**1. Quality Assessment of sequencing data**

Using FastQC software to evaluate the quality of the raw sequencing data.

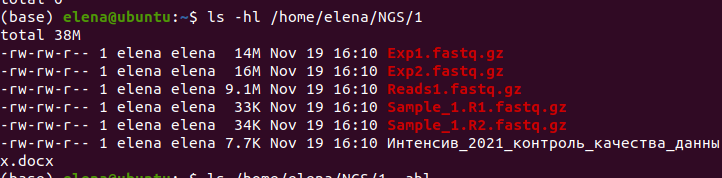
**pwd**

**mkdir /home/elena**

**mkdir /home/elena/NGS**

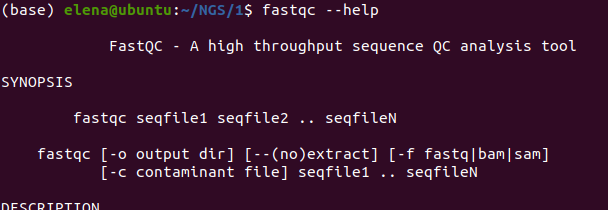
**mkdir /home/elena/NGS/1**

**ls -hl /home/elena/NGS/1**



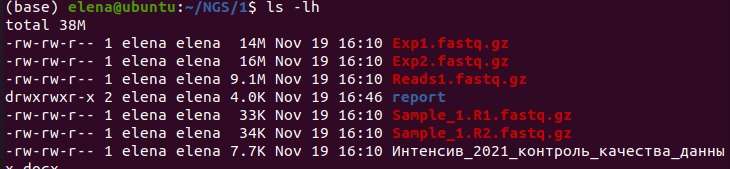


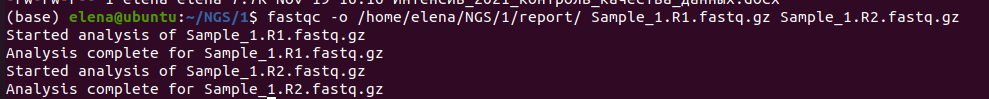
Checking whether the program is clearly visible



The output folder must exist. It must be created before running the program.

**mkdir report**





If done on the virtual machine, copy the results

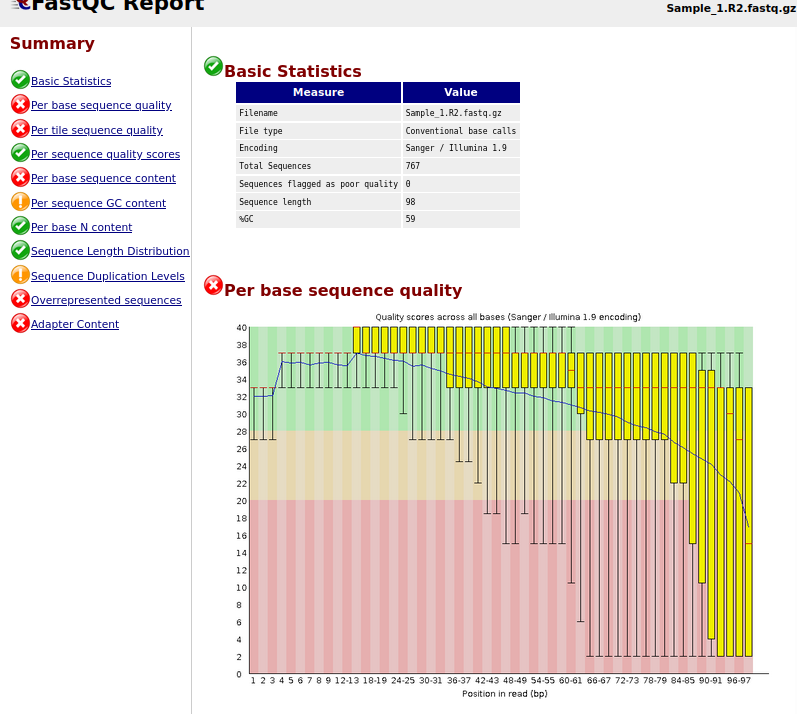


Starting cleaning up reads from adapters and low-quality sequences using Trimmomatic.

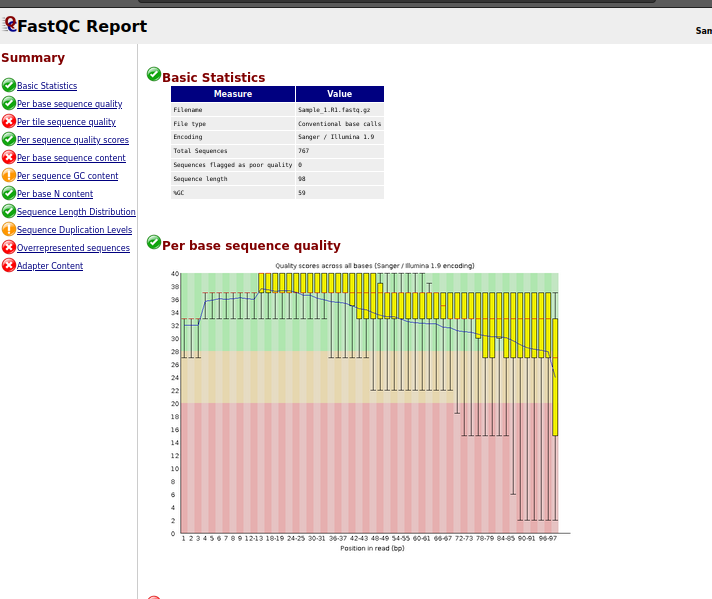
**java -jar /home/elena/Trimmomatic-0.38/trimmomatic-0.38.jar PE -phred33 Sample\_1.R1.fastq.gz Sample\_1.R2.fastq.gz Sample\_1.R1.paired.fastq.gz Sample\_1.R1.unpaired.fastq.gz Sample\_1.R2.paired.fastq.gz Sample\_1.R2.unpaired.fastq.gz ILLUMINACLIP:/home/elena/Trimmomatic-0.38/adapters/NexteraPE-PE.fa:2:30:10 LEADING:10 TRAILING:10 SLIDINGWINDOW:4:10 MINLEN:50**

Evaluating data quality after trimming using FastQC

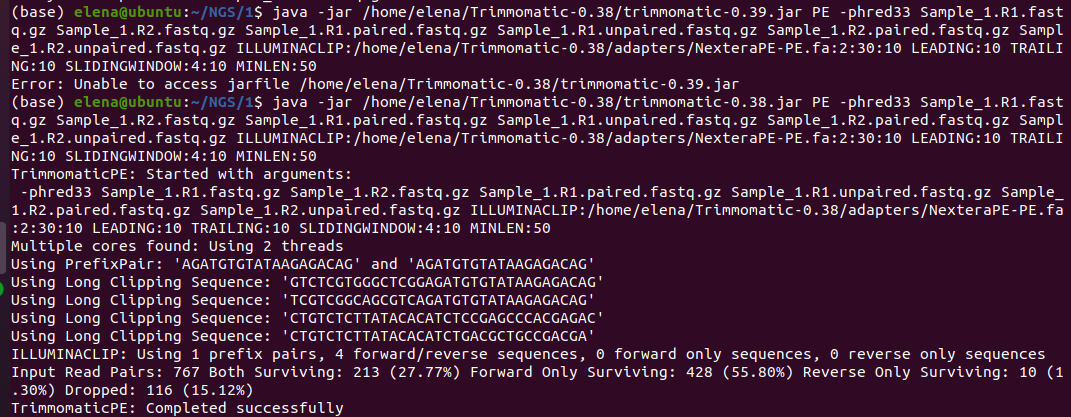
Before



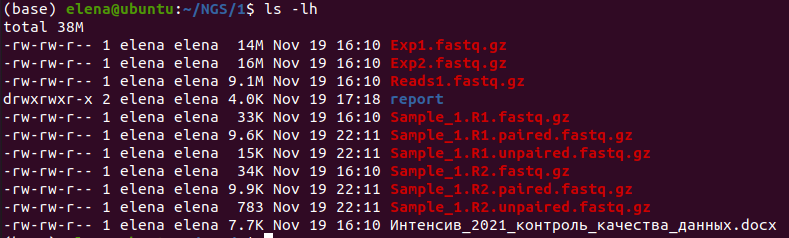
After



The program often shows errors due to a small genome.



**ls -lh**



2. **Reference alignment, calling options**

Antibiotic resistance of microorganisms is a significant concern in the treatment of bacterial infections. Evolution is the main algorithm for finding a way to survive, and bacteria can often resist a particular antibiotic. Therefore, it is vital for scientists and doctors to know the exact resistance mechanism in a bacterial strain. This knowledge can help develop new, better drugs or even help doctors decide which alternative antibiotics to use if a patient is not treated.

This task will work with real-life sequencing data for an *E. coli* strain resistant to the antibiotic ampicillin.

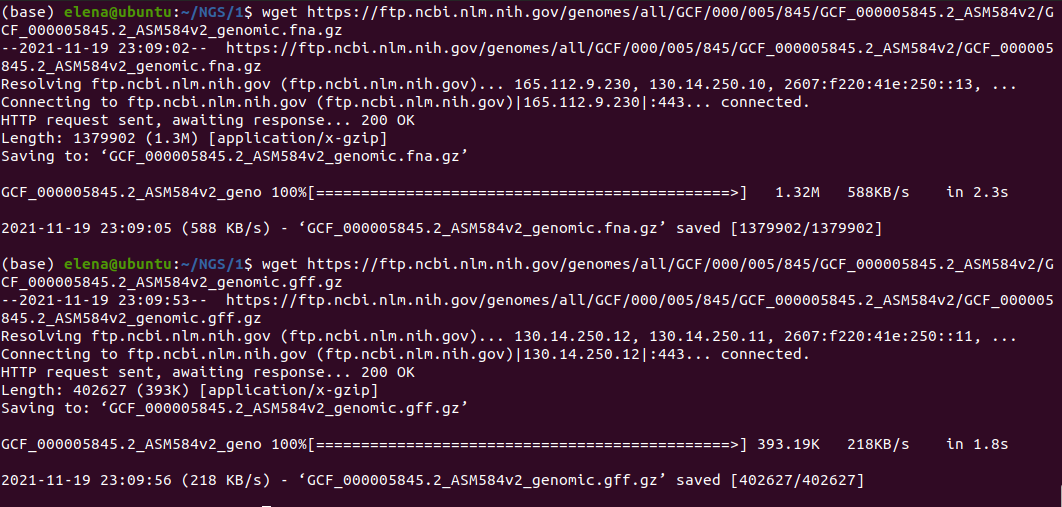
The purpose is to analyze this sequencing data and find the mutations responsible for making *E. coli* antibiotic-resistant. In addition, we will discover in which genes the mutations have occurred and determine the emerging resistance mechanism.

2.1. **Downloading reference genomes  
There are many programs (aligners) available for aligning reads. We will be using the bwa program, which uses the Burrows-Wheeler Transform (BWT).**

Reference sequence of the original (non-resistant) *E. coli* strain.  
This is the K-12 MG1655 strain, a well-known model organism. More information on this genome can be found at <http://www.ncbi.nlm.nih.gov/genome/167>).

Sequence (.fna) and annotation file (.gff).  
To download files from a link using the command line, you can use the wget command:  
**wget** [**https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF\_000005845.2\_ASM584v2/GCF\_000005845.2\_ASM584v2\_genomic.fna.gz**](https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM584v2_genomic.fna.gz)

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



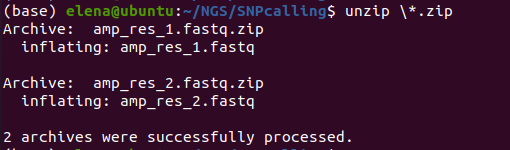
**grep ">" GCF\_000005845.2\_ASM584v2\_genomic.fna**



**less GCF\_000005845.2\_ASM584v2\_genomic.gff**

Unpack Illumina sequencing raw ampicillin resistant strain

**unzip \\*.zip**   
(not unzip \*.zip, \ literally removes the character from the name \*)



**2. 2. Alignment to reference sequence**

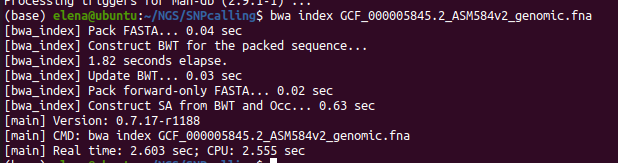
To get some meaningful information from our raw data, we will match it to our reference sequence (already assembled by the genome of a typical, unstable E. coli). In the course of "mapping" for each read, its optimal local alignment to the reference is found. The read is matched to a region of the reference to which it aligns well (i.e., above a particular threshold value) - this indicates that the read is likely to correspond to that part of the E. coli genome.  
  
There are many programs (aligners) available for aligning reads. Today we will be using the bwa program, which uses the Burrows-Wheeler Transform (BWT).

2.1. **Indexing the reference file**

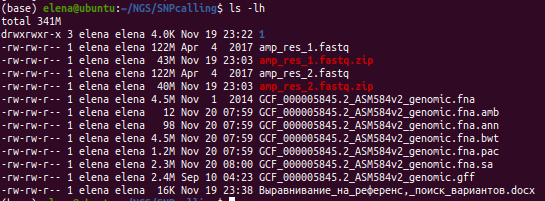
First, we have to index the reference genome. There is a special command for this in bowtie2:

**sudo apt install bwa**

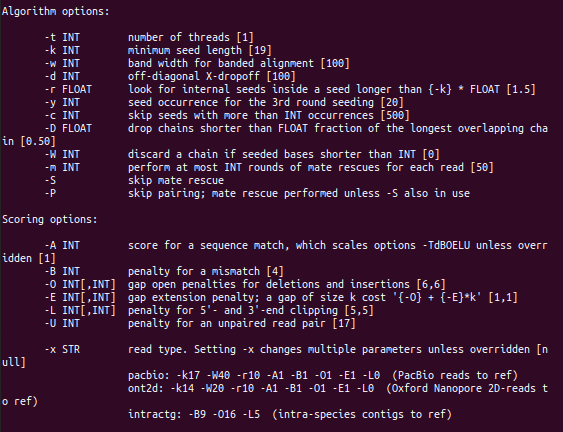
**bwa index GCF\_000005845.2\_ASM584v2\_genomic.fna**

****

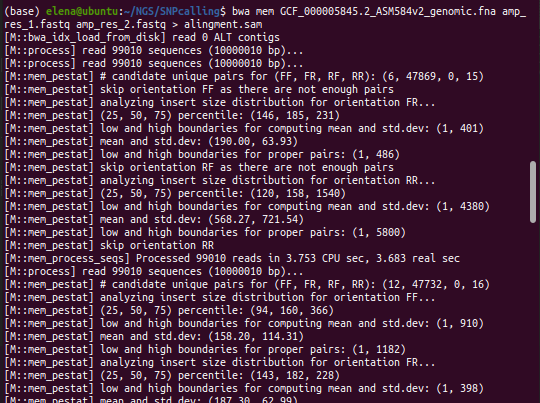
**Ls -lh**



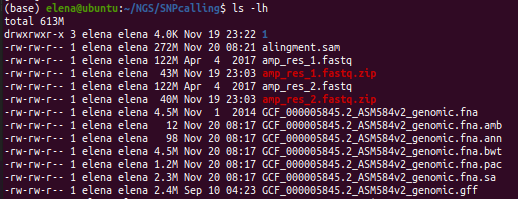
**bwa mem**



2.3. **Read alignment**Align the reads using the newly created index.  
The ">" symbol allows you to redirect the data stream. The redirection takes the previous command's output and places it in a new file specified after the '>.'  
Alignment may take a few minutes  
BWA outputs data in "SAM" format.  
  
**bwa mem GCF\_000005845.2\_ASM584v2\_genomic.fna amp\_res\_1.fastq amp\_res\_2.fastq > alingment.sam**



**Ls -lh**

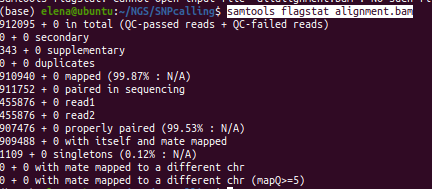


Not all reads will be successfully aligned to the reference. The SAM file contains all reads regardless of whether they were aligned successfully or not. For further analysis, we must know how much of our reads are aligned. If there is a lot of unaligned data, it could indicate that our DNA was contaminated by another way or something went wrong with the sequence.

2.3. **Compress SAM file**  
sudo apt install samtools

First, we need to compress and sort the SAM file using the commands below. A compressed SAM file is called a BAM (Binary Alignment Map) file. We can convert a SAM file to a BAM file as follows:  
**samtools view -S -b alingment.sam > alignment.bam**

Get basic alignment options  
**samtools flagstat alignment.bam**



2.4. **Sorting and indexing the file**

To work with a BAM file, it must be sorted and indexed. This indexing is different from indexing the reference - we are not using BWT here. We want to access the positions in the BAM file quickly.

In the case of indexing a reference, we create an FM BWT index, giving quick access to all substrings and positions where they appear. When indexing a BAM file, having a file sorted by coordinates, you can very quickly pull out the reads from the desired positions without having to iterate over the entire file every time you search for a specific coordinate. This is required for some of the following commands we will be using.

Sort the BAM file by coordinates in the reference sequence:

**samtools sort alignment.bam -o alignment\_sorted.bam**  
  
Index the BAM file for faster search

**samtools index alignment\_sorted.bam**

Our data with IGV

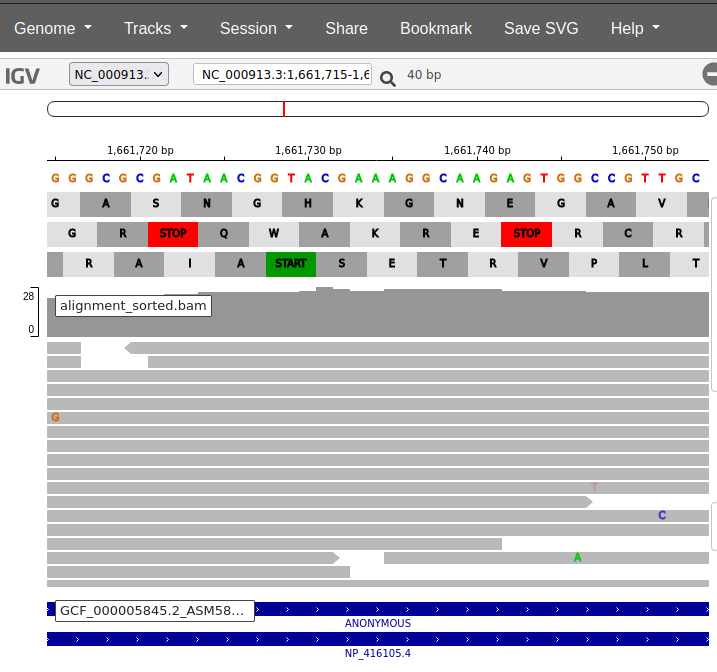
<https://igv.org/app/>

A genomic file will not be read without an index

Select both files .fna, .fai (genome)

bam, bai ( tracks)

Select annotation (tracks)

****

3. **Variant Calling**

Now the challenge is to look at our data, and for each position in the reference genome, see how many reads have a mutation at the same position - we want to distinguish between actual mutations and sequencing errors. The solution is to create an intermediate file type, so-called mpileup - in this file, for each reference position, the number of nucleotides that match or correspond to the reference is stored.

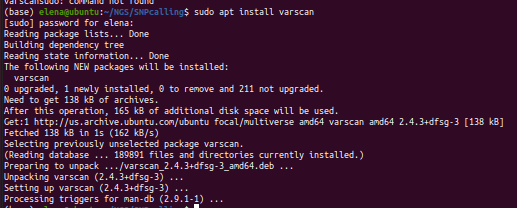
A sorted and indexed BAM file is required to get the mpileup file. Run the command below. This may take a few minutes.

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

****

We use a VarScan to select realistic options.

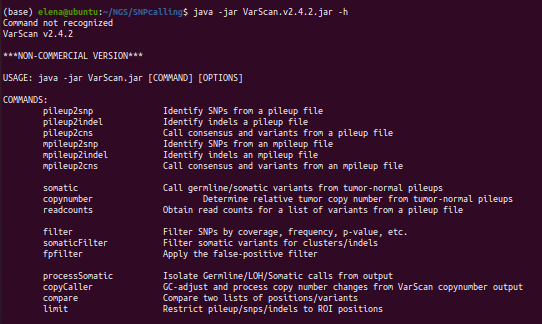
**sudo apt install varscan**



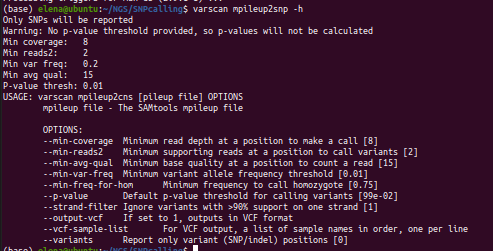
Or

**wget** [**https://github.com/dkoboldt/varscan/releases/download/2.4.2/VarScan.v2.4.2.jar**](https://github.com/dkoboldt/varscan/releases/download/2.4.2/VarScan.v2.4.2.jar)

**java -jar VarScan.v2.4.2.jar -h**



We are interested in the mpileup2snp command

**varscan mpileup2snp -h  
**

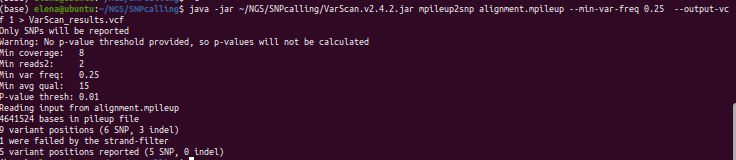
There are many ways to filter our data. For example, VarScan allows you to define your constraints for the selection of final options. Other programs use sophisticated statistical procedures to estimate the likelihood of natural mutations.

The only option we want to change today is --min-var-freq. It specifies the minimum% non-referential baseline required to call it a mutation in a sample.

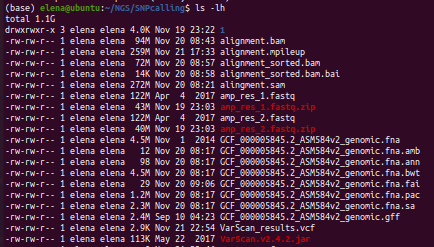
Run the program with the threshold that we think is best (replace our decimal number corresponding to the selected percentage, for example, 50% = 0.50).

**-variants** only displays positions with mutation-selection above our threshold.   
**--output-VCF 1** forces the output to be produced in a single data format called VCF (Variant Call Format).   
Therefore, the command may take several minutes.

**java -jar ~/NGS/SNPcalling/VarScan.v2.4.2.jar mpileup2snp alignment.mpileup --min-var-freq 0.25 --output-vcf 1 > VarScan\_results.vcf**

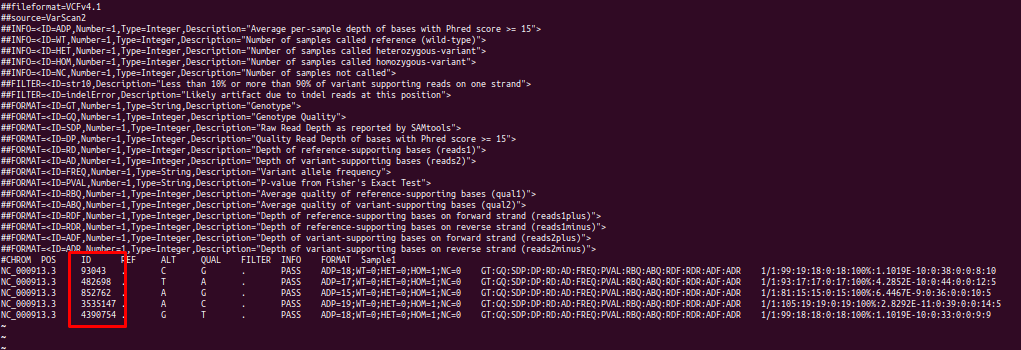
****

**ls -lh**

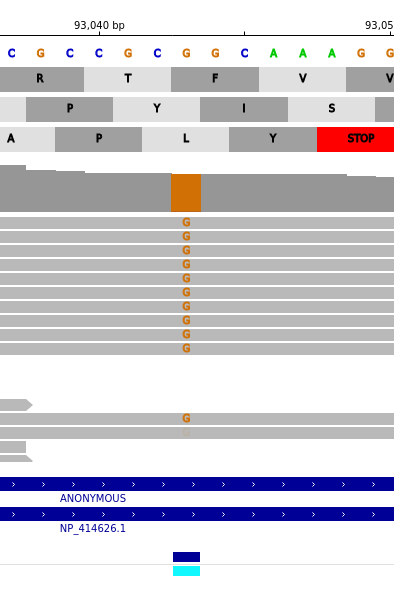
****

5 mutations visible

**less VarScan\_results.vcf**

****

1-st mutation on <https://igv.org/app/>

****

4. **Predicting the effect of mutation**

Where are these mutations found, and whether they change any host proteins (when different codons appear for the same amino acid occur)?

IGV Browser

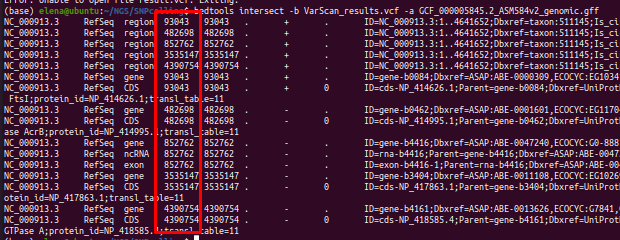
**sudo apt install bedtools**

Add two more "tracks" - a vcf file and an annotation in gff format.

It is possible to cross automatically the variants with annotation:

**bedtools intersect -b VarScan\_results.vcf -a GCF\_000005845.2\_ASM584v2\_genomic.gff**

Intersecting the mutation vcf file and the annotation gff file

****

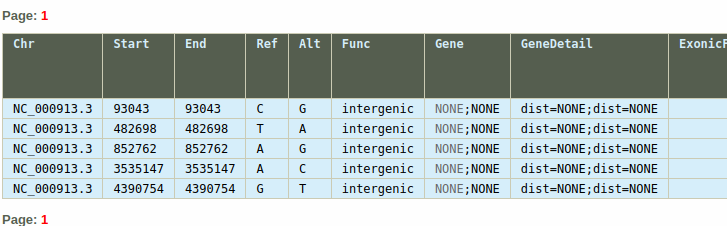
We study all mutations and find out about each of them: is it in a gene, is it a missense mutation (whether it changes the amino acid sequence), a nonsense mutation (leads to a frameshift or a stop codon), or a silent mutation (without replacing an amino acid ). For missense and nonsense mutations, find out what the gene is called.

Simply knowing these mutations will not help a doctor decide how to treat a patient.

Based on the found genes, an assumption about the mechanism of antibiotic resistance.

**Using wANNOVAR**

<http://wannovar.wglab.org/>



**Genome-wide association study**

Exome pre-processed sequencing data in VCF format.

with frequent variants ( > 5%).

Analyzing the associations using a quantitative trait (body mass index, BMI).

Phenotypic data is in the pheno.txt file,

genotypic data - in the gwas-prac.vcf.gz file (necessary to download its index along with the file).

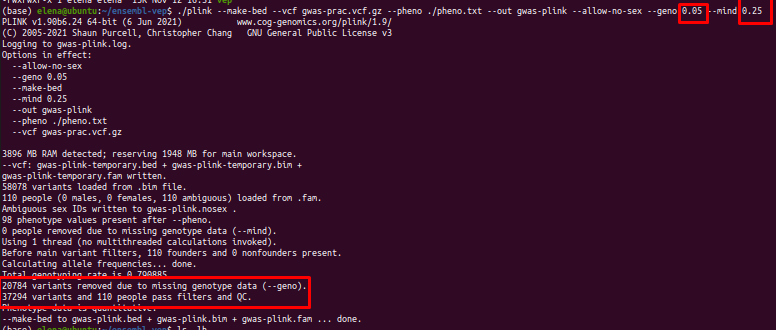
The files are in the plink folder.

Сonverting the data to the PLINK data format

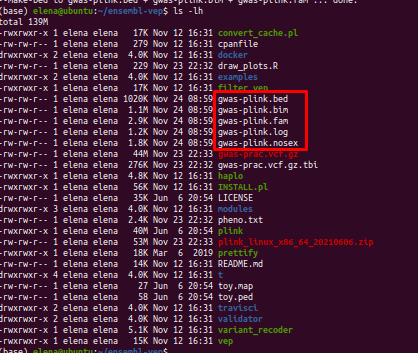
./plink --make-bed --vcf gwas-prac.vcf.gz --pheno ./pheno.txt --out gwas-plink --allow-no-sex --geno <> --mind <>

Specify the desired thresholds for the variant call rate and sample call rate as values of the 'geno' and 'mind' parameters (% of missing genotypes for a particular variant or sample)

**./plink --make-bed --vcf gwas-prac.vcf.gz --pheno ./pheno.txt --out gwas-plink --allow-no-sex --geno 0.05 --mind 0.25**

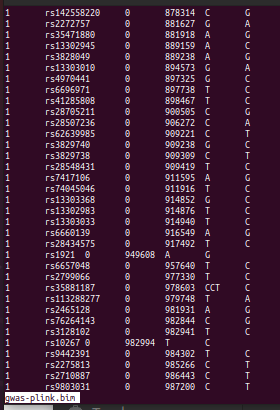


New files were created with different extensions  
Storing genetic information efficiently by compression thought program ‘plink’



General genetic information, a set of genotyped variants

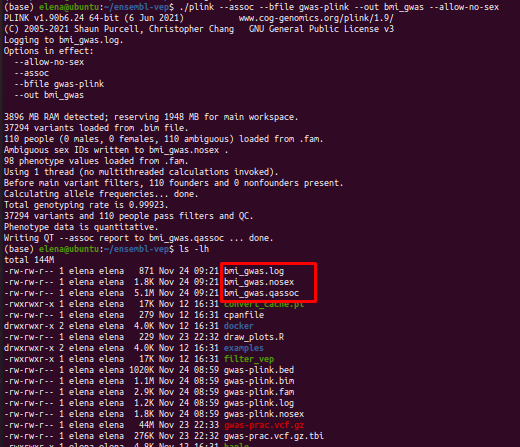




Gwas-plink.bed - binary file with information about each sample

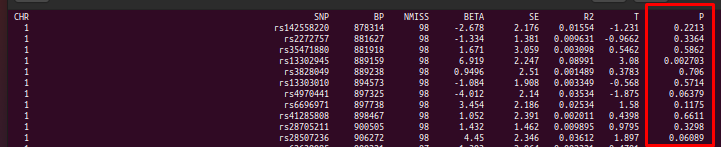
Analysis of associations

**./plink --assoc --bfile gwas-plink --out bmi\_gwas --allow-no-sex**

****

bmi\_gwas.qassoc - the main file for gwas





The key metric is the P or P-value, which is statistically calculated using linear regression.

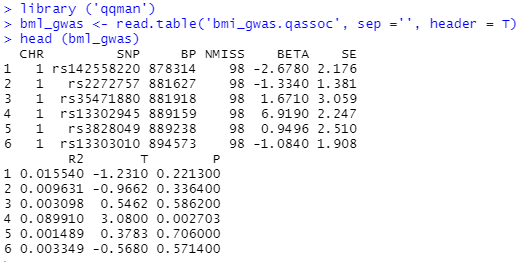
A file with the .qassoc extension containing the analysis results was received.

Basic graph from the draw\_plots.R file in RStudio.

> **library ('qqman')**

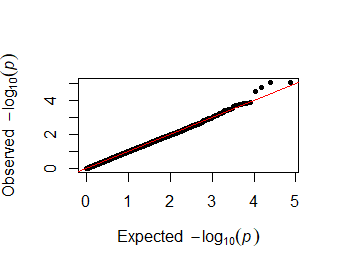
> **bml\_gwas <- read.table('bmi\_gwas.qassoc', sep ='', header = T)**

> **head (bml\_gwas)**



**qq(bml\_gwas$P)**





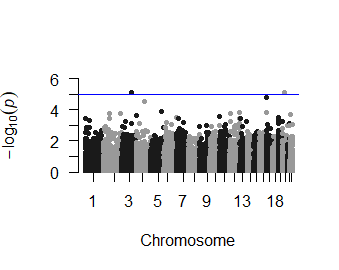
Getting the results of the analysis in the form of QQ and Manhattan graphs

The QQ plot shows matching if there is no effect and actual sample values.  
In our example, there are no population factors since all genomes are representatives of a homogeneous population. Quality control has done well

An upward deviation of the graph towards the end indicates a positive, pronounced signal. Thus, the data is associated with a feature more significantly than expected.

> **manhattan(na.omit(bml\_gwas))**





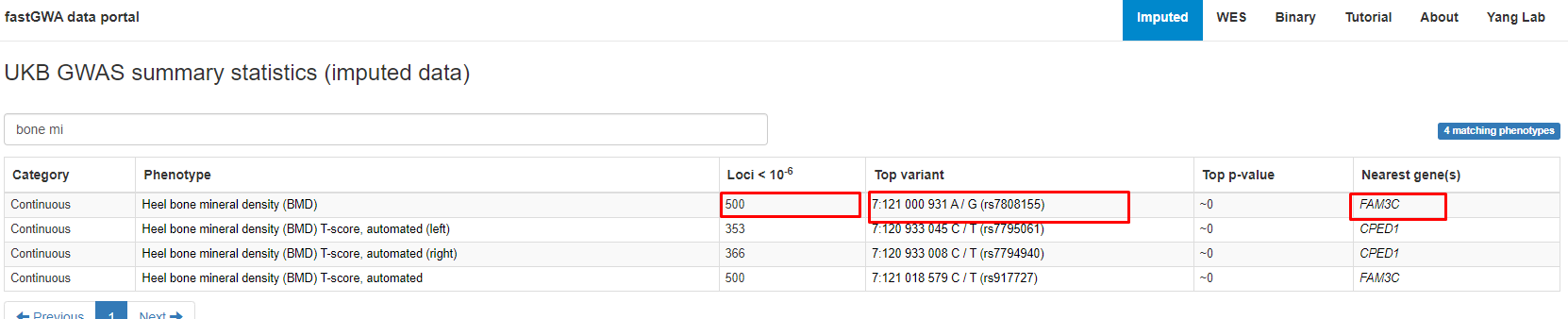
On this plot, we should see the bars that are located in separate regions of the genome. Unfortunately, there are no columns on our data, and it requires further analysis.

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

<https://yanglab.westlake.edu.cn/resources/ukb_fastgwa/imp/phenotypes>

Data on various phenotypes from British bio bank.

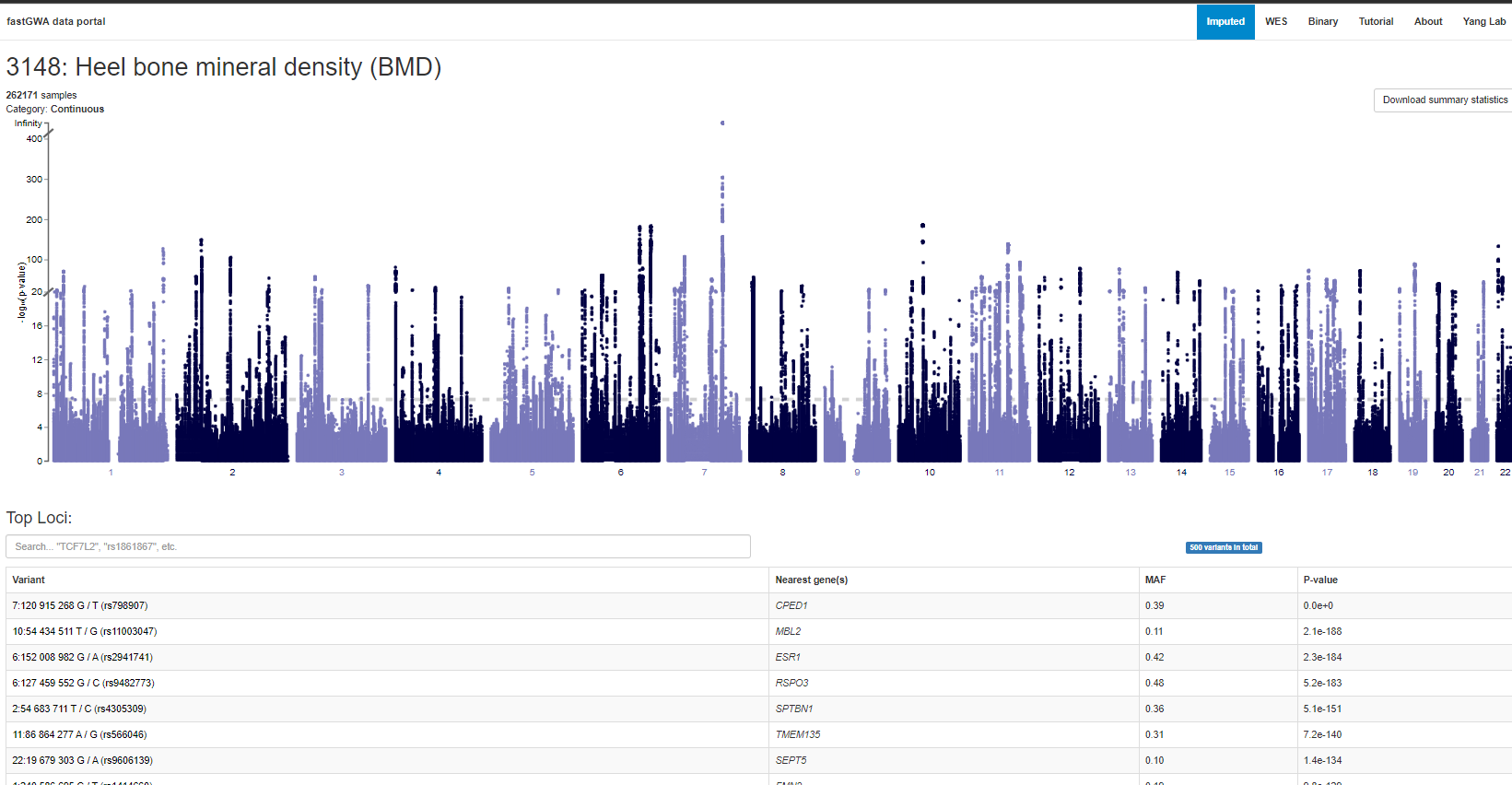
For example, bone mineral density.



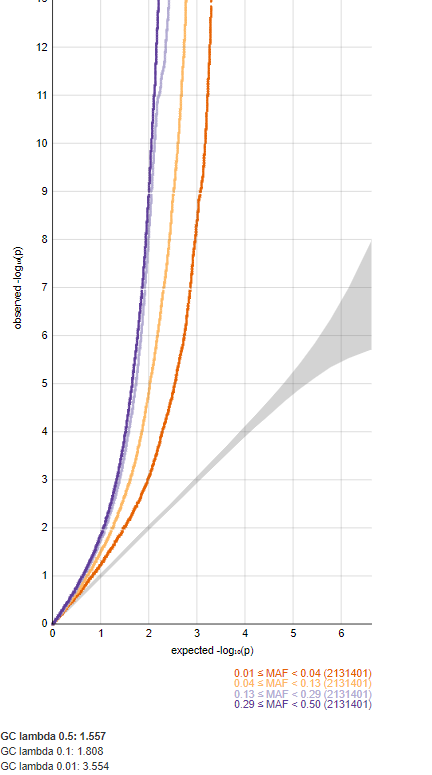
Shows the number of snips (several hundred) with a certain p-value 10^(- 6), the most significant variant and its p-value (almost 0), and the closest gene.

GWAS (*manhattan plot)* finds points that contribute a little to the development of a trait. There can be several hundred of them, which affect bone mineral density.

Each point can have a contribution of about 1-2%. And it isn't easy to conclude individual points, but the situation becomes more evident if we can see a complex plot.



*QQ plot*



If we take thrombosis (blood clot), we get a more typical situation when cutting off most values. We get 5-6 peaks that cross the threshold of significance. QQ plot is also more common, with a deviation in the second half of the schedule.

