







1 camlhmp: A simple framework for building 2 reproducible microbial genome-based typing tools

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

6 **1** Wyoming Public Health Laboratory, Wyoming Department of Health, Cheyenne, Wyoming, USA **2**
7 Center of Applied Pathogen Epidemiology and Outbreak Response, Georgia PGCoe, Atlanta, Georgia,
8 USA **3** Division of Infectious Diseases, Department of Medicine, Emory University School of Medicine,
9 Atlanta, Georgia, USA **4** Computational Systems Biology, Department of Microbial and Molecular
10 Systems, KU Leuven, Belgium

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Submitted: 16 September 2025

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))

11 Summary

12 Sequence-based typing (SBT) is critical for microbial genomics, yet existing tools are often
13 developed in isolation, leading to duplicated efforts, inconsistent formats, and limited commu-
14 nity participation. Here we present camlhmp (Classification through yAML Heuristic Mapping
15 Protocol; pronounced “camel hump”), a flexible framework for creating, executing, and main-
16 taining SBT tools. To demonstrate its application, we developed three camlhmp-powered tools
17 for typing *Pseudomonas aeruginosa*, *Staphylococcus aureus*, and *Streptococcus pneumoniae*.
18 camlhmp is available from PyPi, Bioconda, and at <https://github.com/rpetit3/camlhmp>

Statement of Need

19 In microbiology, genetic typing is commonly used to define genotypes and infer phenotypes.
20 Many assays have been developed for the laboratory using PCR-based NAATs (Nucleic Acid
21 Amplification Tests) ([Vaneeschoutte & Van Eldere, 1997](#)). However, with the prevalence
22 of whole-genome sequencing, new bioinformatics-based approaches are increasingly being
23 developed ([Simar et al., 2021](#)). Developing a new SBT tool requires both extensive knowledge
24 of the target organism and expertise in bioinformatics. These tools are often developed without
25 adhering to standard practices, hindering community contributions.

26 Recognizing the need for a standardized framework to develop easy to use and accessible SBT
27 tools, we developed camlhmp. camlhmp is a Python-based framework designed to simplify the
28 development and management of SBT tools. It uses YAML ([YAML Ain't Markup Language](#)
29 ([YAML™](#)) [Revision 1.2.2, n.d.](#)), a simple human-readable data serialization language, to define
30 a typing schema. From the schema, camlhmp will produce genetic typing results for an input
31 genome in a consistent tab-delimited format.
32

33 camlhmp Design

34 The camlhmp (v1.1.0) framework consisted of a user-supplied YAML schema and FASTA format
35 sequence file, a command-line interface (CLI), and an application programming interface (API).

36 User-supplied files: YAML and FASTA

37 The YAML format was selected primarily for its human readability compared to formats such
38 as JSON (*JavaScript Object Notation*, n.d.) or TOML (*Tom's Obvious, Minimal Language*,
39 n.d.). Each YAML schema was composed of specific sections, including metadata, engine,
40 targets, aliases, and types. metadata included fields to describe the schema, such as name
41 and description. engine included a description of the software tool used to compare query
42 sequences to the input genome. targets included a list of all the targets within the schema,
43 each represented in the corresponding reference FASTA file. aliases allowed for a name
44 to represent a group of targets. Finally, types defined each type based on the presence of
45 specified targets or aliases.

46 Command-line interface

47 camlhcp (v1.1.0) included three CLI tools: camlhcp-blast-alleles, camlhcp-blast-regions,
48 and camlhcp-blast-targets.

49 Each command expected an camlhcp schema and the corresponding target FASTA file, as
50 well as the FASTA-formatted assembly, to be typed. Each utilized BLAST+ (*Camacho et al.*,
51 2009) to align the target sequences to the sample while differing in the type of test performed.
52 camlhcp-blast-alleles expected target alleles for loci to be aligned to, with alleles assigned
53 by perfect matches. camlhcp-blast-regions expected large genomic regions to be aligned
54 in single or multiple hits, with types assigned based on percent coverage and fewest hits.
55 camlhcp-blast-targets expected individual target sequences to be aligned with the types
56 assigned based on the group of targets with a match.

57 Application programming interface

58 The camlhcp (v1.1.0) API was grouped into the following types: engine, framework, parser,
59 and utility. engine included modules for executing bioinformatic tools. framework included
60 modules for working with the camlhcp schema files. parser included modules for parsing the
61 outputs from the engine modules. Finally, utility included generic modules for reading,
62 writing, and validating data.

63 Application of camlhcp

64 To demonstrate the application of camlhcp, we developed schemas for the bacterial pathogens
65 *Pseudomonas aeruginosa*, *Streptococcus pneumoniae*, and *Staphylococcus aureus*.

66 pasty, for serogrouping *P. aeruginosa* samples, used camlhcp-blast-regions to align user
67 assemblies to representative O-specific antigen (OSA) clusters (*Thrane et al.*, 2016) to user
68 assemblies.

69 pbptyper, for typing the penicillin-binding protein (PBP) in *S. pneumoniae* (*Chambers*, 1999),
70 used the camlhcp API to align representative PBPs alleles (*Li et al.*, 2016) against user
71 assemblies.

72 sccmec, for typing SCCmec cassette in *S. aureus* samples (*Uehara*, 2022), used camlhcp-
73 blast-targets to align user assemblies against known target SCCmec (*Wolska-Gębarzewska*
74 et al., 2023) associated genes.

75 Each can be used as stand-alone tools or from the Bactopia pipeline (v3.2.0) (*Petit & Read*,
76 2020).

Conclusion

We developed camlhcp to address the challenge in microbiology of creating and maintaining SBT tools that are both standardized and accessible. We used YAML for defining, executing, and maintaining these tools, lowering the barrier for researchers to develop organism-specific typing schemas. To demonstrate camlhcp's flexibility, we developed three camlhcp-powered typing tools: pasty, pbptyper, and sccmec. As genomic surveillance continues to grow, tools like camlhcp will play an essential role in supporting sustainable typing efforts across diverse microbial pathogens.

Code Availability

camlhcp is available at GitHub, PyPi, and Bioconda:

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

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

camlhcp-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>

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgements

RP3 and TDR 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.

References

- Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics*, 10, 421.
- Chambers, H. F. (1999). Penicillin-binding protein-mediated resistance in pneumococci and staphylococci. *J. Infect. Dis.*, 179 Suppl 2(s2), S353–9.
- JavaScript object notation*. (n.d.). <https://www.json.org/json-en.html>.
- Li, Y., Metcalf, B. J., Chochua, S., Li, Z., Gertz, R. E., Jr, Walker, H., Hawkins, P. A., Tran, T., Whitney, C. G., McGee, L., & Beall, B. W. (2016). Penicillin-binding protein transpeptidase signatures for tracking and predicting β -lactam resistance levels in streptococcus pneumoniae. *MBio*, 7(3).
- Petit, R. A., 3rd, & Read, T. D. (2020). Bactopia: A flexible pipeline for complete analysis of bacterial genomes. *mSystems*, 5(4).
- Simar, S. R., Hanson, B. M., & Arias, C. A. (2021). Techniques in bacterial strain typing: Past, present, and future: Past, present, and future. *Curr. Opin. Infect. Dis.*, 34(4),

- 115 339–345.
- 116 Thrane, S. W., Taylor, V. L., Lund, O., Lam, J. S., & Jelsbak, L. (2016). Application of
117 whole-genome sequencing data for O-specific antigen analysis and in silico serotyping of
118 *Pseudomonas aeruginosa* isolates. *J. Clin. Microbiol.*, 54(7), 1782–1788.
- 119 *Tom's obvious, minimal language*. (n.d.). <https://toml.io/en/v1.0.0>.
- 120 Uehara, Y. (2022). Current status of staphylococcal cassette chromosome mec (SCCmec).
121 *Antibiotics*, 11(1), 86.
- 122 Vaneechoutte, M., & Van Eldere, J. (1997). The possibilities and limitations of nucleic acid
123 amplification technology in diagnostic microbiology. *J. Med. Microbiol.*, 46(3), 188–194.
- 124 Wolska-Gębarzewska, M., Międzobrodzki, J., & Kosecka-Strojek, M. (2023). Current types of
125 staphylococcal cassette chromosome mec (SCCmec) in clinically relevant coagulase-negative
126 staphylococcal (CoNS) species. *Crit. Rev. Microbiol.*, 1–17.
- 127 *YAML ain't markup language (YAML™) revision 1.2.2*. (n.d.). <https://yaml.org/spec/1.2.2/>.

DRAFT