**Supplementary materials**

**Essential mutations in Escherichia coli strain K-12 substrain MG1655 genome associated with antibiotic resistance**

1. **Loading of the data**

We tried to download the reference sequence of the parental *E. coli* strain.

**Command:** *wget ftp://*[*ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia\_coli/all\_assembly\_versions/GCA\_000005845.2\_ASM584v2/GCF\_000005845.2\_ASM584v2\_genomic.fna.gz*](http://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia_coli/all_assembly_versions/GCA_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz)

**Error:** *-2020-10-27 23:09:01-- ftp://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia\_coli/all\_assembly\_versions/GCA\_000005845.2\_ASM584v2/GCF\_000005845.2\_ASM584v2\_genomic.fna.gz*

*=> ‘GCF\_000005845.2\_ASM584v2\_genomic.fna.gz’*

*Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.11, 2607:f220:41e:250::10, 2607:f220:41e:250::7, ...*

*Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.11|:21... connected.*

*Logging in as anonymous ... Logged in!*

*==> SYST ... done. ==> PWD ... done.*

*==> TYPE I ... done. ==> CWD (1) /genomes/genbank/bacteria/Escherichia\_coli/all\_assembly\_versions/GCA\_000005845.2\_ASM584v2 ...*

*No such directory ‘genomes/genbank/bacteria/Escherichia\_coli/all\_assembly\_versions/GCA\_000005845.2\_ASM584v2’.*

**Cause:** mistake in the name of the file

**Command:** *wget ftp://*[*ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF\_000005845.2\_ASM584v2/GCF\_000005845.2\_ASM584v2\_genomic.fna.gz*](http://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz)

**Command:** *wget ftp://*[*ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF\_000005845.2\_ASM584v2/GCF\_000005845.2\_ASM584v2\_genomic.gff.gz*](http://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.gff.gz)

**Command:** *wget* [*ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF\_000005845.2\_ASM584v2/GCF\_000005845.2\_ASM584v2\_cds\_from\_genomic.fna.gz*](ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_cds_from_genomic.fna.gz)

**\_**.fna.gz is sequence in fasta format and \_.gff.gz is annotation of the genome. This is E.coli strain K-12 substrain MG1655.

Then we downloaded raw illumina sequencing reads from shotgun sequencing of E. coli strain with resistance. This was a paired end run so there were two sets of reads.

**Command:** *wget* [*http://public.dobzhanskycenter.ru/mrayko/amp\_res\_1.fastq.zip*](http://public.dobzhanskycenter.ru/mrayko/amp_res_1.fastq.zip)

**Command:** *wget* [*http://public.dobzhanskycenter.ru/mrayko/amp\_res\_2.fastq.zip*](http://public.dobzhanskycenter.ru/mrayko/amp_res_2.fastq.zip)

1. **Examination of files**

We tried to look at sequence files.

**Command:** *head -20 amp\_res\_1.fastq.zip*

**Result:** unreadable symbols

**Cause:** forgot that this is zip format

Remembered utility zcat, which unzips archives.

**Command:** *zcat amp\_res\_1.fastq.zip | head -n 20*

**Result:**

@SRR1363257.37 GWZHISEQ01:153:C1W31ACXX:5:1101:14027:2198 length=101

GGTTGCAGATTCGCAGTGTCGCTGTTCCAGCGCATCACATCTTTGATGTTCACGCCGTGGCGTTTAGCAATGCTTGAAAGCGAATCGCCTTTGCCCACACG

+

@?:=:;DBFADH;CAECEE@@E:FFHGAE4?C?DE<BFGEC>?>FHE4BFFIIFHIBABEECA83;>>@>@CCCDC9@@CC08<@?@BB@9:CC#######

First symbol of each line displayed in red color.

First line - starts with @ and contains some info about the read;

Second line - sequence;

Third line - starts with + and may contain the repeated info;

Fourth line - quality info for each base coded in ASCII (decode each symbol, subtract 33 for Phred33 scale or 64 for Phred64 scale), 0-40, the greater the better accuracy of the base call.

**Command:** *zcat GCF\_000005845.2\_ASM584v2\_genomic.fna.gz | head -n 20*

**Result:**

>NC\_000913.3 Escherichia coli str. K-12 substr. MG1655, complete genome

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGCTTCTGAACTG

We observed another format of the complete genome file, because it is a whole stuff.

After that we counted how many reads were in each fastq file.

**Command:** *zcat amp\_res\_1.fastq.zip | wc -l*

**Result:** 1823504

**Command:** *zcat amp\_res\_2.fastq.zip | wc -l*

**Result:** 1823504

We have 4 lines for each read, so in 1.fastq.zip there are 455876 reads and in 2.fastq.zip there are also 455876 reads.

1. **Quality analysis**

We downloaded fastqc. The Fastqc is a statistics analysis program for sequence data. So we used it to assess the quality of the reads.

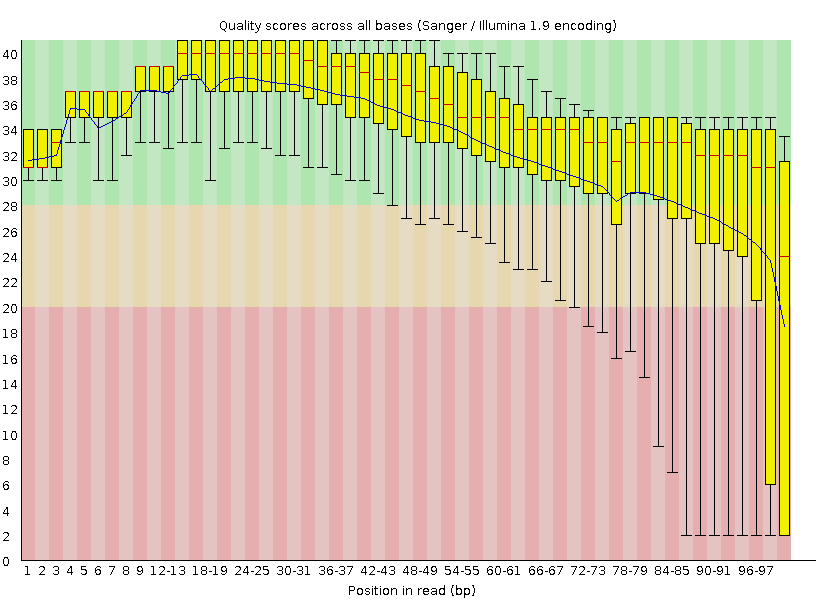
**Command:** *sudo apt-get install fastqc*

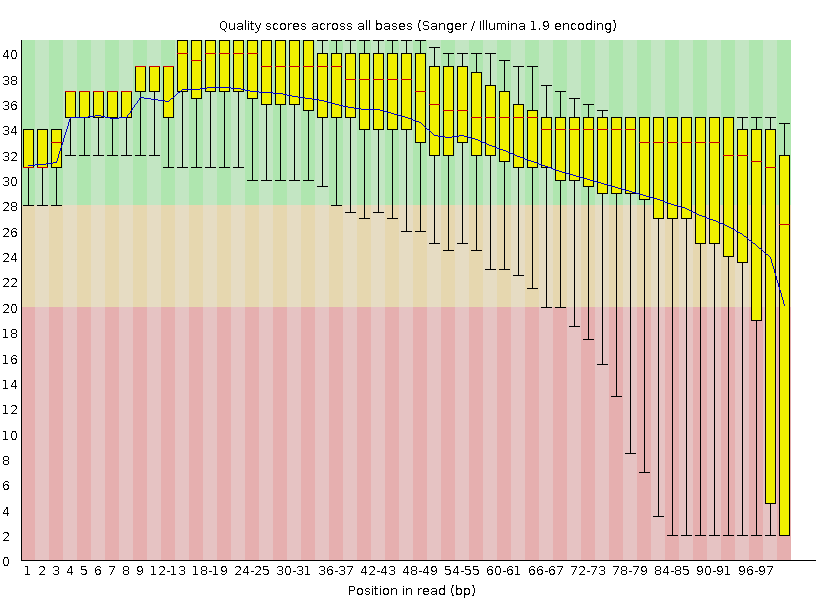
**Command:** *fastqc -o . ./amp\_res\_1.fastq*

**Command:** *fastqc -o . ./amp\_res\_2.fastq*

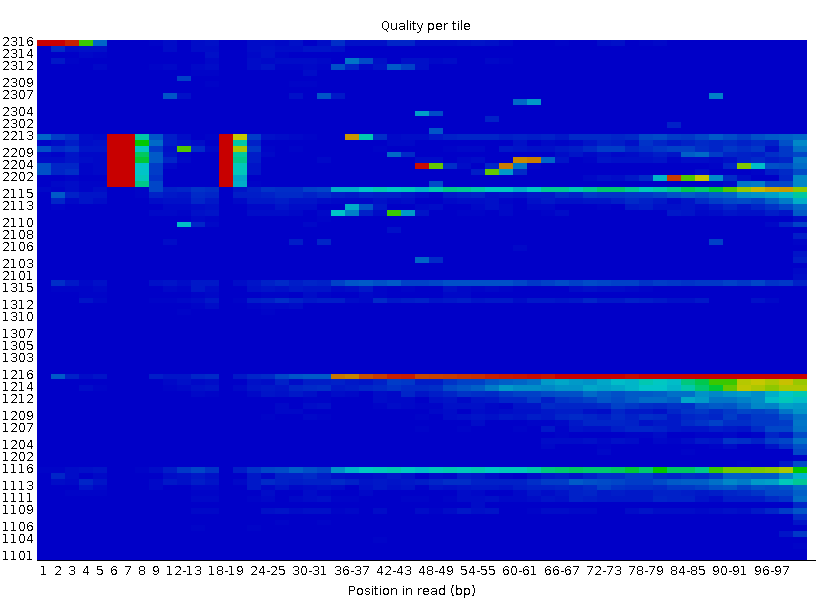
The result is \_.html file. There are two unusual results (red):

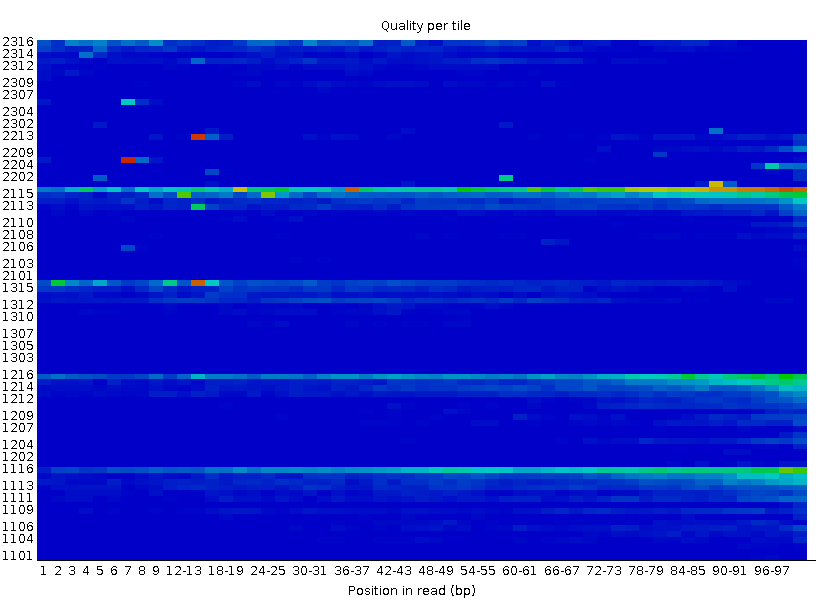
* Rer base sequence quality (because the first quartile of any base less than 5 - degradation of quality over the duration of long runs)





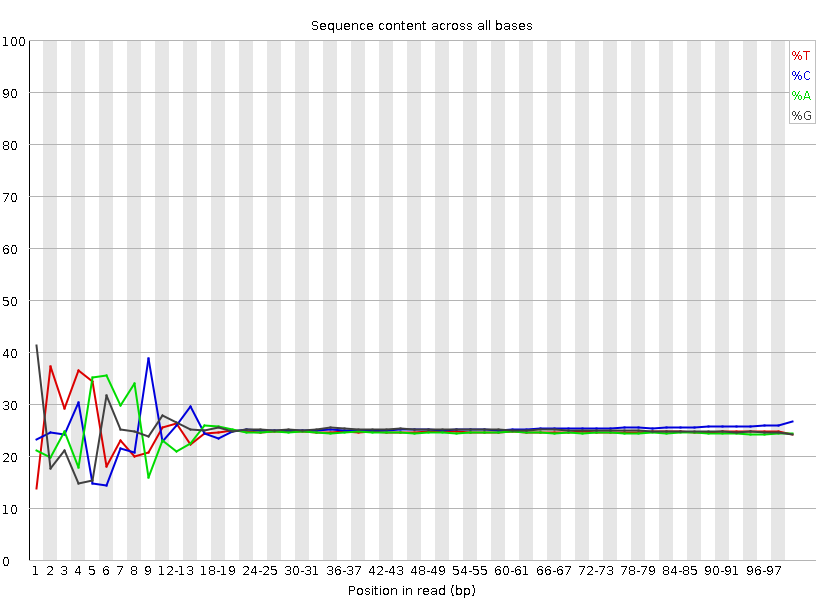
* Per tile sequence quality (because mean Phred score more than 5 less than the mean for that base across all tiles - maybe problems with flowcells, we can see that in second file the result is slightly abnormal)

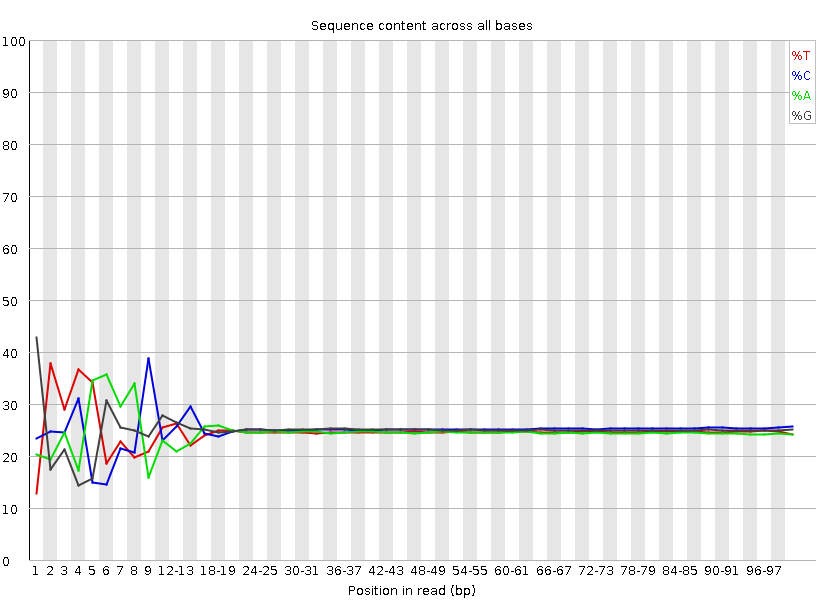




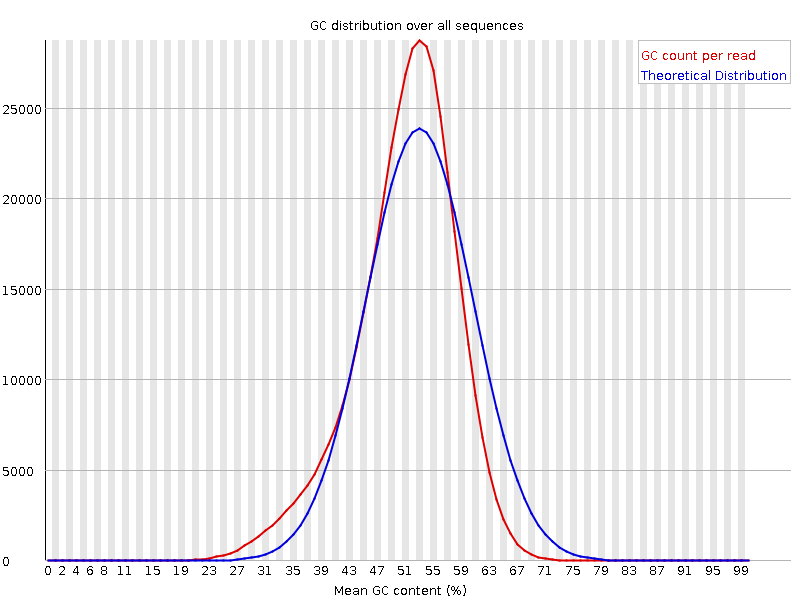
and two abnormal (yellow):

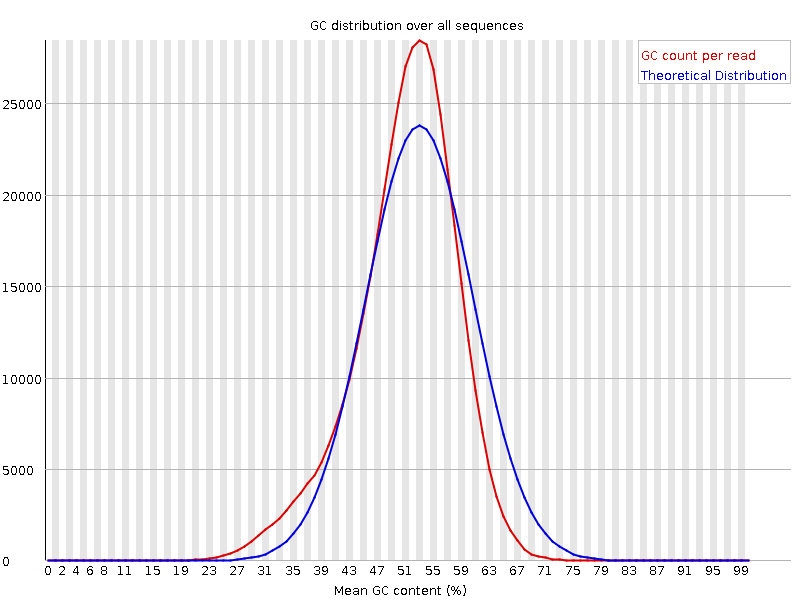
* Per base sequence content (difference between bases more than 10% - biased sequence composition, the library was prepared using random hexamers or through fragmentation with transposases because bias is only around first 18 nucleotides)





* Per sequence GC content (the sum of the deviations from the normal distribution represents more than 15% of the reads, little shift to the left - maybe because of systematic bias)





Per base sequence quality failure can be fixed by trimming and per tile sequence quality failure we can ignore because of specific patterns on the graph (which may be caused by technical problems).

1. **Refinement of the data**

To improve the quality of our reads we used the Trimmomatic program.

**Command:** *sudo apt-get install trimmomatic*

Trimmomatic is a tool for removing low-quality reads and trimming ends of the reads. These procedures improve quality of the reads which helps to achieve better results in further processing of the data. The Trimmomatic can be used on both paired ends and single ends and includes several steps. Set of steps can differ depending on the situation since there is no optimal and universal trimming strategy. List of steps with descriptions is shown in the table below.

|  |  |
| --- | --- |
| Step | Description |
| ILLUMINACLIP | Cut adapter and other illumina-specific sequences from the read. |
| SLIDINGWINDOW | Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold. |
| LEADING | Cut bases off the start of a read, if below a threshold quality |
| TRAILING | Cut bases off the end of a read, if below a threshold quality |
| CROP | Cut the read to a specified length |
| HEADCROP | Cut the specified number of bases from the start of the read |
| MINLEN | Drop the read if it is below a specified length |
| TOPHRED33 | Convert quality scores to Phred-33 |
| TOPHRED64 | Convert quality scores to Phred-64 |

On our run we used the following parameters:

|  |  |
| --- | --- |
| Step | Description |
| LEADING:20 | 20 is a quality threshold |
| TRAILING:20 | 20 is a quality threshold |
| SLIDINGWINDOW:10:20 | 10-base wide window and 20 is a quality threshold |
| MINLEN:20 | 20 is a length threshold |

**Command:** *TrimmomaticPE -phred33 amp\_res\_1.fastq amp\_res\_2.fastq amp\_res\_1P.fq amp\_res\_1U.fq amp\_res\_2P.fq amp\_res\_2U.fq LEADING:20 TRAILING:20 SLIDINGWINDOW:10:20 MINLEN:20*

Deciphering of the arguments: <name of the program> <phred scale> <input file forward> <input file reverse> <output paired forward> <output unpaired forward> <output paired reverse> <output unpaired reverse> <steps> …

In the end of the run we got some back statistics:

*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%)*

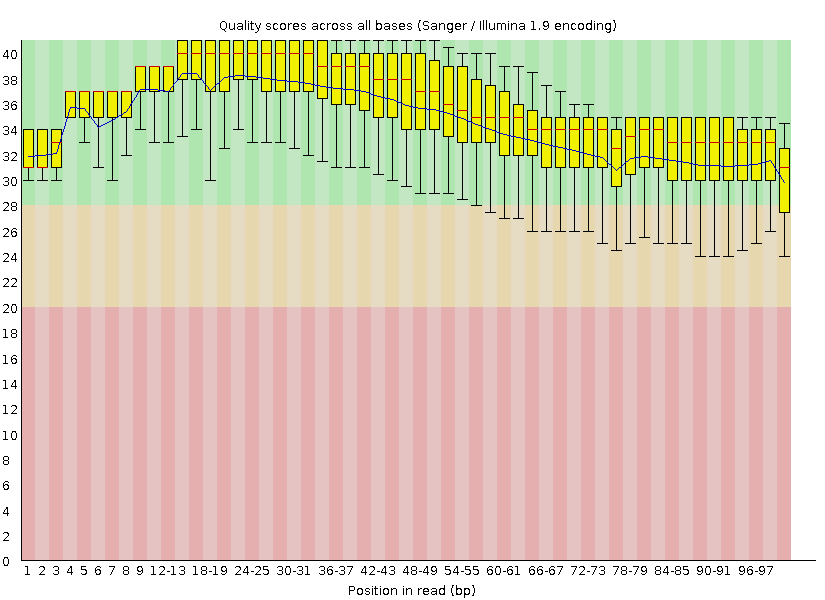
Line count was checked manually, there were 446259 - the same number.

After using TrimmomaticPE we run fastqc again to evaluate changes.

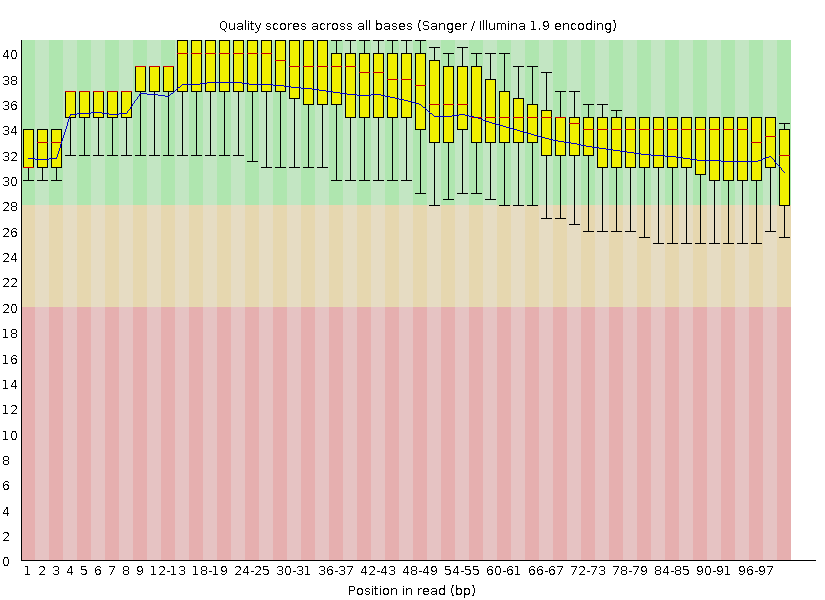
**Command:** *fastqc -o . ./amp\_res\_1P.fq ./amp\_res\_2P.fq*

As can be seen from the results we improved per base sequence quality:

## [OK]**Per base sequence quality, forward**

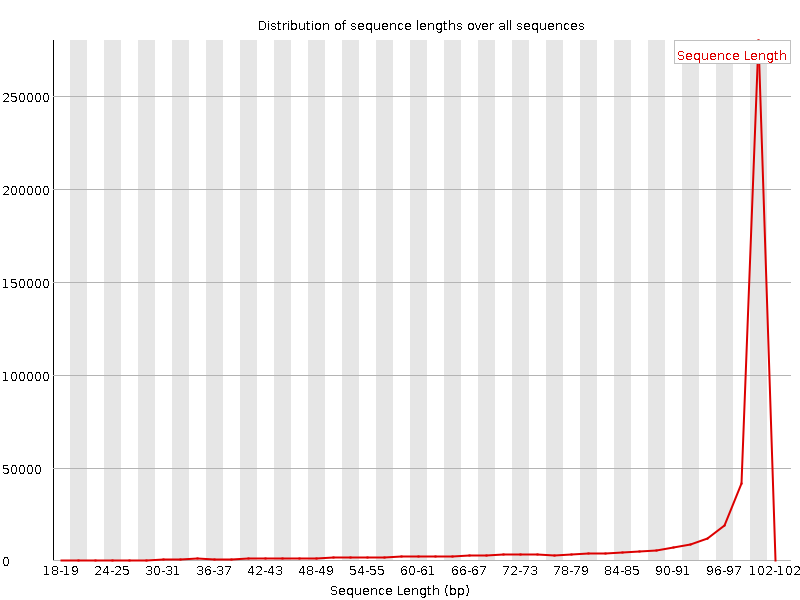


## [OK]**Per base sequence quality, reverse**

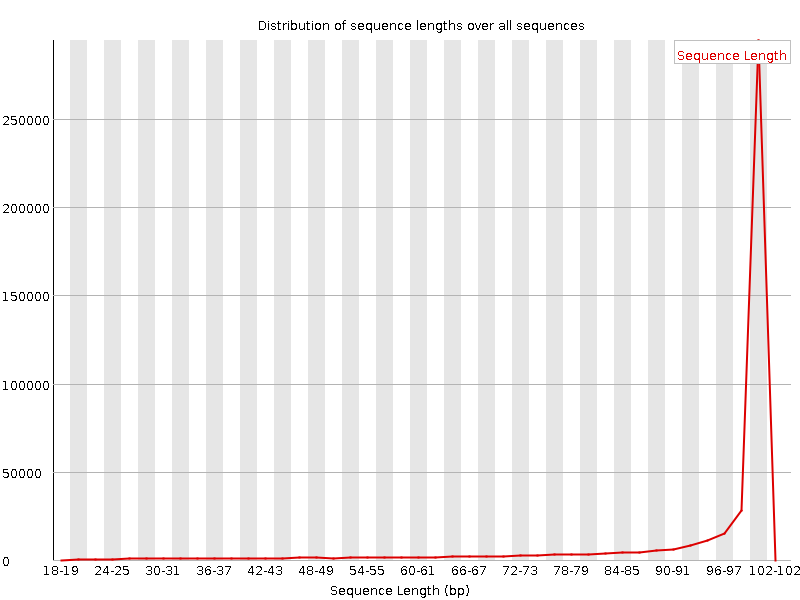


And we got another warning in the “Sequence Length Distribution” section because now our reads are not of the same length:

## [WARN]**Sequence Length Distribution, forward**



## [WARN]**Sequence Length Distribution, reverse**



With changing quality threshold to 30 in Trimmomatic per base quality of the reads became better but per tile quality assessment changed. We got failure in this section now because the score in area colored red (where something happens, for example, bubbles) is more than 5 less than the mean for that base across all tiles (which now have better scores).

1. **Alignment of the reads**

We used a bwa-mem tool to align the reads on the reference genome. This tool is an alignment program and it uses the Burrows-Wheeler Transform. Bwa-mem is optimized for long reads of 100 bp and more. At first step it looks for the maximum exact matches of seeds, and then it extends the seed using fitting or local alignment (depends on alignment score), to map the entire read.

Before indexing reference, file was unzipped:

**Command:** gunzip GCF\_000005845.2\_ASM584v2\_genomic.fna.gz

**Command:** gunzip GCF\_000005845.2\_ASM584v2\_genomic.gff.gz

Many programs that work with sequences use only indexed files so index was created for both files:

**Command:** bwa index GCF\_000005845.2\_ASM584v2\_genomic.fna

**Command:** bwa index GCF\_000005845.2\_ASM584v2\_genomic.gff

Alignment was performed using bwa mem.

**Command:** bwa mem GCF\_000005845.2\_ASM584v2\_genomic.fna.gz amp\_res\_1P.fq amp\_res\_2P.fq > alignment.sam

For the next step we needed to install samtools. This program works with SAM format. The SAM file then was compressed to a BAM file using samtools.

**Command:** sudo apt install samtools

**Command:** samtools view -S -b alignment.sam > alignment.bam

Next, we run samtools with flagstat option to get some statistics:

**Command:** samtools flagstat alignment.bam

**Result:**

892776 + 0 in total (QC-passed reads + QC-failed reads)

0 + 0 secondary

258 + 0 supplementary

0 + 0 duplicates

891649 + 0 mapped (99.87% : N/A)

892518 + 0 paired in sequencing

446259 + 0 read1

446259 + 0 read2

888554 + 0 properly paired (99.56% : N/A)

890412 + 0 with itself and mate mapped

979 + 0 singletons (0.11% : N/A)

0 + 0 with mate mapped to a different chr

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

We got a good result; 99.87% reads were mapped.

1. **Sort and index BAM files**BAM files need to be sorted and indexed.

**Command:** samtools sort alignment.bam -o alignment\_sorted

**Command:** samtools index alignment\_sorted.bam

1. **Variant calling**

To distinguish actual mutations from the sequencing errors we used the VarScan program. Firstly, we needed to make an intermediate file, it goes through each position and “piles up” the reads, tabulating the number of bases that match or don’t match the reference.

**Command:** samtools mpileup -f GCF\_000005845.2\_ASM584v2\_genomic.fna alignment\_sorted.bam > my.mpileup

We run VarScan with parameter --min-var-freq 0.80. This sets the minimum % of non-reference bases at a position required to call it a mutation in the sample.

**Command:** java -jar VarScan.v2.4.0.jar mpileup2snp my.mpileup --min-var-freq 0.80 --variants --output-vcf 1 > VarScan\_results.vcf

1. **Visualizing the data**

To visualize our data, we downloaded the IGV browser from the website. In the menu we selected “Genomes”,” Load Genome from File” and then for our annotation and alignment we selected “File”, “Open from file” to download the data into the IGV browser. The command below runs the IGV browser from the command line.

**Command:** java --module-path=lib -Xmx4g @igv.args --module=org.igv/org.broad.igv.ui.Main

1. **Analyzing the results**

After variant calling, we obtained 5 SNPs. Description of these SNPs is shown below:

1. GCC (A, alanine, +) > GGC (G, glycine, +)  
   Position: 93043  
   Gene: ftsI, peptidoglycan DD-transpeptidase
2. CAG (Q, glutamine, -) > CTG(L, leucine, -)  
   Position: 482698  
   Gene: acrB, multi-drug efflux pump RND permease
3. TTT (noncoding, -) > TCT (noncoding, -)

Position: 852762  
Gene: rybA, small RNA

1. GTA (V, valine, -) > GGT (G, glycine, -)  
   Position: 3535147

Gene: envZ, sensory histidine kinase

1. GCC (A, alanine, -) > GCA (A, alanine, -)  
   Position: 4390754

Gene: rsgA, ribosome small subunit-dependent GTPase A