

camlhmp: a simple framework for building reproducible microbial genome-based typing tools

Robert A. Petit, III,^{1,2} Chayse M. Rowley,¹ Taylor R. Fearing,¹ Stefaan Verwimp,³ Rob G. Christensen,¹ Jim A. Mildenberger,¹ Joseph M. Reed,¹ Timothy D. Read^{2,4}

AUTHOR AFFILIATIONS See affiliation list on p. 2.

ABSTRACT Sequence-based typing is critical for microbial genomics, yet existing tools are often developed in isolation, leading to duplicated effort, inconsistent formats, and limited community participation. To address this, we developed camlhmp (<https://github.com/rpetit3/camlhmp>), a framework for creating, executing, and maintaining typing tools with simple, human-readable YAML files.

KEYWORDS bioinformatics, software, sequence-based typing, microbial genomics

In microbiology, genetic typing by PCR-based NAATs (Nucleic Acid Amplification Tests) (1) is commonly used to define genotypes and infer phenotypes. However, with the prevalence of whole-genome sequencing, new bioinformatics-based approaches are increasingly being developed (2). Developing a new sequence-based typing (SBT) tool requires extensive knowledge of the target organism and bioinformatics expertise. Furthermore, a lack of adherence to standard development practices limits community contributions.

Recognizing the need for a standardized, accessible SBT framework, we developed camlhmp (*Classification through yAML Heuristic Mapping Protocol*), a Python-based framework designed to simplify the creation and management of SBT tools. It uses YAML (3), a simple human-readable data serialization language, to define a typing schema. From the schema, camlhmp uses BLAST+ (4) to align target sequences against an input genome to assign types based on the schema-defined heuristics (e.g. exact match, percent coverage thresholds). camlhmp has been developed for x86-64 Linux and macOS systems and requires Python (v3.11+), BLAST+ (v2.15.0+), Biopython (v1.83+) (5), and supporting open-source Python libraries. To ease dependency management, camlhmp is available from Bioconda (6) and PyPi.

The camlhmp (v1.1.4) framework provided a command-line interface (CLI) and an application programming interface (API), which took a user-supplied YAML schema and FASTA sequence file as inputs.

Each YAML schema contained metadata, engine, targets, aliases, and types sections. metadata included fields to describe the schema, such as name and description. engine included a description of the tool used for alignment. targets included a list of all defined targets within the schema, each with a reference sequence. aliases allowed for a name to represent a group of targets. Finally, types defined each type based on the presence of specified targets or aliases. The camlhmp framework includes schema validation functions and example schemas are provided through stand-alone tools (Table 1) and the test suite.

camlhmp included three CLI tools: camlhmp-blast-alleles, camlhmp-blast-regions, and camlhmp-blast-targets. All commands required the camlhmp schema, target FASTA file, and assembly FASTA file. camlhmp-blast-alleles aligned to target loci with alleles assigned by perfect matches. camlhmp-blast-regions aligned to large genomic regions

Editor Patrick D. Schloss, University of Michigan, Ann Arbor, Michigan, USA

Address correspondence to Robert A. Petit III, robert.petit@wyo.gov.

The authors declare no conflict of interest

Ed. Note: This article was handled by Pat Schloss, who acted as a Guest Editor in consultation with the Editor in Chief, Irene Newton. Dr. Schloss's role as the Guest Editor of this article is outside of his duties as the chair of the Journals Committee.

Received 17 February 2026

Accepted 25 March 2026

Published 20 April 2026

Copyright © 2026 Petit III et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/).

TABLE 1 Tools utilizing the camlhmp framework

Tool	Description	Typing approach
pasty	Serogrouping <i>P. aeruginosa</i> with O-specific antigen clusters (7)	camlhmp-blast-regions CLI
pbptyper	Typing penicillin-binding protein (PBP) in <i>S. pneumoniae</i> (8)	camlhmp API customization
sccmec	Typing SCCmec cassette in <i>S. aureus</i> samples (9, 10)	camlhmp-blast-targets CLI

with types assigned based on percent coverage and fewest hits. camlhmp-blast-targets aligned to individual targets with types assigned based on target matches.

The camlhmp API was grouped into the following types: engine, framework, parser, and utility. engine included modules for executing tools. framework included modules for working with the camlhmp schema. parser included modules for parsing outputs. Finally, the utility included generic modules for reading, writing, and validating data.

To demonstrate the application of camlhmp, we developed schemas for the bacterial pathogens *Pseudomonas aeruginosa*, *Streptococcus pneumoniae*, and *Staphylococcus aureus* (Table 1). These schemas can be used as stand-alone tools or as part of the Bactopia pipeline (v3.2.0) (11).

We developed camlhmp to standardize and simplify the creation and maintenance of SBT tools in microbiology. We demonstrated camlhmp with three tools: pasty, pbptyper, and sccmec. As genomic surveillance expands, tools like camlhmp are vital for sustainable typing of diverse microbial pathogens.

ACKNOWLEDGMENTS

R.A.P.III and T.D.R. received support for this work from the Office of Advanced Molecular Detection, Centers for Disease Control and Prevention (cooperative agreement number CK22-2204) through contract 40500-050-23234506 from the Georgia Department of Public Health.

The GitHub Copilot AI tool was used (i.e., code auto-completion) during development. All code was written and reviewed by the authors.

AUTHOR AFFILIATIONS

¹Wyoming Public Health Laboratory, Wyoming Department of Health, Cheyenne, Wyoming, USA

²Center of Applied Pathogen Epidemiology and Outbreak Response, Georgia PGCoe, Atlanta, Georgia, USA

³Computational Systems Biology, Department of Microbial and Molecular Systems, KU Leuven, Leuven, Flanders, Belgium

⁴Division of Infectious Diseases, Department of Medicine, Emory University School of Medicine, Atlanta, Georgia, USA

AUTHOR ORCIDs

Robert A. Petit, III  <http://orcid.org/0000-0002-1350-9426>

Timothy D. Read  <http://orcid.org/0000-0001-8966-9680>

AUTHOR CONTRIBUTIONS

Robert A. Petit, III, Conceptualization, Data curation, Formal analysis, Project administration, Resources, Software, Validation, Visualization, Writing – original draft | Chayse M. Rowley, Software, Validation, Writing – review and editing | Taylor R. Fearing, Software, Validation, Writing – review and editing | Stefaan Verwimp, Software, Validation, Writing – review and editing | Rob G. Christensen, Software, Writing – review and editing | Jim A. Mildenberger, Software, Writing – review and editing | Joseph M. Reed, Resources, Software, Writing – review and editing | Timothy D. Read, Conceptualization,

Funding acquisition, Methodology, Resources, Software, Validation, Writing – original draft, Writing – review and editing

DATA AVAILABILITY

camlhmp is available at GitHub, PyPi, and Bioconda:

- <https://github.com/rpetit3/camlhmp>
- <https://pypi.org/project/camlhmp/>
- <https://bioconda.github.io/recipes/camlhmp/README.html>

camlhmp is archived on Zenodo at <https://doi.org/10.5281/zenodo.19040444>.

Documentation for camlhmp is available at <https://rpetit3.github.io/camlhmp/>.

camlhmp-powered Typing Tools are available from GitHub, Bioconda, and Bactopia:

- pasty - <https://github.com/rpetit3/pasty>
- pbptyper - <https://github.com/rpetit3/pbptyper>
- sccmec - <https://github.com/rpetit3/sccmec>

REFERENCES

1. Vaneechoutte M, Van Eldere J. 1997. The possibilities and limitations of nucleic acid amplification technology in diagnostic microbiology. *J Med Microbiol* 46:188–194. <https://doi.org/10.1099/00222615-46-3-188>
2. Simar SR, Hanson BM, Arias CA. 2021. Techniques in bacterial strain typing: past, present, and future: past, present, and future. *Curr Opin Infect Dis* 34:339–345. <https://doi.org/10.1097/QCO.0000000000000743>
3. YAML ain't markup language (YAMLTm) revision 1.2.2. Accessed July 8, 2025. <https://yaml.org/spec/1.2.2>.
4. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. 2009. BLAST+: architecture and applications. *BMC Bioinformatics* 10:421. <https://doi.org/10.1186/1471-2105-10-421>
5. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, de Hoon MJL. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25:1422–1423. <https://doi.org/10.1093/bioinformatics/btp163>
6. Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Köster J, Bioconda Team. 2018. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat Methods* 15:475–476. <https://doi.org/10.1038/s41592-018-0046-7>
7. Thrane SW, Taylor VL, Lund O, Lam JS, Jelsbak L. 2016. Application of whole-genome sequencing data for O-specific antigen analysis and in silico serotyping of *Pseudomonas aeruginosa* isolates. *J Clin Microbiol* 54:1782–1788. <https://doi.org/10.1128/JCM.00349-16>
8. Chambers HF. 1999. Penicillin-binding protein-mediated resistance in pneumococci and staphylococci. *J Infect Dis* 179 Suppl 2:S353–9. <https://doi.org/10.1086/513854>
9. Uehara Y. 2022. Current status of staphylococcal cassette chromosome mec (SCCmec). *Antibiotics (Basel)* 11:86. <https://doi.org/10.3390/antibiotics11010086>
10. Wolska-Gębarzewska M, Międzobrodzki J, Kosecka-Strojek M. 2024. Current types of staphylococcal cassette chromosome *mec* (SCCmec) in clinically relevant coagulase-negative staphylococcal (CoNS) species. *Crit Rev Microbiol*:1–17. <https://doi.org/10.1080/1040841X.2023.2274841>
11. Petit RAIII, Read TD. 2020. Bactopia: a flexible pipeline for complete analysis of bacterial genomes. *mSystems* 5:e00190–20. <https://doi.org/10.1128/mSystems.00190-20>