CHAPTER

# 18

# Bioinformatics

*Hector F. Espitia-Navarro*[1,2], *Lavanya Rishishwar*[1,2,3], *Leonard W. Mayer*[2,3], *I. King Jordan*[1,2,3]

[1]School of Biological Sciences, Georgia Institute of Technology, Atlanta, GA, United States; [2]PanAmerican Bioinformatics Institute, Cali, Valle del Cauca, Colombia; [3]Applied Bioinformatics Laboratory, Atlanta, GA, United States

## Molecular epidemiology and typing

Epidemiology entails the study of population distributions of determinants of health and disease, and molecular approaches to epidemiology rely on the analysis of genetically encoded biomarkers and risk factors (Wang et al., 2015). Molecular epidemiology studies are critically important for public health surveillance as well as disease management and control. In the postgenomic era, which is characterized by the rapid accumulation of numerous whole-genome sequences, molecular epidemiology increasingly relies on genome-enabled techniques. Genomic approaches to molecular epidemiology necessitate the use of sophisticated computer algorithms capable of analyzing massive amounts of data for the presence and distribution of genetic markers and risk factors. In this chapter, we cover the state-of-the-art with respect to the computational genomic approaches used to support molecular epidemiology and typing.

Molecular typing refers to the identification of the specific "types" of microbial pathogens that cause infectious disease. For the most part, this concerns the set of procedures used to identify distinct strains of bacteria within a given species. Accordingly, molecular typing techniques require a high level of resolution to distinguish very closely related organisms, which is critically important for molecular epidemiology (Wang et al., 2015). The accurate identification and discrimination of bacterial strains within a given pathogenic species allows scientists to (i) address the underlying biology of bacterial pathogenicity, including virulence, transmissibility, and response to drugs and vaccines, (ii) track the spread of bacterial pathogens locally and globally, (iii) identify natural hosts for bacterial

*Microbial Forensics, Third Edition*
https://doi.org/10.1016/B978-0-12-815379-6.00018-0

pathogens and associate them with specific outbreaks, and (iv) infer the evolution and population structure of bacterial pathogens. The fundamental knowledge gained from the molecular typing of bacterial pathogens facilitates the design of public health strategies for the control and prevention of infectious disease, including tailored treatment schemes, vaccine development, and vaccine surveillance programs.

Early approaches to molecular typing employed a wide variety of surrogate techniques that allowed for the indirect study of genetic variation among bacterial pathogens. These surrogate techniques measured the properties of bacterial proteins or cell surface antigens, via Western or immunoblotting and serotyping, for example, or nucleic acids assayed via nonsequencing-based techniques, such as restriction fragment length polymorphisms or polymerase chain reaction (PCR). While the development and application of these early molecular techniques provided an important advance in bacterial typing, they were difficult to standardize, replicate, and scale-up. Perhaps, most importantly, surrogate techniques for molecular typing did not yield the depth of resolution needed to unambiguously distinguish closely related strains within multiple species of bacterial pathogens. The introduction of genetic sequence–based techniques for molecular typing provided a quantum leap in terms of resolution, stability, and reproducibility for the typing of bacterial pathogens.

## Multilocus sequence typing

The first bona fide gene sequence–based technique developed for bacterial typing is referred to as multilocus sequence typing (MLST). MLST was developed by the group of Martin Maiden at Oxford University for the analysis of *Neisseria meningitidis* and was intended to be a so-called "portable" typing scheme with results that could be directly compared among different laboratories around the world (Maiden et al., 1998). It should be noted that the sequencing and analysis of 16S ribosomal RNA genes (or 16S rRNA) has also been widely used for the characterization of the evolutionary relationships among bacterial species and predates MLST by more than 20 years. However, 16S rRNA sequencing typically does not provide sufficient resolution for the discrimination of distinct strains within bacterial species. Indeed, Maiden and colleagues have provided an overview of the resolution of a variety of sequence-based typing schemes and show that 16S rRNA sequence analysis provides the most reliable resolution at the level of bacterial genus and above (Maiden et al., 2013).

MLST employs typing schemes that are specifically tailored for individual bacterial species. Species-specific MLST typing schemes rely on sequencing fragments of a set of housekeeping genes, typically seven to nine loci, which are distributed around the genome. Essential housekeeping genes are chosen for MLST to ensure that the loci are universally present among isolates that are to be typed. Distinct gene sequences for each locus in an MLST scheme are referred to as alleles, and differences between alleles across all loci in the scheme are used to distinguish specific types (or strains) of bacteria within a species. Each distinct sequence (allele) of a given MLST locus is identified by a gene (locus) name and an integer number that uniquely identifies the allele. Locus-specific integer numbers denote the order of discovery for the alleles at that locus. For example, the ABC transporter ATP-binding gene *abcZ* is one of seven loci used as part of the traditional *N. meningitidis* MLST scheme; unique alleles of *abcZ* are denoted as *abcZ_1*, *abcZ_2*, etc., and as of this writing 881 distinct *abcZ* alleles have been identified in *N. meningitidis*. The combination of alleles characterized across all loci of the MLST scheme defines an allelic profile which is labeled with an arbitrary number that identifies a sequence type (ST). For example, for

*N. meningitidis*, the combination of the alleles *abcZ_1*, *adk_3*, *aroE_4*, *fumC_7*, *gdh_1*, *pdhC_1*, and *pgm_3* results in the allelic profile 1−3−4−7−1−1−3 that represents sequence type 2 (ST2) (Fig. 18.1). Each species-specific MLST scheme uses a database that contains all the known alleles for each locus in the scheme and a table that associates each observed allelic profile with an ST. To characterize an isolate, the seven loci of the scheme of the species under study are sequenced, and each locus-specific sequence is compared to the allele database of the scheme, using a sequence similarity search program such as BLAST+ (Camacho et al., 2009), to generate the allelic profile of the isolate. Finally, the unique ST identifier for the isolate is retrieved from the table of allelic profiles. STs for multiple isolates can be compared, using a minimum spanning tree for example (Fig. 18.1), to get a sense of the scope of diversity found in a given study.

MLST was introduced in 1998, about 6 years before start of the next-generation sequencing (NGS) revolution. At that time, sequencing was done using the Sanger method, which despite numerous technological improvements over the years was still relatively low-throughput, labor-intensive, time-consuming, and expensive. Given the technological limitations at the time, MLST was designed in such a way as to capture genome-wide patterns of sequence variation via sequencing a very small portion of the entire genome. For instance, MLST alleles in the original *N. meningitidis* typing scheme are approximately 450 bp long per locus. The total length of the seven allele sequences in this scheme is 3,284 bp, which represents a mere ~0.1% of an entire 2.3 Mbp *N. meningitidis* genome sequence. It is quite remarkable to consider how successful MLST has been for (fairly) high resolution bacterial typing given the diminishingly small percentage of overall genome sequence diversity that is represented in each scheme.

One way that MLST was scaled-up was through the use of 96-well plates to perform multiple simultaneous PCRs for specific amplicons

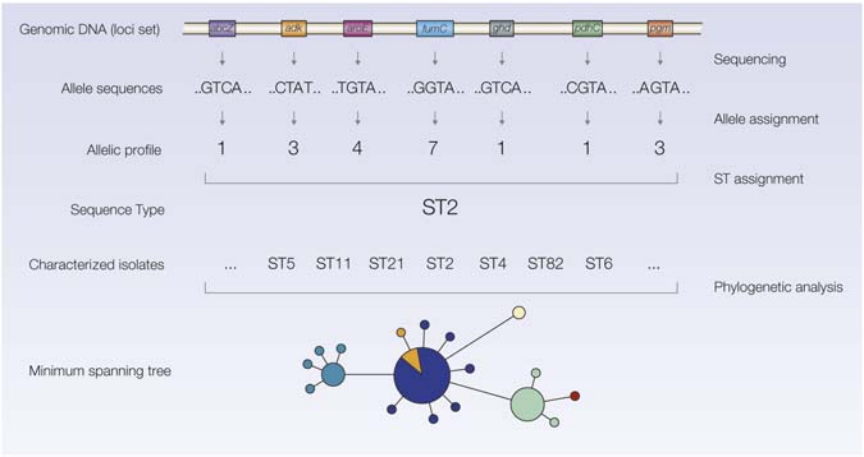


FIGURE 18.1 **Graphic representation of the multilocus sequence typing (MLST) method.** An example is shown for the traditional MLST scheme used for *Neisseria meningitidis*. Seven different loci, distributed around the genome (not shown to scale), are used for this scheme. Unique allele sequences for each locus are characterized and compared against a species-specific MLST database to yield an allelic profile, and each allelic profile is then associated with a specific sequence type (ST). Multiple STs from one or more studies can be compared using phylogenetic analyses to characterize the extent of diversity and relationships seen among a set of bacterial isolates.

across different bacterial isolates. PCR products were then characterized using Sanger sequencing reactions and analyzed on a parallel capillary electrophoresis instrument. MLST software packages, first STARS and later MGIP, were then used to automatically convert Sanger sequencing chromatograms to allele calls and sequence types (Katz et al., 2009). Further extensions of MLST were developed by including additional loci, particularly more variable antigen encoding loci, to yield so-called MLST+ or extended MLST (eMLST) schemes. Extended schemes for *N. meningitidis* typically include combinations of an additional six loci, including the *porA*, *porB*, *fHbp*, and *fetA* antigen encoding genes. The inclusion of antigen encoding genes not only provides additional resolution to traditional MLST schemes but can also yield valuable information with respect to vaccine design and measurement of response.

## Impact of NGS on bacterial typing schemes

The advent of NGS techniques, and the resulting explosion of bacterial genome sequences (Fig. 18.2), has led to the development of new genome-enabled approaches for bacterial typing. First and foremost, it quickly became faster and more cost-effective to sequence an entire genome of a bacterial isolate using NGS platforms (initially Roche 454 and now primarily Illumina) than to amplify multiple specific MLST loci and perform Sanger sequencing on individual amplicons. Whole-genome sequencing obviously yields a massive amount of data far in excess of what is provided by traditional seven to nine loci MLST schemes. This explosion of sequence data presented two distinct opportunities for bacterial typing, each of which came with its own set of computational challenges: (1) the use of whole-genome sequence data for

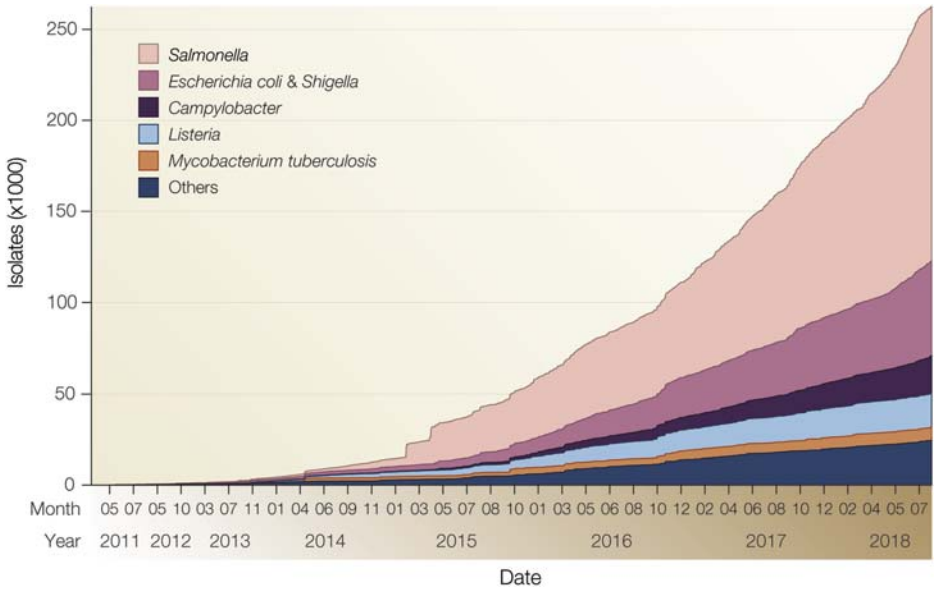


FIGURE 18.2 **Growth in whole-genome sequencing (WGS) of bacterial pathogens in the last 7 years.** The graph represents the number of WGS data submitted to NCBI's Pathogen Detection database since 2011.

existing MLST schemes and (2) the development of novel, larger-scale typing schemes, which avail themselves of the substantial data generated by NGS. We will cover these two broad technological developments in turn, with an emphasis on the computational approaches used for each.

Given the ability to readily generate whole-genome sequences via NGS, one may wonder why a small-scale approach like MLST would be needed at all. It may seem more desirable to simply discard the MLST approach and move on to techniques that better leverage genome-scale datasets. The answer to this question has to do with the vast amount of critically important legacy data that have been generated by the application of MLST schemes to scores of bacterial pathogens over the years. The most widely used MLST scheme database—PubMLST https://pubmlst.org/databases/—currently hosts MLST schemes for 99 species (or genera) of bacterial pathogens along with 10 eukaryotic (fungal) pathogens, bacteriophages, and plasmids. These schemes cover many tens of thousands of distinct allelic sequences and have been widely applied in hundreds of molecular epidemiology studies around the world, including routine surveillance and outbreak investigations. Together, these data and results represent a wealth of information relating bacterial genome sequence variation to determinants of infectious disease. As such, it will remain critically important to continue characterizing bacterial isolates with respect to their MLST sequence types. Of course, with whole-genome sequences in hand, it will also be possible to apply one or more of the new larger-scale typing schemes to the same datasets used to generate MLST sequence types. These two approaches are by no means mutually exclusive.

The remaining importance of MLST in the post-genomic era, combined with the fact that it is now faster and cheaper to sequence whole genomes using NGS platforms than to Sanger sequence MLST amplicons, necessitates the development and application of computational techniques for MLST analysis using NGS datasets. Indeed, there has been a substantial developmental effort for genome-enabled MLST software over the last 8 years. As of this writing, there are at least 13 different genome-based computational methods for MLST analysis (Table 18.1). Our own group developed the program stringMLST, which uses a distinct k-mer-based approach for genome-enabled MLST to yield extremely rapid and 100% accurate MLST sequence types directly from NGS read data. k-mers are sequence substrings, or words, of length *k*. This alignment-free k-mer-based approach represents a substantial technological advance for computational methods for genome-enabled MLST, which other groups have recently extended.

Genome sequence–based approaches for MLST can be broadly classified into two groups—(i) classic alignment–based methods that use genome assembly and/or read mapping and (ii) newer alignment-free approaches that utilize k-mers to derive sequence types directly from NGS read data (Fig. 18.3).

## Alignment-based computational methods

Alignment-based methods for MLST, or other locus-based typing schemes, entail the comparison between isolate allele sequences and typing scheme databases using sequence similarity searches (Fig. 18.3A). A number of these approaches require an assembly step to work with short read data generated by NGS platforms. Once the NGS read data are assembled into longer contiguous (contig) sequences, they are compared with allele and profile databases to generate sequence types. Examples of this kind of typing software include BIGSdb (Jolley and Maiden, 2010), MLSTcheck (Page et al., 2016), and MLSTar (Ferres and Iraola, 2018). Genome assembly is computationally expensive, in terms of both CPU time and memory, and it can require substantial bioinformatics expertise

TABLE 18.1 List of Alignment-based and alignment-free methods for multilocus sequence typing.

| Computational tool | Description | Input data type | User interface | Website | Release year | Reference |
|---|---|---|---|---|---|---|
| *Alignment-based method algorithms that utilize de novo assembly, genome mapping, and/or sequence alignment* | | | | | | |
| BIGSdb | Database and analytical platform designed for microbial loci-based typing schemes. Open-source, freeware, locally installable; base platform for PubMLST website; utilizes BLAST | Genome, gene sequences | Web/GUI | https://pubmlst.org/software/database/bigsdb | 2010 | Jolley & Maiden (2010) |
| MLSTcheck | Automated, scalable command line tool for determining MLST from genome sequences; utilizes BLAST | Genome sequences | CLI | https://www.sanger.ac.uk/science/tools/mlstcheck | 2016 | Page et al. (2016) |
| MLSTar | R-based package to determining MLST from genome sequences; utilizes BLAST | Genome sequences | CLI | https://github.com/iferres/MLSTar | 2018 | Ferres & Iraola (2018) |
| chewBBACA | Comprehensive pipeline for creation of whole- and core-genome MLST (wgMLST and cgMLST) as well as determining wgMLST/cgMLST from genome sequences using BLAST Score Ratio (BSR) | Genome sequences | CLI | https://github.com/B-UMMI/chewBBACA | 2018 | Silva et al. (2018) |
| DTU CGE MLST 2.0 | Web-based application for performing MLST analysis; utilizes de novo assembly and BLAST for MLST | Genome sequences; NGS reads | Web/GUI | https://cge.cbs.dtu.dk/services/MLST | v1: 2012 v2: — | Larsen et al. (2012) |
| SRST/SRST2 | Read-to-genome mapping—based application for performing MLST from NGS read data | NGS reads | CLI | https://katholt.github.io/srst2/ | v1: 2012 v2: 2014 | Inouye et al. (2014) |
| MOST | Modification of SRST2 for MLST analysis and *Salmonella* serotyping from NGS reads | NGS reads | CLI | https://github.com/phe-bioinformatics/MOST | 2016 | Tewolde et al. (2016) |
| ARIBA | Pipeline that performs read-to-gene mapping followed by targeted assembly | NGS reads | CLI | https://github.com/sanger-pathogens/ariba | 2017 | Hunt et al. (2017) |
| Kestral | Novel algorithm that uses k-mers and dynamic programming—based local alignment to perform MLST | NGS reads | CLI | https://github.com/paudano/kestrel | 2017 | Audano et al. (2018) |
| *Alignment-free algorithms that do not utilize assembly- or alignment-based techniques* | | | | | | |
| stringMLST | Loci-based typing using k-mer counting and hash tables | NGS reads | CLI | https://github.com/jordanlab/stringMLST/ | 2017 | Gupta et al. (2017) |
| STing | Computationally efficient implementation of stringMLST; utilizes k-mer frequencies and enhanced suffix arrays | NGS reads | CLI | — | — | Espitia et al. (2017) |

TABLE 18.1 List of Alignment-based and alignment-free methods for multilocus sequence typing.—cont'd

| Computational tool | Description | Input data type | User interface | Website | Release year | Reference |
|---|---|---|---|---|---|---|
| MentaLiST | Loci-based typing using k-mer counting followed by colored de Bruijn graph construction | NGS reads | CLI | https://github.com/WGS-TB/MentaLiST | 2018 | Feijao et al. (2018) |
| Krocus | Loci-based typing from long-read sequencing data; utilizes k-mer counting | Long-read sequences | CLI | https://github.com/andrewjpage/krocus | 2018 | Page & Keane (2018) |

to generate reliable results. As such, assembly represents a major bottleneck for genome-enabled molecular typing studies, and these approaches do not scale well when hundreds of isolates need to be characterized. Assembly-based methods are also difficult to implement for larger-scale locus-based typing schemes that employ hundreds or thousands of genome-wide loci.

Another class of algorithms for bacterial typing with NGS data uses short read mapping to reference sequences as a more computationally tractable alternative to assembly-based methods. These methods can still be considered as alignment-based, because they rely on read-to-genome alignments; nevertheless, they are substantially more efficient compared with assembly-based methods. The Center for Genomic Epidemiology (http://www.genomicepidemiology.org/) provides a genome-based web platform for MLST, which previously implemented an assembly-based approach and has since evolved to use read mapping for allele calling (Larsen et al., 2012). The first program designed specifically to do NGS-based bacterial typing via read mapping was SRST (Inouye et al., 2014), which was subsequently modified by the same group to develop SRST2 and another group to develop the program MOST for *Salmonella* serotyping (Tewolde et al., 2016). More recently, the program ARIBA implemented a hybrid approach that uses read mapping to clusters of related alleles followed by constrained assembly of reads that map to specific clusters (Hunt et al., 2017).

## Alignment-free computational methods

The development of alignment-free methods for genome-based molecular typing with NGS data was a major breakthrough that provided substantial increases in speed and efficiency compared with existing assembly or read mapping approaches. As the name implies, these methods proceed directly from raw NGS sequence read data—without any quality control, alignment, or assembly steps—to call alleles and sequence types (Fig. 18.3B). The program stringMLST, developed by our group, was the first program of this kind designed for bacterial typing directly from NGS data (Gupta et al., 2017). stringMLST was designed and implemented to provide a turn-key solution of bacterial typing from genome sequence data, with minimal requirements for computational capacity or bioinformatics expertise.

The stringMLST algorithm relies on the use of k-mer frequencies and hash tables for characterizing the sequence types of bacterial isolates directly from genome sequence read data. To type bacterial isolates from any given species, stringMLST requires a database built from the alleles of the species-specific typing scheme. To construct the typing scheme database, stringMLST generates all possible k-mers from each allele sequence in the scheme and stores them in a hash table that associates each k-mer with all of the alleles in which it can be found. To characterize an isolate sample, the stringMLST algorithm performs three steps:

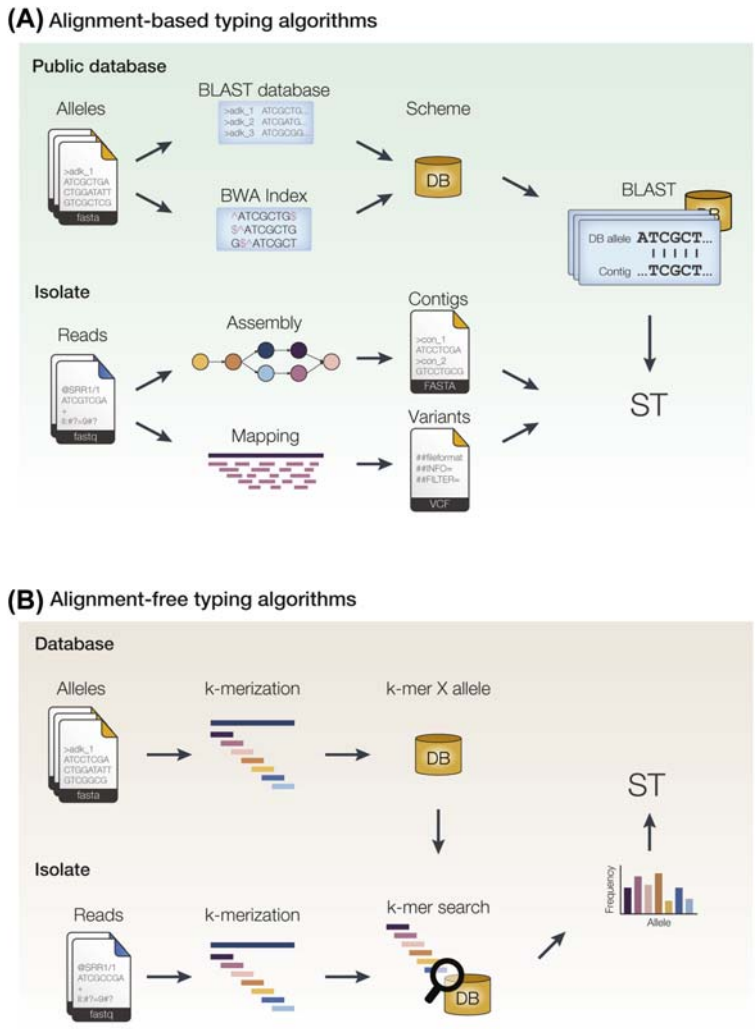


FIGURE 18.3 **Schematic comparison of alignment-based and alignment-free algorithms for sequence typing.** The figure provides the general overview of the two dominant paradigms for performing multilocus sequence typing from whole-genome sequence read datasets. Both methods utilize a database of allele sequences for each locus in the scheme and an allelic profile table that contains the mapping of allele numbers to a sequence type. (A) Alignment-based typing algorithms can be further subcategorized into assembly-based and mapping-based. Assembly-based algorithms make use of de novo genome assembly followed by sequence similarity searching algorithms such as BLAST. Mapping-based algorithms map the read sequences to either a reference genome or loci sequences, followed by variant identification. (B) Alignment-free algorithms utilize exact matching of substrings, also known as k-mers, between NGS reads and allele sequences in the database to identify the sequence type. Exact substring matching is computationally faster than genome assembly or sequence alignment, and these algorithms gain further speed by comparing only a small fraction of the input read dataset and discarding all noninformative reads.

(i) filtering, (ii) k-mer counting, and (iii) reporting. For the filtering step, the algorithm discards a read if the k-mer located in the middle of the read sequence is not present in the allele k-mer database. This heuristic step provides the bulk of the speed and efficiency to the stringMLST algorithm by passing over reads that correspond to genomic regions not covered by the typing

scheme. Because this genomic fraction corresponds to the vast majority of the genome sequence for MLST schemes, only a tiny fraction of the reads need to be fully processed by the algorithm. For the k-mer counting step, if the middle k-mer is found in the allele database, then stringMLST generates all possible k-mers from the read sequence. The algorithm then searches the read k-mers against all k-mers in the database and updates a table of k-mer frequencies for each associated allele. Steps (i) and (ii) are repeated until all of the reads are processed. For the final reporting step, the algorithm then reports the alleles with the maximum k-mer frequency for all loci in the typing scheme, thereby generating an allelic profile and calling the corresponding sequence type.

Compared with existing genome sequence–based typing tools that utilize alignment and/or the assembly, the stringMLST approach is far more efficient and at least as accurate for characterizing bacterial isolates. As reported in Gupta et al. (2017), stringMLST was the only tool able to correctly type each of 40 NGS samples from four different bacterial species (*Campylobacter jejuni*, *Chlamydia trachomatis*, *N. meningitidis*, and *Streptococcus pneumoniae*). It was up to 65× faster than other programs used to process the same datasets, showing an average of 45 s to process each sample read file. In the same study, stringMLST correctly predicted the sequence type for 99.8% of 1002 isolates of *N. meningitidis* requiring an average of 40.7 s and 0.67 MB of RAM to type each sample read file. Page et al. (2017) performed an independent comparison of eight different programs for genome-based MLST, including stringMLST as the only application on the category of alignment-free based methods. In addition to evaluating the accuracy of the tools on NGS data from past outbreaks, they evaluated the impact of sequencing depth and sample contamination on typing speed and accuracy using simulated data. Consistent with our own results, stringMLST was found to be the fastest algorithm by far and also required substantially less computational resources than any of the other programs. In addition, stringMLST proved to be 100% accurate for bacterial typing on outbreak data, comparable to slower and more cumbersome tools that rely on sequence alignment and/or assembly. It is also worth noting that stringMLST does not require any read preprocessing or quality control, making it far easier to use than the other tools and ideally suited for deployment in public health laboratories or in the field. Despite the superior performance of stringMLST for genome-based MLST, it does suffer from scaling issues when applied to larger-scale typing schemes. We cover these issues, and how we are addressing them, in the subsequent sections on genome-scale typing schemes.

Several other groups have introduced k-mer-based typing methods since the development of stringMLST. For example, the program Kestrel (Audano et al., 2018) uses a hybrid approach that combines k-mer analysis with dynamic programming–based local alignment to call MLST alleles and sequence types. However, this approach is far slower and less efficient than the k-mer-only method used by stringMLST, which is 28× faster and requires an average of ~60% of the RAM compared with Kestrel. This performance difference is likely due to the Kestrel algorithm's reliance on the exhaustive dynamic programming step. The program MentaLiST (Feijao et al., 2018) extends the stringMLST approach of using k-mer frequencies and hash tables, by constructing a colored de Bruijn graph for each allele of the typing scheme. With this addition, MentaLiST selects a subset of k-mers that embodies the variation present in the alleles of the typing scheme, resulting in a substantial reduction in the size of the allele database. This database compression allows for substantial improvement of the computational performance on larger typing schemes that utilize hundreds or even thousands of loci genome wide. We cover the computational challenges and opportunities

entailed by these so-called superMLST schemes in the following section. Yet another example of new k-mer-based typing software is Krocus (Page and Keane, 2018), designed for typing from uncorrected long-read sequence data. A problem with these kinds of data is that the current long-read sequencing technologies (Pacific Biosciences and Oxford Nanopore) exhibit high error rates. However, base errors tend to be uniformly distributed, a characteristic exploited by the Krocus developers to circumvent the high error rate problem. Perhaps the most attractive feature of Krocus is that it can type isolates in real time by taking batches of long-reads produced by sequencers that support continuous sequence streaming like those developed by Oxford Nanopore Technologies.

## Genome-enabled bacterial typing schemes

We previously described why whole-genome sequence data are still used for small-scale locus-based typing schemes such as MLST, owing to a combination of the low cost and ease of genome sequencing coupled with the epidemiological importance of MLST legacy data. Nevertheless, the ever increasing availability of numerous whole-genome sequences from bacterial pathogens (Fig. 18.2) provides both challenges and opportunities for the development of novel, large-scale typing schemes, which leverage the analysis of genome-wide variation data. Genome-scale bacterial typing schemes can be broadly categorized as (i) locus-based schemes or (ii) single-nucleotide variant (SNV)—based schemes (Table 18.2). Locus-based typing schemes are direct extensions of MLST that rely on the analysis of hundreds or thousands of loci genome wide, as opposed to the handful of loci used by MLST schemes. For example, core-genome MLST (cgMLST) schemes utilize all of the loci that correspond to the core genome with all genes shared among a set of isolates (i.e., the intersection of genes in a set of genomes). Whole-genome MLST (wgMLST) schemes are even larger-scale and use all of the genes (i.e., the union) found in a set of genomes; this approach includes both the core genome and the accessory genome. These large-scale loci-based bacterial typing schemes provide substantially more resolution than traditional MLST schemes.

In principle, SNV-based approaches to genome analysis provide even more resolution for the delineation of bacterial lineages than the largest-scale loci-based schemes, because there are far more possible single base differences among genomes than the possible number of differences among loci. As such, SNV-based schemes should be able to differentiate extremely closely related strains, down to 1 bp difference in principle. This feature makes SNV-based approaches better for microbial forensics and epidemiological studies that require extreme levels of resolution, such as source attribution in a bioterrorism event (Schmedes et al., 2016; Budowle et al., 2007) or contact tracing studies that seek to characterize the exact origins of bacterial outbreaks and their spread among patients (Stucki et al., 2015). A classic example of this approach is the single base pair resolution typing of *Vibrio cholerae* strains from the 2010 outbreak in Haiti, which ultimately pointed to United Nations peacekeepers from Nepal as the source of the outbreak (Katz et al., 2013).

Nevertheless, there are a number of reasons why locus-based typing schemes are still widely employed for bacterial typing in the postgenomic era. Perhaps, most importantly, locus-based schemes are portable and more reproducible than SNV-based schemes. Because they rely on a predefined set of loci, and the accompanying allele databases, locus-based schemes generate results that can be directly compared among laboratories and among different studies. However, SNV-based schemes rely on the use of one or more reference sequences for variant calling and are thereby more difficult to standardize among groups. The use of reference sequences

**TABLE 18.2** Comparison of locus-based and single-nucleotide variant (SNV)—based typing techniques for bacterial typing.

| | Locus-based typing | Single nucleotide variant (SNV) typing |
|---|---|---|
| Advantages | • Ideal for microbial genome analysis<br>• Allows for comparisons between different studies/outbreaks<br>• Each isolate can be easily computationally represented in a defined space<br>• Availability of several online, publicly accessible resources (tools and large databases)<br>• Standardized pipelines are available<br>• Can be configured to analyze core and accessory genome<br>• Phylogeny reconstruction methods are simpler in nature (UPGMA, eBURST) | • Ideal for identifying and characterizing closely related microbial isolates, e.g., for source attribution or contact tracing studies as well as for complex Eukaryotic genome analysis, such as human<br>• High level of discrimination power; allows inspection of every single-nucleotide change across the genome<br>• Works well if a reference genome is standardized and internationally used<br>• Can be detected using both sequencing and real-time PCR-based methods<br>• Diagnostic SNVs exists for fine subtyping of select agents |
| Disadvantages | • Requires a curated database of alleles and profile definitions<br>• Loci-based schemes are often restricted to genic regions and does not capture variation in intergenic (or intronic) regions<br>• Captures gene presence/absence but fails to capture other large structural variations, e.g., duplications and rearrangements | • Comparison between different studies/outbreaks is limited due to differences in reference genome<br>• Requires an evolutionarily close reference genome, preferably finished<br>• Mostly captures the core genome; misses variations in accessory genome<br>• SNV calls are dependent on filtering criterion used<br>• Does not capture large structural variation events, viz., insertion/deletion (indels), duplications, and rearrangements large<br>• Computational storage grows exponentially as SNV data typically involves representing all sites across the genome<br>• Commonly used phylogenetic methods are computationally intensive (Neighbor-Joining, Maximum Likelihood)<br>• Fails to capture high horizontal gene transfer (HGT) |

for variant calling with SNV-based schemes can also lead to a loss of information with respect to accessory genes, which are often important determinants of virulence for bacterial pathogens. Locus-based schemes, on the other hand, can readily accommodate important accessory genes via presence/absence calls for those loci. Additional details on the relative strengths and weaknesses of locus-based versus SNV-based typing schemes can be found in Table 18.2. Given the continued importance of locus-based typing schemes for genome-enabled bacterial typing, we focus on the computational approaches used for these kinds of schemes in the following sections.

## Computational approaches to large-scale typing schemes

As with MLST, large-scale bacterial typing schemes that leverage genome-wide datasets can be computationally implemented using traditional alignment/assembly-based methods or with the newer k-mer-based approaches. However, it is becoming increasingly apparent that the traditional methods lack the computational speed and efficiency needed to implement such schemes for rapid bacterial typing. For example, approaches that use de novo assembly followed by BLAST can take upward of 12 h for each isolate for cgMLST and/or wgMLST

schemes, which require the analysis of thousands of loci per isolate. As such, the traditional methods will become increasingly irrelevant for epidemiological studies that need to type scores, hundreds, or even thousands of isolates. For this reason, we focus here on the latest developments in the computational approaches for large-scale bacterial typing schemes.

We previously discussed how the application of the first k-mer-based approaches for bacterial typing in the stringMLST algorithm resulted in orders of magnitude speed-up for MLST without any loss of accuracy. However, the stringMLST algorithm did not scale well to large-scale typing schemes like cgMLST. When stringMLST was applied to schemes of this kind, it did not compute any faster than alignment/assembly-based approaches and required an unrealistically large amount of memory to run. This performance was because the underlying hash-table data structure used for the allele k-mer database are not optimally suited for large-scale typing schemes, because it entails the storage of all existing k-mers for thousands of loci. As previously discussed, the more recently developed program MentaLiST addressed this challenge by using a de Bruijn graph to substantially compress the allele k-mer database while also providing for enhanced searching of the database. This revised data structure provides for robust—rapid and accurate—bacterial typing using large-scale typing schemes directly from NGS read data. Our own group is currently developing the algorithm STing (as a successor to stringMLST) that employs a more efficient data structure, thereby allowing for genome-based typing with large-scale schemes.

STing is being developed and implemented for both bacterial typing and gene detection directly from unprocessed NGS read data (Espitia et al., 2017). The STing algorithm stores the allele k-mer databases for large-scale typing schemes using an enhanced suffix array data structure as opposed to the simpler hash table used by stringMLST. The suffix array provides for a substantially compressed representation of the allele k-mer database as well as rapid search capability along the array. STing has been applied to MLST, cgMLST, and wgMLST schemes for a wide variety of bacterial pathogens. It can also be used for automated gene detection directly from read sequences, and this utility is currently being validated in the context of antimicrobial resistance genes and virulence factors (e.g., Shiga toxin). Preliminary results on the performance of STing are very promising, and a more detailed description of both the algorithm and its accuracy is currently in preparation.

## Community adoption of genome-based bacterial typing

As we have mentioned several times, the genome revolution provides both amazing opportunities and profound challenges to the public health community. In principle, genome sequence data provide for unprecedented levels of resolution for bacterial typing, while also generating abundant material for the discovery of the genetic determinants of antibiotic resistance and virulence. Nevertheless, there are substantial technical hurdles that need to be overcome to ensure that the community can fully adopt genome-enabled approaches to molecular epidemiology along with the new bioinformatics techniques that they necessitate.

One key feature of early sequence-based bacterial typing schemes—MLST in particular—was portability in terms of the ability to broadly share uniformly comprehensible typing results among member laboratories distributed among surveillance networks. Portability refers to both the typing techniques, which should be standardized so that they can be carried out in any laboratory, and the typing results, which should have the same representation irrespective of where the results are generated. MLST is ideally suited for portability as it relies on a shared set of loci (allele) sequence definitions and produces

granular and static sequence types from the typing scheme's allelic profiles. Larger-scale typing schemes face a number of challenges to ensure that they both (i) remain completely portable and (ii) allow for comparison with the results of previous generation typing techniques.

The challenge to portability for genome-scale typing schemes is directly related to the scale of these schemes, which can cover hundreds or thousands of loci genome wide. The large scale of these schemes necessitates a highly coordinated effort to standardize the loci (allele) definitions that underlie the schemes and entails far more complicated allelic databases than is the case for MLST schemes, which typically utilize seven to nine loci. With respect to loci definitions, there needs to be an agreement concerning exactly which loci are used for any scheme and which part (i.e., sequence fragment) of each locus is used for typing. This aspect is relatively straightforward for schemes with a few loci but is substantially more complex when hundreds or thousands of loci are used. Furthermore, because genome-scale typing schemes are being independently developed in multiple public health laboratories around the world, numerous different versions of the same typing scheme can end up being used. With respect to allelic databases, despite the fact that thousands of bacterial pathogen genome sequences have already been characterized, allelic and profile databases for larger schemes are either incomplete or do not yet exist. A coordinated effort by the public health community will be needed to address these issues and ensure that genome-enabled typing schemes remain standard and portable. This process needs to happen soon, because it will be difficult for individual laboratories, or particular surveillance networks, to change their typing schemes once they are developed and implemented.

Another critical issue for genome-enabled typing schemes will be the ability to maintain some connection to the vast amount of historical information contained in results generated from smaller-scale legacy typing schemes. In other words, genome-scale typing schemes should be backward compatible, to whatever extent possible, with previous typing schemes such as MLST or even the nonsequence-based pulsed-field gel electrophoresis (PFGE) typing scheme. Public health laboratories will need to dedicate a substantial amount of bioinformatics expertise and effort to map the results of genome-scale typing schemes to the results of legacy typing schemes. An illustrative example of this challenge is the US Centers for Disease Control and Prevention (CDC) PulseNet surveillance network (https://www.cdc.gov/pulsenet/). PulseNet was established in 1996 as a network of public health laboratories around the United States dedicated to surveillance and outbreak detection for food and waterborne illness caused by a prioritized set of bacterial pathogens. PulseNet laboratories use a restriction enzyme—based technique to digest genomes of bacterial pathogen isolates. Subsequently, PFGE generates characteristic DNA fingerprints of the digested genomes, which are captured as distinct banding patterns on a gel. The implementation of PFGE across the PulseNet surveillance network allowed for the discovery of clusters of disease that corresponded to outbreaks, thereby leading to better coordinated and more rapid responses to such public health threats. PulseNet's use of the relatively low resolution and clearly outdated PFGE technique is expected to be phased out starting in 2019, after which time reliance will be exclusively on the far higher-resolution genome-enabled typing schemes. Nevertheless, given the amount of invaluable epidemiological information that is tied to specific PFGE patterns, it will be critically important to be able to relate the results of genome-scale typing schemes to previously characterized patterns. Accordingly, CDC scientists are working to develop approaches for the probabilistic association of PFGE patterns and genome sequence variation, and our own laboratory is involved in this effort via a collaboration with

the CDC's Enteric Diseases Laboratory Branch within the Division of Foodborne, Waterborne, and Environmental Diseases (DFWED).

The challenges for genome-enabled typing schemes outlined above—relating to uniform data standards, typing scheme portability, and backward compatibility—also suggest a pressing need for shared analytical platforms that can be deployed in public health laboratories around the world. Generating whole-genome sequence data is now rapid, cost-effective, and highly standardized. Accordingly, the rate-limiting step for genome-enabled bacterial typing corresponds to the suite of computational analysis tools and methods that need to be used to handle and interpret the massive volumes of data generated by NGS platforms. Here, we are considering mainly the software challenges entailed by the use of NGS data for bacterial typing, but there are also substantial hardware issues that need to be addressed. The sheer volume of data alone poses a fundamental challenge with respect to both computational storage and processing capacity. It is not realistic to expect that all public health laboratories will be able to address these joint challenges via the deployment of local computational capacity. In fact, we are closely reaching the point where it will cost less to sequence bacterial genomes than to store the resulting sequence data for an extended period of time. Similarly, the computational processing power needed to handle hundreds or thousands of genome sequences of bacterial isolates is likely out of reach for all but the most well-funded public health laboratories.

Cloud computing environments, whereby computational storage and processing are provisioned as services that are accessed remotely over the Internet, offer an attractive alternative to the deployment of local computational capacity for bacterial genome analysis. One of the most compelling features of cloud computing is the flexibility entailed by the on-demand model whereby investigators only make use of the amount of computational capacity that they need at any given moment. This relates to both processing power, in terms of the number and architecture of compute cores that can be accessed for any given analysis, and the elastic nature of cloud data storage capacity, with different models of data access for short-term and longer-term storage. Over the last 5 years, there has been a concerted effort to deploy computational genomics algorithms and pipelines across a variety of cloud computing platforms. In Table 18.3, we show examples of cloud computing resources in support of bacterial genome analysis with respect to both specific bioinformatics software packages as well as integrated bioinformatics platforms. The integrated platforms allow users to utilize existing bioinformatics analysis pipelines and/or build their own custom pipelines, which employ multiple applications to execute an entire workflow.

Despite the promise of the cloud computing model for computational genomics, there is currently no standardized cloud computing platform to support genome-enabled bacterial typing. Given the explosion of bacterial genome sequences, coupled with the development of numerous genome-scale typing schemes, we anticipate a pressing need for the cloud deployment of a standardized genome analysis platform in support of genome-enabled bacterial typing in public health laboratories. A shared analytical platform of this kind should consist of (i) a uniform set of bioinformatics analysis tools, (ii) a shared set of standard analysis protocols or pipelines for the use of these tools, and (iii) a set of well-defined data models that cover both input and output standards for the bioinformatics tools as well as the loci (allele) databases that underlie bacterial typing for multiple schemes across multiple species. This platform should also include mechanisms for storing primary NGS data and secondary data (results) generated by the analytical platform along with transparent means for sharing data and communicating results among public health laboratories. Finally, the use of a unified approach to

TABLE 18.3 Examples of integrated bioinformatics cloud computing software and platforms for microbial genome analysis.

| Resource | Description | Website | Reference |
|---|---|---|---|
| Illumina BaseSpace | Illumina's platform for subscription-based bioinformatics data analysis | https://basespace.illumina.com/ | – |
| RAST | Automated microbial genome annotation pipeline | http://rast.nmpdr.org/ | Aziz et al. (2008) |
| Galaxy | Open-source, free-to-use software for a variety of bioinformatics data analyses. Cloud support through Amazon Web Services, CloudMan, Globus Genomics | https://usegalaxy.org/ | Afgan et al. (2018) |
| CloudBioLinux | Community-driven, cloud-based bioinformatics platform | http://cloudbiolinux.org/ | Krampis et al. (2012) |
| CLIMB | UK's nationwide bioinformatics/electronic infrastructure designed to support the needs of the microbiology community | http://bryn.climb.ac.uk/ | Connor et al. (2016) |
| CloVR | A desktop bioinformatics virtual machine capable of utilizing cloud computing resources | http://clovr.org/ | Angiuoli et al. (2011) |
| Nephele | Cloud platform for microbiome data analysis | https://nephele.niaid.nih.gov | Weber et al. (2018) |

collect and distribute epidemiological metadata associated with bacterial isolates characterized as a part of routine surveillance and outbreak investigations will also be a critical component for such a platform.

We envision that the integrated cloud computing service and the standardized bacterial genome analysis platform described above could be unified into a national or global surveillance network with constituent public health laboratories as nodes that are capable of both rapidly typing bacterial isolates and widely sharing the results with other laboratory nodes around the world. To our knowledge, no such integrated platform currently exists, and perhaps even more disconcerting, there is a real possibility that genome-enabled approaches to bacterial typing will ultimately hamper efforts to share bacterial typing results among different laboratories. In particular, if different public health laboratories continue to independently develop their own genome-scale typing schemes, it will become increasing difficult, if not impossible, to meaningfully compare results among laboratories. Obviously, such an outcome should be avoided at all costs; it would be truly unfortunate if the increased resolution afforded by genome-scale typing schemes paradoxically leads to less resolution on the public health challenges to which these schemes are ultimately addressed. Ensuring that such a scenario does not come to pass will require an ongoing effort toward the development, standardization, and sharing of computational approaches to, and platforms for, genome-enabled bacterial typing.

## References

Afgan, E., Baker, D., Batut, B., van den Beek, M., Bouvier, D., Cech, M., et al., 2018. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 46, W537–W544.

Angiuoli, S.V., Matalka, M., Gussman, A., Galens, K., Vangala, M., Riley, D.R., et al., 2011. CloVR: a virtual machine for automated and portable sequence analysis from the desktop using cloud computing. BMC Bioinf. 12, 356.

Audano, P.A., Ravishankar, S., Vannberg, F.O., 2018. Mapping-free variant calling using haplotype reconstruction from k-mer frequencies. Bioinformatics 34, 1659–1665.

Aziz, R.K., Bartels, D., Best, A.A., DeJongh, M., Disz, T., Edwards, R.A., et al., 2008. The RAST Server: rapid annotations using subsystems technology. BMC Genom. 9, 75.

Budowle, B., Beaudry, J.A., Barnaby, N.G., Giusti, A.M., Bannan, J.D., Keim, P., 2007. Role of law enforcement response and microbial forensics in investigation of bioterrorism. Croat. Med. J. 48, 437–449.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., et al., 2009. BLAST+: architecture and applications. BMC Bioinf. 10, 421.

Connor, T.R., Loman, N.J., Thompson, S., Smith, A., Southgate, J., Poplawski, R., et al., 2016. CLIMB (the Cloud Infrastructure for Microbial Bioinformatics): an online resource for the medical microbiology community. Microb. Genom. 2, e000086.

Espitia, H., Chande, A.T., Jordan, I.K., Rishishwar, L., 2017. Method of sequence typing with in silico aptamers from a next generation sequencing platform. In: Office USPaT. US 15/726,005.

Feijao, P., Yao, H.T., Fornika, D., Gardy, J., Hsiao, W., Chauve, C., et al., 2018. MentaLiST–a fast MLST caller for large MLST schemes. Microb. Genom. 4 (2) https://doi.org/10.1099/mgen.0.000146 (https://www.ncbi.nlm.nih.gov/pubmed/29319471).

Ferres, I., Iraola, G., 2018. MLSTar: automatic multilocus sequence typing of bacterial genomes in R. PeerJ 6, e5098.

Gupta, A., Jordan, I.K., Rishishwar, L., 2017. stringMLST: a fast k-mer based tool for multilocus sequence typing. Bioinformatics 33, 119–121.

Hunt, M., Mather, A.E., Sanchez-Buso, L., Page, A.J., Parkhill, J., Keane, J.A., et al., 2017. ARIBA: rapid antimicrobial resistance genotyping directly from sequencing reads. Microb. Genom. 3, e000131.

Inouye, M., Dashnow, H., Raven, L.A., Schultz, M.B., Pope, B.J., Tomita, T., et al., 2014. SRST2: rapid genomic surveillance for public health and hospital microbiology labs. Genome Med. 6, 90.

Jolley, K.A., Maiden, M.C., 2010. BIGSdb: scalable analysis of bacterial genome variation at the population level. BMC Bioinf. 11, 595.

Katz, L.S., Bolen, C.R., Harcourt, B.H., Schmink, S., Wang, X., Kislyuk, A., et al., 2009. Meningococcus genome informatics platform: a system for analyzing multilocus sequence typing data. Nucleic Acids Res. 37, W606–W611.

Katz, L.S., Petkau, A., Beaulaurier, J., Tyler, S., Antonova, E.S., Turnsek, M.A., et al., 2013. Evolutionary dynamics of *Vibrio cholerae* O1 following a single-source introduction to Haiti. mBio 4.

Krampis, K., Booth, T., Chapman, B., Tiwari, B., Bicak, M., Field, D., et al., 2012. Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. BMC Bioinf. 13, 42.

Larsen, M.V., Cosentino, S., Rasmussen, S., Friis, C., Hasman, H., Marvig, R.L., et al., 2012. Multilocus sequence typing of total-genome-sequenced bacteria. J. Clin. Microbiol. 50, 1355–1361.

Maiden, M.C., Bygraves, J.A., Feil, E., Morelli, G., Russell, J.E., Urwin, R., et al., 1998. Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. Proc. Natl. Acad. Sci. U.S.A. 95, 3140–3145.

Maiden, M.C., Jansen van Rensburg, M.J., Bray, J.E., Earle, S.G., Ford, S.A., Jolley, K.A., et al., 2013. MLST revisited: the gene-by-gene approach to bacterial genomics. Nat. Rev. Microbiol. 11, 728–736.

Page, A.J., Keane, J.A., 2018. Rapid multi-locus sequence typing direct from uncorrected long reads using Krocus. PeerJ 6, e5233.

Page, A.J., Taylor, B., Keane, J.A., 2016. Multilocus sequence typing by blast from de novo assemblies against PubMLST. J. Open Source Softw 1, 118-11.

Page, A.J., Alikhan, N.F., Carleton, H.A., Seemann, T., Keane, J.A., Katz, L.S., 2017. Comparison of classical multi-locus sequence typing software for next-generation sequencing data. Microb. Genom. 3, e000124.

Schmedes, S.E., Sajantila, A., Budowle, B., 2016. Expansion of microbial forensics. J. Clin. Microbiol. 54, 1964–1974.

Silva, M., Machado, M.P., Silva, D.N., Rossi, M., Moran-Gilad, J., Santos, S., et al., 2018. chewBBACA: a complete suite for gene-by-gene schema creation and strain identification. Microb. Genom. 4 (3) https://doi.org/10.1099/mgen.0.000166 (https://www.ncbi.nlm.nih.gov/pubmed/29543149).

Stucki, D., Ballif, M., Bodmer, T., Coscolla, M., Maurer, A.M., Droz, S., et al., 2015. Tracking a tuberculosis outbreak over 21 years: strain-specific single-nucleotide polymorphism typing combined with targeted whole-genome sequencing. J. Infect. Dis. 211, 1306–1316.

Tewolde, R., Dallman, T., Schaefer, U., Sheppard, C.L., Ashton, P., Pichon, B., et al., 2016. MOST: a modified MLST typing tool based on short read sequencing. PeerJ 4, e2308.

Wang, X., Jordan, I.K., Mayer, L.W., 2015. A phylogenetic perspective on molecular epidemiology. In: Tang, Y.-W., Sussman, M., Liu, D., Poxton, I., Schwartzman, J. (Eds.), Molecular Medical Microbiology, second ed. Elsevier, Chennai, India, pp. 517–536.

Weber, N., Liou, D., Dommer, J., MacMenamin, P., Quinones, M., Misner, I., et al., 2018. Nephele: a cloud platform for simplified, standardized and reproducible microbiome data analysis. Bioinformatics 34, 1411–1413.