Bioinformatics, 35(21), 2019, 4402–4404
doi: 10.1093/bioinformatics/btz258
Advance Access Publication Date: 14 May 2019
Applications Note

OXFORD

Genome analysis

multiPhATE: bioinformatics pipeline for functional annotation of phage isolates

Carol L. Ecale Zhou[1,*], Stephanie Malfatti[1], Jeffrey Kimbrel[1], Casandra Philipson[2,3], Katelyn McNair[4], Theron Hamilton[2], Robert Edwards[4] and Brian Souza[1]

[1]Global Security Computing Applications Division, Lawrence Livermore National Laboratory, [2]Biological Defense Research Directorate, Naval Medical Research Center, Fort Detrick, MD 21702, USA, [3]Chemical and Biological Research, Defense Threat Reduction Agency and [4]Computational Sciences Research Center, San Diego State University, CA, USA

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on August 13, 2018; revised on March 15, 2019; editorial decision on April 3, 2019; accepted on May 3, 2019

Abstract

Summary: To address the need for improved phage annotation tools that scale, we created an automated throughput annotation pipeline: multiple-genome Phage Annotation Toolkit and Evaluator (multiPhATE). multiPhATE is a throughput pipeline driver that invokes an annotation pipeline (PhATE) across a user-specified set of phage genomes. This tool incorporates a *de novo* phage gene calling algorithm and assigns putative functions to gene calls using protein-, virus- and phage-centric databases. multiPhATE's modular construction allows the user to implement all or any portion of the analyses by acquiring local instances of the desired databases and specifying the desired analyses in a configuration file. We demonstrate multiPhATE by annotating two newly sequenced *Yersinia pestis* phage genomes. Within multiPhATE, the PhATE processing pipeline can be readily implemented across multiple processors, making it adaptable for throughput sequencing projects. Software documentation assists the user in configuring the system.

Availability and implementation: multiPhATE was implemented in Python 3.7, and runs as a command-line code under Linux or Unix. multiPhATE is freely available under an open-source BSD3 license from https://github.com/carolzhou/multiPhATE. Instructions for acquiring the databases and third-party codes used by multiPhATE are included in the distribution README file. Users may report bugs by submitting to the github issues page associated with the multiPhATE distribution.

Contact: zhou4@llnl.gov or carol.zhou@comcast.net

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A bacteriophage (also known as 'phage') is a virus that parasitizes a bacterium by infecting it and reproducing within it. This work was motivated by a need to increase the throughput potential for describing newly sequenced phage genomes. Global pathogen discovery efforts, such as The Global Virome Project (Carrol *et al.*, 2018), are projected to invest billions of dollars to support surveillance projects that characterize the earth's virosphere over the next 10 years. Already, the PhagesDB contains >13 000 phage genomes (Russell and Hatfull, 2017). Phage therapy has resurfaced as a method to combat antimicrobial resistance, and upcoming clinical trials necessitate complete sequencing and characterization of therapeutic candidates, but high-quality gene calling and functional annotation are vital for successful genomic comparison studies and for discovery of new phage-based therapeutic leads (Kutter *et al.*, 2015). Because annotation of phage genomes is a relatively new science, there exist few bioinformatics pipelines for phage analysis that can be readily adapted for use in phage research efforts.

Currently, researchers typically apply bacterial gene callers for annotation of phage DNA, followed by largely manual analyses using web forms, and integration of summary results can be time consuming. Although there exist several codes for identifying prophage sequences in bacterial genomes (Arndt *et al.*, 2016; Kang *et al.*, 2018; Roux *et al.*, 2015; and others), once these sequences have been identified, they are typically annotated using methods developed for sequences from other taxa (Perkel, 2017; Seemann, 2014). Currently there exists only one automated annotation pipeline specifically for phage: Philipson *et al.* (2018) describe a pipeline that identifies features in phage that determine their potential suitability as therapeutic reagents. However, there remains a need for an automated phage annotation pipeline that can be readily implemented on multiple nodes of a local server and that requires minimal software development expertise. To address this need, we present the multiple-genome Phage Annotation Toolkit and Evaluator (multiPhATE) automated high-throughput phage annotation pipeline.

2Description

The PhATE annotation pipeline incorporates four gene callers (if selected): GeneMarkS (Lomsadze *et al.*, 2017), Glimmer (Delcher *et al.*, 2007), Prodigal (Hyatt *et al.*, 2010) and a novel phage-centric gene caller, PHANOTATE (McNair *et al.*, 2019). Functional annotation is achieved by Basic Local Alignment Search Tool (BLAST) and Hidden Markov Model (HMM) searches for homologous sequences in protein- and phage-centric databases. The PhATE workflow is depicted in Supplementary File, 'phate_Fig_1_PhATE_Workflow.pdf'.

2.1 Input

Input to multiPhATE consists of a configuration file that specifies a list of genomes to be processed by PhATE and a set of parameters controlling software execution. The user specifies the names of phage genome fasta files, the names of output subdirectories and other metadata pertaining to the genomes being analyzed. The user also specifies the following optional analyses: (i) gene caller(s) to be run; (ii) gene-caller to use for subsequent annotation (default: PHANOTATE); (iii) blast parameters; (iv) blast databases to be searched; (v) turn hmm search on/off. It is possible to run PhATE using any or all of the specified gene callers, databases and searches. In this way, installation can be achieved one gene-caller or database at a time, with stepwise testing. Also, the user can switch on/off searches (e.g. NR) in order to control execution time (this may be useful in performing preliminary annotation of large numbers of sequences). Although multiPhATE is intended for phage sequence annotation, it would be reasonable to run multiPhATE with bacterial genomes to assist identification of embedded phage sequence.

2.2 Annotation

PhATE begins by performing gene calling using the selected gene caller(s). When two or more are invoked, PhATE outputs a summary table showing a side-by-side comparison of the gene calls, plus summary statistics regarding the numbers and lengths of gene calls for each algorithm, and the numbers of calls in common and unique to each. Next, PhATE uses BLAST+ programs (Camacho *et al.*, 2009) blastn and blastp, and the HMM search program jackhmmer (Johnson *et al.*, 2010), to identify homologs of the input genome and its predicted gene and peptide sequences using several databases: National Center for Biological Information (NCBI) virus genomes, NCBI Refseq proteins, NCBI refseq genes, NCBI virus proteins and Non-Redundant protein sequence database (NR) (NCBI Resource Coordinators, 2016), as well as Swissprot (Bairoch and Apweiler, 2000), Phage Annotation Tools and Methods (PhAnToMe) (www.phantome.org), a virus subset of Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa *et al.*, 2017) and a fasta sequence dataset derived using the database of phage Virus Orthologous Groups (pVOG) identifiers (Grazziotin *et al.*, 2017). The latter database is modified to contain the pVOG identifiers in the fasta headers, by means of scripts included in the multiPhATE distribution.

2.3 Output

PhATE generates the following files and directories: (i) output from the gene-call algorithms and the gene-call comparison (Supplementary Material 'phate_P2_CGC.pdf'); (ii) gene and translated peptide fasta files; (iii) combined-annotation summary files; (iv) directories containing raw BLAST outputs for genome and peptide blast runs; (v) directories with raw HMM search outputs for peptide searches; (vi) alignment-ready fasta files containing each predicted peptide plus the members of each identified pVOG family to which a peptide may be assigned and (vii) log files. BLAST and HMM raw data outputs can be saved or cleaned from the output directories (see README). We demonstrate application of multiPhATE to the annotation of two newly sequenced *Yersinia pestis* phage genomes (see Supplementary Material 'phate_results.pdf'.

Acknowledgements

This work was performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. See Supplementary Material 'phate_author_information.pdf'.

Funding

This work was supported by the Defense Threat Research Agency [grant number 10027-20149].

Conflicts of Interest: none declared.

References

Arndt,D. *et al.* (2016) PHASTER: a better, faster version of the PHAST phage search tool. *Nucleic Acids Res.*, **44**, W16–W21.

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Camacho,C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Carrol,D. *et al.* (2018) The global virome project. *Science*, **359**, 872–874.

Delcher,A.L. *et al.* (2007) Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics.*, **15**, 673–679.

Grazziotin,A.L. *et al.* (2017) Prokaryotic virus orthologous groups (pVOGs): a resource for comparative genomics and protein family annotation. *Nucleic Acids Res.*, **45**, D491–D498.

Hyatt,D. *et al.* (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, **11**, 119.

Johnson,L.S. *et al.* (2010) Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, **11**, 431.

Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.

Kang,H.S. *et al.* (2018) Prophage genomics reveals patterns in phage genome organization and replication. 1–28, doi: 10.1101/114819.

Kutter,E.M. *et al.* (2015) Re-establishing a place for phage therapy in western medicine. *Future Microbiol.*, **10**, 685–688.

Lomsadze,A. *et al.* (2017) Improved prokaryotic gene prediction yields insights into transcription and translation mechanisms on whole genome scale. 1–24. doi: 10.1101/193490.

McNair,K. *et al.* (2019) PHANOTATE: a novel approach to gene identification in phage genomes. *Bioinformatics*, 1–6, doi: 10.1093/bioinformatics/btz265.

NCBI Resource Coordinators. (2016) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **44**, D7–D19.

Perkel,J.M. (2017) Democratizing bioinformatics. *Nature*, **543**, 137–138.

Philipson,C.W. *et al.* (2018) Characterizing phage genomes for therapeutic applications. *Viruses*, **10**, 188.

Roux,S. *et al.* (2015) VirSorter: mining viral signal from microbial genomic data. *PeerJ.*, **3**, e985.

Russell,D.A. and Hatfull,G.F. (2017) PhagesDB: the actinobacteriophage database. *Bioinformatics*, **33**, 784–786.

Seemann,T. (2014) Rapid prokaryotic genome annotation. *Bioinformatics*, **30**, 2068–2069.