E.coli outbreak investigation

Introduction. Tracing the cause of the outbreak.

In April 2011, hundreds of people in Germany were hospitalized with hemolytic uremic syndrome (HUS), a deadly blood disease that often starts as food poisoning with bloody diarrhea and can lead to kidney failure. German health officials immediately informed the World Health Organization about the incident, but they had no idea that it was the beginning of the deadliest *E. coli* outbreak in recent history, caused by a mysterious strain that we will call *E. coli* X. Within a few months, the bacterium had infected thousands and killed 53. But where did it come from?

Researchers initially suspected contaminated food as the source of the *E. coli* outbreak, but they struggled to pin it down to a specific product and location. Shortly after the outbreak started, German authorities found traces of the bacterium in cucumbers imported from Spain. The suspicious vegetables were destroyed, and many Europeans stopped eating cucumbers entirely. A month later, the European Commission backtracked, announcing that cucumbers had nothing to do with the outbreak. (A German court would later order Spanish cucumber growers to be compensated for their financial losses.)

German health officials then linked the outbreak to a restaurant in Lübeck where nearly 20% of the patrons had developed bloody diarrhea in a single week. By analyzing the meals eaten by the guests, researchers found that patrons who had eaten bean sprouts (Figure below) were much more likely to have contracted HUS.

Bean sprouts.

Shortly afterwards, eight people were hospitalized with HUS after eating sprouts in France, and researchers found that the same *E. coli* X strain was to blame. Scientists later tracked the source to a single lot of fenugreek sprout seeds imported from Egypt that had been sold two years previously to distributors in Germany and France. Europe then banned imported fenugreek seeds from Egypt, but Egyptian officials argued that *E. coli* could not have survived for two years on seeds and that handling by the distributor could instead have resulted in sprout contamination. However, biologists know that *E. coli* can survive for years on seeds and still retain its pathogenicity.

The 2011 German outbreak was not the first *E. coli* epidemic to be linked to sprouts, which are usual suspects in *E. coli* epidemics because they are cultivated in humid, bacteria-rich conditions. In fact, *E. coli*-tainted sprouts afflicted nearly ten thousand schoolchildren in Japan in 1996. Yet as we will see, the reaction to the 2011 outbreak was much swifter, thanks to genome sequencing methods and bioinformatics algorithms.

Crowdsourcing bioinformatics analysis of the pathogenic strain

In May 2011, a man and his two children were admitted to an emergency room in Hamburg with bloody diarrhea, having all eaten a salad containing sprouts a week earlier. Doctors first suspected that the family had been infected with the common pathogenic *E. coli* strain O157:H7. This strain causes tens of thousands of hospitalizations annually and often leads to HUS.

In contrast with harmless *E. coli* strains that can easily metabolize sorbitol (which is commonly used as a sugar substitute), some pathogenic *E. coli* strains, including *E. coli* O157:H7 , have difficulty metabolizing sorbitol. Thus, there is a simple test to check for the presence of these strains; when pathogenic *E. coli* grows in a sorbitol-rich medium, its colonies often have a characteristic color, indicating that sorbitol is not being fully fermented. However, all three family samples did not pass the sorbitol test or any other biochemical tests for known HUS-causing *E. coli* strains. At this point, biologists knew that they were facing a previously unknown pathogen, and that traditional methods would not suffice – bioinformatics approaches would be needed.

To investigate the evolutionary origins and pathogenic potential of the outbreak strain, researchers started a crowdsourced research program. They publicized sequencing data of the isolate from the girl in Hamburg, called sample TY2482. A burst of analyses followed, carried out by bioinformaticians on four continents. The researchers even used GitHub for the project: https://github.com/ehec-outbreak-crowdsourced/BGI-data-analysis/wiki

After the genome of the pathogenic strain had been identified, no further outbreak clusters were identified. The 2011 German outbreak, like the 2003 SARS outbreak, presents an early

example of epidemiologists collaborating directly with genomics and bioinformatics experts to stop an epidemic.

In this project, you will follow in the footsteps of the bioinformaticians investigating the outbreak by assembling the genome of the deadly *E. coli* X strain. Specifically, we will provide you with Illumina reads from the TY2482 sample, which were generated at Beijing Genome Institute and deposited into the Short Read Archive (http://www.ncbi.nlm.nih.gov/sra) for public access. After we have assembled the *E. coli* X genome, we will determine which other strain(s) it is most similar to in an effort to determine how it could have arisen. In particular, once we know where it came from, we can ask what new genes it possesses for pathogenicity, and how it evolved these new functions.

Our workflow will be to answer the following questions:

1. What is the genome sequence of *E. coli* X?
2. What strain of *E. coli* is *E. coli* X most similar to? (Where did it come from?)
3. What are the genes that *E. coli* X contains?
4. Which of these genes make *E. coli* X distinct?
5. How did *E. coli* X evolve to obtain these genes?
6. How did *E. coli* X become pathogenic?

But before you embark on analyzing the *E. coli* X strain, we will take the time to explain how a harmless bacterium can become pathogenic in the first place.

How can a bacteria become pathogenic?

There are more than 700 known infectious subspecies, or strains, of *E. coli*. Pathogenicity is determined by virulence factors, various compounds that it produces to facilitate colonizing the host and evading or inhibiting the host's immune system.

Bacteria possess a wide array of virulence factors that may be encoded either in the bacterial chromosome or in extrachromosomal genetic elements called plasmids, independent circular mini-chromosomes that co-exist with the bacterial genome. For example, many *E. coli* strains have plasmids containing genes that have been implicated in antibiotic resistance.

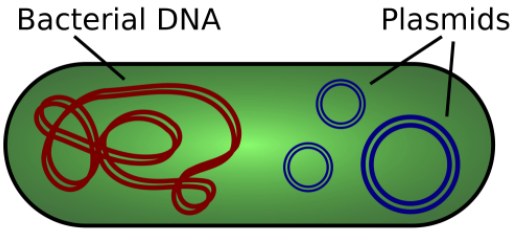


Figure: Plasmids in a bacterial cell.

Many pathogenic bacteria secrete toxins, substances that may inhibit cellular functions in the host. Examples include the tetanus toxin secreted by *Clostridium tetani*, the botulinum toxin secreted by *Clostridium botulinum*, and the anthrax toxin produced by *Bacillus anthracis*. Several strains of *E. coli* and *Shigella* secret Shiga toxins, which are encoded by Stx genes. Shiga toxins are named for Kiyoshi Shiga, who described the bacterial origin of a dysentery outbreak in Japan in 1897 during which nearly 30,000 people died. These toxins cause bleeding by breaking down the lining of the colon, and they can lead to HUS if they reach the kidneys. Shiga toxins attack highly specific receptors on the surface of human cells, and so species that do not have this receptor, such as cows, may harbor toxigenic bacteria without any ill effects.

Another type of pathogenic agents are phages, viruses that cannot replicate on their own and must infect bacteria to do so. Many phages are shaped like lunar landers (Figure below), a design that helps them land on the cell wall of a bacterium and transmit their own genome (called a prophage) into the bacterial genome, so that when the bacterial DNA replicates, it creates new copies of the phage as well.

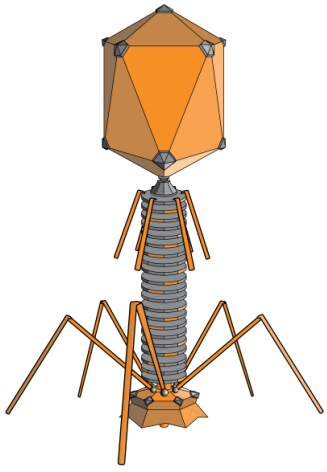


Figure: The structure of a T4 bacteriophage.

Sometimes, the prophage is replicated as a plasmid; in other cases, prophage genes encode recombinases, enzymes that can catalyze DNA exchange reactions between short (30–200 nucleotides) similar sequences. During this process, called site-specific recombination, prophage literally "glued" into host DNA.

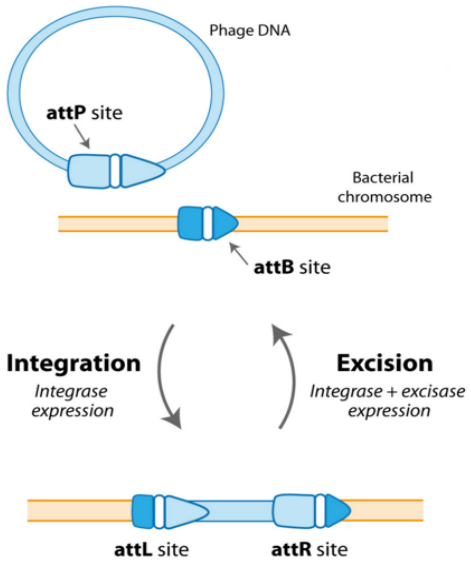


Figure: Phage DNA integration in a bacterial chromosome via recombination between the phage attP and the chromosomal attB recombination sites. Source: Memo-Cell project.

Phages offer an example of a phenomenon that was first witnessed in 1951 when Victor Freeman transferred a viral gene into the bacterium *C. diphtheriae* and transformed an otherwise non-virulent strain into a virulent one. Such horizontal genetic transfer (HGT), or the transfer of genes between two organisms, is in contrast to the usual vertical gene transfer from a cell to its daughter cells during division. For that matter, phages do not always harm the host; by inserting their DNA into the host bacterium, they may actually provide the host with benefits by equipping its genome with new functions. For example, phages can transform harmless strains of *Vibrio cholerae* into the virulent ones that cause cholera. Furthermore, HGT is not limited to transfer between organisms of different types, as bacteria exchange mobile elements such as phages and plasmids to neighboring cells.

Sometimes, genes have been transferred horizontally and become a part of the host's DNA, being passed down to offspring. For example, the cellular organelle called the mitochondrion has its own DNA, and biologists have proposed that the mitochondrion is the descendant of a group of bacteria called alphaproteobacteria, which were engulfed long

ago by eukaryotic cell but not digested. Because sperm cells do not pass mitochondria to zygote, you inherited your “mitochondrial genome” from your mother, who inherited it from her mother, and so on back to the dawn of civilization. A similar example of horizontal gene transfer in plants is the chloroplast, which is a descendant of a cyanobacterium, acquired long ago by a plant ancestor to facilitate photosynthesis.

Assembling the Genome

DNA sequencing technology cannot read whole genomes in one go, so we will assemble genome from the short reads generated by the Illumina instrument. We will use the SPAdes assembler (Bankevich et al., 2012) to assemble the *E. coli* X genome from reads into contigs.

A contig is a “contiguous” segment of the genome that has been reconstructed by an assembly algorithm. In addition to contigs, SPAdes uses information about the distances between reads within read-pairs (called insert size) to combine contigs into ordered collections of adjacent contigs called scaffolds. For example, as illustrated in the figure below, if one read in a read-pair appears in Contig 1, and the other appears in Contig 2, then we may infer that Contig 1 and Contig 2 are neighbors in the same scaffold. Genome assembly programs often fill a gap in a scaffold of length m by a sequence of m occurrences of “N” (a placeholder for unknown nucleotides)

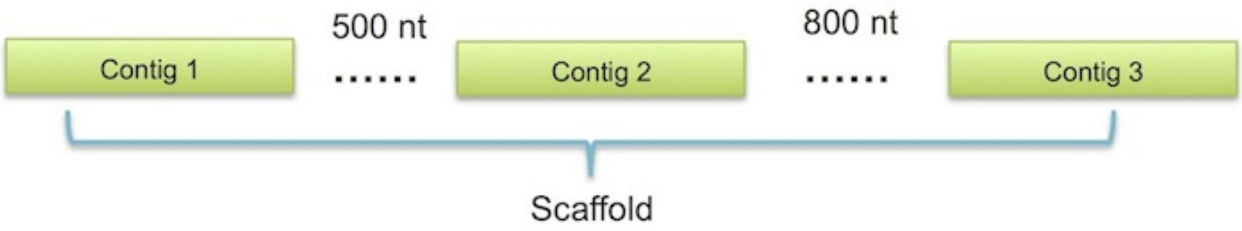


Figure: Three contigs combined into a single scaffold separated by two gaps of length 500 nt and 800 nt.

In modern DNA sequencing projects, DNA fragments are sequenced from both the 5' and 3' ends, giving rise to paired reads separated by some insert size. The forward and reverse read in a paired read are generated from the forward and reverse strand, respectively. A collection of paired reads is called a sequencing library; whereas most paired libraries generate reads with insert sizes below 1 kb, libraries with longer (2 kb-10 kb) insert sizes are called mate-pairs.

Many sequencing projects generate several libraries of paired reads with different insert sizes; for example, the sequencing project at the center of this task employs three libraries of insert lengths 270, 2000, and 6000 nucleotides.

The advantage of using multiple libraries is that libraries with small insert sizes are better suited for resolving short repeats, whereas libraries with larger insert sizes are better suited for resolving long repeats. For example, all bacterial genomes contain ribosomal operons, which are often repeated more than six times and are approximately 5000 nucleotides long. Thus, to resolve ribosomal operons, we need mate-pair libraries with insert sizes exceeding 5000 nt.

Once we have sequenced a genome, there are many metrics for assessing the resulting assembly. We will use QUAST (Gurevich et al., 2013), which takes many metrics into account to evaluate the overall quality of an assembly.

One of the common metrics is the N50 statistic. N50 is defined as the maximal contig length for which all contigs greater than or equal to that length comprise at least half of the sum of the lengths of all the contigs. For example, consider the five toy contigs with the following lengths: [10, 20, 30, 60, 70]. Here, the total length of contigs is 190, and contigs of length 60 and 70 account for at least 50% of the total length of contigs (60 + 70 = 130), but the contig of length 70 does not account for 50% of the total length of contigs. Thus, N50 is equal to 60.

Project description

1. Exploring the dataset

For this project, we provide three libraries from the TY2482 sample with the following insert sizes and orientation:

- SRR292678 - paired end, insert size 470 bp (forward reads, reverse reads, 400 Mb each)
- SRR292862 – mate pair, insert size 2 kb, (forward reads, reverse reads, 200 Mb each)
- SRR292770 – mate pair, insert size 6 kb, (forward reads, reverse reads, 200 Mb each)

Run FastQC on all 6 of the 3 fastq files. Save the number of reads for your report.

For assembly we will be using a de Bruijn graph strategy, which breaks the reads up into k-mers to facilitate assembly of correctly connected contigs. The size and type of these k-mers are important parameters in de novo assembly, so it may be useful to spend some time looking at how k-mers are distributed in our data.

2. (optional, 1 extra credit). K-mer profile and genome size estimation

For our purposes it's OK to only analyze the paired-end library (SRR292678). *To count kmers, you can use the Jellyfish - fast kmer counting program that will count the frequency of all possible k-mers of a given length in our data. It is available via conda (recommended) or with apt-get.*

The "jellyfish count" command takes the following options:

- *-m or "mer" specifies the length*
- *-C tells it to ignore directionality (it treats each read the same as its reverse complement).*
- *-s is an initial estimate for the size of the hash table jellyfish uses, set > genome size*
- *-o specifies the name of the output file. choose a name with the k-mer length in it.*

You can use k-mer sizes of 31, count kmers with `jellyfish count` *and make a histogram file with* `jellyfish histo`.

Take a look at the histo files - on the left is a list of the bins (the number of times a k-mer occurs or its 'depth'), and on the right is the count for the number of k-mers in the data that fit into that category.

The k-mer distribution is easier to understand if we actually visualize it. You can do it in any of your favorite data analysis or spreadsheet packages (~~Excel~~, R, Calc).

Knowing the k-mer distribution, you can estimate the genome size. Knowing the number of reads and average length, you can estimate the total number of bases in all the reads. Use the peak bin you identified, and total number of bases, to estimate the genome size of our bacteria with the following formulas:

$N = (M*L)/(L-K+1)$

$Genome\_size = T/N$

(N: *Depth of coverage*, M: *Kmer peak*, K: *Kmer-size*, L: *avg read length* T: *Total bases*)

3. Assembling E. coli X genome from paired reads

This week for the read correction and assembly we will use assembler SPAdes. Youc an install it via conda (`conda/mamba install spades -c bioconda`)

As an alternative, you can get the binaries for the last version from the corresponding web site. Choose a proper distribution – for Linux or Mac OS (64-bit only). Download and unpack the archive (to unpack a .tar.gz archive, you can use command `tar -xzf filename`).

To verify that the software was installed correctly, run it in the test mode, using the only parameter `--test`. It will assemble a toy dataset, and if you do everything right, the last line of the output will be "Thank you for using SPAdes!"

*Important note: Genome assembly can be a **very** compute-intensive task, raw read files are large, and many of the programs we will encounter take a long time to run on real datasets -- as much as ten hours for some programs.*

We will first try to assemble a single library of sequencing (paired end) reads from *E. coli* X. Let's try to run SPAdes in the paired-end mode, providing paired reads of *E. coli* X. from the library SRR292678 (forward and reverse). SRR292678 here is the ID of the sequencing library.

SPAdes can take up to 40 minutes to finish running, depending on CPU speed and memory. If you face a problem due to lack of computational resources, you can download precomputed results here.

Now that SPAdes has finished running, assess the quality of the resulting assembly. For the subsequent analysis we will be interested only in the “contigs.fasta” and “scaffolds.fasta” files, which define contigs and scaffolds for the *E. coli* assembly, respectively.

You can use the online version of QUAST (*sorry, right now the server is down, power outage*) or the console tool (again, via conda or download from sourceforge).

Consider the “contigs” column in the QUAST report. Note that all statistics are based on contigs of size ≥ 500 bp, unless otherwise noted (the statistics "# contigs (≥ 0 bp)" and "Total length (≥ 0 bp)" refer to all contigs). Biologists are mainly interested in long contigs because short contigs often contain only gene fragments.

3a (Optional, 1 extra credit). Effect of read correction.

SPAdes works in two-step mode: error correction and assembly. It contains corrected reads in the “corrected” folder. Repeat the k-mer profile plotting step and compare with the one for uncorrected reads. Add both plots in your report and explain the difference.

4. Impact of reads with large insert size

As mentioned, there is an advantage in using multiple libraries with different insert sizes, since the library with a small insert size can resolve short repeats, whereas the library with a larger insert size can resolve longer repeats. In particular, we are interested in how the quality of our assembly changes when we add mate-pair libraries.

This time, we will run SPAdes again by consolidating three libraries. Run SPAdes providing all three libraries: SRR292678 as a paired ends, SRR292862 and SRR292770 as a mate pairs.

If you face problems due to lack of computational resources, you can download precomputed results here.

Save the QUAST report data in your lab journal. In your report you should provide the main metrics (N50 and number of contigs) for single-library and three-library assemblies. Also answer, how did the quality of the assembly improve compared to the previous run of SPAdes, and why.

5. Genome Annotation

After assembling a genome, biologists annotate the genome, searching for genes and other important regions. After identifying putative genes, biologists perform functional gene annotation to determine their functions. The "comparative genomics" approach to gene annotation is based on the (not always ideal) assumption that similar genes in different organisms perform similar functions. Finding similar genes can be accomplished by running BLAST or by matching putative proteins against protein domains in the Pfam database.

Now that we have sequenced and assembled the *E. coli* X genome, we will use Prokka for gene prediction and annotation. This tool identifies the coordinates of putative genes within contigs and then uses BLAST for similarity-based annotation using all proteins from sequenced bacterial genomes in the RefSeq database.

Install Prokka on your local computer (I recommend conda/mamba-based approach, but you may consider other ways described there). Run it on the "scaffolds.fasta" file from the SPAdes output with default parameters.

Important note from PROKKA's FAQ:

- *Why can't I load Prokka .GBK files into Mauve?*

Mauve uses BioJava to parse GenBank files, and it is very picky about Genbank files. It does not like long contig names, like those from Velvet or Spades. One solution is to use --centre XXX in Prokka and it will rename all your contigs to be NCBI (and Mauve) compliant

Annotation can take up to 20 minutes. After it has completed, select "scaffolds.gbk" from the output folder and store it on your computer, as we will use it later to compare *E. coli* X to a similar bacterium.

If you face problems due to lack of computational resources, you can download results here: PROKKA.zip

6. Finding the closest relative of *E. coli* X

Our goal is to find the known genome that is the most similar to the pathogenic strain (and infer properties of *E. coli* X from it). We could compare each contig in our assembly against the entire RefSeq database using BLAST, but this could take several hours depending on server workload. A more efficient approach is to select one important and evolutionarily conserved gene for comparison with all other sequenced genomes. The gene that we will use is 16S ribosomal RNA.

First, we need to locate 16S rRNA in the assembled *E. coli* X genome. You can use the rRNA genes prediction tool Barrnap. Find the 16S rRNA in the results of the most recent run of SPAdes.

rRNA genes in bacteria are typically organized in ribosomal operons – set of closely located genes that are activated together. Ribosomal RNA plays a crucial role in protein synthesis, and in order to achieve high growth rate bacteria often possess several copies of this operon. That is why you will probably get several matches here.

We will now use BLAST to search for the genome in the RefSeq database with 16S rRNA that is most similar to the 16S rRNA that we just found. Open the NCBI BLAST homepage (http://blast.ncbi.nlm.nih.gov) and select "Nucleotide blast". To perform the search against complete genomes in the RefSeq database, select the "Reference Genome Database (refseq_genomes)" in the "Database" field, and *Escherichia coli* in the "Organism" field.

To restrict our search to only those genomes that were present in the GenBank database at the beginning of 2011, set the time range using parameter PDAT in the "Entrez Query" field:

```
1900/01/01:2011/01/01[PDAT]
```

Other parameters should be specified as default.

When the search finishes after a few minutes, explore its results – probably the closest relative to the *E. coli* X that can be used as a reference genome.

Click on the “Sequence ID” link under the name of the identified reference in order to open its corresponding GenBank page. Download the genome sequence in FASTA format (in the right upper corner select ”Send” - “Complete Record” - “File” – “Fasta”, and save as “55989.fasta”)

Include the name and GenBank accession number of the reference E.coli strain in the lab report.

Comparing other regions of the *E. coli* X genome against the entire RefSeq database would tell us that this reference is indeed its closest relative, but that *E. coli* X is nevertheless a distinct strain, one whose genome in early 2011 was unknown.

The in-depth study of E. coli X began after the start of the outbreak; after you complete this challenge, we encourage you to perform a search with assembled contigs, removing the restriction in the "Entrez Query" field, and explore the results on your own.

7. What is the genetic cause of HUS?

Now that we have identified the closest relative of the strain that caused the outbreak, it makes sense to examine the original paper about this reference strain (Mossoro et al., 2002). This paper states that our reference belongs to an enteroaggregative *E. coli* (EAEC) strain, harboring the pAA plasmid, which contains aggregative adherence fimbria (AAF) genes allowing bacteria to stick to cells in the intestine (see FAQ: Classification of pathogenic *E. coli* strains).

The reference strain was isolated in the Central African Republic from a stool sample obtained from a man with persistent diarrhea. However, in contrast to patients infected in the 2011 outbreak, this patient had no signs of blood in his stool! This fact suggests that *E. coli* X somehow gained an additional virulence factor over the previously identified reference strain. But what is the virulence factor, and how did *E. coli* X obtain it?

To understand the genetic cause of HUS, we will perform a genome-wide comparison with the reference genome and will analyze the regions where these strains differ from each other. If we find a region where *E. coli* X encodes a new virulence factor or a new gene responsible for antibiotic resistance, it may shed light on the genetic cause of HUS.

We will use a program called Mauve, which visualizes an alignment as a series of conserved segments called **Locally Collinear Blocks (LCBs)**, which are similar to synteny blocks. Insertions and deletions in LCBs correspond to insertions and deletions in a bacterial chromosome. Separate unaligned regions that have no flanking regions from chromosomal DNA, on the other hand, may correspond to extrachromosomal elements such as plasmids. LCBs are indicated using differently colored bars, and you can navigate and zoom using the toolbar at the top of the screen. The screenshot in the figure below shows an example of insertion inside a green LCB of the assembled genome.

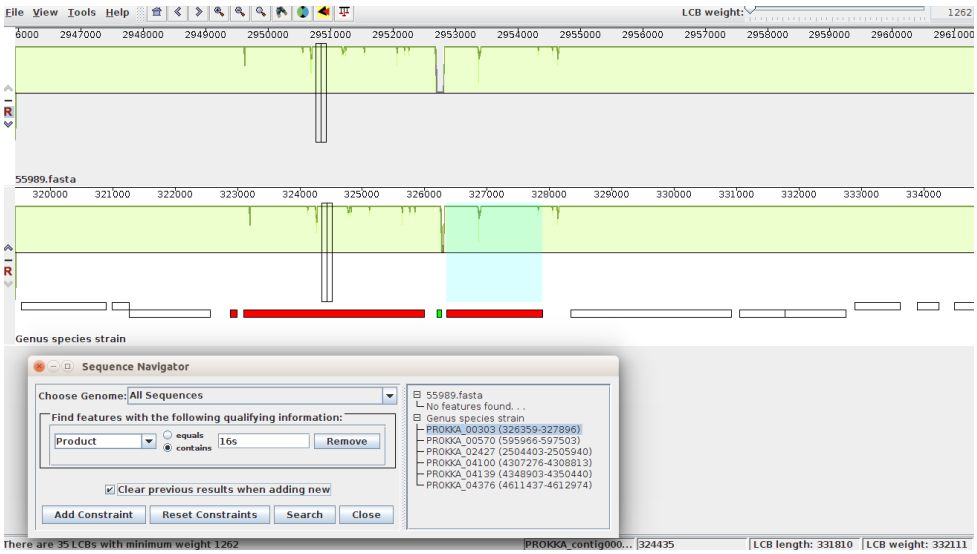
Since bacterial genomes undergo rearrangements, various regions can be rearranged, even between two closely related bacterial species. Collinear blocks allow us to visualize these rearrangements with ease.

To compare the *E. coli* X with the reference genome, first install Mauve (http://darlinglab.org/mauve/download.html) on your computer. Open "Mauve" and select "File" → "Align with progressiveMauve...". Press "Add sequences" and select the reference genome, then the annotated *E. coli* X genome ("scaffolds.gbk" or "scaffolds.gbf", depending on version), and start the alignment.

(Note: Mauve is very picky about folder names. Your paths and filenames should contain English symbols only)

We will now analyze the genome-wide alignment of *E. coli* X and the reference genome. We are interested in unaligned regions in the scaffolds as putative insertions between reference and *E. coli* X to figure out how *E. coli* X acquired the additional virulence factor of causing bleeding. In particular, we will check whether *E. coli* X encodes the Shiga toxins produced by other *E. coli* strains that cause internal bleeding.

You can look for specific genes using Sequence Navigator in Mauve (the icon of the binoculars in the upper line). Select "Product" in the left window and enter the name of the desired gene (or its function) in the right window.



Here is an example of Mauve alignment

Find shiga toxin-related genes and write down its names, length and location in your report.

8. Tracing the source of toxin genes in E. coli X

We have discovered that the E. coli X genome encodes Shiga-like toxin genes. Now let's figure out how this strain has acquired these weapons. Now you can explore the whole segment containing these genes. Zoom out and move your mouse cursor over the genes nearby to see annotation. Based on this annotation, what is the origin of these toxin genes in E.coli X?

If your see a lot of "hypothetical proteins", maybe there is no mobile-element related proteins in your version of Prokka databases - in this case you can check the regions of interest in this .gbk file - I've annotated it on http://rast.nmpdr.org/, it takes time, depends on the server load. They provide slightly different implementations of the .gbk format, and all the interesting information is stored in the "Product" field.

9. Antibiotic resistance detection

During the German outbreak of E. coli X, some patients needed therapeutic or prophylactic administration of antibiotics. The situation was complicated by the fact that some antibiotics, such as ciprofloxacin, can actually activate bacteriophages and increase toxin

production, so additional studies were necessary for these patients in order to choose the proper treatment.

Now that we understand the genetic identity of *E. coli* X, we can use the information we have learned to approach this problem.

To search for genes responsible for antibiotic resistance, we will use ResFinder (https://cge.food.dtu.dk/services/ResFinder/), which specifically searches a database of genes implicated in antibiotic resistance, identifying similarities between the sequenced genome and this database using local alignment.

Visit the ResFinder homepage, upload the "scaffolds.fasta" file from the SPAdes output, and in the field "Select Antimicrobial configuration", select "All". For comparison, do the same for the reference strain.

For your report, write down which antibiotics are *E. coli* X resistant to, and the reference vulnerable to.

10. Antibiotic resistance mechanism

We have discovered that *E. coli* X acquired not only the Shiga toxin but additional antibiotics resistance via HGT! But how?

We know that *E. coli* X is resistant to β-lactam antibiotics, which include penicillin. As we remember from the first week, there are at least four different resistance mechanisms. Besides the efflux pump mutations, some bacteria found more direct defense against these antibiotics by acquiring specific enzymes that can disrupt the antibiotic molecule itself. In our case, one of the common enzymes is β-lactamase (usually encoded by genes called *bla*). Search for these enzymes in the same way that we looked for the toxin genes: by using the Sequence Navigator in Mauve. Determine by looking at neighboring genes how *E. coli* X obtained these genes.

11. Conclusion

This story offers another example of increasing antibiotic resistance. Although doctors have been using antibiotics for less than 75 years, many bacteria have already developed antibiotics resistance during the decades of strong selective pressure. Moreover, bacteria have developed more and more sophisticated mechanisms to evade antibiotics, leading to multidrug-resistant pathogens.

Since their introduction, antibiotics have been viewed as a magic bullet that can cure any bacterial infection. Yet recent losses incurred at the hands of antibiotic-resistant bacteria demonstrate that the magic is waning. In May 2016, Thomas Frieden, Director of the U.S. Center for Disease Control, stated, "The medicine cabinet is empty for some patients. It is the end of the road for antibiotics unless we act urgently." The urgent need certainly involves the development of new antibiotics, but it also will require further bioinformatics research to help contain bacterial outbreaks before they can become epidemics.

For your lab report

For the abstract, remember that your goal is to figure out what was the cause of the *E.coli* outbreak, and describe properties of the novel pathogenic strain.

For the introduction, briefly cover how the *E.coli* can cause hemolytic uremic syndrome, the idea of horizontal transfer of the pathogenic factors. Also tell why it is important in some cases to assemble the genome *de novo* instead of aligning reads to a reference.

In your methods section, describe your assembly and annotation strategies, summarize each step of the analysis.

In your results section, briefly introduce the sequencing data (what are the samples, number of libraries, insert size), provide assembly statistics for one and three libraries, annotation statistics, number and length of 16S rRNA genes.

For discussion, given your results, explain how you think the *E.coli* X strain became pathogenic. Describe how it became resistant to certain antibiotics, describe mechanism of resistance, and suggest alternative treatment for the affected patients.