**BactPrep: A user-friendly whole-genome sequencing analysis platform for the detection of homologous recombination and horizontal gene transfer in bacteria**

Ruijie Xu1,2, Rajeev Sreekumari3, Liliana C. M. Salvador\*1,2,4

\*Corresponding author

1Institute of Bioinformatics, University of Georgia, Athens, GA, 30602, USA

2Center for the Ecology of Infectious Diseases, University of Georgia, Athens, GA, 30602, USA

3Department of Biomedical and Diagnostic Sciences, College of Veterinary Medicine, University of

Tennessee, Knoxville, TN, 37996, USA

4Department of Infectious Diseases, College of Veterinary Medicine, University of Georgia, Athens, GA, 30602, USA

### Abstract

### ****Background****

The acquisition and loss of DNA from bacterial genomes are two of the most effective adaptation mechanisms facilitating bacterial growth and rapid adaptation in changing habitats. In addition, these mechanisms may contribute to loss or gain of virulence and antimicrobial resistance (AMR) development. Thus, identifying these versatile regions in bacterial genomes can shed light on clonal and non-clonal bacterial properties. In recent years, many bioinformatics tools have been developed in response to the rising need in studying the composition of bacterial genomes. However, these tools are diverse in their input file formats, workflows, and implementations, and creating a project-specific workflow is time-consuming. In addition, the changes in genomes because of horizontal gene transfer (HGT) and homologous recombination are known to bias the accuracy of bacteria phylogenetic reconstructions. To respond to the need for a bacterial genomic annotation and analysis platform, we developed BactPrep, a bioinformatics software that 1) extracts the conserved genomic regions for bacterial clonal evolutionary studies, and 2) identifies genetic exchange events to aid for the understanding of bacterial adaptation mechanisms.

**Results**

BactPrep was developed using the Python-based framework Snakemake. The complete workflow includes three separate analyses: 1) identification of recombination events from bacterial genomes, 2) reconstruction of the pan and core genome of a given dataset, and 3) estimation of homologous recombination regions for each core gene. For the users' convenience, BactPrep was implemented in a modular fashion, allowing the user to decide which modules to use. In addition, a command-line interface was designed for BactPrep to replace Snakemake's default interface, which helps BactPrep implemented workflow to run with a single line of command. Finally, BactPrep's complete workflow and correspondent results were demonstrated using a Streptococcus pneumoniae PMEN1 dataset.

**Conclusions**

BactPrep has provided an easy and fast implementation for identifying and extracting recombination and HGT regions from bacterial genomes. The implemented workflow covers a comprehensive range of analyses to understand bacterial genomic plasticity, thus satisfying the need for a broad range of bacterial studies.

**Keywords**

bacteria, whole-genome sequencing, snakemake, software platform, recombination, horizontal gene transfer, clonal genome, evolution, adaptation

### Background

The increase in the number and availability of bacterial whole genome sequences (WGS) opens up an unprecedented opportunity to study the evolution and adaptation of bacterial pathogens [1–4]. Before the genomic era, the comprehension of bacterial genomic compositions was mainly acquired by analyzing one or a few housekeeping genes. These genetic elements were composed of less than 1% of the bacterial genome [1]. This type of analysis often leads to an incomplete understanding of bacterial evolutionary history [5, 6]. The availability of bacterial WGS allows current researchers to investigate the complete set of bacterial genetic compositions and provide more informative and plausible statements for bacterial genotypes, their phylogenetic relationships, and evolutionary histories. However, the biases introduced by homologous recombination and horizontal gene transfer (HGT) are still the major obstacles hindering the further advancement of our understanding of bacterial composition and evolution [2, 5, 7–9]. Homologous recombination (known as just recombination) is the genetic exchange mechanism that takes place with a closely related donor [10–13], and HGT (also known as non-homologous recombination) transfers or exchanges genetic segments with a genetically distinct donor [12, 14]. Incongruent phylogenetic topologies and distorted tree branch lengths are examples of bias sources resulting in inaccurate estimations of bacterial rate of evolution, genetic distances, and evolutionary relationships [15]. In addition to introducing biased inferences at the genomic scale, these mobile genetic regions can also skew our understanding of bacterial population structure, emergence, and transmission dynamics. The information revealed from analyzing bacterial recombination and HGT impacted regions in the non-conserved parts of the genome may bridge the gaps in our understanding of their rapid adaptation mechanisms [16]. Bacterial genomes can be separated into core (conserved) and accessory (non-conserved) genes, reconstructing the pangenome (composed of core and accessory genes) of a bacterial species (Figure 1A) [2]. One approach to resolve the biases introduced by recombination and HGT due to gene gain and loss is to make inferences based only on the core genes of a population of study [14, 17]. Furthermore, by using a pangenome approach, pathogenicity, host adaptation, and the emergence of antimicrobial resistance can be explored from the accessory genes specific to a subgroup of the population sharing the same phenotypic traits or demographical backgrounds. Example studies that adopted this approach include the study of the emergence of multiple-drug resistant *Serratia marcescens* in the United Kingdom and Ireland [18], the *Klebsiella pneumoniae* epidemics in Europe [19], and the 7th cholera pandemic in Africa [20]. Meanwhile, another widely used approach has emerged to study the non-clonal regions in the bacterial genomes is to identify the recombined regions in a WGS alignment (Figure 1D). By identifying and filtering out the potential recombined regions, this approach produces an alignment with only conserved genetic regions, which can be used for downstream clonal genomic analyses. In addition, the positions of the most frequent recombined regions, lineage-shared or phenotypic trait-associated recombined regions are valuable entry points for deciphering non-clonal adaptation mechanisms in bacteria. This approach has also been used in many previous studies, including the emergence of multiple-drug resistant *Mycobacterium abscessus* in humans [21], global surveillance of *Listeria monocytogenes* [22], and the emergence of multiple-drug resistant *Salmonella enterica [23]*. While both of these approaches are widely used to study the plasticity of the bacterial genomes and their diverse evolutionary mechanisms, the occurrences of HGT and recombination are not mutually exclusive to each other and are managed by separate selective pressures [13, 24]. Core genes shared by all the isolates in a bacterial population could also have gone through recombination with the same genes of another isolate from the same population to expedite their process to fitness in a new environment [13], which could also highly impact the results of core genome phylogenetic inference [25]. Thus, detecting both HGT and recombination events from bacterial genomes are necessary analysis steps for obtaining a comprehensive understanding of the underlying mechanism and histories shaping bacterial genomes.

With the increase in bacterial WGS availability, many software programs were developed in response to the rising need to reconstruct pangenomes [26–29], and identify recombination regions in bacterial genomes [12, 30–32]. However, these programs followed different algorithms and were implemented under different computational structures. They also require multiple package dependencies and expect non-standard input files, making the design and the development of a single bacterial genome analysis workflow challenging. To resolve these technical difficulties, we developed BactPrep, a convenient and extensive software platform combining the workflows of identifying core/accessory genome and recombination events from bacterial genomes. Furthermore, to resolve the impact of recombination events within the conserved core genome [25], we extended BactPrep to identify the recombination events from each core gene alignment individually. With the provision of a WGS assemblies dataset, BactPrep will 1) identify the recombination regions from the bacterial WGS alignment and call single nucleotide polymorphisms (SNPs) from recombination-free regions (module "wgsRecomb", Figure 1B); 2) identify core/accessory genes from the WGS input, reconstruct the pangenome (Figure 1A), and concatenate core gene alignments into a single genetically conserved core genome alignment (module "coreGen", Figure 1C) [13]; 3) detect and mask recombination regions from the alignment of the core genes identified in step 2, create a new core genome alignment, and call SNPs from it (module "coreRecomb", Figure 1D). At the end of each workflow, BactPrep will generate phylogenetic trees based on the resulting alignments. The goal of BactPrep is to provide an easily accessible and comprehensive platform that processes bacterial genomic data supplied by the users and performs both gene- and nucleotide-level types of analyses from pangenomic data.

### Implementation

The BactPrep pipeline is composed of three main independent modules, "wgsRecomb", “coreGen”, and “coreRecomb”, and two accessory modules “panRecomb”, and “geneRecomb”. BactPrep pipeline can be run with the ensemble module “ALL”, which calls the core modules “wgsRecomb”, “coreGen”, and “coreRecomb” (Figure 1), or can be run separately by calling each module independently depending on the needs of the user. Figure 2 shows the detailed workflow implemented under each module.

Module “ALL”

This module is the major module for the implementation of the pipeline (Figure 2A). Specifying this module will trigger the analyses under the “wgsRecomb”, “coreGen”, and “coreRecomb” modules. Detailed explanation for each module can be found in their independent sections below.

Module “wgsRecomb”

The “wgsRecomb” module implements the workflow for recombination detection using Gubbins, a software that identifies recombination regions by iteratively identifying genome regions with high densities of mutations (Figure 2B). This module requires the bacterial WGS assemblies as input, and a reference genome for the downstream analysis (-r flag) (Table 1). Each assembly is first aligned with the reference genome to find the single nucleotide polymorphisms (SNPs) using Snippy [33]. After SNP calling, Snippy will combine each alignment with the reference genome to produce a multiple sequence alignment, containing the genomic regions shared among all input assemblies. This multi-sequence alignment will be used as the input for Gubbins [30]. Gubbins will produce a recombination filtered multi-sequence SNPs alignment as the final output [34]. A Maximum Likelihood (ML) phylogeny will be reconstructed for the input dataset with the recombination-free SNP alignment. The SNP alignment and the ML phylogeny will be annotated if the user provides a metadata file.

Module “coreGen”

The “coreGen” module implements the workflow to perform a pangenome construction analysis using the tool Roary [26] with the bacterial WGS assemblies as the input (Figure 2C, Table 1). This module starts with obtaining uniform annotations for all input assemblies using the tool PROKKA [35], which is designed for rapid bacterial, archaeal, and viral genome annotation. The genomic annotations of PROKKA will be used as the input for the Roary analysis, which constructs the pangenome of the input WGS dataset and identifies the core genome of the dataset. By default, these are the genes shared by more than 99% of the isolates within the dataset; however, the parameter can be adjusted using the “-c” flag to change the definition of the core genome. The core genes identified by Roary are individually aligned using MAFFT [36], concatenated into a super-alignment, and provided as an output. The ML phylogeny of the input WGS is inferred from the concatenated core genome alignment using IQ-Tree [37] and also provided as an output. If the user provides a metadata file, a metadata annotated core genome alignment and phylogeny will also be given as outputs.

Module “coreRecomb”

The module “coreRecomb” processes the output of the “coreGen” module (Figure 2D). With the reconstruction of the pangenome dataset, Roary produced individual gene alignments for all genes (clusters of orthologous genes - COG) identified from the WGS assemblies of the input dataset. The software fastGear detects recombination regions for a single gene [32]. Therefore, “coreRecomb” module takes the advantage of Roary [26] output and the tool fastGear to identify recombination from the produced core gene alignments. The algorithm of fastGear will first separate taxa in the input alignment into different lineages based on their sequences similarities using the BAPS algorithm [38]. Ancestral recombination is then inferred as the transfer of genetic segments from all of genomes of one lineage to all of the genomes in the other lineage using a hidden Markov Model (HMM) [39], which produces the log likelihood of the recombination events. The algorithm of fastGear will also report the potential donor lineage and the lineage of the recipient genomes based on the sequences of other lineages at the same position within the dataset using HMM. Recombination regions identified within each gene alignment will be masked before being concatenated into a single super-gene alignment using catfasta2phyml [40]. SNPs from the recombination-free regions (transformed in a super-alignment) will be called using SNP-sites [34]. A ML phylogeny will be reconstructed at the end of the workflow with the taxa annotated if the user provides a metadata file.

Modules “panRecomb” and “geneRecomb”

The modules “panRecomb” and “geneRecomb” are two accessory modules and are not part of the module “ALL”. The module “panRecomb” is designed for users with a need to evaluate recombination from all genes in the pangenome (Figure 2E). It also takes advantage of Roary’s individual gene alignment outputs (just like the module “coreRecomb”) and uses it as the input for fastGear’s gene-wise recombination detection analysis. However, if the user would only like to assess the presence of recombination in a single or a few genes, the “geneRecomb” module allows the input of a single gene alignment or a list of gene alignments for recombination detection (Figure 2F, Table 1).

**Table 1.** BactPrep module composition: description of the software used and input files.

|  |  |  |  |
| --- | --- | --- | --- |
| **Modules** | **Software** | **Required Input Files** | **Optional Input Files** |
| wgsRecomb | Snippy [33], Gubbins [30], IQ-Tree [37] | Reference genome, WGS assemblies | Phage region (BED file), metadata CSV |
| coreGen | PROKKA [35], Roary [26], IQ-Tree [37] | WGS assemblies | Metadata CSV |
| coreRecomb | PROKKA [35], Roary [26], snp-sites [34], Bedtools [41], fastGear [32], IQ-Tree [37] | WGS assemblies | Metadata CSV, Path/to/MCR & Path/to/fastGear executable (if installed before) |
| panRecomb | PROKKA [35], Roary [26], snp-sites [34], Bedtools [41], fastGear [32] | WGS assemblies |
| geneRecomb | PROKKA [35], Roary [26], snp-sites [34], Bedtools [41], fastGear [32] | Gene alignment/List of gene alignments | Path/to/MCR & Path/to/fastGear executable (if installed before) |

### Results

To demonstrate the use of BactPrep, a *Streptococcus pneumoniae* PMEN1 WGS dataset (PMEN1) with 218 isolates was obtained from a previous study [42]. The NCBI accession number for each isolate and the corresponding metadata are provided in the supplementary Table S1. Downloaded WGS reads were first quality checked using fastQC 0.11.9 [43], adapters and low quality reads were removed using Trimmomatic 0.39 [44] with parameter “ILLUMINACLIP:TruSeq2-PE.fa:2:30:10:2:keepBothReads LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36”. Finally, trimmed reads were error-corrected and assembled into WGS assemblies using SPAdes 3.15.0 with parameters "--careful --mismatch-correction”. The assembled WGS of the dataset is available in the link: 10.5281/zenodo.5603335. The BactPrep module “ALL”, which is the ensemble module that implements the workflows for “coreGen”, “wgsRecomb”, and “coreRecomb”, was used to produce the results shown in Figures 3-7. Examples with the implementation of the modules “panRecomb” and “geneRecomb” are also shown in Figure 8 and Table 3, respectively.

Command to run BactPrep with module ‘ALL’ using the PMEN1 WGS dataset:

start\_analysis.py ALL \

-p PMEN1.dated \

-i $WORKPATH/assemblies \

-o $WORKPATH \

-r $WORKPATH/GCF\_000026665.1\_ASM2666v1\_genomic.fna \

-M \

-a $WORKPATH/TableS1.PMEN1.dated.metadata.csv \

-m Year,Country \

-t 5 \

-G " -f 30" \

In this command, we have assigned a project prefix (“PMEN1.dated”) to all the output files using flag "-p". If the project prefix is not specified, the prefix for all output files will be “BactPrep”. The directory that contains all WGS assemblies used as input is provided using the flag “-i”. This flag is wrequired for all modules except for the “geneRecomb” module. BactPrep will extract all WGS assemblies from the specified input folder and will start the downstream analysis. The path to the reference genome is provided using the flag “-r”, which is required if “ALL” or “wgsRecomb” module is intended to be run. The output directory is set with the variable $WORKPATH using the flag “-o”, and the default output directory will be /ABSOLUTE/PATH/TO/BactPrep/results, if not specified. The flag “-M” is used to annotate the output alignments and phylogeny’s taxa. With the “-M” flag specified, a “-a” flag and a “-m” flag must also be specified to provide the path to the metadata file, and the column names of the metadata file that the user would like to use to concatenate at the end of each taxon name. In this example, “Year” and “Country” are the column names of two metadata categories that were concatenated to the end of each taxon (Figures 3 & 4). The metadata file should be a CSV file with the first row as the column name and the first column as the sample ID (sample ID was automatically extracted from the filename of the WGS assemblies provided by the user). The flag “-t” is used to specify the number of threads that the user would like to use to run the analysis. Finally, suppose additional parameters need to be added to Gubbins [30], which is implemented in the “ALL” and “wgsRecomb” modules, a “-G” flag is used to pass any additional parameters to the software. In this example, we used the “-G” flag to pass the parameter “-f 30” into the Gubbins analysis, which changed the upper limit for genomes with percentage of gaps from default 25% to 30%.

The three ML phylogenies were inferred from the output alignments of “wgsRecomb’ (which produced a recombination-free WGS SNPs alignment), “coreGen” (which produced a core gene concatenation alignment), and “coreRecomb” (which produced a recombination-masked core genome SNPs alignment) (Figure 2). All three phylogenies were middle point rooted, which assumes that all sequences have evolved at the same rate. These three phylogenies were reconstructed with 5,557, 44,825, and 42,394 sites, respectively.

Because a metadata file (CSV file) was provided during the analysis (-M & -a flag), the taxa for each ML phylogeny and their corresponding alignments were annotated. The specific columns (“Year” and “Country”) within the metadata file were concatenated to the end of each taxon (using flag “-m”) separated by the “|” delimiter (Figures 3 & 4).

The positions of PMEN1 recombination regions identified by the “wgsRecomb” module can be visualized using phandango [45], which is an interactive viewer for bacterial population genomics (no additional installation needed) (Figure 5). With the provision of the ML phylogenies and predicted recombination positions (stored at gubbins/{project\_prefix}.recombination\_predictions.gff) added to phandango, the visualizer shows the positions of the predicted recombination regions referencing to the reference genome. Red blocks represent ancestral recombination regions shared by more than one closely related isolates, while blue blocks correspond to recent recombination regions identified from a single taxon. The line plot at the bottom of the diagram in Figure 5 shows the recombination frequencies throughout the positions of the PMEN1 genomes. A total of 1,307 recombination regions were identified from the PMEN1 WGS alignment. From these, 1,016 were recent recombination regions identified from a single taxon, while 291 were ancestral recombination regions shared by at least two isolates from the same most recent common ancestor. SNPs called from the recombination regions were filtered from the final alignment, leaving 5,557 recombination filtered SNPs for the construction of the ML final tree (gubbins/{project\_prefix}.final\_tree.tre). The recombination statistics per isolate can be find in the file {project\_prefix}.per\_branch\_statistics.csv.

The output of “coreGen” module can also be visualized using phandango (just like the “wgsRecomb” module). With the provision of the ML phylogeny inferred from the core gene concatenation alignment (roary/roary\_iqtree/{project\_prefix}.treefile) and the gene presence/absence CSV file (roary/gene\_presence\_absence.csv), phandango shows genes shared by all isolates in the dataset and those uniquely present in a single isolate (Figure 6). The frequencies of each gene present in the pangenome of the dataset decreases from left to right. In the case of the PMEN1 dataset (Figure 6), 866 core genes were identified (genes shared by >= 99% of the isolates in the dataset), while 5,179 total genes were identified from the genomes of all the isolates in the dataset (pangenome). Only core genes with single copy per isolate (865 genes) were concatenated into the core genome alignment (roary/core\_gene\_alignment.aln). ML trees were reconstructed based on the core genome of the dataset (roary/roary\_iqtree/{project\_prefix}.treefile) and are shown at the left of Figure 6.

To identify recombination within the core genome of the PMEN1 dataset, the module “coreRecomb” identified recombination from core genes alignments produced by “coreGen” module individually. Detected recombination regions were masked within each gene alignment. These masked core gene alignments were concatenated together again to produce a new core genome alignment. Masked core genome shown in Figure 7 was produced at end of the “coreRecomb” module (fastgear\_core/plot\_coregenome/core\_fastgear\_plot\_heat.pdf). Out of the 865 core genes, 45 were identified with at least one recombination, 30 were identified experiencing only recent recombination events, 11 were identified to experience only ancestral recombination, and 3 were identified to experience both recent and ancestral recombination (statistics can be found in file: fastgear\_core/plot\_coregenome/core\_fastgear\_plot\_scatter\_count.csv). The statistics and positions of recent and ancestral recombination for every core gene can be found in the directory fastgear\_core/loci\_fastGear\_out/{gene\_name}/output.

To demonstrate the results produced by the “panRecomb” module, we have analyzed the PMEN1 dataset using the command below:

start\_analysis.py panRecomb \

-p PMEN1.dated\_fastGear\_pan \

-o $WORKPATH \

-t 10 \

-i $WORKPATH/assemblies

This analysis provided the recombination statistics for all the genes in the pangenome of the PMEN1 dataset, which was reconstructed through the analysis implemented in the “coreGen” module (the “coreGen” analysis will start automatically with the “panRecomb” module specified in the command line). At end of the analysis, a scatter plot showing the numbers of ancestral recombination vs. the numbers of recent recombination events for each gene in the pangenome will be produced (Figure 8). Furthermore, the COG “group\_191” was identified as the gene with the highest frequency of recent recombination, and the COG “group\_1586” and “group\_995” were the genes with the highest frequency of ancestral recombination (Figure 8). The statistics for each gene in the PMEN1 pangenome are available in the fastgear/loci\_fastGear\_out/{gene\_name}/output directory.

To demonstrate the use of the module “geneRecomb”, we used the alignment of the gene *dexB* as an example. The gene was found present in 212 out of 218 WGS assemblies in the analyses implemented in the “coreGen” module. Sequences of the *dexB* gene present in the PMEN1 WGS dataset were also extracted and aligned through the “coreGen” analyses. In the command below, “geneRecomb” module was used to identify the presence of recombination regions in the *dexB* gene alignment:

start\_analysis.py geneRecomb \

-n $WORKPATH/roary/pan\_genome\_sequences/dexB.fa.aln \

-o $WORKPATH

All the WGS assemblies possessing the gene *dexB* were clustered into 3 different lineages (fastgear\_gene/dexB/output/lineage\_information.txt). One ancestral recombination and 5 recent recombinations (fastgear\_gene/dexB/output/recombinations\_recent.txt) were identified from the *dexB* alignment. Table 3 shows the ancestral recombination identified from the *dexB* alignment through the “geneRecomb” analysis (fastgear\_gene/dexB/output/recombinations\_ancestral.txt). The statistics about recent recombinations identified from the *dexB* alignment can be found in the fastgear\_gene/dexB/output/recombinations\_recent.txt file.

**Table 2.** Ancestral recombination identified from the *dexB* gene alignment using the “geneRecomb” module. The “Start” and “End” columns indicate the exact position of recombination present in the gene alignment. Column “Lineage1” shows the potential donor for the recombination and column “Lineage2” shows the recipient lineage of the recombination. Column “Log(BF)” shows the likelihood of the recombination using Bayes Factor (BF) in the log scale.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ANCESTRAL RECOMBINATIONS BY LINEAGE | | | | |
| Start | End | Lineage1 | Lineage2 | Log(BF) |
| 292 | 372 | 2 | 1 | 2.4 |

### Discussion

BactPrep pipeline was developed to detect recombination and HGT regions from bacteria WGS assemblies. Recombination and HGT are the two primary mechanisms driving the non-clonal adaptation in bacteria. A workflow for the two widely used software, Gubbins and Roary, has been implemented in this software platform to identify, extract, and analyze the impact of recombination and HGT in bacteria WGS. While Gubbins identifies recombination regions based on the elevated SNP densities [30], Roary separates the bacteria genomes into core and accessory genes. Gubbins performs best when dealing with datasets from the sub-species level, while datasets with larger genetic divergence (ex. above species-level) are typically analyzed by obtaining the core genome first using Roary [26, 30]. However, bacterial core genes can also experience recombination that will disrupt bacteria clonal evolutionary dynamics. Thus, the gene-wise recombination detection tool, fastGear, was also included to detect recombination events from every individual core gene.

**Platform Interface**

The BactPrep platform was implemented under the conda environment using the Snakemake framework [47]. To provide a more accessible user interface, a command-line user interface was designed for the user to access each module and its specific flags. The command-line interface, in replacement of the Snakemake’s conventional config file specified interface, can provide a more intuitive way for the user to run this tool. The installation of BactPrep requires Conda, which will install the Snakemake framework and the other platform dependencies. To run the three modules “coreRecomb”, “panRecomb”, and “geneRecomb” that are based on fastGear, a matlab runtime complier (MCR) needs to be installed before using the tool. All installation procedures are included in the INSTALL.sh script. The user can install all required dependencies by running the INSTALL.sh script.

**BactPrep advantage**

Roary, Gubbins, and fastGear are all popular tools to study bacterial WGS. However, each tool requires different levels of learning and multiple supporting tools to prepare their specific input files. For example, the input of Roary requires the GFF3 annotation files produced by the tool Prokka, while Gubbins requires the whole genome alignment of all input bacterial assemblies, which is typically obtained through Snippy analyses for each input isolate. Finally, fastGear, a tool designed for detecting recombination regions in a gene-wise fashion, can be difficult to use in the WGS era when a large amount of gene alignments from bacterial genomes needs to be extracted and aligned before being analyzed individually. BactPrep combines pre-analysis workflow for each tool and allows the user to quickly perform a comprehensive genome analysis on their datasets with a single line of code (which greatly reduces the learning curve for learning each tool individually). Moreover, BactPrep also includes the ML reconstruction step for the results of every tool, which allows users to directly visualize the results of each analysis, saving the effort of building a ML phylogeny separately for each individual tool.

**Comparison with other tools**

Recently, multiple tools and pipelines were developed to analyze bacterial genomes. However, each of them was implemented with different goals (Table 3). For example, Bactopia was designed to specifically perform comparative genomic analysis for Illumina sequenced bacterial genomes [48]. Although implemented with a broad range of functionalities, the Bactopia pipeline could not be used to analyze bacterial genomes sequenced by data produced from the emerging long-read sequencing platforms (PacBio or Oxford Nanopore Technologies (ONT)) [48]. An alternative pipeline, ASA3P, which also starts from raw read sequences, has implementations dealing with long-read sequences [49]. However, WGS recombination detection has not been addressed in the ASA3P’s analysis, which has focused only on core/pangenome analyses. The workflow TORMES [50] has the same limitation.

In contrast to ASA3P and TORMES, the pipeline [bactmap](https://github.com/nf-core/bactmap) was designed to only deal with recombination detection without reconstructing the pangenome and extracting core and accessory gene profiles for each isolate [51]. BactPrep was designed and implemented specifically for detecting the non-conserved regions from bacterial WGS assemblies. Without the accessory functionalities implemented in other currently available pipelines, BactPrep provides a lightweight platform to identify mobile genetic events in bacterial WGS and separates the non-conserved and conserved genomic regions for downstream bacterial WGS analyses. By taking WGS assemblies as the input, BactPrep allows users to prepare input assemblies using an assembler of their choice with appropriate parameters, thus avoiding the limitation of only dealing with short-read sequences. More importantly, this implementation allows users to analyze WGS assemblies published in public databases without raw reads records. On the other hand, BactPrep can extract non-conserved genomic regions using both core genome construction and recombination detection methods, thus allowing users to choose their method of preference based on the genetic divergence within the genomes of their datasets. Finally, BactPrep is the first tool designed to deal with gene-wise recombination within the conserved core genome, which has not been implemented in any of the tools mentioned above.

**Table 3.** Tool comparison forbacterial whole-genome sequencing analysis.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Workflow | Input files | WGS recombination detection | Core/ pangenome reconstruction | Core genome recombination detection | Framework | Year | Ref |
| BactPrep | WGS assemblies | Yes  (Gubbins) | Yes  (Roary) | Yes  (fastGear-gene-wise) | Snakemake | 2021 | NA |
| Bactopia | Illumina | No | Yes  (Roary) | Yes (ClonalFrameML-core genome) | Nextflow | 2020 | [48] |
| ASA3P | Illumina/  Pacbio/  ONT | No | Yes  (Roary) | No | Docker/  OpenStack | 2020 | [49] |
| TORMES | Illumina | No | Yes  (Roary) | No | BASH/R | 2019 | [50] |
| bactmap | Illumina/  WGS assemblies | Yes  (Gubbins) | No | No | Nextflow | 2020 | [51] |

### ****Conclusions****

With the decrease in the cost of Next Generation Sequencing, many bacterial genomes have become available to be studied. However, bacterial genome processing and analysis are complex and usually involve a high level of bioinformatics expertise. The development of BactPrep provides a fast and user-friendly platform to analyze the complex evolutionary mechanisms in bacterial genomes. Furthermore, BactPrep identifies and offers annotations for conserved and non-conserved regions in bacterial genomes, which provide opportunities to study both the (conserved) evolutionary dynamics of bacterial genomes as well as the non-clonal regions, contributing significantly to future bacterial genomics studies.

### Availability and requirements **Project name:** BactPrep **Project home page:** https://github.com/rx32940/BactPrep **Operating system(s):** Linux **Programming language:** Python, Bash, Perl **Other requirements:** Python 3.6 or higher, Matlab Runtime **License:**  MIT License **Any restrictions to use by non-academics:** N/A

### List of abbreviations

NGS: Next Generation Sequencing

WGS: Whole Genome Sequences

HGT: Horizontal Gene Transfer

CSV: Comma Separated Values

HMM: Hidden Markov Model

BF: Bayes Factor

**Declarations**

**Ethics approval and consent to participate**

Not applicable

**Consent for publication**

Not applicable

**Availability of data and materials**

The source code files developed during the current study are available in the GitHub Repository: <https://github.com/rx32940/BactPrep.git>. The datasets generated and/or analyzed during the current study are available in the “BactPrep: An automatic pipeline for bacterial whole genome sequencing analysis - Sample Dataset” repository: 10.5281/zenodo.5603335.

**Competing interests**

The authors declare that they do not have competing interests.

**Funding**

This work was supported by the National Science Foundation under Grant No. DGE-1545433 to R.X. and startup funds to L.C.M.S. from the University of Georgia Office of Research.

**Authors' contributions**

R.X. performed research, analyzed data, and wrote first draft of manuscript. R.S. contributed to study interpretation and revised the manuscript. L.C.M.S. supervised research, contributed to study interpretation, and revised the manuscript. All authors read and approved the final manuscript.

**Acknowledgements**

R.X. acknowledges the support provided by the Interdisciplinary Disease Ecology Across Scales (IDEAS) doctoral program at the University of Georgia.

**Authors' information**

R.X. is a PhD candidate at the Institute of Bioinformatics and the Interdisciplinary Disease Ecology Across Scales (IDEAS) doctoral program at the University of Georgia. R.S. is a Professor of Infectious Diseases (Biomedical and Diagnostic Sciences) at the College of Veterinary Medicine at the University Tennessee. L.C.M.S. is an Assistant Professor at the Institute of Bioinformatics and the Department of Infectious Diseases (College of Veterinary Medicine) at the University of Georgia. Both R.X. and L.C.M.S. are members of the Center for the Ecology of Infectious Diseases at the University of Georgia.

## **References**

1. Castillo-Ramírez S, Corander J, Marttinen P, Aldeljawi M, Hanage WP, Westh H, et al. Phylogeographic variation in recombination rates within a global clone of methicillin-resistant Staphylococcus aureus. Genome Biol. 2012;13:R126. doi:10.1186/gb-2012-13-12-r126.

2. Ingle DJ, Howden BP, Duchene S. Development of Phylodynamic Methods for Bacterial Pathogens. Trends Microbiol. 2021. doi:10.1016/j.tim.2021.02.008.

3. Mainda G, Lupolova N, Sikakwa L, Bessell PR, Muma JB, Hoyle DV, et al. Phylogenomic approaches to determine the zoonotic potential of Shiga toxin-producing Escherichia coli (STEC) isolated from Zambian dairy cattle. 2016. doi:10.1038/srep26589.

4. Kamath PL, Foster JT, Drees KP, Luikart G, Quance C, Anderson NJ, et al. Genomics reveals historic and contemporary transmission dynamics of a bacterial disease among wildlife and livestock. Nat Commun. 2016;7:11448. doi:10.1038/ncomms11448.

5. Feil EJ, Holmes EC, Bessen DE, Chan M-S, Day NPJ, Enright MC, et al. Recombination within natural populations of pathogenic bacteria: Short-term empirical estimates and long-term phylogenetic consequences. Proc Natl Acad Sci. 2001;98:182–7. doi:10.1073/pnas.98.1.182.

6. Degnan JH, Rosenberg NA. Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends Ecol Evol. 2009;24:332–40.

7. Sakoparnig T, Field C, van Nimwegen E. Whole genome phylogenies reflect the distributions of recombination rates for many bacterial species. eLife. 2021;10:e65366. doi:10.7554/eLife.65366.

8. Pearson T, Giffard P, Beckstrom-Sternberg S, Auerbach R, Hornstra H, Tuanyok A, et al. Phylogeographic reconstruction of a bacterial species with high levels of lateral gene transfer. BMC Biol. 2009;7:78. doi:10.1186/1741-7007-7-78.

9. Vaughan TG, Welch D, Drummond AJ, Biggs PJ, George T, French NP. Inferring Ancestral Recombination Graphs from Bacterial Genomic Data. Genetics. 2017;205:857–70. doi:10.1534/genetics.116.193425.

10. Majewski J. Sexual isolation in bacteria. FEMS Microbiol Lett. 2001;199:161–9. doi:10.1111/j.1574-6968.2001.tb10668.x.

11. Fraser C, Hanage WP, Spratt BG. Recombination and the Nature of Bacterial Speciation. Science. 2007;315:476–80. doi:10.1126/science.1127573.

12. Marttinen P, Hanage WP, Croucher NJ, Connor TR, Harris SR, Bentley SD, et al. Detection of recombination events in bacterial genomes from large population samples. Nucleic Acids Res. 2012;40:e6. doi:10.1093/nar/gkr928.

13. Vos M. Why do bacteria engage in homologous recombination? Trends Microbiol. 2009;17:226–32. doi:10.1016/j.tim.2009.03.001.

14. Polz MF, Alm EJ, Hanage WP. Horizontal gene transfer and the evolution of bacterial and archaeal population structure. Trends Genet TIG. 2013;29:170–5. doi:10.1016/j.tig.2012.12.006.

15. Hedge J, Wilson DJ. Bacterial Phylogenetic Reconstruction from Whole Genomes Is Robust to Recombination but Demographic Inference Is Not. mBio. 2014;5. doi:10.1128/MBIO.02158-14.

16. Didelot X, Croucher NJ, Bentley SD, Harris SR, Wilson DJ. Bayesian inference of ancestral dates on bacterial phylogenetic trees. Nucleic Acids Res. 2018;46:e134–e134. doi:10.1093/nar/gky783.

17. Costa SS, Guimarães LC, Silva A, Soares SC, Baraúna RA. First Steps in the Analysis of Prokaryotic Pan-Genomes. Bioinforma Biol Insights. 2020;14:1177932220938064. doi:10.1177/1177932220938064.

18. Moradigaravand D, Boinett CJ, Martin V, Peacock SJ, Parkhill J. Recent independent emergence of multiple multidrug-resistant Serratia marcescens clones within the United Kingdom and Ireland. Genome Res. 2016;26:1101–9. doi:10.1101/gr.205245.116.

19. David S, Reuter S, Harris SR, Glasner C, Feltwell T, Argimon S, et al. Epidemic of carbapenem-resistant Klebsiella pneumoniae in Europe is driven by nosocomial spread. Nat Microbiol. 2019;4:1919–29. doi:10.1038/s41564-019-0492-8.

20. Weill F-X, Domman D, Njamkepo E, Tarr C, Rauzier J, Fawal N, et al. Genomic history of the seventh pandemic of cholera in Africa. Science. 2017;358:785–9. doi:10.1126/science.aad5901.

21. Bryant JM, Grogono DM, Rodriguez-Rincon D, Everall I, Brown KP, Moreno P, et al. Emergence and spread of a human-transmissible multidrug-resistant nontuberculous mycobacterium. Science. 2016;354:751–7. doi:10.1126/science.aaf8156.

22. Moura A, Criscuolo A, Pouseele H, Maury MM, Leclercq A, Tarr C, et al. Whole genome-based population biology and epidemiological surveillance of Listeria monocytogenes. Nat Microbiol. 2016;2:1–10. doi:10.1038/nmicrobiol.2016.185.

23. Klemm EJ, Shakoor S, Page AJ, Qamar FN, Judge K, Saeed DK, et al. Emergence of an Extensively Drug-Resistant *Salmonella enterica* Serovar Typhi Clone Harboring a Promiscuous Plasmid Encoding Resistance to Fluoroquinolones and Third-Generation Cephalosporins. mBio. 2018;9. doi:10.1128/mBio.00105-18.

24. Narra HP, Ochman H. Of What Use Is Sex to Bacteria? Curr Biol. 2006;16:R705–10. doi:10.1016/j.cub.2006.08.024.

25. Stott CM, Bobay L-M. Impact of homologous recombination on core genome phylogenies. 2020. doi:10.21203/rs.3.rs-29898/v1.

26. Page AJ, Cummins CA, Hunt M, Wong VK, Reuter S, Holden MTG, et al. Roary: rapid large-scale prokaryote pan genome analysis. doi:10.1093/bioinformatics/btv421.

27. Contreras-Moreira B, Vinuesa P. GET\_HOMOLOGUES, a Versatile Software Package for Scalable and Robust Microbial Pangenome Analysis. Appl Environ Microbiol. 2013;79:7696–701. doi:10.1128/AEM.02411-13.

28. Chaudhari NM, Gupta VK, Dutta C. BPGA- an ultra-fast pan-genome analysis pipeline. Sci Rep. 2016;6:24373. doi:10.1038/srep24373.

29. Ding W, Baumdicker F, Neher RA. panX: pan-genome analysis and exploration. Nucleic Acids Res. 2018;46:e5–e5. doi:10.1093/nar/gkx977.

30. Croucher NJ, Page AJ, Connor TR, Delaney AJ, Keane JA, Bentley SD, et al. Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using Gubbins. Nucleic Acids Res. 2015;43:e15–e15. doi:10.1093/nar/gku1196.

31. Didelot X, Wilson DJ. ClonalFrameML: Efficient Inference of Recombination in Whole Bacterial Genomes. 2015. doi:10.1371/journal.pcbi.1004041.

32. Mostowy R, Croucher NJ, Andam CP, Corander J, Hanage WP, Marttinen P. Efficient Inference of Recent and Ancestral Recombination within Bacterial Populations. Mol Biol Evol. 2017;34:1167–82.

33. Seemann T. tseemann/snippy. 2021. https://github.com/tseemann/snippy. Accessed 22 Jan 2021.

34. Page AJ, Taylor B, Delaney AJ, Soares J, Seemann T, Keane JA, et al. SNP-sites: rapid efficient extraction of SNPs from multi-FASTA alignments. Microb Genomics. 2016;2:e000056.

35. Seemann T. Prokka: Rapid prokaryotic genome annotation. Bioinformatics. 2014;30.

36. Katoh K, Standley DM. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. Mol Biol Evol. 2013;30.

37. Nguyen LT, Schmidt HA, Von Haeseler A, Minh BQ. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol Biol Evol. 2015;32.

38. Corander J, Marttinen P, Sirén J, Tang J. Enhanced Bayesian modelling in BAPS software for learning genetic structures of populations. BMC Bioinformatics. 2008;9:539. doi:10.1186/1471-2105-9-539.

39. Rabiner L, Juang B. An introduction to hidden Markov models. IEEE ASSP Mag. 1986;3:4–16.

40. Nylander J. nylander/catfasta2phyml. Perl. 2021. https://github.com/nylander/catfasta2phyml. Accessed 19 Jul 2021.

41. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2. doi:10.1093/bioinformatics/btq033.

42. Croucher NJ, Harris SR, Fraser C, Quail MA, Burton J, van der Linden M, et al. Rapid pneumococcal evolution in response to clinical interventions. Science. 2011;331:430–4. doi:10.1126/science.1198545.

43. Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. Accessed 8 Feb 2021.

44. Bolger AM, Lohse M, Usadel B. Trimmomatic: A flexible trimmer for Illumina sequence data. Bioinformatics. 2014;30.

45. Hadfield J, Croucher NJ, Goater RJ, Abudahab K, Aanensen DM, Harris SR. Phandango: an interactive viewer for bacterial population genomics. Bioinformatics. 2018;34:292–3. doi:10.1093/bioinformatics/btx610.

46. Corander J, Waldmann P, Sillanpää MJ. Bayesian analysis of genetic differentiation between populations. Genetics. 2003;163:367–74.

47. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. Bioinformatics. 2012;28:2520–2. doi:10.1093/bioinformatics/bts480.

48. Petit RA, Read TD. Bactopia: a Flexible Pipeline for Complete Analysis of Bacterial Genomes. mSystems. 5:e00190-20. doi:10.1128/mSystems.00190-20.

49. Schwengers O, Hoek A, Fritzenwanker M, Falgenhauer L, Hain T, Chakraborty T, et al. ASA3P: An automatic and scalable pipeline for the assembly, annotation and higher-level analysis of closely related bacterial isolates. PLOS Comput Biol. 2020;16:e1007134. doi:10.1371/journal.pcbi.1007134.

50. Quijada NM, Rodríguez-Lázaro D, Eiros JM, Hernández M. TORMES: an automated pipeline for whole bacterial genome analysis. Bioinformatics. 2019;35:4207–12. doi:10.1093/bioinformatics/btz220.

51. Ewels PA, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, et al. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol. 2020;38:276–8. doi:10.1038/s41587-020-0439-x.

**Figure Captions**

**Figure 1.** Background and composition of the BactPrep pipeline module "ALL". BactPrep identifies core/accessory genes from the whole genome sequencing (WGS) input and reconstructs the pangenome of the population(A). Module "All" is composed by the modules "wgsRecomb" (B), "coreGen" (C), and "coreRecomb" (D). Module “wgsRecomb” identifies the recombination regions from the bacterial WGS alignment and calls single nucleotide polymorphisms (SNPs) from recombination-free regions. The module "coreGen" identifies core/accessory genes from the WGS input, reconstructs the pangenome, and concatenates core gene alignments into a single genetically conserved core genome alignment. The module "coreRecomb" detects and masks recombination regions from the alignment of the core genes, creates a new core genome alignment, and calls SNPs from that alignment.

**Figure 2.** Overall implementation of the BactPrep pipeline. The pipeline is composed of three main modules and two accessory ones that produce clonal alignments and recombination-free phylogenies. The core modules in the pipeline are implemented under the module “ALL”, which will perform a detailed and comprehensive analysis for the whole genome sequencing dataset. Each module (core or accessory) can also be run independently if the user wants to perform a specific analysis.

**Figure 3.** Middle point rooted PMEN1 maximum likelihood phylogenies inferred with alignments produced in “coreGen” module (A), which creates a core gene concatenation alignment, “wgsRecomb” module (B), which produces a recombination-filtered SNPs alignment, and “coreRecomb” module (C), which produces a recombination-masked core gene concatenation alignment, respectively.

**Figure 4.** Example output of the multi-sequence alignment of the PMEN1 recombination filtered SNP alignment produced by the module “wgsRecomb”, with information for each isolate annotated with the metadata specified in the example above. An annotated alignment will also be produced by the “coreGen” and “coreRecomb” modules if the user provides a metadata file.**Figure 5**. Recombination regions in the PMEN1 dataset outputted by the “wgsRecomb” module (shown on the right). The recombination regions were predicted by Gubbins and were visualized using Phandango [45], with sequences aligned to the reference genome GCF000026665.1 (.gff file). The maximum likelihood tree inferred from the recombination filtered SNPs alignment produced by the “wgsRecomb” module is shown on the left.

**Figure 6**. Presence and absence of genes in the PMEN1 pangenome reconstructed using the “coreGen” module (shown on the right). The curve shown on the bottom indicates the frequency of each gene present in the PMEN1 dataset. The maximum likelihood tree (on the left) shows the phylogenetic relationships of the isolates inferred from the core gene concatenation alignment, which were also produced by the “coreGen” module.

**Figure 7**. Illustration of recombination regions of the 865 core genes in the PMEN1 dataset identified using the “coreRecomb” module. Roary was the software used to identify the recombination regions. The maximum likelihood phylogeny reconstructed based on the concatenation of the core genome is shown on the left, and the colored matrix on the right represents the recombination regions identified by fastGear from every core gene. The color yellow indicates the conserved region, while the other colors indicate different source lineages within the dataset, which were predicted by the BAPs clustering algorithm [46].

**Figure 8**. Scatter plot showing the number of ancestral recombination versus the number of recent recombination detected for each gene in the pangenome of the PMEN1 dataset identified using analysis implemented in the “panRecomb” module. The top four most frequently recombined genes in the pangenome of the PMEN1 dataset were labelled manually in this plot.