Project #3. E.coli outbreak investigation

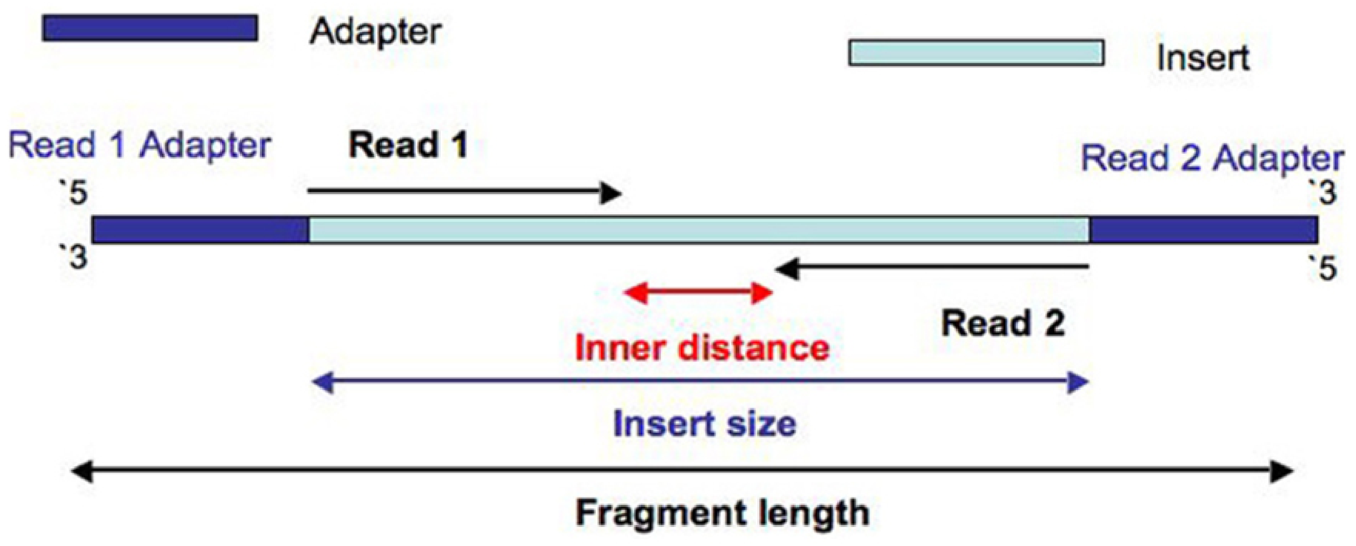
Made by Maria Uzun and Alisa Fedorenko

1. Exploring the dataset

```
mkdir Project3
cd Project3
mkdir raw_data
cd raw_data
```

The samples to work with:

- 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)



Download the data:

```
# SRR292678
wget https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292678sub_S1_L001_R1_001.fastq.gz
wget https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292678sub_S1_L001_R2_001.fastq.gz
# SRR292862
wget https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292862_S2_L001_R1_001.fastq.gz
wget https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292862_S2_L001_R2_001.fastq.gz
# SRR292770
wget https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292770_S1_L001_R1_001.fastq.gz
wget https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292770_S1_L001_R2_001.fastq.gz
```

Make the file with the file names:

```
for file in *R1_001.fastq.gz; do
    echo "${file%_R1_001.fastq.gz}" >> samples.txt
done
```

```
cd ..
mkdir fastqc
cd fastqc
cp ../raw_data/samples.txt .
```

Let’s make some statistics and quality control of the reads:

```
bash seqkit.sh
```

```
#!/bin/bash

: '
Script to run quality control (QC) analyses using seqkit and fastqc.

Iterates over a list of samples provided in samples.txt file.
For each sample, it performs seqkit stats on paired-end FASTQ files and runs FastQC.

Variables:
    - SAMPLE: Represents each sample name read from the samples.txt file.
    - FASTQ1 and FASTQ2: Names of the paired-end FASTQ files for each sample.
    - FASTQ1_PATH and FASTQ2_PATH: Paths to the paired-end FASTQ files.
    - OUTPUT_FILE_SEQKIT: Path to save seqkit stats output for each sample.
    - OUTPUT_DIR_FASTQC: Directory to store FastQC results.

Usage:
    Edit the paths for FASTQ files, output directories, and sample list (samples.txt).

Ensure seqkit and FastQC are installed and accessible from the terminal.

Note: This script assumes proper installation and configuration of seqkit and FastQC.
```

```
Execution:
    Run this script in the terminal using bash: bash script_name.sh
'

while read SAMPLE; do
    echo "Running sample ${SAMPLE}"

    FASTQ1=${SAMPLE}_R1_001.fastq.gz  # fastq names
    FASTQ2=${SAMPLE}_R2_001.fastq.gz

    FASTQ1_PATH="../raw_data/${FASTQ1}"  # <- edit the path to your fastqs
    FASTQ2_PATH="../raw_data/${FASTQ2}"

    OUTPUT_FILE_SEQKIT="../seqkit_statistics/stats_output_${SAMPLE}.txt"  # edit output seqkit file directory


    seqkit stats ${FASTQ1_PATH} ${FASTQ2_PATH} >> ${OUTPUT_FILE_SEQKIT}
    fastqc ${FASTQ1_PATH} ${FASTQ2_PATH} -o ./

done < samples.txt  # txt file with the list of all samples you would like to make QC
```

File	Format	Type	Num Seqs	Sum Len	Min Len	Avg Len	Max Len
SRR292678sub_S1_L001_R1_001.fastq.gz	FASTQ	DNA	5,499,346	494,941,140	90	90	90
SRR292678sub_S1_L001_R2_001.fastq.gz	FASTQ	DNA	5,499,346	494,941,140	90	90	90

File	Format	Type	Num Seqs	Sum Len	Min Len	Avg Len	Max Len
SRR292770_S1_L001_R1_001.fastq.gz	FASTQ	DNA	5,102,041	250,000,009	49	49	49
.SRR292770_S1_L001_R2_001.fastq.gz	FASTQ	DNA	5,102,041	250,000,009	49	49	49

File	Format	Type	Num Seqs	Sum Len	Min Len	Avg Len	Max Len
SRR292862_S2_L001_R1_001.fastq.gz	FASTQ	DNA	5,102,041	250,000,009	49	49	49
SRR292862_S2_L001_R2_001.fastq.gz	FASTQ	DNA	5,102,041	250,000,009	49	49	49

2. K-mer profile and genome size estimation

Install jellyfish:

```
conda install -c conda-forge jellyfish
```

```
mkdir jellyfish
cd ./raw_data # the directory with the raw reads
```

To *count kmers, we 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.*

Jellyfish count:

- m: kmer length, 21 is commonly used
- s: size of hash table: should be genome size + extra kmers from seq errors. However, it does say that hash size will be increased automatically if needed.
- C: canonical. Reverse complement kmers are considered to be identical and are counted as the same thing. This is recommended.
- t: number of threads
- output: mer_counts.jf

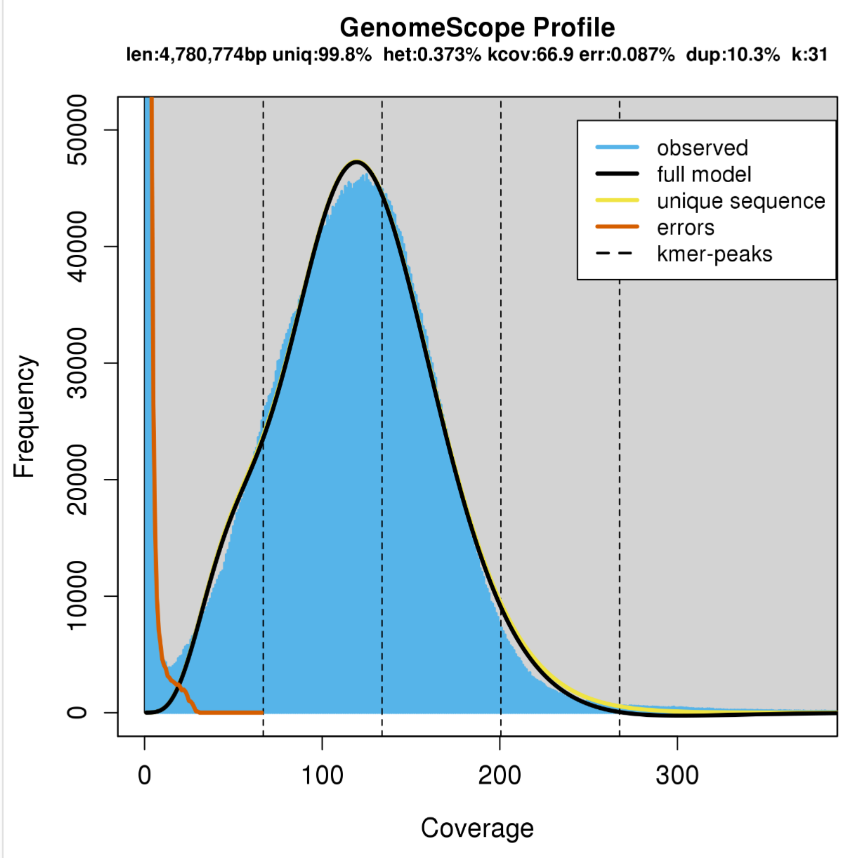
```
jellyfish count -m 31 -s 100M -t 128 -C <(zcat SRR292678sub_S1_L001_R1_001.fastq.gz) <(zcat SRR292678sub_S1_L001_R2_001.fastq.gz) -o ../jellyfish/SRR292678.jf
jellyfish count -m 31 -s 100M -t 128 -C <(zcat SRR292862_S2_L001_R1_001.fastq.gz) <(zcat SRR292862_S2_L001_R2_001.fastq.gz) -o ../jellyfish/SRR292862.jf
jellyfish count -m 31 -s 100M -t 128 -C <(zcat SRR292770_S1_L001_R1_001.fastq.gz) <(zcat SRR292770_S1_L001_R2_001.fastq.gz) -o ../jellyfish/SRR292770.jf
cd ..
```

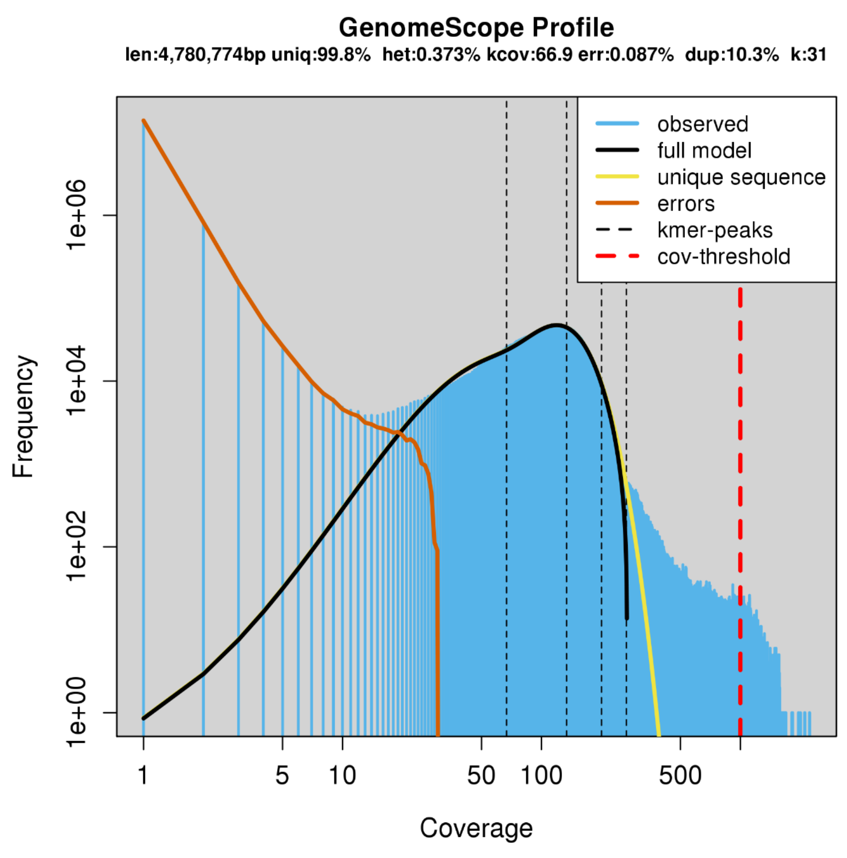
jellyfish histo:

```
jellyfish histo -t 128 --high=1000000 ./jellyfish/SRR292678.jf > ./jellyfish/SRR292678_reads.histo
jellyfish histo -t 128 --high=1000000 ./jellyfish/SRR292862.jf > ./jellyfish/SRR292862_reads.histo
jellyfish histo -t 128 --high=1000000 ./jellyfish/SRR292770.jf > ./jellyfish/SRR292770_reads.histo
```

Vizualizing histo plots for our reads

SRR292678:

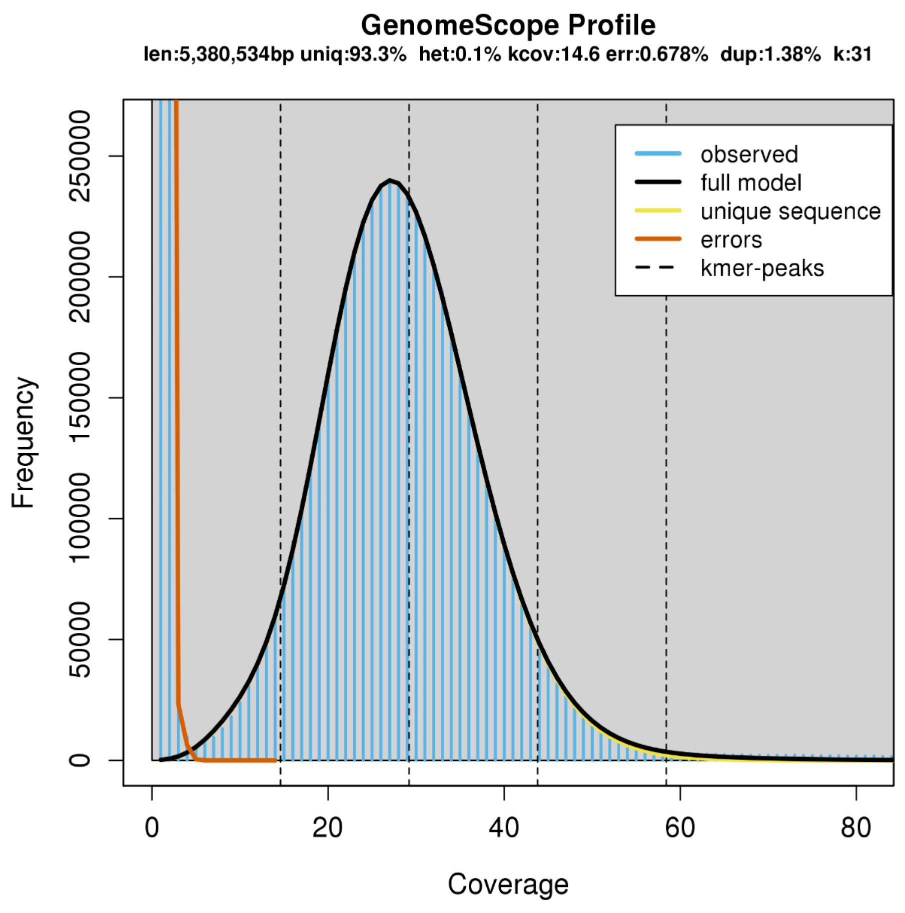


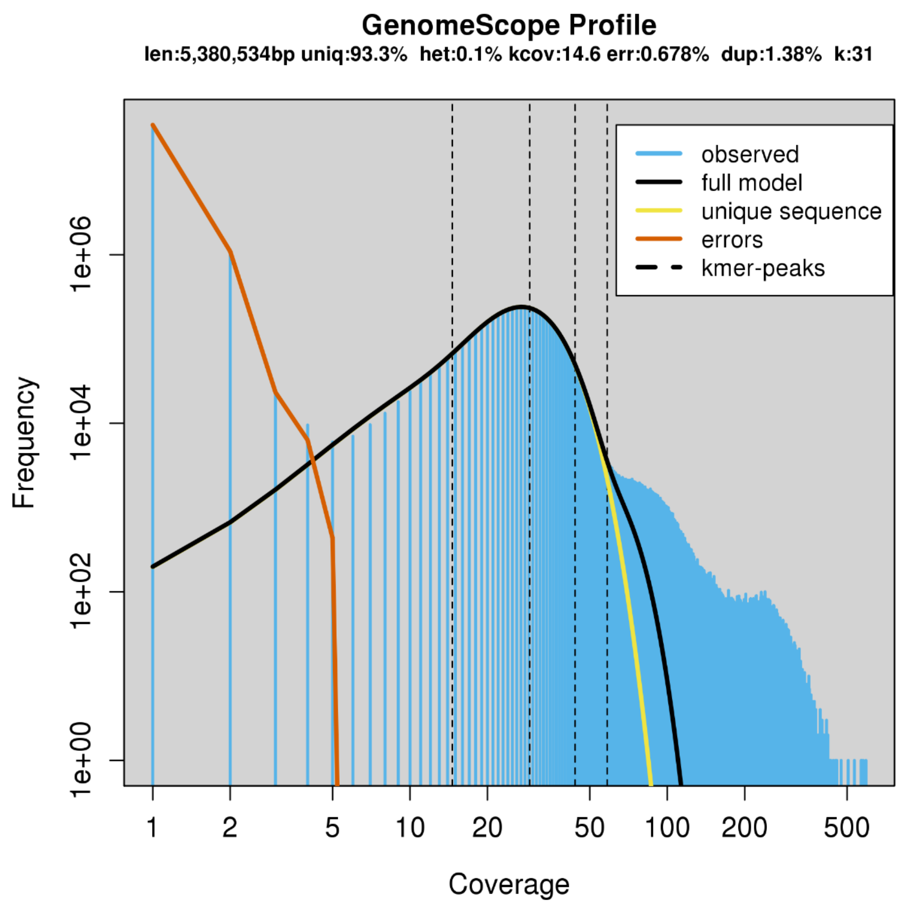


```
GenomeScope version 1.0
k = 31

property                      min                 max
Heterozygosity                0.364681%           0.380497%
Genome Haploid Length         4,761,140 bp        4,780,774 bp
Genome Repeat Length          11,019 bp           11,065 bp
Genome Unique Length          4,750,121 bp        4,769,709 bp
Model Fit                     93.6457%            94.3204%
Read Error Rate               0.0870234%          0.0870234%
```

SRR292862 and SRR292770:





```
GenomeScope version 1.0
k = 31

property                      min                 max
Heterozygosity                0.0966549%          0.103782%
Genome Haploid Length         5,371,456 bp        5,380,534 bp
Genome Repeat Length          358,141 bp          358,746 bp
Genome Unique Length          5,013,315 bp        5,021,788 bp
Model Fit                     97.003%             98.3288%
Read Error Rate               0.677788%           0.677788%
```

According to k-mer distibution. we can estimate genome size. It would be approximately 5.3 Mb.

3. Assembling *E. coli* X genome from paired reads

Install spades:

```
conda install spades -c bioconda
cd ..
mkdir spades
cd spades
```

It is recommended to run SPAdes with the high computational capacity.

For pair-end data:

```
spades.py --pe1-1 ./raw_data/SRR292678sub_S1_L001_R1_001.fastq.gz --pe1-2 ../raw_data/SRR292678sub_S1_L001_R2_001.fastq.gz --threads 128 -o SRR292678sub_S1_spades
```

The output files:

```
# assembly_graph_after_simplification.gfa
# assembly_graph_with_scaffolds.gfa
# contigs.fasta  dataset.info
# K21
# K55
# params.txt
# run_spades.sh
# scaffolds.fasta
# spades.log
# assembly_graph.fastg
# before_rr.fasta
# contigs.paths
# input_dataset.yaml
# K33  misc
# pipeline_state
# run_spades.yaml
# scaffolds.paths  tmp
```

Assess the quality of the resulting assembly:

```
quast.py ./SRR292678sub_S1_spades/scaffolds.fasta -o ./scaffolds_quast_SRR292678sub -t 128
quast.py ./SRR292678sub_S1_spades/contigs.fasta -o ./contigs_quast_SRR292678sub -t 128
```

Results:

Statistics without reference	**scaffolds**	**contigs**
# contigs	221	210
# contigs (>= 0 bp)	372	386
# contigs (>= 1000 bp)	158	159
# contigs (>= 5000 bp)	82	81
# contigs (>= 10000 bp)	67	67
# contigs (>= 25000 bp)	50	50
# contigs (>= 50000 bp)	32	32
Largest contig	300763	300763
Total length	5304595	5295721
Total length (>= 0 bp)	5336365	5334575
Total length (>= 1000 bp)	5259608	5259101
Total length (>= 5000 bp)	5081904	5076685
Total length (>= 10000 bp)	4977737	4977737
Total length (>= 25000 bp)	4714504	4714504
Total length (>= 50000 bp)	4035821	4035821
N50	111860	111860
N75	55279	55279
L50	14	14
L75	31	31
GC (%)	50.53	50.56
Mismatches		
# N's	1790	0
# N's per 100 kbp	33.74	0

4. Impact of reads with large insert size

For pair-end and mate-pair data:

```
spades.py --pe1-1 ./raw_data/SRR292678sub_S1_L001_R1_001.fastq.gz --pe1-2 ./raw_data/SRR292678sub_S1_L001_R2_001.fastq.gz \
    --mp1-1 ./raw_data/SRR292862_S2_L001_R1_001.fastq.gz --mp1-2 ./raw_data/SRR292862_S2_L001_R2_001.fastq.gz \
    --mp2-1 ./raw_data/SRR292770_S1_L001_R1_001.fastq.gz --mp2-2 ./raw_data/SRR292770_S1_L001_R2_001.fastq.gz \
    --threads 128 -o spades_output_3lib
```

The output files:

```
# assembly_graph_after_simplification.gfa
# before_rr.fasta  corrected
# K21
# misc
# run_spades.sh
# scaffolds.paths
# assembly_graph.fastg
# contigs.fasta
# dataset.info
# K33
# params.txt
# run_spades.yaml
# spades.log
# assembly_graph_with_scaffolds.gfa
# contigs.paths
# input_dataset.yaml
# K55
# pipeline_state
# scaffolds.fasta
# tmp
```

Assess the quality of the resulting assembly:

```
quast.py ./spades_output_3lib/scaffolds.fasta -o ./scaffolds_quast_3lib -t 128
quast.py ./spades_output_3lib/contigs.fasta -o ./contigs_quast_3lib -t 128
```

Results:

Statistics without reference	**scaffolds**	**contigs**
# contigs	90	105
# contigs (>= 0 bp)	327	369
# contigs (>= 1000 bp)	54	79
# contigs (>= 5000 bp)	16	33
# contigs (>= 10000 bp)	13	30
# contigs (>= 25000 bp)	10	26
# contigs (>= 50000 bp)	10	22
Largest contig	2815616	698474
Total length	5391554	5350156
Total length (>= 0 bp)	5437160	5403327
Total length (>= 1000 bp)	5365719	5331230
Total length (>= 5000 bp)	5258076	5202939
Total length (>= 10000 bp)	5238939	5183802
Total length (>= 25000 bp)	5200270	5133691
Total length (>= 50000 bp)	5200270	4975501
N50	2815616	335515
N75	391920	143558
L50	1	6
L75	4	13
GC (%)	50.57	50.59
Mismatches		
# N's	33833	0
# N's per 100 kbp	627.52	0

5. Genome Annotation

Download Prokka:

```
conda install -c bioconda prokka
```

Run PROKKA to annotate bacterial scaffolds:

```
prokka --outdir ./prokka_centre_sp13 --centre XXX ./spades_output_3lib/scaffolds.fasta

prokka --outdir ./prokka_centre_sp13_mc --centre XXX /mnt/storage/lab4/Project3/spades_output_3lib_sc_sp13_mc/scaffolds.fasta
prokka --outdir ./prokka_centre_sp15_mc --centre XXX /mnt/storage/lab4/Project3/spades_output_3lib_sp15_mc/scaffolds.fasta
```

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.

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

First, we need to locate 16S rRNA in the assembled *E. coli* X genome. You can use the rRNA genes prediction tool Barrnap.

Download rRNA genes prediction tool Barrnap:

```
conda install -c bioconda -c conda-forge barrnap
```

Run Barrnap to predict 16S RNA:

```
barrnap --quiet PROKKA_11172023.fna -t 128 >> barrnap_results.txt
```

The Barrnap output for 16s RNA:

Genome position	Start	End	E-value	Strand	Length	Product
gnl|BS|PROKKA_000001	326359	327896	0	-	1537	Name=16S_rRNA;product=16S ribosomal RNA
gnl|BS|PROKKA_000001	595966	597503	0	-	1537	Name=16S_rRNA;product=16S ribosomal RNA
gnl|BS|PROKKA_000001	2504403	2505940	0	-	1537	Name=16S_rRNA;product=16S ribosomal RNA
gnl|BS|PROKKA_000005	43835	45372	0	+	1537	Name=16S_rRNA;product=16S ribosomal RNA
gnl|BS|PROKKA_000005	85462	86999	0	+	1537	Name=16S_rRNA;product=16S ribosomal RNA
gnl|BS|PROKKA_000006	111955	113492	0	+	1537	Name=16S_rRNA;product=16S ribosomal RNA
gnl|BS|PROKKA_000071	314	719	9.80E-23	+	405	Name=16S_rRNA;product=16S ribosomal RNA (partial);note=aligned only 25 percent of the 16S ribosomal RNA

We can search the fasta sequences of 16S RNAs in .ffn file to use in for blasting sequences.

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*** **(taxid:562)** 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.

NCBI results:

Name: *Escherichia coli* 55989, complete sequence

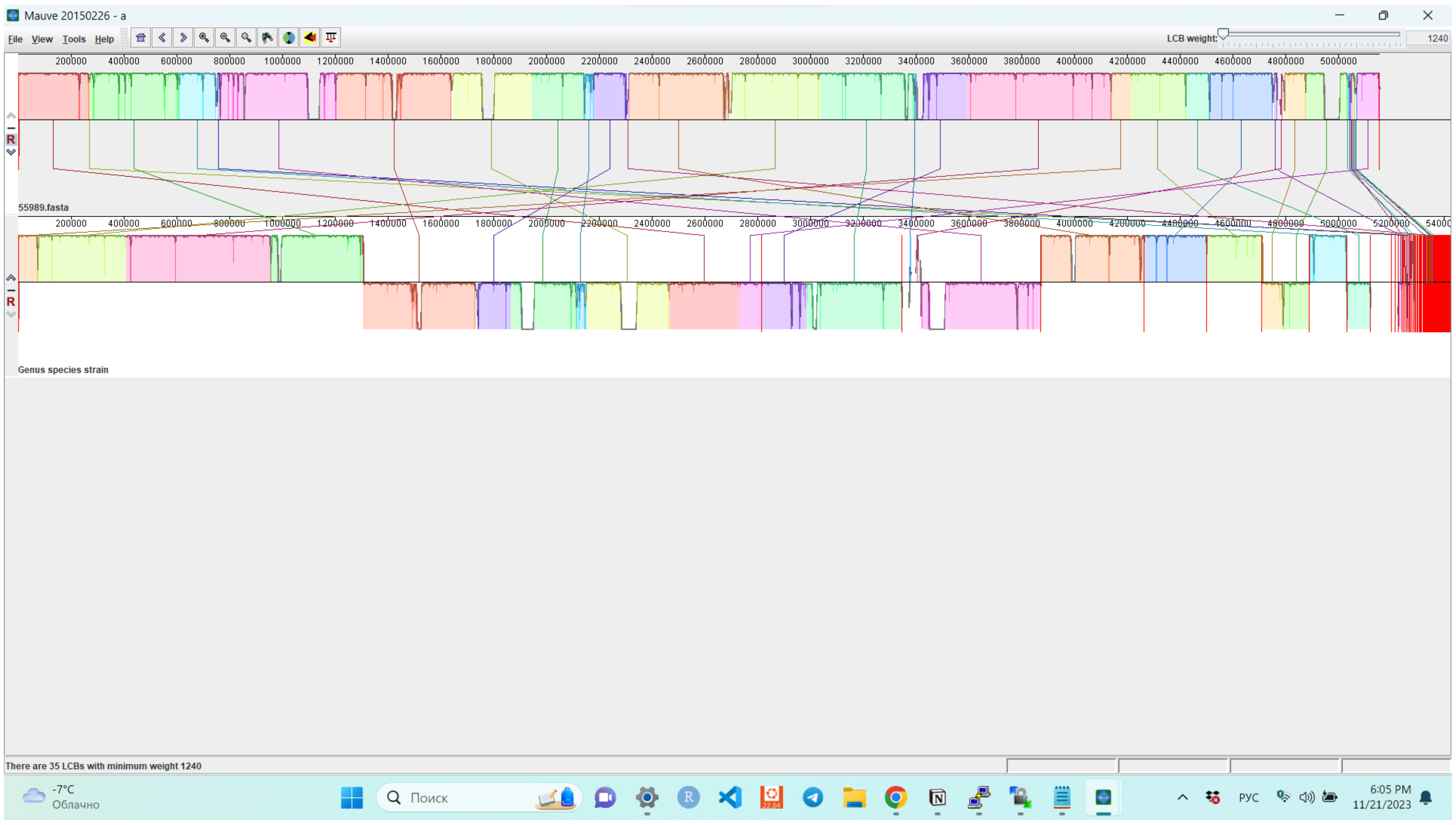
GenBank accession: **NC_011748.1**

7. What is the genetic cause of HUS?

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.

We will now analyze the genome-wide alignment of *E. coli* X and the reference genome.


Mauve 20150226 - a

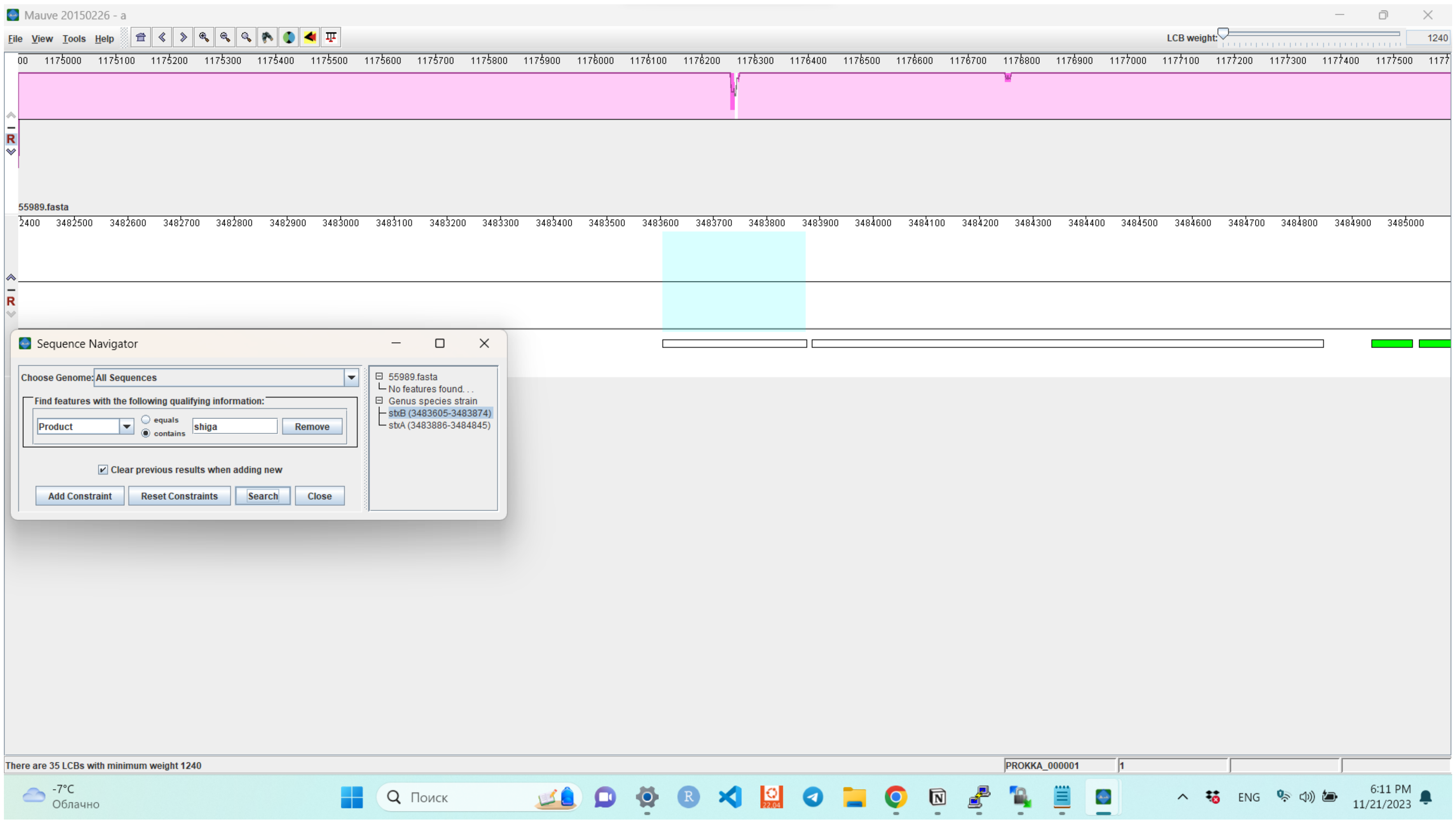
55989.fasta

Genus species strain

LCB weight: 1240

There are 35 LCBs with minimum weight 1240

We found Shiga toxin-related genes:


Mauve 20150226 - a

55989.fasta

Sequence Navigator

Choose Genome: All Sequences

Find features with the following qualifying information:

Product [✓ contains] equals shiga

[✓] Clear previous results when adding new

```
⊟ 55989.fasta
  └ No features found. . .
⊟ Genus species strain
  ├ stxB (3483605-3483874)
  └ stxA (3483886-3484845)
```

There are 35 LCBs with minimum weight 1240

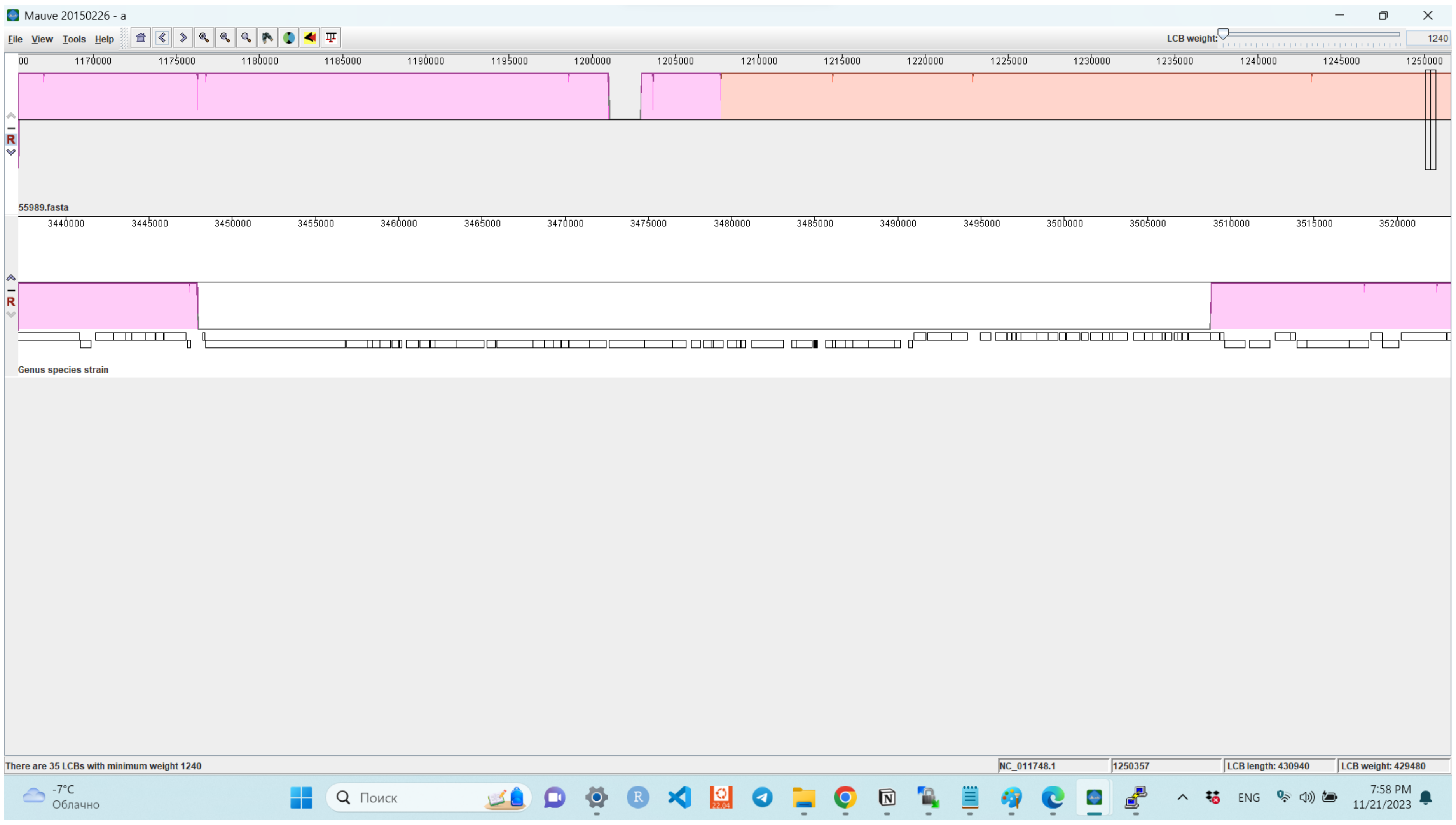
PROKKA_000001 1

***stxB* (Shiga toxin subunit B)**, length - **269** , location 3483605-3483874

***stxA* (Shiga toxin subunit A),** length - **959**, location 3483886-3484845

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

Shiga toxin genes are found in insertion site, and they are surrounded by integrase on one side and a hypothetical protein of the *Escherichia* phage on the other. Moreover, it contained a substantial number of phage genes within its structure. This means that this region was obtained by horizontal gene transfer (HGT).



9. Antibiotic resistance detection

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.

Antibiotics to which reference *E.Coli is* vulnerable to:

Tetracycline								
Resistance gene	Identity	Alignment Length/Gene Length	Position in reference	Contig or Depth	Position in contig	Phenotype	PMID	Accession no.
tet(B)	100.0	1206/1206	1..1206	NC_011748.1 Escherichia coli 55989, complete sequence	4974055..4975260	doxycycline,tetracycline,minocycline	11553538	AF326777

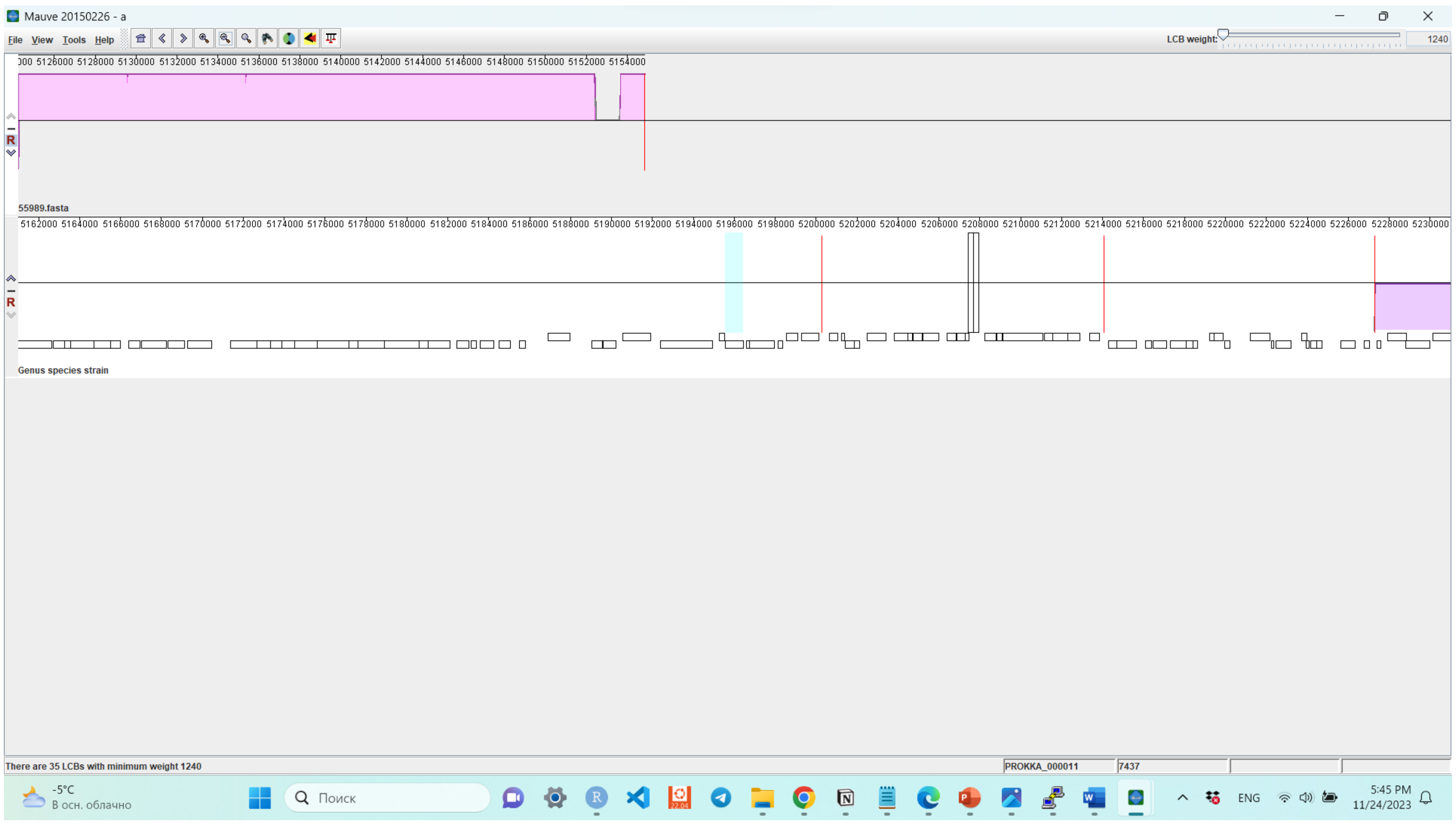
Antibiotics to which *E. coli* X is resistant to, and the reference is vulnerable to:

Resistance gene	Identity	Alignment Length/Gene Length	Coverage	Position in reference	Contig	Position in contig	Phenotype	Hy Accession no.
blaTEM-1B	100	861/861	100	1..861	gnl|BS|PROKKA_000010	78991..79851	Beta-lactam resistance Alternate name; RblaTEM-1	AY458016
blaCTX-M-15	100	876/876	100	1..876	gnl|BS|PROKKA_000010	75294..76169	Beta-lactam resistance Alternate name; UOE-1	AY044436
sul1	100	761/840	90.59524	1..761	gnl|BS|PROKKA_000024	3604..4364	Sulphonamide resistance	AY115475
sul1	100	761/882	86.28118	1..761	gnl|BS|PROKKA_000024	3604..4364	Sulphonamide resistance	DQ914960
sul1	100	761/840	90.59524	1..761	gnl|BS|PROKKA_000024	3604..4364	Sulphonamide resistance	U12338
sul1	100	761/828	91.90821	1..761	gnl|BS|PROKKA_000024	3604..4364	Sulphonamide resistance	AY522923
sul2	100	816/816	100	1..816	gnl|BS|PROKKA_000023	1918..2733	Sulphonamide resistance	HQ840942
qacE	100	282/333	84.68468	1..282	gnl|BS|PROKKA_000024	3263..3544	Disinfectant resistance	X68232
aph(6)-Id	100	837/837	100	1..837	gnl|BS|PROKKA_000023	3597..4433	Aminoglycoside resistance Alternate name; aph(6)-Id	M28829

aph(3'')-Ib	100	804/804	100	1..804	gnl|BS|PROKKA_000023	2794..3597	Aminoglycoside resistance Alternate name; aph(3'')-Ib	AF321551
tet(A)	100	1200/1200	100	1..1200	gnl|BS|PROKKA_000026	956..2155	Tetracycline resistance	AJ517790
dfrA7	100	474/474	100	1..474	gnl|BS|PROKKA_000024	2560..3033	Trimethoprim resistance	AB161450

10. Antibiotic resistance mechanism

We detected genes of beta-lactamase CTX-M-1 (*bla_1*) and beta-lactamase TEM precursor (*bla_2*) in the part of genome, which was not aligned to reference genome.



There are 35 LCBs with minimum weight 1240

We suggested, that this is plasmid. These genes can hydrolyze drug molecules, providing antibiotic resistance. In the neighboring of these genes, phage integrases and mobile element genes were found which means these genes were also acquired through HGT.

Generally, we have discovered that *E. coli* X acquired not only the Shiga toxin but additional antibiotics resistance in its plasmid via HGT.