Real-time search of all bacterial and viral genomic data

Phelim Bradley, Henk C den Bakker, Eduardo P C Rocha, Gil Mcvean, Zamin Iqbal

▶ **To cite this version:**

Phelim Bradley, Henk C den Bakker, Eduardo P C Rocha, Gil Mcvean, Zamin Iqbal. Real-time search of all bacterial and viral genomic data. Nature Biotechnology, 2019, 37 (2), pp.152-159. 10.1038/s41587-018-0010-1 . hal-02329753

HAL Id: hal-02329753
https://hal.science/hal-02329753

Submitted on 23 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Title: Real-time search of all bacterial and viral genomic data

Phelim Bradley[1], Henk C Den Bakker[2], Eduardo P. C. Rocha[4], Gil McVean[5], Zamin Iqbal[1,6]

1 Wellcome Trust Centre for Human Genetics, University of Oxford, UK.
2 Center for Food Safety, Department of Food Science and Technology, University of Georgia, 1109 Experiment Street, Griffin, Georgia 30223
4 Microbial Evolutionary Genomics, Institut Pasteur, CNRS URA2171, Paris, France.
5 Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, Oxford OX3 7LF, United Kingdom.
6 EMBL-EBI, Wellcome Genome Campus, Hinxton, Cambridgeshire, CB10 1SD, UK.

Genome sequencing of pathogens is becoming ubiquitous in microbiology, and automated diagnostics will soon appear. The genome sequence archives, already growing exponentially, are currently not searchable for arbitrary sequence. Such an ability would unlock this resource for science and could underpin global real-time genomic epidemiology and surveillance.

We combine knowledge about bacterial genetic variation with ideas used in web-search, to build a DNA search engine for microbial data that can grow incrementally. We index the complete corpus of bacterial and viral whole genome sequence data as of December 2016 (447,833 genomes, 176 Terabytes), using four orders of magnitude less storage than previous methods, making the global archive for the first time accessible to search, and scaling to millions of genomes. We demonstrate its usefulness with three applications: ultra-fast search for resistance genes MCR1-3, host-range determination for 2827 plasmids, and quantification of the rise of antibiotic resistance prevalence in the archives.

Whole genome sequencing (WGS) offers unparalleled resolution for problems as diverse as contact tracing, mapping the spread of drug resistance, identifying zoonoses, and investigating the underlying biology of infectious diseases. Sequence data is deposited in the global sequence archives (European Nucleotide Archive (ENA), Sequence Read Archive (SRA)) which are doubling every two years, and we expect this to accelerate as affordable WGS-based diagnostic tests become a reality[1-6]. However, it is currently impossible to search the archives for datasets with specific mutations (single nucleotide polymorphisms (SNPs)), genes or mobile elements. The ability to combine these atomic queries would be transformative for global management of infectious disease, allowing instant access to datasets within any specified genetic distance (e.g. "has anyone in the world seen something within 20 SNPs of this strain before?), or with given drug resistance mutations, genes or plasmids.

In the 1990s, when most species had at most one reference genome and within-species variation was less of a focus, BLAST[7] and its successors[8,9] revolutionized bioinformatics by providing online DNA alignment of queries against large databases of reference genomes. However, assemblies constitute only 17% of archived bacterial data (110,898 assemblies versus 554,680 raw read datasets in the European Nucleotide Archive (ENA) as of October 2017) and generally only a fraction of those are indexed for BLAST search. Indeed, although high quality reference genomes of clonal (i.e. containing one genome) samples are the gold standard, these are unachievable with short read data [10-12], and bad references discard or confound data which may be of interest. More fundamentally, many sequence datasets contain populations rather than clonal isolates. Haploid assembly is fundamentally not the right summary for such data. As the archives continue to scale-up, it becomes critical to be able to rapidly filter down to small datasets for careful analysis. The core requirement is therefore to be able to search heterogeneous historical and modern data, whether assembled or not, for presence of arbitrary sequence. In this study, we combine computational techniques previously used in web-search, with knowledge of bacterial population genetics, to develop a data structure, the *Bitsliced Genomic Signature Index (BIGSI*), that solves this problem.

The first scalable search engine for raw sequence was developed in 2016, the Sequence Bloom Tree (SBT)[13], a k-mer (fixed-length DNA word) index, developed to enable detection of specific transcripts in RNA-seq data. This opened up a new field of "experiment discovery" – allowing users to search archives for useful experiments to study further. However, SBT and its recent successors[14-16] shared with previous similar methods (Cortex, vari, McCortex)[17-19] a scaling dependence on the total number of k-mers in the union of datasets indexed. For species where there is considerable k-mer sharing between datasets (e.g. human), this works well, and further efficiencies can be gained by leveraging patterns of sharing between datasets[20]. However, bacteria are fundamentally different: even within a species there can be enormous diversity due to horizontal transfer of DNA (Supplementary Figure 1), and we show this diversity renders previous methods unable to scale. We remove this limitation with the use of BIGSI to index the entire bacterial and viral content of the ENA as of December 2016 (447833 datasets; 170 Tbytes of data), using four orders of

magnitude less storage than previous methods (a schematic outline of these and related methods is shown in Figure 1). This is the first time the archives have been made accessible to search, and we make a version publicly available at http://bigsi.io. We demonstrate applications to basic biology and surveillance: ultra-fast search for the colistin resistance genes MCR1-3, mapping host ranges of 2827 plasmids, and plotting the changing prevalence of antibiotic resistance mutations and genes in the archives.

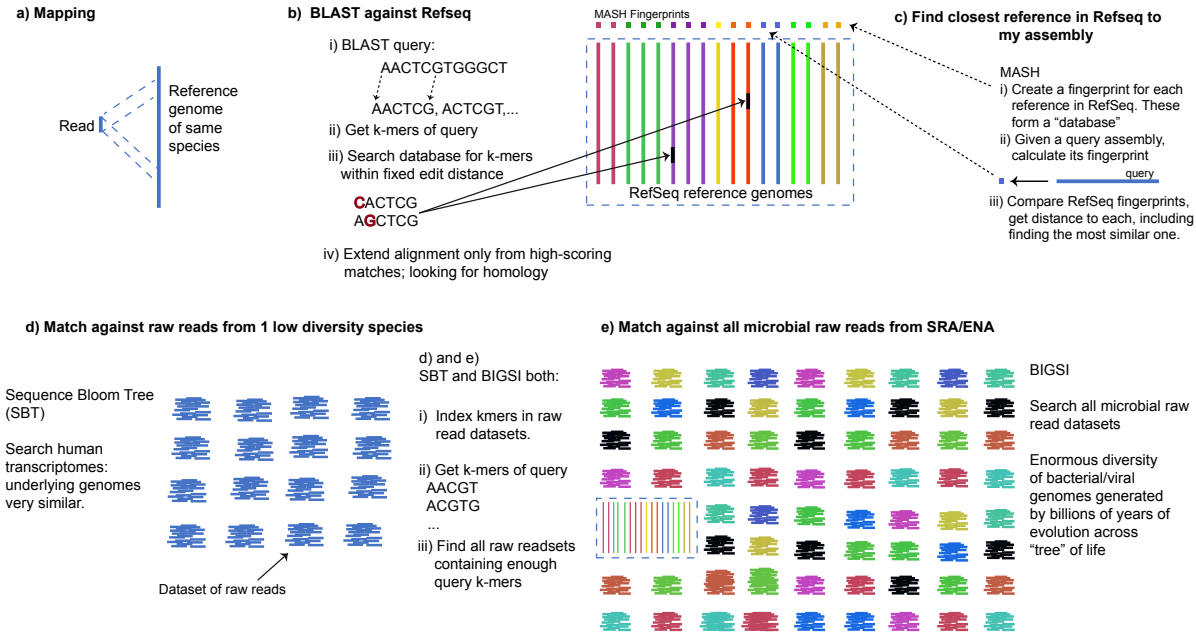

Figure 1: Schematic outline of common sequence matching methods. a) Mapping of sequence reads to a reference genome from the same species, thus assuming relatively low divergence; requirement to map millions of reads in acceptable time and return an alignment and mapping score. Common tools: bwa and bowtie. b) BLAST: Compare a query string with a database of reference genomes (here RefSeq genomes surrounded by dotted box) covering huge phylogenetic range. BLAST takes k-mers from the query, and for each creates a "neighbourhood" of k-mers within a fixed edit distance, and searches for these in the reference genome database. Alignment is only done by extending from these hits. BLAST works for both nucleotide and protein searches and can find both close and remote homology matches. c) MASH stores a tiny fingerprint of each reference in the database (here RefSeq). Querying with an assembly, the fingerprint of the assembly is compared with those of RefSeq to find closest reference. d) Sequence Bloom Tree: first scalable method to search through raw unassembled readsets, by indexing the k-mers within and then compressing the index. Designed for human data, used to find which RNA-seq datasets contain a given transcript. e) BIGSI (this study): Designed to search the complete global corpus of raw sequence data for bacteria and viruses (RefSeq shown within dotted box for sense of scale). This entails different speed/compression trade-offs as the indexed data is far less self-similar than single species data, showing imprint of billions of years of mutation and horizontal gene transfer. Both SBT and BIGSI sacrifice the ability to detect remote homology to enable scalability.

High accuracy queries with a lossy compressed DNA index

We developed a data structure suitable for storing bacterial genomic data, the *bitsliced genomic signature index* (BIGSI). We use the generic term "dataset" to refer to either assembled genomes or unassembled sequence read-files from clonal or non-clonal samples. BIGSI combines a k-mer index with constraints on sequence queries, described below. A bloom filter is a data structure[21] which stores data (here, k-mers) in a bit-vector (array of zeroes and ones) and answers set-membership queries ("is this k-mer contained in the set?") probabilistically. The false negative rate is zero, and the false positive rate is controlled by 2

parameters (size of bit-vector and the number of (hash) functions used to generate the binary encoding, see Figure 2), creating a trade-off between false-positive rate and compression. We describe how we set the bloom filter parameters below. The BIGSI encodes data as a matrix where each column is a bloom filter of the k-mers in a dataset. Figure 2 shows schematics of how data is processed and stored, and details are in Methods. Note that incorporating a new dataset simply requires a new column is added, without needing to rebuild the index. Since appending a new column to a BIGSI requires modifying every key in the index this is an expensive operation. As a result, our approach is to batch new inserts, building a new BIGSI per batch, and merging these together. Our implementation supports both disk-based and in-memory stores; all measurements reported are from the disk-based store. Although we use this as a k-mer index, it can be viewed as a probabilistic coloured de Bruijn graph[17,22].

We note that the Bloom filter parameters and the k-mer size cannot be modified dynamically, so the maximum number of k-mers in any dataset much be set in advance. A simple workaround (not applied in this study) when storing genomes of very different sizes or metagenomic data would be to use an array of BIGSIs , "sharding" across dataset size (choosing which BIGSI to store a dataset in, based on the number of k-mers), and parallelise queries (see Methods). For a comparison of functionality supported by BIGSI compared with other tools, see Methods.

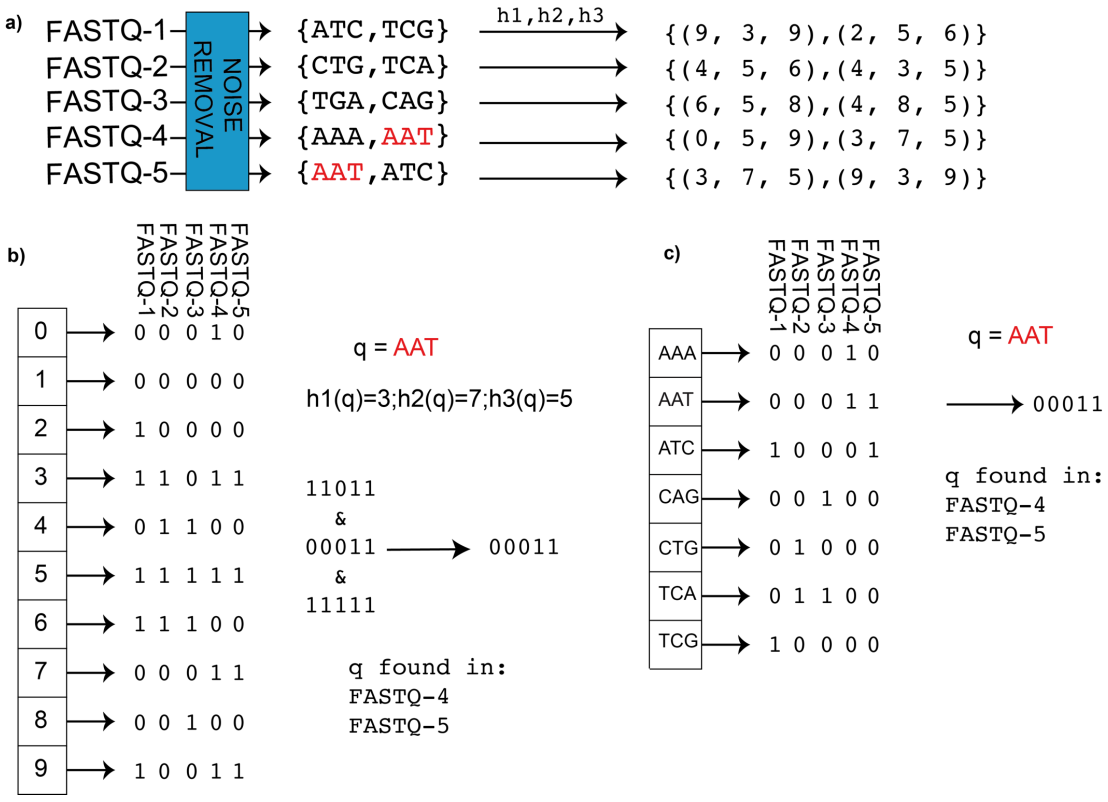

Figure 2: BIGSI encoding compared with naïve approach. a) BIGSI step 1: each input dataset (could be raw sequence data (FASTQ format) or assembly) is converted to a non-redundant list of k-mers (with an optional denoising step to remove sequencing errors, detailed in Methods). A fixed set of η hash functions ($h_1, h_2, \ldots$) is applied to each k-mer (η =3 in this figure), giving a tuple of positions which are all be set to 1 in a bit-vector (a Bloom Filter). b) BIGSI step 2. Each dataset is stored as a fixed length bloom filter, as a column in a rectangular matrix. To query the BIGSI for k-mer AAT, the η hash functions are applied to the query k-mer, returning η rows to be checked (namely 3,7,5 here). All columns (datasets) that have 1 in all of those η rows contain the query k-mer: these rows that are checked are called "bitslices". A hash, mapping from row index to corresponding bit-array is stored to allow fast, i.e. O(1), access to each row when needed.

Adding a new dataset requires just adding a new column. c) Naïve encoding for contrast. A complete list of all k-mers in all datasets form the rows of a large matrix, and columns are datasets. For any given k-mer, entries are set to one for datasets containing that k-mer. When a new dataset is added, the matrix grows vertically (new k-mers added) and horizontally (new column for new dataset).

To genotype a sequence, we query the index for all the k-mers within it. Exact matching requires all k-mers be present (threshold T=100%), and can be implemented as a fast AND operation on bit-vectors. Inexact matching, primarily used for long alleles, requires the presence of some proportion (T<100%) of k-mers be present, and is slower.
The relationship between proportion of k-mers present ("k-mer identity") and the more traditional sequence identity used by BLAST, is non-linear but monotonic. For example, if the k-mer size is 31, each SNP difference causes a window of 31 absent k-mers. Therefore k-mer identity drops more rapidly than sequence identity, and BIGSI is only appropriate for finding matches which are relatively similar (e.g for k=31, matches with sequence identity above 80% (as shown in Supplementary Figure 2)). BIGSI is fundamentally designed for situations needing an exact or moderately close match, or where combinatorial searching is feasible (e.g. querying all sequences 2 SNP differences from a given allele, or all single amino acid changes in a gene). Datasets containing remote matches for a gene could only be sought via searches for subsequences (seeds).

Although BIGSI does not do an alignment, an approximation to a mega-BLAST alignment score can be inferred from the presence/absence pattern of k-mers in the query (details in Methods). We show in Supplementary Figure 3 the strong correlation (r=0.998) between Mega-BLAST score and BIGSI score for 100 *E. coli* AMR genes using a BIGSI of RefSeq-bacteria (release 81).

To search for a single nucleotide polymorphism (SNP), we create a sequence for each allele, with k-mer -1 bases on either side. By requiring multiple k-mers in the sequence be present, we reduce the false positive rate for SNP allele detection exponentially. Indexing at a smaller k-mer (31) than our minimum query length (61) enables both compression and a low error rate. The theoretical false discovery rate for an SNP allele from a probe (flanks plus allele) of length (2k-1) with bloom filter parameters is 10^{-35} per column (see methods) - well below the expected error rate from the underlying sequence data.

We measured query speed by first building a BIGSI of 3,480 datasets of *Mycobacterium tuberculosis* obtained from[23], and genotyping 68,269 SNPs. Searching all datasets for these SNPs took just under 90 minutes on a single CPU core - an effective genotyping rate of above 46,000 genotypes per second.

We validated SNP genotyping accuracy using a subset of 100 of the *M. tuberculosis* datasets for which we had high quality SNP calls using samtools[24] (see Methods). The concordance between methods was 99.997% with a total of only 286/682,690 discrepancies. We measured accuracy of longer allele detection by searching (with T=70% match) for a catalogue of *E. coli* Multi Locus Sequence Type (MLST) alleles and choosing the best scored allele for each gene. We then compared calls on a set of 954 datasets with the MLST allele calls from a high-quality caller: SRST2[25]. Where both methods made a call (6483/6678 alleles), there was 99.9% agreement; otherwise SRST2 failed (n=167), or BIGSI failed to find an allele version above T=70% (n=28).

Benchmarking and scaling.

We benchmarked the empirical scaling properties of BIGSI against the Sequence Bloom Tree (SBT)[13] and the Split Sequence Bloom Tree[14] (SSBT) on a dataset of 10,000 random microbial sequence datasets from the ENA, comparing build and query times, and peak storage requirements on 21 increasing subsets ranging from 100 to the complete 10,000 samples. For each subset, we built a BIGSI, and both SBT and SSBT each using two different parameter settings: one optimized for speed (recommended by the authors in Lemma 1, and subsequent text, of Solomon et al[13] and the SBT/SSBT user manual - termed SBT-fast/SSBT-fast), and another optimized for compression (SBT-small/SSBT-small). See methods for further details. For database sizes >= 2000, peak storage requirements of SBT-fast and SSBT-fast exceeded our available disk space (1Tb; by comparison the total storage required for the input data was 403Gb; see methods).

We queried the resulting indexes for 2157 antimicrobial resistance genes with a mean length of 937bp and total query length 2,021,655bp. SBT-fast, SBT-small, SSBT-fast, SSBT-small and BIGSI returned near-identical hits from the exact match search, with only 1 difference across all queries (measured on N=1000 database size). With the inexact search, concordance was >99% with the differences likely due to the different construction of the underlying bloom filters between the methods.

Inexact (T=40%) query times vs. peak storage requirements for the various methods can be seen in Figure 3; BIGSI maintains good query performance in small space for all input data sizes, whereas (since they need to build an uncompressed tree of bloom filters before compressing it) both SBT and SSBT require trading storage requirements for performance. SSBT-fast and SBT-fast have query time comparable to or better than BIGSI but require orders of magnitudes more storage to build the uncompressed tree. SBT-small and SSBT-small have slower query performance than BIGSI using near equivalent storage due to saturation of internal bloom filters within the sequence bloom tree, a result of the total number of unique k-mers being significantly larger than the number of unique k-mers in any individual dataset. As mentioned above, SBT-fast and SSBT-fast construction failed for databases >=2000 due to excessive disk requirements, but we could nevertheless calculate a lower bound for the peak disk-use for database sizes >=2000; query times were extrapolated linearly (these points are shown as triangles on Figure 3; details of lower bound in Methods). A similar picture is seen for exact queries (T=100%, see Supplementary Figure 4); exact match searches with BIGSI are up to 2-3X faster than inexact queries as they can be found by taking the bit-wise AND of the k-mer lookups and do not require “unpacking” the bit-vectors.

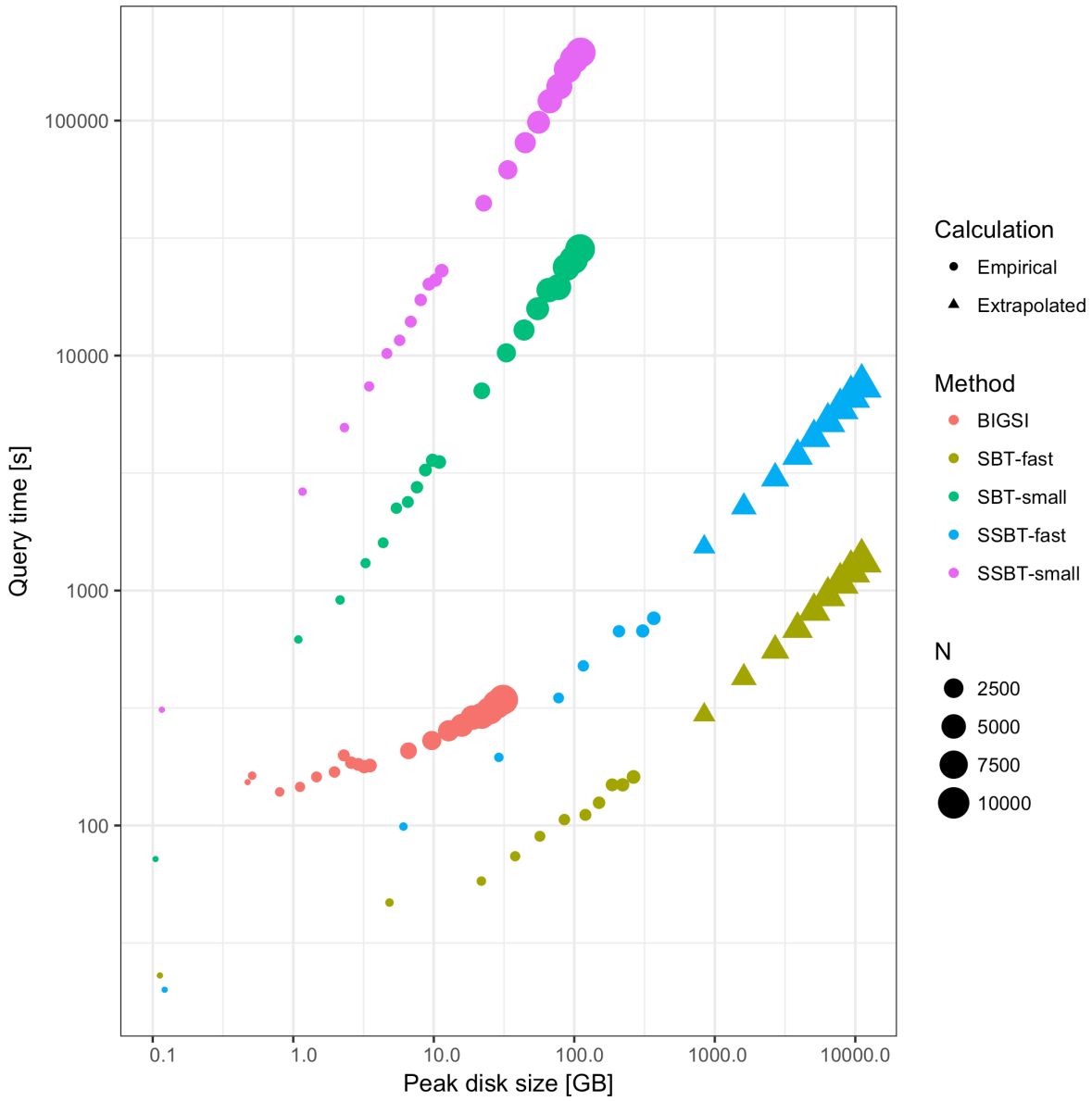

Figure 3: Query time for 2157 antimicrobial resistance genes with T=40% vs. peak disk size when searching databases of sizes from 10–10,000 microbial datasets. Both axes are on a log scale; diameter of dot represents the number of datasets indexed – thus to compare two methods it is necessary to compare dots of the same size. The ideal method would produce dots towards the bottom-left. For the same database size, SBT-small and SSBT-small are slower than BIGSI, as expected due to internal node saturation. Where construction was possible, for the same database size, SBT-fast and SSBT-fast have faster or comparable query times as BIGSI, but require significantly larger disk space to build due to the large number of total k-mers in the database. For database sizes greater than 1000, we were unable to build the SBT-fast/SSBT-fast as their uncompressed disk usage exceeded available space; triangles signify estimated values based a calculated lower bound for disk use (as k-mer content is known), and extrapolated query times (see below and Methods).

There were approximately 4.4×10^9 unique k-mers in the union of all 10,000 datasets, almost 1000 times more than in a typical individual dataset. This would require ~11TB of storage to build SBT-fast, 350X more space than is required by BIGSI. By contrast, querying SBT-small took 83X longer and SSBT-small 568X longer than a BIGSI of the same database size. SSBT-small was nearly 7X slower than SBT-small, perhaps due to the lower proportion of shared k-mers between datasets. Even for this small benchmark, constructing unsaturated

(i.e. fast) SBTs and SSBTs for larger numbers of datasets quickly becomes prohibitive in storage requirements; we address scaling further below. Query times for SBT/SSBT-slow were unacceptable for this benchmark, and so we exclude them as candidates for storing the ENA/SRA, which is 50X bigger.

To ensure our SBT settings were fair, we also benchmarked BIGSI and SBT on the same human RNA-seq dataset used in [13,14,16], using the prebuilt SBT index provided by the authors. We measured the query time of 1000 RNA transcripts randomly selected from the 214,294 known transcripts (reported in [13]). BIGSI has faster query time than SBT and takes smaller space for these datasets (360s and 144Gb for BIGSI, and 2221s and 200Gb for SBT (T=70%); full results including for exact search in Supplementary Table 1). However, we expect that, without additional improvement to BIGSI, both SSBT and Mantis will perform better on human genomic data, as they can take advantage of the higher levels of similarity between human genomes.

Finally, we simulated the scaling of storage requirements required to construct SBT-fast, and BIGSI for data sizes up to 1 million genomes in two regimes: firstly, for genomes with high proportions of k-mer-sharing (e.g. human), and secondly to species with lower proportions of k-mer-sharing (e.g., most bacteria) - see Methods for details, and Supplementary Figure 5). BIGSI scales linearly with the number of datasets, performing identically in both cases. In the low-k-mer sharing regime (which is our focus) an unsaturated SBT/SSBT would require 4 orders of magnitude more storage than BIGSI to construct (tens of Pb rather than 3 Tb).

Indexing all bacterial and viral WGS data

We set out to construct a BIGSI from all bacterial and viral WGS data-sets in the ENA "pathogens endpoint" (which contains all bacteria, all viruses and some eukaryote parasites, totaling 469,654 datasets). After excluding the eukaryotic genomes on the basis of size (see Methods), we were left with 447,833 datasets. The entire index required 1.5TB of storage, <1% of the original data size (170 TBytes) and contained more than 60 billion unique k-mers. Data download took 6 weeks, constructing bloom filters on the fly. Combining the bloom filters afterwards took approximately 2 days (See Methods). We estimate that the intermediate storage required to build an SBT-fast/SSBT-fast of the same data using recommended bloom filter size equal to the number of unique k-mers in the collection, would have been >6.7PB.

We base a number of large-scale analyses below on this index, which we refer to as the all-microbial-index. In order to make statements about which genus certain mobile elements or alleles are found in we estimated the species and abundances present in each dataset that went into the all-microbial-index using the Bayesian abundance estimator Bracken[26] which parses output from the read-classifier Kraken[27] (see Methods). We found over 90% of the datasets were from just 20 genera, and 65% were from the top 5 most common bacterial genera (*Salmonella*, *Streptococcus*, *Staphylococcus*, *Escherischia* and *Mycobacterium*); counts for the most prevalent bacterial genera are shown in Supplementary Figure 6.

Application 1: ultrafast gene search

As a practical example, we searched for exact matches of the colistin resistance genes MCR-1/2/3, subject of intense scrutiny over the last 2 years[28-32] across all 447,833 datasets in the all-microbial-index. Searching for all 3 genes took 1.73 s seconds in total, scanning 10x more genomes than previous publications. MCR-2 was not present, but we found MCR-1 in 169 datasets of 3 species (*E. coli*, *S. enterica*, *E. aerogenes)* and MCR-3 in 34 datasets (*E. coli*, *S. enterica*, *K. pneumoniae*) (see Supplementary Data 1).

Application 2: estimating the host-range of plasmids and conjugative systems

We took 2827 plasmids from the ENA (see Methods, and Supplementary Data 2) and ran an inexact (T=40%) search for these in the all-microbial-index. We filtered these for hits with T>90% for downstream analysis. The total length of query sequence was 227 Mbp, and the query took 2120 CPU hours (11 days real time) on a single server using 8 cores and 1.5 Gb RAM per process. The search returned 665,619 hits with 121,758 unique accessions across 258 genera. Since contamination could confound observations of a plasmid in a genus, we excluded from this analysis (Application 2) all datasets containing evidence of more than one genus at abundance above 0.1%. Only 41% (=184652) of datasets and 62% of search hits passed this filter.

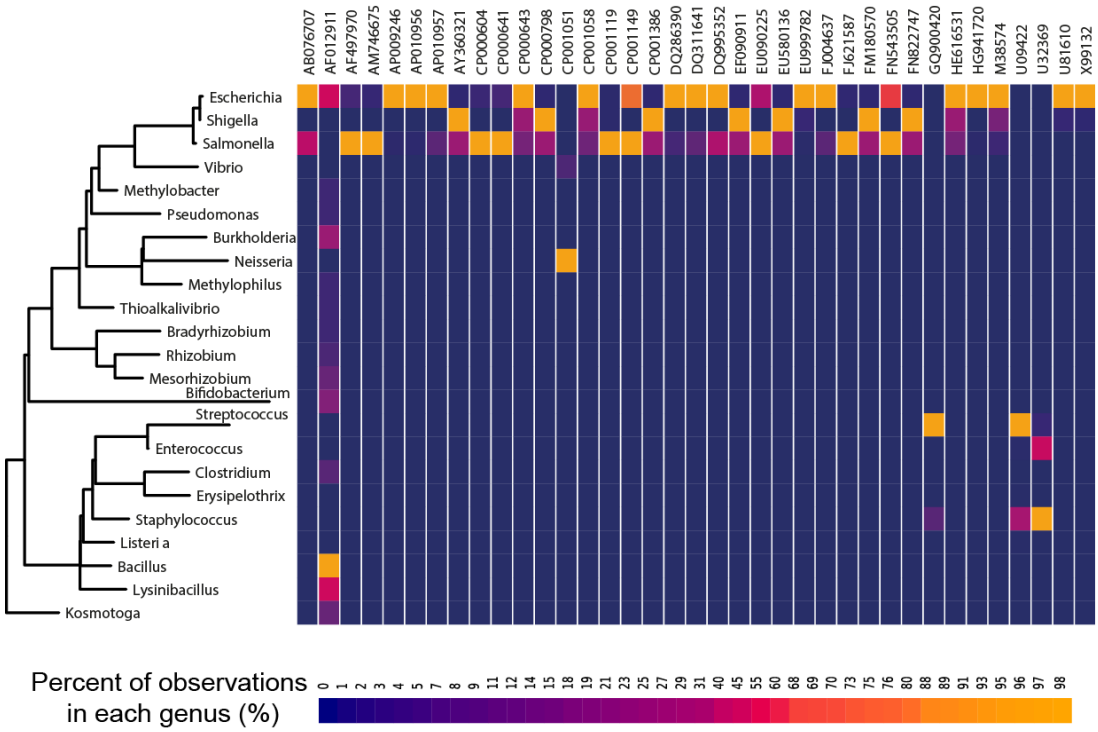

Figure 4: 37 plasmid sequences found at least 5 times in more than one genus in the all-microbial-index. The heatmap shows the frequency of each plasmid within each genus. The plasmids/genera were hierarchically clustered using the UPGMA algorithm and euclidean distance metric. The plasmid at the left (AF012911) with extremely wide phylogenetic distribution is a known cloning vector. The large amount of sharing between *Escherischia*, *Salmonella* and *Shigella* is consistent with known promiscuity within *Enterobacteriaceae*

We often identified plasmids shared by closely related genera, notably among *Escherichia*, *Shigella* and *Salmonella*, and among *Enterococcus*, *Streptococcus* and *Staphylococcus*. We found 37 plasmids present in at least five datasets of at least two genera (Figure 4, Supplementary Data 2 & 3); and 5 in multiple orders and families. The plasmid pETHIS-1

(entry: AF012911) was found in 5 phyla, 10 taxonomic classes, and 17 genera. This plasmid is used as an expression vector and its identification in so many species serves as a positive control, confirming that BIGSI can spot similar plasmids across the database. Of more biological interest, the Tn916 conjugative transposon encoding tetracycline resistance that was first found in *Enterococcus faecium* and known to have broad host range[33] (entry: U09422) was found in Streptococcus (n=3951), Staphylococcus (n=1212), Enterococcus (n=43), Clostridioides (n=29), Listeria (n=19) and Erysipelothrix (n=11).

Sampling biases in the ENA prevent inference about prevalence, but they do allow us to ask if plasmids bearing antibiotic resistance (ABR) genes are more widely phylogenetically distributed than those bearing none. We defined "phylogenetic spread" of a plasmid as the median of the pairwise distances along the tree (incorporating branch lengths) between all pairs of genera in which the plasmid is seen (we used a large subunit rRNA tree). Figure 5 shows the distributions for plasmids bearing at least 3 ABR genes (abbrev. 3ABR, purple), and those bearing none (abbrev. zero-ABR, peach). We test whether 3ABR plasmids are more widely distributed across the phylogeny than zero-ABR plasmids by comparing the 95% quantile of these two distributions, and find that they are (95% quantiles: 1.11 and 1.99 for zero-ABR and 3-ABR respectively), and test for significance using a permutation test (p=0.0024, 1 million replicates, see Supplementary Figure 7). Given the underlying data, we would want to replicate this with wider sampling of the phylogeny to be confident of this result beyond the *Enterobacteriaceae*.

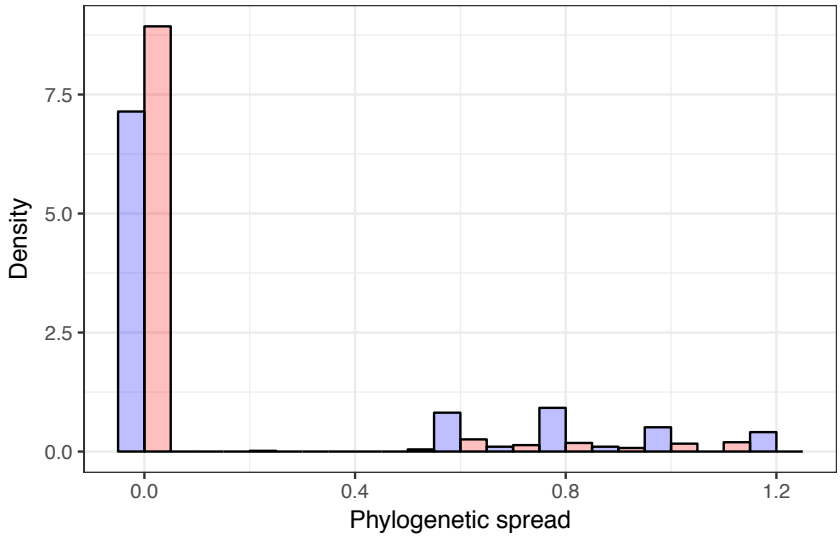

Figure 5: Comparison of phylogenetic spread (median of pairwise distances between all pairs of genera in which a plasmid is seen) of plasmids containing at least 3 antibiotic resistance genes (n=98, purple) with those bearing none (n=665, peach) – histograms are normalized to allow comparison (probability densities). Distance measured on the large subunit rRNA tree from SILVA. Units of phylogenetic spread are substitutions per site; it is possible to have a distance>1 since it is measured up to the common ancestor and back down again.

The distribution of different versions of the machinery (relaxase (MOB) and type IV secretion system (T4SS)) for conjugative transfer of DNA between bacteria have previously been analysed in 1124 genomes[34] using sensitive, but slow, protein profiles searches. We extended this analysis to the whole ENA, searching (exact match) the all-microbial-index for

previously identified MOB and T4SS types. Of the 184,652 datasets, 36030 (19.5%) had a putative conjugative system (i.e. exact matches to at least one MOB type and T4SS) - consistent with the previous estimate of 18%. This proportion varied by phylum, from 0.5% in *Spirochaetes* to 31.7% in *Firmicutes* (see Supplementary Table 2). Supplementary Figure 8 shows the distribution of MOB types in each phylum. At a finer scale these observations provide valuable information on the potential spread of antibiotic resistance genes. For example, focusing on datasets with MOB_T we observe genetic flux between *Staphylococcus* and *Streptococcus*, but not with *Salmonella*, and using this information facilitates the assessment of the risk of spreading between taxa. This flux does not need to strictly follow phylogenetic lines. For example, we observed MOB_Q in *Salmonella* and *Streptococcus* but not *Staphylococcus*, indicating a different probability of cross-genus (and cross-phylum) transfer by conjugation (data in Supplementary Data 4).

Application 3: growth of antibiotic resistance prevalence in the archives

We downloaded all 2157 sequences associated with antibiotic resistance from the CARD database (v1.1.7)[35] searched for these in the all-microbial-index with thresholds of 100% and 70%. An exact search for a gene on average took 1.1s and returned 438 hits. In total this resulted in 944,862 hits in 193,582 unique accessions across 250 genera (Full results in Supplementary Data 5). An inexact search (T=70%) on average took 34.4s and returned 5320 hits. We show in Figure 6a the growing count of ABR genes in the archives, split by year of upload to the SRA/ENA. Restricting to *Staphylococcus*, we find (Figure 6b) the proportion of datasets containing *mecA* gene (causing methicillin resistance) dropping from a high of 70% in 2013 to 40% in 2016, during which period all the *tet* and *aac* genes also drop in prevalence. By contrast, we show in Figure 6c that for *Klebsiella* essentially all resistance genes have gone up in archive prevalence.

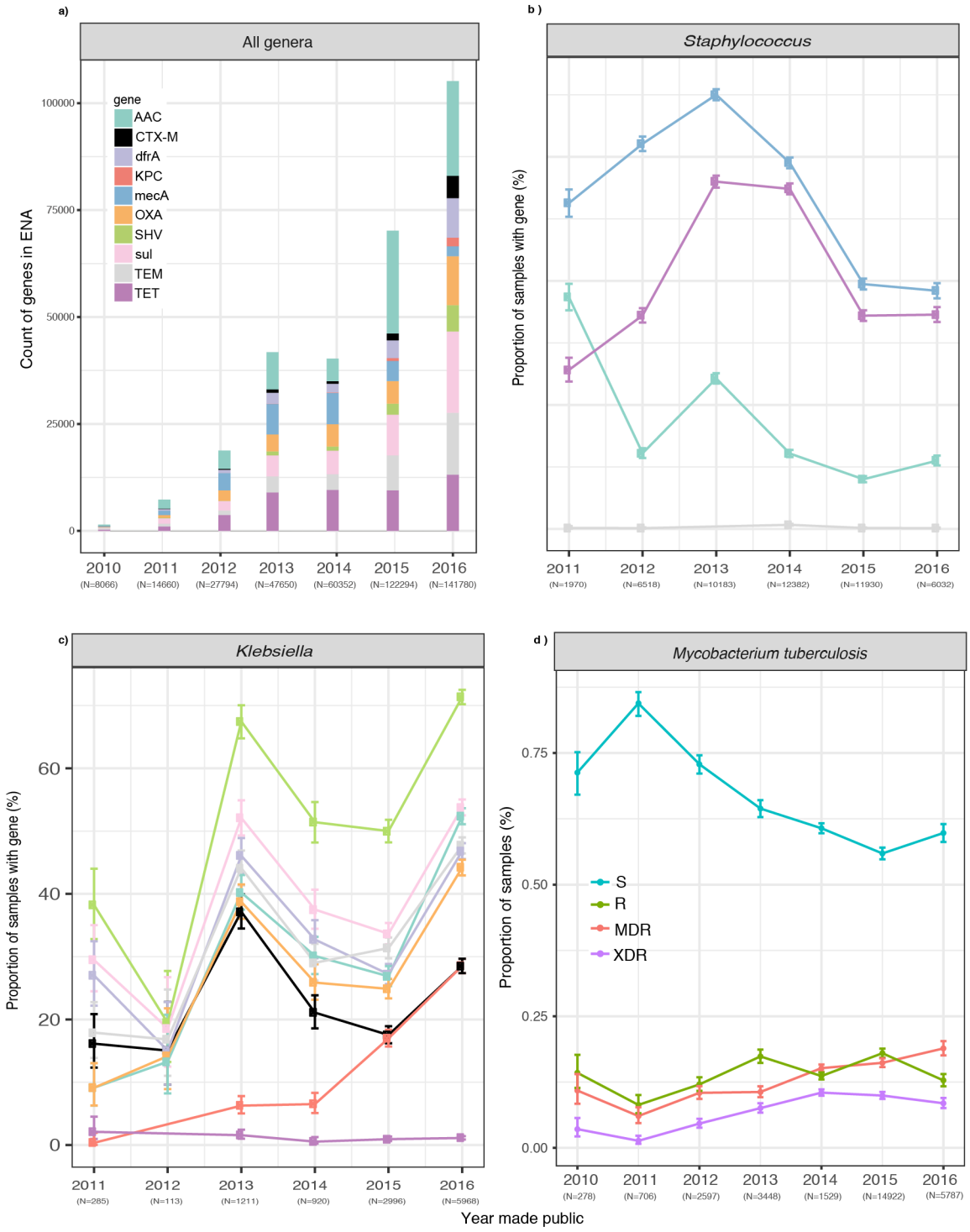

Figure 6: a) counts of samples in the all-microbial index containing a range of ABR genes; each gene treated independently, so a single dataset containing both CTX-M and OXA for example, will be counted twice b) Year-by-year frequency (defined by date of public availability) in *Staphylococci* (dominated by *S. aureus*) of *mecA*, and all *tet* and *aac* genes, which encode resistance to methicillin, tetracycline and aminoglycosides respectively. Archive-prevalence dropping for all since 2013. c) Year-by-year frequency in Klebsiella of various ABR genes; increase in prevalence since 2014 may be due to increased Extended Spectrum Beta-Lactamase surveillance and sampling of KPC resistant *Klebsiella* globally. d) Year-by-year breakdown of *M. tuberculosis* datasets, classified by genotypes as resistant (R), pan-susceptible (S), multiple drug resistant (MDR), extensively drug resistant (XDR) as follows. All datasets were genotyped for variants from the resistance catalog from[23], then classified as resistant or susceptible to 12 antibiotics based on their genotype. Datasets were classed as MDR (multi-drug resistant) if resistant to isoniazid and rifampicin, as XDR (extensively drug-resistant) if MDR and also resistant to a fluoroquinolone, and any of capreomycin, kanamycin and amikacin, and as Resistant if resistant to any antibiotic but not MDR or XDR, and susceptible otherwise.

In *M. tuberculosis,* resistance is driven primarily by mutations affecting amino acids in protein coding genes [1,36]. Genotyping all datasets in the all-microbial-index simultaneously (of which 30,226 were *M. tuberculosis*) for the 206 resistance mutations from Walker et al. [23] took 103 minutes on a single-core, around 10,000x faster than typing each dataset individually with the fast resistance prediction software, *Mykrobe predictor*[1]. The results show (Figure 6d) a rise in prevalence in the archive of MDR-TB since 2011. The contents of the ENA reflect the sampling biases of academic studies, and do not reflect unbiased global sampling. Indeed, the latest WHO estimates for 2016[37], put MDR prevalence at 6.6% (compare: 18.9% in 2016 in Figure 6b).

Discussion

There is an urgent need for a global infrastructure for surveillance and management of infectious disease - microbes know no borders, and evolve faster than we modify our responses[38,39]. Vital analytic and visualization tools for SNP-based analyses are being developed in response to emerging viral outbreaks[40], but the problems of scalability have not been addressed. We foresee a world where millions of bacterial and viral samples have been sequenced and shared, some from very controlled and high quality clinical and public-health sources providing high-value metadata (e.g. the open Genome Trakr database of food-borne pathogens in the USA), and others of varying provenance. A scalable online sequence search facility would be critical to this endeavor. It would provide data not just for urgent outbreaks, but also for monitoring global strain, plasmid and resistance prevalence in humans, animals and the environment. Filtering by metadata content and by provenance would be vital. We have demonstrated the core operations needed for such a search tool, on a scale never before achieved, indexing the entire bacterial and viral WGS content of the global DNA archive. Since new datasets can be rapidly appended to the index, the ability to grow incrementally as new datasets are sequenced is guaranteed. Finally, the method is ready for a future where finished reference genomes become routine, as the index works equally for both raw data and assemblies.

BIGSI was designed with SNP/indel genotyping and allele search queries in mind, allowing the user to zoom in on datasets worthy of detailed study, as well as enabling global monitoring (e.g. of antimicrobial resistance). However, there are limitations. First, as with the SBT, it is a k-mer index and is therefore as lossy as all de Bruijn methods – reconstruction of stored genomes is impossible and repeat regions can not be resolved. Secondly, BIGSI does not store coverage information – this rules out queries where copy number is important, such as detecting azithromycin resistance in *Neisseria gonorrhea* where the level of resistance is mediated by the number of rRNA genes containing a particular SNP. Third, although BIGSI can be considered a coloured de Bruijn graph, it is optimized for search (genotyping), not traversal of the graph. Although one could in theory search for remote homologs using exact matching of short seeds followed by graph traversal, this would take more software development to handle false positive edges. Fourth, BIGSI does not support very short queries, or very low complexity queries where the number of unique k-mers is low, as these result in a high false discovery rate. Fifth, BIGSI currently only supports nucleotide k-mers.

In fact this is not a fundamental limitation - an extension to amino acid search would be straightforward, as the bloom filters are completely agnostic to what they are storing. Finally, BIGSI is optimized for very diverse datasets, where the combined unique k-mer count is much higher than that of any individual sample (e.g. the entire microbial ENA/SRA). Where this is not true (e.g. indexing very similar samples) we expect SSBT or Mantis are likely to be the better choice as they take better advantage of compression from sample similarity.

Searching the DNA archive is one example of a "document retrieval" problem, a subject which has been intensely studied and successfully implemented at massive scale by internet search engines. Our "search terms" are k-mers from SNPs/alleles, and our "web-pages" are raw read datasets or assemblies. Similar approaches to ours (also using bitsliced signatures) have been used for text search[41,42], but were largely abandoned after Zobel et al showed in 1998 that an alternative method (inverted indexes) performed better for natural language[43]. One notable exception in 2017 was the Microsoft Bing search engine[44], which also revives them. For our use case, where each new bacterial dataset brings new variation, bitsliced signatures provide much better scaling than inverted indexes. Web and (microbial) DNA search have different dimensionality, as the language of bacterial genomes is vastly more complex than English. Our dataset was only 10^6 documents but contained 10^{10} unique words, and this would continue to increase with more data, whereas Google indexes 10^{12} documents containing (we estimate) 10^8 words with a much more slowly growing lexicon. As a result, we expect fruitful future interactions between the genomic and document retrieval communities.

In the future we envisage, as sequencing-based diagnostics take-off, with data volumes growing and the user-base expanding to clinical and public health practitioners, there will be a huge need for sequence search across the global corpus of microbial DNA. A combination of MASH, and both nucleotide and protein BIGSI would be extremely powerful. We are currently investigating implementing the BIGSI as a live service at the EMBL-EBI, updated as data is added to the ENA. We believe our approach, and improvements that will surely follow, will put our shared DNA store at everyone's fingertips.

Online Methods

BIGSI construction and querying

BIGSI indexes a set of N (number of datasets) bloom filters by position in the bloom filter. Each bloom filter must be constructed with the same parameters (m, η), where m is the bloom filter's length in bits and η is the number of hash functions applied to each k-mer. The same hash functions must also be used to construct each bloom filter. To construct a BIGSI, the N bloom filters are column-wise concatenate into a matrix. The row index and row bit-vectors are then inserted into a hash table or key-value store as key-value pairs so that row lookups can be done in O(1) time. This set of key-value pairs can be stored on disk, in memory or distributed across several machines and is indexed via a hash index (a b-tree would be an alternative option). To insert a new bloom filter we simply append it as a column to the existing bitmatrix. To query the BIGSI for a k-mer we hash the k-mer η times, look up the resulting keys in the key-value store, and take the bit-wise AND of the resulting bit-vectors (See Figure 1).

Parameter choices

The choice of BIGSI parameters (m, η), depends on: the maximum number of k-mers expected in any dataset (K_{max}), the number of datasets (N) expected, the smallest number of unique k-mers in a query sequence (L_{min}) to be supported, the k-mer size (k) and the maximum number of acceptable false discoveries per query (q_{max}). The expected number of false discoveries (V) for any query can be calculated as $q = E[V] = Np^L$ where p is the false positive rate of the bloom filter and L is the number of unique k-mers in the query, assuming independence of the k-mers and bloom filters. Parameters m and η determine false positive rate for a bloom filter with K_{max} elements – if there are fewer elements, then the false positive rate will be lower. We assume below that all bloom filters have the maximum number of unique k-mers inserted to give an upper bound on error rate. However, the independence assumption mentioned above is strictly speaking false for real data archives (such as the ENA/SRA) which have biased distribution of datasets across the phylogeny. If there is an enrichment of datasets from some genus, then those columns (bloom filters) will be more similar, and conditional on a false positive in one column, the probability of a false positive in the similar columns will be elevated.

To keep q below a chosen threshold $q < q_{max}$ for a given N and k, p must be chosen to satisfy q for the smallest number of unique k-mers in a query L_{min}:

$$p^{L_{min}} = \frac{q_{max}}{N}.$$

Therefore, the desired bloom filter false positive rate is

$$p = \left(\frac{q_{max}}{N}\right)^{\frac{1}{L_{min}}}.$$

Since, for a given number of inserted k-mers (n), and desired false positive rate (p), optimal bloom filter parameters can be determined by the following formula[45]

$$m = -\frac{nln(p)}{\ln(2)^2},$$
$$\eta = -\frac{ln(p)}{ln(2)},$$

which becomes:

$$m = -\frac{K_{max} ln(q_{max}/N)}{\mathcal{L}_{min} ln(2^2)},$$

$$\eta = \frac{ln(q_{max}/N)}{\mathcal{L}_{min} ln(2)}.$$

For example, given

$$N = 10^6; K_{max} = 10^7; L_{min} = 50bp; k = 31; q_{max} = 10^{-6}$$

the resulting expected number of false positives per k-mer-lookup per bloom filter (p), would be:

$$p = \left(\frac{q_{max}}{N}\right)^{\frac{1}{L_{min}}} = \frac{1}{10^{\frac{3}{5}}} = 0.2511\ldots$$

Solving the above equations gives:

$$m = 28{,}755{,}176; \eta = 2$$

Finally we note that any path through the de Bruijn graph which was not in the original genome will be classified as present by BIGSI (as all the k-mers are present), creating a false positive which is not considered in the above modelling. This can only happen if a query includes k-mers repeated in a genome.

BIGSI parameters for all-microbial-index

We assume initially that a bacterial dataset contains at most 10 million k-mers since bacterial genomes are generally under 6 Mb in length, leaving 4 million k-mers available for some sequencing errors which escape de-noising, and plasmid variation. Unless otherwise specified we use BIGSI parameters m=25,000,000, $\eta = 3$ for all analyses. For these parameters, the upper bound on number of false discoveries, q_{max} for an SNP allele from a probe (flanks plus allele) of length (L_{min} =2k-1=61) is 10^{-9} (if $K_{max} = 10^7, N = 10^6$).

BIGSI implementation

An open source implementation of BIGSI can be found at https://github.com/phelimb/BIGSI. BIGSI v0.2.0 supports both disk-based indexing via Berkeley-DB, or rocksDB. It can be extended to in-memory (via python dictionaries), and distributed in-memory (via redis (https://redis.io)) key-value stores. The benchmarking uses the rocksDB key-value store and v0.2.0, and the all-microbial BIGSI uses Berkeley-DB and version v0.1.7.

Comparison with other tools

There are a wide range of tools for sequence matching and alignment, so we provide here a table outlining some of the more popular, and how what they do is (dis)similar to BIGSI.

Table 1: Comparison of DNA-search tools

Tool	Query	Target/Database	Use	Examples
Kraken	Read	Taxonomy of reference genomes	Find lowest place on taxonomy which could emit that read.	Estimating species content of a metagenomic dataset
bwa	read	Reference genome	Map and align	Map reads to a reference of the same species
MASH	Reference	Set of reference	Find closest	Guess species

	genome or raw read dataset	genomes and/or raw read datasets	reference genome or sample. Cannot do sequence search	of a reference, or find a similar sample to your query.
BLAST	Sequence (DNA or Amino-acid)	Set of reference genomes	Alignment, find homology	What species is this read from? Which references have a homolog of MCR-1?
BIGSI	DNA sequence	Set of raw read datasets and/or references and/or assemblies	Find exact match, or inexact match sharing some proportion of k-mers. Optimised for collections with low k-mer similarity between samples (e.g. all microbes)	MLST-type the SRA. How many times has MCR1 been seen? Which species has this plasmid been seen in? Which samples have this SNP?
SBT,SSBT,mantis	DNA sequence	Set of raw read datasets	As for BIGSI but optimised for collections with high k-mer similarity between samples. (e.g. collection of human WGS)	Which datasets have this transcript

Software and public instance

We have made a public instance of our index of the ENA available at http://bigsi.io, where the user can paste sequence and search. This instance uses the redis in-RAM implementation and is hosted by CLIMB (http://www.climb.ac.uk/) on a 3Tb RAM server. The code for BIGSI is open source (MIT license) and available at https://github.com/phelimb/BIGSI.

Scoring of BIGSI queries

BIGSI search hits can optional be scored using an approximation to the ungapped alignment scoring scheme used by megaBLAST. To do this, we take the presence/absence vector for a query of L k-mers. From this, we estimate the approximate number of mismatches of the query from the hit by counting the number of zeroes in contiguous runs of length greater than

1, and dividing by the k-mer size. From these estimated mismatches and matches we calculate a score for an ungapped alignment, with p-values calculated using the same scheme as BLAST. By default the costs are -2 for a mismatch and +1 for matched position.

Benchmarking query time and storage requirements of BIGSI, SBT, and SSBT
We randomly chose 10,000 microbial cleaned de Bruijn graphs from the all-microbial-index accessions and we then further randomly sub-sampled these into collections of 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000 datasets. A BIGSI of each set of datasets was built with parameters (m = 2.5×10^7;η = 3). A SBT and SSBT were built for each set of datasets with $\eta = 1$ and bloom filter size (m) equal to the count of the total number of k-mers in the collections of graphs, as recommend in the text following Lemma 1 of Solomon et al[13], called "SBT-fast" and "SSBT-fast" respectively. Redis hyperloglog was used to count the unique k-mers in the set of cleaned graphs for each increment. A SBT and a SSBT were built for each dataset with m = 2.5×10^7;η = 3 (the same bloom filter parameters as BIGSI), called "SBT-small" and "SSBT-small".

Construction and query time analyses were run on an Amazon Web Service i3.8xlarge instance with 32vCPUs, 224 GiB of memory and 4 x 1.9 TB non-volatile SSD-backed instance storage. SBT-fast construction exceeded 1TB for 2000 datasets, the practical limit we set, and as a result SBT-fast was not built for increments above 1,000 datasets.

The search time comparison was run with 'bt query' and 'bigsi search –seqfile $f', using k-mer thresholds 40% and 100%. A full table of results can be found in Supplementary Data 6.

Simulation of storage requirements for a BIGSI for N datasets is given by:

$$BIGSI_{storage}[bytes] = \frac{mN}{8}$$

Although it is possible to append to an SBT incrementally, as new microbial datasets will keep adding new k-mers, this will lead to saturation of the root-level bloom filter in the SBT, and a collapse in query performance. This can be avoided by reconstructing the SBT, ensuring the bloom filters are large enough to support the full set of k-mers. This was borne out by our benchmarking. In simulating scaling to a million genomes, we therefore focussed on SBT-fast rather than SBT-small. As the best case for a binary tree with N leaves is 2N-1 nodes, we estimate:

$$SBT_{storage}[bytes] = \frac{(2N-1)N_k}{8}$$

where N_k is the total number of k-mers in the combined set of datasets and also equal to the size of the bloom filter required. This is a lower bound for the peak storage use. See Supplementary Data 6, for the close correspondence between this theoretical estimate and the empirical measurement in our benchmark datasets. As a result, we can calculate lower bounds for the peak disk usage for SBT/SSBT; when the explosion of disk usage made construction of indexes for the benchmark datasets with size>1000 datasets, we used this lower bound to plot (suitably labelled) extrapolated datapoints.

No benchmarking against Mantis

We attempted to benchmark against Mantis but despite assistance from the authors we were unable to resolve a number of issues: large numbers of false positive hits on some data, and a bug causing segmentation faults when trying to build more than 3000 datasets. We reluctantly excluded mantis from our benchmarking due to limitations of time.

As currently implemented, the Mantis data structure does not support incremental insertion, as it needs up-front the full set of k-mers and for each k-mer, the list of datasets containing it ("colour class", stored as a bit-vector). There is currently an intermediate stage where the uncompressed colour-class matrix is held in RAM, scaling quadratically in number of datasets.

Benchmarking query time and storage requirements of BIGSI on RNA-seq data

De Bruijn graphs (k=31) were constructed and cleaned from the downloaded RNA-seq fastq files listed in[13]. A BIGSI was built with bloom filter parameters $m = 4000000000$; $\eta = 1$ by chunking into 1,600 batches, building in parallel and then combining into a final index. The SBT and the 214,294 transcripts were provided by the authors (personal correspondence). 1,000 transcripts were randomly selected from the full set and queried with 'bt query' and 'bigsi search –seqfile $f' respectively.

Using an array of BIGSI to support variable dataset size

A limitation of BIGSI is that K_{max}, the maximum number of k-mers per dataset, must be set in advance. One way to extend BIGSI to datasets with varying k-mer cardinality is to build a nested structure of multiple BIGSIs with different K_{max}, e.g. $K_{max} = 10^5, 10^6, 10^7, etc$ ..., and insert each sequence into the appropriate level by k-mer counting before insertion.

Genotyping accuracy measurement on TB

Conservative SNP calls were made using Cortex[17] (independent workflow, k=31) on 3480 *Mycobacterium tuberculosis* datasets from Walker et al[23]. Singleton variants were discarded, and a de-duplicated list of 68,695 SNPs was constructed. We generated "probe sets" consisting of a reference and alternate alleles of these variants from the NC_000962.3 reference. An index of the 3,480 datasets was built and 100 random datasets were genotyped at the 68,695 sites as follows: Each allele of the probe-set is searched for in the BIGSI resulting in Boolean presence/absence of each allele. This requires querying for multiple probes for each variant. If only a reference allele is present the genotype is returned as 0/0, if only an alternate allele 1/1, if both 0/1 and if neither -/-. We compared the concordance of genotypes of the 100 random datasets with those generated with the samtools pipeline from Walker et al[23], excluding filtered positions. As described in the main text, the concordance between methods was 99.997% with a total of 286/682,690 discrepancies. The majority of these discrepancies (203/286) were mixed (heterozygous) calls from BIGSI; samtools had been run with a haploid model it did not make any mixed calls, so we expect some of these were correct.

Indexing of ENA snapshot

The fastqs from accessions listed in Supplementary Data 7 were downloaded via ENA's Globus FTP and included all WGS bacteria and viruses, but also eukaryotic parasites with larger genomes, which we did not intend to index. We removed the eukaryotic genomes implicitly, by setting thresholds to exclude datasets with too many k-mers for a 5Mb

genome. De Bruijn graphs (k=31) were constructed and cleaned from the downloaded fastq files using mccortex[19] v0.0.3-539-g22e27b7.

```
mccortex31 build -t 1 -m 7G -k 31 -s "DATASET_ID" -1 “FASTQ_FILES”
mccortex31 clean -m 7GB -B 2 -U -T
```

De Bruijn graph error cleaning and tip trimming were performed using mccortex. Bloom filters were built using the k-mers from the cleaned graphs with parameters (m = 2.5×10^7;η = 3, $K_{max} = 10^7$) with BIGSI v0.2 as follows:

```
cbg init –k 31 –m 25000000 –h 3
cbg bloom –c “CLEANED_GRAPH_FILE”
```

Of the full set of datasets, 4.6% (21,822/ 469,654) fastq files failed to produce a resulting bloom filter. Of these 21,822: 7,799 exceed the maximum number of k-mers allowed after error cleaning (namely 10^7) and 14,023 exceeded the maximum number of k-mers allowed in the raw dataset (namely ~7x10^9).

The unique k-mers in the union of the cleaned graphs were counted using the redis (v3.2.6) hyperloglog (https://redis.io/commands#hyperloglog) approximate cardinality counter. In the union of all cleaned graphs there were $6.05 \times 10^{10} \pm 5 \times 10^7$ k-mers. We estimate this number would have been at least an order of magnitude higher without the denoising step where mccortex removed sequencing errors.

Species identification

The proportion of species in each dataset was determined using Kraken[27] and Bracken[26]. Kraken v0.10.5 was run on the k-mers from each cleaned de Bruijn graph using the minikraken 20141208 database. The resulting taxonomy labels assigned by Kraken were then analysed by Bracken vfd88a06a to estimate the proportion of k-mers originating from each species present in a dataset. Bracken failed to report species abundance for 12,889 datasets. The taxonomic data for the remaining 434,944 datasets is reported in Supplementary Data 6.

Plasmid search and exclusion of contaminated datasets

2,826 plasmid sequences were taken from the ENA plasmid pages (www.ebi.ac.uk/genomes/plasmid.html; December 2016) (See Supplementary Data 2) and downloaded from the ENA. We then queried the all-microbial-index for these sequences with a proportion of k-mers threshold of 40% (T=40%) present and filtered for hits with T >= 90% for downstream analysis. Queries were run with 1 GB cachesize (memory) per process and parallelised across 8 vCPUs.

In order to determine distribution of plasmids across taxa, while avoiding ENA/SRA metadata errors, we filtered these hits for datasets which (were bacteria and) had no secondary genus above 0.1% frequency. This criterion was chosen to avoid multi copy plasmids from contaminating species establishing false positive hits within a non-host genus. 41% (184652/447,833) of accessions and 62% (415,181/668,720) of search hits passed this

filter. We then filtered for all plasmids which had been seen at least 5 times each in more than one genus and had less than 99% of their observations in the most frequent genus. We found 37 plasmids across 13 genera matched these criteria. By simulating mixtures of *S. enterica* and *E. coli* at relative abundances of 0.0001, 0.001, 0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3 we found we could observe the minority species above 2% frequency (the limit of detection was not lower because we had applied Kraken *after* error-cleaning of de Bruijn graphs). All 37 plasmids reported had at least one observation at a copy number of 5 (which, since we could detect contaminants at 2% frequency, would correspond to a copy number of above 250 if it came from a contaminant) and 16/37 had an observation at 2000x copy number.

Phylogenetic spread of plasmids

We excluded contaminated samples as above, and plasmids with no hits in the all-microbial-index. We used the APE R package to calculate a cophenetic distance matrix between all genera in the Silva large subunit ribosomal RNA tree (release s123_LSU, https://www.arb-silva.de/projects/living-tree/). For all plasmids, we took the N genera in which they were found and calculated the N-choose-2 distances between these using the above matrix, and took the median (mean showed same result).

Conjugative system search

MOB (MOB_B, MOB_C, MOB_CQ, MOB_F, MOB_H, MOB_P,MOB_T, MOB_V) and T4SS sequences (VirB4_TRaU, VirD4_TcpA) as defined in Guglielmini et al [34] and Supplementary Data 8 in the all-microbial-index with T=100%. Full search results are available in Supplementary Data 9. Results were filtered for bacteria and contamination following the same method as described in "Plasmid search". Accessions with at least one MOB and T4SS were said to contain a putative conjugative system. BIGSI does not return copy number, or location on chromosome or plasmid, so it was not possible to determine if the genes were co-located on a chromosome or on a plasmid.

MCR-1,2,3

We searched for MCR-1, MCR-2, MCR-3 in the all-microbial-index using k-mer percent threshold T=100%. See Supplementary Data 1 for sequences and results.

Searching for ABR genes in the ENA

We downloaded all 2157 sequences associated with antimicrobial resistance from the CARD database (v1.1.7)[35]. We searched for these in the all-microbial-index with thresholds of 100% and 70%, using a 1 Gb cache size, and 8 CPUs. A full table of the search results can be found in Supplementary Data 5.

Searching for *M. tuberculosis* variants in the ENA

We searched the all-microbial-index for the variants from the catalogue described in [23] by generating "probe sets" consisting of a reference and alternate alleles of these variants from the NC_000962.3 reference and searching for these alleles. If only a reference allele is present the genotype is returned as 0/0, if only an alternate allele 1/1, if both 0/1 and if neither -/-. From the resulting genotypes we classified each of the datasets as resistant or susceptible to 12 antibiotics following the model described in[23]. The date when this data was first available to the public was extracted from its ENA metadata. Datasets were classed as MDR (multi-drug resistant) if resistant to isoniazid and rifampicin, as XDR (extensively drug-resistant) if MDR and also resistant to a fluoroquinolone, and any of capreomycin, kanamycin and amikacin, and as Resistant if resistant to any antibiotic but not MDR or XDR, and susceptible otherwise.

Acknowledgements
We would like to thank, for critical reading and helpful suggestions: Grace Blackwell, Robyn Ffrancon, Martin Hunt, Jerome Kelleher, Janet Thornton, Rob Patro, Kerri Malone, John Marioni, Andrew Page, Simon Gog, Timo Bellman, Florian Gauger. For enormous assistance with data download from EBI: Robert Esnouf, Guy Cochrane. For hosting our BIGSI demonstration: CLIMB. We acknowledge funding from Wellcome Trust Core Award Grant Number 203141/Z/16/Z. ZI was funded by a Wellcome Trust/Royal Society Sir Henry Dale Fellowship, grant number 102541/A/13/Z. PB was funded by Wellcome Trust Studentship H5RZCO00.

1 Bradley, P. *et al.* Rapid antibiotic-resistance predictions from genome sequence data for Staphylococcus aureus and Mycobacterium tuberculosis. *Nat Commun* **6**, 10063, doi:10.1038/ncomms10063 (2015).
2 Brown, A. C. *et al.* Rapid Whole-Genome Sequencing of Mycobacterium tuberculosis Isolates Directly from Clinical Samples. *J Clin Microbiol* **53**, 2230-2237, doi:10.1128/JCM.00486-15 (2015).
3 Quick, J. *et al.* Real-time, portable genome sequencing for Ebola surveillance. *Nature* **530**, 228-232, doi:10.1038/nature16996 (2016).
4 Schmidt, K. *et al.* Identification of bacterial pathogens and antimicrobial resistance directly from clinical urines by nanopore-based metagenomic sequencing. *J Antimicrob Chemother* **72**, 104-114, doi:10.1093/jac/dkw397 (2017).
5 Votintseva, A. *et al.* Same-Day Diagnostic and Surveillance Data for Tuberculosis via Whole-Genome Sequencing of Direct Respiratory Samples. *J Clin Microbiol* **55**, 1285-1298, doi:10.1128/JCM.02483-16 (2017).
6 Shea, J. *et al.* Comprehensive Whole-Genome Sequencing and Reporting of Drug Resistance Profiles on Clinical Cases of Mycobacterium tuberculosis in New York State. *J Clin Microbiol* **55**, 1871-1882, doi:10.1128/JCM.00298-17 (2017).
7 Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J Mol Biol* **215**, 403-410, doi:10.1016/S0022-2836(05)80360-2 (1990).
8 Kent, W. J. BLAT--the BLAST-like alignment tool. *Genome Res* **12**, 656-664, doi:10.1101/gr.229202. Article published online before March 2002 (2002).
9 Morgulis, A. *et al.* Database indexing for production MegaBLAST searches. *Bioinformatics* **24**, 1757-1764, doi:10.1093/bioinformatics/btn322 (2008).
10 Arredondo-Alonso A, W. R., Schaik WV, Schurch C. On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. *Microbial Genomics*, doi:10.1099/mgen.0.000128 (2017).
11 Bradnam, K. R. *et al.* Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *Gigascience* **2**, 10, doi:10.1186/2047-217X-2-10 (2013).
12 Earl, D. *et al.* Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Res* **21**, 2224-2241, doi:10.1101/gr.126599.111 (2011).
13 Solomon, B. & Kingsford, C. Fast search of thousands of short-read sequencing experiments. *Nat Biotechnol* **34**, 300-302, doi:10.1038/nbt.3442 (2016).

14 Solomon, B. & Kingsford, C. in *International Conference on Research in Computational Molecular Biology.* 257-271 (Springer).
15 Sun, C., Harris, R., Chikhi, R. & Medvedev, P. in *International Conference on Research in Computational Molecular Biology.* 272-286 (Springer).
16 Pandey P, A. F., Bender MA, Ferdman M, Johnson R, Patro R. Mantis: A Fast, Small, and Exact Large-Scale Sequence Search Index. *Biorxiv*, doi:10.1101/217372 (2017).
17 Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. & McVean, G. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat Genet* **44**, 226-232, doi:10.1038/ng.1028 (2012).
18 Muggli, M. *et al.* Succinct colored de Bruijn graphs. *Bioinformatics* **33**, 3181-3187, doi:10.1093/bioinformatics/btx067 (2017).
19 Turner, I., Garimella, K., Iqbal, Z. & McVean, G. Integrating long-range connectivity information into de Bruijn graphs. *Biorxiv*, doi:10.1101/147777 (2017).
20 Almodaresi, F., Pandey, P. & Patro, R. in *17th International Workshop on Algorithms in Bioinformatics (WABI 2017).* (Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik).
21 BH, B. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* **13**, 422-426, doi:10.1145/362686.362692 (1970).
22 Pell, J. *et al.* Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. *Proc Natl Acad Sci U S A* **109**, 13272-13277, doi:10.1073/pnas.1121464109 (2012).
23 Walker, T. M. *et al.* Whole-genome sequencing for prediction of Mycobacterium tuberculosis drug susceptibility and resistance: a retrospective cohort study. *Lancet Infect Dis* **15**, 1193-1202, doi:10.1016/S1473-3099(15)00062-6 (2015).
24 Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078-2079, doi:10.1093/bioinformatics/btp352 (2009).
25 Inouye, M. *et al.* SRST2: Rapid genomic surveillance for public health and hospital microbiology labs. *Genome Med* **6**, 90, doi:10.1186/s13073-014-0090-6 (2014).
26 Lu J, B. F., Thielen P, Salzberg SL. Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science* **3**, doi:10.7717/peerj-cs.104 (2017).
27 Wood, D. E. & Salzberg, S. L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol* **15**, R46, doi:10.1186/gb-2014-15-3-r46 (2014).
28 Hu, Y., Liu, F., Lin, I. Y., Gao, G. F. & Zhu, B. Dissemination of the mcr-1 colistin resistance gene. *Lancet Infect Dis* **16**, 146-147, doi:10.1016/S1473-3099(15)00533-2 (2016).
29 Lu, X. *et al.* MCR-1.6, a New MCR Variant Carried by an IncP Plasmid in a Colistin-Resistant Salmonella enterica Serovar Typhimurium Isolate from a Healthy Individual. *Antimicrob Agents Chemother* **61**, doi:10.1128/AAC.02632-16 (2017).
30 Matamoros S, H. J., Arcilla MS, Willemse N, Melles D, Penders J, Vinh TN, Thi HN, COMBAT consortium, Jong MDd, Schultsz C Global Phylogenetic Analysis Of Escherichia coli And Plasmids Carrying The mcr-1 Gene Indicates Bacterial Diversity But Plasmid Restriction. *Biorxiv* (2017).
31 Xavier, B. B. *et al.* Identification of a novel plasmid-mediated colistin-resistance gene, mcr-2, in Escherichia coli, Belgium, June 2016. *Euro Surveill* **21**, doi:10.2807/1560-7917.ES.2016.21.27.30280 (2016).
32 Yin, W. *et al.* Novel Plasmid-Mediated Colistin Resistance Gene mcr-3 in Escherichia coli. *MBio* **8**, doi:10.1128/mBio.00543-17 (2017).

33 Ciric L, J. A., Elvira de Vries L, Agerso Y, Mullany P, Roberts AP The Tn916/Tn1545 Family of Conjugative Transposons. *Madame Curie Bioscience Database [Internet]* (2013).
34 Guglielmini, J., Quintais, L., Garcillan-Barcia, M. P., de la Cruz, F. & Rocha, E. P. The repertoire of ICE in prokaryotes underscores the unity, diversity, and ubiquity of conjugation. *PLoS Genet* **7**, e1002222, doi:10.1371/journal.pgen.1002222 (2011).
35 Jia, B. *et al.* CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. *Nucleic Acids Res* **45**, D566-D573, doi:10.1093/nar/gkw1004 (2017).
36 Eldholm, V. & Balloux, F. Antimicrobial Resistance in Mycobacterium tuberculosis: The Odd One Out. *Trends Microbiol* **24**, 637-648, doi:10.1016/j.tim.2016.03.007 (2016).
37 Organisation, W. H. Global Tuberculosis Report 2017. (2017).
38 Gardy, J. & Loman, N. Towards a genomics-informed, real-time, global pathogen surveillance system. *Nature Reviews Genetics* (2017).
39 Schatz, M. C. & Phillippy, A. M. The rise of a digital immune system. *Gigascience* **1**, 4, doi:10.1186/2047-217X-1-4 (2012).
40 Hadfield J, M. C., Bell SM, Huddleston J, Potter B, Callender C, Sagulenko P, Bedford T, Neher RA. Nextstrain: real-time tracking of pathogen evolution. doi:https://doi.org/10.1101/224048 (2017).
41 Wong HKT, L. H., Olken F, Rotem D, Wong L. in *Proceedings of the 11th international conference on Very Large Data Bases* Vol. 11 (ed Alain Pirotte and Yannis Vassiliou) 448-457 (Stockholm, Sweden, 1985).
42 Shepherd MA, P. W., Chu CK. A Fixed-Size Bloom Filter for Searching Textual Documents. *The Computer Journal* **32**, doi:10.1093/comjnl/32.3.212 (1989).
43 Zobel, J., Moffat, A. & Ramamohanarao, K. Inverted Files Versus Signature Files For Text Indexing. *ACM Trans. Database Syst.* **23**, 453-490, doi:10.1145/296854.277632 (1998).
44 Goodwin B, H. M., Luu D, Clemmer A, Curmei M, Elnikety S, He Y. in *40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* (ACM).
45 Broder, A. & Mitzenmacher, M. Network Applications of Bloom Filters: A Survey. *Internet Mathematics* **1**, 485-509, doi:10.1080/15427951.2004.10129096 (2004).