Published in final edited form as:
Nat Biotechnol. 2019 February ; 37(2): 152–159. doi:10.1038/s41587-018-0010-1.

Ultra-fast search of all deposited bacterial and viral genomic data

Phelim Bradley[1], **Henk C Den Bakker**[2], **Eduardo P. C. Rocha**[3,4], **Gil McVean**[5], and **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, Georgia, USA

[3]UMR 3525, CNRS, Paris, France

[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, UK

[6]EMBL-EBI, Hinxton, UK

Abstract

Exponentially increasing amounts of unprocessed bacterial and viral genomic sequence data are stored in the global archives. The ability to query these data for sequence search-terms would facilitate both basic research and applications such as real-time genomic epidemiology and surveillance. However, this is not possible with current methods. To solve this problem, we combine knowledge of microbial population genomics with computational methods devised for web-search to produce a searchable data structure named Bitsliced Genomic Signature Index (BIGSI). We indexed the entire global corpus of 447,833 bacterial and viral whole genome sequence datasets using 4 orders of magnitude less storage than previous methods. We applied our BIGSI search function to rapidly find resistance genes MCR-1/2/3, determine the host-range of 2827 plasmids, and quantify antibiotic resistance in archived datasets. Our index can grow incrementally as new (unprocessed or assembled) sequence datasets are deposited and can scale to millions of datasets.

Whole genome sequencing (WGS) of bacteria and viruses offers unparalleled resolution for a whole range of research questions including contact tracing, mapping the spread of drug

*Corresponding author, zi@ebi.ac.uk.

Author Contributions

ZI, GM designed and oversaw the study, PB invented the method, developed software and performed analyses, HdB performed analyses for the plasmid study, EPCR co-designed and analysed the conjugative system and plasmid analysis, ZI wrote the paper, all authors made detailed feedback on the paper.

Competing Financial Interests

GM is a cofounder of, holder of shares in, and consultant to Genomics PLC.

Europe PMC Funders Author Manuscripts

resistance, identifying zoonoses and investigating the biology of infectious diseases. Genome sequence data is deposited in public archives such as the European Nucleotide Archive (ENA) and the Sequence Read Archive (SRA), which mirror their underlying data. The volume of archived microbial sequence data has doubled in size every two years, and is likely to grow at an increasing pace1–6. However, it is currently impossible to search these archives for DNA sequence of interest, such as mutations or genes. This functionality would transform infectious disease management, by enabling rapid identification of already-sequenced organisms that are highly similar to an outbreak strain, or by enabling studies of antibiotic resistance mutations, genes or plasmids.

At first sight BLAST7 and its successor8,9 algorithms seem to enable these searches, by performing alignment of query sequences against large databases. There are two reasons why these tools do not suffice. First, they would not scale to databases the size of the ENA or beyond. Second, they require assembled genomes as input - if applied to raw sequence data they would only find matches completely contained within a single read. However assembly is fundamentally lossy when the input data contains multiple strains, and the highly heterogeneous historical data in the ENA would result in very variable assemblies, particularly of plasmids 10–12.

The first scalable method for searching raw sequences, named Sequence Bloom Tree (SBT)13, comprised a k-mer (fixed-length DNA word) index. SBT enabled detection of specific transcripts in RNA-seq data, and allowed users to search archived datasets for the first time. SBT, methods based on SBTs14–16 and methods such as *Cortex*, *vari* and *McCortex*17–19 are all constrained by a scaling dependence on the total number of k-mers in the union of datasets indexed. For species such as humans, where there is considerable k-mer sharing between datasets, SBTs work very well; efficiencies can also be gained by leveraging patterns of sharing between datasets20. However, bacterial and viral genomes encapsulate billions of years of evolution; indeed even within one bacterial species there can be enormous diversity due to horizontal transfer of DNA (Supplementary Figure 1). We show this diversity renders previous methods unable to scale. We solve the problem with an alternative type of k-mer index, using a fixed-length binary “signature” of each dataset. We call our data structure a Bitsliced Genomic Signature Index (BIGSI).

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. Bitsliced signatures were once used for text search21,22, but were largely abandoned after Zobel et al showed in 1998 that an alternative method (inverted indexes) performed better for natural language23. One notable exception in 2017 was the Microsoft Bing search engine24, which revived their use. For our use case, where each new microbial 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 microbial genomes is vastly more complex than English. Our ENA/SRA dataset was only 10^6 documents but contained 10^{10} unique terms (here, k-mers), and this would continue to increase with more data, whereas Google indexes 10^{12} documents containing (we estimate) 10^8 terms with a much more slowly growing lexicon.

Here, we use BIGSI to index the entire bacterial and viral WGS content in the ENA as of December 2016 (see Figure 1 for context), and apply it to solve basic microbial surveillance problems. We also make a demonstration search publicly available at http://bigsi.io.

Results

We developed a data structure suitable for storing microbial genomic data, the bitsliced genomic signature index (BIGSI). We use the generic term “dataset” to refer to either an assembled genome 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 structure25 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 and Online Methods), creating a trade-off between false-positive rate and compression. We describe how we set the bloom filter parameters below. BIGSI encodes data as a matrix in which each column is a bloom filter of the k-mers in a dataset. Querying for presence of a k-mer involves applying the Bloom filter hash functions to this k-mer, which each return an integer, and taking the rows of the matrix (“bitslices”) indexed by these integers; datasets (columns) containing the k-mer have a 1 in all those rows. Fast (O(1)) access to the rows is achieved using a hash mapping from row index to the corresponding bitarray. Figure 2 shows how data is processed and stored, and how queries are made; details are in Online Methods. Incorporating a new dataset simply requires the addition of a new column, without needing to rebuild the index. However, since appending a new column to a BIGSI requires modifying every key in the index this is an expensive operation. Our solution is to batch new inserts, building a new index per batch, and then merging indexes. Our implementation supports both disk-based and in-memory stores but all measurements in this report are from the disk-based store. Although we use BIGSI as a k-mer index, it can be viewed as a probabilistic coloured de Bruijn graph17,26. A comparison of functionality supported by BIGSI compared with other tools is shown in Supplementary Table 1.

To search for 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 as the bits are unpacked and summed. 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 (Online Methods, Supplementary Figures 2,3). In creating BIGSI we use a relatively high per k-mer error rate of ~0.3 to enable a higher bloom filter compression rate, but require a longer minimum query length (61) (Online Methods). For example, to search for and genotype a single nucleotide polymorphism (SNP), we create a sequence probe for each allele, with k-1 bases on either side, and demand multiple k-mers from the query be present, reducing the false positive rate for SNP allele detection exponentially. The theoretical false discovery rate for an SNP allele from a probe of length (2k-1) with our bloom filter parameters is 10^{-29} per column, which is well below the expected error rate from the underlying sequence data (~10^{-2}).

We measured query speed by first building a BIGSI of 3,480 datasets of *Mycobacterium tuberculosis* obtained from27, and genotyping 68,269 SNPs. Searching all of these datasets for these SNPs took less than 90 minutes on a single CPU core which translates to a genotyping rate of more than 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 samtools28 (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: SRST229. 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 BIGSI

We benchmarked the empirical scaling properties of BIGSI against the Sequence Bloom Tree (SBT)13 and the Split Sequence Bloom Tree15 (SSBT) using a dataset of 10,000 random microbial sequence datasets from the ENA. We compared build and query times, and peak storage requirements on 21 subsets ranging from 100 to 10,000 samples. For each subset, we built a BIGSI, an SBT and an SSBT using two different parameter settings: one optimized for speed (recommended by the authors in Lemma 1, and subsequent text, of Solomon et al13 and the SBT/SSBT user manual) (SBT-fast/SSBT-fast), and another optimized for compression (SBT-small/SSBT-small) (Online methods). For database sizes of more than 2,000 samples, peak storage requirements of SBT-fast and SSBT-fast exceeded our available disk space (1 Tb; by comparison the total storage required for the input data was 403 Gb; Online methods).

We queried the resulting indexes for 2,157 antimicrobial resistance genes with a mean length of 937 bp and total query length 2,021,655 bp. 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 = 1,000 database size). With the inexact search, concordance was >99% with the differences likely due to differences between methods in the construction of the underlying bloom filters. Inexact (T = 40%) query times versus peak storage requirements for the various methods are shown 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 trade storage requirements for performance (we quantify performance below). SSBT-fast and SBT-fast have query time comparable to or better than BIGSI but require orders of magnitude more storage to build the uncompressed tree. SBT-small and SSBT-small have almost equivalent storage as BIGSI, but slower query performance, 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. With 2,000 samples in the index, the SBT-fast query was only slightly slower than BIGSI's (292s vs 274s) but required 132X (878GB/6.6GB) more storage during construction, and 9X (57.5GB/6.6GB) more storage after compression. Further details in Online Methods. Similar

results were found for exact queries (T = 100%, Supplementary Figure 4); exact match searches with BIGSI were up to 2-3X faster than inexact queries.

To ensure our SBT comparison was fair, we also benchmarked BIGSI and SBT on the same human RNA-seq dataset used in 13,15,30, 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), and compressed index size. BIGSI has faster query time than SBT in smaller space for these datasets (360 s and 144 Gb for BIGSI, and 2221 s and 200 Gb for SBT (T = 70%); details in Online methods and full results, including for exact search, in Supplementary Table 2).

Finally, we simulated the scaling of storage requirements required to build SBT-fast, and BIGSI for dataset of up to 1 million genomes of two types: genomes with high proportions of k-mer sharing (e.g. human), and genomes with lower proportions of k-mer sharing (e.g. bacteria) (Supplementary Figure 5). BIGSI scales linearly with number of datasets, performing identically in both cases. For the low k-mer-sharing simulation an unsaturated (i.e. fast) SBT/SSBT would require 4 orders of magnitude more storage than BIGSI to construct (tens of Pb rather than 3 Tb).

An index of all bacterial and viral sequence in the ENA

We downloaded the entire set of 469,654 bacterial, viral and parasitic WGS datasets in the ENA as of December 2016. After a step to exclude eukaryotic genomes on the basis of size (Online Methods), we were left with 447,833 datasets, from which we created a BIGSI index. This index, named the ‘all-microbial index’ required 1.5 TB of storage, <1% of the original data size (170 TB) and contained more than 60 billion unique k-mers. Data download took 6 weeks, and we built bloom filters ‘on the fly’. Combining the complete set of bloom filters took approximately 2 days (Online Methods). We estimate that the intermediate storage required to build an SBT-fast/SSBT-fast of the same data with recommended bloom filter size equal to the number of unique k-mers in the collection, would be >6.7PB.

We used our ‘all-microbial index’ for several analyses. First, we estimated the species and abundances present in each dataset that within the all-microbial-index using the Bayesian abundance estimator Bracken31, which parses output from the read-classifier Kraken32 (Online Methods). Using this method, we found that more than 90% of the input datasets were from only 20 genera, and 65% of the input datasets were from the 5 most common bacterial genera (*Salmonella*, *Streptococcus*, *Staphylococcus*, *Escherichia* and *Mycobacterium*); counts for the most prevalent bacterial genera in the all-microbial index are shown in Supplementary Figure 6.

Ultrafast gene search using the all-microbial index

We searched for exact matches to the mobilized colistin resistance genes *MCR-1/2/3*33–37 in the all-microbial-index. Searching for all 3 genes in 10x more genomes than has been previously reported took just 1.73 s seconds in total. We did not detect *MCR-2*. *MCR-1* was present in 169 datasets from 3 species (*Escherichia coli*, *Salmonella enterica*, *Enterobacter*

aerogenes) and *MCR-3* was present in 34 datasets from 3 species (*E. coli*, *S. enterica*, *Klebsiella pneumoniae*) (Supplementary Data 1). Our all microbial index therefore enables almost instant detection of drug resistance genes, or indeed any other gene studied, across the global corpus.

Host-range of plasmids in the all-microbial index

We downloaded a set of 2,827 plasmids from the ENA (Online Methods, Supplementary Data 2) and ran an inexact (T = 40%) search for them in the all-microbial-index. We restricted analysis to hits with T > 90%. The total length of query sequences was 227 Mbp, and the query took 2,120 CPU hours (11 days) 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 all datasets containing evidence of more than one genus at abundance above 0.1%. Only 41% (184,652) of datasets and 62% of search hits passed this filter.

In the filtered output we identified plasmids shared by closely related genera, for example *Escherichia*, *Shigella* and *Salmonella* or *Enterococcus*, *Streptococcus* and *Staphylococcus*. We found 37 plasmids present in at least five datasets from at least two genera (Figure 4, Supplementary Data 2, 3); and 5 plasmids that were present in multiple orders and families. The plasmid pETHIS-1 (GenBank identifier: 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 detect common plasmids. Of more biological interest, the Tn916 conjugative transposon encoding tetracycline resistance that was first found in *Enterococcus faecium* and is known to have broad host range38 (GenBank identifier: U09422) was found in Streptococcus (n = 3,951), Staphylococcus (n = 1,212), Enterococcus (n = 43), Clostridioides (n = 29), Listeria (n = 19) and Erysipelothrix (n = 11).

Sampling biases in the ENA preclude inference about plasmid distribution, but they do allow us to ask if plasmids bearing antibiotic resistance genes are more widely distributed than those plasmids that do not have antibiotic resistance genes. We defined “phylogenetic spread” of a plasmid as the median of the pairwise distances along a large subunit rRNA tree (incorporating branch lengths) between all pairs of genera in which the plasmid is detected. We tested whether plasmids harboring at least 3 antibiotic resistance genes are more widely distributed across this phylogeny than plasmids with no antibiotic resistance genes by comparing the 95% quantile of these two distributions, and found that they are (Figure 5). We tested for significance using a permutation test (p = 0.0024, 1 million replicates) (Supplementary Figure 7). Since the sampling of the ENA data is biased towards the *Enterobacteriaceae*, we would restrict this conclusion to that family.

The distribution of different versions of the machinery for conjugation of DNA between bacteria has previously been analysed in 1,124 genomes39 using sensitive, but slow, protein profile searches for relaxase (MOB) and type 4 secretion system (T4SS) genes. We undertook a similar analysis of the whole ENA, by using an exact match search of the all-microbial-index for nucleotide alleles corresponding to previously identified MOB and

T4SS “types” (alleles defined in amino acid space). We applied the same contamination filter mentioned above (restricting to 184,652 datasets), and required exact matches to at least one MOB and one T4SS be present to be considered a putative conjugative system. 36,030 datasets met these criteria, giving an estimate of 19.5% of bacteria in the ENA containing a conjugative system, which is consistent with the previous estimate of 18% in Guglielmini et al39. This proportion varied by phylum (minimum 0.5% in *Spirochaetes* and maximum 31.7% in *Firmicutes,* full data in Supplementary Table 3, and MOB type distribution in Supplementary Figure 8). These data could be mined further to explore the potential spread of antibiotic resistance genes. For example, by analysing datasets shown to contain MOB_T we observed genetic flux between *Staphylococcus* and *Streptococcus*, but not between either of them and *Salmonella*. We also found MOB_Q in *Salmonella* and *Streptococcus* but not *Staphylococcus*, indicating a different probability of cross-genus (and cross-phylum) transfer by conjugation (Supplementary Data 4).

Antibiotic resistance genes in the ENA

To understand whether antibiotic resistance gene prevalence in the ENA has changed over time, we downloaded the set of 2,157 sequences associated with antibiotic resistance from the the Comprehensive Antibiotic Resistance Database (CARD) database (v1.1.7)40 searched for this set of genes in the all-microbial-index with thresholds of 100% and 70% (full results for both searches are in Supplementary Data 5). An exact search for a single gene took an average of 1.1s and returned an average of 438 hits, whereas the corresponding figures for an inexact search (T = 70%) were 34.4s and 5,320 hits. We show the prevalence over time in 3 genera, and in total, in Figure 6. Focusing only on *Staphylococcus*, we found that the proportion of datasets containing *mecA* gene (methicillin resistance), fell from 70% in 2013 to 40% in 2016, and that prevalence of *tet* and *aac* genes also decreased (Figure 6b). For *Klebsiella* however, the prevalence of all resistance genes that we screened for in the all-microbial dataset increased over time (Figure 6c).

In *M. tuberculosis,* antibiotic resistance mainly arises through amino acid mutations in specific genes 1,41. We genotyped the all-microbial-index, which contains 30,226 *M. tuberculosis* genome datasets, for the 206 resistance mutations enumerated by Walker *et al.* 27. This exercise took only 103 minutes on a single-core, which is approximately 10,000x faster than typing each dataset individually with the fast resistance prediction software *Mykrobe predictor*1. Our results indicate an increase in prevalence of MDR-TB in the ENA since 2011 (Figure 6d). It should be noted that one cannot infer anything about global prevalence from this, as the contents of the ENA are subject to sampling biases (that is, are dependent on specific research studies rather than containing an unbiased sampling of isolates from tuberculosis patients). Indeed, the latest WHO estimates for 201642, put MDR prevalence at 6.6% as compared with the 18.9% of datasets deposited in the ENA/SRA in 2016 that we detected (Figure 6d).

Discussion

Hundreds of thousands of bacterial and viral samples are sequenced and shared every year, and this number is growing exponentially. A scalable online sequence search facility that

could access all deposited sequence data would unlock the archives for fundamental research, clinical microbiology and public health, in much the same way that web-search allows people to narrow down to web-pages of interest. In particular, there is a pressing need for a global infrastructure for surveillance and management of infectious diseases43,44. Until now, this has not been possible. We have developed a data structure (BIGSI) to meet this need, and have indexed the entire bacterial and viral WGS content of the global DNA archive. BIGSI allows tracking of several key entities: genes, SNPs, plasmids, MLST types, or clusters defined by SNPs. Importantly, new datasets can be readily and quickly added to our index, allowing it to grow as new datasets are sequenced. Our indexing method works for both raw data and assembled genomes, so it is poised for a future in which finished reference genomes are routine.

BIGSI was designed with SNP or indel genotyping and allele search queries in mind. It allows a user to identify datasets worthy of in-depth study, as well as enabling global monitoring of specific alleles. Nevertheless, BIGSI has limitations, which it shares with previous k-mer tools such as the SBT; we give the main ones here, but also see Online Methods. 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 cannot be resolved. Also BIGSI does not store coverage information which rules out queries in which copy number is important. One example of such a query is detecting azithromycin resistance in *Neisseria gonorrhoeae*, in which the level of resistance is mediated by the number of rRNA genes containing a particular SNP45. BIGSI only supports nucleotide k-mers although we would note that an extension to amino acid search would be straightforward, because bloom filters are agnostic to what they are storing. Finally, BIGSI is optimized for very diverse datasets, in which the combined unique k-mer count is much higher than that of any individual sample e.g. the entire microbial ENA/SRA. Where samples are less diverse (eg. human), we expect SSBT or Mantis to be a better choice, as they take advantage of compression from sample similarity.

In the future we envisage that sequencing data volumes will continue to grow and the user-base that wants to access and exploit those data will increase to include clinical and public health practitioners. There will be a pressing need for quick and accurate sequence searches across the global corpus of microbial sequences. MASH combined with both nucleotide and protein BIGSI could be a powerful combined approach to achieve this goal. We are now investigating implementing our BIGSI as a live service at the EMBL-EBI, to be updated as data is added to the ENA. These and complementary approaches will put shared DNA resources at everyone’s fingertips.

Online Methods

Code availability

An open source implementation of BIGSI can be found at https://github.com/phelimb/BIGSI. BIGSI v0.3.0 supports disk-based indexing via Berkeley-DB, or rocksDB, and distributed in-memory (via redis (https://redis.io)) key-value stores and can be extended to any key-value store. The benchmarking uses the rocksDB key-value store, and v0.2.0 of BIGSI, and the all-microbial index uses Berkeley-DB and BIGSI version v0.1.7.

Europe PMC Funders Author Manuscripts

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 allowable 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 formula48

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

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

which becomes:

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

$$\eta = -\frac{ln(q_{max}/N)}{L_{min} ln(2)}.$$

For example, given

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

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

Europe PMC Funders Author Manuscripts

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 $K_{max} = 10^7$, $N = 10^6$).

K-mer identity and scoring

If, for example, the k-mer size is 31, each SNP difference between the query and the nearest sequence in the index causes a window of 31 absent k-mers. Therefore k-mer identity drops more rapidly than sequence identity; e.g for k=31, matches with sequence identity above 80% can be detected (Supplementary Figure 2). Therefore, BIGSI most naturally enables exact or close match searches, or situations where combinatorial searching is feasible, such as querying all sequences 2 SNP different from a given allele, or all single amino acid changes in a gene. Datasets containing matches for a more divergent allele can be sought by searching for subsequences (seeds).

Although BIGSI does not carry out an alignment, an approximation to a megaBLAST alignment score can be inferred from the presence/absence pattern of k-mers in the query. To approximate a megaBLAST score, 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. This approximation deteriorates as k-mer identity drops (see Supplementary Figure 2). Nevertheless, we show the strong correlation (r=0.998) between MegaBLAST score and BIGSI score for 100 *E. coli* AMR genes using a BIGSI of RefSeq-bacteria (release 81) in Supplementary Figure 3.

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 ($m = 2.5 \times 10^7; \eta = 3$). 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 al13, 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 $m = 2.5 \times 10^7; \eta = 3$ (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 3000 datasets, the practical limit we set, and as a result SBT-fast was not built for increments above 2,000 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.

As mentioned above, SBT-fast and SSBT-fast construction failed for databases > 2,000 due to excessive disk requirements, but we were able to calculate a lower bound for the peak disk-use for database sizes > 2,000; query times were extrapolated linearly (these points are shown as triangles on Figure 3; details of lower bound below).

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. It would require ~11TB of storage to build SBT-fast, 350X more space than is required by BIGSI. By contrast, querying SBT-small took 65X longer and SSBT-small 448X longer than a BIGSI of the same database size (N=10,000). 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 dataset, constructing unsaturated (i.e. fast) SBTs and SSBTs for larger numbers of datasets is prohibitive in terms of storage requirements; we address scaling further below. Query times for SBT/SSBT-slow were unacceptable for this benchmark, and so we excluded them as candidates for storing the ENA/SRA, which is 50X bigger than this benchmark dataset.

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>2000 datasets, we used this lower bound to plot (suitably labelled) extrapolated datapoints.

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 in13. 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. Since the SBT was pre-built by the authors, and therefore compressed, we used a RocksDB back-end store for BIGSI, which used Snappy (https://google.github.io/snappy/) for bitslice compression. This gave us a compressed BIGSI index, allowing us to compare compressed index size.

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 (e.g. metagenomic data) is to build 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. Queries can then be sent to all the indexes in parallel and then results combined.

Genotyping accuracy measurement on TB

Conservative SNP calls were made using Cortex17 (independent workflow, k=31) on 3480 *Mycobacterium tuberculosis* datasets from Walker et al27. 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 al27, 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 mccortex19 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 ($m = 2.5 \times 10^7; \eta = 3$, $K_{max} = 10^7$) 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 Kraken32 and Bracken31. 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

Europe PMC Funders Author Manuscripts

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 39 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)40. 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 27 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 in27. 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.

Trade-offs and limitations in BIGSI

BIGSI can be considered a coloured de Bruijn graph, but we would note that it is optimized for search (genotyping), not traversal of the graph. Although one could search for remote homologs using exact matching of short seeds followed by graph traversal, this would require additional software development to handle false positive edges. BIGSI is able to achieve speed, compression and accuracy by using queries that are longer than the indexing k-mer size (here 31) – the relatively highly false-positive rate at a single k-mer is dropped exponentially with each extra unique k-mer in the query. This does however mean that short or low-complexity queries should only be used in circumstances where a higher error rate is acceptable (e.g. we require >=63bp queries and >=21 unique kmers for the all-microbial-index).

Data availability

All of the underlying genomic data for this study is publicly available at the ENA.

Supplementary Tables 1-3 are available in a single supplementary document.

Supplementary Figures 1-8 are available in a single supplementary document.

Supplementary Data files 1-7 can be found here: ftp.ebi.ac.uk/pub/software/bigsi/nat_biotech_2018/supp. Contents are : data 1-MCR search results, data 2 – plasmid search results, data 3 – counts of 5 specific plasmids across genera, data 4 – counts of MOB types across genera, data 5 – CARD antibiotic resistance gene search results (T=70%), data 6 – benchmarking results, data 7 – Bracken taxonomic results, data 8 – MOB type definition fasta, data 9 - MOB and T4SS search results (T=100%).

The all-microbial-index itself is available here: ftp.ebi.ac.uk/pub/software/bigsi/nat_biotech_2018/all-microbial-index/. In order to facilitate reproducibility for others without having to download and process 170Tb of raw data, we make the 26Tb of cleaned de Bruijn (binary) graph files for the entire all-microbial-index snapshot of the ENA available at ftp.ebi.ac.uk/pub/software/bigsi/nat_biotech_2018/ctx. An archive of our code, plus the underlying data for our figures, can be found here: at ftp.ebi.ac.uk/pub/software/bigsi/nat_biotech_2018/code and at ftp.ebi.ac.uk/pub/software/bigsi/nat_biotech_2018/data_underlying_figures

We have also 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 BIGSI v0.1.7 (using berkeleyDB) and is hosted by CLIMB (http://www.climb.ac.uk/) on a 3Tb RAM server.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

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. GM was funded by Wellcome Trust grant 100956/Z/13/Z. ER was funded by the ANR MAGISBAC grant number ANR-14-CE10-0007-02. PB was funded by Wellcome Trust Studentship H5RZCO00.

References

1. Bradley P, et al. Rapid antibiotic-resistance predictions from genome sequence data for Staphylococcus aureus and Mycobacterium tuberculosis. Nat Commun. 2015; 6doi: 10.1038/ncomms10063
2. Brown AC, et al. Rapid Whole-Genome Sequencing of Mycobacterium tuberculosis Isolates Directly from Clinical Samples. J Clin Microbiol. 2015; 53:2230–2237. DOI: 10.1128/JCM.00486-15 [PubMed: 25972414]
3. Quick J, et al. Real-time, portable genome sequencing for Ebola surveillance. Nature. 2016; 530:228–232. DOI: 10.1038/nature16996 [PubMed: 26840485]
4. Schmidt K, et al. Identification of bacterial pathogens and antimicrobial resistance directly from clinical urines by nanopore-based metagenomic sequencing. J Antimicrob Chemother. 2017; 72:104–114. DOI: 10.1093/jac/dkw397 [PubMed: 27667325]
5. Votintseva A, et al. Same-Day Diagnostic and Surveillance Data for Tuberculosis via Whole-Genome Sequencing of Direct Respiratory Samples. J Clin Microbiol. 2017; 55:1285–1298. DOI: 10.1128/JCM.02483-16 [PubMed: 28275074]
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. 2017; 55:1871–1882. DOI: 10.1128/JCM.00298-17 [PubMed: 28381603]
7. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990; 215:403–410. DOI: 10.1016/S0022-2836(05)80360-2 [PubMed: 2231712]
8. Kent WJ. BLAT--the BLAST-like alignment tool. Genome Res. 2002; 12:656–664. DOI: 10.1101/gr.229202 [PubMed: 11932250]
9. Morgulis A, et al. Database indexing for production MegaBLAST searches. Bioinformatics. 2008; 24:1757–1764. DOI: 10.1093/bioinformatics/btn322 [PubMed: 18567917]
10. Arredondo-Alonso AWR, Schaik WV, Schurch C. On the (im)possibility of reconstructing plasmids from whole-genome short-read sequencing data. Microbial Genomics. 2017; doi: 10.1099/mgen.0.000128
11. Bradnam KR, et al. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. Gigascience. 2013; 2:10.doi: 10.1186/2047-217X-2-10 [PubMed: 23870653]
12. Earl D, et al. Assemblathon 1: a competitive assessment of de novo short read assembly methods. Genome Res. 2011; 21:2224–2241. DOI: 10.1101/gr.126599.111 [PubMed: 21926179]
13. Solomon B, Kingsford C. Fast search of thousands of short-read sequencing experiments. Nat Biotechnol. 2016; 34:300–302. DOI: 10.1038/nbt.3442 [PubMed: 26854477]
14. Pandey P, et al. Mantis: A Fast, Small, and Exact Large-Scale Sequence-Search Index. Cell Syst. 2018; 7:201–207 e204. DOI: 10.1016/j.cels.2018.05.021 [PubMed: 29936185]

15. Solomon, B; Kingsford, C. International Conference on Research in Computational Molecular Biology; Springer; 257–271.
16. Sun, C; Harris, R; Chikhi, R; Medvedev, P. International Conference on Research in Computational Molecular Biology; Springer; 272–286.
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. 2012; 44:226–232. DOI: 10.1038/ng.1028 [PubMed: 22231483]
18. Muggli M, et al. Succinct colored de Bruijn graphs. Bioinformatics. 2017; 33:3181–3187. DOI: 10.1093/bioinformatics/btx067 [PubMed: 28200001]
19. Turner I, Garimella K, Iqbal Z, McVean G. Integrating long-range connectivity information into de Bruijn graphs. Biorxiv. 2017; doi: 10.1101/147777
20. Almodaresi, F; Pandey, P; Patro, R. 17th International Workshop on Algorithms in Bioinformatics (WABI 2017); Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik;
21. Wong, HKT; L, H; Olken, F; Rotem, D; Wong, L. In: Pirotte, Alain; Vassiliou, Yannis, editors. Proceedings of the 11th international conference on Very Large Data Bases; Stockholm, Sweden: 1985. 448–457.
22. Shepherd MA, P W, Chu CK. A Fixed-Size Bloom Filter for Searching Textual Documents. The Computer Journal. 1989; 32doi: 10.1093/comjnl/32.3.212
23. Zobel J, Moffat A, Ramamohanarao K. Inverted Files Versus Signature Files For Text Indexing. ACM Trans Database Syst. 1998; 23:453–490. DOI: 10.1145/296854.277632
24. Goodwin, B; H, M; Luu, D; Clemmer, A; Curmei, M; Elnikety, S; He, Y. 40th International ACM SIGIR Conference on Research and Development in Information Retrieval; ACM;
25. BH B. Space/time trade-offs in hash coding with allowable errors. Commun ACM. 1970; 13:422–426. DOI: 10.1145/362686.362692
26. Pell J, et al. Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. Proc Natl Acad Sci U S A. 2012; 109:13272–13277. DOI: 10.1073/pnas.1121464109 [PubMed: 22847406]
27. Walker TM, et al. Whole-genome sequencing for prediction of Mycobacterium tuberculosis drug susceptibility and resistance: a retrospective cohort study. Lancet Infect Dis. 2015; 15:1193–1202. DOI: 10.1016/S1473-3099(15)00062-6 [PubMed: 26116186]
28. Li H, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009; 25:2078–2079. DOI: 10.1093/bioinformatics/btp352 [PubMed: 19505943]
29. Inouye M, et al. SRST2: Rapid genomic surveillance for public health and hospital microbiology labs. Genome Med. 2014; 6:90.doi: 10.1186/s13073-014-0090-6 [PubMed: 25422674]
30. Pandey P, A F, Bender MA, Ferdman M, Johnson R, Patro R. Mantis: A Fast, Small, and Exact Large-Scale Sequence Search Index. Biorxiv. 2017; doi: 10.1101/217372
31. Lu J, B F, Thielen P, Salzberg SL. Bracken: estimating species abundance in metagenomics data. PeerJ Computer Science. 2017; 3doi: 10.7717/peerj-cs.104
32. Wood DE, Salzberg SL. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. 2014; 15:R46.doi: 10.1186/gb-2014-15-3-r46 [PubMed: 24580807]
33. Hu Y, Liu F, Lin IY, Gao GF, Zhu B. Dissemination of the mcr-1 colistin resistance gene. Lancet Infect Dis. 2016; 16:146–147. DOI: 10.1016/S1473-3099(15)00533-2 [PubMed: 26711359]
34. 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. 2017; 61doi: 10.1128/AAC.02632-16
35. 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
36. Xavier BB, et al. Identification of a novel plasmid-mediated colistin-resistance gene, mcr-2, in Escherichia coli, Belgium, June 2016. Euro Surveill. 2016; 21doi: 10.2807/1560-7917.ES.2016.21.27.30280
37. Yin W, et al. Novel Plasmid-Mediated Colistin Resistance Gene mcr-3 in Escherichia coli. MBio. 2017; 8doi: 10.1128/mBio.00543-17

38. 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. 2013
39. Guglielmini J, Quintais L, Garcillan-Barcia MP, de la Cruz F, Rocha EP. The repertoire of ICE in prokaryotes underscores the unity, diversity, and ubiquity of conjugation. PLoS Genet. 2011; 7:e1002222.doi: 10.1371/journal.pgen.1002222 [PubMed: 21876676]
40. Jia B, et al. CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. Nucleic Acids Res. 2017; 45:D566–D573. DOI: 10.1093/nar/gkw1004 [PubMed: 27789705]
41. Eldholm V, Balloux F. Antimicrobial Resistance in Mycobacterium tuberculosis: The Odd One Out. Trends Microbiol. 2016; 24:637–648. DOI: 10.1016/j.tim.2016.03.007 [PubMed: 27068531]
42. Organisation WH. Global Tuberculosis Report 2017. 2017.
43. Gardy J, Loman N. Towards a genomics-informed, real-time, global pathogen surveillance system. Nature Reviews Genetics. 2017
44. Schatz MC, Phillippy AM. The rise of a digital immune system. Gigascience. 2012; :1–4. DOI: 10.1186/2047-217X-1-4 [PubMed: 23587310]
45. Eyre DW, et al. WGS to predict antibiotic MICs for Neisseria gonorrhoeae. J Antimicrob Chemother. 2017; 72:1937–1947. DOI: 10.1093/jac/dkx067 [PubMed: 28333355]
46. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics. 2009; 25:1754–1760. DOI: 10.1093/bioinformatics/btp324 [PubMed: 19451168]
47. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. 2009; 10:R25.doi: 10.1186/gb-2009-10-3-r25 [PubMed: 19261174]
48. Broder A, Mitzenmacher M. Network Applications of Bloom Filters: A Survey. Internet Mathematics. 2004; 1:485–509. DOI: 10.1080/15427951.2004.10129096

Europe PMC Funders Author Manuscripts

Editors summary

The global set of bacterial and viral sequences can be rapidly searched using a data structure inspired by web-search algorithms.

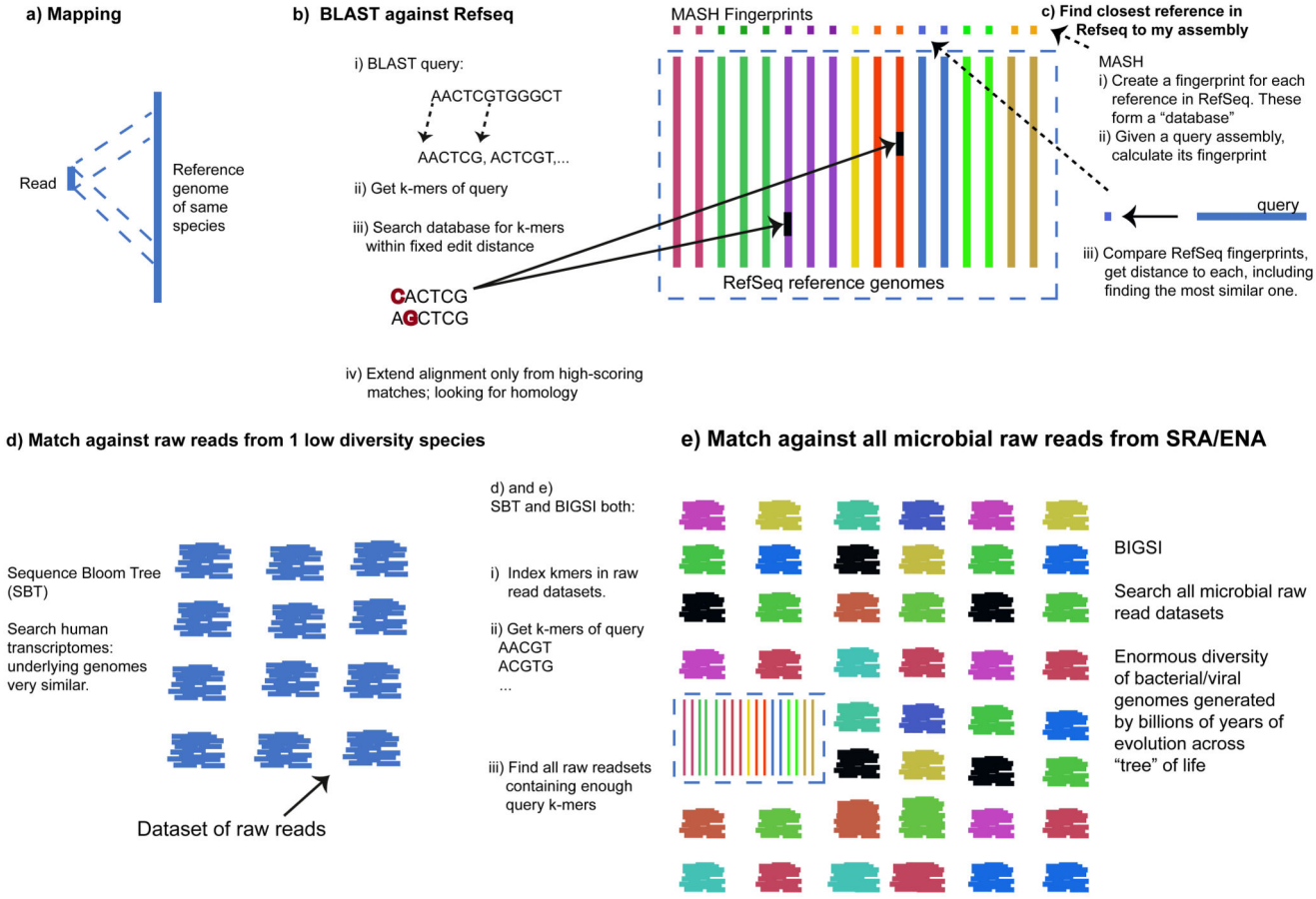

Figure 1. Sequence matching methods.

a) Mapping of sequence reads to a reference genome from the same species, assuming relatively low divergence; requirement to map millions of reads in acceptable time and return an alignment and mapping score. Common tools: bwa46 and bowtie47. b) BLAST7 compares a query string with a database of reference genomes (in the figure we show RefSeq genomes in a dotted box) covering a massive phylogenetic range. BLAST takes k-mers from the query, and for each k-mer it creates a "neighbourhood" of k-mers within a fixed edit distance (edits are shown in red, b(iii)), and searches for these in the reference genome database. Alignment is only done by extending from these hits. Blast can be applied to nucleotide and protein searches and can find close and remote homology matches. c) MASH stores a tiny fingerprint of each reference in the database (in this case RefSeq). Querying with an assembly, the fingerprint of the assembly is compared with that of RefSeq to find the closest reference. d) Sequence Bloom Tree13 was the first scalable method to search through raw unassembled readsets (unassembled readsets are shown as "piles" of reads (short lines), all in same colour to signify same species), by indexing the k-mers in the data and then compressing the index. Designed for human data, SBT can be applied to find which RNA-seq datasets contain a given transcript. e) BIGSI can search the complete set of raw sequence data for bacteria and viruses. RefSeq is shown in a dotted box amongst unassembled readsets; different colours to signify the massive range of species and phyla.

Europe PMC Funders Author Manuscripts

The different input data for SBT and BIGSI mean that these methods have different speed and compression trade-offs.

Europe PMC Funders Author Manuscripts

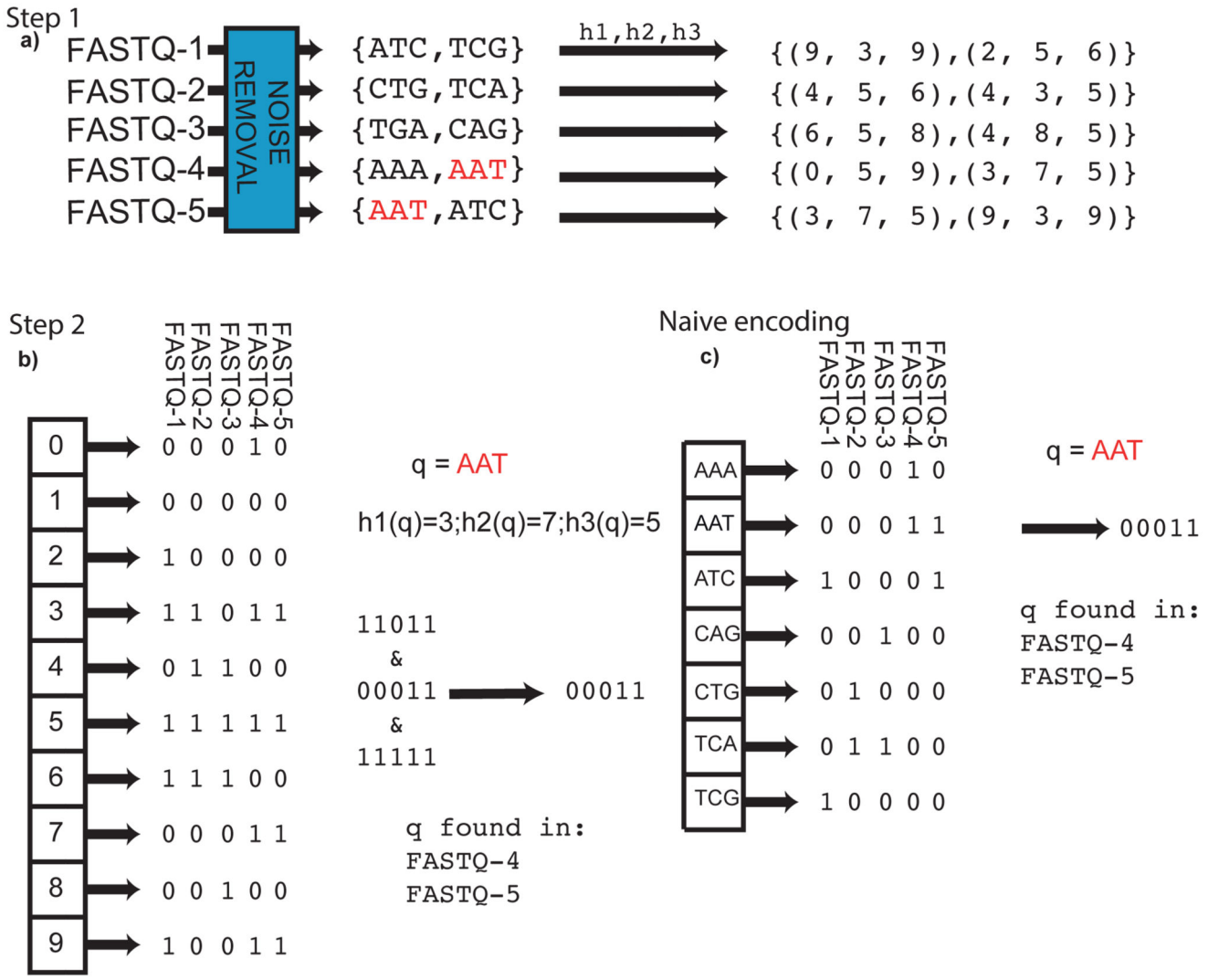

Figure 2. BIGSI encoding.

a) In step 1, each input dataset (raw sequence data in 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 Online Methods). A fixed set of η hash functions (h_1, h_2, …) is applied to each k-mer (η=3 in this figure), giving a tuple of positions which are all set to 1 in a bit-vector (a Bloom Filter). b) In 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 adding a new column. c) Naïve encoding is shown to contrast with the BIGSI approach. 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).

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

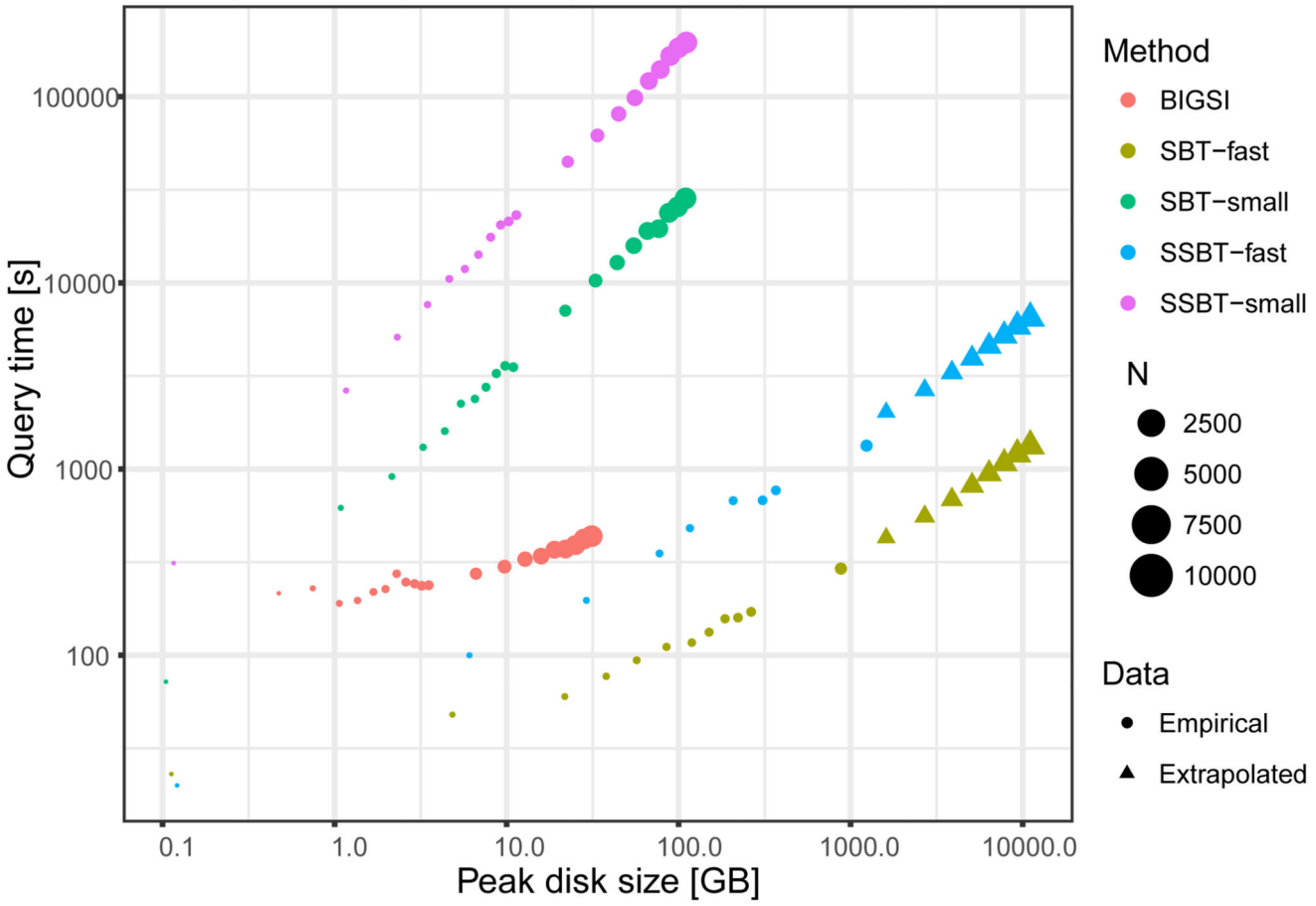

Figure 3. Speed and space trade-offs as index grows.

Benchmarking of BIGSI against fast and small versions of each of SBT and SSBT, using a set of 2,157 antimicrobial resistance genes as a query dataset. We performed inexact search (T = 40%) and show query speed vs. peak disk size when searching databases of sizes from 10–10,000 microbial datasets. Both axes are on a log scale; the diameter of a dot represents the number of datasets indexed. 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 database sizes greater than 2000, 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 (Online Methods).

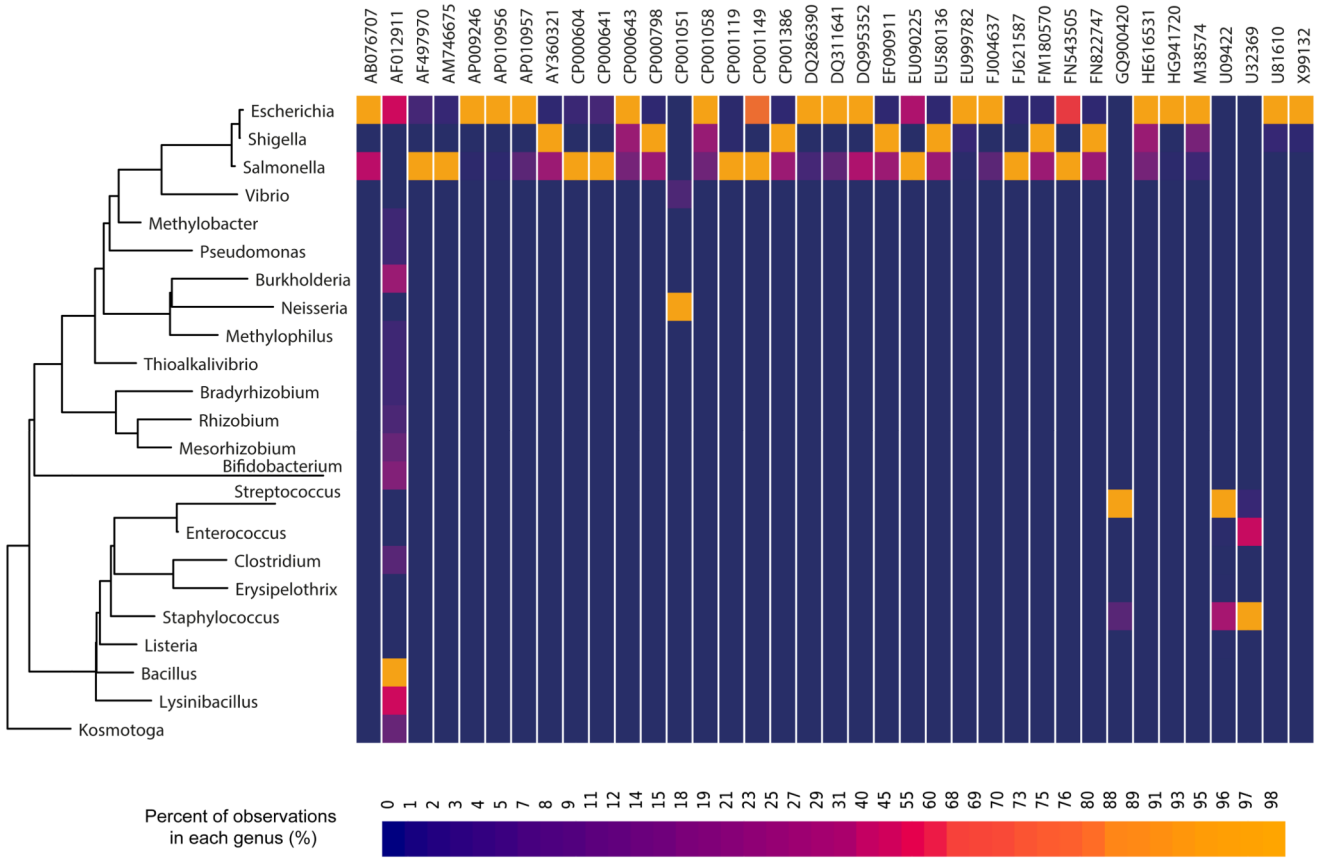

Figure 4. Phylogenetic distribution of plasmid sequences.

Plasmid sequences (37) that are found at least 5 times in more than one genus in the all-microbial-index are shown. The heatmap shows the frequency of each plasmid in 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 *Escherichia*, *Salmonella* and *Shigella* is consistent with known promiscuity within *Enterobacteriaceae*.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

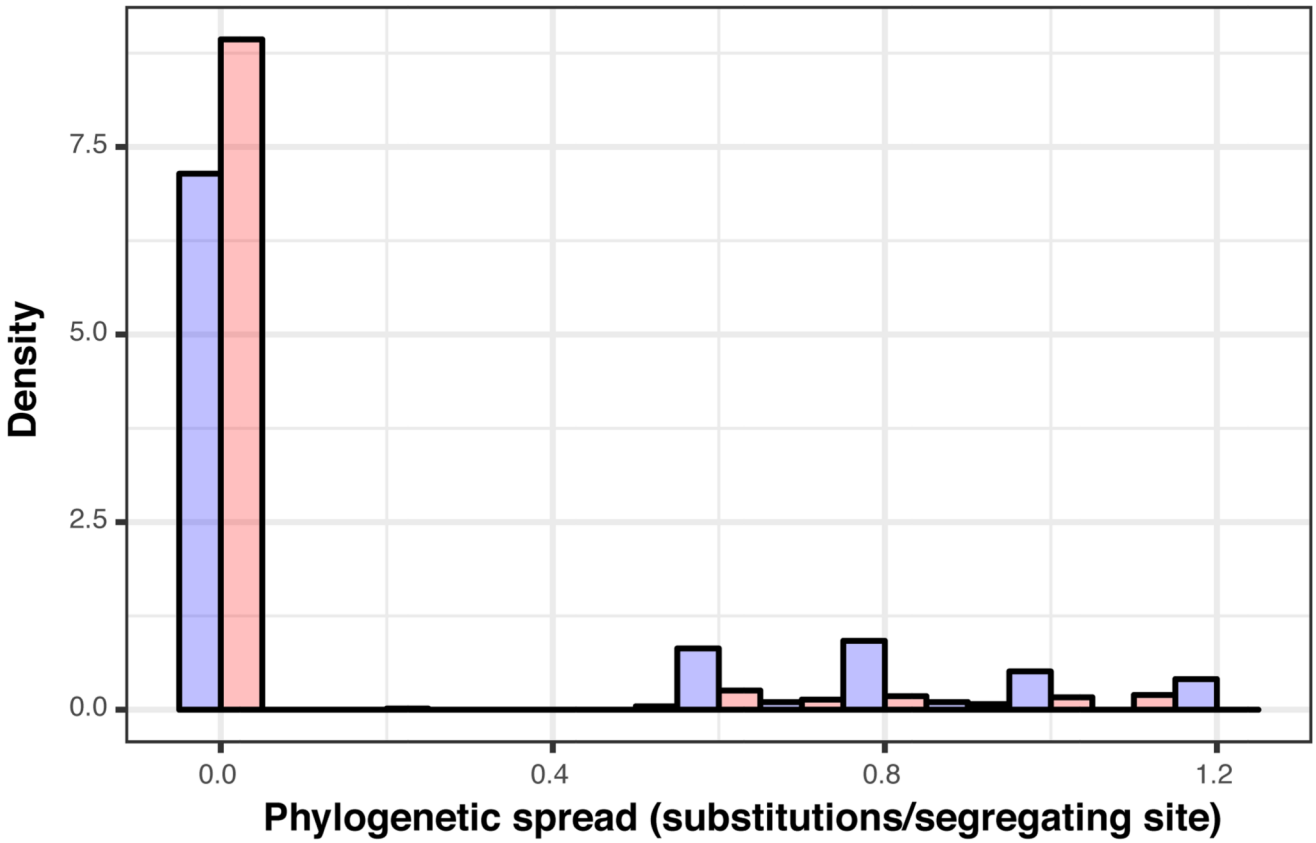

Figure 5. Plasmid spread and antibiotic resistance genes.

A 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) is shown. Histograms are normalized to allow comparison (probability densities). The 95% quantiles of the distributions are 1.11 (no resistance genes) and 1.99 (>=3 genes). 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.

Europe PMC Funders Author Manuscripts

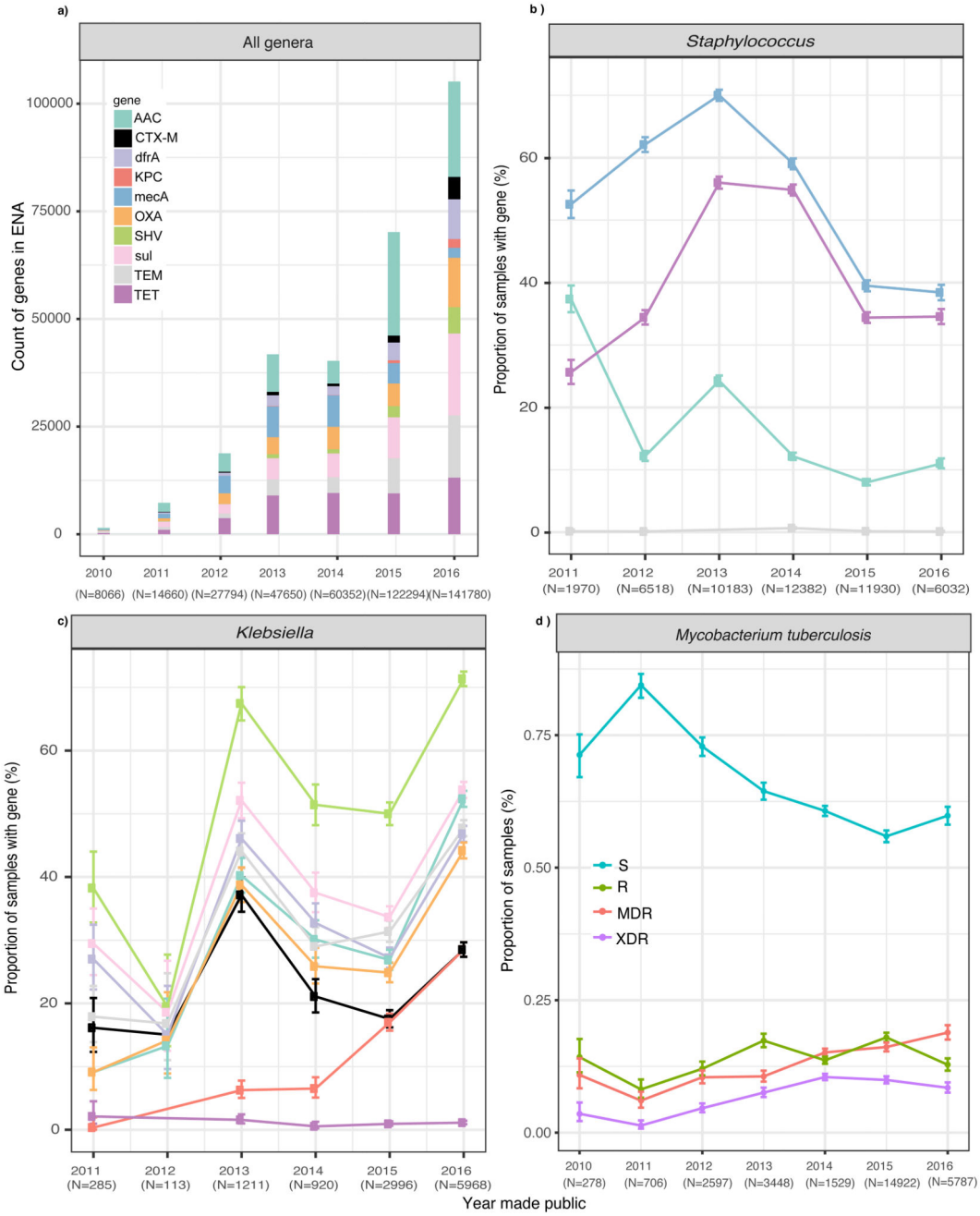

Figure 6. Antibiotic resistance gene prevalence in ENA over time.

a) Counts of samples in the all-microbial index containing a range of antibiotic resistance 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. c) Year-by-year frequency in *Klebsiella* of various antibiotic resistance 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 from27, 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. Error bars around the estimated frequency (mean) in (b–d) show the 95% confidence interval calculated using the Wilson binomial confidence test.

Europe PMC Funders Author Manuscripts