Genomic Epidemiology with Mixed Samples

Tommi Mäklin[1,*], Teemu Kallonen[2,3], Jarno Alanko[4], Veli Mäkinen[4], Jukka Corander[1,2,3], and Antti Honkela[4,*]

1 Helsinki Institute for Information Technology HIIT, Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland
2 Department of Biostatistics, University of Oslo, Oslo, Norway
3 Wellcome Sanger Institute, Hinxton, Cambridgeshire, UK
4 Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Helsinki, Finland
* Corresponding author, tommi.maklin@helsinki.fi, antti.honkela@helsinki.fi

Abstract

Genomic epidemiology is an established tool for investigation of outbreaks of infectious diseases and wider public health applications. It traces transmission of pathogens based on whole-genome sequencing of colony picks from culture plates enriching the target organism(s). In this article, we introduce the mGEMS pipeline for performing genomic epidemiology directly with plate sweeps representing mixed samples of the target pathogen in a culture plate, skipping the colony pick step entirely. By requiring only a single culturing and library preparation step per analyzed sample, we address several key issues in the current approach relating to its cost, practical application and sensitivity. Our pipeline significantly improves upon the state-of-the-art in analysing mixed short-read sequencing data from bacteria, reaching accuracy levels in downstream analyses closely resembling colony pick sequencing data that allow reliable SNP calling and subsequent phylogenetic analyses. The fundamental novel parts enabling these analyses are the mGEMS read binner for probabilistic assignments of sequencing reads and the high-throughput exact pseudoaligner Themisto. In conjunction with recent advances in probabilistic modelling of mixed bacterial samples and genome assembly techniques, these tools form the mGEMS pipeline. We demonstrate the effectiveness of our approach using closely related samples in a nosocomial setting for the three major pathogens *Enterococcus faecalis*, *Escherichia coli* and *Staphylococcus aureus*. Our results lend firm support to more widespread consideration of genomic epidemiology with mixed infection samples.

Keywords

genomic epidemiology, strain identification, plate sweeps, probabilistic modelling, pseudoalignment

Introduction

Public health epidemiology for bacterial infections has been transformed by the use of high-throughput sequencing data to analyze and identify the source of an outbreak and to trace circulating pathogenic strains based on routine surveillance (Deng et al. 2016; Tang et al. 2017; Van Goethem et al. 2019; Grad and Lipsitch 2014; Kwong et al. 2015). Standard genome-based epidemiological linking of cases requires accurate genome sequences for the pathogens derived from high coverage sequencing data for pure-colony isolates. The isolates are obtained by an enrichment and separation step in the form of a plate culture and subsequent colony picks based e.g. on morphology and colour. Typical workflow of genomic epidemiology may thus necessitate multiple colony picks per sample and the corresponding DNA library preparation and sequencing steps for each of them. Combined, these steps require a significant amount of laboratory effort and time, and lead to increased costs since the price of library preparation is becoming comparable to the cost of sequencing itself (Rossen et al. 2018). This can act as a barrier to more widespread genomic pathogen surveillance even in well-resourced public health laboratories, and prevent application of genomic epidemiology altogether in poorer settings.

Whole-genome shotgun metagenomics has been proposed as a solution for getting rid of the culturing step entirely. In this approach, sequencing is performed directly on the DNA extracted from the original sample and the resulting reads computationally binned or assembled. While tools capable of pangenome-based analyses (Scholz et al. 2016), metagenome assembly (Nurk et al. 2017; Li et al. 2016, 2015; Peng et al. 2012), or taxonomic binning (Sieber et al. 2018; Kang et al. 2019; Wu et al. 2016) from metagenomic short-read sequencing data have been developed, these methods typically require that the samples do not contain many closely related organisms. In particular the strain-variation within a species is assumed to be large enough not to be confused with sequencing errors or variation in the assembly graph (Breitwieser et al. 2019). When more complex strain-level diversity is present, benchmarking these tools shows reduced performance in both taxonomic binning and metagenomic assembly (Sczyrba et al. 2017; McIntyre et al. 2017; Vollmers et al. 2017; Meyer et al. 2018). In practice, natural strain-level variation is harbored ubiquitously in epidemiologically relevant samples (Greenblum et al. 2015; The Human Microbiome Project Consortium 2012; Ellegaard and Engel 2016) and it is reflected by the transmission events occurring between individuals and their environment (Stoesser et al. 2015). Although some sample types may be dominated by one or two strains (Truong et al. 2017), direct environmental sequencing may result in an overabundance of host DNA (Whelan et al. 2020; Ivy et al. 2018; Gu et al. 2019), or lack detection power for strains with low abundance in environments with high species diversity (Whelan et al. 2020; Quince et al. 2017; Vollmers et al. 2017). These challenges are overcome in genomic epidemiology by enriching the target species through the use of plate cultures. Since established protocols and growth media are available for most bacteria of clinical relevance (Lagier et al. 2015), enrichment provides an effective means to deplete the host DNA and increase the sequencing depth for target organisms when working with well-characterized species.

In this article, we introduce the mGEMS pipeline for performing genomic epidemiology with mixed cultures from samples that may harbor multiple closely related bacterial lineages. mGEMS requires only a single culturing and library preparation step per sample, which can significantly reduce the cost of performing genomic epidemiology in the standard public health setting and make the whole process more streamlined. We demonstrate the effectiveness of our approach in SNP calling and phylogenetic analyses by using synthetic mixed culture samples of closely related samples from previous genomic epidemiology studies (Brodrick et al. 2017; Raven et al. 2016; Paterson et al. 2015) executed in a standard manner in nosocomial settings for the three major pathogens *E. faecalis*, *E. coli* and *S. aureus*. Our results illustrate that accurate transmission and case-linking analyses are possible at reduced cost levels by enabling sample de-mixing and subsequent variant calling.

Key parts of our pipeline presented in this paper are the mGEMS binner for short-read sequencing data, and the scalable pseudoaligner Themisto, which provides an exact version of the kallisto pseudoalignment algorithm (Bray et al. 2016) for large reference databases of single-clone sequenced bacterial pathogens. Together with recent advances in both probabilistic modelling of mixed bacterial samples (Mäklin et al. 2020) and genome assembly techniques (Seemann 2018), these methods form the mGEMS pipeline. A central step in mGEMS is an application of the recent mSWEEP method (Mäklin et al. 2020), which estimates the relative abundance of reference bacterial lineages in mixed samples using pseudoalignment and Bayesian mixture modelling. While Themisto enables upscaling of mSWEEP to significantly larger reference databases, the mGEMS binner is a novel sequencing read binning approach. Our binner is based on leveraging probabilistic sequencing read classifications to reference lineages from mSWEEP, and notably allowing a single read to be assigned to multiple bins. Using mGEMS to bin the reads in the original mixed samples produces sets of reads closely resembling standard isolate sequencing data and additionally acts as a denoising step for removing possible contaminant DNA. These advances allow a subsequent efficient use of the existing leading tools for genomic epidemiology in the analysis of mixed culture samples, which can pave way to a more widespread consideration of genomic epidemiology for public health applications.

Results

Read binning and genome assembly from mixed samples with mGEMS

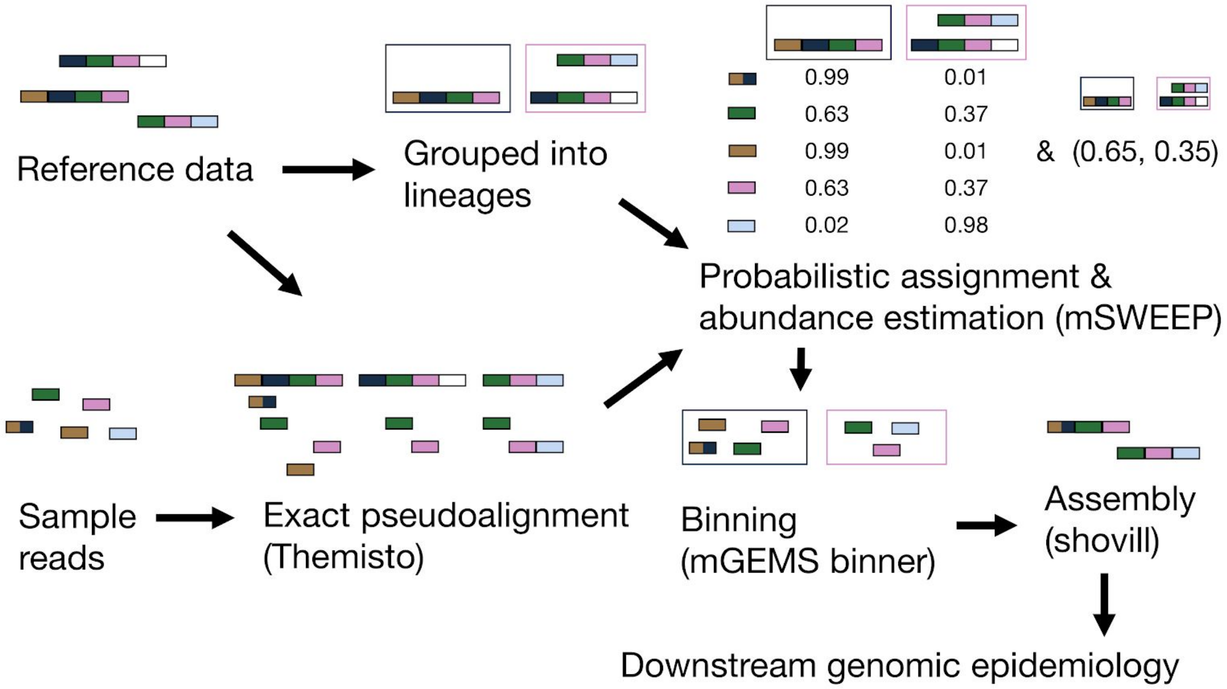


Figure 1 Flowchart describing a genomic epidemiology workflow with the mGEMS pipeline. The figure shows the various steps of the pipeline. Steps with program names in brackets constitute the parts of the mGEMS pipeline. Presented values from mSWEEP and mGEMS binner are the actual results of running the pipeline with the described input.

Our mGEMS read binning algorithm, part of the mGEMS pipeline (Figure 1), requires probabilistic assignments of sequencing reads to reference taxonomic units (lineages or sequences) and an estimate of the relative sequence abundance of these same references in the full set of reads. mGEMS then bins the reads by assigning a read to a bin (corresponding to a target sequence from a given reference lineage) if the read-level probabilistic assignment value of the lineage is greater or equal to the sequence abundance of that particular lineage in the full set of reads. Notably, this algorithm allows a single sequencing read to be assigned to multiple bins which is a crucial feature for considering strain-level variation. As shown in the Methods section, this algorithm assigns reads to reference lineages only if the sequence represented by a read is likely contained in a target sequence that belongs to the reference lineage. In the rest of the pipeline (Figure 1), we use our own more efficient and accurate implementation of the pseudoalignment algorithm in kallisto (Bray et al. 2016), called Themisto, to pseudoalign the sequencing reads against the reference sequences. Themisto is based on using colored de Bruijn graphs to represent the reference sequences and disk storage to control the amount of memory required in constructing the pseudoalignment index. These choices lead to Themisto aligning a similar number of reads per hour as kallisto, while being 70 times faster to load in an example pseudoalignment index consisting of 3682 *E. coli* sequences (28 minutes for kallisto and 0.55 minutes for Themisto; Supplementary Methods 1). Implementation of the method is described in more detail in Supplementary Methods 1. The pseudoalignments from Themisto are used as input to the mSWEEP method (Mäklin et al. 2020) to estimate the probabilistic read assignments and whole-sample relative sequence abundances. These values provide the necessary input to the mGEMS binner which assigns the sequencing reads to the bins. Finally, we use the shovill (Seemann 2018) assembly optimizer for the SPAdes assembler (Bankevich et al. 2012; Nurk et al. 2013) to assemble the bins. On an example synthetic mixed sample (the *E. coli* sample with the most reads), the full mGEMS pipeline took 112 minutes to run (Themisto 26 min, mSWEEP 4 min, mGEMS binner 16 min, and shovill 66 min) using two threads on a laptop computer with two processor cores and 16 gigabytes of memory. C++ implementations of both the mGEMS binner and the Themisto pseudoaligner are freely available on GitHub (https://github.com/PROBIC/mGEMS, MIT license, and https://github.com/algbio/themisto, GPLv2 license).

Overview of the experiments used in benchmarking mGEMS

We assessed the accuracy and effectiveness of mGEMS by considering data from three genomic epidemiological studies (Brodrick et al. 2017; Raven et al. 2016; Paterson et al. 2015). We mixed colony pick isolate sequencing data from these studies synthetically and compared the pipeline outputs against the benchmark of having non-mixed data available for the epidemiological analysis. The synthetic experiments presented are: 1) mixing reads from three clones of *E. coli* sequence type (ST) 131 sublineages obtained from a study of multidrug-resistant *E. coli* ST131 strains circulating in a long-term care facility in the UK (Brodrick et al. 2017), 2) mixing reads from seven *E. faecalis* STs identified in a study of the population structure of hospital-acquired vancomycin-resistant *E. faecalis* lineages in the UK and Ireland (Raven et al. 2016), and 3) mixing reads from three *S. aureus* ST22 sublineages from a study of the transmission network of methicillin-resistant *S. aureus* (MRSA) among staff and patients at an UK veterinary hospital (Paterson et al. 2015). We also provide three different approaches to constructing the reference datasets for the pseudoalignment step: 1) a national (UK) collection of *E. coli* ST131 isolates associated with bacteremia (Kallonen et al. 2017), 2) a global collection of all available *E. faecalis* genome assemblies from the NCBI as of 2 February 2020, and 3) a local collection of *S. aureus* sequencing data from the staff members at the veterinary hospital at the earliest possible time point in the same study (Paterson et al. 2015). A detailed description of the generated experiments and the accession numbers of the isolate sequencing and reference data used is presented in the Methods section.

SNPs from binned reads match SNPs called from isolate data

First, we compared the accuracy of SNP calling with the snippy software (version 4.4.5) (Seemann 2014) from the bins obtained by processing the abundance estimation results from the mixed samples with the mGEMS binner with the results of the same analyses from the isolate sequencing data (Figure 2). In the *E. coli* and *E. faecalis* experiments (Figure 2 panels a and b respectively), the SNPs were called from assembled contigs while in the *S. aureus* experiment (Figure 2 panel c), we called the SNPs directly from the sequencing reads because calling the SNPs from the contigs resulted in poorer performance (Supplementary Figure 1). In all experiments, the SNPs called from the mixed samples closely resemble the results of isolate sequencing data in both the samples that are similar and dissimilar to the reference sample. Although in the *E. coli* experiment using mGEMS produced more SNPs on average, the results were consistently higher for all samples and did not affect the results of the analyses presented further in this article.

We suspected that the observed differences in the SNP counts may have been caused by issues in the sequence assembly due to mGEMS allowing a read to belong to multiple bins, which results in variable coverage between the regions with and without the clade-specific SNPs. We tested this assumption by replacing the shovill assembler in the mGEMS pipeline with metagenomic assemblers, which naturally handle variable coverage. However, while using the metagenomic assemblers marginally improved the results in some of the experiments (Figure 2 panel d, Supplementary Figure 2), the improvements were not drastic enough to decisively confirm our suspicions about the accuracy of the SNP calling being limited by the choice of the assembler. We did observe that when measured by reference-independent assembly statistics (sum of all contig lengths, total number of contigs, sequence length of the shortest contig at 50% genome length N50, and the smallest number of contigs whose sum is at least 50% of the genome length L50), the statistics obtained from the standard configuration of mGEMS with the shovill assembler resemble those from isolate sequencing data.

Further assessment of the accuracy of our called SNPs was done by fitting a Bayesian linear regression model to the same SNP data with the isolate results as the sole explanatory variable and the results from the bins or the metagenomic assemblers as the response variable (Figure 2 and Supplementary Figure 2) using the brms R package (Bürkner 2017, 2018; Carpenter et al. 2017). In both the *E. coli* ST131 sublineage and the *E. faecalis* experiments, the 95% posterior credible interval for the slope from mGEMS with all assembler choices except metaSPAdes contains the correct value of 1.0. The *S. aureus* experiments produce worse 95% credible intervals for the slope with none of the intervals containing the correct value. However, the number of SNPs between the clades is three to four orders of magnitude less than in the *E. coli* and *E. faecalis* experiment and the practical differences in the values are quite small.

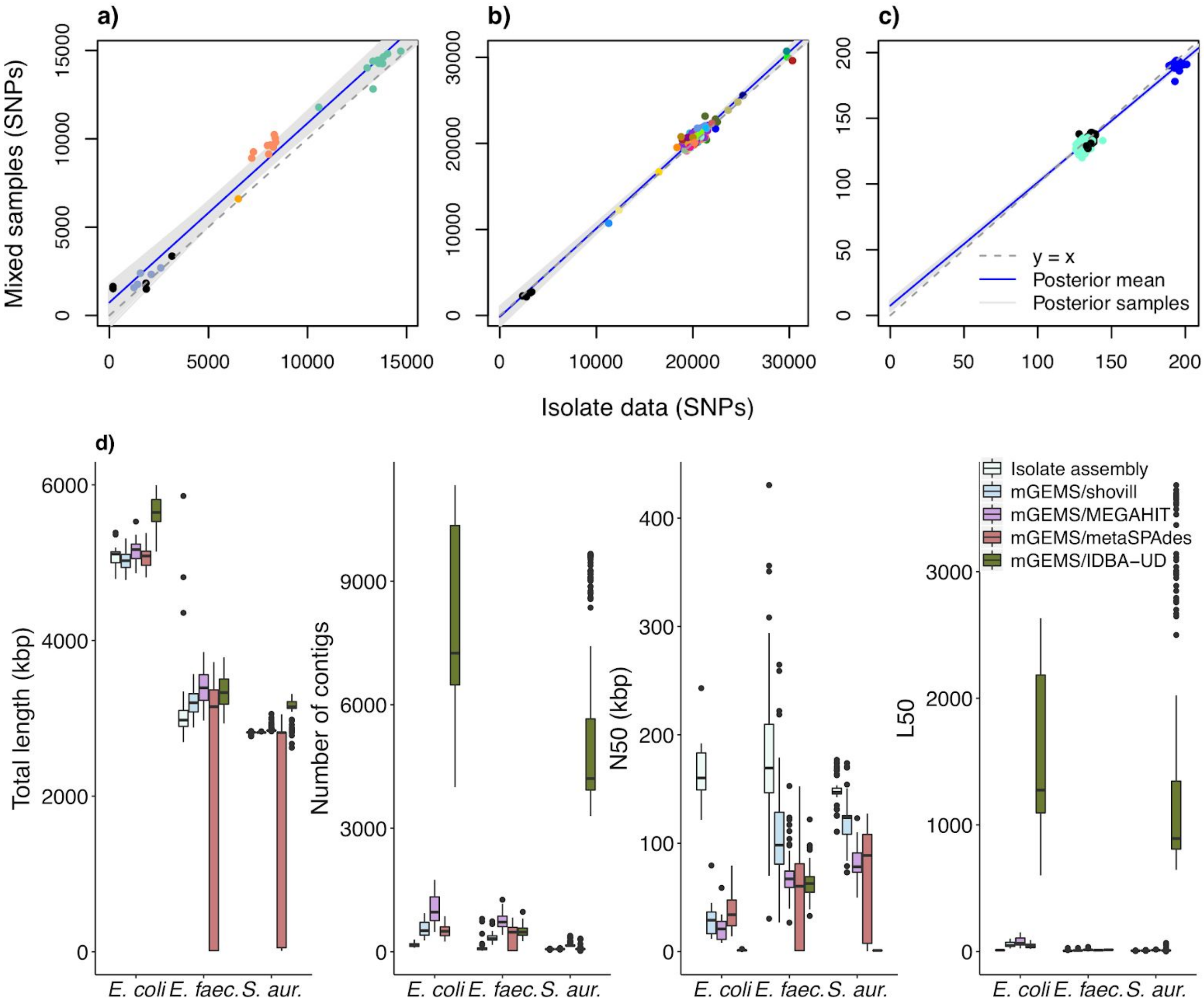


Figure 2 Comparing mGEMS with isolate sequencing data. Panels **a**, **b**, and **c** compare the results of SNP calling from mixed samples with the mGEMS pipeline against the results from isolate sequencing data. Panel **d** compares reference-free assembly statistics from mGEMS pipeline with different assemblers against the results from assembling the isolate sequencing data with shovill. The results in panel **a** are for the *E. coli* ST131 isolates, panel **b** the *E. faecalis* isolates, and panel **c** the *S. aureus* ST22 isolates. In panels **a** and **b**, SNPs were called from contigs after assembling the reads. In panel **c**, the SNPs were called directly from the reads. Points are colored according to the lineage within the species. The dashed gray line represents a hypothetical perfect match between the binned and isolate reads. The blue line is the posterior mean while the shaded area contains the 95% posterior credible region calculated from 10 000 posterior samples from a Bayesian regression model with the SNPs from the binned reads as the response and the SNPs from the isolate sequencing data as the sole explanatory variable. In panel **d**, the boxes are colored according to the type of assembly. The presented statistics are the summed lengths of all contigs (total length), the number of contigs, the sequence length of the shortest contig at 50% genome length (N50), and the smallest number of contigs whose sum of lengths is at least 50% of the genome length (L50).

Split-*k*-mer comparison between isolate reads and mGEMS bins

We also examined the accuracy of the mGEMS binner without assembling by using the split *k*-mer analysis provided by the SKA software (version 1.0) (Harris 2018). In a split-*k*-mer analysis, each basepair in the read is flanked by two *k*-mers. The base pair in the middle position plus the flanking *k*-mers constitute a single split-*k*-mer. If the split-*k*-mers are calculated for all base pairs in two samples, they can be used to compare the samples on the basis of matching or mismatching split-*k*-mers or to call SNPs by comparing two split-*k*-mers where the flanking *k*-mers match but the base pair in between does not.

We first used SKA to call split-*15*-mer-SNPs in the three reference sequences from the binned sequencing reads, and calculated the difference in the count of SNPs called in the reference sequence between the isolate and the binned reads (Supplementary Figure 3). Since the results in Figure 2 for *S. aureus* were obtained without assembly, there is no notable difference when compared to the SKA results. However, the SKA results for *E. coli* and *E. faecalis* contain fewer SNPs called from the binned reads, implying that binning with mGEMS acts as filtering for the sequencing data, since the results from the assemblies display no stark differences. Next, we performed pairwise comparisons within the separate sets of 1) all isolate reads and 2) the binned reads by calculating the pairwise matching and mismatching split-*15*-mers and calling pairwise split-*15*-mer-SNPs. Then, we compared the pairwise results from the isolate reads to the results from the binned reads. While comparing these pairwise differences shows more discrepancy (Supplementary Figure 4) than the comparison considering only SNPs called in the reference genome, the pairwise SNP counts are still relatively well preserved in all three experiments.

Phylogenetic analysis of *Escherichia coli* ST131 sublineages in a long-term care facility

We used a set of 30 multidrug-resistant *E. coli* ST131 strains sequenced from the residents of a long-term care facility in the UK (Brodrick et al. 2017) to produce a total of 10 synthetic mixed samples. Each sample was the result of mixing isolate sequencing data from three *E. coli* ST131 sublineages (one from each of the main lineages A, B, or C) together. We attempted to preserve the potential sequencing errors and biases by using all available reads from each of the isolate samples. We applied the mGEMS pipeline to the 10 synthetic mixed samples with a national (from the UK) collection of *E. coli* ST131 strains as the reference data (Kallonen et al. 2017), and used RAxML-NG (version 0.8.1) (Kozlov et al. 2019) to infer a phylogenetic tree from both assemblies obtained from the isolate sequencing data (ground truth) and the assemblies from the mGEMS pipeline. Comparing these two trees (Figure 2), shows that the overall structure of the trees is remarkably similar, with the global structure between the clades completely recovered and, locally, most leaves having the same neighbors. While the phylogeny inferred with the mGEMS pipeline tends to contain longer branch lengths within the clades, the bootstrap support values do not show overly confident predictions — instead being in line with the values from the isolate data.

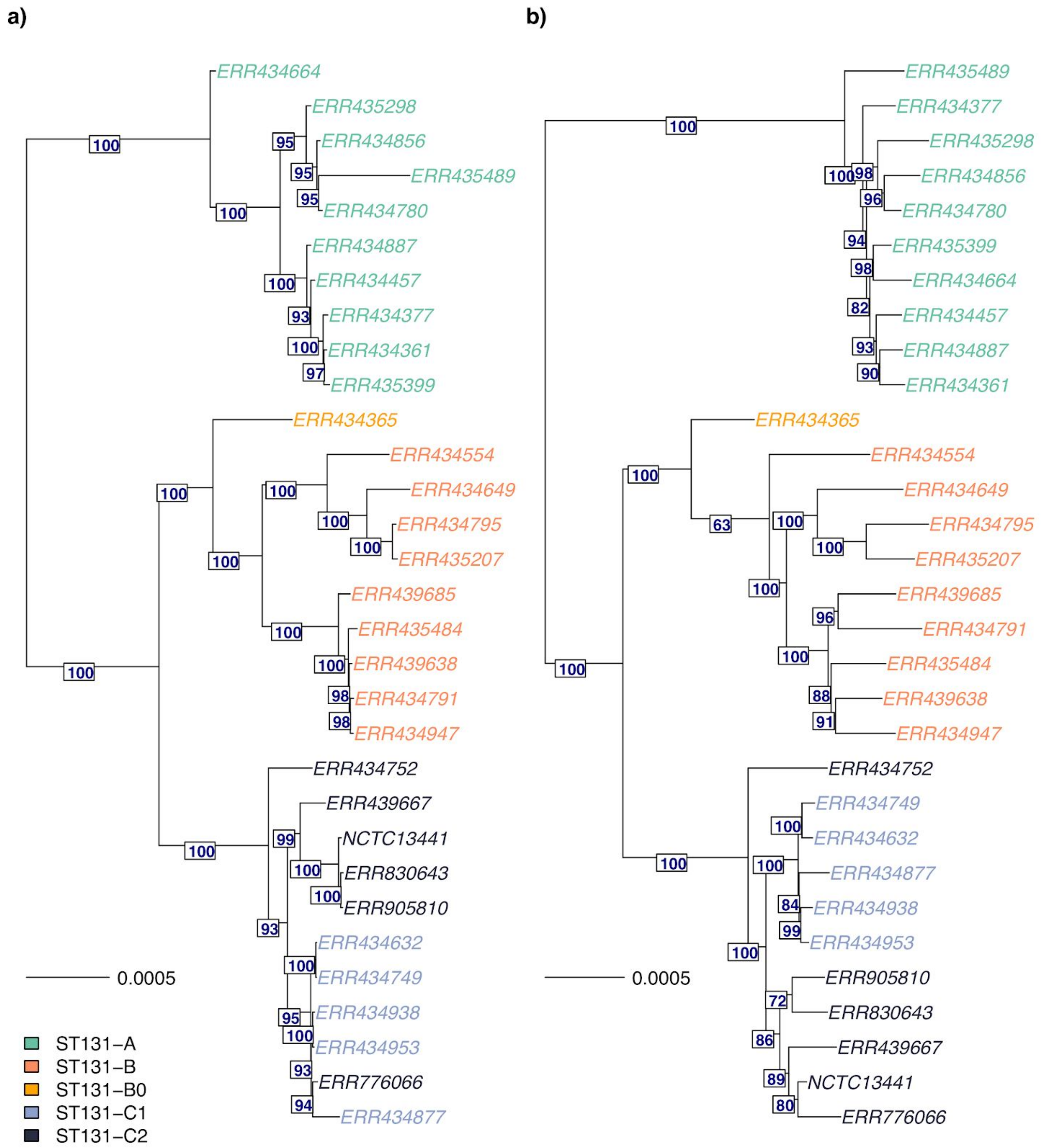


Figure 3 Midpoint-rooted maximum likelihood trees from core SNP alignment of *Escherichia coli* ST131 strains. The phylogeny in panel **a** was constructed from isolate sequencing data from 30 *E. coli* ST131 strains, and the phylogeny in panel **b** with the mGEMS pipeline from 10 synthetic plate sweep samples, each mixing three isolate samples from the three main ST131 lineages (A, B, and C; one strain from each per sample). Both phylogenies were inferred with RAxML-NG. Numbers below the edges are the Felsenstein bootstrap support values from RAxML-NG for the next branch. Leaves are coloured according to the *E. coli* ST131 sublineage (A, B, B0, C1, or C2), and branch lengths in the tree scale with the mean number of nucleotide substitutions per site on the respective branch (GTR+G4 model). Leaves are labeled with the ENA accession number and the leaf labeled *NTCC13411* corresponds to the reference strain used in calling the core SNPs.

Population structure of nosocomial *Enterococcus faecalis* infections in the UK

Our next experiment was performed on sequencing data from bloodstream-infection-associated *E. faecalis* strains with a high prevalence of vancomycin-resistance circulating in hospitals in the UK (Raven et al. 2016). In this experiment, we mixed together isolate sequencing data from seven distinct *E. faecalis* STs (Ruiz-Garbajosa et al. 2006), producing a total of 12 synthetic mixed samples with seven clones present in each. Each synthetic mixed sample included all sequencing reads from the mixed isolate sequencing data similarly to the *E. coli* experiment. We used a global collection of *E. faecalis* strains (all *E. faecalis* genome assemblies submitted to the NCBI as of 2 February 2020) as the reference data for the mGEMS pipeline, and again inferred the phylogenies for assemblies from both the isolate sequencing data and the results of the mGEMS pipeline. The more complex structure of these phylogenies was compared by plotting the two phylogenies against each other in a tanglegram (Figure 4). Apart from a few structural mismatches in branches with poor bootstrap support values in both phylogenies (indicating uncertainty in the structure to begin with), the tree structure is strikingly well-recovered from the binned reads.

In fact, the tree inferred with the mGEMS pipeline has better bootstrap support values in the lower parts of the tree, suggesting that using mGEMS provides a better phylogeny than using the isolate sequencing data alone. We suspect this improvement in the bootstrap support values was caused by contamination in the isolate sequencing data for BSAC ec750, which produces an assembly 5.8Mb long — nearly twice the length of the reference *E. faecalis* strain V583 (3.2Mb). Similar changes in the bootstrap support values and additional structural changes occur in the parts of the tree containing the isolates BSAC ec294 and BSAC ec655 which both produce abnormally long assemblies (4.8Mb and 4.4Mb, respectively). The assembly lengths for both the isolate and mGEMS-binned sequencing reads are provided in Supplementary Table 1.

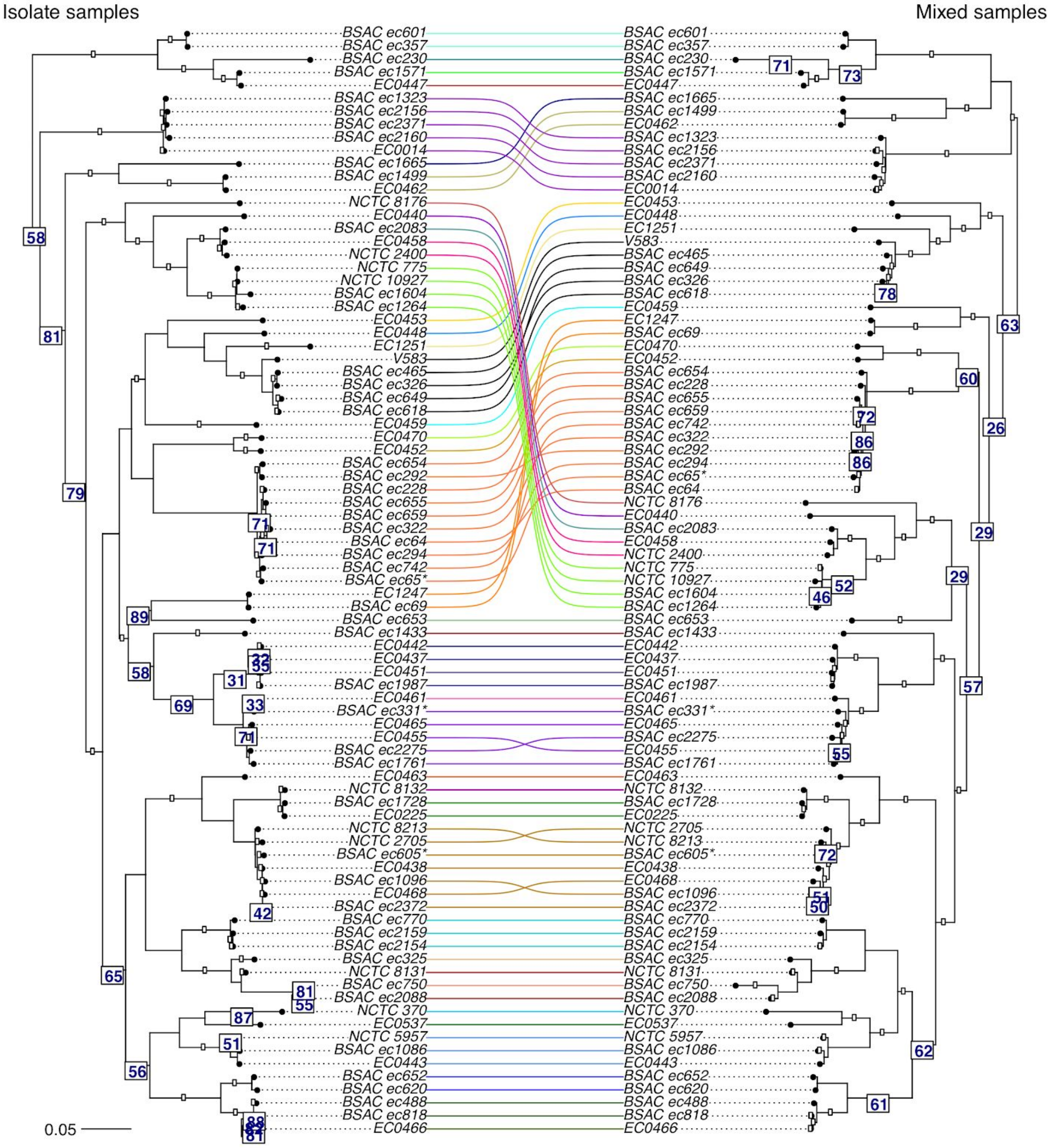


Figure 4 Tanglegram of two midpoint-rooted maximum likelihood trees from core SNP alignment of *Enterococcus faecalis* strains. The phylogeny labelled *Isolate samples* was inferred with RAXML-NG from assembling the isolate sequencing data from 84 *E. faecalis* strains. The phylogeny labelled *Mixed samples* was inferred from 12 synthetic mixed samples, each containing sequencing data from seven different *E. faecalis* STs randomly chosen from the isolate sequencing data. Numbers below the edges indicate Felsenstein bootstrap support values from RAxML-NG for the next branch towards the leaves of the tree. Only support values *less than* 90 are shown. Branches are coloured according to the *E. faecalis* STs, and branch lengths in the tree scale with the mean number of nucleotide substitutions per site on the respective branch (GTR+G4 model). Leaves are labeled with the strain name from NCBI and the leaf labeled *V583* corresponds to the reference strain for calling the core SNPs.

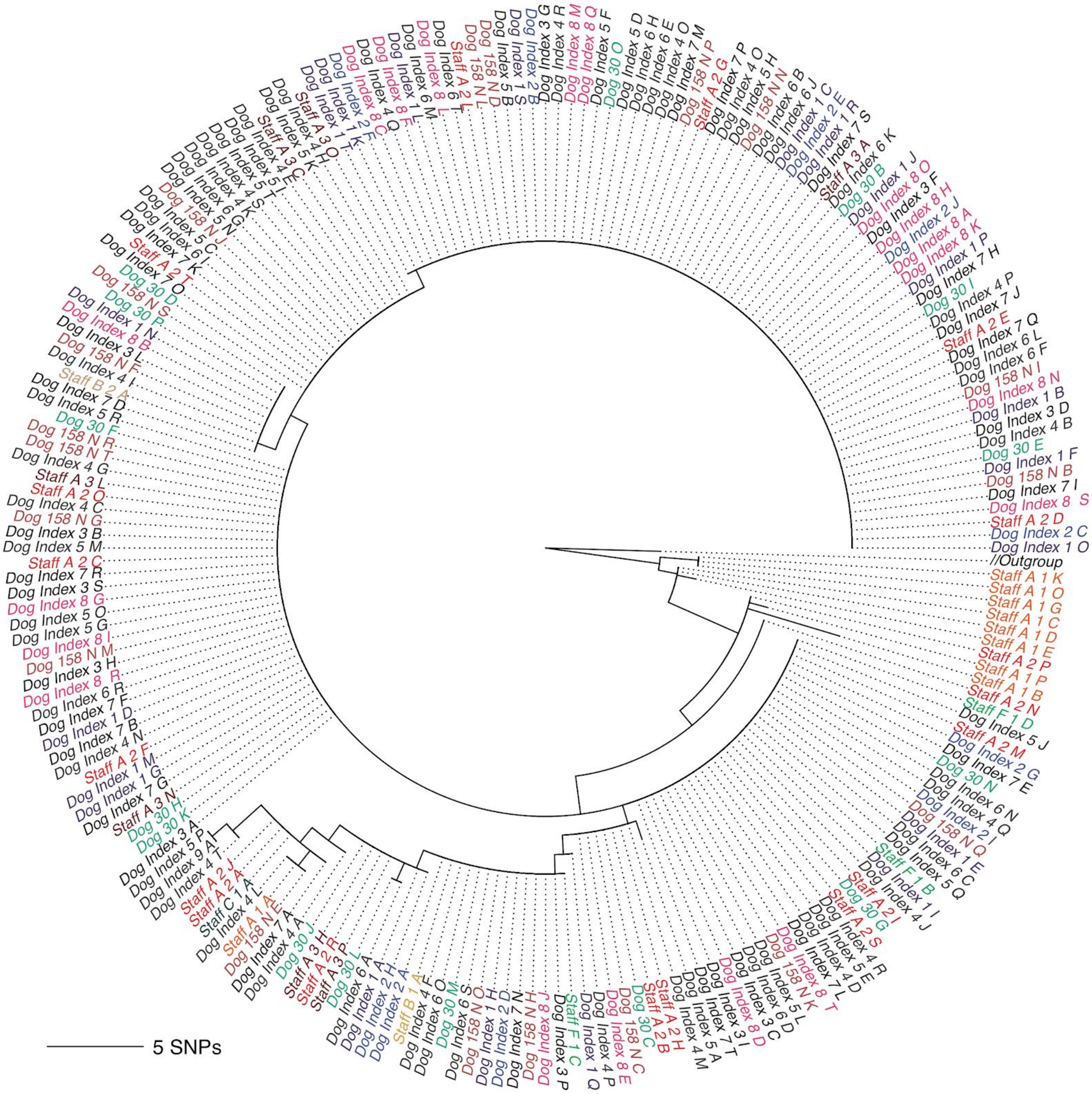


Figure 5 Midpoint-rooted maximum likelihood tree from core SNP alignment of *Staphylococcus aureus* ST22 showing the clade 1 strains. The phylogeny was inferred from a combined set of assemblies from 60 isolate sequencing samples (leaves labelled Staff A-G 1 A-T, corresponding to the temporally first samples from each staff member) and 312 assemblies obtained from the mGEMS pipeline applied to synthetic mixed samples of sequencing data from each of the three different *S. aureus* ST22 clades (1, 2, and 3). The mixed samples were produced from the isolate sequencing data collected from the patients, or from the staff members after the first sampling time. The branch labelled Outgroup leads to clades 2 and 3, which are not shown. Branch labels are coloured according to the plate the isolate sequencing data was picked from. Branch lengths in the phylogeny scale with the mean number of SNPs obtained by multiplying the mean nucleotide substitutions per site on the respective branch (GTR+G4 model) with the total

number of alignment sites. Leaves are labeled with the format: staff or patient, a letter indicating the donor, plate number (ascending in time), and a letter indicating the colony pick id.

In our last experiment, we used a dataset containing *S. aureus* ST22 sublineages (clade 1, clade 2, and clade 3) circulating among the staff and patients at a veterinary hospital in the UK (Paterson et al. 2015) and separated by less than 150 SNPs. Because of the minimal differences between the clades, and a lack of isolates from these clades in published sources, we decided to use the isolates from the temporally first sample from the staff members as the reference data (representing a local reference collection). We separated the reference isolates from our experiment cases, and proceeded to mix the remaining isolate sequencing data together. We generated a total of 312 synthetic mixed samples, each containing the sequencing data from three isolate samples from each of the three clades. Because the numbers of samples in each clade were not equal, the data from some of the isolate samples was contained in multiple mixed samples. Since we wanted to represent each isolate with only a single instance in the phylogeny, we randomly chose one corresponding bin from mGEMS as the representative for an isolate that was included in multiple mixed samples.

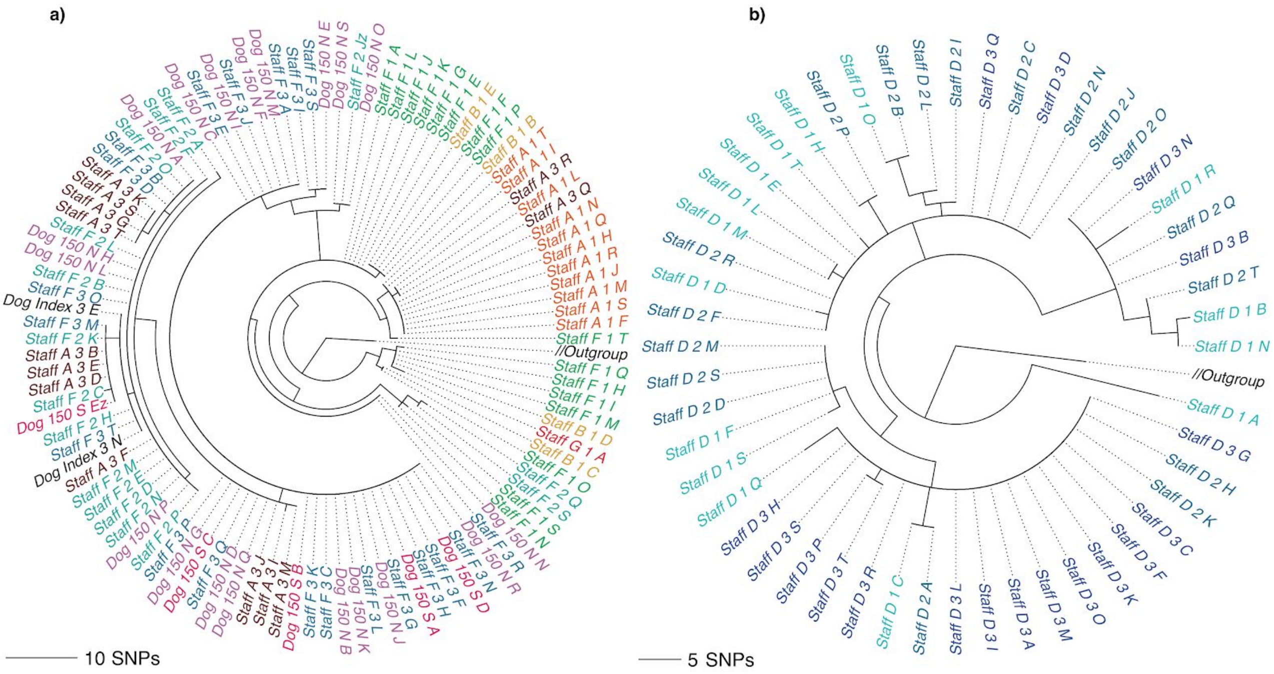


Figure 6 Midpoint-rooted maximum likelihood trees from core SNP alignment of *Staphylococcus aureus* ST22 showing clade 2 and clade 3 strains. The underlying phylogeny is the same as in Figure 3. The phylogeny in panel **a** contains the clade 2 strains, and panel **b** the clade 3 strains. Branches leading to clade 1 were removed. Branch labels are coloured according to the plate the isolate sequencing data was originally picked from with darker shades indicating later sampling times. Branch lengths in the phylogeny scale with the mean number of SNPs obtained by multiplying the mean nucleotide substitutions per site on the respective branch (GTR+G4 model) with the total number of alignment sites. Leaves are labeled with the format: staff or patient, a letter indicating the donor, plate number (ascending in time), and a letter indicating the colony pick id.

The phylogenies in Figures 5 and 6 were inferred with RAxML-NG (version 0.8.1) (Kozlov et al. 2019) from the results of the mGEMS pipeline. We plotted the phylogenies separately for the clade 1 isolates (Figure 5) and clade 2 and 3 isolates (Figure 6) without changing the underlying tree structure. Phylogenies inferred from the isolate sequencing data using the same pipeline are available in Supplementary Figure 5 and Supplementary Figure 6. In the original study (Paterson et al. 2015), Staff member A was inferred as having introduced the MRSA strain from Clade 1 into the veterinary hospital. In our phylogeny, Staff member A's initial samples (timepoints labels 1 and 2) are indeed contained at the root of the tree inferred from the mGEMS pipeline, although the placement of the strains further up the tree vary when compared to the results presented in the original study. The original study performed manual quality control of the SNP data by removing transposable elements which was not replicated in our experiment, possibly explaining some of the observed differences between the tree structures. The phylogenies for clades 2 and 3 (Figure 6) follow the results of the original study more closely with most subclades found in both the isolate and the mixed sample phylogenies. Importantly, in all three clades no assembly from the mGEMS pipeline was assigned to the wrong clade in the phylogeny despite the minimal distances between the clades.

Comparison with metagenomic assemblers

We benchmarked our method against three metagenomic assemblers: IDBA-UD (v1.1.3) (Peng et al. 2012), MEGAHIT (v1.2.9) (Li et al. 2016, 2015), and metaSPAdes (v3.14.0) (Nurk et al. 2017). These tools represent the state-of-the-art in metagenomic assembly based on their performance in benchmarking studies (Sczyrba et al. 2017; McIntyre et al. 2017; Vollmers et al. 2017; Meyer et al. 2018). We compared the results of assembling the synthetic mixed *E. coli*, *E. faecalis*, and *S. aureus* sample with these three methods with assemblies obtained from the mGEMS pipeline. The reference-dependent assembly statistics (Figure 7) were obtained by comparing both the mGEMS assemblies, with the standard configuration using shovill as the assembler, and the metagenomic assemblies against reference genomes constructed by assembling the isolate sequencing data contributing to the mixed samples. We used metaQUAST (v5.0.2) (Mikheenko et al. 2016) to calculate the comparison statistics. Based on the results from running the metagenomic assemblers, we opted not to run taxonomic contig binners since their performance on alignment-based statistics will necessarily be worse than that of the metagenomic assembler used as the input.

Our results (Figure 7) show that the metagenomic assemblers struggle in all three experiment sets on all four presented statistics (fraction of bases in the reference assembly that a base from the compared assembly aligns to, the sequence length of the shortest contig at 50% of total reference genome length, difference in the total number of bases in the reference genome minus the total number of aligned bases in the compared sequence, and the number of mismatches per 1000 aligned base pairs). In contrast, assembling the bins from mGEMS produces significantly better results in nearly all experiments and assembly statistics ($p < 10^{-4}$ in the cases where mGEMS is better, Wilcoxon rank-sum test, rejecting the hypotheses: 1) the values of genome fraction or NGA50 from mGEMS are less than those from the compared methods, or 2) absolute values of the length difference or mismatches per 1000 bp from mGEMS are greater than those from the compared methods), with the exception of the NGA50 value in the *E. coli* experiments where MEGAHIT and metaSPAdes outperform mGEMS ($p < 10^{-9}$, Wilcoxon rank-sum test, rejecting the hypothesis: the values from metaSPAdes or MEGAHIT for NGA50 are less than from mGEMS).

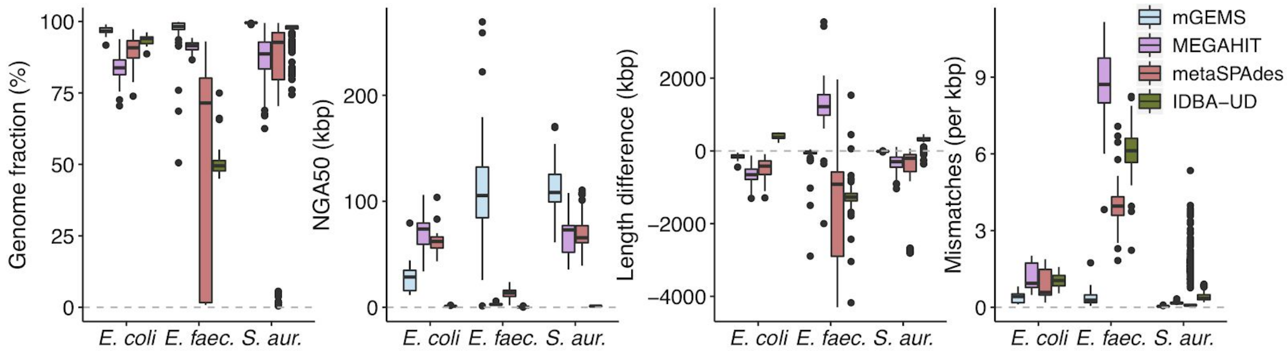


Figure 7 Comparison between mGEMS and other methods on three sets of synthetic mixed samples. The figure shows four assembly statistics: percentage of aligned bases in the reference genome (genome fraction), the sequence length of the shortest contig at 50% of total reference genome length (NGA50), the difference between the total number of bases in the reference genome and the total number of aligned bases in the compared genome (length difference), and the number of mismatches in aligned bases per 1000 base pairs (mismatches). The dashed gray line shows the location of zero in each of the subplots. The statistics were calculated by assembling either the bins obtained with the mGEMS pipeline or the full mixed sample with MEGAHIT, metaSPAdes, or IDBA-UD. We used metaQUAST to align the resulting contigs against ground-truth genomes obtained by assembling the isolate reads contained in the mixed samples. The boxplots are grouped based on the species contained in the sample, and the colors indicate the four different methods used for obtaining the assemblies.

Discussion

Adopting a plate-sweep approach, where DNA from the individual bacteria growing on the same plate is prepared and sequenced en masse, shows clear promise in reducing the amount of manual and costly laboratory work that has been identified as an emerging bottleneck for epidemiological analyses at many public health laboratories (Rossen et al. 2018). In this article we have introduced the mGEMS pipeline, which includes novel pseudoalignment and read binning methods, for genomic epidemiological analyses of plate sweeps. Our pipeline provides means to accurately recover the genomes, or corresponding sequencing reads, from mixed samples with extremely closely related strains separated by less than a few dozen SNPs. In these settings, where the differences between the strains are at or under the sequence type level, isolate sequencing is traditionally required to draw epidemiological conclusions. We have demonstrated that the same conclusions can be robustly made from plate sweeps by using mGEMS. Additionally, since the pipeline relies on modelling pseudoalignments against reference sequences, mGEMS acts as quality control for sequencing reads from samples that inadvertently contain multiple lineages or contamination, which can disrupt downstream analyses like SNP calling (Goig et al. 2020). Our pipeline also significantly outperforms the current state-of-the-art in analysing sequencing data from closely related mixed samples, reaching accuracy levels likely constrained by technical variation in the sequencing data and limitations in assembling sequencing data with variable coverage. To our knowledge, mGEMS is the first tool capable of reliable recovery of the full strain variety in complex mixed samples.

mGEMS demonstrates the power of plate sweep sequencing in genomic epidemiology and enables a change in the currently dominant framework that confers multiple benefits over both whole-genome shotgun metagenomics and isolate sequencing. Studies of the population structures of opportunistic pathogens have revealed extensive strain-level within-host variation (Stoesser et al. 2015; Golubchik et al. 2013; Paterson et al. 2015; Greenblum et al. 2015; Brodrick et al. 2017; Lieberman et al. 2014) with adverse implications for transmission analyses relying solely on isolate sequencing (Worby et al. 2014; Stoesser et al. 2015) and longitudinal studies reporting the absence or re-emergence of strains in a host based on colony picks (Paterson et al. 2015; Brodrick et al. 2016, 2017). While whole-genome shotgun metagenomics solves these issues to some extent (Gu et al. 2019; Forbes et al. 2017), the culture-free nature suffers from issues with both bacterial and host DNA contamination particularly affecting the sensitivity for detecting strains in low abundance (Whelan et al. 2020; Ivy et al. 2018; McArdle and Kaforou 2020; Salter et al. 2014). Using mGEMS in conjunction with plate sweep sequencing data avoids these issues altogether, paving way for more representative studies of pathogen population structure and providing higher-resolution data for more complex models of transmission dynamics incorporating within-host variation and evolution (Maio et al. 2018; Worby et al. 2017; Skums et al. 2018).

Since our method relies on available single-clone genomic reference data and plate cultures of the bacteria to sequence them at a sufficient depth for assembly, it obviously cannot be applied to the study of uncharacterized or unculturable species. However, culture media do exist for most human pathogens of public health relevance (Lagier et al. 2015) or can be developed for some of the allegedly unculturable bacteria (Stewart 2012; Vartoukian et al. 2010; Ito et al. 2019). Moreover, the availability of single-clone genome sequences is still increasing at a high rate, such that for many of them plenty of sufficiently representative reference sequences would be available (Forster et al. 2019; Zou et al. 2019). In these cases, the drastic reduction in the costs of library preparation, and the better capture of the underlying genomic variation between closely related bacteria in a set of mixed samples provided by mGEMS is extremely valuable. We hope that by enabling significant streamlining of the process of producing data for public health genomic epidemiology, our approach inspires both applications and further method development within this exciting research area.

Methods

mGEMS workflow

Our pipeline for performing genomic epidemiology with short-read sequencing data from mixed samples, mGEMS, requires as input the sequencing reads and a reference database representing the clonal variation in the organisms likely contained in these reads. The reference database must additionally be grouped accordingly into clonal groups representing lineages within the species. We used either the multilocus sequence types (*E. faecalis* experiments) or sublineages within the sequence types (*E. coli* and *S. aureus* experiments) as the clonal grouping. With these preprocessing steps performed, the first step in the mGEMS pipeline is to pseudoalign the sequencing reads against the reference database using our scalable implementation of (exact) pseudoalignment with the Themisto software (in this article we used v0.1.1 with the optional setting to also align the reverse complement of the reads enabled). The pseudoalignments and the clonal grouping are then supplied as input to the mSWEEP software (v1.3.2; doi: 10.5281/zenodo.3631062, with default settings) (Mäklin et al. 2020) which estimates the relative sequence abundances of the clonal groups in the mixed sample. Consequently, mSWEEP produces a probabilistic classification of the sequencing reads to the different reference clonal groups. This classification is subsequently processed by the mGEMS binner (v0.1.1, default settings), which assigns the sequencing reads to bins that correspond to a single reference clonal group — with a possibility for a sequencing read to belong to multiple bins. As the last step, the bins are (optionally) assembled with the shovill (v0.9.0, with default settings) (Seemann 2018) assembly pipeline. mGEMS and Themisto are freely available on GitHub (https://github.com/PROBIC/mGEMS and https://github.com/algbio/themisto).

Reference data

We used three different sets of sequencing data as the reference for the three different experiments presented. The three different reference datasets represent a local (*S. aureus* experiment), a national (*E. coli*), and a global collection (*E. faecalis*) of strains from these species. Accession numbers and multilocus sequence types for the reference data are available in Supplementary Table 2 accompanied with rudimentary assembly statistics from both the isolate sequencing data and the assemblies from the mGEMS pipeline. In each experiment, we only aligned against the reference sequences from the relevant species.

In the *E. coli* experiments, our collection of 218 *E. coli* ST131 isolates originated from the British Society for Antimicrobial Chemotherapy's bacteraemia resistance surveillance program and were originally isolated from 11 hospitals across England

(Kallonen et al. 2017) . These isolates were assigned to five ST131 sublineages (A, B0, B, C1, or C2) as described previously (Kallonen et al. 2017) . As the reference sequence for calling the SNPs in building the phylogeny, we used the ST131 strain NCTC13441 (European Nucleotide Archive [ENA] sequence set UFZF01000000).

The global collection of *E. faecalis* reference data was obtained by downloading all available *E. faecalis* assemblies (1484 as of 2 February 2020) from the NCBI, which were assigned to STs with the mlst software (version 2.18.1) (Jolley et al. 2018; Seemann 2015; Ruiz-Garbajosa et al. 2006). Sequence type could not be determined for 177 assemblies. These were discarded, leaving a total of 1307 assemblies assigned to 203 distinct sequence types. We used the ST6 strain V583 (Paulsen et al. 2003) as the reference for SNP calling (NCBI RefSeq sequences NC_004668.1-NC_004671.1).

The *S. aureus* reference data was obtained from the same study as the experiment data (Paterson et al. 2015). We used shovill (version 0.9.0 with default settings) (Seemann 2018) to assemble the isolate sequencing reads from the first sampling of the staff members at the veterinary hospital, and assigned the assembled sequences to the ST22 sublineages according to the information provided in original study (Paterson et al. 2015). The reference sequence used in calling the SNPs was the ST22 strain HO 5096 0412 (Holden et al. 2013) (ENA sequence HE681097.1)

If the reference sequence in any of the experiments consisted of multiple contigs, we concatenated the contigs together by adding a 100-base gap between them. The final reference file that was used as input for Themisto indexing was produced by concatenating all reference sequences processed in this way together.

Synthetic experiment generation

We produced our three synthetic experiment sets by synthetically mixing together the isolate sequencing data from distinct lineages in each of the three studies. In the *E. coli* experiments, we produced 10 mixed samples with one strain from each of the three main ST131 lineages (A, B, or C) in each sample. In the *E. faecalis* experiments, we mixed together seven strains from seven different sequence types to produce a total of 12 mixed samples. The strains included in each sample were chosen at random without replacement in the *E. coli* and *E. faecalis* experiments. The *S. aureus* mixed samples were produced by randomly mixing together one strain from each of the three sublineages with replacement while ensuring that each strain appears at least once. The sequencing data that was used in the reference dataset was not included in any of the experiments. In all three experiment sets, we used all available sequencing data in the mixed samples, resulting in 8-15 million reads in the experiments. Supplementary Table 1 contains the accession numbers and lineage assignments of the isolate sequencing data in each sample, as well as the assembly statistics from both isolate sequencing and the synthetic mixed samples processed with mGEMS.

Pseudoalignment

We used Themisto (v0.1.1) with the default settings. Themisto is a *k*-mer-based pseudoalignment tool which encodes sets of *k*-mers as a succinct colored de Bruijn graph. A read is considered to pseudoalign against a reference sequence if at least one *k*-mer of the read is found in the reference, and each *k*-mer of the read is either found in the reference or not found at all in the database of all references. This can be seen as an exact version of the pseudoalignment algorithm implemented by the tool Kallisto (Bray et al. 2016).

The index was constructed using *31*-mers. Themisto does not distinguish between paired-end reads and single reads, so we decided to consider a paired-end read as pseudoaligned only when both fragments pseudoaligned. We have included this functionality for supporting paired-end reads in both the mSWEEP and mGEMS software implementations.

Abundance estimation and probabilistic read assignment

We used the mSWEEP (Mäklin et al. 2020) software (v1.3.2; doi: 10.5281/zenodo.3631062) with default settings. The program was altered to support pseudoalignments from Themisto, and to output the read-level probabilistic assignments to the reference lineages. We also improved the scalability of mSWEEP by parallelizing the abundance estimation part and reducing memory consumption. These alterations have been included in versions v1.3.2 (Themisto and mGEMS support) and v1.4.0 (parallelization and memory usage improvements) of the software.

Read binning

In order to collect all reads in a mixed sample that likely originate from the same target lineage, we consider a binning strategy that allows associating the same read with multiple reference lineages. We assume that each reference lineage is represented by, at most, only one target sequence in the mixed sample, and that the sets of reference sequences capture the variation in the reference lineages sufficiently to use them as a substitute for the target sequence which may not be included in the reference sequences. In our formal treatment of the task of binning a set of sequencing reads, we define the task in terms of finding K subsets (bins), one for each reference lineage $k = 1, ..., K$, of the full sets of reads $R = \{r_1, ..., r_N\}$ denoted by $G_k \subset R$ that contain reads likely originating from the target sequence belonging to the reference lineage k. The reads assigned to each subset G_k are determined based on read-level probabilities $\gamma_{n,k}$, $\sum_{k=1}^{K} \gamma_{n,k} = 1$, $n = 1, ..., N$ to classify the read r_n into the reference lineage k by defining the subsets G_k such that

$$G_k = \left\{ r_n : \gamma_{n,k} \geq q_k \right\},$$

Equation 1

holds for some threshold $q_k \in [0, 1]$ which may vary between the lineages k. The formulation in Equation (1) has the benefit of allowing the read r_n to possibly belong to several subsets G_k, which is an important property for dealing with multiple closely related lineages in the same mixed sample.

In order to find a suitable value for the threshold q_k, and to determine the corresponding assignment rule, we consider two binary events: 1) $I_{n,k}$: the reference lineage k generated the read r_n, and 2) $J_{n,k}$: the true nucleotide sequence represented by the read r_n is part of the target sequence belonging to the reference lineage k. Knowing the probability of the event $J_{n,k}$ would directly enable us to assess the plausibility of assigning the read r_n to the reference lineage k but its value is difficult to estimate directly. However, we can determine and write down the values of the conditional probabilities $P[I_{n,k} = 1 \mid J_{n,k} = 0]$ and $P[I_{n,k} = 1 \mid J_{n,k} = 1]$ as

$$P[I_{n,k} = 1 \mid J_{n,k} = 0] = 0, \text{ and}$$

$$P[I_{n,k} = 1 \mid J_{n,k} = 1] = \frac{\theta_k}{\sum_{c : J_{n,c} = 1} \theta_c},$$

Equation 2

where θ_k is the proportion of reads from the reference lineage k, and $\sum_{c : J_{n,c} = 1} \theta_c$ is the proportion of reads from any reference lineages $\{c : J_{n,c} = 1\}$ which contain the sequence represented by the read r_n. The conditional probabilities in Equation (2) allow us to write the unconditional probability $P[I_{n,k} = 1]$ as

$$P[I_{n,k} = 1] = P[I_{n,k} = 1 \mid J_{n,k} = 0]P[J_{n,k} = 0] + P[I_{n,k} = 1 \mid J_{n,k} = 1]P[J_{n,k} = 1]$$

$$\Leftrightarrow P[I_{n,k} = 1] = \frac{\theta_k}{\sum_{c : J_{n,c} = 1} \theta_c} P[J_{n,k} = 1].$$

Equation 3

Using the formulation in Equation (3) and the fact that we can approximate

$\frac{\theta_k}{\sum_{c\,:\,J_{n,c}\,=\,1} \theta_c} \approx \theta_k$ if we assume that the mixed sample is mostly composed of closely related organisms (the denominator $\sum_{c\,:\,J_{n,c}\,=\,1} \theta_c$ approaches 1), we can rewrite Equation (3) as

$$P[I_{n,\,k} = 1] \approx \theta_k P[J_{n,\,k} = 1].$$

Equation 4

Equations (4) and (3) together imply that if the value of the probability $P[I_{n,\,k} = 1]$ that the read r_n was generated from the lineage k exceeds the relative abundance θ_k of that lineage in whole sample ($P[I_{n,\,k} = 1] \geq \theta_k$), then the value of the probability $P[J_{n,\,k} = 1]$ that the nucleotide sequence represented by the read r_n is contained in the target sequence from the reference lineage k must be "large" ($P[J_{n,\,k} = 1] \rightarrow 1$). This statement about the magnitude of $P[J_{n,\,k} = 1]$ derives from our assumption that the denominator in Equation (3) is close to 1.

Since we have an estimate of the probabilities $P[I_{n,\,k} = 1]$ available in the form of the read-level probabilistic assignments $\gamma_{n,\,k} \approx P[I_{n,\,k} = 1]$, we can plug these values in Equation (4) and use the result to derive the assignment rule

$$\text{if } \gamma_{n,\,k} \geq \theta_k \text{, assign the read } r_n \text{ to } G_k.$$

Equation 5

The assignment rule in Equation (5) gives us a way to assess the validity of the statement contained in the probability $P[J_{n,\,k} = 1]$ which we could not estimate directly.

Because of computational accuracy, we cannot obtain meaningful relative abundance estimates θ_k for lineages with a relative abundance less than $\frac{1}{N}$ (less than one read from the lineage k in the sample). Since there are K lineages in total, in the worst-case scenario $K\frac{1}{N}$ units of the relative abundance fall into this meaningless range. Therefore only a fraction of the total relative abundance of 1 can be considered to be accurately determined when using computed values of θ_k, and this fraction d is determined in the worst-case scenario through the formula

$$d = 1 - K\frac{1}{N}.$$

Equation 6

Equation (6) means that when evaluating the validity of the assignment rule presented in Equation (5) with computed values, we have to replace θ_k with the value $d\theta_k$ which depends on the value of d in Equation (6). Merging the result from Equations (5) and (6) leads us to the final assignment rule (Equation 7) of

$$\text{if } \gamma_{n,k} > d\theta_k \text{, assign the read } r_n \text{ to } G_k.$$

Equation 7

In practice, reads which pseudoalign to exactly the same reference sequences have identical values $\gamma_{n,k}$. The reads can thus be assigned to equivalence classes defined by their pseudoalignments, which enables a speedup in the implementation of the binning algorithm by considering each equivalence class as a single read. Due to this speedup and the computational simplicity of evaluating the assignment rule in Equation (7), the memory footprint of the mGEMS binner is determined by the number of equivalence classes and reference lineages in the input pseudoalignment and the runtime limited by disk I/O performance.

Genome assembly

After binning the sequencing reads in our experiments with the aforementioned assignment rule, we assembled the sequencing reads assigned to the bins using the shovill (version 0.9.0, default settings) (Seemann 2018) assembly optimizer for the SPAdes assembler (Bankevich et al. 2012; Nurk et al. 2013). This step concludes what we in this article call the mGEMS pipeline.

SNP calling and phylogeny reconstruction

We used snippy (version 4.4.5) (Seemann 2014) to produce a core SNP multiple-sequence alignment from the assembled contigs. Since the *E. coli* and *S. aureus* strains used were from the same sequence type, the core alignment for these two species contained almost the whole genome. After running snippy, RAxML-NG (version 0.8.1) (Kozlov et al. 2019) was used to infer the maximum-likelihood phylogeny from the alignment. Since some of the *S. aureus* strains from the same clade were identical, we changed the default value of the minimum branch length parameter in RAxML-NG to 10^{-10} in the *S. aureus* experiments and printed the branch length with eight decimal precision to identify branches of length zero. In all experiments, we ran RAxML-NG with 100 random and 100 maximum parsimony starting trees, and performed 1000 bootstrapping iterations to infer Felsenstein bootstrap support values for the branches. We used the phytools R package (v0.6-99) (Revell 2012) to perform midpoint rooting for the tree, and the ape R package (v5.3) (Paradis and Schliep 2019) to create the visualizations.

Data Access

Source code and precompiled binaries (generic Linux and macOS) for both mGEMS and Themisto are freely available in GitHub at https://github.com/PROBIC/mGEMS (MIT license) and at https://github.com/algbio/themisto (GPLv2.0 license). A tutorial describing how to reproduce the synthetic mixed samples, bin the mixed reads, and infer the phylogenies is available in the mGEMS GitHub repository. The reference data used is available from Zenodo (*E. coli* doi: 10.5281/zenodo.3724111, *E. faecalis* doi: 10.5281/zenodo.3724101, *S. aureus* doi: 10.5281/zenodo.3724135).

Acknowledgements

TM and AH were supported by the Academy of Finland grant no. 310261 as well as the Flagship programme (Finnish Center for Artificial Intelligence FCAI; to JC and AH). TK and JC were supported by the JPI-AMR consortium SpARK (MR/R00241X/1). JC was funded by the ERC grant no. 742158. TK was funded by the Norwegian Research Council JPIAMR grant no. 144501. JA and VM were supported by the Academy of Finland grant no. 309048. The authors wish to acknowledge CSC – IT Center for Science, Finland, and the Finnish Grid and Cloud Infrastructure (persistent identifier urn:nbn:fi:research-infras-2016072533) for providing computational resources.

Author Contributions

TM, TK, JC and AH conceived the study, developed the full mGEMS pipeline, and designed the benchmarking experiments. TM and AH developed the mGEMS binning algorithm. JA and VM developed the Themisto pseudoaligner. TM implemented the mGEMS binner. JA implemented the Themisto pseudoaligner. TM ran the experiments and created the visualizations. TM, TK, JC and AH interpreted the results and wrote the main article. JA wrote the supplementary file describing Themisto. All authors participated in reviewing and editing the article and discussed the results.

Disclosure Declaration

The authors declare that they have no competing interests.

References

Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, Lesin VM, Nikolenko SI, Pham S, Prjibelski AD, et al. 2012. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *J Comput Biol* **19**: 455–477.

Bray NL, Pimentel H, Melsted P, Pachter L. 2016. Near-optimal probabilistic RNA-seq quantification. *Nat Biotechnol* **34**: 525–527.

Breitwieser FP, Lu J, Salzberg SL. 2019. A review of methods and databases for metagenomic classification and assembly. *Brief Bioinform* **20**: 1125–1136.

Brodrick HJ, Raven KE, Harrison EM, Blane B, Reuter S, Török ME, Parkhill J, Peacock SJ. 2016. Whole-genome sequencing reveals transmission of vancomycin-resistant Enterococcus faecium in a healthcare network. *Genome Med* **8**: 4.

Brodrick HJ, Raven KE, Kallonen T, Jamrozy D, Blane B, Brown NM, Martin V, Török ME, Parkhill J, Peacock SJ. 2017. Longitudinal genomic surveillance of multidrug-resistant Escherichia coli carriage in a long-term care facility in the United Kingdom. *Genome Med* **9**: 70.

Bürkner P-C. 2018. Advanced Bayesian Multilevel Modeling with the R Package brms. *R J* **10**: 395–411.

Bürkner P-C. 2017. brms: An R Package for Bayesian Multilevel Models Using Stan. *J Stat Softw* **80**: 1–28.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A. 2017. Stan: A Probabilistic Programming Language. *J Stat Softw* **76**: 1–32.

Deng X, den Bakker HC, Hendriksen RS. 2016. Genomic Epidemiology: Whole-Genome-Sequencing-Powered Surveillance and Outbreak Investigation of Foodborne Bacterial Pathogens. *Annu Rev Food Sci Technol* **7**: 353–374.

Ellegaard KM, Engel P. 2016. Beyond 16S rRNA Community Profiling: Intra-Species Diversity in the Gut Microbiota. *Front Microbiol* **7**. https://www.frontiersin.org/articles/10.3389/fmicb.2016.01475/full (Accessed February 17, 2020).

Forbes JD, Knox NC, Ronholm J, Pagotto F, Reimer A. 2017. Metagenomics: The Next Culture-Independent Game Changer. *Front Microbiol* **8**. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5495826/ (Accessed February 20, 2020).

Forster SC, Kumar N, Anonye BO, Almeida A, Viciani E, Stares MD, Dunn M, Mkandawire TT, Zhu A, Shao Y, et al. 2019. A human gut bacterial genome and culture collection for improved metagenomic analyses. *Nat Biotechnol* **37**: 186–192.

Goig GA, Blanco S, Garcia-Basteiro AL, Comas I. 2020. Contaminant DNA in bacterial sequencing experiments is a major source of false genetic variability. *BMC Biol* **18**: 24.

Golubchik T, Batty EM, Miller RR, Farr H, Young BC, Larner-Svensson H, Fung R, Godwin H, Knox K, Votintseva A, et al. 2013. Within-Host Evolution of Staphylococcus aureus during Asymptomatic Carriage. *PLoS ONE* **8**. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3641031/ (Accessed February 24, 2020).

Grad YH, Lipsitch M. 2014. Epidemiologic data and pathogen genome sequences: a powerful synergy for public health. *Genome Biol* **15**: 538.

Greenblum S, Carr R, Borenstein E. 2015. Extensive Strain-Level Copy-Number Variation across Human Gut Microbiome Species. *Cell* **160**: 583–594.

Gu W, Miller S, Chiu CY. 2019. Clinical Metagenomic Next-Generation Sequencing for Pathogen Detection. *Annu Rev Pathol Mech Dis* **14**: 319–338.

Harris SR. 2018. SKA: Split Kmer Analysis Toolkit for Bacterial Genomic Epidemiology. *bioRxiv* 453142.

Holden MTG, Hsu L-Y, Kurt K, Weinert LA, Mather AE, Harris SR, Strommenger B, Layer F, Witte W, Lencastre H de, et al. 2013. A genomic portrait of the emergence, evolution, and global spread of a methicillin-resistant Staphylococcus aureus pandemic. *Genome Res* **23**: 653–664.

Ito T, Sekizuka T, Kishi N, Yamashita A, Kuroda M. 2019. Conventional culture methods with commercially available media unveil the presence of novel culturable bacteria. *Gut Microbes* **10**: 77–91.

Ivy MI, Thoendel MJ, Jeraldo PR, Greenwood-Quaintance KE, Hanssen AD, Abdel MP, Chia N, Yao JZ, Tande AJ, Mandrekar JN, et al. 2018. Direct Detection and Identification of Prosthetic Joint Infection Pathogens in Synovial Fluid by Metagenomic Shotgun Sequencing. *J Clin Microbiol* **56**. https://jcm.asm.org/content/56/9/e00402-18 (Accessed February 17, 2020).

Jolley KA, Bray JE, Maiden MCJ. 2018. Open-access bacterial population genomics: BIGSdb software, the PubMLST.org website and their applications. *Wellcome Open Res* **3**: 124.

Kallonen T, Brodrick HJ, Harris SR, Corander J, Brown NM, Martin V, Peacock SJ, Parkhill J. 2017. Systematic longitudinal survey of invasive Escherichia coli in England demonstrates a stable population structure only transiently disturbed by the emergence of ST131. *Genome Res*.

Kang DD, Li F, Kirton E, Thomas A, Egan R, An H, Wang Z. 2019. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* **7**: e7359.

Kozlov AM, Darriba D, Flouri T, Morel B, Stamatakis A. 2019. RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics* **35**: 4453–4455.

Kwong JC, Mccallum N, Sintchenko V, Howden BP. 2015. Whole genome sequencing in clinical and public health microbiology. *Pathology (Phila)* **47**: 199–210.

Lagier J-C, Edouard S, Pagnier I, Mediannikov O, Drancourt M, Raoult D. 2015. Current and Past Strategies for Bacterial Culture in Clinical Microbiology. *Clin Microbiol Rev* **28**: 208–236.

Li D, Liu C-M, Luo R, Sadakane K, Lam T-W. 2015. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* **31**: 1674–1676.

Li D, Luo R, Liu C-M, Leung C-M, Ting H-F, Sadakane K, Yamashita H, Lam T-W. 2016. MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods* **102**: 3–11.

Lieberman TD, Flett KB, Yelin I, Martin TR, McAdam AJ, Priebe GP, Kishony R. 2014. Genetic variation of a bacterial pathogen within individuals with cystic fibrosis provides a record of selective pressures. *Nat Genet* **46**: 82–87.

Maio ND, Worby CJ, Wilson DJ, Stoesser N. 2018. Bayesian reconstruction of transmission within outbreaks using genomic variants. *PLOS Comput Biol* **14**: e1006117.

Mäklin T, Kallonen T, David S, Boinett CJ, Pascoe B, Méric G, Aanensen DM, Feil EJ, Baker S, Parkhill J, et al. 2020. High-resolution sweep metagenomics using fast probabilistic inference. *Wellcome Open Res* **5**: 14.

McArdle AJ, Kaforou M. 2020. Sensitivity of shotgun metagenomics to host DNA: abundance estimates depend on bioinformatic tools and contamination is the main issue. *Access Microbiol*. https://www.microbiologyresearch.org/content/journal/acmi/10.1099/acmi.0.000104 (Accessed February 24, 2020).

McIntyre ABR, Ounit R, Afshinnekoo E, Prill RJ, Hénaff E, Alexander N, Minot SS, Danko D, Foox J, Ahsanuddin S, et al. 2017. Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biol* **18**: 182.

Meyer F, Hofmann P, Belmann P, Garrido-Oter R, Fritz A, Sczyrba A, McHardy AC. 2018. AMBER: Assessment of Metagenome BinnERs. *GigaScience* **7**. https://academic.oup.com/gigascience/article/7/6/giy069/5034950 (Accessed February 23, 2020).

Mikheenko A, Saveliev V, Gurevich A. 2016. MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics* **32**: 1088–1090.

Nurk S, Bankevich A, Antipov D, Gurevich A, Korobeynikov A, Lapidus A, Prjibelsky A, Pyshkin A, Sirotkin A, Sirotkin Y, et al. 2013. Assembling Genomes and Mini-metagenomes from Highly Chimeric Reads. In *Research in Computational Molecular Biology* (eds. M. Deng, R. Jiang, F. Sun, and X. Zhang), *Lecture Notes in Computer Science*, pp. 158–170, Springer, Berlin, Heidelberg.

Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. 2017. metaSPAdes: a new versatile metagenomic assembler. *Genome Res* **27**: 824–834.

Paradis E, Schliep K. 2019. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* **35**: 526–528.

Paterson GK, Harrison EM, Murray GGR, Welch JJ, Warland JH, Holden MTG, Morgan FJE, Ba X, Koop G, Harris SR, et al. 2015. Capturing the cloud of diversity reveals complexity and heterogeneity of MRSA carriage, infection and transmission. *Nat Commun* **6**: 6560.

Paulsen IT, Banerjei L, Myers GSA, Nelson KE, Seshadri R, Read TD, Fouts DE, Eisen JA, Gill SR, Heidelberg JF, et al. 2003. Role of Mobile DNA in the Evolution of Vancomycin-Resistant Enterococcus faecalis. *Science* **299**: 2071–2074.

Peng Y, Leung HCM, Yiu SM, Chin FYL. 2012. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics* **28**: 1420–1428.

Quince C, Walker AW, Simpson JT, Loman NJ, Segata N. 2017. Shotgun metagenomics, from sampling to analysis. *Nat Biotechnol* **35**: 833–844.

Raven KE, Reuter S, Gouliouris T, Reynolds R, Russell JE, Brown NM, Török ME, Parkhill J, Peacock SJ. 2016. Genome-based characterization of hospital-adapted Enterococcus faecalis lineages. *Nat Microbiol* **1**: 15033.

Revell LJ. 2012. phytools: an R package for phylogenetic comparative biology (and other things). *Methods Ecol Evol* **3**: 217–223.

Rossen JWA, Friedrich AW, Moran-Gilad J. 2018. Practical issues in implementing whole-genome-sequencing in routine diagnostic microbiology. *Clin Microbiol Infect* **24**: 355–360.

Ruiz-Garbajosa P, Bonten MJM, Robinson DA, Top J, Nallapareddy SR, Torres C, Coque TM, Cantón R, Baquero F, Murray BE, et al. 2006. Multilocus Sequence Typing Scheme for Enterococcus faecalis Reveals Hospital-Adapted Genetic Complexes in a Background of High Rates of Recombination. *J Clin Microbiol* **44**: 2220–2228.

Salter SJ, Cox MJ, Turek EM, Calus ST, Cookson WO, Moffatt MF, Turner P, Parkhill J, Loman NJ, Walker AW. 2014. Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biol* **12**: 87.

Scholz M, Ward DV, Pasolli E, Tolio T, Zolfo M, Asnicar F, Truong DT, Tett A, Morrow AL, Segata N. 2016. Strain-level microbial epidemiology and population genomics from shotgun metagenomics. *Nat Methods* **13**: 435–438.

Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Dröge J, Gregor I, Majda S, Fiedler J, Dahms E, et al. 2017. Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nat Methods* **14**: 1063–1071.

Seemann T. 2015. *mlst*. GitHub https://github.com/tseemann/mlst.

Seemann T. 2018. *shovill*. GitHub https://github.com/tseemann/shovill.

Seemann T. 2014. *snippy: fast bacterial variant calling from NGS reads*. GitHub https://github.com/tseemann/snippy.

Sieber CMK, Probst AJ, Sharrar A, Thomas BC, Hess M, Tringe SG, Banfield JF. 2018. Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. *Nat Microbiol* **3**: 836–843.

Skums P, Zelikovsky A, Singh R, Gussler W, Dimitrova Z, Knyazev S, Mandric I, Ramachandran S, Campo D, Jha D, et al. 2018. QUENTIN: reconstruction of disease transmissions from viral quasispecies genomic data. *Bioinformatics* **34**: 163–170.

Stewart EJ. 2012. Growing Unculturable Bacteria. *J Bacteriol* **194**: 4151–4160.

Stoesser N, Sheppard AE, Moore CE, Golubchik T, Parry CM, Nget P, Saroeun M, Day NPJ, Giess A, Johnson JR, et al. 2015. Extensive Within-Host Diversity in Fecally Carried Extended-Spectrum-Beta-Lactamase-Producing Escherichia coli Isolates: Implications for Transmission Analyses. *J Clin Microbiol* **53**: 2122–2131.

Tang P, Croxen MA, Hasan MR, Hsiao WWL, Hoang LM. 2017. Infection control in the new age of genomic epidemiology. *Am J Infect Control* **45**: 170–179.

The Human Microbiome Project Consortium. 2012. Structure, function and diversity of the healthy human microbiome. *Nature* **486**: 207–214.

Truong DT, Tett A, Pasolli E, Huttenhower C, Segata N. 2017. Microbial strain-level population structure and genetic diversity from metagenomes. *Genome Res* **27**: 626–638.

Van Goethem N, Descamps T, Devleesschauwer B, Roosens NHC, Boon NAM, Van Oyen H, Robert A. 2019. Status and potential of bacterial genomics for public health practice: a scoping review. *Implement Sci* **14**: 79.

Vartoukian SR, Palmer RM, Wade WG. 2010. Strategies for culture of 'unculturable' bacteria. *FEMS Microbiol Lett* **309**: 1–7.

Vollmers J, Wiegand S, Kaster A-K. 2017. Comparing and Evaluating Metagenome Assembly Tools from a Microbiologist's Perspective - Not Only Size Matters! ed. F. Rodriguez-Valera. *PLOS ONE* **12**: e0169662.

Whelan FJ, Waddell B, Syed SA, Shekarriz S, Rabin HR, Parkins MD, Surette MG. 2020. Culture-enriched metagenomic sequencing enables in-depth profiling of the cystic fibrosis lung microbiota. *Nat Microbiol* 1–12.

Worby CJ, Lipsitch M, Hanage WP. 2017. Shared Genomic Variants: Identification of Transmission Routes Using Pathogen Deep-Sequence Data. *Am J Epidemiol* **186**: 1209–1216.

Worby CJ, Lipsitch M, Hanage WP. 2014. Within-Host Bacterial Diversity Hinders Accurate Reconstruction of Transmission Networks from Genomic Distance Data. *PLoS Comput Biol* **10**. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3967931/ (Accessed February 24, 2020).

Wu Y-W, Simmons BA, Singer SW. 2016. MaxBin 2.0: an automated binning algorithm to recover genomes from multiple metagenomic datasets. *Bioinformatics* **32**: 605–607.

Zou Y, Xue W, Luo G, Deng Z, Qin P, Guo R, Sun H, Xia Y, Liang S, Dai Y, et al. 2019. 1,520 reference genomes from cultivated human gut bacteria enable functional microbiome analyses. *Nat Biotechnol* **37**: 179–185.

Pseudoalignment in the mGEMS pipeline

Tommi Mäklin Teemu Kallonen Jarno Alanko
Veli Mäkinen Jukka Corander Antti Honkela

In this supplement we describe the pseudoalignment algorithm and implementation used in the mGEMS pipeline. The implementation is called Themisto, and is freely available at https://github.com/algbio/themisto under the GPLv2.0 license. Pseudoalignment is an approximate form of alignment that reports only whether a read matches to a reference sequence or not, without necessarily returning the genomic coordinates of the match. Pseudoalignment can be much cheaper computationally than regular alignment.

1 The pseudoalignment criterion

Our pseudoalignment is based on the pseudoalignment algorithm used in the transcript abundance quantification tool Kallisto [1]. The pseudoalignment criterion we use is defined as follows. Suppose we want to pseudoalign a read against a set of reference sequences $T_1, \ldots, T_m$. The read is considered to pseudoalign against reference T_i if at least one k-mer of the read is found in T_i and for each k-mer x of the read, one of the following holds:

1. x is a k-mer of T_i
2. x is not a k-mer of any of $T_1, \ldots, T_m$

This criterion closely replicates the pseudoalignment of Kallisto, with the difference that Kallisto uses a heuristic based on the topology of the de Bruijn graph of $T_1, \ldots, T_m$ to skip over some k-mers of the read for efficiency. More specifically, if the current k-mer is in a non-branching path of the graph, Kallisto skips a number of k-mers of the read equal to the distance to reach the next branching node. If the k-mers before and after the skip are found in the same reference sequences, the skip is considered valid, and otherwise Kallisto falls back to checking all k-mers of the read individually. However, even if the skip is considered valid, it could be the case that a skipped k-mer would have affected the result of the pseudoalignment. On the other hand, we implement the described pseudoalignment criterion exactly, and observe a very slight improvement in accuracy compared to using Kallisto's pseudoalignments. The difference in accuracy could also be due to small implementation differences, since we designed our tool around the high-level description in Kallisto's manuscript [1] rather than the source code itself.

2 Implementation overview

The pseudoalignment criterion we have chosen effectively reduces each reference sequence and each read into unordereded sets of k-mers. This loses some information, but in turn it allows for more efficient data structures and algorithms. The pseudoalignment could in principle be implemented on top of any data structure for indexing k-mer sets.

Indexing k-mer sets efficiently is currently a very active field of research [2]. In k-mer data structures, each reference sequence is usually given a unique identifier, called the *color* of the sequence. Each k-mer is associated with a *color set*, which is defined to be the set of colors of the reference sequences that contain that k-mer. The basic query on a k-mer data structure is to retrieve the color set of a given k-mer. Our pseudoalignment criterion can be computed against all references at once by intersecting the non-empty color sets of all k-mers in a read.

We chose to implement our own k-mer index. The main design goal was that the index should be memory-efficient to build and use, because the size of the reference dataset can be large. To this end, we index the k-mer sets as a *succinct colored de Bruijn graph*. The nodes of the graph represent k-mers and the edges represent $(k + 1)$-mers. The graph is encoded with a variant of the BOSS representation [3] and each node is linked to the corresponding color set with a separate coloring data structure which is unique to our implementation. Each query read is aligned as both the reverse complement string and the forward string, and we return the union of the pseudoalignments of both directions. Figure 1 illustrates the approach.

A speciality of our implementation is that the construction can be done almost entirely on disk, using only a minimal amount of RAM. This is made possible by designing the construction pipeline around two well-studied primitive operations: k-mer counting and disk-based sorting. The next section gives the technical details of the index and the construction pipeline.

3 Implementation details

The reference sequences are modeled as strings from an alphabet Σ of size σ (for DNA, $\Sigma = \{A,C,G,T\}$ and $\sigma = 4$). Let us denote the set of references with $T_1, \ldots, T_m$. First, we build the BOSS data structure of the de Bruijn graph, implemented in terms of the generic Wheeler graph framework introduced by Gagie et al. [4].

Let $T = T_1\$T_2\$ \cdots T_m\$$ be a dollar-separated concatenation of the reference sequences, where the dollar is a special symbol such that $\$ \notin \Sigma$. Let $f_\ell(x)$ be

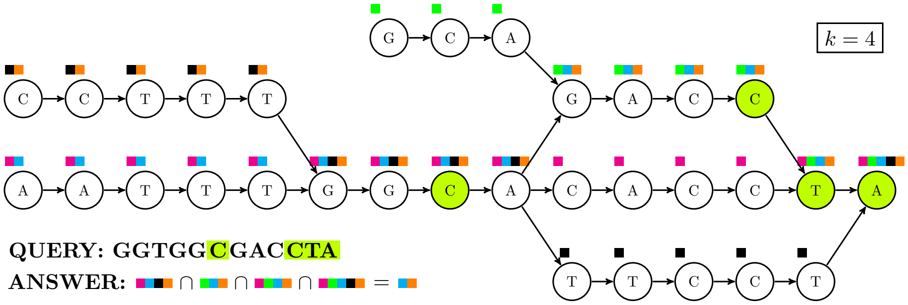


Figure 1: A colored de Bruijn graph of order $k = 4$. Each reference sequence in the graph is assigned a unique color. The color sets of nodes are drawn above the nodes. In the example, the query GGTGGCTGACCTA is pseudoaligned against the graph. Four of the k-mers of the query are found in the graph. The representative nodes of those k-mers are highlighted with green. The pseudoalignment returns the intersection of the color sets of the highlighted nodes.

the set of distinct characters that are found to the left of k-mer x in T and let $f_r(x)$ be the set of distinct characters that are found to the right of x in T.

To build the Wheeler graph data structure, we iterate the sets $f_\ell(x)$ and $f_r(x)$ in colexicographic[1] order of the k-mers x of T. To do this, we first list all distinct $(k+2)$-mers of T to disk. Then we sort the $(k+2)$-mers x in increasing order of the colexicographic rank of the middle k-mer $x[2..(k+1)]$ using a disk-based sorting algorithm. Next, we stream the sorted $(k+2)$-mers from disk. For every run of $(k+2)$-mers with an identical middle k-mer y, we collect the sets $f_\ell(y)$ and $f_r(y)$ by looking at the first and last characters of the $(k+2)$-mers in the run. Building the Wheeler graph data structure is straightforward from this information.

After this, we have a working index of the de Bruijn graph (V, E) of the references $T_1, \ldots, T_m$. If $(u, v) \in E$, we call u a predecessor of v, and v a successor of u. Next, we add the colors to nodes of the graph. To eliminate redundancy, we only store colors for a subset $V' \subseteq V$, where $v \in V'$ iff at least one of the following conditions hold:

1. Node v represents the first k-mer of a reference sequence.
2. A predecessor of v represents the last k-mer of a reference sequence.
3. Node v has multiple predecessors
4. Node v has a predecessor that has multiple successors.

[1] The colexicographic order of strings is like the standard lexicographic order, but characters are compared starting starting from the end. The index can be build with either lexicographic or colexicographic sorting, but we choose to follow the colexicographic convention of the Wheeler graph framework. The indexed graph can be traversed in both directions.

If $v \notin V'$, then its color set has to be the same as its predecessor's color set. We can find out the color set of v by walking backward to the nearest node $u \in V'$. Node u is guaranteed to exist because the first node of every reference sequence is always in V'. The nodes in V' can be found and marked by using the BOSS index.

However, with this setup, finding node u might take a long time if we are in the middle of a long unitig (non-branching path), so we also store the color sets for some nodes inside long unitigs. Let S be the set of nodes such that the distance backward to the nearest node in V' is an integer multiple of s for some global integer parameter s. We also store the color sets for all nodes in S. This way, we can find a color set of a node in at most s backward steps. The sampling parameter s can be tuned to obtain different time-space tradeoffs.

The color sets are computed with two disk-based sortings as follows. Assume we have marked all nodes in $V' \cup S$. Assign the reference sequences $T_1, \ldots, T_m$ colors such that the color of sequence T_i is i. For each $i = 1, \ldots, m$, walk the de Bruijn graph according to T_i using the constructed BOSS index, and for each node $v \in V' \cup S$ encountered, print to disk a pair (v, i). After all sequences T_i have been processed, sort the pairs on disk by the node identifiers v, and scan the sorted list, writing to another file pairs (v, C_v), where C_v is the list of colors of node v. Then, sort the new pairs by the color sets and scan the resulting sorted list to obtain a list of pairs (X_v, C_v), where X_v is the set of nodes with color set C_v. Finally, store all distinct color sets to a file, and for each node in the sets X_v, store a pointer to the corresponding color set.

It remains to be described how the color sets are stored in a succinct and accessible way. Let us denote the set of distinct color sets with $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$. The color sets are stored in a concatenated form $C_1 \cdots C_{|C|}$. We mark with a bit vector all positions in the concatenation where a new color set starts, and index the bit vector for constant-time select queries to be able to locate the i-th distinct color set in constant time. A pointer to color set C_i is just the integer i, which can be represented in $\lceil \log |\mathcal{C}| \rceil$ bits. By choosing the sampling parameter $s = \lceil \log |\mathcal{C}| \rceil$, the size of S is at most $|V| / \log |\mathcal{C}|$, so the total size taken by the sampling pointers is only $|V|$ bits, and we obtain a worst-case color set lookup time of $O(\log s) = O(\log |\mathcal{C}|)$. With this, the whole coloring data structure takes on the order of $|V'| \log |\mathcal{C}| + |V| + \sum_{C \in \mathcal{C}} |C| \log m$ bits of space. The Wheeler graph data-structure takes $|V| \log \sigma + 2|V| + \sigma \log |V| + o(|V| \log \sigma)$ bits space, where σ is the size of the alphabet.

Most of the heavy work is done by the subroutines for k-mer listing and for disk-based sorting. In our implementation, we used the highly optimized parallel tool KMC3 for k-mer listing, and a custom ℓ-way disk-streaming mergesort with parallel merges for the sorting. The sorting implementation first divides the input into blocks that fit in the given RAM limit, sorts the blocks in RAM to disk, and then merges the blocks. Extra memory can speed up the sorting.

Any general purpose tool for the sorting and k-mer listing subroutines could be plugged into the pipeline with no changes to the rest of the pipeline. We believe this property could allow our construction pipeline to scale even to a distributed cluster of machines, as there are distributed implementations for both k-mer counting and sorting.

4 Performance

We benchmarked the construction performance of our implementation on a dataset of 3682 *Escherichia coli* genomes downloaded from the NCBI archives[2]. There were 19.0 billion nucleotides in this dataset.

Given 20GiB of RAM, Themisto builds the *E. coli* index for $k = 32$ in 6 hours and 16 minutes[3]. The main drawback is that the construction takes 375 GiB of disk space. Large disk usage is a common problem with sorting-based de Bruijn graph construction algorithms, such as the VARI-merge construction algorithm [5].

The final size of our index was 7.8GiB. The BOSS component of the index takes only 364MiB, and the rest of the space is taken by the coloring data structure. The concatenation of distinct color sets takes 6.6GiB of space. The distribution of the sizes of the color sets is shown in Figure 2. The index contains 325 million distinct k-mers.

Our implementation pseudoaligns reads from *E. coli* strains collected from across England [6] against the index at a rate of 1.4 billion nucleotides per hour using 8 threads, after loading the index into memory in 33 seconds. The alignment speed depends on the number matching k-mers and sizes of the color sets of the k-mers.

In comparison, Kallisto takes 4 hours and 57 minutes to construct an index for the same dataset, requiring as much as 287 GiB of memory. The index size on disk is 83 GiB, and 128 GiB in memory. The pseudoalignment throughput is approximately 2.1 billion nucleotides per hour using 8 threads, after loading the index to memory in 28 minutes. Table 1 summarizes key performance metrics for both Kallisto and Themisto on our benchmark.

	Index in disk	Index in RAM	Indexing time	Indexing RAM	Indexing disk	Pseudoalignment throughput
Themisto	7.8GiB	7.8GiB	6h 16min	20GiB	375GiB	$(1.4 \cdot 10^9)$ nt/h
Kallisto	83GiB	142GiB	4h 57min	287GiB	-	$(2.1 \cdot 10^9)$ nt/h

Table 1: Themisto versus Kallisto on our benchmark dataset. The unit of throughput is nucleotides per hour.

[2] Assemblies from ftp://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/assembly_summary.txt with the organism name "*Escherichia coli*".

[3] Hardware: Intel Xeon E7-8890 CPU (2.2GHz, 60M Cache, 9.6GT/s QPI 24C/48T, HT, Turbo 165W) with 48 × 64GB LRDIMM memory (2400MT/s, Quad Rank, x4 Data Width), running on top of a distributed Lustre file system.

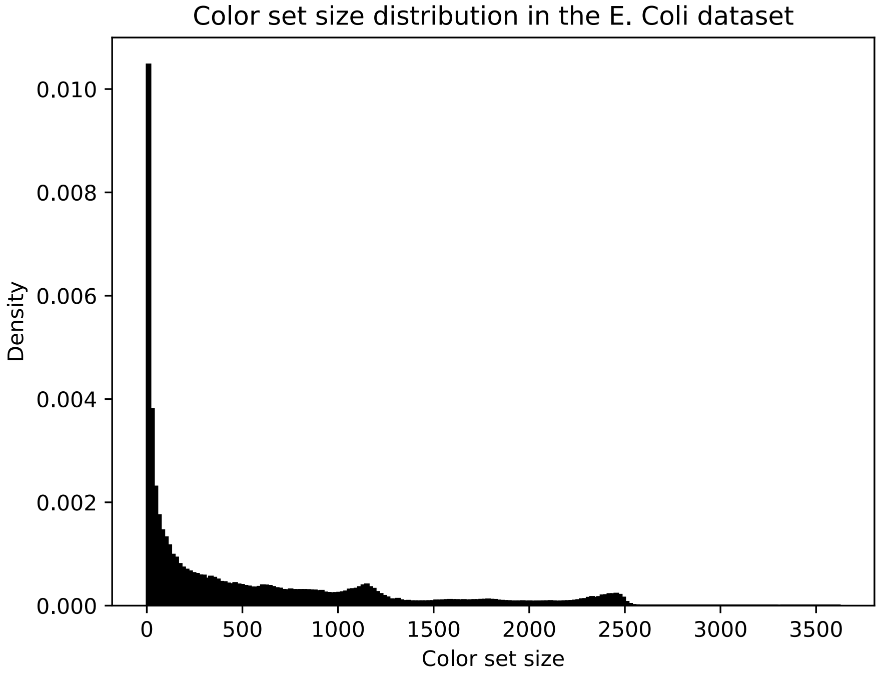


Figure 2: Color set size distribution for the dataset of 3682 E. Coli genomes each having a unique color.

References

[1] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature biotechnology*, 34(5):525–527, 2016.

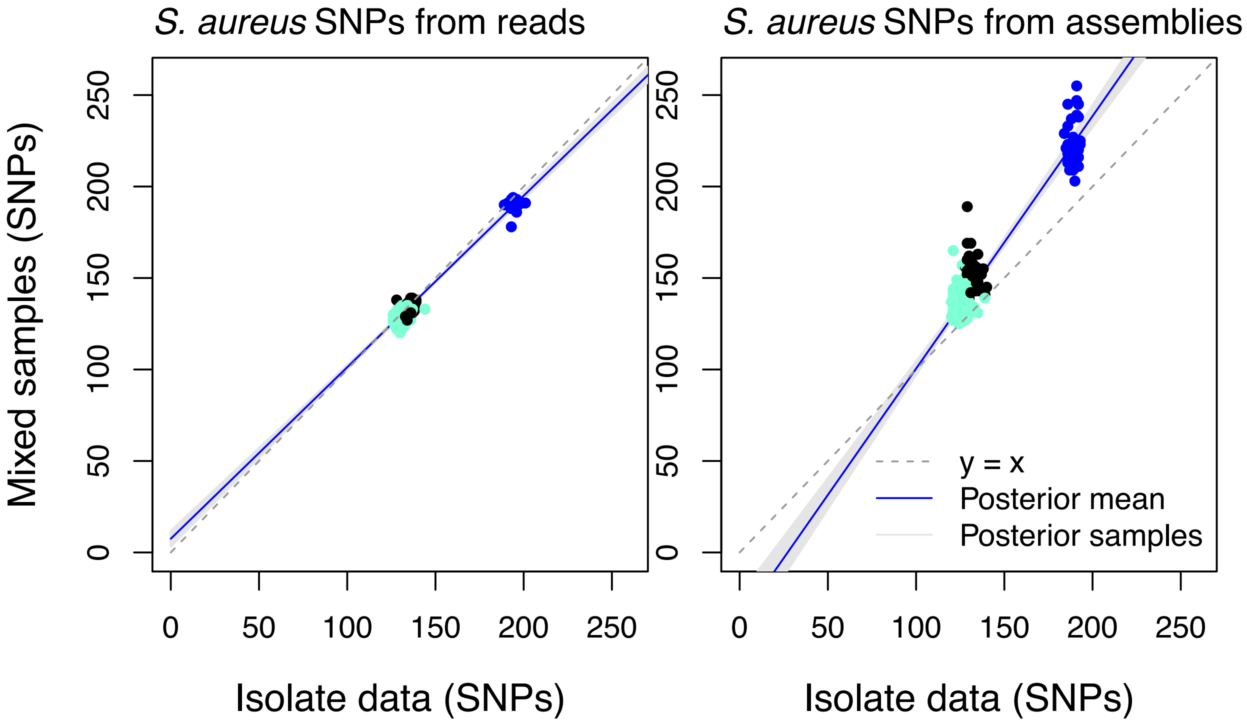
[2] Camille Marchet, Christina Boucher, Simon J Puglisi, Paul Medvedev, Mikaël Salson, and Rayan Chikhi. Data structures based on k-mers for querying large collections of sequencing datasets. *bioRxiv*, page 866756, 2019.

[3] Alexander Bowe, Taku Onodera, Kunihiko Sadakane, and Tetsuo Shibuya. Succinct de Bruijn graphs. In *International workshop on algorithms in bioinformatics*, pages 225–235. Springer, 2012.

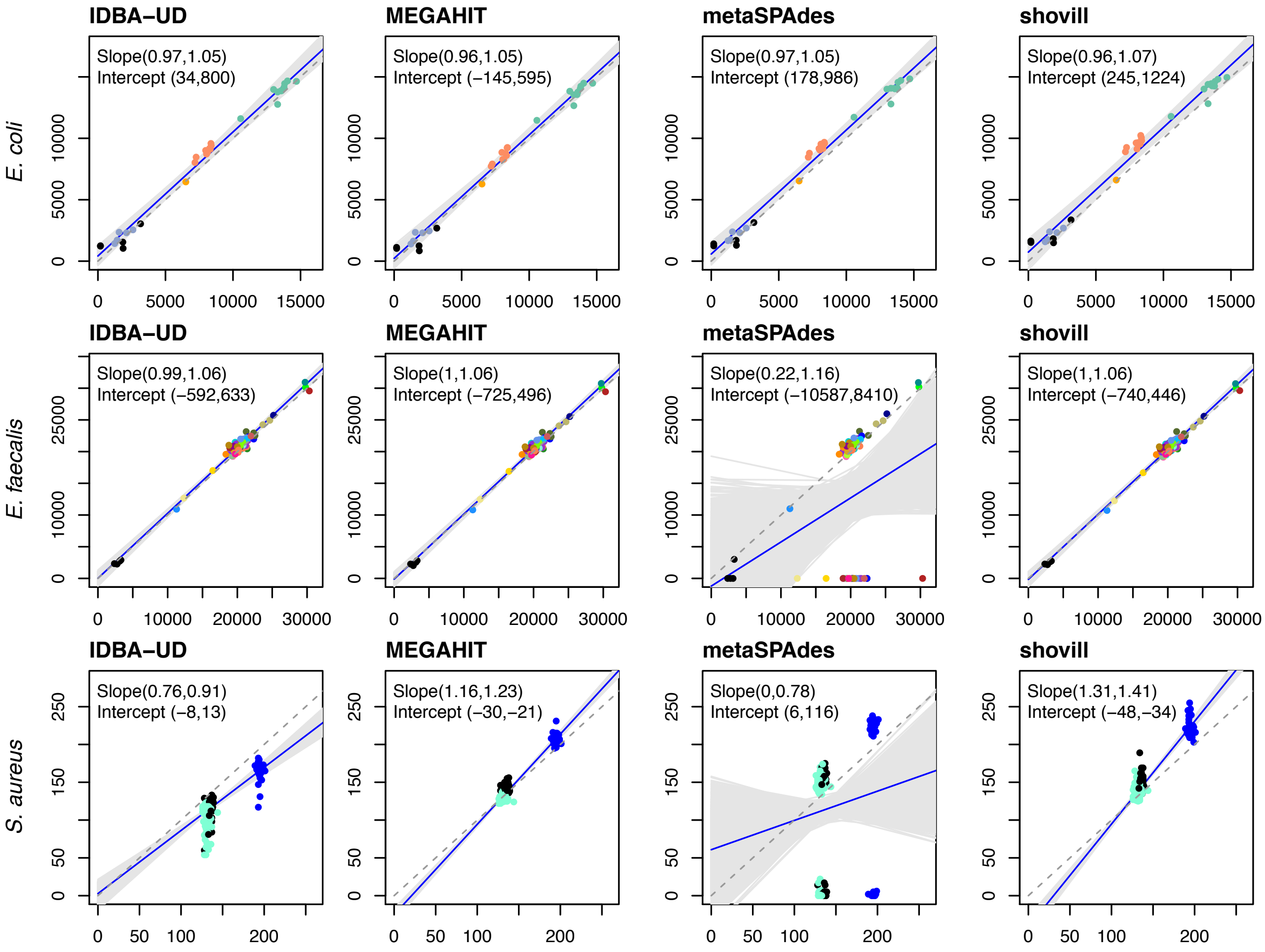
[4] Travis Gagie, Giovanni Manzini, and Jouni Sirén. Wheeler graphs: A framework for BWT-based data structures. *Theoretical computer science*, 698:67–78, 2017.

[5] Martin D Muggli, Bahar Alipanahi, and Christina Boucher. Building large updatable colored de Bruijn graphs via merging. *Bioinformatics*, 35(14):i51–i60, 2019.

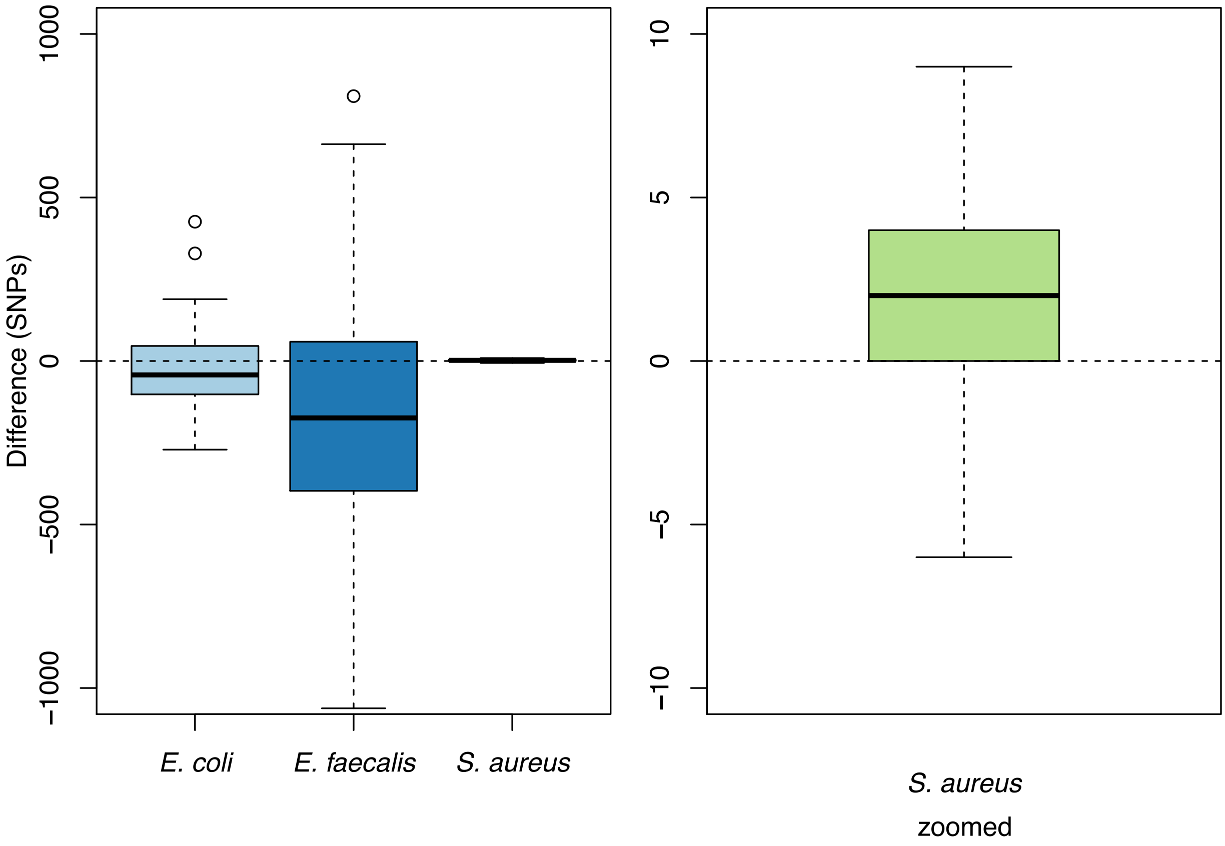
[6] Teemu Kallonen, Hayley J Brodrick, Simon R Harris, Jukka Corander, Nicholas M Brown, Veronique Martin, Sharon J Peacock, and Julian Parkhill. Systematic longitudinal survey of invasive Escherichia coli in England demonstrates a stable population structure only transiently disturbed by the emergence of ST131. *Genome research*, 27(8):1437–1449, 2017.



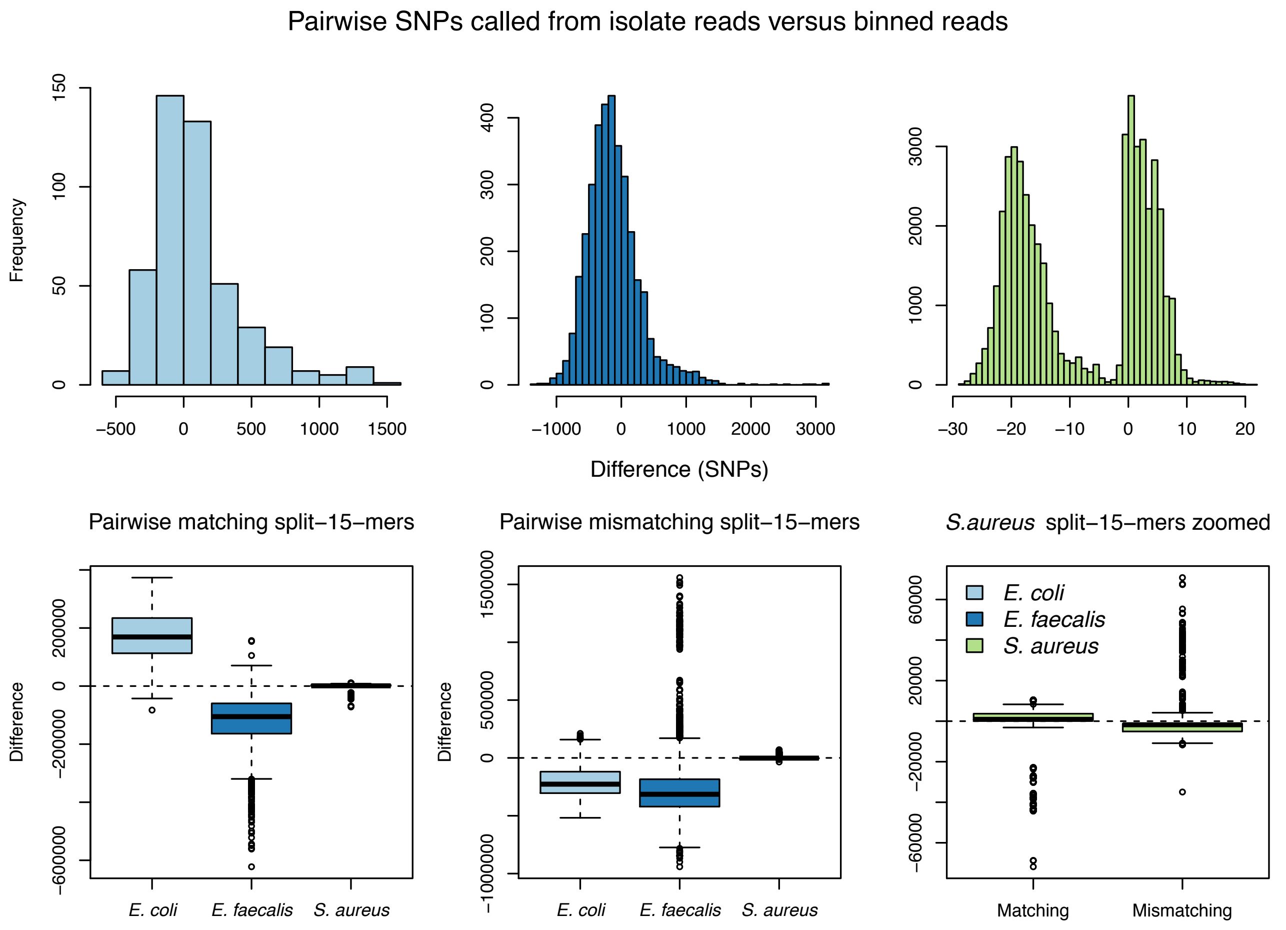
Supplementary Figure 1 *S. aureus* SNPs called from reads vs. assemblies from the mGEMS pipeline.



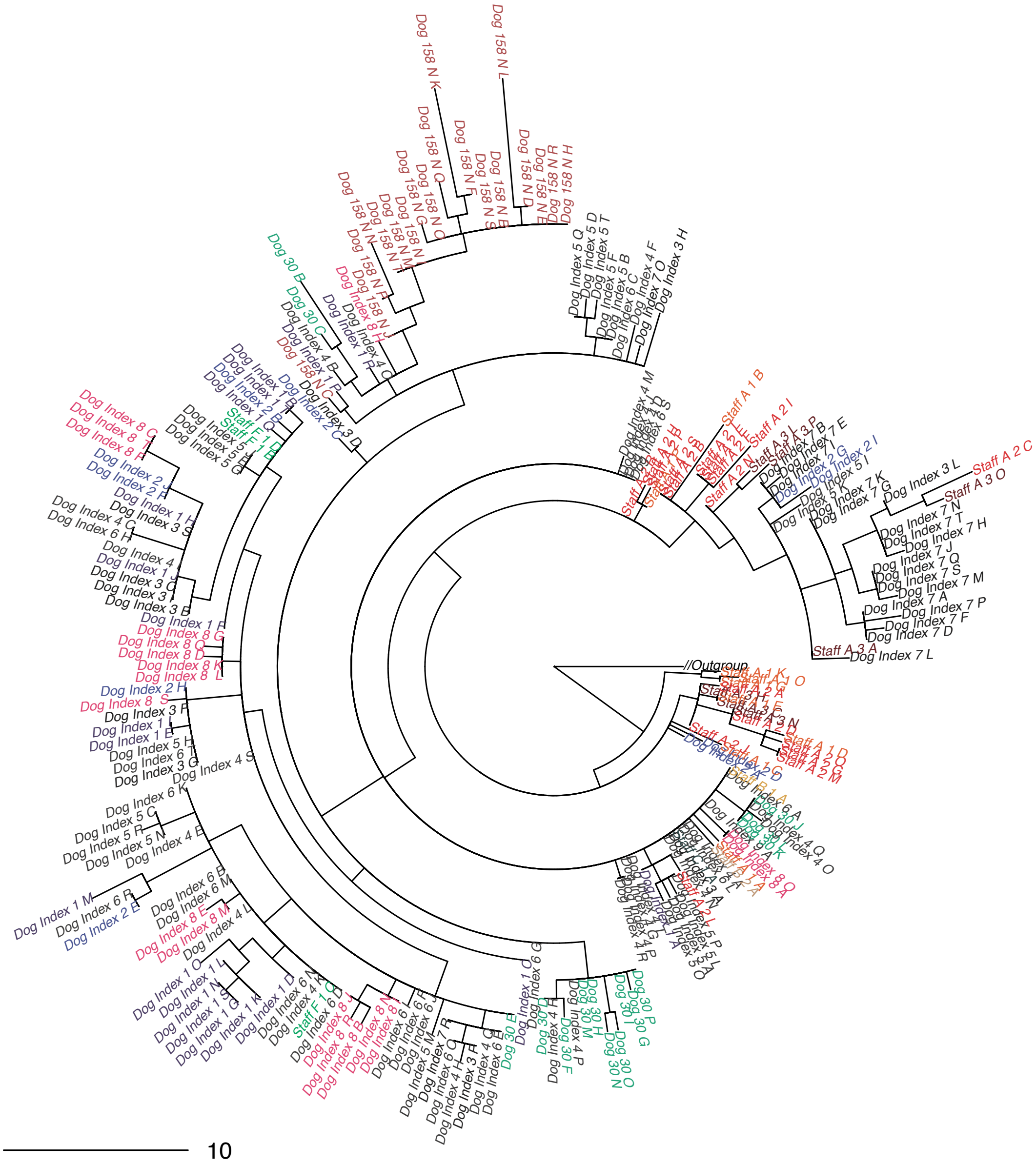
Supplementary Figure 2 SNP calling with the mGEMS pipeline using different assemblers.



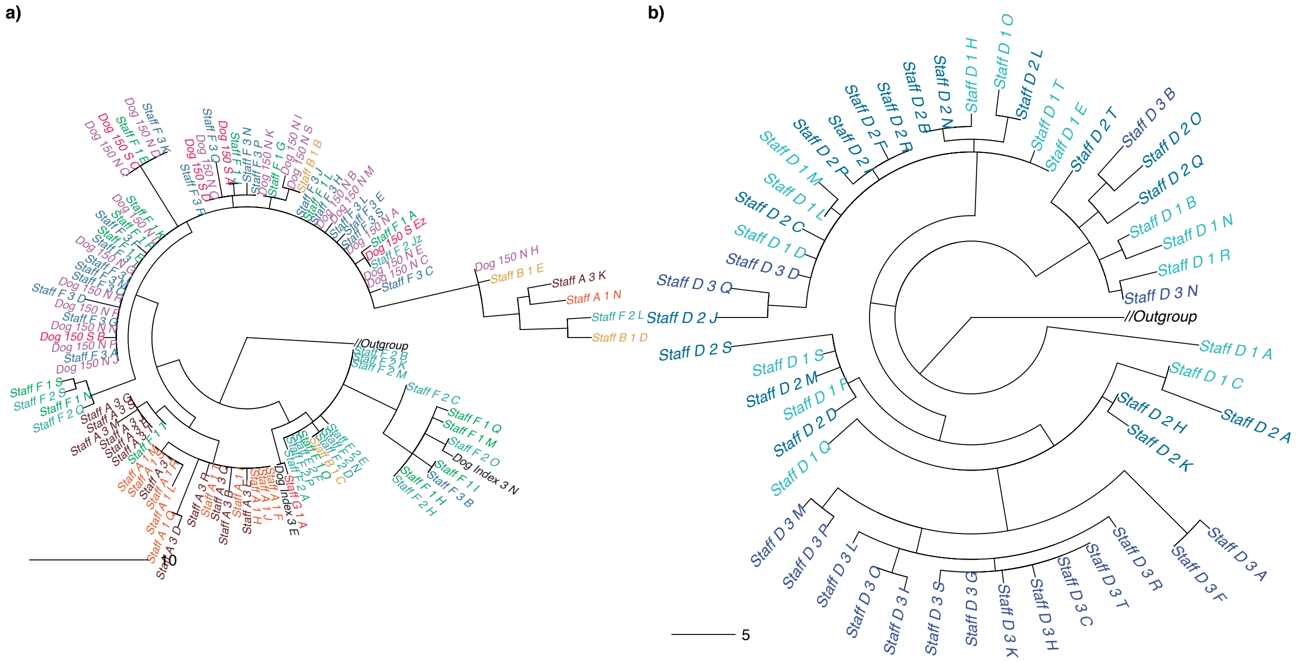
Supplementary Figure 3 Difference in SKA-SNPs called in the reference genome from isolate reads versus mixed samples binned with the mGEMS pipeline.



Supplementary Figure 4 Pairwise SKA-SNPs called from isolate reads versus mixed samples binned with the mGEMS pipeline.



Supplementary Figure 5 Midpoint-rooted maximum likelihood tree from core SNP alignment of *Staphylococcus aureus* ST22 isolate sequencing data showing the clade 1 strains.



Supplementary Figure 6 Midpoint-rooted maximum likelihood trees from core SNP alignment of *Staphylococcus aureus* ST22 isolate sequencing datashowing clade 2 and clade 3 strains.