FastMLST: A multi-core tool for multilocus sequence typing of draft genome assemblies

Enzo Guerrero-Araya[1,2,*], Marina Muñoz[1,2,3], César Rodríguez[3] and Daniel Paredes-Sabja[1,2,5,*]

[1] Microbiota-Host Interactions and Clostridia Research Group, Facultad de Ciencias de la Vida, Universidad Andrés Bello, Santiago, Chile. [2] ANID – Millennium Science Initiative Program - Millennium Nucleus in the Biology of the Intestinal Microbiota, Santiago, Chile. [3] Grupo de Investigaciones Microbiológicas-UR (GIMUR), Departamento de Biología, Facultad de Ciencias Naturales, Universidad del Rosario, Bogotá, Colombia. [4] Facultad de Microbiología & Centro de Investigación en Enfermedades Tropicales (CIET), Universidad de Costa Rica, San José, Costa Rica. [5] Department of Biology, Texas A&M University, College Station, TX, 77843, USA.

* To whom correspondence should be sent

Contact: Enzo Guerrero-Araya: enzoguerreroaraya@gmail.com and Daniel Paredes-Sabja: dparedes-sabja@bio.tamu.edu

ABSTRACT

Multilocus Sequence Typing (MLST) is a precise microbial typing approach at the intra-species level for epidemiological and evolutionary purposes. It operates by assigning a sequence type (ST) identifier to each specimen, based on a combination of allelic sequences obtained for multiple housekeeping genes included in a defined scheme. The use of MLST has multiplied due to the availability of large numbers of genomic sequences and epidemiological data in public repositories. However, data processing speed has become problematic due to datasets' massive size. Here, we present FastMLST, a tool that is designed to perform PubMLST searches using BLASTn and a divide-and-conquer approach. Compared with mlst, CGE/MLST, MLSTar, and PubMLST, FastMLST takes advantage of current multi-core computers to simultaneously type thousands of genome assemblies in minutes, reducing processing times by at least 4-fold and with more than 99.95% consistency.

Availability and Implementation: The source code, installation instructions and documentation are available at https://github.com/EnzoAndree/FastMLST

Abstract word count: 151

Word count (main text): 992/1000

1. Introduction

Multilocus Sequence Typing was a revolutionary attempt to standardize the molecular typing of *Neisseria meningitidis* (Maiden, et al., 1998) that has been replicated in multiple pathogens of global relevance (Dingle, et al., 2001; Griffiths, et al., 2010; Martin-Rodriguez, et al., 2019; Meats, et al., 2003). This technique assigns a bacterial isolate a sequence type (ST) based on a

combination of alleles from an optimal set of housekeeping genes defined for each species. Even though there are methods with higher discrimination power (cgMLST/wgMLST), MLST offers several advantages, such as (1) a common language for typing, (2) simple implementation, and (3) a vast amount of information collected throughout history (Kimura, 2018).

Currently available programs for ST determination from assembled genomes i.e. mlst (Seemann, 2015), CGE/MLST (Larsen, et al., 2012), MLSTar (Ferrés and Iraola, 2018), and the online tool PubMLST (Jolley, et al., 2018) deliver results in 1-9 minutes when dealing with hundreds of genomes. However, since most of them do not (or inefficiently) implement parallel computing, processing of tens of thousands of genomes can be extremely time-consuming. In addition, their outputs are typically restricted to a table with allelic profiles and do not deliver a file with concatenated allele sequences necessary for subsequent phylogenetic analyses (Glaeser and Kampfer, 2015).

In response to these limitations, we developed FastMLST, a tool focused on parallel computing via a divide-and-conquer approach (Smith, 1985) that performs BLASTn searches (Camacho, et al., 2009) against the sequences of MLST schemes deposited in the PubMLST database (Jolley, et al., 2018). FastMLST delivers two main outputs: a table the allelic profiles detected for a genome query and a multi-FASTA file containing their sequences.

2. Methods

2.1. Implementation

FastMLST is an open source standalone software implemented in Python 3 that only requires BLASTn as an external program. FastMLST is easy to install via the Anaconda Package Manager (https://anaconda.org/bioconda/fastmlst) and is in constant development.

2.2. Key analysis steps

PubMLST database setup

FastMLST can retrieve and update the PubMLST database (Jolley, et al., 2018) with its '--update-mlst' option. It automatically processes the 150 currently available schemes (October 2020) and prepares the system for subsequent deployment.

Allele searches

FastMLST is based on a divide-and-conquer approach, whereby each genome is processed in parallel. Allele searches are performed with BLASTn using the public, regularly updated, and centralized database maintained at PubMLST (Jolley, et al., 2018).

Scheme identification

The MLST scheme to use for each query genome is inferred by an automatic scoring system, similar to the one reported by (Seemann, 2015). This system assigns known, full-length alleles a maximum score of 100 points, 70% of the score to alleles of the expected length but with SNPs (95% identity by default), or 20% of the score to alleles with unexpected lengths. At the end, the MLST scheme with the highest overall score is selected for further analyses. Alternatively, it is possible to choose among schemes manually. When multiple perfect hits to a single scheme are found, FastMLST reports all hits to alert the user of potential contaminations.

3. Performance benchmark

We assessed the capability of FastMLST to identify STs correctly for three bacterial species with unequal numbers of scheme alleles. To this end, we measured the concordance over ST assignment between FastMLST and PubMLST (Jolley, et al., 2018). The species tested were

79 *Cutibacterium acnes, Enterococcus faecium*, and *Streptococcus pneumoniae*, which as of
80 September 11th 2020 included 153, 1,833 and 16,301 STs, respectively. Besides, we compared the
81 processing speeds of FastMLST and four others widely used software: mlst v2.11, CGE/MLST
82 v2.0.4, MLSTar v1.0, and the online tool PubMLST.

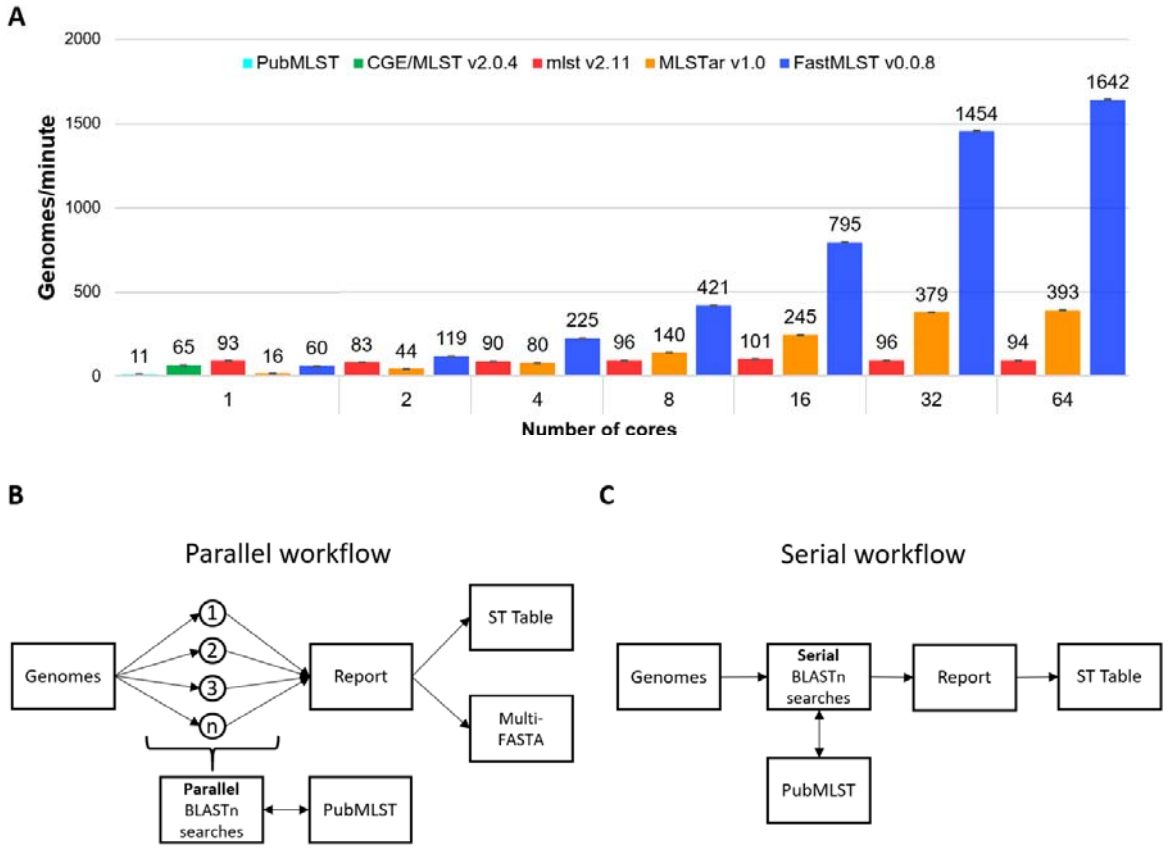
83 A total of 278 genomes for *C. acnes*, 1,915 for *E. faecium*, and 7,996 for *S. pneumoniae* obtained
84 from GenBank using ncbi-genome-download v0.2.11 (https://github.com/kblin/ncbi-genome-
85 download) were used in the analysis of ST assignment concordance (Supplementary Table 1).
86 Compared with PubMLST result (Supplementary Tables 2, 3 and 4), we show that FastMLST
87 (Supplementary Tables 5, 6 and 7) reaches a concordance of 99.96% (n=9,502/9,506) (Table 1). As
88 to the inconsistencies detected, FastMLST failed to assign STs to 4 genomes with multiple perfect
89 hits against the same locus, indicating that they are possibly contaminated (Supplementary Table 6
90 and 7). Additionally, FastMLST reported 401 STs that remained undetected by PubMLST (7 in *C.*
91 *acnes*, 216 in *E. faecium*, and 178 in *S. pneumoniae*) (Supplementary Table 5, 6 and 7). These new
92 ST are derived from new alleles or new combinations of known alleles (Table 1). On the other
93 hand, FastMLST missed some full-length alleles due to incomplete coverage (<100%) (Table 1).

94 **Table 1. Concordance analysis between FastMLST and PubMLST using 10,189 genomes.**

Dataset	Genomes	%GC	ST assigned by PubMLST[a]	ST assigned by FastMLST[b]	Concordance[c]	New ST with full-length alleles[d]	Missing alleles (<100% coverage)[e]
C. acnes	278	60%	259	259	259/259 (100%)	7	12
E. faecium	1915	38%	1545	1544	1544/1545 (99.94%)	216	155
S. pneumoniae	7996	39%	7702	7699	7699/7702 (99.96%)	178	119

[a] A genome was included in this category if an ST was assigned using PubMLST.
[b] A genome was included in this category if an ST was assigned using FastMLST.
[c] A ST assignment was considered concordant if the ST assigned by PubMLST and FastMLST was identical.
[d] A genome with novel alleles (100% coverage). It is a designation only present in FastMLST.
[e] Details of genomes with missing alleles detected using FastMLST.

95 The speed of FastMLST, MLSTar, PubMLST and mlst (expressed in genomes/minute) was
96 compared in a workstation equipped with 2 Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10GHz using 1, 2,
97 4, 8, 16, 32 or 64 cores. Each program was run in triplicate with a sample of 278 *C. acnes* genomes
98 each time. The results for PubMLST were obtained remotely because it does not offer a
99 standalone version. When a single core was used, mlst v2.11 achieved the best performance, with
100 93 genomes/minute (Figure 1A). However, mlst v2.11 did not speed up when its multiple
101 processing option was enabled (Figure 1A). MLSTar can also use multiple processors, but it is at
102 least 4-fold slower than FastMLST (Figure 1A and 1B). FastMLST´s processing speed grew
103 exponentially as more cores were deployed (Figure 1B), reaching a speed of 1,642
104 genomes/minute using 64 cores. This allows processing of 16,000 genomes in less than 10
105 minutes.



106

Figure 1. Processing speed and simplified comparative-workflow. (A) Processing speed (genomes/minute) of FastMLST and standard software for MLST determination (mlst, PubMLST, CGE/MLST and MLSTar) was compared by triplicate using a dataset composed of 278 *C. acnes* genomes and 1, 2, 4, 8, 16, 32, and 64 cores. Error bars represent the standard error of three replicates. Only mlst, MLSTar and FastMLST natively support parallel processing. **(B)** FastMLST and MLSTar workflow. **(C)** mlst, PubMLST and CGE/MLST workflow.

4. Final remarks

FastMLST assigns STs to thousands of genomes in minutes with the additional plus of generating a multi-FASTA file with concatenated allele sequences suitable for downstream phylogenetic analysis. These unique features have the potential to boost future research.

Funding

This work was supported by a Doctoral fellowship 21181536 from ANID to E.G-A., and by EU-Lac project "Genomic Epidemiology of *Clostridium difficile* in Latin America (T020076)", ANID – Millennium Science Initiative Program – NCN17_093, and Start-up funds of Texas A&M University to D.P-S.

References

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

Dingle, K.E., *et al.* Multilocus sequence typing system for *Campylobacter jejuni*. *J Clin Microbiol* 2001;39(1):14-23.

Ferrés, I. and Iraola, G. MLSTar: automatic multilocus sequence typing of bacterial genomes in R. *PeerJ* 2018;6:e5098-e5098.

Glaeser, S.P. and Kampfer, P. Multilocus sequence analysis (MLSA) in prokaryotic taxonomy. *Syst Appl Microbiol* 2015;38(4):237-245.

Griffiths, D., *et al.* Multilocus sequence typing of *Clostridium difficile*. *J Clin Microbiol* 2010;48(3):770-778.

Jolley, K.A., Bray, J.E. and Maiden, M.C.J. Open-access bacterial population genomics: BIGSdb software, the PubMLST.org website and their applications. *Wellcome Open Res* 2018;3:124.

Kimura, B. Will the emergence of core genome MLST end the role of in silico MLST? *Food Microbiol* 2018;75:28-36.

Larsen, M.V., *et al.* Multilocus sequence typing of total-genome-sequenced bacteria. *J Clin Microbiol* 2012;50(4):1355-1361.

Maiden, M.C., *et al.* Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. *Proc Natl Acad Sci U S A* 1998;95(6):3140-3145.

Martin-Rodriguez, A.J., *et al.* Multilocus sequence typing of *Shewanella algae* isolates identifies disease-causing *Shewanella chilikensis* strain 6I4. *FEMS Microbiol Ecol* 2019;95(1).

Meats, E., *et al.* Characterization of encapsulated and noncapsulated *Haemophilus influenzae* and determination of phylogenetic relationships by multilocus sequence typing. *J Clin Microbiol* 2003;41(4):1623-1636.

Seemann, T. 2015. mlst. https://github.com/tseemann/mlst

Smith, D.R. The design of divide and conquer algorithms. *Science of Computer Programming* 1985;5:37-58.