ProkEvo: an automated, reproducible, and scalable framework for high-throughput bacterial population genomics analyses

Natasha Pavlovikj[1,*], Joao Carlos Gomes-Neto[2,3,*], Jitender S. Deogun[1], Andrew K. Benson[2,3]

[1] Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, Nebraska, United States of America

[2] Department of Food Science and Technology, University of Nebraska-Lincoln, Lincoln, Nebraska, United States of America

[3] Nebraska Food for Health Center, University of Nebraska-Lincoln, Lincoln, Nebraska, United States of America

[*] These authors contributed equally to this work.

Corresponding Author:
Andrew K. Benson[2,3]
115 Food Innovation Center, Lincoln, Nebraska, 68588-6205, USA
Email address: abenson1@unl.edu

Abstract

Whole Genome Sequence (WGS) data from bacterial species is used for a variety of applications ranging from basic microbiological research to diagnostics, and epidemiological surveillance. The availability of WGS data from hundreds of thousands of individual isolates of a given microbial species poses a tremendous opportunity for discovery and hypothesis-generating research, but such opportunity is limited by scalability and user-friendliness of existing pipelines for population-scale inquiry. Here, we present ProkEvo, an automated, scalable, and open-source framework for bacterial population genomics analyses using WGS data. ProkEvo was specifically developed to achieve the following goals: 1) Automating and scaling the computational analysis of many thousands of bacterial genomes starting from raw Illumina paired-ended reads; 2) Using workflow management systems (WMS) such as Pegasus WMS to ensure reproducibility, scalability, modularity, fault-tolerance, and robust file management throughout the process; 3) Utilizing high-performance and high-throughput computational platforms; 4) Generating population-based genotypic analysis at different levels of resolution using the core-genome as an input, and allelic-based or Bayesian statistical tools as classification methods; and 5) Detecting antimicrobial resistance (AMR) genes using varying databases, putative virulence factors, plasmids, and producing pan-genome annotations and data compilation that can be further utilized for analysis. The scalability of ProkEvo is shown by using two datasets with significantly different genome sizes – one with ~2,400 genomes, and the second one an order of magnitude larger containing ~23,000 genomes. Because of its modularity, the running time of ProkEvo varied from ~3-26 days depending on the dataset and the computational platform used. However, if all ProkEvo steps were ran sequentially, the running time would have varied from ~3 months to 13 years. While the running time depends on multiple factors, there is a significant advantage of using such scalable, parallelizable, and automated pipeline. ProkEvo can be used with virtually any bacterial species and the Pegasus WMS enables easy addition or removal of programs from the workflow or modification of options within them. To show this, we used ProkEvo with three important serovars of the foodborne pathogen *Salmonella enterica*, as well as *Campylobacter jejuni* and *Staphylococcus aureus*. These three pathogens all used different MLST scheme, and the program SISTR, which among many functions does cgMLST calls, was only applied to the *S. enterica* serovars. All the dependencies of ProkEvo can be distributed via conda environment or Docker image. To demonstrate ProkEvo's applicability, we have carried a population-based analysis along with the distribution of antimicrobial-associated resistance loci across datasets, and showed how to combine phylogenies with metadata using reproducible Python and R scripts. Collectively, our study shows that ProkEvo presents a viable option for scaling and automating analyses of bacterial populations with direct applications for basic microbiology research, clinical microbiological diagnostics, and epidemiological surveillance.

Introduction

Due to the advances in WGS technology, decreasing costs, and the proliferation of publicly available tools and genomics datasets, the field of bacterial genomics has evolved rapidly from comparative analysis of a few strains of a given species, to analyzing many thousands of genomes [1,2,3,5]. The applications of WGS-based genomics are many, ranging from basic research, public health, pathogen surveillance, clinical diagnostics, and ecological and evolutionary studies of pathogenic and non-pathogenic species [3,4]. Indeed, use of WGS by public health agencies is becoming the standard for epidemiological surveillance, outbreak detection, and source-tracking by providing unprecedented levels of resolution and accuracy [6,7,8].

Within the context of public health, WGS data from populations of pathogenic bacteria such as *Salmonella enterica*, *Campylobacter jejuni* and *Staphylococcus aureus* (when collected temporally from clinical samples, food animals, and food production environments) also create opportunity for ecological and evolutionary inquiry at unprecedented scales of genomic resolution. Powered statistically by the large number of genomes available from surveillance, the data can also be used for complex evolutionary inquiry and predicting features of the genomic architecture that may have been fixed in certain populations due to selection and ecological adaptation in these environments [9,10]. Genomic segments under different patterns of selection or associated with distinct populations based on serovars [13,14], or genotypes at different scales of resolution [15], can further be tested *in silico* to predict potential functional characteristics of populations (e.g. antimicrobial resistance (AMR) [11], virulence and metabolic attributes [10,12]), leading to important hypotheses about the selective forces that are shaping these populations.

Currently, there are small number of automated pipelines available for analysis and genotypic classification of bacterial genomes: EnteroBase [17], TORMES [18], Nullarbor [19], ASA3P [20]. These pipelines each have unique advantages, but differ in the programming language used, the size and type of supported input data, the supported bioinformatics tools, and the computational platform used. Accordingly, these pipelines support different types of biological inquiry. Our work was motivated by the need for a scalable WGS pipeline that can be used broadly for population-based inquiry (ecological, evolutionary, epidemiological). To accommodate the complex combinations of multiple, sequential steps, where each step performs different task and requires different software, we developed a pipeline managed by a Workflow Management System (WMS) [21,22,23,24], which facilitates with managing massive numbers of computational operations in high-performance computing environments, including University or publicly available clusters [25,26], clouds [27], or distributed grids [28,29].

In this paper, we describe ProkEvo – an automated and user-friendly pipeline for population-based inquiry of bacterial species that is managed through the Pegasus WMS and is portable to computing clusters, clouds, and distributed grids. ProkEvo works with raw paired-ended Illumina reads, and is composed of multiple sequential steps. These steps include trimming and quality control, as well as serovar prediction in the case of Salmonella, Multilocus-sequence typing

(MLST) using seven or approximately 300 loci, Bayesian genomic admixture analysis at different scales of resolution, screening for AMR and putative virulence genes, plasmid identification, and pan-genome analysis.

Here, we show the utility and adaptability of ProkEvo for basic metrics of population genetic analysis of using three serovars of the enteric pathogen *S. enterica* (serovars Typhimurium, Newport and Infantis), as well as foodborne pathogens *C. jejuni* and *S. aureus*. We test the scalability and modularity of ProkEvo by using two datasets with ~2,400 and ~23,000 genomes each. In order to show portability and implications to performance, these analyses were each performed on two different computational platforms, the University of Nebraska high-performance computing cluster (Crane) and the Open Science Grid (OSG), a distributed, high-throughput cluster. Additionally, we take an extra step and provide some initial guidance to researchers on how to utilize a few of the output files generated by ProkEvo to perform meaningful population-based analyses in a reproducible fashion using a combination of R and Python scripts. Combined, ProkEvo presents a reliable, efficient, and practical platform for researchers performing bacterial population genomics analyses that can lead to novel discoveries of candidate loci or genotypes of ecological relevance, while generating testable hypothesis related to physiological and virulence attributes of a given population.

Materials & Methods

Overview of ProkEvo

The ProkEvo pipeline is capable of processing tens of thousands of raw, paired-end Illumina reads obtained from NCBI utilizing high-performance and high-throughput computational resources. The pipeline is composed of two sub-pipelines: 1) The first sub-pipeline performs the standard genomics analyses, such as sequence trimming, *de novo* assembly, and quality control; 2) The second sub-pipeline uses the assemblies that have passed the quality control and performs specific population-based classifications (serotype prediction specifically for Salmonella, genotype classification at different scales of resolution, analysis of core- and pan-genomic content). Pegasus WMS manages and splits each sub-workflow into as many independent tasks as possible to take advantage of many computational resources.

A text file of SRA identifications corresponding to the raw Illumina reads deposited to the Sequence Read Archive (SRA) database in NCBI (NCBI SRA) is used as an input to the pipeline. The first step of the pipeline and the first sub-workflow is downloading genome data from NCBI SRA [30]. This is done using the package parallel-fastq-dump [31]. The SRA files are downloaded using the prefetch utility, and the downloaded files are converted into paired-end fastq reads using the program parallel-fastq-dump. While the SRA Toolkit [30] provides the same functionality, this toolkit can be slow sometimes and show intermittent timeout errors, especially when downloading many files. parallel-fastq-dump is a wrapper for SRA Toolkit that speeds the process by dividing the conversion to fastq files into multiple threads. After the raw paired-end fastq files are generated, quality trimming and adapter clipping is performed using

Trimmomatic [32]. FastQC is used to check and verify the quality of the trimmed reads [33]. FastQC is run independently for each paired-end dataset and all output files are concatenated at the end for a summary. The paired-end reads are assembled *de novo* into contigs using SPAdes [34]. These assemblies are generated using the default parameters. The quality of the assemblies is evaluated using QUAST [35]. The information obtained from QUAST is used to discard assemblies with 0 or more than 300 contigs, or assemblies with N50 value of less than 25,000. The filtering of the assemblies concludes the first part or first sub-pipeline of the workflow. Each of these steps is independent of the input data and each task is performed on one set of paired-end reads using one computing core. This makes the analyses modular and suitable for high-throughput resources with many available cores. Moreover, having many independent tasks significantly reduces the memory and time requirements while generating the same results as when the analyses are done sequentially. Theoretically, if a dataset has 2,000 paired-end reads and a computational platform has 2,000 available cores, ProkEvo can scale and utilize all these resources at the same time. Needless to say, this is extremely important for any real-time large-scale population genomics analyses.

The second sub-pipeline uses the assemblies that passed the quality control to perform more specific population-based characterizations, including genotypic classifications, serovar prediction exclusively for Salmonella, gene-based annotations, and pan-genome outputs. PlasmidFinder is used to identify plasmids in the assemblies [36]. PlasmidFinder comes with curated database of plasmid replicons to identify plasmids in sequences from the Enterobacteriaceae species. SISTR is used for Salmonella only and produces serovar prediction and *in silico* molecular typing by determination of antigen gene and core-genome multilocus sequence typing (cgMLST) gene alleles [14]. SISTR generates multiple output files. The one we are interested the most for further downstream analyses is the primary output file named sistr_output.csv. The filtered assemblies are annotated using Prokka [37]. Prokka comes with a set of core and HMM databases for the most common bacterial species. If needed, one can customize and create their own annotation database. In addition to the other files, Prokka produces annotation files in GFF3 format that are used with Roary [38] to calculate the pan-genome and generate the core-genome alignment. The produced core-genome alignment file is then used with fastbaps, an improved version of the BAPS clustering method [39], to hierarchically cluster the genetic sequences from the multiple sequence alignment in varying numbers of stratum. Multilocus-sequence typing is performed using MLST [40]. Here, the isolates are characterized by being compared to sequences of seven ubiquitous, house-keeping genes [41] using the filtered genome assemblies. In addition to these analyses, the filtered assemblies are screened for AMR and virulence associated genes using ABRicate [42]. ABRicate comes with multiple comprehensive gene-based mapping databases, and the ones used in ProkEvo are NCBI [43], CARD [44], ARG_ANNOT [45], Resfinder [46], and VFDB [47]. Prokka, SISTR, PlasmidFinder, MLST, and ABRicate are independent of each other, and they are all run simultaneously in parallel. Moreover, Prokka, SISTR and PlasmidFinder perform their computations per filtered assembly, while MLST and ABRicate require all filtered assemblies to be used together. Running multiple independent jobs simultaneously is one of the key factors to maximize computational efficiency. At the end, once the SISTR analyses finish for all assemblies, the generated independent sistr_output.csv files are concatenated. This aggregation of files can be done because the genome annotation to serovars and cgMLST lineages done by SISTR occurs completely independent for each genome. Each tool executed in ProkEvo is run with specific options. While the options used in this paper fit the presented case studies, these options are easily adjustable and configurable in the pipeline. Because we developed ProkEvo initially for studying the bacterial pathogen *Salmonella enterica*, due to our specific needs, the pipeline was specifically designed to implement SISTR, which accurately assigns serovar based on the Kauffman-White scheme [56]. However, the overall pipeline is not specific to this bacterial species and other serotype prediction modules can be substituted for SISTR to accommodate user-specific needs. Additionally, MLST program can be directed to other species-specific sets of genetic loci, as shown with the *Campylobacter jejuni* and *Staphylococcus aureus* datasets. Or else, the user may decide to bypass genotype-calling tools altogether, and carry out other types of analyses. ProkEvo is amenable to such modifications.

The modularity of ProkEvo allows us to decompose the analyses into multiple tasks, some of which can be run in parallel, and utilize a WMS. ProkEvo is dependent on many well-developed bioinformatics tools and databases which setup and installation are not trivial. In order to ease this process, reduce the technical complexity, and allow reproducibility, we provide two software distributions for ProkEvo. The first distribution is a conda environment that contains all software dependencies [48], and the second one is a Docker image that can be used with Singularity [49]. Both distributions are supported by the majority of computational platforms and integrate well with ProkEvo, and can be easily modified to include other tools and steps. The code for ProkEvo, and both the conda environment and the Docker image, are publicly available at our GitHub repository (https://github.com/npavlovikj/ProkEvo).

Pegasus Workflow Management System

ProkEvo uses the Pegasus WMS which is a framework that automatically translates abstract, high-level workflow description into concrete efficient scientific workflow, that can be executed on different computational platforms, such as clusters, grids, and clouds. The abstract workflow of Pegasus WMS contains information and description of all executable files (transformations) and logical names of the input files used by the workflow. On the other hand, the concrete workflow specifies the location of the data and the execution platform [24]. The workflow is organized as a directed acyclic graph (DAG), where the nodes are the tasks and the edges are the dependencies. Next, the workflow is submitted using HTCondor [50]. Pegasus WMS uses DAX (directed acyclic graph in XML) files to describe an abstract workflow. These files can be generated using programming languages such as Java, Perl, or Python. The high-level of abstraction of Pegasus allows scientists to ignore low-level configurations required by the underlying execution platforms. Pegasus WMS is an advanced system that supports data management and task execution in automated, reliable, efficient, and scalable manner. This whole process is monitored, and the workflow data is tracked and staged. The requested output results are presented to the researchers, while all intermediate data can be removed or re-used. In case of errors, jobs are automatically retried. If the errors persist, a checkpoint file is produced so the job can be resubmitted and resumed. Pegasus WMS supports sub-workflows, task clustering and defining memory and time resources per task. Pegasus WMS generates web dashboard for each workflow for better workflow monitoring, debugging, and analyzing. Pegasus WMS comes with a set of useful command-line tools that help researchers to submit and analyze workflows and generate useful statistics and plots about the workflow performance, running time, and machines used.

ProkEvo uses Python to create the workflow description. Each step of the pipeline is a computational job represented as a node in the DAG. Two nodes are connected with an edge if the two jobs need to be run one after another. The input and output files are defined in the DAG as well. All the jobs that are not dependent on each other can be run concurrently. Each job uses its own predefined script that executes the program the job requires with the specified options. This script can be written in any programming language. The bioinformatics tools and programs required by ProkEvo can be distributed through conda environment [48] or Docker image [49]. The predefined scripts are already part of ProkEvo, and no further changes or modifications are needed. With the modularity of Pegasus, each job requests its own run time and memory resources. Exceeding the memory resources is a common occurrence in any bioinformatics analysis. Thus, if exceeding the memory is a reason for a job failure, Pegasus retires the job with increased requirements. Higher memory requirements may mean longer waiting times for resources, and it is really important and efficient to use high memory requirements only when needed, which is allowed by Pegasus WMS. ProkEvo is written such that supports execution on high-performance and high-throughput computational platforms. In the analyses for this paper, we use both the University cluster and OSG, and working versions for both platforms are available in our GitHub repository (https://github.com/npavlovikj/ProkEvo).

Computational execution platforms

Traditionally, scientific workflows have been executed on high-performance and high-throughput computational platforms. While high-performance platforms provide resources for analyses that require lots of cores, time, and memory, high-throughput platforms are suitable for many small and short independent tasks. The design of ProkEvo fits University and other publicly or privately available clusters and grids, providing flexibility in the computational platform.

University cluster (Crane), a high-performance computational platform

University and other public clusters are shared among all users and enforce fair-share scheduling and file and disk spaces quotas. The clusters are suitable for various types of jobs, such as serial, parallel, GPU, and high memory specific jobs, thus the high-performance. Crane [25] is one of the high-performance computing clusters at the University of Nebraska Holland Computing

Center (HCC). Crane is Linux cluster, has 548 Intel Xeon nodes, with RAM ranging from 64GB to 1.5TB, and supports Slurm and HTCondor as job schedulers. In order to use Crane, one needs an HCC account associated with a University of Nebraska faculty or research group. While we use Crane as a computational platform for ProkEvo, the majority of University and publicly available high-performance clusters are administered in a similar way and can be used to run ProkEvo.

Crane has support for Pegasus and HTCondor, and no further installation is needed in order to run ProkEvo. Due to the limited resources and fair-share policy on Crane, tens to hundreds of independent jobs can be run concurrently. We provide a version of ProkEvo suitable for Crane with conda environment, which contains all required software. Crane has a shared file system where the data is accessible across all computing nodes. Depending on the supported file system, Pegasus is configured separately and handles the data staging and transfer accordingly. However, users do not need any advanced experience in high-performance computing to run ProkEvo on Crane, or any other University or publicly available cluster. Users only need to provide list of SRA identifications and run the submit script that distributes the jobs automatically as given in our GitHub repository (https://github.com/npavlovikj/ProkEvo).

Open Science Grid (OSG), a distributed, high-throughput computational platform

The Open Science Grid (OSG) is a distributed, high-throughput distributed computational platform for large-scale scientific research [28,29]. OSG is a national consortium of more than 100 academic institutions and laboratories that provide storage and tens of thousands of resources to OSG users. These sites share their idle resources via OSG for opportunistic usage. Because of its opportunistic approach, OSG as a platform is ideal for running massive numbers of independent jobs that require less than 10GB of RAM, less than 10GB of storage, and less than 24 hours running time. If these conditions are fulfilled, in general, OSG can provide unlimited resources with the possibility of having hundreds or even tens of thousands of jobs running at the same time. The OSG resources are Linux-based, and due to the different sites involved, the hardware specifications of the resources are different and vary. Using OSG is free for academic usage. The host institution does not need to be part of OSG for a researcher to use this platform.

All steps from the population genomics analyses fulfill the conditions for OSG-friendly jobs. Thus, ProkEvo can efficiently utilize these distributed high-throughput resources, and run thousands of analyses concurrently if the resources are available. OSG supports Pegasus and HTCondor, so no installation steps are required. We provide version of ProkEvo suitable for OSG (https://github.com/npavlovikj/ProkEvo). This version uses the Docker image with all software requirements via Singularity and supports non-shared file system. In non-shared systems, the resources do not share the data. The data are read and written from a staging location, and all of this is handled by Pegasus WMS. In order to run ProkEvo on OSG, users only need to provide list of SRA identifications and run the submit script without any advanced experience in high-throughput computing.

Population genomics analyses

The population-based analyses performed in this paper provide an initial guidance on how to comprehensively utilize the following output files produced by ProkEvo: 1) MLST output (.csv); 2) SISTR output (.csv); 3) BAPS output (.csv); 4) Core-genome alignment file (core_gene_alignment.aln) for phylogenetic analysis; and 5) Resfinder output (.csv) containing AMR genes. We use both R and Python 3 Jupyter Notebooks for all our data analyses (https://github.com/npavlovikj/ProkEvo). The input data used for these analyses is available on Figshare (https://figshare.com/projects/ProkEvo/78612).

A first general step in this type of analysis is opening all files in the preferred environment (i.e., RStudio or JupyterHub), and merging them into a single data frame based on the SRA (genome) identification. Next, we perform quality control (QC) of the data, focusing on identifying and dealing with missing values, or cells of the data frame containing erroneous characters such as hyphens (-) and interrogation marks (?). For that, we demonstrate our approach for cleaning up the data prior to conducting exploratory analysis and generating all visualizations.

In the case of Salmonella datasets, we used an additional important "checking/filtering" step after the QC is done. Since the program SISTR provides a serovar call based on genotypic information, one can opt for keeping those genomes that do not match the original serovar identification in the analysis, or excluding them. Both approaches are justifiable with the latter one being more conservative, and it specifically assumes that the discordance between data entered in NCBI and genotypic prediction done by SISTR is accurate. However, it is important to remember that we initially expect that the dataset belongs to a particular serovar because of the keywords we used to search the NCBI SRA database, such as: "*Salmonella* Newport", "*Salmonella* Typhimurium", or "*Salmonella* Infantis". Typically, the proportion of genomes that are classified by SISTR as other serovars can be somewhat minor, but may also bias the analysis depending on the size of the dataset (~ < 3% for any given Salmonella dataset equals "miscalls"). In our case, for example, we were conservative and either filtered the "miscalls" out of the data for some analysis, or kept it as a separate group called "other serovars". The latter approach was done for some specific analysis, such as phylogenetics, whereby the program we used required us to have all data points in place (e.g. ggtree in R). That is the case because the core-genome alignment used for the phylogeny is generated by Roary without considering the SISTR prediction for serovar calls. If that is of interest, the user can add a conditional to the pipeline to run Roary after considering SISTR results, but that only applies to Salmonella genomes. However, we do note that stringent requirements for serotype classification (i.e. filtering out "miscalls" based on SISTR predictions) could eliminate important variants that may genotypically match known populations of the serovar, but which have acquired mutations or recombination events at serotype-determining loci. The larger the datasets are, the more influential that percentage of discordant calls can be. Hence, for Salmonella specifically, this has to be considered carefully depending on the research question to be answered and database being utilized. Our suggestion is that for any predictive analysis, one should either filter out, or at least, classify the potential miscalls as other serovars after running SISTR.

To define metrics for the *S*. Typhimurium and *S*. Newport populations for population structure analysis, the pipeline combines MLST-based genotypes at different scales of resolution with Bayesian-based predictions of genomic structure. Combining these varying genotyping approaches allows for classification and quantification of relative frequency distribution of genotypes/haplotypes, as well as the visualization of their genetic relationships. In this version of ProkEvo, we have implemented legacy MLST for ST calls using seven loci, core-genome MLST (cgMLST) that uses approximately 330 loci for MLST analysis, and a Bayesian-based BAPS haplotype classification using six layers of BAPS (BAPS1 being the lowest level of resolution and BAPS6 being the highest). We also use a hierarchical approach for exploring the relative frequencies of genotypic and genomic classifications one to another. For example, genomes can be classified based on BAPS1 and the distribution of legacy STs can be assessed relative to the BAPS-inferred genomic structures populations. Likewise, the genetic relationships of thousands of cgMLST genotypes can also be assessed with respect to the BAPS-based and ST-linked genomic architecture at different levels of resolution to infer evolutionary relationships. This hierarchical approach was possible for the *S*. Newport dataset of ~2,400 genomes (USA data), but the core-genome alignment step was not scalable to the 10-fold larger dataset of *S*. Typhimurium (~23,000 genomes – worldwide data), which required split into twenty smaller datasets during the core-genome alignment step. Basically, our empirical experience has been that Roary performs without errors and converges when having approximately up to 2,000 genomes. Although random partitioning of the subsets should yield the same classifications of dominant genomic groups, the Bayesian classification algorithm (BAPS) may not necessarily assign grouping numbers for different genomic types in a standardized manner across different subsets of a larger dataset. Aggregation of the BAPS data from subsets therefore requires user-based input. On the other hand, sub-setting the data is advantageous for downstream data science and machine learning analyses since they require a nested cross-validation approach for feature selection and predictive analytics. Herein, we use a random sampling approach to split the data for *S*. Typhimurium used with Roary. Based on the number of genomes, we created 20 groups such that each has 1,076-1,077 genomes. Next, from the GFF files produced by Prokka, we randomly selected and assigned genomes to each group using custom Bash scripts. Both Roary and fastbaps were run per group, resulting in 20 independent runs and output files. In addition to these analyses, we check the count of haplotypes within a major cgMLST (i.e. epidemiological clone) vs. others using all six layers of the BAPS clustering algorithm (BAPS1-6). A highly clonal population of a given cgMLST is expected to display very few genotypes at all six levels of BAPS. In contrast, a diverse population of a given cgMLST or highly related cgMLST genotypes may partition between different BAPS-based genomic groups. In practice, this analysis is important to examine how homogenous or heterogenous a population is, which has implications for ecological and epidemiological inference. Complementary to this population structure analysis, we demonstrated the distribution of some AMR genes within and between

Salmonella serovars, including *S*. Infantis (~1,700 genomes – USA data), and within them across their respective major ST populations. For that, we selected the Resfinder outputs as an example to show the identification of putative AMR genes. We arbitrarily selected genes with proportion higher than or equal to 25% for *S*. Newport, *S*. Infantis, and *S*. Typhimurium, for visualizations which were produced with ggplot2 in R [52]. The respective scripts are provided in our repository (see GitHub link for code).

In order to demonstrate how versatile ProkEvo can be, we also conducted a population-based analysis of *C. jejuni* and *S. aureus* datasets from USA, each containing 21,919 and 11,990 genomes, respectively. For both datasets, we analyzed the population structure using BAPS1 and STs. The same hierarchical population basis described for Salmonella applies here, with BAPS1 coming first and STs next in terms of population ranking. We used a random sample of ~1,000 genomes of each species to demonstrate the distribution of BAPS1 and STs onto the phylogenetic structure. Phylogenies were constructed using the core-genome alignment produced by Roary, and by applying the FastTree program [53] using the generalized time-reversible (GTR) model of nucleotide evolution (see GitHub link for code). Additionally, we showed the distribution of STs within each bacterial species (only showed STs with proportion higher than 1%), and the relationship between the relative frequencies of dominant STs and AMR genes. Genes with relative frequency below 25% were filtered out of the data. All visualizations were generated with ggplot2 in R, and the scripts are also provided in our repository.

Results

Overview of ProkEvo

Figure 1 shows the overall flow of tasks performed by ProkEvo including all specific bioinformatics tools used for each task. On the other hand, *Fig. 2* presents the Pegasus WMS design of ProkEvo. The DAG shown contains all independent input and output files, tasks, and the dependencies among them. The modularity of ProkEvo allows every single task to be executed independently on a single core. As seen on *Fig. 2*, there are approximately 10 tasks executed per one genome. When ProkEvo is used with whole bacterial populations of thousands of genomes, the number of total tasks is immense. Advanced WMS such as Pegasus allow scaling of these tasks independently and utilizing diverse computational platforms. *Figure 3* provides an example of running ProkEvo on the two different computational platforms used in this paper - the University of Nebraska high-performance computing cluster (Crane) and the Open Science Grid (OSG), a distributed, high-throughput cluster, using two datasets of significantly different size (~2,400 genomes [1X] vs. ~23,000 genomes [10X]). The ProkEvo code available on our GitHub page supports both platforms, and the researcher can choose which one to use. Both platforms have different structure and have their own advantages and disadvantages that are highlighted in *Fig. 3*.

Performance evaluation

To measure the scalability and adaptability of ProkEvo, we used two datasets with significantly different genome sizes – one with ~2,400 genomes (*S*. Newport), and the second one an order of magnitude larger (~23,000 genomes from *S*. Typhimurium). We ran ProkEvo on two different computational platforms, the high-performance cluster at the University of Nebraska (Crane) and the OSG, a distributed, high-throughput cluster. Each dataset was run once on the two platforms and statistics about the Pegasus WMS workflow were generated. Of note, there may be variation in the ProkEvo runtime from project to project based on the availability of resources on each platform. As an HPC resource of the Holland Computing Center, the Crane cluster is managed by fair-share scheduling, while as an opportunistic HTC resource, the OSG resources may be dynamically de-provisioned or having intermittent issues. These factors may impact the future predictability of running time and performance of ProkEvo on both platforms. In average, on Crane we had hundred jobs running at the time, and due to the similar type of nodes available, the runtime should be similar for multiple runs of the same workflow. On the other hand, the nodes on OSG are more diverse and the runtime and the number of jobs for multiple runs can be significantly different (from few jobs running at the same time to few tens of thousand).

ProkEvo consists of two sub-workflows, with number of jobs varying from a few thousands to a few hundreds of thousands, depending on the dataset used. "pegasus-statistics" generates summary statistics regarding the workflow performance, such as the total number of jobs, total run time, number of jobs that failed and succeeded, task and facility information, etc. Some of these statistics are demonstrated in *Table 1*. The total distributed running time is the total running time of ProkEvo from the start of the workflow to its completion. The total sequential running time is the total running time if all steps in ProkEvo are run one after another. In case of retries, the running times of all re-attempted jobs are included in these statistics as well. Beside the workflow runtime information, *Table 1* also shows the maximum total number of independent jobs ran on Crane and OSG within one day. Moreover, the total count of succeeded jobs is shown for both computational platforms and datasets.

When ran on Crane, ProkEvo with *S*. Newport completely finished in 3 days and 15 hours. If this workflow were run sequentially on Crane, its cumulative running time would be 115 days and 18 hours. On the other hand, ProkEvo with *S*. Newport finished in 7 days and 4 hours when OSG was used as a computational platform. Similarly, if this workflow were run sequentially on OSG, its cumulative running time would be 1 year and 69 days. As it can be observed, the workflow running on OSG took longer than the workflow running on Crane. OSG provides variable resources with different configuration and hardware, and depending on that, the performance may vary significantly. Also, the OSG jobs may be preempted if the resource owner submits more jobs. In this case, the preempted job is retried, but that additional time is added to the workflow wall time. While the maximum number of independent jobs ran on Crane in one day is 2,377, this number is 8,606 when OSG was used. This is where the importance of using HTC resources such as OSG can be observed - the high number of jobs and nodes that can be run and used simultaneously, which is often a limit for University clusters. The total number of successful jobs ran with ProkEvo with the *S*. Newport dataset is 9,281 on Crane and 16,624 on

OSG. Due to the opportunistic nature of the OSG resources, a running job can be cancelled and retried again, thus the higher number of jobs reported by OSG. Similar pattern can be observed when ProkEvo was run with the *S*. Typhimurium dataset. When ran on Crane, ProkEvo with *S*. Typhimurium completely finished in 15 days and 22 hours. If this workflow was run sequentially on Crane, its cumulative running time would be 2 years and 268 hours. On the other hand, the ProkEvo run for *S*. Typhimurium finished in 26 days and 6 hours, when using OSG as a computational platform. Similarly, if this workflow were run sequentially on OSG, its cumulative running time would be 13 years and 50 days. The maximum number of independent jobs ran on Crane and OSG is 12,382 and 25,540 respectively. The total number of successful jobs ran with the *S*. Typhimurium dataset is 217,942 on Crane and 232,422 on OSG.

Although the workflow run time was better when Crane was used as a computational platform, it can be noticed that the bigger the dataset and the more jobs are running, the higher the efficiency of using OSG is. As long as resources are available and no preemption occurs, workflows running on OSG can have a great performance. On OSG, ProkEvo ran on resources shared by thirty-four different facilities. Failures and retries are expected to occur on OSG, and their proportion may vary. From our experience, the number of failures and retries took up ~0.3%-30% of the total number of jobs. However, the OSG support staff acts promptly on isolating these issues, which can also be masked by a resilient and fault-tolerant workflow management systems like Pegasus WMS. All the data, intermediate and final files generated by ProkEvo are stored under the researcher's allocated space on the file system on Crane. Depending on the file system, it is possible that there are file count and disk space quotas. When large ProkEvo workflows are run, these quotas may be exceeded. On the other hand, due to the non-shared nature of the file system of OSG, intermediate files are stored on different sites, and exceeding the quotas is usually not an issue.

Both Crane and OSG are computational platforms that have different structure and target different type of scientific computation. All analyses performed with ProkEvo fit both platforms well. Thus, we provide an unambiguous comparison of both platforms and show their advantages and drawbacks when large-scale workflows such as ProkEvo are run.

Applications

In this Section, we present a diverse array of analysis carried out across three important zoonotic serovars of Salmonella, and two other widespread species of foodborne pathogens, namely *C. jejuni* and *S. aureus*. While these data were collected from a recognizably biased database that is inflated with clinical isolates, we are focusing on demonstrating some of the utilities and approaches that can result from using ProkEvo for population-based analysis. Therefore, we are limiting ourselves from making any generalizable inference about the ecology and epidemiology of these populations. However, the analytical framework is still valid and applicable for analyzing more sophisticatedly designed collections of isolates, or even doing pattern searching with publicly available databases. More specifically, our goal is to demonstrate how to conduct an initial population-based analysis with some of ProkEvo's outputs. To achieve that objective, we present a series of independent case studies that encapsulate some of the most common approaches for studying bacterial populations. Prior to discussing those case studies, we highlight some important concepts regarding bacterial population genetics and ecology. We have selected to work with these three species of foodborne pathogens because of our specific research interest. However, ProkEvo can be used with other bacterial species with a few limitations: 1) The MLST program only works if the target bacterial species has an allelic profile present in the database, or is incorporated by the user; and 2) SISTR is designed to only work for Salmonella, but can be easily blocked out from the pipeline by the user.

Overview of the population structure and ecology for Salmonella, *C. jejuni* and *S. aureus*

To understand the real applicability of ProkEvo, it is important to provide some insights regarding the most relevant aspects of the biology of the target organisms. Foodborne gastroenteritis is among the most prevalent zoonotic infectious illnesses of humans, with the pathogenic bacterial species such as *S.* enterica lineage I, *C. jejuni*, and *S. aureus* being one of the most prevalent causative agents worldwide [54].

Salmonella populations can be found as common inhabitants of the gastrointestinal tract in a wide range of mammals, birds, reptiles, and insects and these organisms are often transmitted to humans through contaminated animal products, vegetables, fruits, and processed foods [55]. The genus Salmonella comprises two primary species (*S. enterica* and *S. bongori*), which are believed to have diverged from their last common ancestor approximately 40 million years ago [88]. Worldwide, *S. enterica* is the most frequently isolated species from human clinical cases and from most environments. After the ancestral divergence from the common ancestor with *S. bongori*, the *S. enterica* lineage has further diversified into six different sub-species. The vast majority (>90%) of known human cases are caused by populations descending from a single sub-species, namely *S. enterica subsp*. enterica (lineage I). Even within lineage I, there is still tremendous genetic and phenotypic diversity, as the lineage has diverged into a diverse array of distinct sub-types or sub-populations that have classically been differentiated by serological typing of markers on their cell surface (lipopolysaccharide molecules and major protein components of the flagellum) [56,89]. The >2,500 known, serologically-distinct serovars represent relevant biological units for epidemiological surveillance and tracking because isolates belonging to the same serovar show much less variation with respect to important traits such as range of host species, survival in the environment, efficiency of transmission to humans, and virulence characteristics, than isolates from different serovars [7,89]. Indeed, the diverse array of serotypic markers, host ranges, and human disease phenotypes are covariates with the population structure of *S. enterica* lineage I, with most serovars marking unique clonal lineages. Thus, different isolates of a given serotype share more recent ancestry to one another than they do to isolates of any other serotype [89]. In fact, the serotype of most isolates can be predicted accurately from ST distributions. Herein, we have decided to use an example of genomes representing the following three serovars of *S. enterica* lineage I: *S.* Infantis, *S.* Newport, and *S.* Typhimurium. These are among the top twenty-five most prevalent and important zoonotic serovars of Salmonella according to the Center for Disease Control and Prevention [57]. All three serovars are capable of gastroenteritis in humans and their typical reservoir is livestock. Bovine appear to be the most common source for *S*. Infantis and *S*. Newport, while *S*. Typhimurium has a generalist life-style and can be found in swine, poultry, bovine, etc. [55].

The population structure of Salmonella is largely clonal and hierarchical genotyping schemes such as MLST show that isolates having genetically-related core-genome MLST (cgMLST) genotypes (high-resolution based on ~330 highly conserved genes) are mostly embedded within clonally-related STs defined at lower resolution by seven-gene MLST [7]. Thus, the *S. enterica* lineage I population structure can be hierarchically analyzed by first identifying the serovar in question, and then breaking it down into ST and cgMLST. However, at high levels of resolution (cgMLST), inferring the phylogenetic relationships across thousands of different cgMLST genotypes is computationally not scalable, especially if having to account for horizontal gene transfer (HGT) by removing putative recombination events across divergent lineages. To overcome this problem, genotypic classification of isolates can be combined with scalable Bayesian-based computational approaches such as BAPS, which determines evolutionary relationships based on compositional features of the core-genome at different scales of resolution. Thus, evolutionary relationships of ST clonal complexes and cgMLST genotypes can be inferred efficiently by using a hierarchical classification with six BAPS levels (BAPS1 being the lowest level, and BAPS6 the highest level of resolution and population fragmentation). In our heuristic-based approach, we use the following hierarchical level of population structure analysis for Salmonella: 1) Serovar; 2) BAPS1; 3) STs; and 4) cgMLSTs. Our empirical experience has been that multiple STs can be part of the same sub-group within BAPS1, implying they have shared a common ancestor more recently than the divergent ones. This BAPS1 vs. ST hierarchical relationship has been shown before for Salmonella [58], and even for a completely unrelated species, such as *Enterococcus faecium* [59]. Of note, epidemiological clones, which comprise a homogenous population of isolates related to an outbreak, are typically genotyped as cgMLSTs. That happens because cgMLST offers the appropriate level of granularity to define genotypes at the highest level of resolution while considering the shared genomic variation across isolates (i.e. all shared, or >99% loci) [7].

Besides *S. enterica* Lineage I, there are two major species of Campylobacter associated with gastrointestinal diseases in humans, namely *C. jejuni* and *C. coli* [60]. *Campylobacter jejuni* is more often associated with outbreaks in developed countries such as the USA, with poultry and dairy products being the most common sources of the pathogen [61]. As in the case of Salmonella, *C. jejuni* population structure can be studied using a hierarchical approach, excluding serovars, but including BAPS1, STs, and cgMLSTs. One unique aspect of *C. jejuni* population biology is the potential for high frequency of HGT, which not only affects the acquisition of novel loci, but also the population structure of the microorganism [60]. That is, *C. jejuni* is less clonal than any given serovar of *S. enterica* lineage I, and it contains a variety of widespread STs, for which the diversification patterns appear to be strongly affected by the host colonized with this pathogen [60,62].

Whereas, Salmonella and *C. jejuni* are gram-negative bacteria that belong to the same phylum Proteobacteria, *S. aureus* is a gram-positive species that pertains to the phylum Firmicutes. *Staphylococcus aureus* can cause a diverse array of diseases in humans including skin infections, endocarditis, among others, but it is also a foodborne pathogen [63]. Gastroenteritis caused by this pathogen is due to the production of enterotoxins. Livestock are one of *S. aureus* reservoirs, but it can also live in human skin and nasal cavity. In the case of *S. aureus*-associated foodborne illnesses, humans ingest products such as milk-derivatives (e.g. cheese), and meat that are contaminated with enterotoxins produced by the pathogen, which appear to occur due to the environmental stress caused by those specimens [64]. *Staphylococcus aureus* population can be structured the same way as that of Salmonella and *C. jejuni* using BAPS1, STs, and cgMLSTs. However, this pathogen is not as diverse as *C. jejuni* at the ST level, but its degree of clonality is more comparable to those serovars within *S. enterica* lineage I. Altogether, our approach here is to use these different levels of genotypic resolutions to demonstrate some of the aspects of the population structure of these organisms, while highlighting their degree of clonality and relatedness since those may reflect important ecological characteristics of the pathogen. Also, from an epidemiological point of view, using ST and cgMLST identifications is a manner to which researchers and microbiologists can standardize the nomenclature to discuss specific aspect of a population that might be multi-drug resistance and/or a culprit in an outbreak.

In this era of systems biology and multi-omics methodologies, it is highly desirable to link genetic classifications of isolates (e.g. serovar, MLST, cgMLST genotypic classifications, and BAPS-based genetic relationships) to important phenotypes associated with resistance to antimicrobial agents, virulence, host adaptation, transmission, and environmental survival. Although linked genotypic and phenotypic data can certainly inform epidemiological surveillance, the linkage affords an even greater opportunity to identify signatures of evolutionary processes (selection) and ecological fitness of the different pathogenic populations in animal and food production environments at the molecular scale [90]. Genes and pathways marked by these processes may illuminate selective pressures and better inform risk assessments as well as development of strategies to mitigate spread. Therefore, here we provide a practical example of how to link the distribution of known AMR genes to the population structure of the organism using serovars and STs in the case of *S. enterica* lineage I serovars, and STs for *C. jejuni* and *S. aureus*. We chose to use known AMR loci for its association with the spread of STs and epidemiological clones worldwide, as in the case of Salmonella [65], *C. jejuni* [66], and *S. aureus* [67].

<u>Case study 1: *S.* Newport population structure analysis</u>

The *S. enterica* serovar Newport is a zoonotic pathogen that ranks among the top 25 serovars considered as emerging pathogens by public health agencies due to several recent outbreaks of foodborne gastroenteritis in humans [91]. Unlike most serovars of *Salmonella enterica* lineage I, which comprise worldwide populations dominated by a single ST clonal complex, the *S.* Newport serovar has diversified into four distinct STs (*Fig. 4A*). The genetic diversity detected in

S. Newport isolates is surprising given its relatively low representation in the NCBI SRA database when only selecting genomes from the USA (total of 2,392 isolates). Thus, this serotype provides a robust example for analysis of a moderately complex population structure through ProkEvo. After the pre-processing steps, assemblies from 2,365 isolates passed the filtering step. The total output data produced by ProkEvo for *S*. Newport was 131GB. After filtering for potentially misclassified genomes using the output of SISTR, we were left with 2,317 genomes that were annotated as *S*. Newport and predicted as *S*. Newport genotypically (*Fig. S2* and *Fig. S3*). Specifically, SISTR-based serovar predictions suggest that 2.03% of the genomes were misclassified as Newport. Using the genotypes assigned by the MLST, cgMLST, and BAPS-based genomic composition programs implemented in ProkEvo, we next defined the relative frequency of each genotype among 2,317 isolates (*Fig. 4A-H*). This analysis identified the expected structure with four dominant STs in the following descending order: ST118, ST45, ST5, and ST132. The cgMLST distribution identified a total of 764 unique cgMLST genotypes, with the cgMLST genotype 1468400426 representing the most frequent lineage or epidemiological clone (*Fig. 4B*).

To circumvent the scalability problem of phylogeny inferred from thousands of core-genome alignments, we next examined genetic relationships of cgMLST genotypes using the scalable Bayesian-based approach in BAPS to define haplotypes based on the relative degrees of admixture in the core-genome composition at different scales of resolution. As expected, BAPS-based haplotypes at increasing levels of resolution (BAPS1-BAPS6) increasingly fragmented the *S*. Newport into: 9 sub-groups for BAPS1, 32 sub-groups for BAPS2, 83 sub-groups for BAPS3, 142 sub-groups for BAPS4, 233 sub-groups for BAPS5, and 333 sub-groups for BAPS1, discrete haplotypes (*Fig. 4C-H*). We next used a hierarchical analysis to group the *S*. Newport STs and cgMLSTs based on shared genomic admixtures at BAPS level 1 (BAPS1). At BAPS1, the lowest level of resolution, there are 9 total haplotypes. This analysis showed that the dominant BAPS1 haplotype (BAPS1 sub-group 8) is shared by two of the dominant STs, ST118 and ST5 (*Fig. S1A*). The shared BAPS haplotype implies that the two clonal complexes defined by these dominant STs are more related to each other than ST45 or ST132, which is consistent with the genetic relationships of these STs predicted by e-BURST [7]. Further analysis of the BAPS1 sub-group 8 haplotype for the major cgMLST lineages also showed 307, 149, and 23 cgMLST genotypes derived from the ST118, ST5, and ST350 clonal complexes, respectively. Having more cgMLST genotypes may suggest that ST118 is a more diverse population, which can be influenced by sample bias and size. Of note, there was not a dominant cgMLST within any of BAPS1 sub-group 8 STs 118, 5, or 350. An interesting question though would be if there is a correlation between the cgMLST diversity across STs and their ecological dispersion. Perhaps more diverse clonal complexes would be able to survive more readily in distinct habitats, say for instance bovine vs. lettuce. Consistent with the genetic relationships of STs predicted by shared BAPS1 sub-group 8 haplotypes, we also found that ST45 belongs to a distinct BAPS1 haplotype (sub-group 1), with a total of 152 cgMLST genotypes, and that the most frequent cgMLST lineage is cgMLST 1468400426, which happens to be the most dominant lineage for the entire *S*.

Newport data. This predominance of cgMLST 1468400426 within ST45 and across STs, could be due several reasons, including, but not limited to: 1) Sampling effect; 2) Recent outbreaks; 3) Founder effect with a new introduction of a clone in a population; or 4) Selective sweep at the whole-genome level in the population due to a selective advantage. Obviously, selection or founder effects can be an explanation for the emergence of epidemiological clone capable of causing an outbreak [68,69]. Our point is that these are some of the patterns one can discover when using population-based analysis, that can generate testable hypotheses of what might have happened or is occurring. That emphasizes the importance of metadata and having a carefully designed collection of isolates, because by knowing for instance temporal patterns, we can capture potential cgMLST successions in a population that might be linked to actions previously taken in a farm or food production site.

After identifying the dominant cgMLST lineage 1468400426, we assessed the degree of clonality or genotypic homogeneity of its population when compared to all other cgMLSTs combined, exclusively within BAPS1 and ST45 population (*Fig. S2A-E*). We do that by examining the frequency of sub-groups within each BAPS level from 2 to 6. To visualize the partitioning, we first select only genomes belonging to BAPS1 and ST45, and then we classified the data at each level of BAPS2-BAPS6 into two groups: one group contains cgMLST 1468400426 (numbered 1), while the second group contained all other cgMLSTs (numbered 0). If the dominant cgMLST 1468400426 is highly clonal, it will be present in one or only a few of the BAPS subgroups at each level of BAPS resolution. This is exactly what was observed in *Fig. S2A-E*, where the dominant cgMLST 1468400426 genotype was always found within a single BAPS subgroup, even at the highest level of resolution (BAPS6). Notably, at each BAPS level, there are other cgMLST genotypes that also map to the same BAPS subgroup as the dominant cgMLST 1468400426, and the frequency of these other cgMLST genotypes that share BAPS with the dominant cgMLST 1468400426 clone is essentially stable as the BAPS resolution increases. These shared BAPS subgroupings at different levels are indicative of these cgMLST genotypes sharing recent evolutionary relationships. Importantly, we are just analyzing this pattern of population stratification within BAPS1 and ST45 clonal complex. We have also found that cgMLST 1468400426 can be rarely found within ST3045 and ST4493, with only one genome of this cgMLST found in each of these two STs. That makes sense in terms of evolutionary history, because ST3045, ST3494, ST3783, and ST493 are the other STs that may have shared a recent ancestor with ST45, since they all belong to BAPS1 sub-group 1.

Collectively, this hierarchical analysis of the genomic relatedness of ST and dominant cgMLST genotypes provides a systematic way to understand population structure and evolutionary relationships of cgMLST genotypes without the need for computationally intensive phylogeny. These relationships are important as they can yield interesting hypotheses about shared ecological and epidemiological patterns among cgMLSTs that are closely related evolutionarily. It is important to reiterate that this sample of *S*. Newport genomes is from USA. Scaling this analysis to other continents across the globe could reveal what genotypes are predominant, what the relationships are with host and environmental variations, and ultimately what the genomic events associated with them are and which pathways are represented in it. All the steps of these analyses are publicly available in a Jupyter Notebook (https://github.com/npavlovikj/ProkEvo), and the files used can be found on Figshare (https://figshare.com/projects/ProkEvo/78612).

Case study 2: *S*. Typhimurium population-based analysis

S. Typhimurium is the most widespread serovar of *S. enterica* worldwide [92]. Its dominance is partially attributed to its inherited capacity to move across a variety of animal reservoirs including poultry, bovine, swine, plants and ultimately being capable of infecting humans to cause gastroenteritis or non-Typhoidal Salmonellosis [93,94]. This serovar is phenotypically divided into biphasic and monophasic sub-populations based on their expression of major flagellin proteins from both (biphasic) or only one (monophasic) of the two major flagellin genes [92]. Monophasic *S*. Typhimurium is an emerging zoonotic sub-population that is often multi-drug and heavy-metal (copper, arsenic, and silver) resistant [92,95,96]. Due to its relevance as a major zoonotic pathogen and its frequent isolation from clinical and environmental samples, *S*. Typhimurium genomes from a large number of isolates are available (23,045 genomes from various continents – not filtered for USA only). The geographical location from where the genomes were isolated could not be ascertained for this population, because of unreliability of the metadata deposited to NCBI SRA. More importantly, the *S*. Typhimurium dataset is a good measure of the scalability of ProkEvo, since it is an order of magnitude larger than *S*. Newport in the number of genomes. After the download and the pre-processing steps, 21,534 assemblies passed the filtering step. The total output data produced by ProkEvo for *S*. Typhimurium was 1.2TB.

As with *S*. Newport, we also conducted an analysis of the population structure based on MLST and cgMLST. Briefly, the reason for not including the BAPS1-6 outputs is because we have divided the dataset into smaller sub-samples for computational purposes, which due to the nature of Bayesian programming requires user input as described in the Methods section. After quality controlling and filtering the data, we ended up with 20,239 genomes of *S*. Typhimurium biphasic and monophasic combined. In order to present various ways of conducting population-based analyses using ProkEvo, for the analysis with the *S*. Typhimurium dataset we use combination of three pieces of information: 1) Whether or not the genome is classified as biphasic or monophasic based on the SISTR algorithm (.csv); 2) The ST clonal complexes calls using the legacy MLST (.csv); and 3) The cgMLST genotypic classification based on SISTR (.csv) [55]. It is important to note that SISTR makes predictions of serotypes based on genotypic information solely. In Salmonella that is possible, because of the high degree of linkage disequilibrium between the clonal frame (i.e. genome backbone) and loci that generate the O and H antigens [59]. In this dataset, 72.6%, 25%, 2.4% of the quality-controlled genomes were classified as Biphasic, Monophasic, or other serovars, respectively. From the Biphasic population, 78.4%, 9.62%, 5.35%, 2.09% of the isolates belonged to ST19, ST313, ST36, and ST34, respectively (*Fig. 5A*). Whereas, for Monophasic, 93%, 5.79%, 0.094% of the isolates belonged to ST34, ST19, and ST36, respectively (*Fig. 5A*). First, it is known that the ST34 complex predominantly comprises the population of *S*. Typhimurium Monophasic, which reflects its high degree of clonality [65]. As for the Biphasic population, ST19 dominates and likely contains the ancestor of the other ST clonal complexes. Most likely, the ST19 dominance is a consequence of its dispersal capacity and ability to spread across a variety of reservoirs, including different species of livestock, and other environments [65]. Now, ST313 has recently emerged in Africa, and is associated with non-Typhoidal Salmonellosis in humans. This host-restriction is generally associated with gene loss and auxotrophic formation in the population of the pathogen [70]. ST36 represents a minor clonal complex within *S*. Typhimurium Biphasic that appears to either be restricted ecologically, or has not had the appropriate selective force facilitating its expansion in the overall population, but it is capable of causing gastroenteritis in humans [65].

In terms of cgMLST genotypic distributions, Biphasic and Monophasic had 5,162 vs. 1,161 unique cgMLST genotypes, respectively. That is expected given the three-times larger estimated population size for Biphasic (~75%) vs. Monophasic (~25%). Notably, within the Biphasic population there was not a dominance pattern for the distribution of cgMLST lineages. However, in the Monophasic population, cgMLST 1652656062 and cgMLST 860079270 lineages comprised 32.33% and 19.62% of the isolates, respectively (*Fig. 5B*). As an attempt to explain such a scenario for Monophasic, we could list the following hypotethical reasons for such a unique pattern: 1) Founder effect – new epidemiological clones are introduced simultaneously in different locations, perhaps as part of distinct outbreaks; or 2) Parallel evolution with selection operating separately to facilitate their expansion in different reservoirs. These are important questions that we should be asking about these populations, but to switch from a hypothetical scenario to systematically developing these ideas, we need reliable metadata, and sampling done not only for clinical isolates, but also environmental ones across the food chain. For instance, if the distribution of cgMLST lineages for Biphasic were truly well-represented here, one could hypothesize that they either colonized different reservoirs or have equivalent fitness within the same environment based on their proportionality. We are using this platform to propose ideas of how we can connect computational analysis of population genomics to the biology of these microorganisms. All the steps for this analysis are shown in our Jupyter Notebook (https://github.com/npavlovikj/ProkEvo), and the input files can be found on Figshare (https://figshare.com/projects/ProkEvo/78612).

Case study 3: Distribution of known AMR loci between and within *S*. Infantis, *S*. Newport, and *S*. Typhimurium

In case study 3, we demonstrate the distribution of known AMR conferring loci based on the Resfinder database across three widely spread zoonotic serovars of *S*. *enterica* lineage I (*S*. Infantis, *S*. Newport, and *S*. Typhimurium). Our choice of only showing the Resfinder-specific results is due to its current utilization in the fields of ecology and genomic epidemiology [71,72]. However, as described in the Methods section, results for other databases are provided, and the researcher may choose a different one based on preference, or may even decide to report the results comparatively since ProkEvo gives that option. One cautionary note is that just finding an AMR gene is not sufficient to predict the phenotype accurately. For instance, deleterious mutations can happen rendering the gene afunctional, or allelic variation can generate varying degrees of resistance in the population [97]. Additionally, we have used USA-only data for *S.* Infantis for two main reasons: 1) It has a higher degree of clonality than *S.* Newport and *S.* Typhimurium which provides a contrast for population-based comparison [7,73]; and 2) It has multi-drug resistance clones recently being associated with outbreaks linked to food products [74]. It is also important to note that we have arbitrarily chosen to show data for genes above a certain threshold (>=25%) because of its potential relevance in distribution across the population, and to facilitate visualizations. Moreover, we are not accounting for the potential correlated distribution of genes across genomes for this demonstrative analysis, but that would be important in more advanced work. Linked genomic variation can mask the differentiation between causative genes to hitchhikers when studying the underlying basis for traits such as antimicrobial resistance [16]. With that being pointed out, our goal here is to show the relationship between the population structure and independent AMR loci distribution in the population.

First, the genes demonstrated here are known to confer resistance to the following classes of antibiotics: tetracyclines (*tet* genes), sulfonamides (*sul* genes), macrolides (*mdf* genes), florfenicol and chloramphenicol (*florR* and *catA* genes), trimethoprim (*dfrA* genes), beta-lactamases (*bla* family of genes), and aminoglycosides including streptomycin and spectinomycin (*aph*, *ant*, *aadA*, and *aac* genes) [98]. When comparing across serovars, we found 72, 125, and 408 unique loci for *S.* Infantis, *S.* Newport, and *S.* Typhimurium, respectively. After filtering for the most frequent ones based on our threshold, three overall points stand out: 1) *S.* Infantis population has more loci with higher frequency (> 25%); 2) *S.* Typhimurium appears to have a higher diversity of genetic elements which comes with higher sparsity as well (i.e. the majority of loci are present in very low frequency in the population); and 3) The mdf(A)_1 and aac(6')-Iaa_1 loci appear to be widespread across all serovars (*Fig. 6A*). Obviously, these pairwise comparisons are confounded by sample size, number of outbreaks, geographical distribution (USA vs. worldwide), etc. But if these results were representative of the overall population, it would be expected for *S.* Typhimurium to have a higher diversity because it is more widespread across hosts and environments, which may yield more opportunities for gene acquisition by HGT [55]. In the case of *S.* Infantis, its high clonality can be observed since the overall serovar distribution matches that of the most dominant clonal complex ST32 (*Fig. 6B*). Specifically, the total number of unique loci found in *S.* Infantis, or ST32 only, matched to 72 genes. Interestingly, the distribution of both mdf(A)_1 and aac(6')-Iaa_1 loci is very comparable across all serovars (*Fig. 6A),* and within them between major clonal complexes and others STs *(Fig. 6B-D)*. That is an indicative that those elements are more likely to be ancestrally acquired than recently derived in these populations [75]. Overall, ST118, ST5, and ST45 have 57, 33, and 84 unique loci found in them, respectively (*Fig. 6C*). Lastly, for *S.* Typhimirium, ST19, ST313, ST34, and ST36 had a total of 301, 112, 249, and 130 unique AMR loci in their populations. Given that ST19 and ST34 are the most frequent clonal complexes found in this serovar, it is not a surprise that their repertoire of genes would be higher than the others [7,65] (*Fig. 6D*). Noticeably, ST32 for *S.* Infantis, ST45 for *S.* Newport, and STs 313 and 36 for *S.* Typhimurium have a higher frequency of the most dominant genes across their populations. That is most likely a reflex of the high degree of clonality for those clonal complexes. As stated before, high degree of population homogeneity can be an artefact of oversampling clinical isolates during outbreaks without accounting for the overall environmental diversity. Also, it is important to mention that we are not differentiating between genes present in chromosome vs. plasmids. The latter are more promiscuous and facilitate HGT between closely related, or divergent populations [99].

Case study 4: Population structure and AMR loci distribution for *C. jejuni* and *S. aureus*

In contrast to *S.* Infantis, *S.* Newport, and *S.* Typhimurium, *C. jejuni* has a more diverse population at the level of clonal complexes (STs) (*Fig. 7A*). Visibly, we can have a higher number of dominant STs which, in parts, reflect the more accentuated degree of HGT of this species compared to *S. enterica* and *S. aureus*, and the impact of host-associated diversification [60]. At least some of *C. jejuni* STs appear to behave similarly to *S.* Typhimurium by having a somewhat generalist behavior in terms of host distribution, but host-specialization can occur as well. For instance, ST21 can be found in the gastrointestinal tract of poultry and humans; whereas, ST45 can be found in the gastrointestinal tract of bovine and humans; but that does not prevent their movement across other livestock species. This potential for ecological encounter in a reservoir would facilitate the occurrence of HGT, which in turn creates a degree of admixture in the population [76,77]. By consequence, drawing true phylogenetic relationships becomes cumbersome because of the impact of recombination events on the clonal frame [78]. Of note, we chose to show STs with a proportion higher than 1% in order to facilitate visualization for both *C. jejuni* and *S. aureus*. Contrary to *C. jejuni*, *S. aureus* has a higher degree of clonality, which can be seen based on having fewer dominant STs, and with STs 8, 5, and 105 comprising more than 80% of the population (*Fig. 7B*). ST8 is known to be associated with community-acquired infections in the form of either methicillin susceptible or resistant strains (MSSA or MRSA) [79]. ST5 can also cause skin infections and is often found as MRSA [80]; whereas, ST105 is closely related to ST5 and both can carry the *SCCmec* element II [81]. In terms of AMR loci, we found 256 vs. 164 unique genetic elements for *C. jejuni* and *S. aureus*, respectively. Within *C. jejuni*, the top 8 most frequent STs had the following total number of loci: ST353 (29), ST45 (30), ST982 (20), ST48 (24), ST50 (31), ST8 (20), ST806 (19), and ST459 (15). As for *S. aureus*, the top 6 most frequent STs had the following total number of loci: ST8 (88), ST5 (85), ST105 (52), ST398 (39), ST609 (20), and ST45 (24). Of note, identical ST numbers across different bacterial species do not belong to the same population. ST numbers are both data- and species-dependent.

In the *C. jejuni* data we can see an overall trend for widespread distribution of two genes: tet(O)_1 and blaOXA-193_1, which confer resistance to tetracyclines and beta-lactamases, respectively (*Fig. 7C*). A parsimonious explanation for it would be that these genes are vertically acquired by an ancestral population, and consequently lost many independent times due to drift or selection across different clonal complexes [82]. This idea is corroborated by the diversity and dispersion shown in the overlaid disposition of the core-genome phylogeny of *C. jejuni* with BAPS1 and ST hierarchical groupings (*Fig. 8A*). BAPS1 sub-groups are comprised of unique dominant STs that are scattered around the tree, instead of having closely related STs sharing sub-groups which would indicate the presence of very recent common ancestors across them. Hence, in such scenario, genes that are in higher frequency across divergent populations are more likely to have been acquired vertically from a common ancestor, rather than independently while STs diversify in the environment. But those are not mutually exclusive scenarios, and these data cannot prove or the other. In contrast, the *cfr(C)_1* locus appears uniquely in the ST806 clonal complex when comparing across the dominant STs, suggesting a more recent acquisition of this gene. The *cfr* gene is of extreme relevance because it has a pleiotropic effects, conferring resistance to a variety of AMR classes, such as: phenicol, lincosamide, oxazolidinone, pleuromutilin, streptogramin A, and other macrolides [83]. Of note, the phylogenetic tree calculated here did not account for HGT, which can be a confounding factor for accurately estimating evolutionary relationships for highly recombining species such as *C. jejuni*. Removing putative recombining regions from core-genome alignment belonging to divergent STs, while scaling the analysis, is a computational problem yet to be solved.

In the case of *S.* aureus, we see a similar trend in the distribution of the most common AMR genes for STs 5 and 105 (*Fig. 7D*), which are confirmed to be more closely related to each other than the other dominant STs, based on them being part of BAPS1 sub-group 5 (*Fig. 8B*). ST8 and ST609 also share evolutionary history, since they belong to BAPS1 sub-group 6 (*Fig. 8B*). Now, ST398 and ST45 pertain to BAPS1 sub-groups 1 and 4, respectively. This potential differential ancestral pattern is somewhat reflected on the overall distribution of AMR genes for *S. aureus* (*Fig. 7D*). In contract to *C. jejuni*, there is not a common trend across STs with the exception of the *mecA_6* locus. That pattern suggests that some of these elements are being acquired independently by HGT, which includes plasmid transmission as well, and perhaps, some are acquired vertically by loss across generations. Having multiple STs as part of a single BAPS1 sub-group reinforces the knowledge that *S. aureus* is more clonal than *C. jejuni*, for instance. Another interesting statistic is that, *C. jejuni* contains 24 BAPS level 1 sub-groups as opposed to only 7 being present in the *S. aureus* population. Even though we have selected USA genomes for both species, there are many other ecological and epidemiological factors limiting our interpretation of the data. Interestingly, when compared to the three *S. enterica* lineage I serovars and *C. jejuni*, *S. aureus* population has some unique loci that comprise the list of most prevalent ones such as those associated to resistance to: 1) Erythromycin and streptogramin B (*msr*, *mph,* and *erm* genes); 2) Penicillin and methicillin (*mecA* and *blaZ* family of genes); and 3) Fosfomycin (*fosD* gene) [98]. To some extent that reflects the biology of those organisms with *S. aureus* being the only gram positive, but perhaps that could also be explained with this species being able to colonize a different ecological habitat such as the mammary gland of bovine and nasal cavity of humans and livestock [84]. It is worth reinforcing that we cannot differentiate between genetic elements present in either the bacterial chromosome or plasmid based on the analysis presented here. It would be intuitive to expect genes that are in high frequency across very divergent STs to be in the chromosome, but it is also possible that a common plasmid containing the locus is shared across them, or the gene is widespread across various distinct plasmids [99].

Discussion

The continuous increase in the volume of WGS data is practically driving the field of bacterial genomics towards implementing large-scale data science approaches to learn from the data. Mining bacterial population-based datasets through genomics can be very revealing of the population structure, geographical and temporal distributions, and epidemiological patterns that may reflect adaptive evolution and ecological adaptation [3,4,7,10,16]. However, scaling and automating WGS analyses can be a challenging task that comes with its own costs and benefits. The trade-off of automating is that users end-up relying on underlying "black-box" to generate data without considering parameter tuning and optimization very seriously. On the other hand, there is a large number of biology/microbiology laboratories that can immediately benefit from such automation to generate a variety of hypotheses that can then be tested more rigorously with *in vitro* and *in vivo* experimentation approaches. ProkEvo fills that gap by allowing researchers to scale the analyses from hundreds to many thousands of genomes without having to write scripts and programs from scratch. ProkEvo takes advantage of a set of well-developed and robust bioinformatics tools that combined produce a reproducible, and scalable workflow.

ProkEvo is modular – when feasible, each genome is analyzed independently. In theory, if a dataset has *n* genomes and a computational platform has *n* available cores, ProkEvo can easily scale linearly and utilize all these resources at the same time using execution platforms such as clusters and grids. By using the already existing pipeline for ProkEvo, modifying and expanding it with additional steps, tools, and databases becomes straight-forward. ProkEvo only needs a list of NCBI SRA (genome) identifications as an input, and Pegasus submit script. The computational resources used for the steps in ProkEvo are specified per tool and are not fixed. This is an important feature of ProkEvo that allows faster allocation of resources and requiring high resources only when needed. While the scripts for executing the tools in ProkEvo are written to consider possible errors with the program, such as bad data or exceptions, failures due to rare cases are still possible. In this case, only the failed job is retried, and possibly terminated. This individual failure does not affect the continuity of the pipeline and the remaining jobs keep running. This is really useful especially when analyzing large datasets, in which out of tens of thousands of genomes, few may have faulty reads and should not have an impact over the rest of the workflow. These are only a few of the advantages of ProkEvo. Most of them come as a consequence of using robust, reliable, and automated workflow management system such as Pegasus. Pegasus WMS has been used for development of small and large-scale processing and computational pipelines for various projects. Some of these projects include the LIGO gravitational wave detection analysis [51], the structural protein-ligand interactome (SPLINTER) project [85], the Soybean Knowledge Base (SOyKB) pipeline [86], the Montage project for generating science-grade mosaics of the sky [87]. The scalability and handling large sets of data and computations, the portability to different computational platforms, and its ease of use are just few of the reasons why we chose Pegasus WMS to develop ProkEvo.

Besides ProkEvo, several other automated pipelines for analyses of bacterial genomes have been developed over the years, such as EnteroBase [17], TORMES [18], Nullarbor [19], and ASA3P [20]. EnteroBase is an online resource for identifying and visualizing bacterial species-specific genotypes at scale by utilizing a high-performance cluster at the University of Warwick. TORMES is a whole bacterial genome sequencing pipeline that works with raw Illumina paired-end reads, and is written in Bash. Nullarbor is a Perl pipeline for performing analyses and generating web reports of bacterial sequenced isolates for public health microbiology laboratories. ASA3P is an automated and scalable assembly annotation and analyses pipeline for bacterial genomes written in Groovy. While some of these pipelines' future plans are to use robust workflow management systems, to the best of our knowledge none of them is using one yet. Moreover, these computational platforms have been tested using tens to a few thousands of genomes in general. This is sufficient for some research questions, and the existing pipelines can perform well on this scale. However, for understanding ecological and evolutionary patterns of populations, analyzing moderate to large scale genomic datasets of a population is needed. As of today, *S. enterica*, *C. jejuni*, and *S. aureus* have more than 300,000, 50,000, and 70,000 genomes available, respectively. Performing analyses on such an enormous scale, and tracking steps, data and errors is a challenging task that requires not only using scalable programming languages and advanced computational approaches, but powerful execution platforms as well. ProkEvo efficiently addresses some of these issues with using reliable and robust management system and high-throughput and high-performance computational platforms. However, future testing needs to be done to evaluate and improve ProkEvo's performance with more than hundredths of thousands of genomes, and its portability to cloud environments such as the Amazon Web Service. Of note, one particular bottleneck is generating core-genome alignments with Roary. This step is important since it precedes population structure analysis using fastbaps or doing phylogenetics. However, this step can run indefinitely when the number of genomes is large, which is often the case. One possible workaround is to randomly divide the dataset into samples of up to 2,000 genomes, which allows ProkEvo to perform all jobs efficiently. However, that comes with some consequences: 1) fastbaps uses Bayesian computations which may prevent direct data aggregation afterwards; 2) The user will have to generate multiple phylogenetic trees; and 3) Pan-genome annotation may vary in gene identity with inconsistent callings, which particularly affects the identification of hypothetical proteins. However, there are other computational approaches that can be used for phylogenetic inference such as kmer-based construction of distance matrices using assemblies directly [100]. Although these can be hurdles, we anticipate that novel algorithmic approaches in addition to large-scale computing will facilitate the generation of novel solutions for these problems. An advantage of ProkEvo is that by using the Pegasus workflow, novel software can be added to the platform without disrupting any pre-established tasks. Hence, users should be able to incorporate new solutions or alternative steps or programs easily.

Analyzing data more rapidly and automatically solves only part of the problem. We still, as a community, need to learn how to mine these data in light of principles of population genetics and ecology, in addition to using more modern tools such as machine learning and pattern searching algorithms [101,102,103]. Only a combination of these philosophies can accelerate our discovery rate regarding the biology of these microorganisms at the population level. As such, we provide a preliminary guidance on how to examine the population structure of bacteria using varying genotypic resolutions. Our approach shows how to find population-based patterns when analyzing the frequency distribution of genotypes at different scales. Of note, these varying levels of genotypic resolution are fundamentally based on mining the shared genomic variations present in the core-genome (i.e. ubiquitous loci spread across the entire or vast majority (> 99%) of a given species-specific bacterial population). By identifying high-frequency sub-populations we can then search for genes that are uniquely present (i.e. loci present in the accessory genome), or over-represented in them. This approach can be useful in revealing the pathways that may be essential for major epidemiological clones, pathogenic variants, or clonal complexes, to spread successfully through animal and environmental reservoirs [104]. For instance, clinical isolates of *C. jejuni* clonal complexes ST21 and ST45 appear to preferentially have acquired loci conferring the capacity to proliferate in the presence of oxygen, in addition to utilizing formate and savaging nucleotides, which in turn maximizes their survival and spread across the poultry food chain [10]. This is example of how specific populations can have a fitness advantage by acquiring niche-transcending genes, since aerobic respiration is not a particular attribute of a single macro- or micro-habitat. The identification of niche-transcending vs. niche-specifying genes can be very informative of different ecological attributes present in a bacterial population. Population-based selective sweeps (i.e. purged genomic variation at the whole genome level) can happen by a simple acquisition of a locus or loci capable of providing novel physiological or pathogenic capacity [75]. This could be reflected on a temporal change of cgMLST epidemiological clones in a population, whereby a single cgMLST takes over, and comparative population genomics links unique accessory loci to the genome backbone of that lineage. By linking the genotypic variation to reliable epidemiological information, we might be able to discern and experimentally test which selective factors contributed to such a dynamic. Clearly, having reliable and accurate metadata for such a modeling approach would not only be enriching, but crucial. Currently, we are limited to the meta information the public databases are populated with. This is indeed a major factor that needs to be addressed by the community at large. We need a minimal amount of useful and reliable epidemiological data while considering data privacy and litigation issues.

Altogether, we believe that creating an automated, robust, and scalable platform for carrying out population-based analysis can maximize our discoveries and aid in the development of hypothesis-driven work and epidemiological surveys of pathogens. This powerful combination of population-based pattern searching with experimentation may provide new insights of the evolution of these populations, and perhaps yield novel applications for surveillance and disease mitigation in the case of major foodborne pathogens such as *S. enterica* lineage I, *C. jejuni*, and *S. aureus*. Similarly, this approach can be used for other bacterial species, such as beneficial microbes that are or can be putative probiotic candidates. In general, our platform aims at leveraging the microorganismal population structure to identifying patterns that can be useful for understanding ecological and evolutionary processes shaping populations. This top-down based analysis has the advantage of using agnostic principles and inquiries to learn from the large-scale data in order to get novel insights about the fundamental biology of the species, while discovering novel and practical information. However, this is only possible because ProkEvo allows us to conduct the analysis in a reproducible, scalable and expandable fashion, permitting us to identify novel population patterns with different levels of resolution.

Conclusions

In this paper we present **ProkEvo**, which is: 1) An automated, user-friendly, reproducible, and open-source pipeline for bacterial population genomics analyses that uses the Pegasus Workflow Management System; 2) Pipeline that can scale the analysis from at least a few to tens of thousands of bacterial genomes using high-performance and high-throughput computational resources; 3) An easily modifiable and expandable pipeline to include additional steps, custom scripts and software, user databases, and species-specific data; 4) Modular pipeline that can run many thousands of analyses concurrently, if the resources are available; 5) Pipeline for which the memory and run time allocations are specified per job, and automatically increases its memory in the next retry; 6) Distributed with conda environment and Docker image for all bioinformatics tools and databases needed to perform population genomics analyses; and ultimately includes: 7) An initial guidance on how to perform population-based analyses using its output files with reproducible Jupyter Notebooks and R scripts. One important advantage of ProkEvo is its adaptability to the user needs. Also, we intend to keep on improving this pipeline to include new computational branches that will potentially add the following functionality: 1) cgMLST genotyping for non-Salmonella genomes; and 2) Integrating the population-based analysis and predictive pan-genome computations to identify genes uniquely present in sub-populations defined based on STs, cgMLSTs, etc. These functions can add tremendous value to research and clinical microbiological purposes. First, cgMLST genotyping is directly applicable for epidemiological surveillance of populations. Finally, an automated population-based and pan-genome analyses can allow researchers and clinical microbiologists to find unique genes that are enriched in a target population, which may in turn reflect past selection and ecological adaptation to a particular environment or host. Ideally, we, as a community would have access to a minimal amount of epidemiological information that would facilitate discovering novel potential genomic signatures associated with different environments and hosts. While the latter remains a large issue to be dealt with, ProkEvo has the potential to be implemented as an open-source science gateway, which remains a long-term goal.

Acknowledgements

This work was supported by funding from the IANR Agricultural Research Division and the National Institute for Antimicrobial Resistance Research and Education. This work was completed by utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative, and using resources provided by the Open Science Grid, which is supported by the National Science Foundation and the U.S. Department of Energy's Office of Science. We would like to greatly thank Mats Rynge for his extensive assistance and valuable suggestions while setting up and running ProkEvo on the Open Science Grid. We also thank Dr. Derek Weitzel and Karan Vahi for their technical support.

This paper is dedicated to the memory of Dr. David Swanson, the former director of the Holland Computing Center, who passed away before this project was completed. Dr. David Swanson was an amazing individual, a true leader and an inspiration to us all. He was a strong advocate of computational research and literacy, and passionate about finding ways to do large-scale science attainably, better, and faster. Our collaboration and shared endeavors would not have been possible without him.

References

1. Quainoo S, Coolen JPM, van Hijum SAFT, Huynen MA, Melchers WJG, van Schaik W, et al. Whole-Genome Sequencing of Bacterial Pathogens: the Future of Nosocomial Outbreak Analysis. Clinical Microbiology Reviews. 2017 Aug 30;30(4):1015–63.
2. Pallen M, Wren B. Bacterial pathogenomics. Nature. 2007;449(7164):835-842.
3. Sheppard S, Guttman D, Fitzgerald J. Population genomics of bacterial host adaptation. Nature Reviews Genetics. 2018;19(9):549-565.
4. Joseph S, Read T. Bacterial population genomics and infectious disease diagnostics. Trends in Biotechnology. 2010;28(12):611-618.
5. Land M, Hauser L, Jun S, Nookaew I, Leuze M, Ahn T et al. Insights from 20 years of bacterial genome sequencing. Functional & Integrative Genomics. 2015;15(2):141-161.
6. Zhou Z, Alikhan N, Sergeant M, Luhmann N, Vaz C, Francisco A et al. GrapeTree: visualization of core genomic relationships among 100,000 bacterial pathogens. Genome Research. 2018;28(9):1395-1404.
7. Alikhan N, Zhou Z, Sergeant M, Achtman M. A genomic overview of the population structure of Salmonella. PLOS Genetics. 2018;14(4):e1007261.
8. Dallman T, Byrne L, Ashton P, Cowley L, Perry N, Adak G et al. Whole-Genome Sequencing for National Surveillance of Shiga Toxin–Producing Escherichia coliO157. Clinical Infectious Diseases. 2015;61(3):305-312.
9. Croucher N, Coupland P, Stevenson A, Callendrello A, Bentley S, Hanage W. Diversification of bacterial genome content through distinct mechanisms over different timescales. Nature Communications. 2014;5(1).
10. Yahara K, Méric G, Taylor A, de Vries S, Murray S, Pascoe B et al. Genome-wide association of functional traits linked with Campylobacter jejuni survival from farm to fork. Environmental Microbiology. 2017;19(1):361-380.
11. McDermott PF, Tyson GH, Kabera C, Chen Y, Li C, Folster JP, et al. Whole-Genome Sequencing for Detecting Antimicrobial Resistance in Nontyphoidal Salmonella. Antimicrobial Agents and Chemotherapy. 2016 Jul 5;60(9):5515–20.
12. Laabei M, Recker M, Rudkin J, Aldeljawi M, Gulay Z, Sloan T et al. Predicting the virulence of MRSA from its genome sequence. Genome Research. 2014;24(5):839-849.
13. Ingle DJ, Valcanis M, Kuzevski A, Tauschek M, Inouye M, Stinear T, et al. In silico serotyping of E. coli from short read data identifies limited novel O-loci but extensive diversity of O:H serotype combinations within and between pathogenic lineages. Microbial Genomics. 2016 Jul 11;2(7).
14. Yoshida CE, Kruczkiewicz P, Laing CR, Lingohr EJ, Gannon VPJ, Nash JHE, et al. The Salmonella In Silico Typing Resource (SISTR): An Open Web-Accessible Tool for Rapidly Typing and Subtyping Draft Salmonella Genome Assemblies. Hensel M, editor. PLOS ONE. 2016 Jan 22;11(1):e0147101.

15. Sheppard SK, Jolley KA, Maiden MCJ. A Gene-By-Gene Approach to Bacterial Population Genomics: Whole Genome MLST of Campylobacter. Genes. 2012 Apr 12;3(2):261–77.
16. Power R, Parkhill J, de Oliveira T. Microbial genome-wide association studies: lessons from human GWAS. Nature Reviews Genetics. 2016;18(1):41-50.
17. Zhou Z, Alikhan N-F, Mohamed K, Fan Y, Achtman M. The EnteroBase user's guide, with case studies on Salmonella transmissions, Yersinia pestis phylogeny, and Escherichia core genomic diversity. Genome Research. 2019 Dec 6;30(1):138–52.
18. Quijada NM, Rodríguez-Lázaro D, Eiros JM, Hernández M. TORMES: an automated pipeline for whole bacterial genome analysis. Valencia A, editor. Bioinformatics. 2019 Apr 8;35(21):4207–12.
19. Seemann T, Goncalves da Silva A, Bulach DM, Schultz MB, Kwong JC, Howden BP. Nullarbor. GitHub. 2020. Available: https://github.com/tseemann/nullarbor.
20. Schwengers O, Hoek A, Fritzenwanker M, Falgenhauer L, Hain T, Chakraborty T, Goesmann A. ASA3P: An automatic and scalable pipeline for the assembly, annotation and higher level analysis of closely related bacterial isolates. PLoS computational biology. 2020 Mar 5;16(3):e1007134.
21. Koster J, Rahmann S. Snakemake--a scalable bioinformatics workflow engine. Bioinformatics. 2012 Aug 20;28(19):2520–2.
22. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature Biotechnology. 2017 Apr;35(4):316–9.
23. Apache Airflow. Apache Airflow. Available: http://airflow.incubator.apache.org/.
24. Deelman E, Singh G, Su M-H, Blythe J, Gil Y, Kesselman C, et al. Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems. Scientific Programming. 2005;13(3):219–37.
25. HCC. Holland Computing Center | Nebraska. Available: https://hcc.unl.edu/.
26. Towns J, Cockerill T, Dahan M, Foster I, Gaither K, Grimshaw A, et al. XSEDE: Accelerating Scientific Discovery. Computing in Science & Engineering. 2014 Sep;16(5):62–74.
27. Langmead B, Nellore A. Cloud computing for genomic data analysis and collaboration. Nature Reviews Genetics. 2018 Jan 30;19(4):208–19.
28. Pordes R, Petravick D, Kramer B, Olson D, Livny M, Roy A, et al. The open science grid. Journal of Physics: Conference Series. 2007 Jul 1;78:12057.
29. Sfiligoi I, Bradley DC, Holzman B, Mhashilkar P, Padhi S, Wurthwein F. The Pilot Way to Grid Resources Using glideinWMS. In: 2009 WRI World Congress on Computer Science and Information Engineering. IEEE; 2009.
30. Leinonen R, Sugawara H, Shumway M. The Sequence Read Archive. Nucleic Acids Research. 2010 Nov 9;39(Database):D19–21.

31. Valieris R. parallel-fastq-dump. GitHub. 2020. Available: https://github.com/rvalieris/parallel-fastq-dump.

32. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics. 2014 Apr 1;30(15):2114–20.

33. Andrews S. FASTQC. A quality control tool for high throughput sequence data. 2010.

34. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. Journal of Computational Biology. 2012 May;19(5):455–77.

35. Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome assemblies. Bioinformatics. 2013 Feb 19;29(8):1072–5.

36. Carattoli A, Zankari E, García-Fernández A, Voldby Larsen M, Lund O, Villa L, et al. In SilicoDetection and Typing of Plasmids using PlasmidFinder and Plasmid Multilocus Sequence Typing. Antimicrobial Agents and Chemotherapy. 2014 Apr 28;58(7):3895–903.

37. Seemann T. Prokka: rapid prokaryotic genome annotation. Bioinformatics. 2014 Mar 18;30(14):2068–9.

38. Page AJ, Cummins CA, Hunt M, Wong VK, Reuter S, Holden MTG, et al. Roary: rapid large-scale prokaryote pan genome analysis. Bioinformatics. 2015 Jul 20;31(22):3691–3.

39. Tonkin-Hill G, Lees JA, Bentley SD, Frost SDW, Corander J. Fast hierarchical Bayesian analysis of population structure. Nucleic Acids Research. 2019 May 11;47(11):5539–49.

40. Seemann T. MLST. GitHub. 2020. Available: https://github.com/tseemann/mlst.

41. Jolley KA, Maiden MC. BIGSdb: Scalable analysis of bacterial genome variation at the population level. BMC Bioinformatics. 2010 Dec;11(1).

42. Seemann T. ABRicate. GitHub. 2020. Available: https://github.com/tseemann/abricate.

43. Feldgarden M, Brover V, Haft DH, Prasad AB, Slotta DJ, Tolstoy I, et al. Validating the AMRFinder Tool and Resistance Gene Database by Using Antimicrobial Resistance Genotype-Phenotype Correlations in a Collection of Isolates. Antimicrobial Agents and Chemotherapy. 2019 Aug 19;63(11).

44. Jia B, Raphenya AR, Alcock B, Waglechner N, Guo P, Tsang KK, et al. CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. Nucleic Acids Research. 2016 Oct 26;45(D1):D566–73.

45. Gupta SK, Padmanabhan BR, Diene SM, Lopez-Rojas R, Kempf M, Landraud L, et al. ARG-ANNOT, a New Bioinformatic Tool To Discover Antibiotic Resistance Genes in Bacterial Genomes. Antimicrobial Agents and Chemotherapy. 2013 Oct 21;58(1):212–20.

46. Zankari E, Hasman H, Cosentino S, Vestergaard M, Rasmussen S, Lund O, et al. Identification of acquired antimicrobial resistance genes. Journal of Antimicrobial Chemotherapy. 2012 Jul 10;67(11):2640–4.

47. Chen L, Zheng D, Liu B, Yang J, Jin Q. VFDB 2016: hierarchical and refined dataset for big data analysis—10 years on. Nucleic Acids Research. 2015 Nov 17;44(D1):D694–7.

48. Anaconda | The World's Most Popular Data Science Platform. Anaconda. Available: https://www.anaconda.com/.
49. Empowering App Development for Developers | Docker. Available: https://www.docker.com/.
50. Computing with HTCondor. HTCondor. Available: http://research.cs.wisc.edu/htcondor.
51. Usman SA, Nitz AH, Harry IW, Biwer CM, Brown DA, Cabero M, et al. The PyCBC search for gravitational waves from compact binary coalescence. Classical and Quantum Gravity. 2016 Oct 10;33(21):215004.
52. Wickham H. ggplot2. Wiley Interdisciplinary Reviews: Computational Statistics. 2011;3(2):180-185.
53. Price M, Dehal P, Arkin A. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. PLoS ONE. 2010;5(3):e9490.
54. Abebe E, Gugsa G, Ahmed M. Review on Major Food-Borne Zoonotic Bacterial Pathogens. Journal of Tropical Medicine. 2020;2020:1-19.
55. Ferrari R, Rosario D, Cunha-Neto A, Mano S, Figueiredo E, Conte-Junior C. Worldwide Epidemiology of Salmonella Serovars in Animal-Based Foods: a Meta-analysis. Applied and Environmental Microbiology. 2019;85(14).
56. Rowe B, Hall ML. Kauffman-White scheme. Public Health Laboratory Service, London, UK. 1989.
57. Snapshots of Salmonella Serotypes | Salmonella Atlas | Reports and Publications | Salmonella | CDC. Available: https://www.cdc.gov/salmonella/reportspubs/salmonella-atlas/serotype-snapshots.html.
58. Connor T, Owen SV, Langridge G, Connell S, Nair S, Reuter S, Dallman TJ, Corander J, Tabing KC, Le Hello S, Fookes M. What's in a name? Species wide whole genome sequencing resolves invasive and non-invasive Salmonella Paratyphi B. mBio. 2016 Aug 23;7(4).
59. Moradigaravand D, Gouliouris T, Blane B, Naydenova P, Ludden C, Crawley C, Brown NM, Török ME, Parkhill J, Peacock SJ. Within-host evolution of Enterococcus faecium during longitudinal carriage and transition to bloodstream infection in immunocompromised patients. Genome medicine. 2017 Dec;9(1):1-1.
60. Sheppard SK, Maiden MC. The evolution of Campylobacter jejuni and Campylobacter coli. Cold Spring Harbor perspectives in biology. 2015 Aug 1;7(8):a018119.
61. Outbreaks Involving Campylobacter | CDC. Available: https://www.cdc.gov/campylobacter/outbreaks/outbreaks.html.
62. Griekspoor P, Colles FM, McCarthy ND, Hansbro PM, Ashhurst-Smith C, Olsen B, Hasselquist D, Maiden MC, Waldenström J. Marked host specificity and lack of phylogeographic population structure of Campylobacter jejuni in wild birds. Molecular ecology. 2013 Mar;22(5):1463-72.

63. Tong SY, Davis JS, Eichenberger E, Holland TL, Fowler VG. Staphylococcus aureus infections: epidemiology, pathophysiology, clinical manifestations, and management. Clinical microbiology reviews. 2015 Jul 1;28(3):603-61.

64. Fetsch A, Johler S. Staphylococcus aureus as a foodborne pathogen. Current Clinical Microbiology Reports. 2018 Jun 1;5(2):88-96.

65. Bawn M, Alikhan NF, Thilliez G, Kirkwood M, Wheeler NE, Petrovska L, Dallman TJ, Adriaenssens EM, Hall N, Kingsley RA. Evolution of Salmonella enterica serotype Typhimurium driven by anthropogenic selection and niche adaptation. Plos Genetics. 2020 Jun 8;16(6):e1008850.

66. Mourkas E, Florez-Cuadrado D, Pascoe B, Calland JK, Bayliss SC, Mageiros L, Méric G, Hitchings MD, Quesada A, Porrero C, Ugarte-Ruiz M. Gene pool transmission of multidrug resistance among Campylobacter from livestock, sewage and human disease. Environmental microbiology. 2019 Dec;21(12):4597-613.

67. Holden MT, Hsu LY, Kurt K, Weinert LA, Mather AE, Harris SR, Strommenger B, Layer F, Witte W, de Lencastre H, Skov R. A genomic portrait of the emergence, evolution, and global spread of a methicillin-resistant Staphylococcus aureus pandemic. Genome research. 2013 Apr 1;23(4):653-64.

68. Grad YH, Lipsitch M, Feldgarden M, Arachchi HM, Cerqueira GC, FitzGerald M, Godfrey P, Haas BJ, Murphy CI, Russ C, Sykes S. Genomic epidemiology of the Escherichia coli O104: H4 outbreaks in Europe, 2011. Proceedings of the national academy of sciences. 2012 Feb 21;109(8):3065-70.

69. Fraser C, Hanage WP, Spratt BG. Neutral microepidemic evolution of bacterial pathogens. Proceedings of the National Academy of Sciences. 2005 Feb 8;102(6):1968-73.

70. Okoro CK, Barquist L, Connor TR, Harris SR, Clare S, Stevens MP, Arends MJ, Hale C, Kane L, Pickard DJ, Hill J. Signatures of adaptation in human invasive Salmonella Typhimurium ST313 populations from sub-Saharan Africa. PLoS Negl Trop Dis. 2015 Mar 24;9(3):e0003611.

71. Perron GG, Whyte L, Turnbaugh PJ, Goordial J, Hanage WP, Dantas G, Desai MM. Functional characterization of bacteria isolated from ancient arctic soil exposes diverse resistance mechanisms to modern antibiotics. PLoS One. 2015 Mar 25;10(3):e0069533.

72. Cooper AL, Low AJ, Koziol AG, Thomas MC, Leclair D, Tamber S, Wong A, Blais BW, Carrillo CD. Systematic Evaluation of Whole Genome Sequence-Based Predictions of Salmonella Serotype and Antimicrobial Resistance. Frontiers in Microbiology. 2020 Apr 3;11:549.

73. Gymoese P, Kiil K, Torpdahl M, Østerlund MT, Sørensen G, Olsen JE, Nielsen EM, Litrup E. WGS based study of the population structure of Salmonella enterica serovar Infantis. BMC genomics. 2019 Dec 1;20(1):870.

74. Kawakami V, Bottichio L, Lloyd J, Carleton H, Leeper M, Olson G, Li Z, Kissler B, Angelo KM, Whitlock L, Sinatra J. Multidrug-Resistant Salmonella I 4,[5], 12: i:– and

Salmonella Infantis Infections Linked to Whole Roasted Pigs from a Single Slaughter and Processing Facility. Journal of food protection. 2019 Sep;82(9):1615-24.

75. Cohan FM. Transmission in the origins of bacterial diversity, from ecotypes to phyla. Microbial Transmission. 2019 Mar 1:311-43.

76. Berthenet E, Thépault A, Chemaly M, Rivoal K, Ducournau A, Buissonnière A, Bénéjat L, Bessède E, Mégraud F, Sheppard SK, Lehours P. Source attribution of Campylobacter jejuni shows variable importance of chicken and ruminants reservoirs in non-invasive and invasive French clinical isolates. Scientific reports. 2019 May 30;9(1):1-8.

77. Sheppard SK, Didelot X, Meric G, Torralbo A, Jolley KA, Kelly DJ, Bentley SD, Maiden MC, Parkhill J, Falush D. Genome-wide association study identifies vitamin B5 biosynthesis as a host specificity factor in Campylobacter. Proceedings of the national academy of sciences. 2013 Jul 16;110(29):11923-7.

78. Hanage WP, Fraser C, Spratt BG. Fuzzy species among recombinogenic bacteria. BMC biology. 2005 Dec;3(1):1-7.

79. Glaser P, Martins-Simões P, Villain A, Barbier M, Tristan A, Bouchier C, Ma L, Bes M, Laurent F, Guillemot D, Wirth T. Demography and intercontinental spread of the USA300 community-acquired methicillin-resistant Staphylococcus aureus lineage. MBio. 2016 Mar 2;7(1).

80. Baines SL, Howden BP, Heffernan H, Stinear TP, Carter GP, Seemann T, Kwong JC, Ritchie SR, Williamson DA. Rapid emergence and evolution of Staphylococcus aureus clones harboring fusC-containing staphylococcal cassette chromosome elements. Antimicrobial agents and chemotherapy. 2016 Apr 1;60(4):2359-65.

81. Challagundla L, Reyes J, Rafiqullah I, Sordelli DO, Echaniz-Aviles G, Velazquez-Meza ME, Castillo-Ramírez S, Fittipaldi N, Feldgarden M, Chapman SB, Calderwood MS. Phylogenomic classification and the evolution of clonal complex 5 methicillin-resistant Staphylococcus aureus in the Western Hemisphere. Frontiers in Microbiology. 2018 Aug 22;9:1901.

82. Bobay LM, Ochman H. Factors driving effective population size and pan-genome evolution in bacteria. BMC evolutionary biology. 2018 Dec;18(1):1-2.

83. Atkinson GC, Hansen LH, Tenson T, Rasmussen A, Kirpekar F, Vester B. Distinction between the Cfr methyltransferase conferring antibiotic resistance and the housekeeping RlmN methyltransferase. Antimicrobial agents and chemotherapy. 2013 Aug 1;57(8):4019-26.

84. Roberson JR, Fox LK, Hancock DD, Gay JM, Besser TE. Ecology of Staphylococcus aureus isolated from various sites on dairy farms. Journal of dairy science. 1994 Nov 1;77(11):3354-64.

85. Quick R, Hayashi S, Meroueh S, Rynge M, Teige S, Wang B, et al. Building a Chemical-Protein Interactome on the Open Science Grid. Proceedings of Science, International Symposium on Grids and Clouds (ISGC) 2015, 2015.

86. Liu Y, Khan SM, Wang J, Rynge M, Zhang Y, Zeng S, et al. PGen: large-scale genomic variations analysis workflow and browser in SoyKB. BMC Bioinformatics. 2016 Oct;17(S13).
87. Berriman GB, Deelman E, Good JC, Jacob JC, Katz DS, Kesselman C, et al. Montage: a grid-enabled engine for delivering custom science-grade mosaics on demand. In: Optimizing Scientific Return for Astronomy through Information Technologies. SPIE; 2004.
88. Fookes M, Schroeder G, Langridge G, Blondel C, Mammina C, Connor T et al. Salmonella bongori Provides Insights into the Evolution of the Salmonellae. PLoS Pathogens. 2011;7(8):e1002191.
89. Achtman M, Wain J, Weill F, Nair S, Zhou Z, Sangal V et al. Multilocus Sequence Typing as a Replacement for Serotyping in Salmonella enterica. PLoS Pathogens. 2012;8(6):e1002776.
90. Cury J, Oliveira P, de la Cruz F, Rocha E. Host Range and Genetic Plasticity Explain the Coexistence of Integrative and Extrachromosomal Mobile Genetic Elements. Molecular Biology and Evolution. 2018;35(9):2230-2239.
91. Schneider J, White P, Weiss J, Norton D, Lidgard J, Gould L et al. Multistate Outbreak of Multidrug-Resistant Salmonella Newport Infections Associated with Ground Beef, October to December 2007. Journal of Food Protection. 2011;74(8):1315-1319.
92. Sun H, Wan Y, Du P, Bai L. The Epidemiology of Monophasic Salmonella Typhimurium. Foodborne Pathogens and Disease. 2020;17(2):87-97.
93. Crump J, Sjölund-Karlsson M, Gordon M, Parry C. Epidemiology, Clinical Presentation, Laboratory Diagnosis, Antimicrobial Resistance, and Antimicrobial Management of Invasive Salmonella Infections. Clinical Microbiology Reviews. 2015;28(4):901-937.
94. Ferrari R, Rosario D, Cunha-Neto A, Mano S, Figueiredo E, Conte-Junior C. Worldwide Epidemiology of Salmonella Serovars in Animal-Based Foods: a Meta-analysis. Applied and Environmental Microbiology. 2019;85(14).
95. Branchu P, Charity O, Bawn M, Thilliez G, Dallman T, Petrovska L et al. SGI-4 in Monophasic Salmonella Typhimurium ST34 Is a Novel ICE That Enhances Resistance to Copper. Frontiers in Microbiology. 2019;10.
96. Arai N, Sekizuka T, Tamamura Y, Kusumoto M, Hinenoya A, Yamasaki S et al. Salmonella Genomic Island 3 Is an Integrative and Conjugative Element and Contributes to Copper and Arsenic Tolerance of Salmonella enterica. Antimicrobial Agents and Chemotherapy. 2019;63(9).
97. Knopp M, Andersson DI. Predictable phenotypes of antibiotic resistance mutations. MBio. 2018 Jul 5;9(3).
98. McArthur AG, Waglechner N, Nizam F, Yan A, Azad MA, Baylay AJ, Bhullar K, Canova MJ, De Pascale G, Ejim L, Kalan L. The comprehensive antibiotic resistance database. Antimicrobial agents and chemotherapy. 2013 Jul 1;57(7):3348-57.

99. Achtman M, Zhou Z. Distinct genealogies for plasmids and chromosome. PLoS Genet. 2014 Dec 18;10(12):e1004874.

100. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM. Mash: fast genome and metagenome distance estimation using MinHash. Genome biology. 2016 Dec 1;17(1):132.

101. Wheeler N, Gardner P, Barquist L. Machine learning identifies signatures of host adaptation in the bacterial pathogen Salmonella enterica. PLOS Genetics. 2018;14(5):e1007333.

102. Schrider D, Kern A. Supervised Machine Learning for Population Genetics: A New Paradigm. Trends in Genetics. 2018;34(4):301-312.

103. Lupolova N, Lycett S, Gally D. A guide to machine learning for bacterial host attribution using genome sequence data. Microbial Genomics. 2019;5(12).

104. Azarian T, Huang IT, Hanage WP. Structure and Dynamics of Bacterial Populations: Pangenome Ecology. InThe Pangenome 2020 (pp. 115-128). Springer, Cham.

Table 1: Comparison of ProkEvo's performance on Crane and OSG with two datasets with significant difference in size and number of genomes.

	Crane	**OSG**	**Crane**	**OSG**
Number of genomes	2,392		23,045	
Total distributed running time*	3 days 15 hours	7 days 4 hours	15 days 22 hours	26 days 6 hours
Total estimated sequential running time**	115 days 18 hours	1 year 69 days	2 years 268 days	13 years 5 days
Maximum jobs ran in a day***	2,377	8,608	12,382	25,540
Total number of jobs ran	9,281	16,624	217,942	232,422
Output data size	131 GB		1.2 TB	

* Total distributed running time is calculated when many independent tasks are executed simultaneously while utilizing a single core each of them. This is the default behavior of ProkEvo.

** Total estimated sequential running time is calculated when all steps from the pipeline are assumed to be run sequentially, on a single core.

*** The number of maximum jobs ran in a day depends on the type and length of the job, and is not linear, i.e. some tasks run faster than others which is directly dependent of the type of job being done.

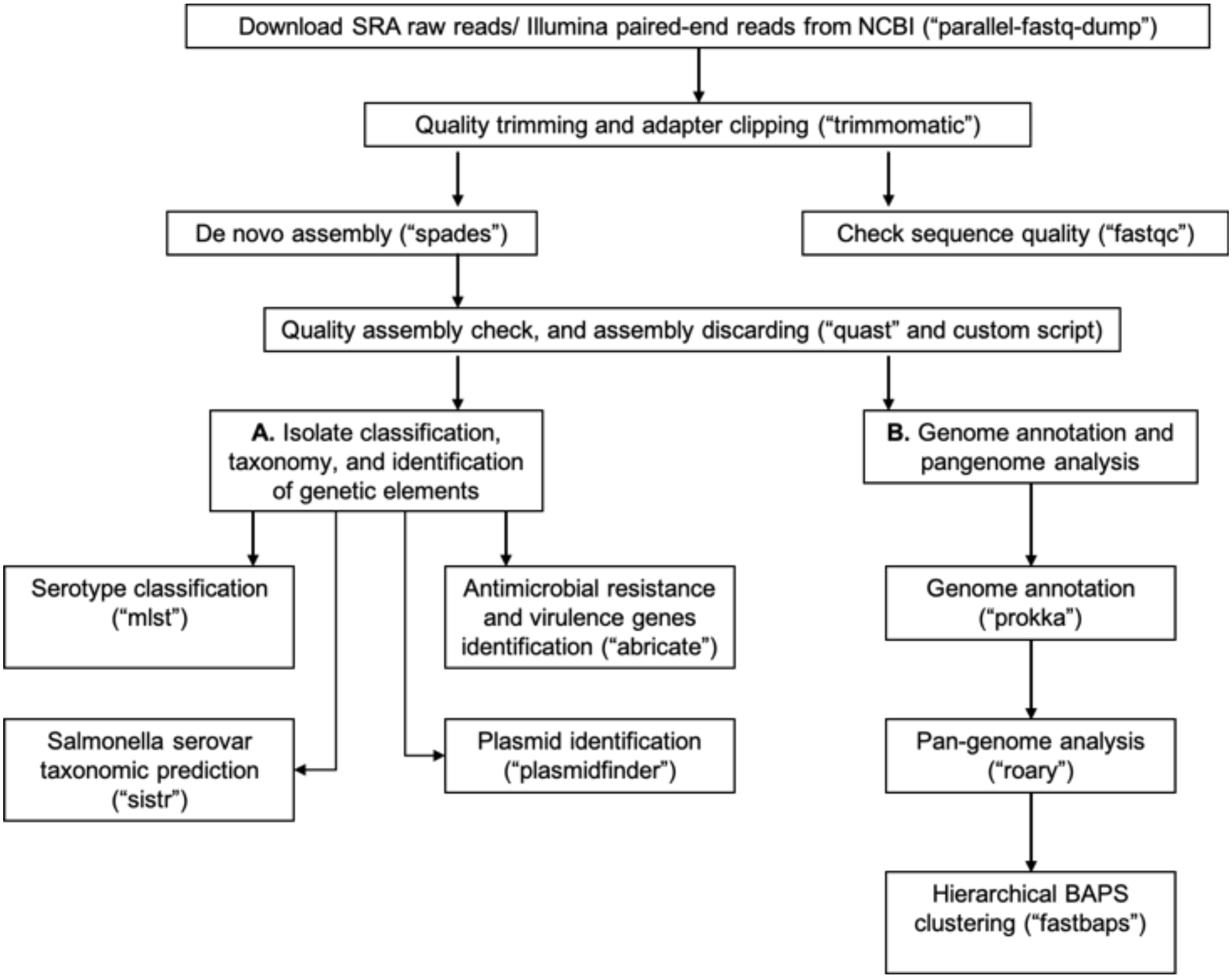
Download SRA raw reads/ Illumina paired-end reads from NCBI ("parallel-fastq-dump")
Quality trimming and adapter clipping ("trimmomatic")
De novo assembly ("spades")
Check sequence quality ("fastqc")
Quality assembly check, and assembly discarding ("quast" and custom script)
A. Isolate classification, taxonomy, and identification of genetic elements
B. Genome annotation and pangenome analysis
Serotype classification ("mlst")
Antimicrobial resistance and virulence genes identification ("abricate")
Salmonella serovar taxonomic prediction ("sistr")
Plasmid identification ("plasmidfinder")
Genome annotation ("prokka")
Pan-genome analysis ("roary")
Hierarchical BAPS clustering ("fastbaps")

Figure 1: Overall ProkEvo's computational workflow.

Top-down flow of tasks for the ProkEvo pipeline. The squares represent the steps, where the bioinformatics tool used for each step is shown in brackets. The pipeline starts with downloading raw Illumina sequences from NCBI, after providing a list of SRA identifications, and subsequently performing quality control. Next, *de novo* assembly is performed on each genome using SPAdes and the low-quality contigs are removed. This concludes the first part of the pipeline, the first sub-workflow. The second sub-workflow is composed of more specific population-genomics analyses, such as genome annotation and pangenome analyses (with Prokka and Roary) and isolate serotype predictions from genotypes in the case of Salmonella (SISTR), genotyping using core-genome (fastbaps, MLST, and cgMLST genotyping with SISTR), and identifications of genetic elements with ABRicate and Plasmidfinder.

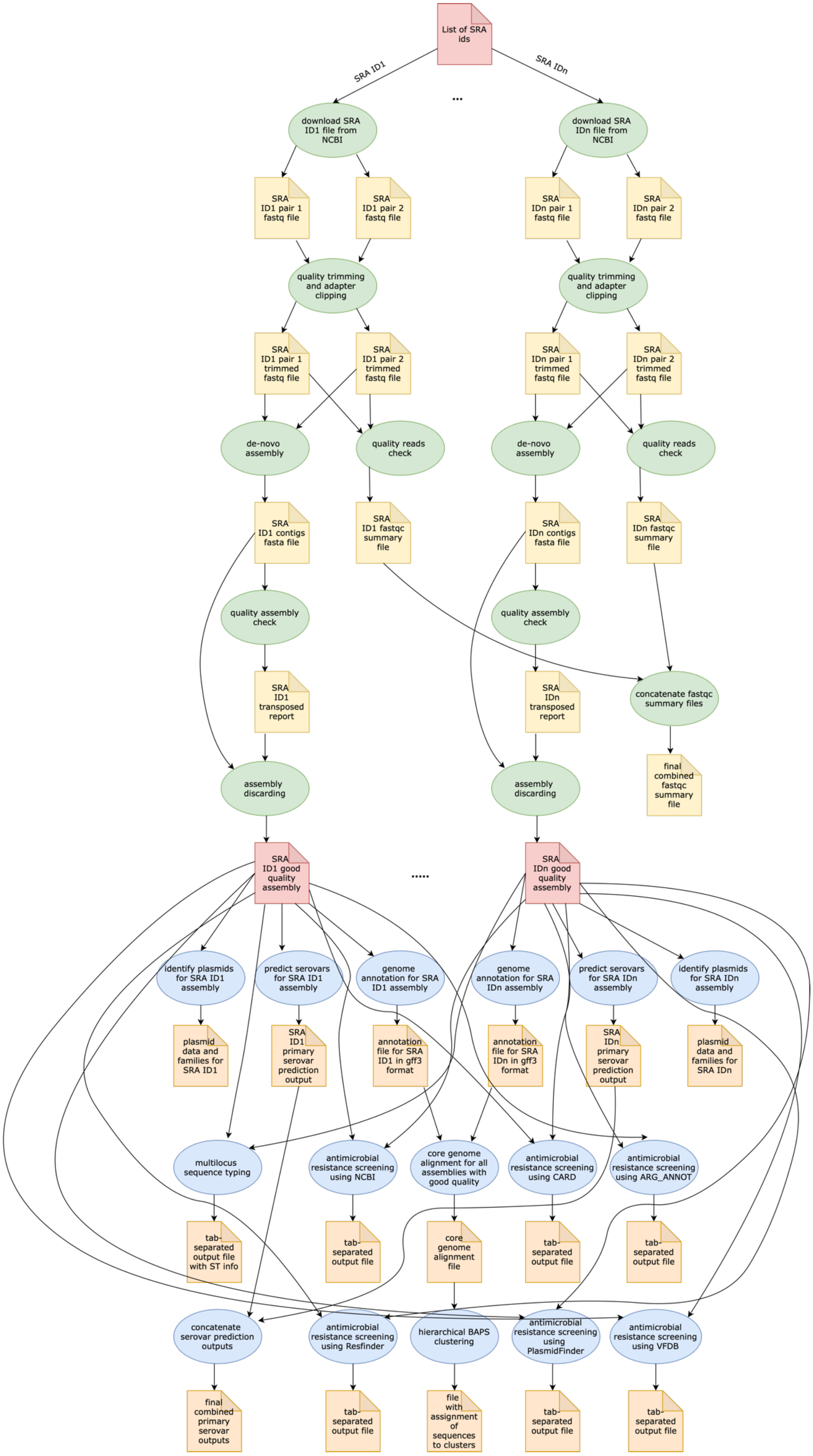
List of SRA ids
SRA ID1
SRA IDn
...
download SRA ID1 file from NCBI
download SRA IDn file from NCBI
SRA ID1 pair 1 fastq file
SRA ID1 pair 2 fastq file
SRA IDn pair 1 fastq file
SRA IDn pair 2 fastq file
quality trimming and adapter clipping
quality trimming and adapter clipping
SRA ID1 pair 1 trimmed fastq file
SRA ID1 pair 2 trimmed fastq file
SRA IDn pair 1 trimmed fastq file
SRA IDn pair 2 trimmed fastq file
de-novo assembly
quality reads check
de-novo assembly
quality reads check
SRA ID1 contigs fasta file
SRA ID1 fastqc summary file
SRA IDn contigs fasta file
SRA IDn fastqc summary file
quality assembly check
quality assembly check
SRA ID1 transposed report
SRA IDn transposed report
concatenate fastqc summary files
assembly discarding
assembly discarding
final combined fastqc summary file
SRA ID1 good quality assembly
.....
SRA IDn good quality assembly
identify plasmids for SRA ID1 assembly
predict serovars for SRA ID1 assembly
genome annotation for SRA ID1 assembly
genome annotation for SRA IDn assembly
predict serovars for SRA IDn assembly
identify plasmids for SRA IDn assembly
plasmid data and families for SRA ID1
SRA ID1 primary serovar prediction output
annotation file for SRA ID1 in gff3 format
annotation file for SRA IDn in gff3 format
SRA IDn primary serovar prediction output
plasmid data and families for SRA IDn
multilocus sequence typing
antimicrobial resistance screening using NCBI
core genome alignment for all assemblies with good quality
antimicrobial resistance screening using CARD
antimicrobial resistance screening using ARG_ANNOT
tab-separated output file with ST info
tab-separated output file
core genome alignment file
tab-separated output file
tab-separated output file
concatenate serovar prediction outputs
antimicrobial resistance screening using Resfinder
hierarchical BAPS clustering
antimicrobial resistance screening using PlasmidFinder
antimicrobial resistance screening using VFDB
final combined primary serovar outputs
tab-separated output file
file with assignment of sequences to clusters
tab-separated output file
tab-separated output file

Figure 2: Pegasus workflow of ProkEvo.
Pentagons represent the input and output files, the ovals represent the tasks (jobs), and the arrows represent the dependency order among the tasks. Pentagons are colored in red for the input files used for the first and second sub-workflow, respectively. The yellow pentagons and the green ovals represent the input and output files, and tasks (jobs) that are part of the first sub-workflow. The pentagons colored in orange and the ovals colored in blue are the input and output files, and tasks used in the second sub-workflow. While the first sub-workflow is more modular, most of the tasks from the second sub-workflow are performed on all processed genomes together. Here, the steps of the analyses for two genomes are shown, and those steps and tasks remain the same regardless of the number of genomes. The number of tasks significantly increases with the number of genomes used, and because of the modularity of ProkEvo, each task is run on a single core which facilitates parallelization at large scale. Theoretically, if there are *n* cores available on the computational platform, ProkEvo can utilize all of them and run *n* independent tasks, simultaneously (1:1 correspondence).

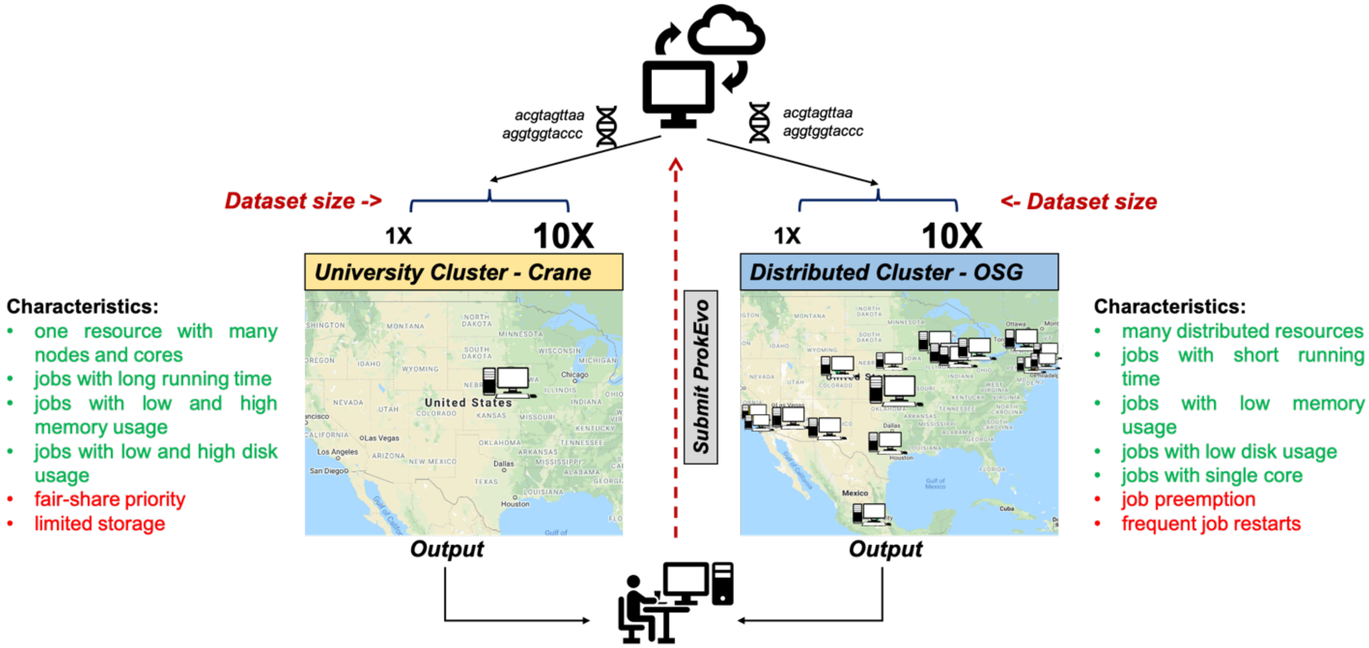
acgtagttaa
aggtggtaccc
acgtagttaa
aggtggtaccc
Dataset size ->
<- Dataset size
1X
10X
1X
10X
University Cluster - Crane
Distributed Cluster - OSG
Submit ProkEvo
Characteristics:
one resource with many nodes and cores
jobs with long running time
jobs with low and high memory usage
jobs with low and high disk usage
fair-share priority
limited storage
Characteristics:
many distributed resources
jobs with short running time
jobs with low memory usage
jobs with low disk usage
jobs with single core
job preemption
frequent job restarts
Output
Output

Figure 3: Computational experimental approach to test the performance of ProkEvo using two different computational platforms with datasets of different size.

To test how ProkEvo would perform with a small (1X) vs. moderately large (10X) datasets, in addition to using different computational resources, we have designed the following experiment: 1) Selected two adequately sized datasets including genomes from *S*. Newport (1X – from USA) and *S*. Typhimurium (10X – worldwide); 2) Used two different types of computational platforms: Crane, the University of Nebraska high-performance computing cluster, and the Open Science Grid, as a distributed high-throughput computing cluster; 3) We then ran both datasets on the two platforms with ProkEvo, and collected the statistics for the performance in order to provide a comparison between the two different computational platforms, as well as possible guidance for future runs. Of note, the text in green and red correspond to advantages and disadvantages of using each computational platform, respectively.

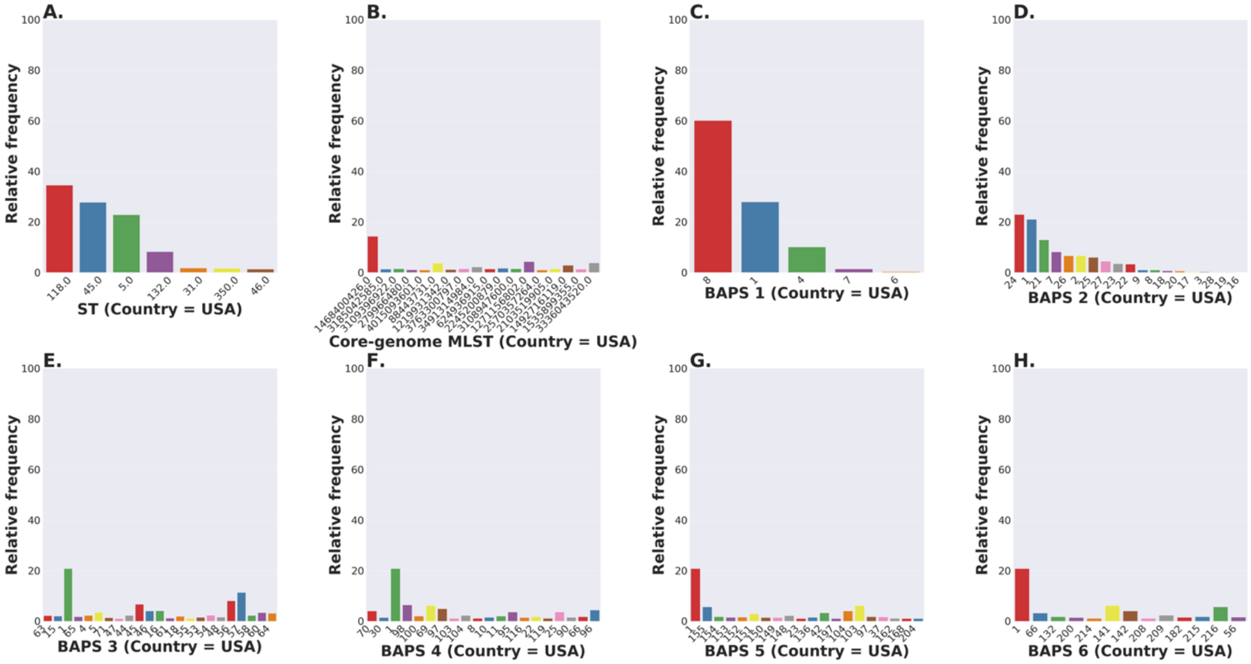

Figure 4: *Salmonella* Newport (USA) population stratification by genotype classification using two methods: allelic calls (ST and cgMLST) and a heuristic Bayesian approach (BAPS).

(A) ST distribution based on seven ubiquitous and genome-scattered loci using the MLST program, which is based on the PubMLST typing schemes (plot excludes STs with relative frequency below 1%). (B) Core-genome MLST distribution based on SISTR which uses ~330 ubiquitous loci (plot excludes STs with relative frequency below 1%). (C-H) BAPS levels 1-6 relative frequencies. For BAPS levels 3-6, we have excluded sub-groups that were below 1% in relative frequency in order to facilitate visualization. The initial number of genomes used as an input was 2,392, while these analyses were run with 2,365 genomes that passed the post-assembly filtering steps.

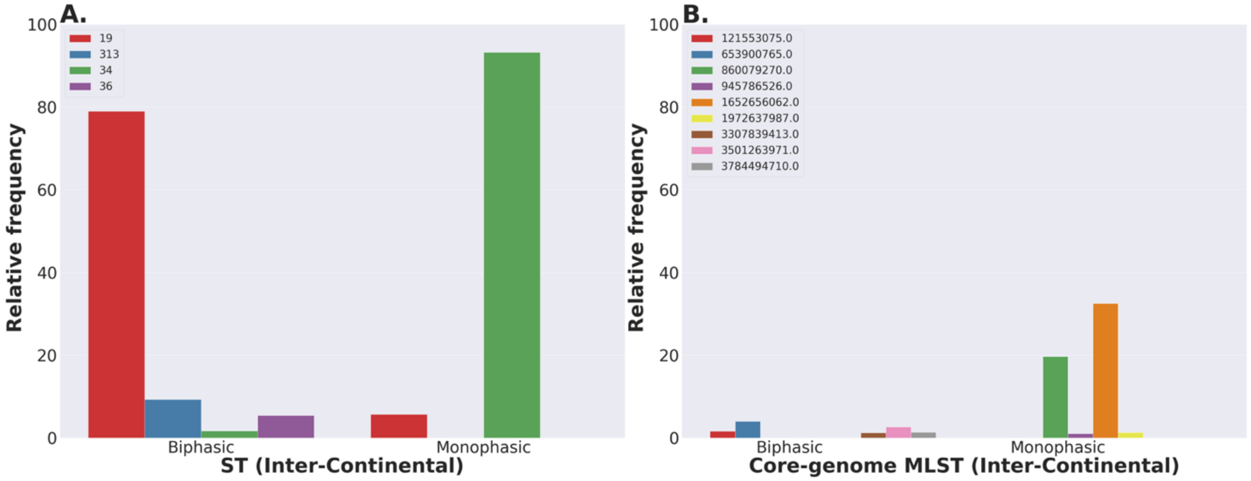

Figure 5: Inter-continental distribution of *Salmonella* Typhimurium STs and core-genome MLSTs.

(A-B) Relative frequencies of STs and core-genome MLSTs between Monophasic and Biphasic populations across multiple continents (STs and core-genome MLSTs with proportion below 1% were excluded from the graph). The initial number of genomes used as an input was 23,045, while these analyses were run with 21,534 genomes that passed the filtering steps. Raw sequences were downloaded from NCBI SRA without filtering for USA isolates exclusively. Hence, the name “Inter-Continental”. However, we cannot break the data down into continents, because the metadata was unreliable.

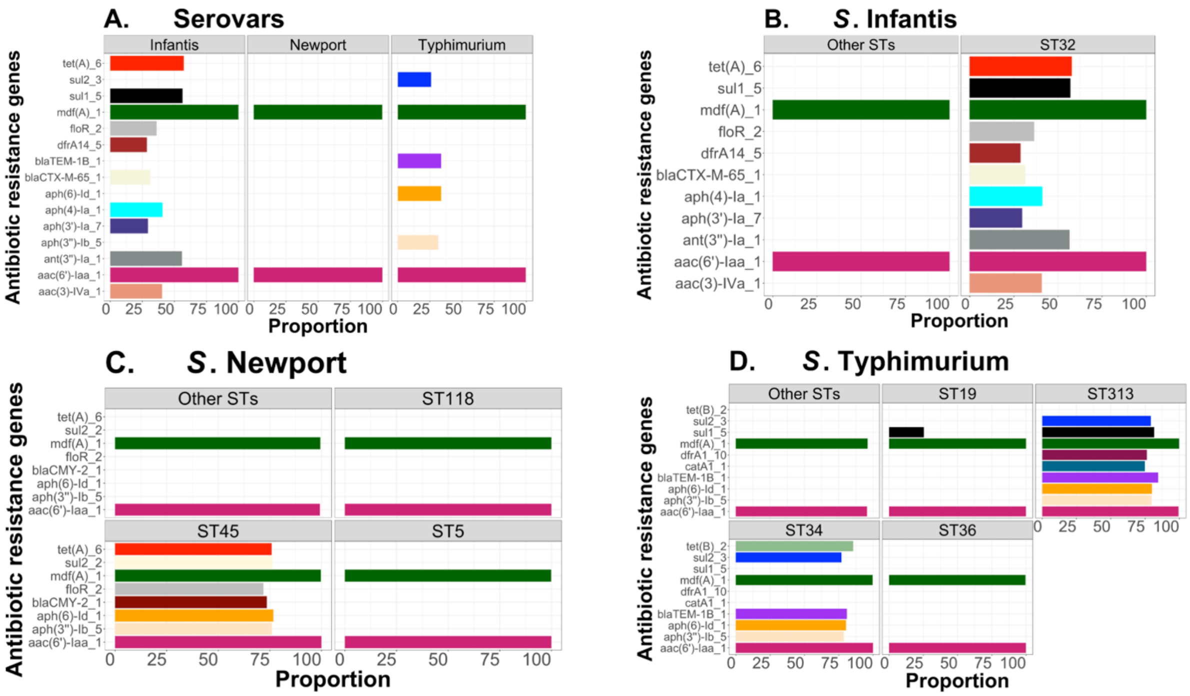

Figure 6: Antibiotic-associated resistance genes distribution between and within three serovars of *S. enterica* lineage I.

(A) Proportion of genomes containing antibiotic-associated resistance genes within each serovar. (B-D) Proportion of antibiotic-associated resistance genes within major vs. other STs for *S.* Infantis, *S.* Newport, and *S.* Typhimurium, respectively. For the plots, (B-D), the population was initially aggregated based on the dominant STs vs. the others, prior to calculating the relative frequency of genomes containing each antibiotic-resistance gene. Only proportions equal to or greater than 25% are shown. For *S.* Infantis and *S.* Newport, only USA data were used; whereas, for *S.* Typhimurium we did not filter based on geography in order to have a larger dataset used to test ProkEvo's computational performance. Datasets were not filtered for any other epidemiological factor. The total number of genomes used for this analysis was 1,684, 2,365, 21,509 for *S.* Infantis, *S.* Newport, and *S.* Typhimurium, respectively, after filtering out all missing or erroneous values. Also, there were 18 and 1666 genomes for "Other STs" and ST32 within the *S.* Infantis data, respectively. For *S.* Newport, there were 393, 800, 643, and 529 genomes of the following groups: Other STs, ST118, ST45, and ST5, respectively. Lastly, for *S.* Typhimurium, there were 1,430, 12,477, 1,493, 5,274, and 835 genomes for either Other STs, ST19, ST313, ST34, or ST36, respectively.

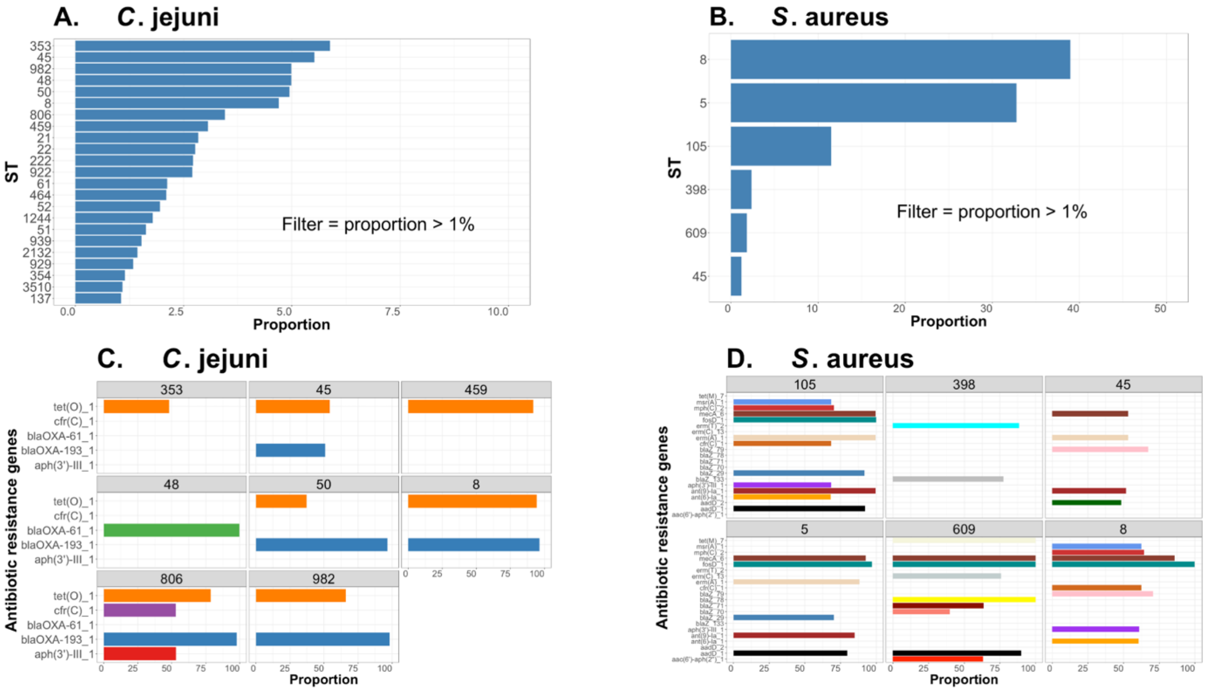

Figure 7: ST-based population structure and distribution of antibiotic-associated resistance genes for two major foodborne pathogens.
(A-B) Proportion of the most dominant STs within *C. jejuni* and *S. aureus* populations (only proportions > 1% are shown). (C-D) Proportion of genomes containing antibiotic-resistance genes within ST populations for *C. jejuni* and *S. aureus* (only proportions > 25% are shown). Both datasets only included genomes from USA and were not filtered for any other epidemiological factor. The total number of genomes entered in this analysis was 18,845 and 11,597, for *C. jejuni* and *S. aureus*, respectively, after filtering out all missing or erroneous values. For *C. jejuni*, there were 886, 1,041, 940, 932, 1,108, 577, 651, and 940 genomes of the following groups: ST8, ST45, ST48, ST50, ST353, ST459, ST806, and ST982, respectively. Lastly, for *S. aureus*, there were 4,518, 3,801, 1,334, 276, 211, and 141 genomes for either ST8, ST5, ST105, ST398, ST609, or ST45, respectively.

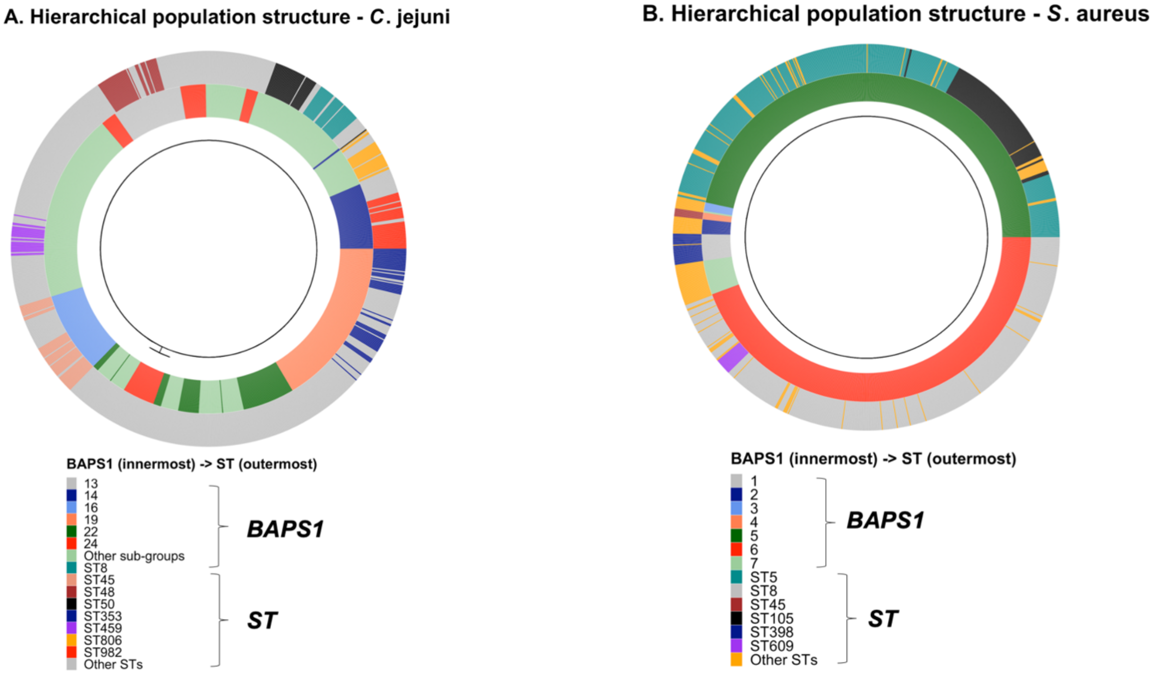

Figure 8: Relationship between the core-genome phylogeny and population structure of *C. jejuni* and *S. aureus*.

(A-B) Population structure using BAPS1 and ST for genotypic classifications were overlaid onto the core-genome phylogeny of both *C. jejuni* and *S. aureus*, respectively. BAPS1 was used as the first layer of classification to demonstrate how each sub-group can be comprised of multiple STs. For instance, STs that cluster together, and belong to the same BAPS1 sub-group, are more likely to have shared a most recent common ancestor. This represents a hierarchical population-based analysis going from BAPS1 to STs. For this analysis and visualization, we have used a random sample composed of 1,044 and 1,193 genomes for *C. jejuni* and *S. aureus*, respectively.