format [73].

Given its widespread use, we compared the output of annotation with the tools in CPT Galaxy to RASTtk for five phages representing all the major morphotypes: four that we originally annotated, and phage T1 from RefSeq (S5 Table). The final number of genes called was similar, within six of each other, with CPT Galaxy annotations always having the higher count. The phages annotated using the CPT Galaxy-Apollo platform had fewer proteins without a functional prediction (hypothetical or phage protein). Finally, the number of genes called with a valid Shine-Dalgarno sequence was higher for CPT Galaxy, but only by a maximum of seven genes in total. In addition to missing more complex genome features, such as terminal repeat regions and rho-independent terminators, RASTtk did not specifically assign function to various known phage proteins with bioinformatically recognizable characteristics, such as the *spans* and translationally frameshifted tape measure protein chaperones.

No currently available automated annotation algorithm can reasonably be expected to reach the highest quality product that manual, human effort can produce, and that in turn is inferior to empirical data to support annotations. However, it is not always feasible to invest significant researcher time on upstream annotation processes. While experienced annotators can be quite efficient, when speed is of the essence, automated pipelines can be combined with the convenient browsing and editing tools available in the CPT Galaxy for polishing. Additionally, the annotation can be performed collaboratively across groups or institutions, making use of the “Google Docs”-type functionality, as we have done in several instances [74–76]. Finally, the system’s application to a classroom setting is also immediately apparent, even a virtual classroom, where teams can work on the same genome, and all contributions are logged per user.

Conclusion

Here, we have described a complete, easy-to-use web-based platform for phage annotation. The powerful suite of tools housed in the CPT Galaxy instance provides a dynamic, scalable

framework for genome data analyses that can be used independently of, or in conjunction with the community genome editor, Apollo. In both research and educational contexts, the CPT Galaxy and Apollo system allows users to focus on the biology behind annotation rather than the minutiae of server maintenance, command-line operations and file management. This is essential to allow us to use the flood of sequencing data to move deeper into the understanding of phage biology and the fundamental principles that govern their bacterial hosts, and make informed decisions about their applications in the future.

Supporting information

S1 Table. A list of all tools used in the structural (S), functional (F), and comparative genomics (C) workflows. The current working versions of the tools presented here are available at <https://github.com/TAMU-CPT/galaxy-tools>.

(XLSX)

S2 Table. The consensus and derivative possible Shine-Dalgarno sequences that are the default in ShineFind. By default, the ShineFind tool searches within 3–24 nucleotides upstream of a given start codon for the longest, or first, possible Shine-Dalgarno sites that match to the following, starting with the *E. coli* consensus.

(XLSX)

S3 Table. List of common names and identifiers used in BLAST. Each category may be used separately or together to restrict a BLAST analysis by TaxID. Phage common names and NCBI accession numbers are included in the custom canonical phages database, which is also pre-compiled as a BLAST database. Bolded rows contain the well-studied representatives of each morphotype.

(XLSX)

S4 Table. Helpful CPT Galaxy workflow links. Current versions of all published workflows can be accessed at https://cpt.tamu.edu/galaxy-pub/workflows/list_published.

(XLSX)

S5 Table. Comparison of phage genome annotations performed using different methods. Five phages, the canonical phage T1 and four previously annotated at the CPT, were assessed with RASTtk and features were compared. The number of gene features listed is strictly a count, not reflecting the possibility that completely different genes, or genes with alternate starts, were called. The hypothetical protein count is meant to be a proxy for functional annotation. Since RASTtk also uses the nonspecific name “phage protein”, this term was separately tallied. In the additional features row are listed genes and other aspects of phage genomes that are typically only identified by manual inspection, and which only our pipeline identifies here. tmp = tape measure protein, and terminators refer to rho-independent terminators predicted by TransTermHP. Compiled annotation data can be browsed by the public at <https://cpt.tamu.edu/apollo/jbrowse/>.

(XLSX)

Acknowledgments

The Center for Phage Technology received an Initial University Multidisciplinary Research Initiative from Texas A&M University and Texas AgriLife. We are also grateful to the Texas A&M Department of Biochemistry and Biophysics for support. Many CPT staff, undergraduates and graduate students in the BICH 464 Bacteriophage Genomics course of the Department of Biochemistry and Biophysics at Texas A&M have tested the system through their use since the

initiation of this project. We thank Rodolfo Aramayo for early suggestions to implement Galaxy. We thank the Galaxy community contributors for their development and maintenance of this open resource, including the format of the training materials. Additionally, we thank Suzanna Lewis and Nathan Dunn for their support in adapting Apollo to phage annotation.

Author Contributions

Conceptualization: Helena Rasche, Jason J. Gill.

Data curation: Jolene Ramsey, Helena Rasche, Mei Liu, Ry Young, Jason J. Gill.

Formal analysis: Helena Rasche.

Funding acquisition: James C. Hu, Ry Young, Jason J. Gill.

Investigation: Jolene Ramsey, Jason J. Gill.

Methodology: Helena Rasche, Cory Maughmer, Anthony Criscione, Eleni Mijalis.

Project administration: Mei Liu, Ry Young, Jason J. Gill.

Resources: Cory Maughmer, Ry Young, Jason J. Gill.

Software: Helena Rasche, Cory Maughmer, Anthony Criscione, Eleni Mijalis.

Supervision: Jolene Ramsey, Helena Rasche, Cory Maughmer, Mei Liu, James C. Hu, Ry Young, Jason J. Gill.

Validation: Jolene Ramsey, Helena Rasche, Cory Maughmer, Eleni Mijalis, Mei Liu, Ry Young, Jason J. Gill.

Visualization: Helena Rasche.

Writing – original draft: Jolene Ramsey.

Writing – review & editing: Jolene Ramsey, Helena Rasche, Cory Maughmer, Anthony Criscione, Eleni Mijalis, Mei Liu, Ry Young, Jason J. Gill.

References

1. Ofir G, Sorek R. Contemporary Phage Biology: From Classic Models to New Insights. *Cell*. 2018; 172:1260–1270. <https://doi.org/10.1016/j.cell.2017.10.045> PMID: 29522746
2. Salmond GPC, Fineran PC. A century of the phage: past, present and future. *Nat Rev Microbiol*. 2015; 13:777–786. <https://doi.org/10.1038/nrmicro3564> PMID: 26548913
3. Young R, Gill JJ. MICROBIOLOGY. Phage therapy redux—What is to be done? *Science (New York, NY)*. 2015; 350:1163–1164. PMID: 26785457
4. Pirnay J-P, Verbeken G, Ceysens P-J, Huys I, Vos DD, Ameloot C, et al. The Magistral Phage. *Viruses*. 2018; 10:64. <https://doi.org/10.3390/v10020064> PMID: 29415431
5. Pouillot F, Blois H, Iris F. Genetically engineered virulent phage banks in the detection and control of emergent pathogenic bacteria. *Biosecurity Bioterrorism Biodefense Strategy Pract Sci*. 2010; 8:155–69. <https://doi.org/10.1089/bsp.2009.0057> PMID: 20569057
6. Hatfull GF, Hendrix RW. Bacteriophages and their genomes. *Current opinion in virology*. 2011; 1:298–303. <https://doi.org/10.1016/j.coviro.2011.06.009> PMID: 22034588
7. Brister JR, Ako-Adjei D, Bao Y, Blinkova O. NCBI viral genomes resource. *Nucleic acids research*. 2015; 43:D571–7. PMID: 25428358
8. Shean RC, Makhsous N, Stoddard GD, Lin MJ, Greninger AL. VAPiD: a lightweight cross-platform viral annotation pipeline and identification tool to facilitate virus genome submissions to NCBI GenBank. *BMC bioinformatics*. 2019; 20:48. <https://doi.org/10.1186/s12859-019-2606-y> PMID: 30674273
9. Zhou CLE, Malfatti S, Kimbrel J, Philipson C, McNair K, Hamilton T, et al. multiPhATE: bioinformatics pipeline for functional annotation of phage isolates. Hancock J, editor. *Bioinformatics*. 2019;44W16. <https://doi.org/10.1093/bioinformatics/btz258>

10. Philipson C, Voegtly L, Lueder M, Long K, Rice G, Frey K, et al. Characterizing Phage Genomes for Therapeutic Applications. *Viruses*. 2018; 10: 188–20. <https://doi.org/10.3390/v10040188> PMID: 29642590
11. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Cech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic acids research*. 2018; 46:W537–W544. PMID: 29790989
12. Lee E, Helt GA, Reese JT, Munoz-Torres MC, Childers CP, Buels RM, et al. Web Apollo: a web-based genomic annotation editing platform. *Genome biology*. 2013; 14:R93. <https://doi.org/10.1186/gb-2013-14-8-r93> PMID: 24000942
13. Dunn NA, Unni DR, Diesh C, Munoz-Torres M, Harris NL, Yao E, et al. Apollo: Democratizing genome annotation. Darling AE, editor. *Plos Comput Biol*. 2019; 15:e1006790. <https://doi.org/10.1371/journal.pcbi.1006790> PMID: 30726205
14. Skinner ME, Uzilov AV, Stein LD, Mungall CJ, Holmes IH. JBrowse: a next-generation genome browser. *Genome Res*. 2009; 19:1630–8. <https://doi.org/10.1101/gr.094607.109> PMID: 19570905
15. Blankenberg D, Kuster GV, Bouvier E, Baker D, Afgan E, Stoler N, et al. Dissemination of scientific software with Galaxy ToolShed. *Genome Biol*. 2014; 15:403. <https://doi.org/10.1186/gb4161> PMID: 25001293
16. Cock PJA, Grüning BA, Paszkiewicz K, Pritchard L. Galaxy tools and workflows for sequence analysis with applications in molecular plant pathology. *Peerj*. 2013; 1:e167. <https://doi.org/10.7717/peerj.167> PMID: 24109552
17. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*. 2009; 25:1422–1423. PMID: 19304878
18. Consortium U. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res*. 2019; 47:D506–D515. PMID: 30395287
19. Buels R, Yao E, Diesh CM, Hayes RD, Munoz-Torres M, Helt G, et al. JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol*. 2016; 17:66. <https://doi.org/10.1186/s13059-016-0924-1> PMID: 27072794
20. Batut B, Hilteemann S, Bagnacani A, Baker D, Bhardwaj V, Blank C, et al. Community-Driven Data Analysis Training for Biology. *Cell Syst*. 2018; 6:752–758.e1. <https://doi.org/10.1016/j.cels.2018.05.012> PMID: 29953864
21. Mungall CJ, Batchelor C, Eilbeck K. Evolution of the Sequence Ontology terms and relationships. *J Biomed Inform*. 2011; 44:87–93. <https://doi.org/10.1016/j.jbi.2010.03.002> PMID: 20226267
22. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, et al. The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol*. 2005; 6:R44. <https://doi.org/10.1186/gb-2005-6-5-r44> PMID: 15892872
23. Kongari R, Rajaure M, Cahill J, Rasche E, Mijalis E, Berry J, et al. Phage spanins: diversity, topological dynamics and gene convergence. *BMC bioinformatics*. 2018; 19:326. <https://doi.org/10.1186/s12859-018-2342-8> PMID: 30219026
24. Babu MM, Priya ML, Selvan AT, Madera M, Gough J, Aravind L, et al. A Database of Bacterial Lipoproteins (DOLOP) with Functional Assignments to Predicted Lipoproteins. *J Bacteriol*. 2006; 188:2761–2773. <https://doi.org/10.1128/JB.188.8.2761-2773.2006> PMID: 16585737
25. von Heijne G. The structure of signal peptides from bacterial lipoproteins. *Protein Eng Des Sel*. 1989; 2:531–534. PMID: 2664762
26. Cahill J, Young R. Phage Lysis: Multiple Genes for Multiple Barriers. *Adv Virus Res*. 2018; 103:33–70. <https://doi.org/10.1016/bs.avir.2018.09.003> PMID: 30635077
27. Juncker AS, Willenbrock H, Heijne GV, Brunak S, Nielsen H, Krogh A. Prediction of lipoprotein signal peptides in Gram-negative bacteria. *Protein science: a publication of the Protein Society*. 2003; 12:1652–1662. PMID: 12876315
28. Krogh A, Larsson B, von Heijne G, Sonnhammer EL. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of Molecular Biology*. 2001; 305:567–580. <https://doi.org/10.1006/jmbi.2000.4315> PMID: 11152613
29. Lambowitz AM, Zimmerly S. Group II introns: mobile ribozymes that invade DNA. *Csh Perspect Biol*. 2011; 3:a003616. <https://doi.org/10.1101/cshperspect.a003616> PMID: 20463000
30. Candales MA, Duong A, Hood KS, Li T, Neufeld RAE, Sun R, et al. Database for bacterial group II introns. *Nucleic Acids Res*. 2011; 40:D187–D190. PMID: 22080509
31. Leclercq S, Gilbert C, Cordaux R. Cargo capacity of phages and plasmids and other factors influencing horizontal transfers of prokaryote transposable elements. *Mob Genetic Elements*. 2012; 2:115–118. <https://doi.org/10.4161/mge.20352> PMID: 22934247

32. Edgell DR, Gibb EA, Belfort M. Mobile DNA elements in T4 and related phages. *Virology*. 2010; 7:290. <https://doi.org/10.1186/1743-422X-7-290> PMID: 21029434
33. Petrov VM, Nolan JM, Bertrand C, Levy D, Desplats C, Krisch HM, et al. Plasticity of the Gene Functions for DNA Replication in the T4-like Phages. *J Mol Biol*. 2006; 361:46–68. <https://doi.org/10.1016/j.jmb.2006.05.071> PMID: 16828113
34. Petrov VM, Ratnayaka S, Karam JD. Genetic Insertions and Diversification of the PoIB-Type DNA Polymerase (gp43) of T4-Related Phages. *J Mol Biol*. 2010; 395:457–474. <https://doi.org/10.1016/j.jmb.2009.10.054> PMID: 19896487
35. Baranov PV, Fayet O, Hendrix RW, Atkins JF. Recoding in bacteriophages and bacterial IS elements. *Trends Genet*. 2006; 22:174–181. <https://doi.org/10.1016/j.tig.2006.01.005> PMID: 16460832
36. Atkins JF, Loughran G, Bhatt PR, Firth AE, Baranov PV. Ribosomal frameshifting and transcriptional slippage: From genetic steganography and cryptography to adventitious use. *Nucleic Acids Res*. 2016; 44:7007–78. PMID: 27436286
37. Xu J, Hendrix RW, Duda RL. Conserved translational frameshift in dsDNA bacteriophage tail assembly genes. *Molecular cell*. 2004; 16:11–21. <https://doi.org/10.1016/j.molcel.2004.09.006> PMID: 15469818
38. Garcia P, Rodriguez I, Suarez JE. A -1 Ribosomal Frameshift in the Transcript That Encodes the Major Head Protein of Bacteriophage A2 Mediates Biosynthesis of a Second Essential Component of the Capsid. *J Bacteriol*. 2004; 186:1714–1719. PMID: 14996802
39. Auzat I, Dröge A, Weise F, Lurz R, Tavares P. Origin and function of the two major tail proteins of bacteriophage SPP1. *Molecular Microbiology*. 2008; 70:557–569. <https://doi.org/10.1111/j.1365-2958.2008.06435.x> PMID: 18786146
40. Condrón BG, Atkins JF, Gesteland RF. Frameshifting in gene 10 of bacteriophage T7. *J Bacteriol*. 1991; 173:6998–7003. <https://doi.org/10.1128/jb.173.21.6998-7003.1991> PMID: 1938901
41. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. *BMC bioinformatics*. 2009; 10:421. <https://doi.org/10.1186/1471-2105-10-421> PMID: 20003500
42. Hatfull GF. Innovations in Undergraduate Science Education: Going Viral. *J Virol*. 2015; 89:8111–3. <https://doi.org/10.1128/JVI.03003-14> PMID: 26018168
43. Merrill BD, Ward AT, Grose JH, Hope S. Software-based analysis of bacteriophage genomes, physical ends, and packaging strategies. *BMC genomics*. 2016; 17:679. <https://doi.org/10.1186/s12864-016-3018-2> PMID: 27561606
44. O'Leary NA, Wright MW, Brister JR, Ciuffo S, Haddad D, McVeigh R, et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res*. 2015; 44:D733–45. PMID: 26553804
45. Shen W, Xiong J. TaxonKit: a cross-platform and efficient NCBI taxonomy toolkit. *Biorxiv*. 2019;513523. <https://doi.org/10.1101/513523>
46. Dion MB, Oechslin F, Moineau S. Phage diversity, genomics and phylogeny. *Nat Rev Microbiol*. 2020; 18:125–138. <https://doi.org/10.1038/s41579-019-0311-5> PMID: 32015529
47. Garneau JR, Depardieu F, Fortier L-C, Bikard D, Monot M. PhageTerm: a tool for fast and accurate determination of phage termini and packaging mechanism using next-generation sequencing data. *Scientific reports*. 2017; 7:8292. <https://doi.org/10.1038/s41598-017-07910-5> PMID: 28811656
48. Delcher AL, Harmon D, Kasif S, White O, Salzberg SL. Improved microbial gene identification with GLIMMER. *Nucleic acids research*. 1999; 27:4636–4641. PMID: 10556321
49. 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 research: an international journal for rapid publication of reports on genes and genomes*. 2008; 15:387–396. PMID: 18940874
50. Kingsford CL, Ayanbule K, Salzberg SL. Rapid, accurate, computational discovery of Rho-independent transcription terminators illuminates their relationship to DNA uptake. *Genome biology*. 2007; 8:R22. <https://doi.org/10.1186/gb-2007-8-2-r22> PMID: 17313685
51. Laslett D, Canback B. ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic acids research*. 2004; 32:11–16. PMID: 14704338
52. Miller ES, Kutter E, Mosig G, Arisaka F, Kunisawa T, Ruger W. Bacteriophage T4 Genome. *Microbiol Mol Biol R*. 2003; 67:86–156. <https://doi.org/10.1128/mmlr.67.1.86-156.2003> PMID: 12626685
53. Kang HS, McNair K, Cuevas DA, Bailey BA, Segall AM, Edwards RA. Prophage genomics reveals patterns in phage genome organization and replication. *Biorxiv*. 2017;114819. <https://doi.org/10.1101/114819>

54. McNair K, Zhou C, Dinsdale EA, Souza B, 2019. PHANOTATE: A novel approach to gene identification in phage genomes. *Bioinformatics*. 2019. PMID: [31329826](#)
55. 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. PMID: [22584627](#)
56. Quevillon E, Silventoinen V, Pillai S, Harte N, Mulder N, Apweiler R, et al. InterProScan: protein domains identifier. *Nucleic Acids Res*. 2005; 33:W116–W120. PMID: [15980438](#)
57. Zdobnov EM, Apweiler R. InterProScan—an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*. 2001; 17:847–848. PMID: [11590104](#)
58. Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, et al. InterPro: the integrative protein signature database. *Nucleic Acids Res*. 2008; 37:D211–5. PMID: [18940856](#)
59. Jones P, Binns D, Chang H-Y, Fraser M, Li W, McAnulla C, et al. InterProScan 5: genome-scale protein function classification. *Bioinformatics*. 2014; 30:1236–1240. PMID: [24451626](#)
60. Kent WJ, Zweig AS, Barber G, Hinrichs AS, Karolchik D. BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics*. 2010; 26:2204–2207. PMID: [20639541](#)
61. Käll L, Krogh A, Sonnhammer ELL. A combined transmembrane topology and signal peptide prediction method. *J Mol Biol*. 2004; 338:1027–1036. <https://doi.org/10.1016/j.jmb.2004.03.016> PMID: [15111065](#)
62. Krumsiek J, Arnold R, Rattei T. Gepard: a rapid and sensitive tool for creating dotplots on genome scale. *Bioinformatics*. 2007; 23:1026–1028. PMID: [17309896](#)
63. Darling AE, Mau B, Perna NT. progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement. Stajich JE, editor. *PloS one*. 2010; 5:e11147. <https://doi.org/10.1371/journal.pone.0011147> PMID: [20593022](#)
64. Sullivan MJ, Petty NK, Beatson SA. Easyfig: a genome comparison visualizer. *Bioinform Oxf Engl*. 2011; 27:1009–10. PMID: [21278367](#)
65. Dong C, Hao G-F, Hua H-L, Liu S, Labena AA, Chai G, et al. Anti-CRISPRdb: a comprehensive online resource for anti-CRISPR proteins. *Nucleic acids research*. 2018; 46:D393–D398. PMID: [29036676](#)
66. Jordan TC, Burnett SH, Carson S, Caruso SM, Clase K, DeJong RJ, et al. A broadly implementable research course in phage discovery and genomics for first-year undergraduate students. Losick R, editor. *mBio*. 2014; 5:e01051–13. <https://doi.org/10.1128/mBio.01051-13> PMID: [24496795](#)
67. Tatusova T, DiCuccio M, Badretdin A, Chetvernin V, Nawrocki EP, Zaslavsky L, et al. NCBI prokaryotic genome annotation pipeline. *Nucleic Acids Res*. 2016; 44:6614–6624. PMID: [27342282](#)
68. Haft DH, DiCuccio M, Badretdin A, Brover V, Chetvernin V, O'Neill K, et al. RefSeq: an update on prokaryotic genome annotation and curation. *Nucleic Acids Res*. 2017; 46:D851–D860. PMID: [29112715](#)
69. Tanizawa Y, Fujisawa T, Nakamura Y. DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication. Hancock J, editor. *Bioinformatics*. 2018; 34:1037–1039. PMID: [29106469](#)
70. Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, et al. The RAST Server: rapid annotations using subsystems technology. *Bmc Genomics*. 2008; 9:75. <https://doi.org/10.1186/1471-2164-9-75> PMID: [18261238](#)
71. Overbeek R, Olson R, Pusch GD, Olsen GJ, Davis JJ, Disz T, et al. The SEED and the Rapid Annotation of microbial genomes using Subsystems Technology (RAST). *Nucleic Acids Res*. 2014; 42:D206–14. PMID: [24293654](#)
72. Brettin T, Davis JJ, Disz T, Edwards RA, Gerdes S, Olsen GJ, et al. RASTtk: a modular and extensible implementation of the RAST algorithm for building custom annotation pipelines and annotating batches of genomes. *Sci Rep-uk*. 2015; 5:8365. <https://doi.org/10.1038/srep08365> PMID: [25666585](#)
73. McNair K, Aziz RK, Pusch GD, Overbeek R, Dutilh BE, Edwards R. Phage Genome Annotation Using the RAST Pipeline. *Methods in molecular biology* (Clifton, NJ). 2018; 1681:231–238. https://doi.org/10.1007/978-1-4939-7343-9_17 PMID: [29134599](#)
74. Marc A, Cater K, Kongari R, Hatoum-Aslan A, Young RF, Liu M. Complete Genome Sequence of *Staphylococcus aureus* Siphophage Lorac. Dennehy JJ, editor. *Microbiol Resour Announc*. 2019; 8:603. <https://doi.org/10.1128/MRA.00586-19> PMID: [31270202](#)
75. Culbertson EK, Bari SMN, Dandu VS, Kriznik JM, Scopel SE, Stanley SP, et al. Draft Genome Sequences of *Staphylococcus* Podophages JBug18, Pike, Pontiff, and Pabna. Dennehy JJ, editor. *Microbiol Resour Announc*. 2019; 8:555. <https://doi.org/10.1128/mra.00054-19> PMID: [30834363](#)
76. Freeman ME, Kenny SE, Lanier A, Cater K, Wilhite MC, Gamble P, et al. Complete Genome Sequences of *Staphylococcus epidermidis* Myophages Quidividi, Terranova, and Twillingate. Dennehy JJ, editor. *Microbiol Resour Announc*. 2019; 8:555. <https://doi.org/10.1128/mra.00598-19> PMID: [31248994](#)