Chapter 5

Step-by-Step Bacterial Genome Comparison

Dennis Carhuaricra-Huaman and João Carlos Setubal

Abstract

Thanks to advancements in genome sequencing and bioinformatics, thousands of bacterial genome sequences are available in public databases. This presents an opportunity to study bacterial diversity in unprecedented detail. This chapter describes a complete bioinformatics workflow for comparative genomics of bacterial genomes, including genome annotation, pangenome reconstruction and visualization, phylogenetic analysis, and identification of sequences of interest such as antimicrobial-resistance genes, virulence factors, and phage sequences. The workflow uses state-of-the-art, open-source tools. The workflow is presented by means of a comparative analysis of *Salmonella enterica* serovar Typhimurium genomes. The workflow is based on Linux commands and scripts, and result visualization relies on the R environment. The chapter provides a step-by-step protocol that researchers with basic expertise in bioinformatics can easily follow to conduct investigations on their own genome datasets.

Key words Comparative genomics, Pangenome, Bacteria, Phylogenetic analysis

1 Introduction

Bacterial genomics started in 1995 with the publication of the complete genome of *Haemophilus influenzae*, enabling a more profound comprehension of the organism's biology [1]. Shortly after, the *Mycoplasma genitalium* genome was sequenced. The availability of these two genomes gave rise to comparative genomics, revolutionizing prokaryotic biology [2]. Comparative genomics includes many analyses, the most basic of which is the determination of which genes are present or absent in a particular genome with respect to others. Such information helps understand the genome evolution and the basis of phenotypic differences among related organisms.

Early comparative studies already showed that large differences in gene content may occur between genomes of the same prokaryotic species. When three *Escherichia coli* genomes became available in 2002, comparative analysis revealed that only 39.2% of the genes were shared by these three genomes [3]. Soon, other similar

João Carlos Setubal et al. (eds.), *Comparative Genomics: Methods and Protocols*, Methods in Molecular Biology, vol. 2802,
https://doi.org/10.1007/978-1-0716-3838-5_5,

observations were made about other bacterial species, showing the remarkable plasticity of such genomes, eventually giving rise to the pangenome concept. The pangenome is the set of all non-redundant genes present in a given set of genomes. The differential gene content among genomes from the same species comes about because of extensive horizontal gene transfer (HGT) and gene loss, two of the main forces driving the evolution of prokaryotes [4].

In the last 20 years, the development of cheaper and more accessible sequencing methods caused an exponential increase in the amount of bacterial genomes [5]. This, in turn, stimulated the development of genome informatics, or computational methods for genome analysis, in particular, comparative analysis. A few examples are the genomic epidemiology of pathogenic bacteria [6], pathogenesis and niche specialization [7], discovery of genes associated with virulence and antimicrobial resistance [8, 9], identification of antigens through reverse vaccinology [10], discovery of antiphage defense systems [11], among others.

The use of massive sequencing and large genomic datasets may lead to important discoveries in various fields, such as medicine and plant pathology. However, frequently, there is a mismatch between the capability of obtaining large genomic datasets and the ability of effectively analyzing such data. Genome sequences are only useful if there are adequate capabilities for annotation and comparative analysis, including computational infrastructure and skilled scientists in data analysis and programming.

This chapter describes a step-by-step genomic comparison protocol, using the bacterium *Salmonella enterica* serovar Typhimurium (*S.* Typhimurium) as an example. *S.* Typhimurium is one of the most important gastrointestinal pathogens of humans, and is carried by livestock. In this chapter, we use genomes of *S.* Typhimurium downloaded from a public database.

The protocol presented in this study uses state-of-the-art bioinformatics tools that are freely available. This protocol begins with genome annotation and pangenome reconstruction, followed by gene content analysis and visualization and phylogenetic reconstruction. The protocol concludes with the identification of sequences of interest, including antimicrobial-resistance genes (ARGs), virulence genes, and phage sequences. The aim of this protocol is to provide a user-friendly guide that can be used as a template by researchers who are interested in applying the same analyses to their own genome datasets.

2 Requirements and Assumptions

This chapter assumes basic knowledge of Unix/Linux and R. All analyses can be run on a desktop computer running Linux/Unix or Mac OS. Most programs can be executed using bash shell commands. We adopt the convention of presenting commands

executed on the Linux shell preceded by the "$" symbol and the label **bash shell**. We also present R code, which can be executed in the RStudio environment (https://posit.co/download/rstudio-desktop/). R code sections are preceded by the label **R script**.

This protocol was tested on a desktop computer with 8 GB of RAM, processor Intel Core i7-6500U CPU 2.50 GHz, with 200 GB of disk space, and running Linux (Ubuntu version 20.04 system 64-bit).

3 Datasets

Bacterial genome sequences can be retrieved from a variety of public repositories; examples include GenBank [12], BIGSdb [13], IMG [14], BV-BRC [15], or Enterobase [16]. In this chapter, we use genome sequences of *Salmonella enterica* ser. Typhimurium, which we abbreviate as **SeT**. Our dataset consists of 12 genomes from strains LT2, 798, D23580, DT104, DT2, L-3553, SL1344, T000240, U288, SO4698-09, SO9207-07, and SO9304-02, which have been isolated from different hosts [17], and two SeT genomes isolated from guinea pigs (SMVET11 and SMVET22) sequenced by our group [18]. The accessions of these 14 SeT genomes are shown in Table 1 and can be downloaded from the GenBank genome database. Table 1 also

Table 1
The genomes of *S. Typhimurium* used in this chapter. Each genome sequence can be downloaded from GenBank using the Assembly identifier in the last column

Genome	Host	Country	MLST	Phylogroup	Assembly
798	Pig	USA	ST19	β	GCA_000252875.1
D23580	Human	Africa	ST313	β	GCA_000027025.1
DT104	Cattle	NA	ST19	α	GCA_000493675.1
DT2	Pigeon	Germany	ST128	β	GCA_000493535.2
L-3553	Cattle	Japan	ST19	β	GCA_000828595.1
LT2	Human	USA	ST19	α	GCA_000006945.2
SL1344	Cattle	UK	ST19	β	GCA_000210855.2
T000240	Human	Japan	ST19	α	GCA_000283735.1
U288	Pig	UK	ST19	α	GCA_000380325.1
SO4698-09	Cattle	UK	ST34	α	GCA_001540845.1
SO9207-07	Pig	UK	ST19	α	GCA_903989485.1
SO9304-02	Cattle	UK	ST19	β	GCA_902500315.1
SMVET11	Guinea Pig	Peru	ST19	α	GCA_024721515.1
SMVET22	Guinea Pig	Peru	ST19	α	GCA_024721395.1

contains additional information associated with each sample, such as host source, country, and genetic characteristics that were identified in the mentioned studies, such as the sequence type (ST), a method based on the allelic profile of seven housekeeping genes; and the phylogroup, which denotes a strain's placement in the SeT phylogenetic tree, in clades α or β [17].

4 Software

The software tools that will be used in this workflow are freely available and summarized in Table 2, including links to the websites from which these can be downloaded and installed.

Table 2
List of software tools used to perform comparative genomics in this chapter

Tool	Description	References	Source
ABRicate	Genome screening for ARGs and VFs	[19]	https://github.com/tseemann/abricate
Easyfig	Genome sequence comparison	[20]	https://mjsull.github.io/Easyfig/
eggNOG-mapper v.2	Functional enrichment	[21]	https://github.com/eggnogdb/eggnog-mapper
GenBank	Genome database	[12]	https://www.ncbi.nlm.nih.gov/genbank/
Gephi	Network visualization	[22]	https://gephi.org/users/download/
ggplot2	R package	[23]	https://cran.r-project.org/web/packages/ggplot2/
ggtree	R package	[24]	https://github.com/YuLab-SMU/ggtree
Gubbins	Recombination	[25]	https://github.com/nickjcroucher/gubbins
IQ-TREE 2	Phylogeny	[26]	https://github.com/iqtree/iqtree2
limma	R package	[27]	https://bioconductor.org/packages/release/bioc/html/limma.html
micropan	R package	[28]	https://github.com/larssnip/micropan
Panaroo	Pangenomic analysis	[29]	https://github.com/gtonkinhill/panaroo
PPanGGOLiN	Pangenome graph	[30]	https://github.com/labgem/PPanGGOLiN
pheatmap	R package	[31]	https://github.com/cran/pheatmap
Prokka	Genome annotation	[32]	https://github.com/tseemann/prokka
R	Environment for analysis	[33]	https://www.r-project.org/

(continued)

Table 2
(continued)

Tool	Description	References	Source
Scoary	Pangenome-wide association studies analysis	[34]	https://github.com/AdmiralenOla/Scoary
SNP-sites	SNPs extraction for phylogenetic analysis	[35]	https://github.com/sanger-pathogens/snp-sites
VirSorter2	Phage sequence prediction	[36]	https://github.com/jiarong/VirSorter2

5 Genome Annotation

Genome annotation is the process whereby the location and functional characteristics of genes and other genetic elements are added to the raw genome sequence. Nowadays, it is an automated process, with occasional manual curation in special cases and/or for particular genes. With more than a million prokaryotic genomes available in public databases, accurate automated genome annotation is crucial for many downstream genomic analyses [37]. **Prokka** [32] is the most cited command-line tool for prokaryote genome annotation, with an easy installation and short runtime performance (5 min for a typical bacterial genome of about 5 Mbp). The Prokka pipeline uses Prodigal [38] to predict coding sequences (CDSs) and other software tools for RNA annotation. **Bakta** [39] was introduced using a similar workflow but providing a more comprehensive annotation. Bakta predicts pseudogenes and small proteins that are not annotated by Prokka. Both tools provide a variety of output files, such as .gff, .gbk, and .faa files, which are commonly used for comparative analysis.

Here, we annotate all 14 genomes of our dataset with Prokka. We loop through all the genomes using the following command (which assumes that the FASTA files for all genomes are available in the directory from which the command is issued).

Bash Shell

```
$ for i in *.fasta; do prokka --kingdom Bacteria --genus Salmonella
--prefix "${i%%.*}" --locustag "${i%%.*}" --outdir "${i%%.*}"
--compliant "$i"; done
```

Here we are asking Prokka to annotate all assemblies in FASTA format (.fasta); name the output files (--prefix) and locus tag (--locustag) as the isolate name; and make the annotations compliant with NCBI standards (--compliant).

Table 3
Annotation statistics of SeT genomes. In the last column, the number represents the total number of rRNA gene units that were detected (among three possible: 5S, 23S, 16S)

Genome	Size (bp)	# CDS	# tRNA	# rRNA units
798	4,970,096	4685	83	22
D23580	4,879,400	4554	88	22
DT104	5,027,665	4743	85	22
DT2	4,814,801	4583	84	22
L-3553	5,184,452	4925	85	22
LT2	4,951,383	4620	86	22
SL1344	5,067,450	4763	86	22
T000240	5,069,994	4784	84	22
U288	5,017,059	4707	85	22
SO4698-09	5,037,238	4750	85	22
SO9207-07	4,916,754	4585	88	22
SO9304-02	5,045,986	4789	86	22
SMVET11	4,851,410	4542	76	8
SMVET22	5,095,938	4863	75	8

For each genome annotated, a directory with the same name is created containing annotation files. Three output files (.gff, .gbk, and .faa) will be used for downstream analysis. Table 3 summarizes the annotation features of all 14 genomes.

We now describe the tool eggNOG-mapper, which can map protein-coding genes, or coding sequences (CDSs), in a genome to the orthologous families in the eggNOG database [40]. eggNOG-mapper offers functional annotation that includes KEGG pathways [41], COG functional categories [42], carbohydrate-active enzymes (CAZymes) families [43], and gene ontology terms [44]. Protein sequences (.faa file) generated by Prokka can be uploaded to the eggNOG-mapper website (http://eggnog-mapper.embl.de/), which uses precomputed orthologous groups from the eggNOG database v.6 (http://eggnog6.embl.de) for fast functional annotation. eggNOG-mapper can also be installed locally, but the storage requirement is high due to database size (around 40 GB is needed for the eggNOG annotation databases and additional disk space if the HMMER option is used). eggNOGG-mapper can be executed locally with the following command.

Bash Shell

```
$ for i in *.faa; do emapper.py -i "$i" --output "$i"_eggnog; done
```

For each input, eggNOGG-mapper returns an annotation file (*samplename.emapper.annotations*) which provides the predictions for each query in TSV format (tab-separated values). Relevant columns are as follows:

Query: The query sequence name

GOs: List of predicted GO terms

COG_category: List of predicted COG categories

KEGG_ko: List of predicted KEGG orthologs

CAZy: List of predicted CAZy orthologs

6 Pangenome Reconstruction and Visualization

The pangenome is the set of all gene families present in a given group of genomes belonging to a specific taxon [45]. The gene set can be subdivided into the "core" and "accessory" genomes. The core genome is composed of genes that are present in all members of the group, whereas those genes that are only present in some members represent the accessory genome [46]. This protein-coding gene-based approach is the most common in prokaryotic pangenome reconstruction, and it requires a preliminary step of gene annotation to locate and extract the coding sequences. Depending on the program used to compute the pangenome, the categories "core" and "accessory" can be further subdivided (as will be the case in the example of this chapter, as discussed later).

Before delving into the nuts and bolts of pangenome computation for the genome set we are using, it is worth citing the web-based tool PanExplorer [47], which provides a workflow for this computation, including visualization components. It is a useful and effective tool, which saves users from having to run individual programs to achieve the results illustrated here.

6.1 Ortholog Gene Computation and Clustering

Identifying orthologous genes in different genomes is the first step for pangenome reconstruction and gene content comparison. Orthologs can be identified by carrying out similarity searches between genes from different genomes using, for example, BLAST [48], CD-HIT [49], or DIAMOND [50], and then clustering the results into orthologous groups using the Markov clustering algorithm (MCL) or by looking at triangles of pairwise best hits [51, 52].

Panaroo [29] is currently one of the most popular tools for pangenome reconstruction and is the tool we use in this example. Given a set of annotated genomes, in the form of .gff or .gbk files, Panaroo uses CD-HIT for sequence similarity search and clustering in order to obtain gene clusters with a high similarity threshold (98% by default). Some of these clusters are then merged according to synteny information, which is also used to find missing genes and correct for possible errors in assembly and annotation.

In order to reconstruct the pangenome of our dataset of SeT genomes, the annotation files (.gff format) previously generated by Prokka are used as input to Panaroo. The basic command for pangenome calculation is as follows.

Bash Shell

```
$ panaroo -i *.gff -o results_SeT --clean-mode
strict -a core --aligner mafft
```

The parameter "aligner" indicates the program that Panaroo should use to perform multiple alignments of the genes in each cluster. In this case, we use the program MAFFT [53]. Panaroo will create a directory called "results_ST" containing a set of output files. The most important ones are as follows:

summary_statistics.txt: A summary text file reports the number of genes discovered in the analyzed data, categorized into core, soft-core, shell, and cloud, based on their occurrence frequency within the studied genomes. Soft-core, shell, and cloud genomes are concepts specific to Panaroo, and are a refinement of the accessory genome concept; their definitions are as follows. These categories can be represented in pie charts (Fig. 1a).

pan_genome_reference.fa: A multi-fasta file that contains a unique representative nucleotide sequence extracted from each cluster present in the pangenome.

gene_presence_absence.csv: This spreadsheet contains the description of each gene in the pangenome.

gene_presence_absence.Rtab: A binary tab-separated matrix of presence/absence of each gene in the pangenome, the presence of a gene is coded as 1 and absence as 0.

core_gene_alignment.aln: A file that contains an alignment of all core genes (by default, at least 95% of all samples). This can be used for phylogenetic analysis.

After constructing the pangenome, all CDSs are now clustered into orthologous groups of genes (OGs). Hereinafter, OGs will be referred to simply as genes. Panaroo assigns genes to the core (present in at least 99% of genomes) or to the accessory genomes,

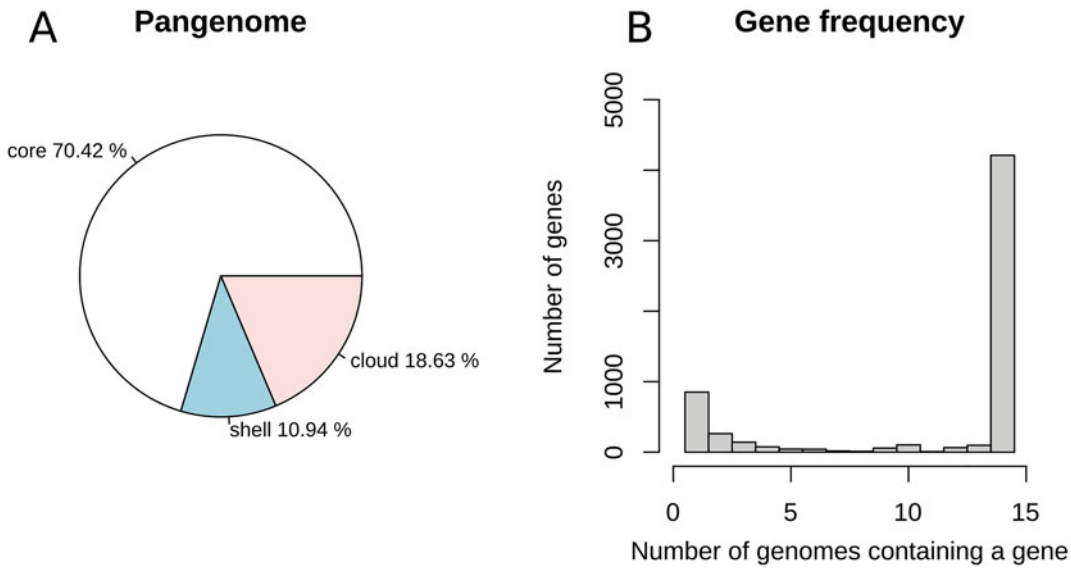

Fig. 1 (**a**) The pie chart displays the proportion of core, shell, and cloud of the SeT pangenome using 14 genomes. (**b**) Gene frequency distribution across the number of genomes; the typical asymmetric U-shape is observed with most genes present in all genomes

which is subdivided into the soft-core (95–99% of genomes), shell (15–95% of genomes), and cloud (less than 15% of genomes). These statistical data of the pangenome can be represented in a pie chart and a histogram of gene frequencies. To generate these plots, we will use the binary tab-separated matrix of presence/absence (gene_presence_absence.Rtab) in the following R script.

R Script

```
# Read the presence/absence matrix obtained from Panaroo
data <- read.table("gene_presence_absence.Rtab",sep = "\t", row.names = 1,
header = T, check.names = F)
# Calculate pangenome size
pangenome_size = nrow(data)
# Calculate the core, shell and cloud sizes as assigned by panaroo
core_size <- length(rowSums(data)[rowSums(data) >= 0.99*ncol(data)])
shell_size <- length(rowSums(data)[rowSums(data) < 0.99*ncol(data) &
rowSums(data) > 0.15*ncol(data)])
cloud_size <- length(rowSums(data)[rowSums(data) <=  0.15*ncol(data)])
par(mfrow = c(1, 2),pin = c(2.5, 2.5))
# Plot a pie chart displaying the core and accessory proportions; this is
Figure 1A
slices <- c(core_size,shell_size,cloud_size)
pct <- round(slices/sum(slices)*100,2)
lab <- paste(c("core", "shell", "cloud"),pct,"%",sep=" ")
pie(slices, labels = lab, main="Pangenome", cex=0.8)
```

```
# Plot a histogram of genes families in the 14 S. Typhimurium genomes;
this is Figure 1B
hist(rowSums(data), xlab = "Number of genomes containing a gene",
     ylab = "Number of genes", main = "Gene frequency",
     ylim = c(0,5000), xlim = c(0,ncol(data)+1),
     breaks = seq(min(rowSums(data))-0.5, max(rowSums(data))+0.5, by = 1))
```

In our example, the core genome represents around 70% (4210 genes) of the total SeT pangenome (5978 genes), whereas the accessory section (shell and cloud) represents about 30% (1768 genes). A value of 70% for the core genome is relatively high and can be explained by the fact that the genomes chosen are all strains of one serovar of the *Salmonella enterica* species; such genomes tend to share a large fraction of their protein-coding genes.

In most species, when the number of genomes analyzed increases, the size of the core genome tends to decrease, because newly added genomes may not have all the genes that are part of the previous core. The opposite happens with the accessory genome, which tends to increase with more genomes added to the analysis (however, *see* Subheading 6.2, for the concept of open and closed genomes). It is important to note that different pangenome reconstruction programs may yield different pangenome estimates because they use distinct ortholog identification methods, identity cutoff values, or they may differentially account for assembly and annotation errors [54].

The gene frequency histogram (Fig. 1b) displays a "U-shape" distribution, where most genes are present either in only one genome (single-genome accessory genes) or in all genomes (core genes); the intermediate-frequency accessory genes generally have lower counts. This "U-shape" distribution is typically found in prokaryote genomes and is the result of the interplay between gene loss and horizontal gene transfer [4].

6.2 Open and Closed Pangenomes

When analyzing a set of genomes using the concept of pangenome, one important question to ask is whether the pangenome for that particular set is open or closed. A pangenome is classified as "open" when it always grows when new genomes are added to the computation. By contrast, a closed pangenome means that, after a certain number of genomes have been added, any new genomes for that taxon will not contain any genes not seen before.

The openness/closeness of a pangenome can be estimated by constructing rarefaction curves and applying a statistical model as

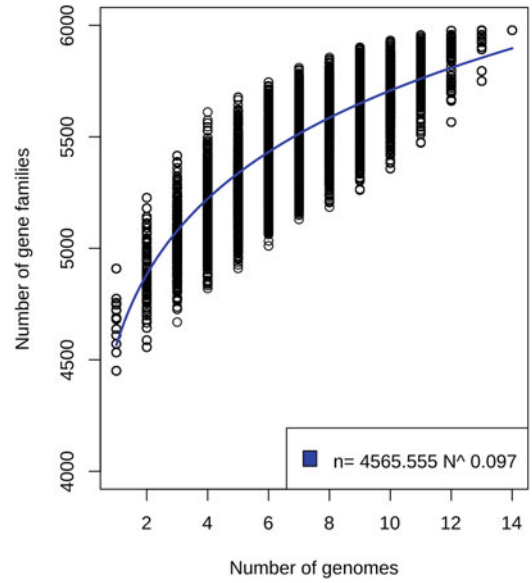

Fig. 2 Rarefaction curve of the SeT pangenome calculated from random combinations of strains. Fit to the displayed formula is shown in blue. The legend shows the power law function with parameter "γ" of 0.097, indicating an open pangenome

proposed by Tettelin et al. [45]. A rarefaction curve is the cumulative number of unique ortholog genes we observe as more and more genomes are added to the dataset (Fig. 2). The Heaps law model fits the rarefaction curve of the pangenome according to the function:

$$n = k \times N^{\gamma}$$

where:

- n is the expected number of distinct genes for a given number of genomes (N) and
- k and γ are free parameters that are determined empirically.

According to Heaps' law (which is a power law), when $\gamma > 0$, the pangenome is considered open, and when $\gamma < 0$, the pangenome is considered closed [45].

The rarefaction curve was generated using *rarefaction()* function of the micropan package [28] with 100 permutations. We then fitted a power law function and recovered the value of k and γ parameters. The script in R is shown as follows.

R Script

```
library(micropan)
data <- read.table("gene_presence_absence.Rtab", sep = "\t",
header = T, row.names = 1, check.names = F)
# transpose dataframe
df_t <- t(data)
rownames(df_t) <- colnames(data)
colnames(df_t) <- rownames(data)
# generate rarefaction matrix
rar<-rarefaction(df_t, n.perm = 1000)
mrar <- as.matrix(rar)
# melt rat
rm <- melt(mrar[-1,-1])
# fit the power law to the rarefaction data
fit_h<-lm(log(value) ~ log(Var1), data = rm)
# extract parameters k and γ (gamma)
k <- exp(summary(fit_h)$coef[1])
gamma <- summary(fit_h)$coef[2]
alpha <- 1 -gamma

 # plot the rarefaction curve and power law function
 plot(value ~ Var1,data = rm,xlab="Number of genomes",
         ylab="Number of gene families",ylim=c(4000, 6000))
 curve(k*x^gamma,1,14,add=TRUE, col="blue",lwd=2)
 legend(x = "bottomright", legend =
 paste("n=",round(k,3),"N^",round(gamma,3)), fill = "blue")
```

The value we obtained for the γ parameter was 0.097, suggesting that the SeT pangenome is open; this is consistent with results from the literature [55, 56]. However, there is also evidence that the pangenome for *Salmonella enterica* is closed [57]. Conclusions regarding the pangenome openness and closeness of species can be inconsistent between studies when a small number of genomes are used. The fact that the value for the γ parameter that we obtained is close to 1 is also indicative that the closeness or openness of this pangenome should not be a firm conclusion.

Alternative metrics can quantify pangenome diversity such as genomic fluidity. This is a metric used to quantify the degree of dissimilarity in gene content between genomes. In the case of two genomes, genomic fluidity is calculated as the proportion of genes that are specific to one genome out of the total of genes present in both genomes. For a population, fluidity is determined by averaging genome fluidity calculation over all pairs of genomes [58, 59].

6.3 Comparison of Gene Content

Graphical representations of gene content variation using the presence/absence matrix can be depicted by Venn diagrams, presence/absence binary maps, or principal component analysis (PCA).

Venn diagrams are only useful to represent gene content relationships for, at most, a handful of genomes. Above that, Venn diagrams become progressively more complicated. The function *vennDiagram ()* of the *limma* R package can be used to draw a gene content Venn diagram as illustrated in Fig. 3. The code used to plot Fig. 3 from the presence/absence matrix in the R environment is as follows.

R Script

```
library(limma)
# import the gene presence/absence matrix generated by panaroo
data <- read.table("gene_presence_absence.Rtab", sep = "\t", header = T,
check.names = F, row.names = 1)
# select only four genomes from the dataset
counts <- vennCounts(data[6:9])
# plot Venn diagram
vennDiagram(counts, circle.col = c("red", "blue", "green3", "yellow"), cex=1)
```

To visualize variation in the genetic content of more than a handful of genomes, the presence/absence heatmap is more practical. In presence/absence heatmaps, each row represents a genome, while each column represents a gene (orthologous group); the presence of a gene is denoted as a colored cell, whereas an absent gene is represented by a non-colored cell. The clustering methods to group patterns of gene presence/absence in both genomes (rows) and genes (columns) help identify genes that are unique to certain subsets of genomes or that are shared among many genomes. We used the *pheatmap* package [31] to generate a heatmap representation of the pangenome from the presence/absence matrix (Fig. 4). The script in R is shown as follows.

R Script

```
library(pheatmap)
# read presence/absence matrix of gene content from panaroo
data <- read.table("gene_presence_absence.Rtab",sep = "\t",header = T,
row.names = 1, check.names = F)
# transpose dataframe
df_t <- t(data)
rownames(df_t) <- colnames(data)
colnames(df_t) <- rownames(data)
# convert dataframe to matrix
pange <- as.matrix(df_t, as.numeric)
rownames(pange) <- colnames(data)
# plot heatmap showed in Figurer 4A
hm <- pheatmap(pange, clustering_distance_rows = "manhattan",
```

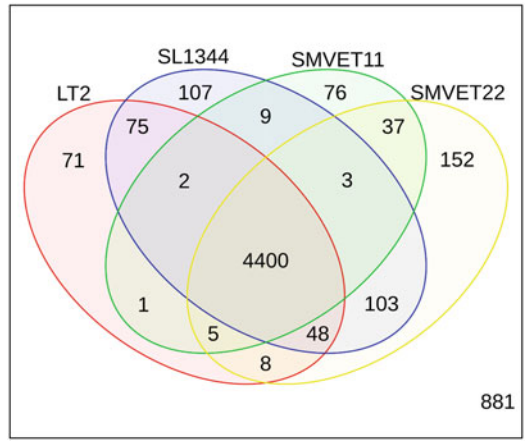

Fig. 3 Illustration of a gene content Venn diagram for four SeT genomes (LT2, SMVET11, SMVET22, and SL1344). The numbers inside each region indicate the number of genes shared in that region. For example, the region that represents the intersection of all four ellipses contains 4400 genes, meaning that the four genomes share these many genes. The number outside all ellipses (881) represents the number of genes that are absent in these four genomes but are present in the other ten genomes of the dataset

```
      clustering_method = "ward.D", color = c("white", "skyblue4"),
      clustering_distance_cols = "manhattan", show_colnames = F,
      cluster_cols = T, cluster_rows = T, legend = F)
# recover data matrix from heatmap after clustering
reorder <- data[hm$tree_col[["order"]],]
reorder <- reorder[,hm$tree_row[["order"]]]
# plot heatmap of a selected region of the pangenome matrix (Figure
4B)
p2 <- pheatmap(t(reorder[5816:5890,]), show_colnames = T,
      cluster_cols = F, cluster_rows = F, legend = F,
      color = c("white", "skyblue4"))
```

Figure 4a shows the full pangenome gene content with the presence of a gene in blue and the absence without color. The hierarchical clustering of genomes and genes allows us to observe different presence/absence patterns. For example, there is one cluster of 85 genes absent in four SeT isolates (SO4698-09, SO9207-07, D23580, and DT2), but present in the other ten genomes (Fig. 4b). Virulence plasmid-borne *spvB* and *spvC* genes stand out among these clusters of genes. The isolates that lack *spvB* and *spvC* genes may exhibit reduced virulence [60].

PCA is a statistical method that helps analyze genomic diversity and identify possible associations of genomes based on gene content [61]. In a PCA, the gene presence/absence matrix is first

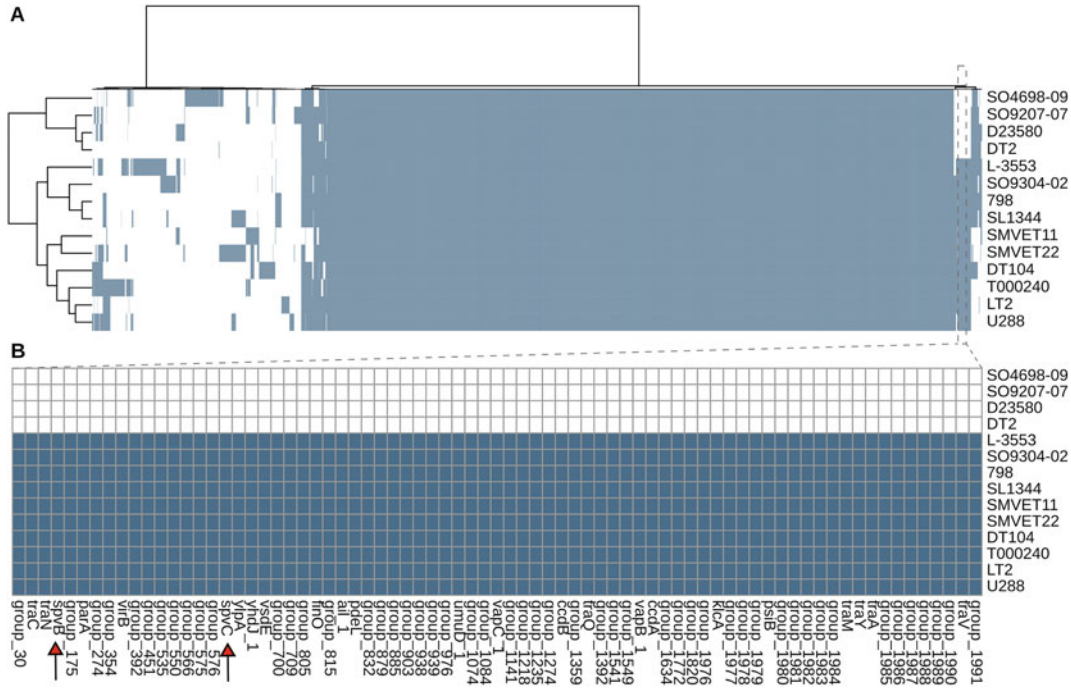

Fig. 4 (**a**) The heatmap represents the gene content, in terms of presence/absence of genes in the SeT pangenome inferred from the comparison of fourteen genomes. Gene presence is shown in blue and gene absence in white. A total of 5976 orthologous groups (genes) representing the full pangenome size are displayed. Dendrograms clustering genes (*X*-axis) and genomes (*Y*-axis) according to gene presence/absence pattern were produced in *pheatmap* package using hierarchical clustering based on the "ward.D" method with "manhattan" distance. (**b**) The heatmap highlights a cluster of 85 genes (dashed box in (**a**)) absent in four genomes (SO4698-09, SO9207-07, D23580, and DT2) but present in the other ten genomes. These include the *spvB* and *spvC* genes (red arrows), which are plasmid-carried virulence factors

transformed into a set of principal components that capture the variation in gene content across the genomes. The principal components are then plotted in a two-dimensional space, where each point represents a genome, the position of the point reflects its gene content, and the distance between points is a measure of how different two genomes are in terms of gene content. The points can be colored according to any characteristic of the strain (e.g., host, isolation country, sequence type). Since we have metadata associated with each SeT strain in our dataset (Table 1), we will use it to color the strains according to the host source and phylogroup variables. The R package ggplot2 can be used to perform a PCA from the gene presence/absence matrix.

R Script

```
library(ggplot2)
# read the presence/absence matrix generated by panaroo
data <- read.table("gene_presence_absence.Rtab", sep = "\t", header = T,
row.names = 1, check.names = F)
# transpose dataframe
df_t <- t(data)
rownames(df_t) <- colnames(data)
colnames(df_t) <- rownames(data)
# read the metadata (table 1 of this chapter) as a dataframe
meta <- read.table("metadata.tab", header = T, sep = "\t", row.names = 1)
# merge presence/absence matrix with metadata into one dataframe
dafr <- data.frame(merge(df_t, meta, by = 0))
# compute principal components on the accessory portion of the pangenome
PC<-prcomp(dafr[, c(4211:5979)])
# you can use "Phylogroup" instead of "Host"
PCi<-data.frame(PC$x, Host=dafr$Host)
# plot PCA labeled by Host (or Phylogroup)
ggplot(PCi, aes(x=PC1,y=PC2,fill=Host)) +
  geom_point(size = 5, alpha = 0.5, shape = 21) +
  scale_fill_brewer(palette = "Set1") +
  theme_bw()
```

The PCA analysis of SeT gene content was unable to distinguish isolates from different host sources (Fig. 5a). However, isolates from phylogroups α and β were separated by the second component (Fig. 5b), which means that there is differential gene content between isolates from these two phylogroups.

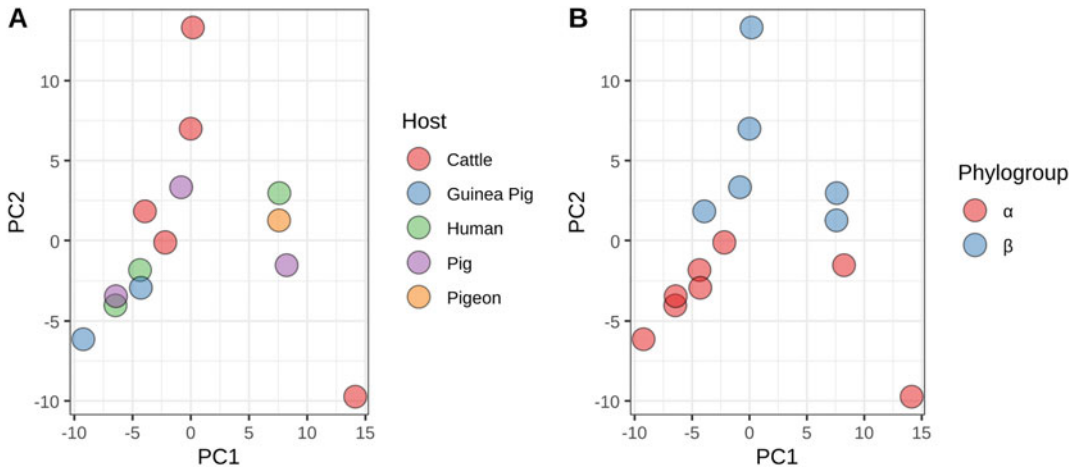

Fig. 5 Principal component analysis of the 14 SeT genomes. The graph is generated from the gene presence/absence matrix. Each dot represents a genome, which is colored according to host source (**a**) or phylogroup (**b**). A horizontal line around *y* value 0 in (**b**) provides a good separation between the two phylogroups, suggesting that phylogroup is indeed a good determinant of gene content, and that separation is provided by the second principal component (PC2)

6.4 Pangenome-Wide Association Studies

Genome-wide association study (GWAS) is an approach for studying genotype-phenotype associations. In prokaryotes, the GWAS approach is applied to pangenomes (pan-GWAS) in order to identify genes associated with specific phenotypes, such as host source, virulence, and antibiotic resistance [62]. Scoary [34], a popular tool for pan-GWAS analysis, correlates gene presence/absence from pangenome analysis with phenotypic traits. The presence/absence matrix generated by Panaroo can be used as input for Scoary.

Scoary needs two inputs: the *gene_presence_absence.csv* file generated by Panaroo and a trait file (csv format) containing phenotypic traits. For example, if the trait is resistant to tetracycline and the categories are resistant and susceptible, we could use "0" to indicate susceptibility and "1" to indicate resistance.

Bash Shell

```
$ scoary.py -g <gene_presence_absence.csv> -t
<traits.csv>
```

Here, we cannot use our dataset because GWAS requires more than just 14 genomes to assess the association to a specific phenotype (for additional information, see [63]). The power to find statistically significant associations is affected by several factors such as sample size, allele frequency, population diversity and structuring [64]. A large number of genomes (hundreds) are typically used in this kind of analysis [62].

7 Phylogenetic Tree Based on Core Genome Alignment

Phylogeny inference is a vast topic; we include here an example of phylogeny inference because it is relatively straightforward to obtain a phylogeny given the alignment file *core_gene_alignment.aln* generated by Panaroo. For more information on phylogeny inference, we refer the reader to [65]. Most of the phylogeny inference tools use a maximum likelihood approach, such as RAxML [66], FastTree [67], and IQ-TREE 2 [26].

It is important to consider the effect of homologous recombination when reconstructing phylogenies in prokaryotes. Some tools like Gubbins [25] or ClonalFrame [68] can be used to mask recombinant regions before reconstructing a phylogeny.

IQ-TREE 2 offers multiple options, including ultrafast bootstrapping (-B) and ModelFinder to find the best substitutions model (-m). The tree is typically created in Newick format and can be visualized by ggtree package [24] in R. Commands for recombinant regions detection and phylogenetic reconstruction using Gubbins and IQ-TREE 2 starting from *core_gene_alignment.aln* file generated by Panaroo are shown as follows.

Bash Shell

Run Gubbins for detection of recombination in the alignment.

```
$ run_gubbins -p gubbins core_gene_alignment.aln
```

Extract SNPs from the alignment using SNP-sites.

```
$ snp-sites -c gubbins.filtered_polymorphic_sites.
fasta > clean.core.aln
```

Reconstruct the phylogenetic tree using IQ-TREE 2 with 1000 of bootstrap (-B).

```
$ iqtree2 -s clean.core.aln -B 1000 --prefix tree_
clean_ST
```

IQ-TREE 2 takes a few minutes on a standard laptop and generates several files. The tree file (.treefile) in Newick format is used for *ggtree* and *phangorn* package in R for tree visualization.

R Script

```
library(ggplot2)
library(ggtree)
library(phangorn)
# read Newick file generated by IQ-TREE2
tree <- read.tree("tree_clean_ST.treefile")
# set midpoint root
treeMP<-midpoint(tree)
# draw phylogenetic tree
gg <- ggtree(treeMP, layout= "rectangular", right=F) +
  geom_tiplab(size=2.8, linesize=.5,offset = 0.0003,align = T) +
  geom_text2(aes(subset = !isTip, label=label), size = 2,
             hjust=1.2,vjust = -0.3)+
  geom_treescale()
# read metadata information (table 1)
meta <- read.table("metadata.tab", header = T, sep = "\t")
# add host source information to the tree
p1 <- gg %<+% meta[,c(1:2)]
# draw tippoint colored according to host information
p2 <- p1 + geom_tippoint(aes(fill=Host, shape=Host),
                         size=5, alpha=1, colour = "black") +
  scale_fill_brewer(palette = "Set1")+
  scale_shape_manual(values = rep(21,each=12))+
  theme(legend.position = "right")
# draw vertical bars on clades representing α and β clades
p3 <- p2 + geom_cladelab(node=20,barsize=1.5, label="α",
                         offset=0.0008,offset.text=.0) +
           geom_cladelab(node=27,barsize=1.5, label="β",
           offset=0.0008, offset.text=.0)
# plot
p3
```

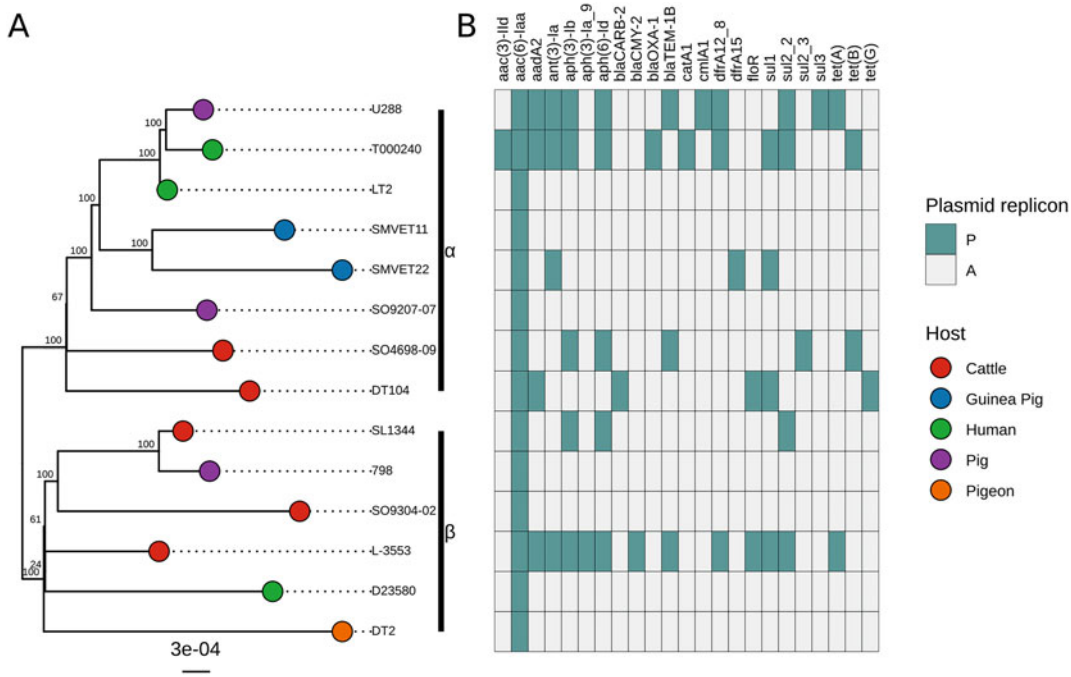

Fig. 6 (**a**) Phylogenetic tree of *S.* Typhimurium reconstructed from the core genome alignment of fourteen genomes using IQ-TREE 2. Tip point circle shapes are colored according to the host source. Vertical black bars represent well-supported clades (α and β phylogroups). (**b**) heatmap of ARGs predicted by ABRicate using the ResFinder database. Turquoise boxes indicate that the gene is present; white boxes show the gene is absent

The phylogenomic tree of SeT (Fig. 6a) shows two main groups, which represent the well-known α and β clades [17], both with high bootstrap support. A probable association of lineages to certain hosts is not evident due to the small number of isolates. However, previous works have shown that certain lineages are associated with some hosts, such as DT8 associated with ducks or ST313 associated with humans in Africa [69].

8 Pangenome Graphs

The most common methods used to generate prokaryotic pangenomes provide a matrix indicating the presence or absence of genes, without regard for gene order or orientation [52, 70]. However, more detailed information on the variability of both gene content and genome structure within a group of genomes may provide valuable information about the evolution of the corresponding species.

Novel algorithms have emerged to address this issue, focused on extending the pangenome framework of microbial diversity to graphical models [71]. A pangenome graph is a model in which nodes represent gene families and edges represent a relation of

genetic contiguity, allowing the exploration of structural and sequence order variation in genomes. Current tools that use this approach in prokaryotic genomes are PPanGGOLiN [30] and PanGraph [72].

While PPanGGOLiN follows the common gene-based approach using annotated .gff genomes as input, PanGraph uses a sequence-based approach to group homologous sequences through nucleotide alignments alone, without relying on annotation. However, this method requires the use of high-quality genomes.

In PPanGGOLiN, a statistical approach is used to classify the gene presence/absence of the pangenome into persistent (gene families present in almost all genomes), shell (gene families present at intermediate frequencies), and cloud (gene families present at low frequency) partitions. PPanGGOLiN uses gff/gbk annotation files as input and yields multiple output files such as gene_presence_absence.Rtab and matrix.cvs, which are also produced by Panaroo. The most important output of PPanGGOLiN is an HDF-5 file named pangenome.h5. It stores all information about the pangenome, including data to construct the graph. We will now use PPanGGOLiN to compute the pangenome from the GFF files of the 14 *S.* Typhimurium genomes.

Bash Shell

```
$ ppanggolin workflow --anno gff_list.tab
```

The gff_list.tab is a tab-separated file containing a list of the strain names and paths to the associated GFF files of all genomes used in the analysis. After the running is completed, PPanGGOLiN produces a directory containing all outputs. To visualize the partitioned pangenome graph we load the pangenome Graph_light.gexf.gz file (this is the file that contains the graph description in terms of nodes and edges) into the Gephi program.

The Gephi software is an open-source tool for exploration and visualization of networks and graphs. We use Gephi with the ForceAtlas2 algorithm and the following parameters: Scaling = 10,000, Stronger Gravity = True, Gravity = 2.0, and edge weight influence = 2.0. The final plot is shown in Fig. 7.

The PPanGGOLiN pangenome graph in Fig. 7 displays the overall gene content diversity of the 14 *S.* Typhimurium genomes. The persistent partition (orange nodes) is the most abundant and corresponds to gene families present in at least 13 genomes (core genome). The shell partition (green nodes) is represented by gene families present in 6–12 genomes and these are arranged in several continuous groups surrounded by longer regions with showing synteny. For example, a region of contiguous genes encoding plasmid proteins belongs to the pLST plasmid of *S.* Typhimurium

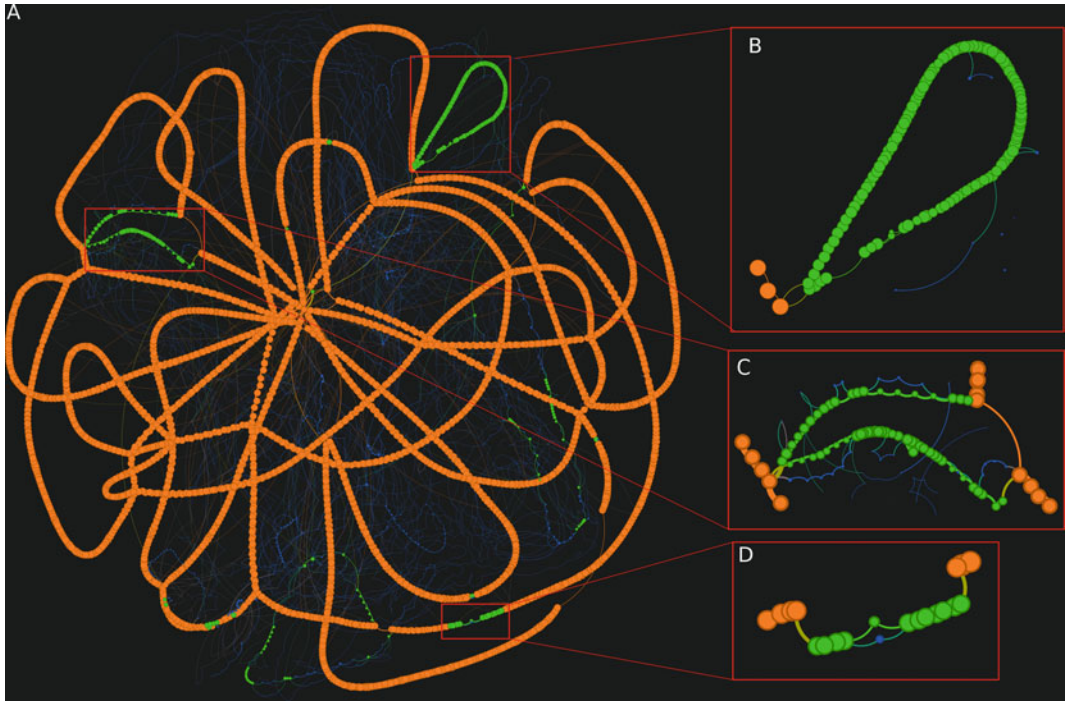

Fig. 7 (**a**) The partitioned pangenome graph was calculated from 14 SeT genomes. Nodes represent gene families and edges represent a relation of genetic contiguity. The persistent, shell, and cloud nodes are colored in orange, green, and blue, respectively. The size of the nodes is proportional to the number of strains which share that gene. The insets (**b**–**d**) depict subgraphs corresponding to the shell partition inserted between conserved sequences

present in most genomes but not in all (Fig. 7b). Likewise, a group of prophage genes is present only in some genomes and inserted (alternative path) between conserved regions (Fig. 7c). Figure 7d shows a region where some genomes have lost genes (no alternative path is observed); this region corresponds to the second flagellar phase commonly lost in monophasic variants. For instance, the SMVET11 strain of our dataset was previously described as a monophasic *S.* Typhimurium [18].

9 Identification of Sequences of Interest in Genomic Data

9.1 Prediction of Antimicrobial-Resistance Genes and Virulence Factors

Identification of antimicrobial-resistance genes (ARGs) and virulence factors (VFs) in genomic data is standard in clinically associated bacteria. ARGs search in assembled genomes can be carried out using curated databases such as ResFinder [73], CARD [74], or NCBI-ARMFinderPlus [75], whereas VFs can be identified using the virulence factor database (VFDB) [76]. The ABRicate pipeline (https://github.com/tseemann/abricate) is frequently used to screen assemblies against the databases mentioned above. It runs

a BLAST/DIAMOND search in FASTA files (assemblies) with customizable identity/coverage cutoffs and also allows combining reports of different runs into a matrix of gene presence/absence.

We use the genomes in FASTA format downloaded from GenBank as input to run ABRicate with the ResFinder database as shown by the following commands.

Bash Shell

Run ABRicate with identity/coverage cutoff of 90/90% respectively.

```
$ abricate --db resfinder *.fasta --minid 90 --
mincov 90 > result_resfinder.tab
```

Combine reports into presence/absence matrix.

```
$ abricate --summary result_resfinder.tab > resfin-
der_summary.tab
```

ABRicate generates an antimicrobial-resistance gene presence/absence matrix with a present gene represented by its "% of coverage" and an absent gene denoted by a point (".") To visualize the gene presence/absence matrix in a heatmap alongside the phylogenetic tree we use the *gheatmap()* function of *ggtree* package in R with the following code.

R Script

```
library(ggnewscale)
# read the gene presence/absence matrix (resfinder_summary.tab)
df1 <- read.table("resfinder_summary.tab", sep = "\t", header = T,
check.names = F, row.names = 1)
# remove first row from the dataframe containing the number of ARGs
df2 <- df1[,-1]
# rename coverage values as 'P' if present and 'A' if absent
df2[df2 >= 90] <- 'Present'
df2[df2 < 90] <- 'Absent'
# start with p3 object generated in the section 7
p4 <- p3 + new_scale_fill()
# generate the gene presence/absence heatmap alongside the tree
p5 <- gheatmap(p4, df2, width = 1.2, font.size = 2.8,
      color ="black", colnames_offset_y = -0.4, colnames_position =
      "top", offset = 0.0012, hjust = 0, colnames_angle = 90) +
        scale_fill_manual(breaks = c("Present", "Absent"),
                          values = c("#6da3a3", "gray95"),
                          name = "Plasmid replicon") +
       theme(legend.position = "right")
p5 + ylim(NA,16)
```

In Fig. 6b we observe that the 14 genomes we are analyzing have different ARG content. Moreover, only one ARG (*aac(6')-Iaa*) is present in all strains. *aac(6')-Iaa* gene encodes an acetyltransferase that confers resistance to aminoglycoside drugs. It was acquired before *Salmonella enterica* serotype diversification, and is present in almost all isolates of the Typhimurium serotype [77]. Diverse ARGs were detected in humans and livestock-associated SeT strains (the columns of Fig. 6). Resistance to quinolone is typically found in *Salmonella* and is due to point mutations in DNA gyrase and topoisomerase IV genes. ABRicate only detects acquired resistance genes, so we recommend using PointFinder [78] to detect chromosomal mutations predictive of drug resistance.

9.2 Phage Sequence Prediction

Phage prediction tools are used to identify prophages and other viral sequences in assembled bacterial genomes. There are several different phage prediction tools available, with each using different algorithms and databases. Two of the most commonly used phage prediction tools are PHASTER [79] and VirSorter2 [36]. PHASTER runs on a web server and uses a combination of sequence similarity, gene prediction, and structural analysis to identify prophages in bacterial genomes. It also provides information on the location and orientation of the prophage, as well as the predicted functions of its genes. VirSorter2 is a standalone program that can be run locally and uses a machine-learning approach to classify viral sequences in genomic and metagenomic datasets. It can identify complete or partial phage genomes, as well as other types of viruses, and provides information on their taxonomy, gene content, and potential hosts.

To predict phage sequences for the genomes in our dataset, we loop VirSorter2 through all the genomes using the following command.

Bash Shell

```
$ for i in *.fasta; do virsorter run -w "${i%%.*}"_vir -i
"$i" --include-groups "dsDNAphage,ssDNA" -j 4 --min-score
0.8 --min-length 10000; done
```

VirSorter2 produces two output files: final-viral-boundary.tsv and final-viral-boundary.fasta. The first is a tab-separated table that contains the start and end base-pair positions of all predicted phage sequences in the genome and the other file contains the sequence of these phages in FASTA format.

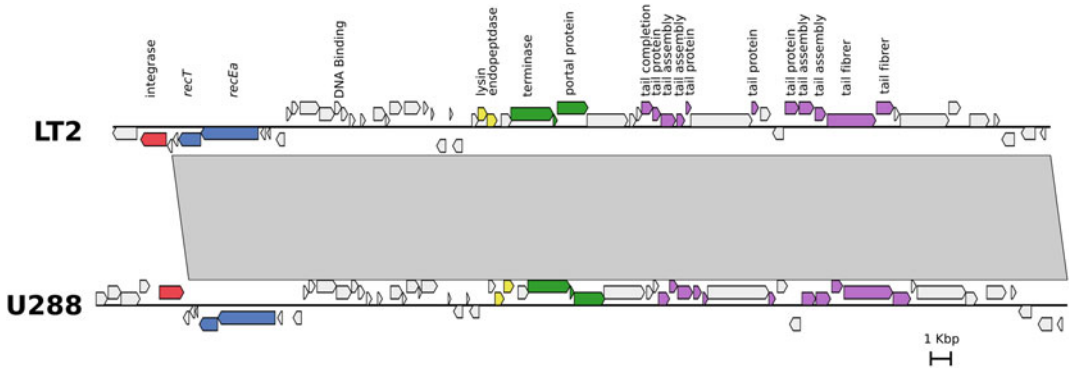

Fig. 8 Easyfig output image of the phage sequences of two *S.* Typhimurium isolates (LT2 and U288). Coding regions are shown as arrows. Selected open reading frames are colored in relation to their functions. The percentage of sequence similarity is indicated by the intensity of the gray color

We can compare phage sequences and evaluate their conservation and synteny plotting comparison figures of multiple genomes. To achieve this, we can use Easyfig [20], an application written in Python with an easy-to-use graphical user interface used for creating linear comparison figures of multiple genomic loci from annotation files (e.g., GenBank).

The gbk annotation file of LT2 and U288 genomes generated by Prokka in Subheading 5 was uploaded to Easyfig to generate a comparative plot of phage sequences. In Easyfig, set the start and end positions of the phage predicted by VirSorter2. Then, generate a BLAST comparison file by clicking on "generate blastn files." Finally, "Create Figure" to generate the image that can be exported in SVG format as shown in Fig. 8.

In Fig. 8, we observe sequence conservation and synteny in most of the phage sequences of two SeT genomes. However, distinct integrase genes were present in the genomes. Previous works have revealed that phages are responsible for the main genetic content variation in *Salmonella* genomes [17, 80].

10 Conclusion

In this chapter, we used freely available software to compare 14 SeT genomes and described the characteristics of the pangenome, explored the genetic content variation, and performed a phylogenomic analysis, starting from genomes downloaded from GenBank. Although in our example we used a small dataset for illustrative purposes, it revealed an open pangenome for SeT of size 5978 genes and a core genome with 4210 genes. We identified important variations in terms of gene content, including differential presence of virulence genes between isolates, gene content distribution

according to phylogroups, and high synteny conservation. Additionally, we predicted ARGs in livestock- and human-associated isolates and annotated phage sequences. The programs and codes presented can be used by the reader interested in carrying out a comparative analysis of genomes from any bacterial species.

Acknowledgments

This work was supported in part by a CNPq senior researcher fellowship to JCS.

References

1. Fleischmann RD, Adams MD, White O et al (1995) Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science 269:496–512. https://doi.org/10.1126/science.7542800
2. Mushegian AR, Koonin EV (1996) A minimal gene set for cellular life derived by comparison of complete bacterial genomes. Proc Natl Acad Sci USA 93:10268–10273
3. Welch RA, Burland V, Plunkett G et al (2002) Extensive mosaic structure revealed by the complete genome sequence of uropathogenic Escherichia coli. Proc Natl Acad Sci 99:17020–17024. https://doi.org/10.1073/pnas.252529799
4. Arnold BJ, Huang I-T, Hanage WP (2022) Horizontal gene transfer and adaptive evolution in bacteria. Nat Rev Microbiol 20:206–218. https://doi.org/10.1038/s41579-021-00650-4
5. Kim Y, Gu C, Kim HU, Lee SY (2020) Current status of pan-genome analysis for pathogenic bacteria. Curr Opin Biotechnol 63:54–62. https://doi.org/10.1016/j.copbio.2019.12.001
6. Ruan Z, Yu Y, Feng Y (2020) The global dissemination of bacterial infections necessitates the study of reverse genomic epidemiology. Brief Bioinform 21:741–750. https://doi.org/10.1093/bib/bbz010
7. Hurtado R, Carhuaricra D, Soares S et al (2018) Pan-genomic approach shows insight of genetic divergence and pathogenic-adaptation of *Pasteurella multocida*. Gene 670:193–206. https://doi.org/10.1016/j.gene.2018.05.084
8. Mageiros L, Méric G, Bayliss SC et al (2021) Genome evolution and the emergence of pathogenicity in avian Escherichia coli. Nat Commun 12:765. https://doi.org/10.1038/s41467-021-20988-w
9. The CRyPTIC Consortium (2022) Genome-wide association studies of global Mycobacterium tuberculosis resistance to 13 antimicrobials in 10,228 genomes identify new resistance mechanisms. PLoS Biol 20:e3001755. https://doi.org/10.1371/journal.pbio.3001755
10. Seib KL, Zhao X, Rappuoli R (2012) Developing vaccines in the era of genomics: a decade of reverse vaccinology. Clin Microbiol Infect 18:109–116. https://doi.org/10.1111/j.1469-0691.2012.03939.x
11. Doron S, Melamed S, Ofir G et al (2018) Systematic discovery of antiphage defense systems in the microbial pangenome. Science 359:eaar4120. https://doi.org/10.1126/science.aar4120
12. Benson DA, Cavanaugh M, Clark K et al (2018) GenBank. Nucleic Acids Res 46:D41–D47. https://doi.org/10.1093/nar/gkx1094
13. Jolley KA, Bray JE, Maiden MCJ (2018) Open-access bacterial population genomics: BIGSdb software, the PubMLST.org website and their applications. Wellcome Open Res 3:124. https://doi.org/10.12688/wellcomeopenres.14826.1
14. Markowitz VM, Chen I-MA, Palaniappan K et al (2012) IMG: the Integrated Microbial Genomes database and comparative analysis system. Nucleic Acids Res 40:D115–D122. https://doi.org/10.1093/nar/gkr1044
15. Olson RD, Assaf R, Brettin T et al (2023) Introducing the Bacterial and Viral Bioinformatics Resource Center (BV-BRC): a resource combining PATRIC, IRD and ViPR. Nucleic Acids Res 51:D678–D689. https://doi.org/10.1093/nar/gkac1003

16. Zhou Z, Alikhan N-F, Mohamed K et al (2020) The EnteroBase user's guide, with case studies on *Salmonella* transmissions, *Yersinia pestis* phylogeny, and *Escherichia* core genomic diversity. Genome Res 30:138–152. https://doi.org/10.1101/gr.251678.119
17. Bawn M, Alikhan N-F, Thilliez G et al (2020) Evolution of *Salmonella enterica* serotype Typhimurium driven by anthropogenic selection and niche adaptation. PLoS Genet 16: e1008850. https://doi.org/10.1371/journal.pgen.1008850
18. Carhuaricra Huaman DE, Luna Espinoza LR, Rodríguez Cueva CL et al (2022) Genomic characterization of *Salmonella* Typhimurium isolated from Guinea Pigs with Salmonellosis in Lima, Peru. Microorganisms 10:1726. https://doi.org/10.3390/microorganisms10091726
19. Seemann T (2023) ABRicate
20. Sullivan MJ, Petty NK, Beatson SA (2011) Easyfig: a genome comparison visualizer. Bioinformatics 27:1009–1010. https://doi.org/10.1093/bioinformatics/btr039
21. Cantalapiedra CP, Hernández-Plaza A, Letunic I et al (2021) eggNOG-mapper v2: functional annotation, orthology assignments, and domain prediction at the metagenomic scale. Mol Biol Evol 38:5825–5829. https://doi.org/10.1093/molbev/msab293
22. Bastian M, Heymann S, Jacomy M (2009) Gephi: an open source software for exploring and manipulating networks. Proc Int AAAI Conf Web Soc Media 3:361–362. https://doi.org/10.1609/icwsm.v3i1.13937
23. Wickham H (2016) ggplot2: elegant graphics for data analysis, 1st edn. Springer, New York
24. Yu G, Smith DK, Zhu H et al (2017) ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. Methods Ecol Evol 8: 28–36. https://doi.org/10.1111/2041-210X.12628
25. Croucher NJ, Page AJ, Connor TR et al (2015) Rapid phylogenetic analysis of large samples of recombinant bacterial whole genome sequences using Gubbins. Nucleic Acids Res 43:e15. https://doi.org/10.1093/nar/gku1196
26. Minh BQ, Schmidt HA, Chernomor O et al (2020) IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era. Mol Biol Evol 37:1530–1534. https://doi.org/10.1093/molbev/msaa015
27. Smyth G, Hu Y, Ritchie M, et al (2023) limma: linear models for microarray data
28. Snipen L, Liland KH (2015) micropan: an R-package for microbial pan-genomics. BMC Bioinform 16:79. https://doi.org/10.1186/s12859-015-0517-0
29. Tonkin-Hill G, MacAlasdair N, Ruis C et al (2020) Producing polished prokaryotic pangenomes with the Panaroo pipeline. Genome Biol 21:180. https://doi.org/10.1186/s13059-020-02090-4
30. Gautreau G, Bazin A, Gachet M et al (2020) PPanGGOLiN: depicting microbial diversity via a partitioned pangenome graph. PLoS Comput Biol 16:e1007732. https://doi.org/10.1371/journal.pcbi.1007732
31. Kolde R (2019) pheatmap: Pretty Heatmaps
32. Seemann T (2014) Prokka: rapid prokaryotic genome annotation. Bioinformatics 30:2068–2069. https://doi.org/10.1093/bioinformatics/btu153
33. R Core Team (2023) R: a language and environment for statistical computing
34. Brynildsrud O, Bohlin J, Scheffer L, Eldholm V (2016) Rapid scoring of genes in microbial pan-genome-wide association studies with Scoary. Genome Biol 17:238. https://doi.org/10.1186/s13059-016-1108-8
35. Page AJ, Taylor B, Delaney AJ et al (2016) SNP-sites: rapid efficient extraction of SNPs from multi-FASTA alignments. Microb Genom 2:e000056. https://doi.org/10.1099/mgen.0.000056
36. Guo J, Bolduc B, Zayed AA et al (2021) VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. Microbiome 9:37. https://doi.org/10.1186/s40168-020-00990-y
37. Salzberg SL (2019) Next-generation genome annotation: we still struggle to get it right. Genome Biol 20:92. https://doi.org/10.1186/s13059-019-1715-2
38. Hyatt D, Chen G-L, LoCascio PF et al (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinform 11:119. https://doi.org/10.1186/1471-2105-11-119
39. Schwengers O, Jelonek L, Dieckmann MA et al (2021) Bakta: rapid and standardized annotation of bacterial genomes via alignment-free sequence identification. Microb Genom 7. https://doi.org/10.1099/mgen.0.000685
40. Hernández-Plaza A, Szklarczyk D, Botas J et al (2023) eggNOG 6.0: enabling comparative genomics across 12 535 organisms. Nucleic Acids Res 51:D389–D394. https://doi.org/10.1093/nar/gkac1022

41. Kanehisa M, Goto S, Sato Y et al (2014) Data, information, knowledge and principle: back to metabolism in KEGG. Nucleic Acids Res 42: D199–D205. https://doi.org/10.1093/nar/gkt1076
42. Galperin MY, Wolf YI, Makarova KS et al (2021) COG database update: focus on microbial diversity, model organisms, and widespread pathogens. Nucleic Acids Res 49:D274–D281. https://doi.org/10.1093/nar/gkaa1018
43. Drula E, Garron M-L, Dogan S et al (2022) The carbohydrate-active enzyme database: functions and literature. Nucleic Acids Res 50: D571–D577. https://doi.org/10.1093/nar/gkab1045
44. The Gene Ontology Consortium (2019) The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Res 47:D330–D338. https://doi.org/10.1093/nar/gky1055
45. Tettelin H, Riley D, Cattuto C, Medini D (2008) Comparative genomics: the bacterial pan-genome. Curr Opin Microbiol 11:472–477. https://doi.org/10.1016/j.mib.2008.09.006
46. McInerney JO, McNally A, O'Connell MJ (2017) Why prokaryotes have pangenomes. Nat Microbiol 2:1–5. https://doi.org/10.1038/nmicrobiol.2017.40
47. Dereeper A, Summo M, Meyer DF (2022) PanExplorer: a web-based tool for exploratory analysis and visualization of bacterial pan-genomes. Bioinformatics 38(18): 4412–4414. https://doi.org/10.1093/bioinformatics/btac504
48. Altschul SF, Gish W, Miller W et al (1990) Basic local alignment search tool. J Mol Biol 215:403–410. https://doi.org/10.1016/S0022-2836(05)80360-2
49. Fu L, Niu B, Zhu Z et al (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 28:3150–3152. https://doi.org/10.1093/bioinformatics/bts565
50. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods 12:59–60. https://doi.org/10.1038/nmeth.3176
51. Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: identification of Ortholog Groups for eukaryotic genomes. Genome Res 13:2178–2189. https://doi.org/10.1101/gr.1224503
52. Contreras-Moreira B, Vinuesa P (2013) GET_HOMOLOGUES, a versatile software package for scalable and robust microbial Pan-genome analysis. Appl Environ Microbiol 79: 7696–7701. https://doi.org/10.1128/AEM.02411-13
53. Katoh K, Standley DM (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol 30:772–780. https://doi.org/10.1093/molbev/mst010
54. Colquhoun RM, Hall MB, Lima L et al (2021) Pandora: nucleotide-resolution bacterial pan-genomics with reference graphs. Genome Biol 22:267. https://doi.org/10.1186/s13059-021-02473-1
55. Alikhan N-F, Zhou Z, Sergeant MJ, Achtman M (2018) A genomic overview of the population structure of Salmonella. PLoS Genet 14: e1007261. https://doi.org/10.1371/journal.pgen.1007261
56. Park S-C, Lee K, Kim YO et al (2019) Large-scale genomics reveals the genetic characteristics of seven species and importance of phylogenetic distance for estimating pan-genome size. Front Microbiol 10:834
57. Jacobsen A, Hendriksen RS, Aaresturp FM et al (2011) The *Salmonella enterica* Pan-genome. Microb Ecol 62:487–504. https://doi.org/10.1007/s00248-011-9880-1
58. Kislyuk AO, Haegeman B, Bergman NH, Weitz JS (2011) Genomic fluidity: an integrative view of gene diversity within microbial populations. BMC Genomics 12:32. https://doi.org/10.1186/1471-2164-12-32
59. Domingo-Sananes MR, McInerney JO (2021) Mechanisms that shape microbial Pangenomes. Trends Microbiol 29:493–503. https://doi.org/10.1016/j.tim.2020.12.004
60. Matsui H, Bacot CM, Garlington WA et al (2001) Virulence plasmid-borne spvB and spvC genes can replace the 90-Kilobase plasmid in conferring virulence to *Salmonella enterica* Serovar Typhimurium in subcutaneously inoculated mice. J Bacteriol 183:4652–4658. https://doi.org/10.1128/JB.183.15.4652-4658.2001
61. Ma S, Dai Y (2011) Principal component analysis based methods in bioinformatics studies. Brief Bioinform 12:714–722. https://doi.org/10.1093/bib/bbq090
62. Allen JP, Snitkin E, Pincus NB, Hauser AR (2021) Forest and trees: exploring bacterial virulence with genome-wide association studies and machine learning. Trends Microbiol 29: 621–633. https://doi.org/10.1016/j.tim.2020.12.002
63. Didelot X (2021) Phylogenetic methods for genome-wide association studies in bacteria. Methods Mol Biol Clifton NJ 2242:205–220.

https://doi.org/10.1007/978-1-0716-1099-2_13

64. Coll F, Gouliouris T, Bruchmann S et al (2022) PowerBacGWAS: a computational pipeline to perform power calculations for bacterial genome-wide association studies. Commun Biol 5:1–12. https://doi.org/10.1038/s42003-022-03194-2
65. Patané JSL, Martins J, Setubal JC (2018) Phylogenomics. In: Setubal JC, Stoye J, Stadler PF (eds) Comparative genomics: methods and protocols. Springer, New York, pp 103–187
66. Stamatakis A (2014) RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics 30:1312–1313. https://doi.org/10.1093/bioinformatics/btu033
67. Price MN, Dehal PS, Arkin AP (2010) FastTree 2 – approximately maximum-likelihood trees for large alignments. PLoS One 5:e9490. https://doi.org/10.1371/journal.pone.0009490
68. Didelot X, Wilson DJ (2015) ClonalFrameML: efficient inference of recombination in whole bacterial genomes. PLoS Comput Biol 11:e1004041. https://doi.org/10.1371/journal.pcbi.1004041
69. Branchu P, Bawn M, Kingsley RA (2018) Genome variation and molecular epidemiology of *Salmonella enterica* Serovar Typhimurium Pathovariants. Infect Immun 86:e00079–e00018. https://doi.org/10.1128/IAI.00079-18
70. Page AJ, Cummins CA, Hunt M et al (2015) Roary: rapid large-scale prokaryote pan genome analysis. Bioinforma Oxf Engl 31:3691–3693. https://doi.org/10.1093/bioinformatics/btv421
71. Guarracino A, Heumos S, Nahnsen S et al (2022) ODGI: understanding pangenome graphs. Bioinformatics 38:3319–3326. https://doi.org/10.1093/bioinformatics/btac308
72. Noll N, Molari M, Neher RA (2022) PanGraph: scalable bacterial pan-genome graph construction. 2022.02.24.481757
73. Florensa AF, Kaas RS, Clausen PTLC et al (2022) ResFinder - an open online resource for identification of antimicrobial resistance genes in next-generation sequencing data and prediction of phenotypes from genotypes. Microb Genomics 8. https://doi.org/10.1099/mgen.0.000748
74. Alcock BP, Raphenya AR, Lau TTY et al (2020) CARD 2020: antibiotic resistome surveillance with the comprehensive antibiotic resistance database. Nucleic Acids Res 48:D517–D525. https://doi.org/10.1093/nar/gkz935
75. Feldgarden M, Brover V, Gonzalez-Escalona N et al (2021) AMRFinderPlus and the Reference Gene Catalog facilitate examination of the genomic links among antimicrobial resistance, stress response, and virulence. Sci Rep 11:12728. https://doi.org/10.1038/s41598-021-91456-0
76. Liu B, Zheng D, Jin Q et al (2019) VFDB 2019: a comparative pathogenomic platform with an interactive web interface. Nucleic Acids Res 47:D687–D692. https://doi.org/10.1093/nar/gky1080
77. Liao J, Orsi RH, Carroll LM et al (2019) Serotype-specific evolutionary patterns of antimicrobial-resistant *Salmonella enterica*. BMC Evol Biol 19:132. https://doi.org/10.1186/s12862-019-1457-5
78. Zankari E, Allesøe R, Joensen KG et al (2017) PointFinder: a novel web tool for WGS-based detection of antimicrobial resistance associated with chromosomal point mutations in bacterial pathogens. J Antimicrob Chemother 72:2764–2768. https://doi.org/10.1093/jac/dkx217
79. Arndt D, Grant JR, Marcu A et al (2016) PHASTER: a better, faster version of the PHAST phage search tool. Nucleic Acids Res 44:W16–W21. https://doi.org/10.1093/nar/gkw387
80. Mottawea W, Duceppe M-O, Dupras AA et al (2018) *Salmonella enterica* prophage sequence profiles reflect genome diversity and can be used for high discrimination subtyping. Front Microbiol 9:836. https://doi.org/10.3389/fmicb.2018.00836