Project # 1

What causes antibiotic resistance?

18.10.19 The reference sequence of the parental *E. coli* strain K-12 substrain MG1655 was downloaded from NCBI FTP (<ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/005/845/GCA_000005845.2_ASM584v2>) – sequence in .fasta format (GCF\_000005845.2\_ASM584v2\_genomic.fna) and annotation in .gff format (GCF\_000005845.2\_ASM584v2\_genomic.fna). More information here: <https://www.ncbi.nlm.nih.gov/genome/167>.

Raw illumina sequencing reads from shotgun sequencing of an *E. coli* strain that is resistant to the antibiotic ampicillin was downloaded from <http://public.dobzhanskycenter.ru/mrayko/amp_res_1.fastq.zip> (forward); <http://public.dobzhanskycenter.ru/mrayko/amp_res_2.fastq.zip> (reverse).

How many lines are there in each fastq file? Using **wc -l amp\_res\_1.fastq**, **wc -l amp\_res\_2.fastqc** it was found that the first file and the second one contain each 1823504 lines.

Fastqc was installed (<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>) – a program for statistical analysis.

Getting some statistics concerning raw reads: **fastqc -o . /[path\_to\_file\_1]/amp\_res\_1.fastqc /[path\_to\_file\_2]/amp\_res\_2.fastqc**. 4 files were generated: amp\_res\_1\_fastqc.zip, amp\_res\_1\_fastqc.html, amp\_res\_2\_fastqc.zip, amp\_res\_2\_fastqc.html. The basic statistics match what was calculated before: 1823504 / 4 = 455876 (each read in .fastq file have four lines of information). Red circles: “per base sequence quality”, “per tile sequence quality” (first file), “per base sequence quality” (second file).

Trimmomatic was installed (<http://www.usadellab.org/cms/?page=trimmomatic>) – a program for filtering the reads.

**java -jar ~/ib/Project1/trimmomatic-0.39.jar PE amp\_res\_1.fastq amp\_res\_2.fastq out1.fq out2u.fq out3r.fq out4ru.fq LEADING:20 TRAILING:20 SLIDINGWINDOW:10:20 MINLEN:20**

Output:

Multiple cores found: Using 4 threads

Quality encoding detected as phred33

Input Read Pairs: 455876 Both Surviving: 446259 (97,89%) Forward Only Surviving: 9216 (2,02%) Reverse Only Surviving: 273 (0,06%) Dropped: 128 (0,03%)

TrimmomaticPE: Completed successfully

The second attempt:

**amp\_res\_1.fastq amp\_res\_2.fastq out51.fq out52u.fq out53r.fq out54ru.fq LEADING:30 TRAILING:30 SLIDINGWINDOW:4:20 MINLEN:20**

Output:

Multiple cores found: Using 4 threads

Quality encoding detected as phred33

Input Read Pairs: 455876 Both Surviving: 423785 (92,96%) Forward Only Surviving: 22067 (4,84%) Reverse Only Surviving: 6817 (1,50%) Dropped: 3207 (0,70%)

TrimmomaticPE: Completed successfully

Results generated for the second time were used.

out51.fq, out51r.fq: both the forward and reverse read passed the trimming filter.

out51u.fq, out51ru.fq: only one read passed the trimming filter.

20.10.2019 bwa was installed (Li and Durbin, 2009) – an aligner program

Indexing reference file: **bwa index GCF\_000005845.2\_ASM584v2\_genomic.fna**

Aligning: **bwa mem GCF\_000005845.2\_ASM584v2\_genomic.fna out51.fq out53r.fq > alignment.sam**

Samtools was installed ((<https://github.com/samtools/samtools>) – a set of utilities for interacting with and post-processing short DNA sequence read alignments in the SAM (Sequence Alignment/Map), BAM (Binary Alignment/Map) and CRAM formats.

Compressing .SAM file into .BAM file: **samtools view -S -b alignment.sam > alignment.bam**

Getting some statistics: **samtools flagstat alignment.bam**

Output:

samtools flagstat alignment.bam

847778 + 0 in total (QC-passed reads + QC-failed reads)

0 + 0 secondary

208 + 0 supplementary

0 + 0 duplicates

846529 + 0 mapped (99.85% : N/A)

847570 + 0 paired in sequencing

423785 + 0 read1

423785 + 0 read2

843592 + 0 properly paired (99.53% : N/A)

845302 + 0 with itself and mate mapped

1019 + 0 singletons (0.12% : N/A)

0 + 0 with mate mapped to a different chr

0 + 0 with mate mapped to a different chr (mapQ>=5)

Sorting .BAM file: **samtools sort alignment.bam -o alignment\_sorted.bam**

Indexing .BAM file: **samtools index alignment\_sorted.bam**

IGV Browser was installed (<https://software.broadinstitute.org/software/igv/>) – Integrative Genomics Viewer, a visualisation tool

Visualisation: IGV Browser -> “Genomes” -> “Load Genome from File” (GCF\_000005845.2\_ASM584v2\_genomic.fna); “File” -> “Open from File” (alignment\_sorted.bam)

Going through data, examining each position in the reference genome to see how many reads have a mutation at the same position: **samtools mpileup -f GCF\_000005845.2\_ASM584v2\_genomic.fna alignment\_sorted.bam > my.mpileup**

21.10.2019 VarScan was installed (<http://dkoboldt.github.io/varscan/>) – a variant caller, version 2.3.9

Variant calling: **java -jar VarScan.v2.3.9.jar mpileup2snp my.mpileup --min-var-freq 0.50**

**--variants --output-vcf 1 > VarScan\_results.vcf**

Visualisation via IGV: annotation in .gff format and a .vcf file were added.

Five mutations were found.