Published online 24 October 2006 *Nucleic Acids Research, 2006, Vol. 34, No. 20* ***5839–5851***
doi:10.1093/nar/gkl732

Phage_Finder: Automated identification and classification of prophage regions in complete bacterial genome sequences

Derrick E. Fouts*

The Institute for Genomic research, 9712 Medical Center Drive, Rockville, MD 20850, USA

Received July 27, 2006; Revised and Accepted September 21, 2006

ABSTRACT

***Phage_Finder*, a heuristic computer program, was created to identify prophage regions in completed bacterial genomes. Using a test dataset of 42 bacterial genomes whose prophages have been manually identified, *Phage_Finder* found 91% of the regions, resulting in 7% false positive and 9% false negative prophages. A search of 302 complete bacterial genomes predicted 403 putative prophage regions, accounting for 2.7% of the total bacterial DNA. Analysis of the 285 putative attachment sites revealed tRNAs are targets for integration slightly more frequently (33%) than intergenic (31%) or intragenic (28%) regions, while tmRNAs were targeted in 8% of the regions. The most popular tRNA targets were Arg, Leu, Ser and Thr. Mapping of the insertion point on a consensus tRNA molecule revealed novel insertion points on the 5′ side of the D loop, the 3′ side of the anticodon loop and the anticodon. A novel method of constructing phylogenetic trees of phages and prophages was developed based on the mean of the BLAST score ratio (BSR) of the phage/prophage proteomes. This method verified many known bacteriophage groups, making this a useful tool for predicting the relationships of prophages from bacterial genomes.**

INTRODUCTION

Bacteriophages (phages) are viruses that infect bacteria. Phages not only play important roles in the biology of their hosts, but they also have a major influence on the ecology of the oceans by cycling limiting nutrients (1) and boosting photosynthesis (2). Temperate phages (phages that integrate into the host genome) can provide essential virulence and fitness factors, affecting metabolism, bacterial adhesion, colonization, invasion, spread, resistance to immune responses, exotoxin production, serum resistance, destruction of competing bacteria and resistance to antibiotics (3,4). These capabilities can be generated by the introduction of novel genes or by altering expression of existing genes. Phages have also contributed significantly to our understanding of many cellular processes and have been a source of countless enzymes used routinely in molecular biology and biotechnology. Furthermore, phages themselves (5) or proteins produced by them (6,7) may be used as antimicrobials to cure bacterial infections in humans and are being developed as household disinfectants.

More than 5000 phages have been classified since 1959 by electron microscopy (8), yet <3% of these have been completely sequenced and deposited in public databanks. Considering the estimate of $\sim 10^{30}$ phages in the world (9), the diversity of phages is likely to be great which means more complete phage genome sequences will be needed to fully comprehend the true extent of genetic diversity, the capacity for genetic mobilization/exchange and the evolution of bacteriophages. Until this happens, there is an enormous, poorly explored, publicly available resource of bacteriophage genomes within the complete sequence of bacterial genomes. The phage genomes that can be recovered in this way are the double-stranded DNA tailed phages in the order *Caudovirales* and single-stranded DNA filamentous phages in the order *Inoviridae* that are known to integrate into the genome of their host.

Current bacterial genome sequencing projects poorly identify and annotate regions of bacteriophage origin and there are no standard definitions for the classification of these regions. There are multiple reasons for this. Many groups identify phage regions by manual inspection (10). Some consider any region with matches to phage sequence a prophage, while others have a more stringent definition and require complete or nearly complete sets of phage genes within the region to be considered a prophage. The annotation of the genes within these regions is even more variable because each group has different criteria for gene annotation. One of the greatest problems with the identification of prophage regions is the enormous diversity within the phage population, which can be observed in the sequence

*Tel: +1 301 795 7874; Fax: +1 301 838 0208; Email: dfouts@tigr.org

of many of the complete phage genomes where >50% of open reading frames (ORFs) have no database match. Another problem is that some regions are not complete phages, but just contain mainly tail fibers and lytic enzymes that have been hijacked by the host and used as weaponry to fight off competitors. These regions would be considered bacteriocins and have been called by many different names, depending on the species of origin (i.e. monocin for *Listeria monocytogenes*, pyocin for *Pseudomonas aeruginosa*) (11). Some investigators try to rely on altered G + C content of the region or the disruption of genes (particularly, the targeting or tRNAs) as methods of defining the boundaries of prophage regions. We have shown that prophage regions do not always have atypical G + C nucleotide composition (12), so this is not a reliable method. Phages do not always integrate into coding regions and do not exclusively use tRNAs as the target site for integration, making scanning for disrupted genes or tRNAs as a standalone method inadequate for finding prophage regions.

Phage_Finder was developed as a heuristic computer program to identify prophage regions within bacterial genomes and is freely available (http://www.tigr.org/software/ or http://phage-finder.sourceforge.net). It uses tab-delimited results from NCBI BLASTALL (13) or WU BLASTP 2.0 (http://blast.wustl.edu) (14) searches against a collection of bacteriophage sequences and results from HMMSEARCH (15) analysis of 441 phage-specific hidden Markov models (HMMs) to locate prophage regions. By using FASTA33 (16), MUMMER (17) or BLASTN (14), it can find potential attachment (att) sites of the phage region(s). Data from tRNAscan-SE (18) and Aragorn (19) are used to determine whether a tRNA or tmRNA served as the putative target for integration. Additionally, by looking for the presence or absence of specific proteins using HMMs, *Phage_Finder* can predict whether the region is most likely prophage and which type (Mu, P2, or retron R73), an integrated element, a plasmid, or a degenerate phage region. The pipeline was tested against a set of manually-defined prophage regions (20). *Phage_Finder* found 91% of these regions, resulting in 7% false positives and 9% false negatives with a test dataset using default settings (Table 2). To test the robustness of the pipeline, 302 complete bacterial genomes were processed through the pipeline. A total of 403 putative prophage regions were identified, which accounted for 1.7% of the total bacterial DNA. In addition to finding prophage regions, *Phage_Finder* has found integrated elements and integrated plasmids. A novel method for constructing phylogenetic trees of phages and prophages was developed based on the mean of the BLAST score ratio (BSR) of bacteriophage proteins.

MATERIALS AND METHODS

System and software requirements

Phage_Finder was written in PERL (http://www.perl.org) and tested using PERL version 5.8.5+ on Linux and Mac OS X 10.3/10.4 operating systems, but should work in most Unix environments if the following helper programs are installed and functional: NCBI BLASTALL (13) or WUBLAST 2.0 (http://blast.wustl.edu) (14) for BLAST searching, HMMSEARCH (15) to find HMM matches, tRNAscan-SE (18) to find the location of tRNAs, Aragorn (19) to locate tmRNA sequences, and FASTA33 (16), MUMMER (17), or BLASTN (13) to find att sites. *Phage_Finder.pl* utilizes the Math::Round PERL module that was written by Geoffrey Rommel to round numbers by defined multiples and is freely available (http://search.cpan.org/~grommel/Math-Round-0.05/Round.pm).

Input requirements

If running the included *Phage_Finder.sh* BASH script, the input requirements are as follows: name of file containing the protein sequences of the bacterial genome to be searched in FASTA format, name of BLAST-formatted phage protein sequences, the name of the file containing the DNA sequence of the entire bacterial genome to be searched (not the coding sequences), and the *Phage_Finder* information file (tab-delimited: contig_ID, size_of_contig, feat_name, end5, end3, annotation) or a GenBank .ptt file.

If invoking *Phage_Finder.pl* directly, then the following tab-delimited files are required for full functionality: NCBI or WU BLASTP data, HMMSEARCH data, tRNAscan-SE data, Aragorn data and either a *Phage_Finder* information file or GenBank .ptt file. If the HMM data is not provided then, the search for att sites will not be performed. If the data from tRNAscan-SE/Aragorn is not provided, then the *Phage_Finder.pl* will not associate any putative target-site duplications with *tRNA*/*tmRNA* genes.

Identification of prophage regions

One of the original intentions of *Phage_Finder* was to have a program that can distinguish between largely intact, possibly functional prophages versus small regions or clusters of prophage remnants and other mobile elements. It takes advantage of several features of functional prophages to filter out unwanted fragmented regions. Since functional temperate phages integrate as linear molecules in a size range of 18–150 kb, good candidate prophage regions should have clusters of phage-like genes in this size range. Functional phages also have a large fraction of hypothetical or conserved hypothetical proteins. These stretches of phage-like and unknown genes are consecutive, not broken up by operons of house-keeping genes, although an occasional metabolic enzyme can be encoded on a phage. Tailed phages will have a conserved late gene operon that is responsible for packaging and head morphogenesis (21–23). This conserved region includes a small and large terminase subunit to recognize pac or cos sites and to cleave phage genome concatemers for packaging of the phage genome into capsids (24); a portal protein to form a hole for passage of the phage genome during packaging and release (25); a prohead protease to generate mature capsids (23,26); and the major capsid protein that forms the bacteriophage capsid or head (23). A functional prophage region will also lack ribosomal RNA sequences.

The boundaries of functional prophages that integrated into specific locations can be determined by locating a site-specific recombinase at one of the ends of the phage region. Phages can integrate into *tRNA*/*tmRNA* genes, other conserved genes or intragenic regions. Since many phages

and genetic elements tend to have an affinity for *tRNA* genes as the target for integration, any *tRNA*/*tmRNA* gene present within the putative phage region is checked first as a target for integration. The integrase can integrate into the anticodon-loop, the T-loop, or the 3′ end of the *tRNA*/*tmRNA* gene (27). The phage genome will contain sequence near the integrase gene that is homologous to the 3′ part of its target to avoid inactivating the gene after insertion (28). Following integration, the target gene will be a fusion with the 5′ end being of bacterial origin and the 3′ end of phage origin and the original bacterial-derived 3′ end on the other side of the inserted phage genome. By searching the other end of the putative phage region with the sequence of the *tRNA*/*tmRNA* gene, including extra sequence in case of miscalculation of the boundaries, one can identify what looks like a target-site duplication, the sequence of the replaced 3′ end of the *tRNA*/*tmRNA* gene. This homologous sequence, flanking the genome of the integrated phage genome, is referred to as the core attachment site (att_{core}) and the two half sites are *attL* (the phage-derived sequence) and *attR* (the original bacterial sequence). Sequences that are 3′ of the *tRNA*/*tmRNA* gene can also be part of the att_{core} (27).

Phage_Finder overview

The first analysis that *Phage_Finder* does is to count the number of valid BLASTP phage matches within a user-defined window size, sliding by a user-defined step size until the size of the genome is reached. I have determined that a window size of 10 000 bp and a step size of 5000 bp are optimal settings for defining clusters of phage hits with the least amount of noise. These setting are the default window and step size settings. The center of each prophage region is then defined as the window with the greatest number of phage database matches in a region that begins with a window having at least the user-defined number of hits per window (default is four hits per window).

If at least one region is found within the minimum number of hits per window, then the 5′ and 3′ boundaries of each region is roughly determined. Beginning with the previously defined center of each region, *Phage_Finder.pl* slides gene by gene toward the 5′ and 3′ ends of the region, making a decision about inclusion. The decision to include a gene within a particular phage region is made in the following order: (i) if the protein has a phage HMM hit, then include; (ii) if it has a phage database BLASTP valid match, then include; (iii) if the gene is a *tRNA* or *tmRNA*, then include; (iv) if either of the next three genes are tRNAs or tmRNAs, then include; (v) if the gene has annotation that has been observed in known phages and there are at least three valid BLASTP phage database matches in the current window, then include (this 'ok annotation' is described below); (vi) if the gene has annotation that has been observed in known phages and there are at least two valid BLASTP phage database matches in the next window, then include; (vii) if there are at least three valid BLASTP phage database matches in the next window, then include; lastly, (viii) if there are at least two valid BLASTP matches in the current step and the current gene is before the matching gene, then include every gene up to the gene with the database match. If a putative prophage region contained valid HMM matches to XerC/D (TIGR02224 and TIGR02225) or integron (TIGR02249) integrases, then these regions would be excluded from further consideration.

Before further defining the boundaries of each putative prophage region, the program attempts to define whether the regions are type prophage or type bacteriocin and whether regions can be further classified as Mu-like, retron phage R73-like, P2-like or P4-like. Degenerate regions are determined after analysis of putative attachment sites. A region is defined as type prophage if: (i) it has a core HMM match or (ii) it has a lytic HMM match and a tail HMM match and an integrase HMM match. If the region has a lytic HMM match and a tail HMM match, but no integrase HMM match, then the region is defined as type bacteriocin, which is analogous to the phage-like bacteriocins (pyocins) in *P.aeruginosa* or monocins in *L.monocytogenes* (11). The R-type and F-type pyocins are probably the best studied, being defined phenotypically and genetically (29). These pyocins encode headless phage tails, regulatory proteins and lysis proteins for the production and release of phage tails that are specific to closely related *Pseudomonads*, resulting in destruction of the target cell through membrane disruption (30). The region is sub classified as Mu-like if there are proteins within the region that match the following Mu-specific HMMs: PF02316 (Mu DNA-binding domain), PF02914 (Bacteriophage Mu transposase), PF06074 [labeled as protein of unknown function (DUF935), but matches the Mu portal, *gp29*] and PF07030 [phage conserved hypothetical protein (DUF1320), matching Mu *gp36*]. PF06074 and PF07030 were determined to be Mu-specific by searching the phage database with these models and via mapping to single-linkage clusters of the phage database searched against itself by BLASTP (data not shown). Only Mu-like phages were hit by these HMMs. A combination of specific HMM matches and region length were used to distinguish between the retron phage R73, P2 and P4. The retron phage R73 is a P4-like cryptic prophage from a clinical *Escherichia coli* isolate, containing a retroelement (31). Bacteriophage P4 is a satellite phage that can use the head and tail genes of coliphage P2 to package inself into infectious viral particles (32). All three of these phages can be identified as having HMM hits to PF04606 (Phage transcriptional activator, Ogr/Delta) and TIGR01613 (phage/plasmid primase, P4 family, C-terminal domain) and an integrase match. The region is classified as retron R73-like if there is a match to PF00078 [Reverse transcriptase (RNA-dependent DNA polymerase)]. If the length of the region is >25 000 bp, then it is considered P2-like and P4-like if the size is <15 000 bp.

Provided that a file containing the DNA sequence of the genome and HMM or tRNA/tmRNA data was provided and the region is not Mu-like, a search for putative phage attachment sites is conducted. The user can specify BLASTN, FASTA33 or MUMMER to do the nucleotide similarity searches. BLASTN and FASTA33 have the advantage in that imperfect direct repeats can be identified, while MUMMER only looks for exact matches. BLASTN is the default nucleotide similarity search tool because it appears to do a better job of finding more significant matches. Only the top two matches are processed.

If *tRNA*/*tmRNA* gene(s) are within a putative prophage region, the sequence of the inmost *tRNA* gene is used as

the query to search the remaining 20% of the region plus 15 000 additional nucleotides for a similar direct repeat. If a direct repeat is identified, then *Phage_Finder* attempts to extend the region of homology by searching against the *tRNA*/*tmRNA* sequence plus 200 additional nucleotides at the 3′ end of the *tRNA*/*tmRNA* gene.

If no homologies are found, the program will identify the outermost integrase gene. The sequence beginning 1 nt outside the outermost end of the integrase gene and extending outside of the phage region by 400 nt is used as the query to search the remaining 20% of the region plus 15 000 additional nucleotides for a similar direct repeat. If a putative att site is identified, the coordinates are used to identify a putative target gene. Analysis of the distance between integrase and att sites of known phages was conducted to determine that 400 nt are sufficient sequence to include the att site. The coordinates of *tRNA*/*tmRNA* genes are used to determine whether a putative att site targeted a *tRNA*/*tmRNA* gene, in case the incorrect tRNA was chosen from a series of multiple tRNAs.

The final step is to format the data for printing a summary to the terminal and multiple output files to disk. If there are more contigs/assemblies to analyze, the program begins by finding clusters of phage database BLASTP matches on the next contig. This is repeated until there are no further contigs to search.

Input. *Phage_Finder* begins by checking for the two required data files: NCBI or WUBLAST BLASTP tab-delimited data where the query bacterial genome protein sequences were searched against a BLAST-formatted database of bacteriophage protein sequences and the *Phage_Finder* information file or GenBank .ptt file (described above). If both files are given and present, then the data from the *Phage_Finder* information file or GenBank .ptt file is processed, otherwise, the program terminates with a help menu. A flowchart of *Phage_Finder* logic is presented in Figure 1. If provided, data from tRNAscan-SE and Aragorn are processed and stored for future use. Other files, that change infrequently, are stored in the *Phage_Finder* home directory and are hard-coded into the program.

A list of satisfactory gene annotations is read from a file so the program can differentiate between 'house-keeping' genes and those genes that have been previously associated with functional phages or can be associated with phages (i.e. hypothetical proteins, conserved proteins and regulatory proteins). This list of 'ok annotation' was generated by parsing the annotation associated with every phage sequence in the phage database used in BLAST searches, removing spaces and making non-redundant. This also allows those genes that are actually phage-derived, but not matching anything in the current, limited phage database, to be included in a phage region. If orthologs of every phage protein were already in the database, this step would not be necessary.

Because orthologs of some proteins that are found in the genomes of functional phages are also present in bacterial genomes in non-phage regions, matches to these proteins are not specific to phage regions and must be excluded from analysis. Examples of proteins that fall into this category are transposases (non-Mu-like), ABC transporters, ribonucleotide reductase and certain other enzymes

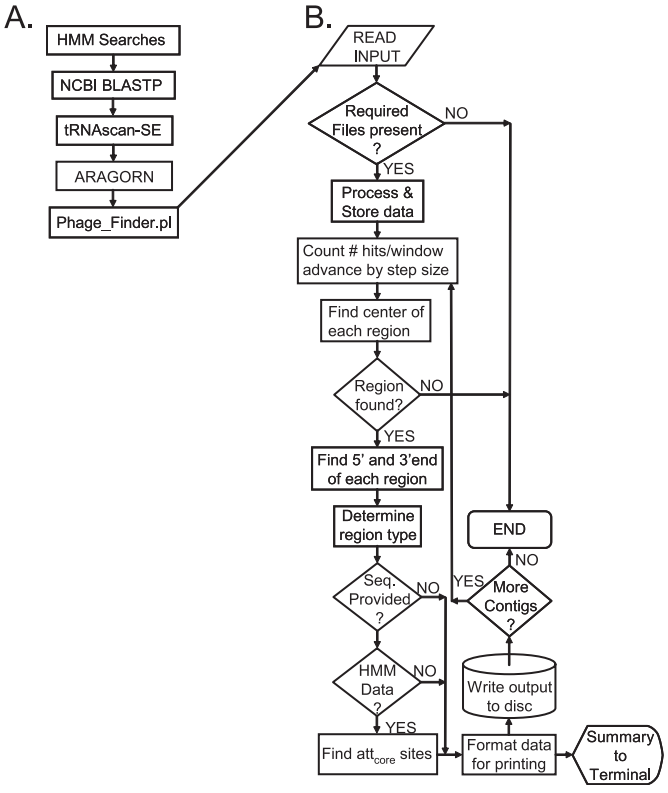


Figure 1. Flow chart of *Phage_Finder* pipeline (**A**) and *Phage_Finder.pl* script (**B**) logic. Standard symbols for constructing flow charts were used.

commonly encoded in the genomes of lytic phages. The accession numbers of these proteins are stored in the phage_exclude.lst file, which is used by *Phage_Finder.pl* to exclude these protein database matches from analysis.

A list of core phage HMMs as well as lists of HMMs that are specific for phage lysis genes and tail proteins are read in from separate files (Table 1). Matches to these HMMs as well as the integrase HMMs (PF00239, PF00589, or PF02899) are used to characterize the putative prophage regions as either prophage or bacteriocin. Matches to other specific phage HMMs and the length of the region are used to distinguish between Mu-like regions, retron phage R73, P2 and P4.

Tab-delimited data from BLASTP searches against a database of bacteriophage protein sequences is read in and processed. Data from NCBI BLASTP option −m 8 or WUBLASTP that has been processed with the BTAB BLAST output parser (33) are acceptable formats. Only valid matches are considered for further analysis. A valid match is the top or best match whose subject accession number is not in the exclude list and has an *E*-value less than or equal to the specified cut-off (default is 10^{-6}).

The results of phage-specific HMM searches are then processed. Protein sequences are searched with HMMSEARCH against 441 total models [295 glocal-mode (built with hmmls mode) and 146 fragment-mode (built with hmmfs mode)]. For an explanation of the glocal/hmmls and fragment/hmmfs alignment modes, please refer to the HMMER User's Guide (http://hmmer.wustl.edu/). Eight Pfam fragment-mode models were removed due to frequent matches to non-phage

Table 1. List of HMMs used to categorize putative prophage regions

Name	Description
Core phage HMMs	
Large terminase	
PF03354	Terminase_1: phage terminase, large subunit, putative
PF04466	Terminase_3: Phage terminase large subunit
PF05876	Terminase_GpA: Phage terminase large subunit (GpA)
PF06056	Terminase_5: Putative ATPase subunit of terminase (gpP-like)
PF07570	Protein of unknown function (DUF1545)
TIGR01547	phage_term_2: phage terminase, large subunit, PBSX family
TIGR01630	psiM2_ORF9: phage uncharacterized protein, C-terminal domain
Small terminase	
PF03592	Terminase_2: Terminase small subunit
PF05119	Terminase_4: Phage terminase, small subunit
PF05944	Phage_term_smal: Phage small terminase subunit
PF07141	Phage_term_sma: Putative bacteriophage terminase small subunit
PF07471	Phage_Nu1: Phage DNA packaging protein Nu1
TIGR01558	sm_term_P27: phage terminase, small subunit, putative, P27 family
Portal	
PF04860	Phage_portal: Phage portal protein
PF05133	Phage_prot_Gp6: Phage portal protein, SPP1 Gp6-like
PF05136	Phage_portal_2: Phage portal protein, lambda family
PF06074	DUF935: Protein of unknown function (DUF935)
TIGR01537	portal_HK97: phage portal protein, HK97 family
TIGR01538	portal_SPP1: phage portal protein, SPP1 family
TIGR01539	portal_lambda: phage portal protein, lambda family
TIGR01540	portal_PBSX: phage portal protein, PBSX family
TIGR01542	A118_put_portal: phage portal protein, putative, A118 family
Capsid/head/coat	
PF01819	Levi_coat: Levivirus coat protein
PF02305	Phage_F: Capsid protein (F protein)
PF03864	Phage_cap_E: Phage major capsid protein E
PF05065	Phage_capsid: Phage capsid family
PF05125	Phage_cap_P2: Phage major capsid protein, P2 family
PF05126	Phage_min_cap: Phage minor capsid protein
PF05356	Phage_Coat_B: Phage Coat protein B
PF05357	Phage_Coat_A: Phage Coat Protein A
PF05371	Phage_Coat_Gp8: Phage major coat protein, Gp8
PF06673	Phage_min_cap2: Phage minor capsid protein 2
PF07068	L_lactis_ph-MCP: *Lactococcus lactis* bacteriophage major capsid protein
TIGR01551	major_capsid_P2: phage major capsid protein, P2 family
TIGR01554	major_cap_HK97: phage major capsid protein, HK97 family
Capsid prot.	
PF03420	Peptidase_U9: Prohead core protein protease, T4 family
PF04586	Caudo_protease: Caudovirus prohead protease
TIGR01543	proheadase_HK97: phage prohead protease, HK97 family
Head-tail joining	
PF02831	gpW: gpW [head-tail-joining]
PF05352	Phage_connector: Phage Connector (GP10)
PF05354	Phage_attach: Phage Head-Tail Attachment
PF05521	Phage_H_T_join: Phage head-tail joining protein
PF06264	DUF1026: Protein of unknown function (DUF1026)
TIGR01563	gp16_SPP1: phage head-tail adaptor, putative
Tape measure	
PF06120	Phage_HK97_TLTM: Tail length tape measure protein
PF06791	TMP_2: Prophage tail length tape measure protein
TIGR01541	tape_meas_lam_C: phage tail tape measure protein, lambda family
TIGR01760	tape_meas_TP901: phage tail tape measure protein, TP901 family, core region
Virion morphogenesis	
PF02924	HDPD: Bacteriophage lambda head decoration protein D
PF02925	gpD: Bacteriophage scaffolding protein D
PF03863	Phage_mat-A: Phage maturation protein
PF04233	Phage_Mu_F: Phage Mu protein F like protein
PF05396	Phage_T7_Capsid: Phage T7 capsid assembly protein
PF05926	Phage_GPL: Phage head completion protein (GPL)
PF05929	Phage_GPO: Phage capsid scaffolding protein (GPO)
PF07230	Phage_T4_Gp20: Bacteriophage T4-like capsid assembly protein (Gp20)
TIGR01641	phageSPP1_gp7: phage putative head morphogenesis protein, SPP1 gp7 family
Other functions	
PF02914	Mu_transposase: Bacteriophage Mu transposase
PF03374	ANT: Phage antirepressor protein
PF04687	Microvir_H: Microvirus H protein (pilot protein)
PF05135	Phage_QLRG: Phage QLRG family, putative DNA packaging
PF05435	Phi-29_GP3: Phi-29 DNA terminal protein GP3
PF05894	Podovirus_Gp16: Podovirus DNA encapsidation protein (Gp16) [terminal protein]
PF07026	DUF1317: phage conserved hypothetical protein
PF07030	DUF1320: phage conserved hypothetical protein
PF07880	T4_gp9_10: Bacteriophage T4 gp9/10-like protein [baseplate]
TIGR01560	put_DNA_pack: uncharacterized phage protein (possible DNA packaging)
TIGR02215	phage_chp_gp8: phage conserved hypothetical protein, phiE125 gp8 family
Other phage HMMs	
Lysis	
PF00959	Phage_lysozyme: Phage lysozyme
PF01464	SLT: Transglycosylase SLT domain
PF01473	CW_binding_1: Putative cell wall binding repeat
PF03245	Phage_lysis: Bacteriophage lysis protein
PF04517	Microvir_lysis: Microvirus lysis protein (E), C terminus
PF04531	Phage_holin_1: Bacteriophage holin
PF04550	Phage_holin_2: Phage holin family 2
PF04688	Phage_holin: Phage lysis protein, holin
PF04936	DUF658: Protein of unknown function (DUF 658)
PF05102	Holin_BlyA: holin, BlyA family
PF05105	Phage_holin_4: Holin family
PF05106	Phage_holin_3: Phage holin family (Lysis protein S)
PF05289	BLYB: Borrelia hemolysin accessory protein [holin]
PF05382	Amidase_5: Bacteriophage peptidoglycan hydrolase
PF05449	DUF754: Protein of unknown function (DUF754)
PF06714	Gp5_OB: Gp5 N-terminal OB domain
PF06715	Gp5_C: Gp5 C-terminal repeat (3 copies)
PF06737	Transglycosylas: Transglycosylase-like domain
PF06946	Phage_holin_5: Phage holin
PF07066	Phage_Lacto_M3: Lactococcus phage M3 protein
TIGR01592	holin_SPP1: holin, SPP1 family
TIGR01593	holin_tox_secr: toxin secretion/phage lysis holin
TIGR01594	holin_lambda: phage holin, lambda family
TIGR01598	holin_phiLC3: holin, phage phi LC3 family
TIGR01606	holin_BlyA: holin, BlyA family
TIGR01673	holin_LLH: phage holin, LL-H family
Tails/tail fibers	
PF02306	Phage_G: Major spike protein (G protein)
PF02413	Caudo_TAP: Domain of unknown function DUF144
PF03335	Phage_fiber: Phage tail fiber repeat
PF03406	Phage_fiber_2: Phage tail fiber repeat
PF03903	Phage_T4_gp36: Phage T4 tail fibre
PF03906	Phage_T7_tail: Phage T7 tail fiber protein
PF04630	Phage_tail: Phage major tail protein
PF04717	Phage_base_V: Phage-related baseplate assembly protein
PF04865	Baseplate_J: Baseplate J-like protein
PF04883	DUF646: Bacteriophage protein of unknown function (DUF646)
PF04984	Phage_sheath_1: Phage tail sheath protein
PF04985	Phage_tube: Phage tail tube protein FII
PF05017	TMP: TMP repeat
PF05069	Phage_tail_S: Phage virion morphogenesis family
PF05100	Phage_tail_L: Phage minor tail protein L

Table 1. *Continued*

Name	Description
PF05268	GP38: Phage tail fibre adhesin Gp38
PF05489	Phage_tail_X: Phage Tail Protein X
PF05939	Phage_min_tail: Phage minor tail protein
PF06141	Phage_tail_U: Phage minor tail protein U
PF06158	Phage_E: Phage tail protein E
PF06199	Phage_tail_2: Phage major tail protein 2
PF06222	Phage_TAC: Phage tail assembly chaperone
PF06223	Phage_tail_T: Minor tail protein T
PF06274	Mu-like_GpL: Bacteriophage Mu tail sheath protein (GpL)
PF06341	DUF1056: Protein of unknown function (DUF1056)
F06488	L_lac_phage_MSP: *L.lactis* bacteriophage major structural protein
PF06528	Phage_P2_GpE: phage tail protein, P2 GpE family
PF06763	Minor_tail_Z: Prophage minor tail protein Z (GPZ)
PF06805	Lambda_tail_I: Bacteriophage lambda tail assembly protein I
PF06810	Phage_GP20: Phage minor structural protein GP20
PF06820	Phage_fiber_C: Putative prophage tail fibre C-terminus
PF06841	Phage_T4_gp19: T4-like virus tail tube protein gp19
PF06890	Phage_Mu_Gp45: Bacteriophage Mu Gp45 protein
PF06891	P2_Phage_GpR: P2 phage tail completion protein R (GpR)
PF06893	Phage_Mu_P: Bacteriophage Mu P protein
PF06894	Phage_lambd_GpG: Bacteriophage lambda minor tail protein (GpG)
PF06995	Phage_P2_GpU: Phage P2 GpU
PF07409	GP46: Phage protein GP46
PF07484	Collar: Phage Tail Collar Domain
TIGR01600	phage_tail_L: phage minor tail protein L
TIGR01603	maj_tail_phi13: phage major tail protein, phi13 family
TIGR01611	tail_tube: phage major tail tube protein
TIGR01633	phi3626_gp14_N: phage putative tail component, N-terminal domain
TIGR01634	tail_P2_I: phage tail protein I
TIGR01635	tail_comp_S: phage virion morphogenesis protein
TIGR01644	phage_P2_V: phage baseplate assembly protein V
TIGR01665	put_anti_recept: phage minor structural protein, N-terminal region
TIGR01674	phage_lambda_G: phage minor tail protein G
TIGR01715	phage_lam_T: phage tail assembly protein T
TIGR01725	phge_HK97_gp10: phage protein, HK97 gp10 family
TIGR02126	phgtail_TP901_1: phage major tail protein, TP901-1 family
TIGR02242	tail_TIGR02242: phage tail protein domain

proteins (PF05442, PF07352, PF05012, PF06992, PF06806, PF07068, PF05037 and PF07230). A valid HMM match is recorded if the total score is greater than the noise cut-off (for glocal-mode) or is greater than the trusted cut-off for the fragment-mode). Unfortunately, due to a lack of sequence diversity that was included in the HMM seed for several glocal-mode model Pfams, several reasonable matches were not used by *Phage_Finder.pl* because the total score was between the noise and trusted cut-offs. There were five Pfam glocal-mode models (PF00589, PF02316, PF02914, PF06074 and PF07030) where the noise cut-off had to be set to a lower value within the *Phage_Finder.pl* program to increase the number of valid matches to these models. This was not the case for the fragment-mode models, where only scores above the trusted cut-off appear reliable.

Output. *Phage_Finder* will generate at least eight different filetypes as output if a phage-like region is identified. These include the following: (i) a log file that gives useful information about how *Phage_Finder* processed the data and a summary of the findings; (ii) a file that can be imported into XGRAPH (http://www.xgraph.org/), which plots the number of phage matches to the database per window and step size; (iii) a tab-delimited report file that shows (coordinate incremented by the step size, # hits per window, and the feat_name or locus name of the hits); (iv) a file containing the 5′ end of each gene, tRNA or att site within each region, the name of the feature and the annotation/database match/HMM match as well as the G + C% content of each region, a best guess for the type of region and the coordinates of each region with or without att site adjustments. There are three different names for this file, depending on the size of the regions (1–10 000, 10 001–18 000 and >18 001 bp); (v) a tab-delimited file containing (contig_id, size of the genome, G + C% content of the genome, 5′ end of the phage region, 3′ end of the phage region, size of region in bp, label (small, medium, large), region type (prophage, integrated element, degenerate), sequence of attR, sequence of attL, name of integration target, G + C% of region, 5′ feat_name or locus name, 3′ feat_name or locus name, # integrase HMM hits, # core_HMM hits, # above noise core_HMM hits, # lytic gene HMM hits, # tail HMM hits, # Mu HMM hits, orientation of the prophage based on orientation of the target or the position of the integrase, the distance from att site to integrase, and the number of genes in the region; (vi) a file in FASTA format containing the DNA sequence of the phage region; (vii) a file in FASTA format containing the DNA sequence of each gene within the phage region; and (viii) a file in FASTA format containing the protein sequence of each gene within the phage region.

Calculation of distance from BSR

The BSR has been used to compare three genomes at a time (34). This approach can be expanded to compare *n* number of genomes by computing the average of BSRs between each genome. For the purpose of tree building, a Phylip-style distance matrix was required (35,36). BLASTP was used to identify bidirectional matching protein sequences as described previously (37). The BSR was calculated on all protein matches that met the prerequisites as described previously (37). Those proteins that failed to meet the prerequisites were given a BSR value of zero. Because the Phylip-style distance matrix uses a different numerical scale than the BSR, a simple calculation (Dab = −99.999999 × BSR +99.999999) was used to convert the BSR (1 = exact match, 0 = no match) into a phylip distance (0 = exact match, 99.999999 = no match). To ensure that Dab equals Dba for all the proteins between two genomes (a and b), the following calculation was used to compute the mean of the distance between genomes a and b: $D = (\sum \mathrm{Dab_i} + \sum \mathrm{Dba_i})/N$, where N is the total number of proteins in the subject and query genomes a and b. This number, D, for every genome pairwise combination was formatted as a Phylip-style distance matrix file that was used as input for the Phylip NEIGHBOR program.

Sources of genome sequence data and phage database

The majority of phage sequences that were included in the phage database were obtained from NCBI's phage page (http://www.ncbi.nlm.nih.gov/genomes/static/phg.html). Additional phage sequences were obtained from GenBank and some prophages were obtained from whole genome projects at TIGR. Complete bacterial genome sequences were obtained from NCBI (http://www.ncbi.nlm.nih.gov/genomes/static/eub_g.html).

RESULTS

Testing the program

Phage_Finder was written for the purpose of automated identification of prophage regions in bacterial genomes. As with any new piece of software, one would like to have a measure of performance—a benchmark. Because there are no other publicly available programs that perform a similar function, we were unable to benchmark *Phage_Finder* against other programs, as was done with tRNA-finding programs (19,38). Instead, the number of false positive and false negative prophage regions were calculated based on specific criteria and based on comparison to a manually curated set of prophages. Perhaps the best list of manually curated prophages was compiled by Sherwood Casjens (20). Rob Edwards provided the sequences and putative att sites of these prophages in Genbank format (http://phage.sdsu.edu/~rob/phage/). Using the two resources, *Phage_Finder.pl* was run on 42 bacterial genomes that had manually curated prophages with putative att sites. Of the 118 manually curated prophages, *Phage_Finder* found 107 (91%). This translates into 9% false negatives (11/118, Table 2). If the *E.coli*

Table 2. Testing *Phage_Finder* against a known dataset

Organism	Known # Prophage[a]	# ORFs	Predicted # Prophage[a]	# ORFs	# False +	# False −
Bacillus anthracis Ames	3	147	3	170	0	0
Bacillus halodurans C-125	1	44	1	44	0	0
Bacillus subtilus 168[b]	2	247	2	162	1	0
Bifidobacterium longum NCC2705 1	19	1	19	0	0	
C.jejuni RM1221	1	57	1	57	0	0
Clostridium perfringens 13	1	44	1	42	0	0
Clostridium tetani E88	2	68	2	69	0	0
D.vulgaris Hildenborough	4	201	4	207	0	0
Enterococcus faecalis V583	5	288	5	268	0	0
E.coli CFT073	5	298	3	200	0	2
E.coli K-12	4	98	4	97	0	0
E.coli O157:H7 EDL933	10	429	8	472	0	2
E.coli O157:H7 VT-2 Sakai	11	598	10	677	0	1
L.lactis IL1403	6	254	4	240	0	2
Listeria innocua CLIP 11262	5	302	5	321	0	0
L.monocytogenes EGD-e	1	62	1	64	0	0
Mesorhizobium loti MAFF303099	2	95	1	36	0	1
Methylococcus capsulatus Bath	1	58	1	55	0	0
Mycobacterium tuberculosis CDC1551	1	14	1	14	0	0
M.tuberculosis H37Rv	2	29	2	36	0	0
P.putida KT2440	2	125	2	125	0	0
Pseudomonas syringae DC3000	1	45	1	45	0	0
Ralstonia solanacearum GMI1000	3	120	3	146	3	0
Salmonella enterica serovar typhi CT18	3	122	2	146	0	1
Salmonella typhimurium LT2	5	207	5	212	0	0
Shewanella oneidensis MR-1	1	75	1	75	0	0
Shigella flexneri 2a	2	39	2	64	2	0
Staphylococcus aureus COL	1	72	1	72	1	0
S.aureus Mu50	2	132	2	133	1	0
S.aureus MW2	2	123	2	151	0	0
S.aureus N315	1	65	1	90	0	0
Staphylococcus epidermidis RP62A	1	154	1	154	0	0
Streptococcus agalactiae 2603 V/R	2	112	2	112	0	0
S.pyogenes M1 GAS	4	171	3	157	0	1
S.pyogenes M3 MGAS315	6	338	6	359	0	0
S.pyogenes M18 MGAS8232	5	294	5	332	0	0
Treponema denticola ATCC 35405	1	41	1	41	0	0
Vibrio cholerae N16961	1	13	0	0	0	1
Xanthomonas axonopodis 306	2	51	2	97	0	0
Xanthomonas campestris ATCC33913	1	51	1	51	0	0
Xylella fastidiosa 9a5c	3	168	3	169	0	0
X.fastidiosa Temecula1	1	22	1	22	3	0
Total	118	5892	107	6003	11	11

[a]Only those regions with predicted att sites and contain core phage genes are listed.
[b]SPβ was split into two regions.

genomes were omitted, which have tandom (piggy-back) prophages and fragmented prophage regions, *Phage_Finder* found 82 of 88 (93%) of the manually curated prophage regions. Of the remaining six false negative prophages not reported by *Phage_Finder* using the default −S (strict mode) option, five of these regions were found when omitting the strict option and one was piggy-back with another phage. The piggy-back phages are either fused into one large phage region or one of the two phages is not reported. The number of false positive prophage regions was determined from the tab-delimited output file to be 7% (11/151), using default settings (Table 2). A region was considered false positive if the region was small and fragmented (<10 kb) or consisted mainly of transposons, restriction systems, transporters, plasmid-like genes or enzymes that could be host-derived in the absence of phage-specific genes.

Application of the program

After having demonstrated that a tool was produced that could identify prophage regions >90% of time, it was time to put the program to use on all completed bacterial genomes. At the time of writing, 302 complete bacterial genomes were available at NCBI. About half of all bacterial genomes processed contained a putative prophage region (154 out of 302). This accounted for 2.74% of the total bacterial chromosomal DNA or ~2.6 phage per genome. A little more than half of predicted prophage regions had putative att sites (285 out of 403). Mu-like phages accounted for roughly 8% of all predicted prophage regions.

Since there were so many putative attachment sites predicted, new questions could be asked that were not previously possible due to an insufficient amount of data. Do phage prefer tRNAs, tmRNAs, other genes, or intergenic regions as targets for integration? Put another way, what is the distribution of tRNA targets? What is the average distance from integrase to att site? Are certain tRNAs targeted more frequently? Which part of the tRNA do phages target? The distribution of 285 putative attachment sites is depicted as a pie chart in Figure 2A. It appears that tRNAs are targets for integration slightly more frequently (33%) than intergenic (31%) or intragenic (28%) regions, but because these numbers are so close, they each account for roughly a third of the total att sites. tmRNAs appear to be less prevalent as targets for prophage integration, accounting for 8% of the predicted prophage regions. As improved att site prediction algorithms are developed, these numbers will likely change. The most popular tRNA targets appear to be Arg, Leu, Ser and Thr (Figure 2B). The mapping of the insertion point on a consensus tRNA molecule revealed novel insertion points not previously characterized (27). In particular, the 5′ side of the D loop, the 3′ side of the anticodon loop and the anticodon were not previously known to be targeted (Figure 2C). When the frequency distribution was plotted linearly (Figure 2D), two distinct peaks were observed, spaced ~30 bp apart (or three turns of B-form DNA). These apparent 'hot spots' were confirmed independently (27).

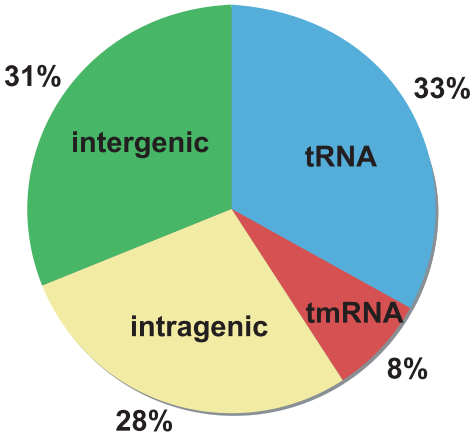
With the number of known phages and predicted prophages being well over 600, a novel way to display their relationships was needed. Rohwer and Edwards (39) developed a method of constructing phage trees using multiple sequence alignments, but with so many phages, a more rapid method of determining distance was required. The BSR has been used to compare up to three genomes at a time (34) and the mean of the BSR has been used to identify top phage matches (40). By converting the mean of the BSR of a phage proteome into a distance matrix, a tree can be generated. In-house PERL scripts were created to convert the BSR into a distance matrix. NEIGHBOR (35,36) was used to convert the distance matrix into a Neighbor-Joining tree (Figure 3). A test tree produced by this pipeline was compared against previously published *Campylobacter* 16S rRNA and concatenated protein trees and found to produce results similar to those produced using concatenated protein sequences (37). Given this benchmark, it is reasonable to believe that this method of tree building based on the BSR is a reliable method to infer genetic or evolutionary relationships.

A total of 679 phages, prophages and predicted prophages were processed with the mean BSR/distance matrix/Neighbor-Joining pipeline. PHYLODRAW (41) was used to generate an unrooted radial tree (Figure 4). Though this method most likely does not reflect true evolutionary decent, due to the mosaicism of phage genomes (42,43), some useful associations can be gained. There was clustering of known phages that do not integrate into the host chromosome and are known to be very similar, like the T4 and T5 phages and the Bam35 and PRD1 groups of Tectiviruses. As expected, there are no predicted or known prophages that cluster with these lytic phages. An exception to this is the *Pseudomonas putida* putative prophage Φ3, which clusters with the T7-like phages (Figure 4). It was previously shown that this *P.putida* prophage had best BLAST matches to T7-like phages (12). There are examples of clades or clusters (gold color, Figure 4) that are made up exclusively of predicted prophages from *Phage_Finder*, which suggests these prophage sequences are novel, lacking close relationship to known sequenced phages. There are a number of examples where the phage clusters agree with current knowledge, while there are also examples to the contrary. Specifically, most of the Mu-like phages cluster together, except for *Deinococcus radiodurans* RadMu and *Campylobacter jejuni* CMLP1 (Figure 4). The previously published c2-like and sk1-like relationships were confirmed (43); however, the Sfi21- and Sfi11-like phages were broken into separate groups. The K1-specific podophages K1F and K1E clustered with T7 and SP6, respectively, which agrees with previous observations (44). The Lambdoid phages λ, HK97, HK022 and N15 did not all cluster together. HK97 and HK022 clustered together as predicted (45), and *E.coli* N15 clustered with *Klebsiella oxytoca* ΦKO2 and *Yersinia enterocolitica* PY54 as previously shown (46). On the other hand, Lambda clustered with Coliphage 21 and with *E.coli* and *Shigella* predicted prophages instead of these other Lambdoid phages.

DISCUSSION

A software package has been described that searches complete bacterial genomes for the presence of bacteriophage-like regions and generates several output files. To test the accuracy of prediction, the program was ran against a set of

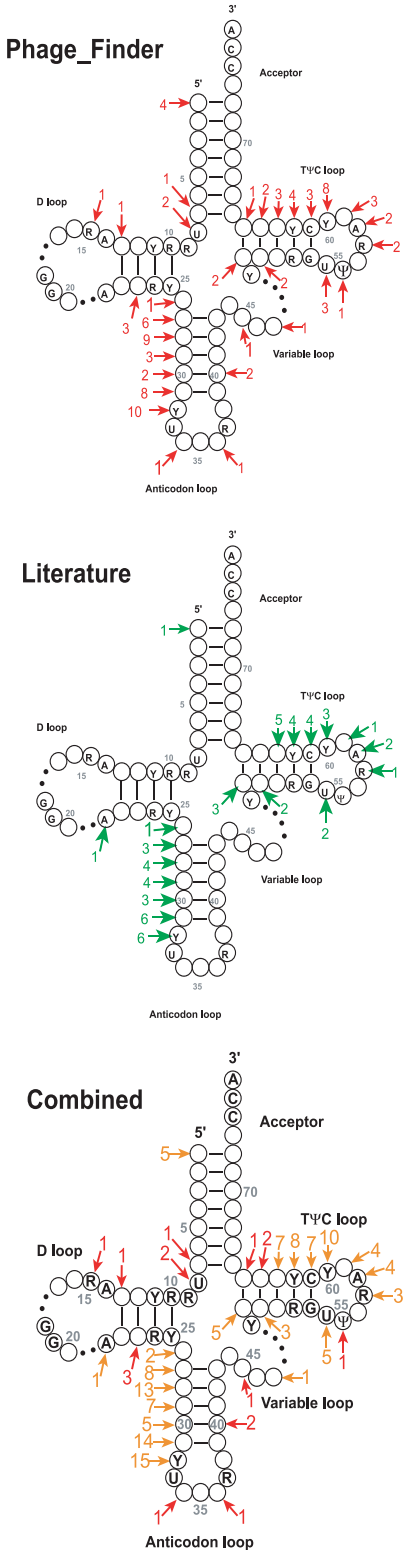
A.



B.

First position (5'end)	Second position								Third position (3' end)
	U		C		A		G		
U	Phe	0:0	Ser	0:0	Tyr	0:0	Cys	0:0	U
	Phe	4:2	Ser	1:4	Tyr	1:0	Cys	1:3	C
	Leu	1:3	Ser	5:3	Stop	0:0	Stop	1:0	A
	Leu	4:7	Ser	1:6	Stop	0:0	Trp	0:0	G
C	Leu	0:0	Pro	0:0	His	0:0	Arg	1:0	U
	Leu	1:2	Pro	0:4	His	0:1	Arg	0:0	C
	Leu	1:2	Pro	0:2	Gln	0:0	Arg	0:0	A
	Leu	1:0	Pro	2:0	Gln	0:1	Arg	1:2	G
A	Ile	0:0	Thr	0:0	Asn	0:0	Ser	0:0	U
	Ile	1:0	Thr	2:2	Asn	1:4	Ser	1:0	C
	Ile	0:0	Thr	1:2	Lys	1:5	Arg	4:10	A
	Met	1:3	Thr	4:7	Lys	0:2	Arg	5:6	G
G	Val	0:0	Ala	0:0	Asp	0:0	Gly	0:0	U
	Val	2:3	Ala	0:0	Asp	1:0	Gly	1:0	C
	Val	1:3	Ala	0:1	Glu	0:1	Gly	1:1	A
	Val	1:1	Ala	1:0	Glu	0:1	Gly	1:5	G

C.



D.

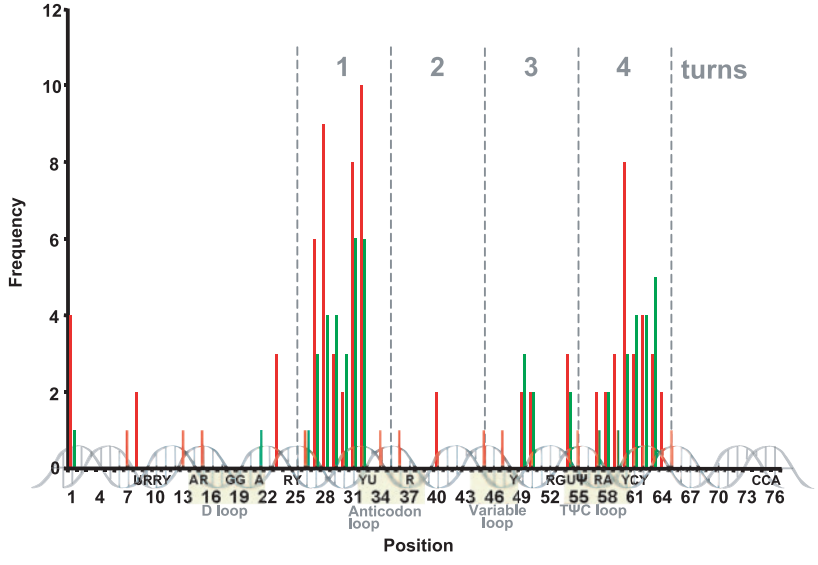


Figure 2. Predicted prophage target-site distributions. The distribution of targets where *Phage_Finder* found putative attachment sites (**A**). The genetic code table indicates the distribution of tRNA targets (**B**). For each codon, the number of phages from Williams, 2002 and the number of predicted prophages from this study are indicated, separated by a colon. The gray-highlighted numbers demarcate those codons that are targeted six or more times. The point of insertion on a consensus tRNA molecule was mapped (**C**) for *Phage_Finder* predicted prophages (upper), phages and prophages from the literature [(27), middle] and the two datasets combined (lower). The arrows point to the nucleotide insertion point while the numbers indicate number of insertions at each insertion point. Red arrows and numbers in the combined dataset show those locations that are unique to either dataset, while gold colored arrows and numbers highlight common insertion points between the two datasets. The frequency and position of insertion into a consensus tRNA gene is noted in (**D**). Red bars indicate *Phage_Finder*-predicted insertion events while green bars represent insertion events reported from the literature (27).

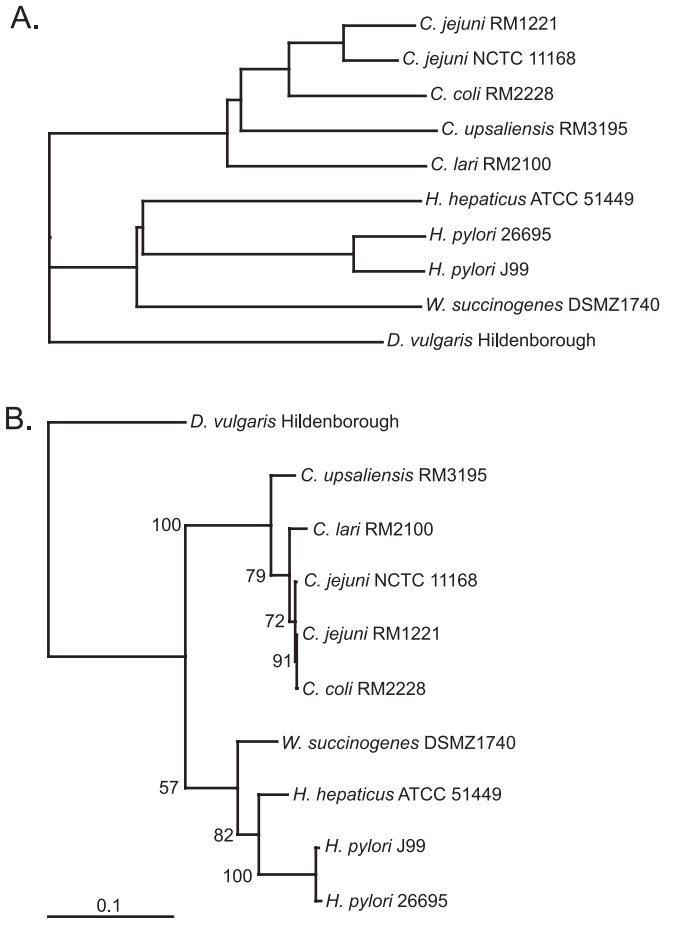


Figure 3. Test phylogenetic tree generated by converting the BSR of BLASTP bidirectional matches into distance (**A**). Whole genome BLASTP data from Fouts *et al*., 2005 (37) was used to compute this tree. The previously published 16S rRNA tree (**B**) is shown for comparison.

42 complete bacterial genomes that had manually curated prophages with putative attachment sites. *Phage_Finder* found all but 11 manually identified prophage regions. The missed regions fall into one of two categories (regions that did not meet the 'strict' definition and regions that have more than one prophage integrated into the same attachment site in tandom—so called 'piggy-back' prophages). The piggy-back phage regions were either fused into one large region or truncated. It is difficult for the program to tease these regions apart because there are multiple possible attachment sites (one pair marking the boundary of the first prophage and another pair marking the boundary of the entire tandem series). Since this happens infrequently (3% of the test dataset), future updates to the program will deal with this issue. There are still many putative prophage regions that do not have core HMM matches (large terminase, portal, major capsid), which could be the reason why some of the manually identified prophages were missed under the strict mode. As more HMMs are built to these important phage proteins, the robustness of phage detection will increase. The accuracy of boundary prediction was measured by calculating the difference in ORF counts (between the known and predict prophage regions) per bacterial genome (17 ORFs/genome, Table 2). Currently, the program chooses between one of the two top matches to either a tRNA sequence or the sequence extending 400 bp outside of the integrase gene. If the query sequence is a tRNA, then the longest att site is chosen, but if the query sequence was from near the integrase gene, then a different strategy is taken. The best att site is determined as one that is either in the 3′ end of a gene or has the longest length. Future updates will likely focus on a more sophisticated strategy for attachment site prediction, since the longest att site is not necessarily the one used by the phage integrase.

Never before the creation of this tool has it been possible to analyze such a large dataset of prophage sequences (447 prophage or prophage regions encoding 6171 proteins). Initial studies of the data in this report looked at the distribution of 285 putative attachment sites (Figure 2) and the distribution of 679 phages, prophages and predicted prophages (Figure 4). From data presented in Figure 2, it was concluded that tRNAs, intergenic and intragenic regions are targeted with about the same frequency (roughly a third of the time). tmRNAs were targeted infrequently at 8% of the time. The number of putative intergenic insertions may change over time as better attachment site prediction algorithms are developed. It is also possible that the number of intragenic insertions is artificially too high since very small regions of nucleotide similarity can result in inaccurate target site prediction. The region of the *tRNA* gene that is targeted is very specific (the 3′ end of the gene in most instances) and can be quite large (up to 121 bp), which increases the confidence level of prediction over intragenic regions. The 5′ side of the D loop, the 3′ side of the anticodon loop and the anticodon were not previously known to be targeted by site-specific integrases (27). It remains to be determined whether these integrases actually function by inserting into the predicted locations in the tRNA genes.

A new method for constructing phage trees was developed as a way to display the output of *Phage_Finder* and to determine whether such an approach might be useful as a way of grouping or classifying phages, prophages and predicted prophage regions from genomic sequences. Instead of using protein alignments (39), the mean of the BSR was used to generate a PHYLIP distance matrix. Since the test tree, consisting of *Campylobacter* sequences rooted with *Desulfovibrio vulgaris* (Figure 3) was able to generate a tree that was consistent with the published concatenated protein tree (37), the results of such analysis are credible. Further validity of this method came from the clustering of many known phages. One interesting result from Figure 4 was that RadMu did not cluster with other Mu-like phages. Perhaps RadMu is defective and has changed considerably such that it doesn't match any other phage very well. Alternatively, RadMu could represent a unique clade of Mu-like phage that has no other sequenced members. Though this method is potentially very exciting because it is less computationally intensive than creating whole protein alignments and because there is no universal phylogenetic marker for studying phage evolution, caution must be exercised since many phages are mosaics of other phages (9,42,45). However, this procedure does have utility for clustering related groups of phages and for classifying uncultured phages. Furthermore, in a number of examples (*Campylobacter*, *Listeria*, *Bacillus*, *Mycobacterium*, *Staphylococcus* and *Streptococcus*

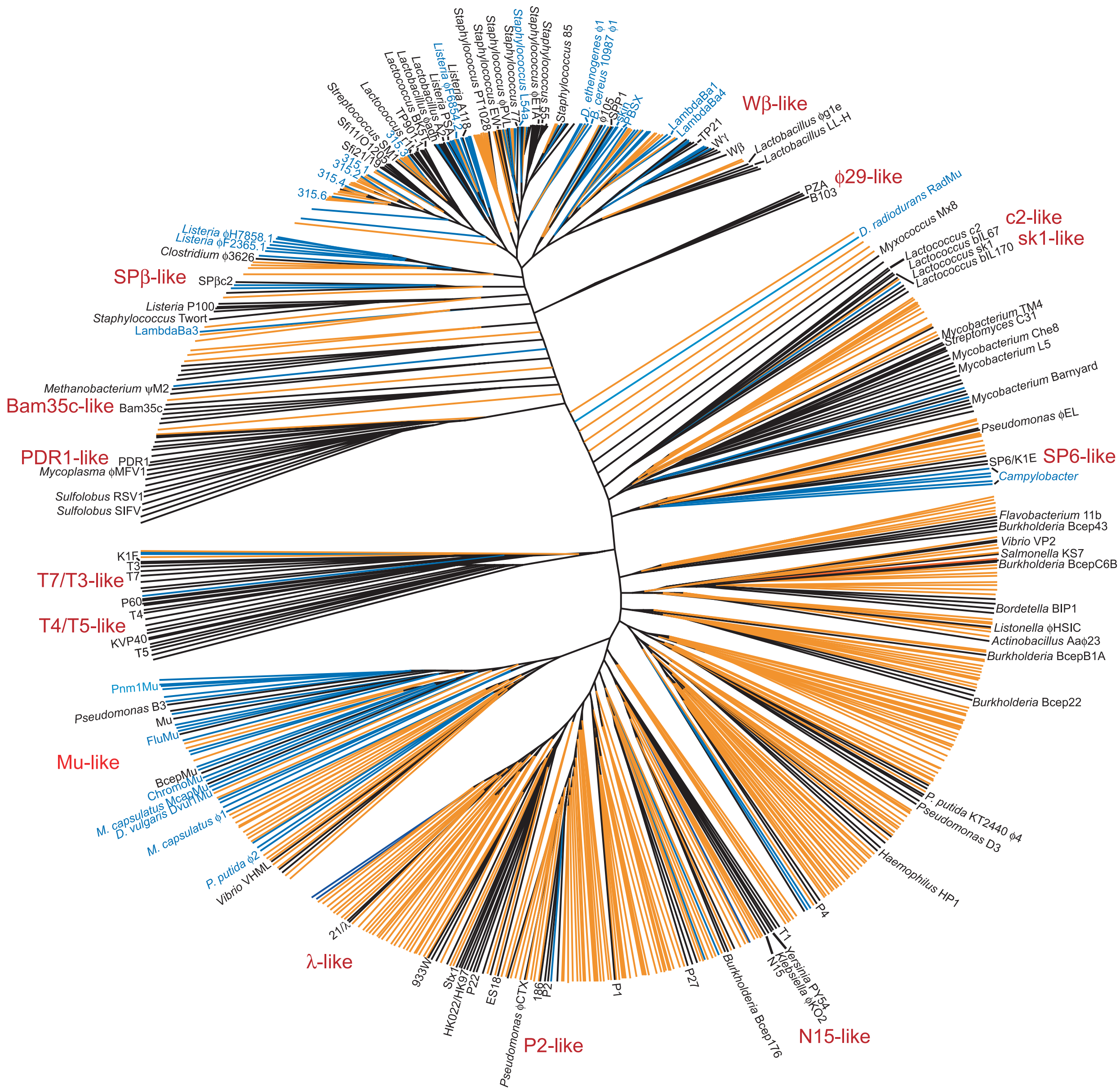


Figure 4. Phylogenetic analysis of *Phage_Finder* predicted prophages, known prophages and sequenced phage genomes. The radial tree was constructed with branch length extensions. The branches were colored as follows: sequenced phage genomes (black), known prophages (blue), *Phage_Finder* predicted prophage regions (gold). Only key phages or prophages are noted for clarity. Known phage groups are indicated in red.

phages and prophages), there is evidence to suggest coevolution of phages with host bacteria (Figure 4). Future versions might include either bonuses or penalties for synteny or the lack of synteny, which may resolve mosaic phages.

Phage_Finder was initially developed to aid in the identification of prophage regions in complete bacterial genomes and to improve annotation of these genomes by associating some level of function to the many hypothetical and conserved hypothetical proteins that are encoded in bacterial genomes. It has been very successful in these goals and has even surfaced a few surprises. For example, 10% of all proteins (20% of the hypothetical proteins) in the genome of *Enterococcus faecalis* V285 were within *Phage_Finder*-predicted putative prophage regions (D. E. Fouts, unpublished data). Furthermore, even though *E.coli* O157:H7 Sakai had the greatest number of predicted prophage regions (13 under strict settings), *Streptococcus pyogenes* MGAS315, with six predicted prophage regions, had the highest percentage (13.6%) of its genome as prophage DNA. Another surprise was how well *Phage_Finder* could identify genomic islands (pathogenecity islands, *mec* regions, integrated plasmids, which lead to the implementation of the 'strict' (-S) option. This raises the question of whether phages evolved from these mobile elements or whether these mobile elements evolved from phages. *Phage_Finder* was run on all the complete archael genomes, but found no prophage regions under any setting. It is not clear whether these archael genomes do not have integrated phages or whether the prophage regions are not detected because phages of this type are lacking from the BLAST-formated database.

The *Phage_Finder* pipeline will be a valuable tool for the scientific community and has been made publicly available (http://www.tigr.org/software/ or http://phage-finder.sourceforge.net). Future plans are to integrate this pipleline into the existing infrastructure at TIGR and to make the results of *Phage_Finder* analysis publicly available. It may also be possible to modify the program to identify integrated viruses in eukaryotic genomes, which would greatly facilitate the identification of retroviruses.

ACKNOWLEDGEMENTS

The author wishes to thank Timothy D. Read for encouragement to pursue this project, Karen E. Nelson for support, and Daniel H. Haft for construction of some core phage HMMs, the XerC/XerD HMMs (TIGR02224 and TIGR02225) and the integron-specific HMM TIGR02249. Special thanks also go to Rob Edwards for providing the sequences and putative att sites of a manually curated prophage test dataset (http://phage.sdsu.edu/~rob/phage/) and to Mihai Pop for help with PERL theory and for reviewing the manuscript. I also thank Robert T. DeBoy, Pawel Gajer, Gagan Pandya, and Pratap Venepally for critical discussions on tRNA insertion frequencies and Claudia Haywood for legal assistance with the open source license. Funding to pay the Open Access publication charges for this article was provided by NSF-EF-0412091.

Conflict of interest statement. None declared.

REFERENCES

1. Sullivan,M.B., Waterbury,J.B. and Chisholm,S.W. (2003) Cyanophages infecting the oceanic cyanobacterium *Prochlorococcus*. *Nature*, **424**, 1047–1051.
2. Mann,N.H., Cook,A., Millard,A., Bailey,S. and Clokie,M. (2003) Bacterial photosynthesis genes in a virus. *Nature*, **424**, 741.
3. Wagner,P.L. and Waldor,M.K. (2002) Bacteriophage control of bacterial virulence. *Infect. Immun.*, **70**, 3985–3993.
4. Schuch,R. and Fischetti,V.A. (2006) Detailed genomic analysis of the Wbeta and Gamma phages infecting *Bacillus anthracis*: implications for evolution of environmental fitness and antibiotic resistance. *J. Bacteriol.*, **188**, 3037–3051.
5. Stone,R. (2002) Bacteriophage therapy. Stalin's forgotten cure. *Science*, **298**, 728–731.
6. Schuch,R., Nelson,D. and Fischetti,V.A. (2002) A bacteriolytic agent that detects and kills *Bacillus anthracis*. *Nature*, **418**, 884–889.
7. Liu,J., Dehbi,M., Moeck,G., Arhin,F., Bauda,P., Bergeron,D., Callejo,M., Ferretti,V., Ha,N., Kwan,T. *et al.* (2004) Antimicrobial drug discovery through bacteriophage genomics. *Nat. Biotechnol.*, **22**, 185–191.
8. Ackermann,H.W. (2001) Frequency of morphological phage descriptions in the year 2000. Brief review. *Arch. Virol.*, **146**, 843–857.
9. Brüssow,H. and Hendrix,R.W. (2002) Phage genomics: small is beautiful. *Cell*, **108**, 13–16.
10. Fouts,D.E. (2004) In Fraser,C.M., Read,T.D. and Nelson,K.E. (eds.), *Microbial Genomes*. Humana Press Inc., Totowa, NJ, pp. 71–91.
11. Daw,M.A. and Falkiner,F.R. (1996) Bacteriocins: nature, function and structure. *Micron*, **27**, 467–479.
12. Nelson,K.E., Weinel,C., Paulsen,I.T., Dodson,R.J., Hilbert,H., Martins dos Santos,V.A., Fouts,D.E., Gill,S.R., Pop,M., Holmes,M. *et al.* (2002) Complete genome sequence and comparative analysis of the metabolically versatile *Pseudomonas putida* KT2440. *Environ. Microbiol.*, **4**, 799–808.
13. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
14. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
15. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
16. Pearson,W.R. (1990) Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.*, **183**, 63–98.
17. Delcher,A.L., Kasif,S., Fleischmann,R.D., Peterson,J., White,O. and Salzberg,S.L. (1999) Alignment of whole genomes. *Nucleic Acids Res.*, **27**, 2369–2376.
18. Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.*, **25**, 955–964.
19. Laslett,D. and Canback,B. (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Res.*, **32**, 11–16.
20. Casjens,S. (2003) Prophages and bacterial genomics: what have we learned so far? *Mol. Microbiol.*, **49**, 277–300.
21. Desiere,F., Lucchini,S. and Brüssow,H. (1999) Comparative sequence analysis of the DNA packaging, head, and tail morphogenesis modules in the temperate cos-site *Streptococcus thermophilus* bacteriophage Sfi21. *Virology*, **260**, 244–253.
22. Desiere,F., McShan,W.M., van Sinderen,D., Ferretti,J.J. and Brüssow,H. (2001) Comparative genomics reveals close genetic relationships between phages from dairy bacteria and pathogenic *Streptococci*: evolutionary implications for prophage-host interactions. *Virology*, **288**, 325–341.
23. Duda,R.L., Martincic,K. and Hendrix,R.W. (1995) Genetic basis of bacteriophage HK97 prohead assembly. *J. Mol. Biol.*, **247**, 636–647.
24. Ackermann,H.W. (1999) Tailed bacteriophages: the order *Caudovirales*. *Adv. Virus Res.*, **51**, 135–201.
25. Casjens,S. and Hendrix,R. (1988) In Calendar,R. (ed.), *The Bacteriophages*. Plenum Press, New York and London, Vol. 1, pp. 15–91.
26. Black,L.W., Showe,M.K. and Steven,A.C. (1994) In Karam,J. (ed.), *Molecular Biology of Bacteriophage T4*. American Society for Microbiology Press, Washington, DC, pp. 518–558.
27. Williams,K.P. (2002) Integration sites for genetic elements in prokaryotic tRNA and tmRNA genes: sublocation preference of integrase subfamilies. *Nucleic Acids Res.*, **30**, 866–875.
28. Zhao,S. and Williams,K.P. (2002) Integrative genetic element that reverses the usual target gene orientation. *J. Bacteriol.*, **184**, 859–860.
29. Nakayama,K., Takashima,K., Ishihara,H., Shinomiya,T., Kageyama,M., Kanaya,S., Ohnishi,M., Murata,T., Mori,H. and Hayashi,T. (2000) The R-type pyocin of *Pseudomonas aeruginosa* is related to P2 phage, and the F-type is related to lambda phage. *Mol. Microbiol.*, **38**, 213–231.
30. Uratani,Y. and Hoshino,T. (1984) Pyocin R1 inhibits active transport in *Pseudomonas aeruginosa* and depolarizes membrane potential. *J. Bacteriol.*, **157**, 632–636.
31. Sun,J., Inouye,M. and Inouye,S. (1991) Association of a retroelement with a P4-like cryptic prophage (retronphage ΦR73) integrated into the selenocystyl tRNA gene of *Escherichia coli*. *J. Bacteriol.*, **173**, 4171–4181.
32. Dehò,G. and Ghisotti,D. (2006) In Calendar,R. and Abedon,S.T. (eds.), *The Bacteriophages*. Oxford University Press, Inc., New York, Vol. 1, pp. 391–408.
33. Dubnick,M. (1992) Btab—a Blast output parser. *Comput. Appl. Biosci.*, **8**, 601–602.
34. Rasko,D.A., Myers,G.S. and Ravel,J. (2005) Visualization of comparative genomic analyses by BLAST score ratio. *BMC Bioinformatics*, **6**, 2.
35. Felsenstein,J. (1989) PHYLIP—Phylogeny Inference Package (Version 3.2). *Cladistics*, **5**, 164–166.
36. Felsenstein,J. (2006) PHYLIP (Phylogeny Inference Package) version 3.65.. Distributed by the author, Department of Genome Sciences, University of Washington, Seattle.
37. Fouts,D.E., Mongodin,E.F., Mandrell,R.E., Miller,W.G., Rasko,D.A., Ravel,J., Brinkac,L.M., DeBoy,R.T., Parker,C.T., Daugherty,S.C. *et al.* (2005) Major structural differences and novel potential virulence mechanisms from the genomes of multiple *Campylobacter* species. *PLoS Biol.*, **3**, e15.

38. Laslett,D., Canback,B. and Andersson,S. (2002) BRUCE: a program for the detection of transfer-messenger RNA genes in nucleotide sequences. *Nucleic Acids Res*., **30**, 3449–3453.
39. Rohwer,F. and Edwards,R. (2002) The Phage Proteomic Tree: a genome-based taxonomy for phage. *J. Bacteriol*., **184**, 4529–4535.
40. Fouts,D.E., Rasko,D.A., Cer,R.Z., Jiang,L., Fedorova,N.B., Shvartsbeyn,A., Vamathevan,J.J., Tallon,L., Althoff,R., Arbogast,T.S. *et al*. (2006) Sequencing *Bacillus anthracis* typing phages gamma and cherry reveals a common ancestry. *J. Bacteriol*., **188**, 3402–3408.
41. Choi,J.H., Jung,H.Y., Kim,H.S. and Cho,H.G. (2000) PhyloDraw: a phylogenetic tree drawing system. *Bioinformatics*, **16**, 1056–1058.
42. Lawrence,J.G., Hatfull,G.F. and Hendrix,R.W. (2002) Imbroglios of viral taxonomy: genetic exchange and failings of phenetic approaches. *J. Bacteriol*., **184**, 4891–4905.
43. Proux,C., van Sinderen,D., Suarez,J., Garcia,P., Ladero,V., Fitzgerald,G.F., Desiere,F. and Brussow,H. (2002) The dilemma of phage taxonomy illustrated by comparative genomics of Sfi21-like *Siphoviridae* in lactic acid bacteria. *J. Bacteriol*., **184**, 6026–6036.
44. Stummeyer,K., Schwarzer,D., Claus,H., Vogel,U., Gerardy-Schahn,R. and Muhlenhoff,M. (2006) Evolution of bacteriophages infecting encapsulated bacteria: lessons from *Escherichia coli* K1-specific phages. *Mol. Microbiol*., **60**, 1123–1135.
45. Juhala,R.J., Ford,M.E., Duda,R.L., Youlton,A., Hatfull,G.F. and Hendrix,R.W. (2000) Genomic sequences of bacteriophages HK97 and HK022: pervasive genetic mosaicism in the lambdoid bacteriophages. *J. Mol. Biol*., **299**, 27–51.
46. Casjens,S.R., Gilcrease,E.B., Huang,W.M., Bunny,K.L., Pedulla,M.L., Ford,M.E., Houtz,J.M., Hatfull,G.F. and Hendrix,R.W. (2004) The pKO2 linear plasmid prophage of *Klebsiella oxytoca*. *J. Bacteriol*., **186**, 1818–1832.