Project №1: What causes antibiotic resistance?” Alignment to reference, variant calling

Description:

- Initial data:
  - reference genom of E.Coli without ampicillin resistance
  - Illumina sequencing reads from shotgun sequencing of an E. coli strain with ampicillin resistance
- Goal: find mutations responsible for ampicillin resistance

0. Programs used in the course of work

In [1]:
```
! fastqc --version
```

```
FastQC v0.11.9
```

In [3]:
```
! java -jar /usr/share/java/trimmomatic.jar -version
```

```
0.39
```

In [38]:
```
! bwa
```

```
Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.17-r1188
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index         index sequences in the FASTA format
         mem           BWA-MEM algorithm
         fastmap       identify super-maximal exact matches
         pemerge       merge overlapping paired ends (EXPERIMENTAL)
         aln           gapped/ungapped alignment
         samse         generate alignment (single ended)
         sampe         generate alignment (paired ended)
         bwasw         BWA-SW for long queries

         shm           manage indices in shared memory
         fa2pac        convert FASTA to PAC format
         pac2bwt       generate BWT from PAC
         pac2bwtgen    alternative algorithm for generating BWT
         bwtupdate     update .bwt to the new format
         bwt2sa        generate SA from BWT and Occ

Note: To use BWA, you need to first index the genome with `bwa index'.
      There are three alignment algorithms in BWA: `mem', `bwasw', and
      `aln/samse/sampe'. If you are not sure which to use, try `bwa mem'
      first. Please `man ./bwa.1' for the manual.
```

In [45]:
```
! samtools --version
```

```
samtools 1.10
Using htslib 1.10.2-3
Copyright (C) 2019 Genome Research Ltd.
```

In [55]:
```
! java -jar "/home/gressy/Programs/VarScan.v2.3.9.jar"
```

```
VarScan v2.3

USAGE: java -jar VarScan.jar [COMMAND] [OPTIONS]

COMMANDS:
        pileup2snp              Identify SNPs from a pileup file
        pileup2indel            Identify indels a pileup file
        pileup2cns              Call consensus and variants from a pileup file
        mpileup2snp             Identify SNPs from an mpileup file
        mpileup2indel           Identify indels an mpileup file
        mpileup2cns             Call consensus and variants from an mpileup file

        somatic                 Call germline/somatic variants from tumor-normal pileups
        copynumber                      Determine relative tumor copy number from tumor-normal pileups
        readcounts              Obtain read counts for a list of variants from a pileup file

        filter                  Filter SNPs by coverage, frequency, p-value, etc.
        somaticFilter           Filter somatic variants for clusters/indels
        fpfilter                Apply the false-positive filter

        processSomatic          Isolate Germline/LOH/Somatic calls from output
        copyCaller              GC-adjust and process copy number changes from VarScan copynumber output
        compare                 Compare two lists of positions/variants
        limit                   Restrict pileup/snps/indels to ROI positions
```

1. Getting data

- Download reference sequence of the E. coli strain (without ampicillin resistance)

In [3]:
```
!wget https://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia_coli/reference/GCA_000005845.2_ASM584v2/GCA_000005845.2_ASM584v2_genomic.fna.gz
!gzip -d GCA_000005845.2_ASM584v2_genomic.fna.gz
```

```
--2021-10-14 20:52:29--  https://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia_coli/reference/GCA_000005845.2_ASM584v2/GCA_000005845.2_ASM58
4v2_genomic.fna.gz
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.7, 130.14.250.10, 2607:f220:41e:250::7, ...
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.7|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1379898 (1,3M) [application/x-gzip]
Saving to: ‘GCA_000005845.2_ASM584v2_genomic.fna.gz’

GCA_000005845.2_ASM 100%[===================>]   1,32M   119KB/s    in 14s

2021-10-14 20:52:45 (93,1 KB/s) - ‘GCA_000005845.2_ASM584v2_genomic.fna.gz’ saved [1379898/1379898]
```

- Download reference sequence annotation

In [4]:
```
!wget https://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia_coli/reference/GCA_000005845.2_ASM584v2/GCA_000005845.2_ASM584v2_genomic.gff.gz
!gzip -d GCA_000005845.2_ASM584v2_genomic.gff.gz
```

```
--2021-10-14 21:03:58--  https://ftp.ncbi.nlm.nih.gov/genomes/genbank/bacteria/Escherichia_coli/reference/GCA_000005845.2_ASM584v2/GCA_000005845.2_ASM58
4v2_genomic.gff.gz
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 165.112.9.228, 165.112.9.229, 2607:f220:41e:250::13, ...
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|165.112.9.228|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 380579 (372K) [application/x-gzip]
Saving to: ‘GCA_000005845.2_ASM584v2_genomic.gff.gz’

GCA_000005845.2_ASM 100%[===================>] 371,66K   254KB/s    in 1,5s

2021-10-14 21:04:00 (254 KB/s) - ‘GCA_000005845.2_ASM584v2_genomic.gff.gz’ saved [380579/380579]
```

- Download Illumina sequencing reads with ampicillin resistance

  Notification: you need to look at the file number at figshare.com to download these files

In [1]:
```
#forward reads - 1
! wget -O amp_res_1.fastq.gz https://figshare.com/ndownloader/files/23769689
! gzip -d amp_res_1.fastq.gz
```

```
--2021-10-14 21:28:27--  https://figshare.com/ndownloader/files/23769689
Resolving figshare.com (figshare.com)... 52.19.36.181, 63.32.80.207, 2a05:d018:1f4:d000:f5f7:3cea:611b:d23d, ...
Connecting to figshare.com (figshare.com)|52.19.36.181|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://s3-eu-west-1.amazonaws.com/pfigshare-u-files/23769689/amp_res_1.fastq.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=10&X-Amz-Signe
dHeaders=host&X-Amz-Signature=e7e6488a34d94285337a37c9e4e76946367a5b19cec9bb0a143a0f5925a5847e&X-Amz-Date=20211014T182828Z&X-Amz-Credential=AKIAIYCQYOYV
5JSSROOA/20211014/eu-west-1/s3/aws4_request [following]
--2021-10-14 21:28:28--  https://s3-eu-west-1.amazonaws.com/pfigshare-u-files/23769689/amp_res_1.fastq.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires
=10&X-Amz-SignedHeaders=host&X-Amz-Signature=e7e6488a34d94285337a37c9e4e76946367a5b19cec9bb0a143a0f5925a5847e&X-Amz-Date=20211014T182828Z&X-Amz-Credenti
al=AKIAIYCQYOYV5JSSROOA/20211014/eu-west-1/s3/aws4_request
Resolving s3-eu-west-1.amazonaws.com (s3-eu-west-1.amazonaws.com)... 52.218.0.35
Connecting to s3-eu-west-1.amazonaws.com (s3-eu-west-1.amazonaws.com)|52.218.0.35|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 44390940 (42M) [application/gzip]
Saving to: ‘amp_res_1.fastq.gz’

amp_res_1.fastq.gz  100%[===================>]  42,33M   176KB/s    in 3m 58s

2021-10-14 21:32:26 (182 KB/s) - ‘amp_res_1.fastq.gz’ saved [44390940/44390940]
```

In [2]:
```
#reversed reads - 2
! wget -O amp_res_2.fastq.gz https://figshare.com/ndownloader/files/23769692
! gzip -d amp_res_2.fastq.gz
```

```
--2021-10-14 21:32:40--  https://figshare.com/ndownloader/files/23769692
Resolving figshare.com (figshare.com)... 52.19.36.181, 63.32.80.207, 2a05:d018:1f4:d000:f5f7:3cea:611b:d23d, ...
Connecting to figshare.com (figshare.com)|52.19.36.181|:443... connected.
HTTP request sent, awaiting response... 302 Found
```

```
Location: https://s3-eu-west-1.amazonaws.com/pfigshare-u-files/23769692/amp_res_2.fastq.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=10&X-Amz-Signe
dHeaders=host&X-Amz-Signature=7c0eaa657f39d68c12200918c171305b4c92d4d937f6ec057e6a8ca21b4710a2&X-Amz-Date=20211014T183240Z&X-Amz-Credential=AKIAIYCQYOYV
5JSSROOA/20211014/eu-west-1/s3/aws4_request [following]
--2021-10-14 21:32:40--  https://s3-eu-west-1.amazonaws.com/pfigshare-u-files/23769692/amp_res_2.fastq.gz?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires
=10&X-Amz-SignedHeaders=host&X-Amz-Signature=7c0eaa657f39d68c12200918c171305b4c92d4d937f6ec057e6a8ca21b4710a2&X-Amz-Date=20211014T183240Z&X-Amz-Credenti
al=AKIAIYCQYOYV5JSSROOA/20211014/eu-west-1/s3/aws4_request
Resolving s3-eu-west-1.amazonaws.com (s3-eu-west-1.amazonaws.com)... 52.218.57.211
Connecting to s3-eu-west-1.amazonaws.com (s3-eu-west-1.amazonaws.com)|52.218.57.211|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 44375379 (42M) [application/gzip]
Saving to: ‘amp_res_2.fastq.gz’

amp_res_2.fastq.gz  100%[===================>]  42,32M   151KB/s    in 4m 51s

2021-10-14 21:37:32 (149 KB/s) - ‘amp_res_2.fastq.gz’ saved [44375379/44375379]
```

Results of section:

- GCA_000005845.2_ASM584v2_genomic.fna, GCA_000005845.2_ASM584v2_genomic.gff - reference sequence of the parental (non-evolved, antibiotic-resistant) strain of E. coli and annotation
- amp_res_1.fastqc, amp_res_2.fastqc - raw Illumina sequencing data obtained from sequencing of the E. coli strain resistant to the antibiotic ampicillin (1 and 2 refer to forward and reverse)

2. Inspecting raw sequencing data manually

The goal of the section: to understand the structure of the fasta file

In [3]:
```
#number of lines in reads fasta file
! wc -l amp_res_1.fastq
```

```
1823504 amp_res_1.fastq
```

So, there are 1823504 lines in this file

In [4]:
```
#see first 20 lines of file
!head -20 amp_res_1.fastq
```

```
@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#######
@SRR1363257.46 GWZHISEQ01:153:C1W31ACXX:5:1101:19721:2155 length=101
GTATGAGGTTTTGCTGCATTCTCTGNGCGAATATTAACTCCNTNNNNNTTATAGTTCAAAGCAAGTACCTGTCTCTTATACACATCTCCGAGCCCACGAGC
+
@@<?=D?D==?<AFGDF+AIHEACH#22<:?E8??:9??GG#0#####000;CF=C)4.==CA@@@)=7?C7?E37;3@>;;(.;>AB##############
@SRR1363257.77 GWZHISEQ01:153:C1W31ACXX:5:1101:5069:2307 length=101
GCTTCTCTTAACTGAGGTCACCATCATGCCGTTAAGTCCCTACCTCTCTTTTGCCGGTAACTGTTCCGCCGCGATTGCCTTTTATCTGTCTCTTATACACC
+
??<DBD;4C2=<BB>:AC;<CF<CE@FE9@E1C@891CD*9:?:3D@DD4?D<DD:0;@A=AEIDDA##################################
@SRR1363257.78 GWZHISEQ01:153:C1W31ACXX:5:1101:5178:2440 length=101
GCATAAGGACGATCGCTCCAGAGTAAAATAAATACGCGCATGTGATACTCACAATACCAATGGTGAAGTTACGGGACTTAAACAAACTGAGATCAAGAATC
+
CCCFFFFFHHHHHJJJJJJJJJJFFHIJJJJJJJJJJJJJJJJJJJJJJJIJHHHHHHFDEDF;AEEEEEEDDDDDBBACDDDCDDDDCCDDDDDDCCDC3
```

```
@SRR1363257.96 GWZHISEQ01:153:C1W31ACXX:5:1101:6707:2460 length=101
TCATTAAGCCGTGGTGGATGTGCCATAGCGCACCGCAAAGTTAAGAAACCGAATATTGGGTTTAGTCTTGTTTCATAATTGTTGCAATGAAACGCGGTGAA
+
CCCFFFFFHHHHHJHIIJIIIIJJJJJJGIJJJJJIJJIIGHJJJJJIIJJDHFFFFFEDACDDDCDDDDCCDDECACCDCCCDACDDDDCCDDDDDBD@A
```

We see that there are 4 lines per read:

1. identifiers and information about this read
2. read sequence
3. + (sometimes this line have the identifier and info repeated)
4. quality string (ASCII characters encode the quality score for each base). The quality score ranges from 0 to about 40 (the higher the number, the greater the accuracy of the base call)

   To get the actual quality score, you need to figure out the value of the symbol, then subtract 33 (this is called ‘sanger’ scaling type, because it's the same scaling that people use with traditional sanger sequencing). With some older Illumina data (pre version 1.8), you subtract other numbers, like 64 - this is called Phred33 (sanger) or Phred64 scale, respectively.

Therefore, the number of reads:

In [5]:
```
1823504/4
```

Out[5]:
```
455876.0
```

3. Inspecting raw sequencing data with fastqc

The goal of this section: view fasta files information (like quality) using special program Fastqc

Fastqc is a simple program for analyzing fastq statistics

In [8]:
```
# -o .  to output files to the current directory
! fastqc -o . /home/gressy/Current\ work/Bioinf\ 2021-2022/Projects/Project1/amp_res_1.fastq /home/gressy/Current\ work/Bioinf\ 2021-2022/Projects/Proje
```

```
Started analysis of amp_res_1.fastq
Approx 5% complete for amp_res_1.fastq
Approx 10% complete for amp_res_1.fastq
Approx 15% complete for amp_res_1.fastq
Approx 20% complete for amp_res_1.fastq
Approx 25% complete for amp_res_1.fastq
Approx 30% complete for amp_res_1.fastq
Approx 35% complete for amp_res_1.fastq
Approx 40% complete for amp_res_1.fastq
Approx 45% complete for amp_res_1.fastq
Approx 50% complete for amp_res_1.fastq
Approx 55% complete for amp_res_1.fastq
Approx 60% complete for amp_res_1.fastq
Approx 65% complete for amp_res_1.fastq
Approx 70% complete for amp_res_1.fastq
Approx 75% complete for amp_res_1.fastq
Approx 80% complete for amp_res_1.fastq
Approx 85% complete for amp_res_1.fastq
Approx 90% complete for amp_res_1.fastq
Approx 95% complete for amp_res_1.fastq
Analysis complete for amp_res_1.fastq
Started analysis of amp_res_2.fastq
```

```
Approx 5% complete for amp_res_2.fastq
Approx 10% complete for amp_res_2.fastq
Approx 15% complete for amp_res_2.fastq
Approx 20% complete for amp_res_2.fastq
Approx 25% complete for amp_res_2.fastq
Approx 30% complete for amp_res_2.fastq
Approx 35% complete for amp_res_2.fastq
Approx 40% complete for amp_res_2.fastq
Approx 45% complete for amp_res_2.fastq
Approx 50% complete for amp_res_2.fastq
Approx 55% complete for amp_res_2.fastq
Approx 60% complete for amp_res_2.fastq
Approx 65% complete for amp_res_2.fastq
Approx 70% complete for amp_res_2.fastq
Approx 75% complete for amp_res_2.fastq
Approx 80% complete for amp_res_2.fastq
Approx 85% complete for amp_res_2.fastq
Approx 90% complete for amp_res_2.fastq
Approx 95% complete for amp_res_2.fastq
Analysis complete for amp_res_2.fastq
```

The result of previous program is 2 html files with lot's of information on reads. First of all, we are interested in quality control: colored circles on the lest of the html page:

- green circle - normal
- yellow circle - slightly abnormal
- red circle - very unusual

There is more detailed information about Fastqc here .

FastQC Report

Thu 14 Oct 2021
amp_res_1.fastq

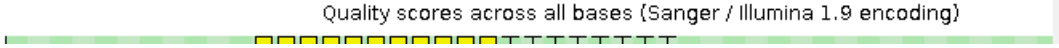
Summary

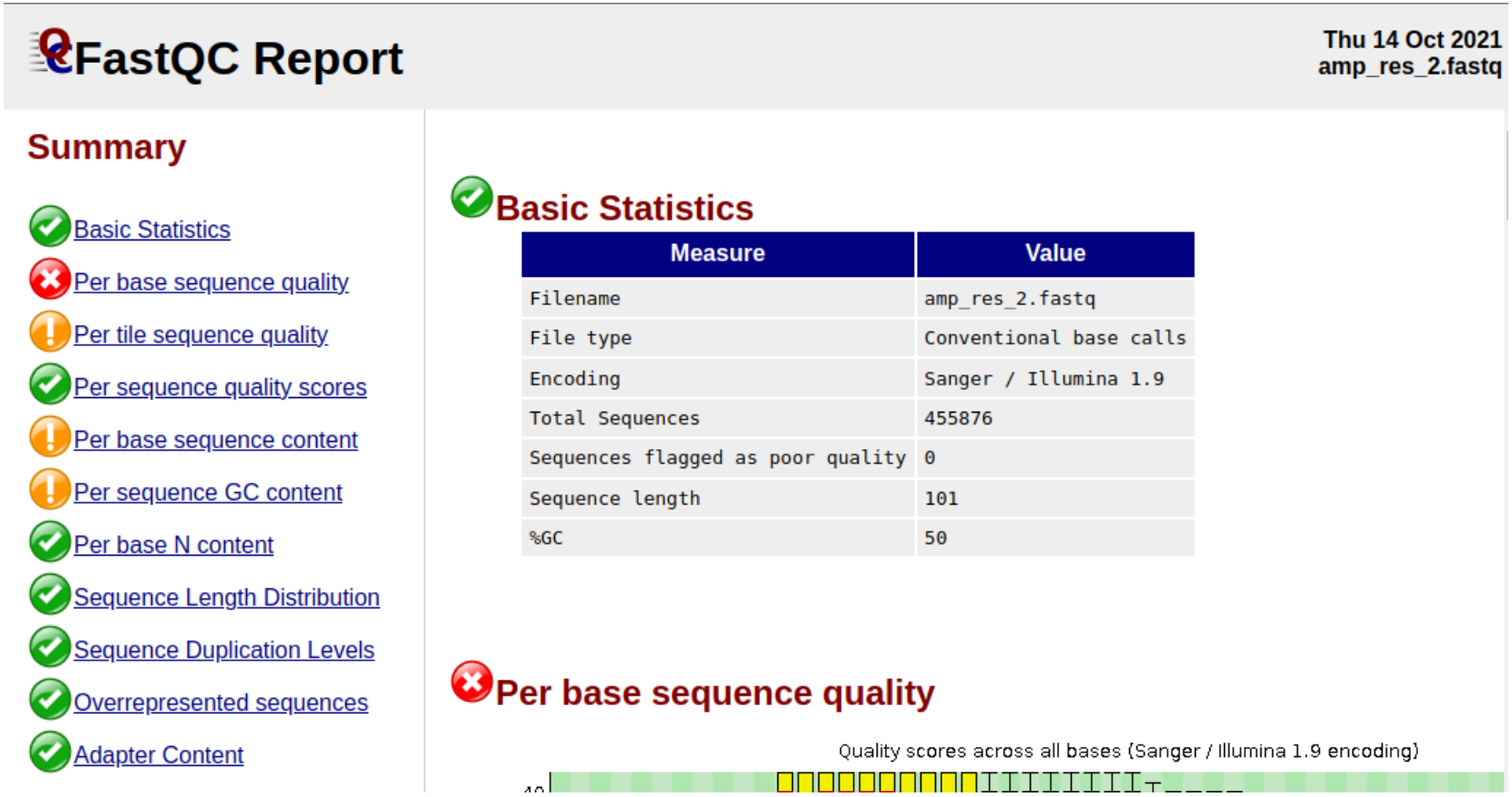
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	amp_res_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	455876
Sequences flagged as poor quality	0
Sequence length	101
%GC	50

Per base sequence quality





FastQC Report

Thu 14 Oct 2021
amp_res_2.fastq

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	amp_res_2.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	455876
Sequences flagged as poor quality	0
Sequence length	101
%GC	50

Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)

So, per base sequence quality is bad both for forward and reversed reads. We can also see bad per tile sequence quality for forward reads.

4. Filtering the reads

Previous section shows bad quality results for reads. That is why it is nesessary to improve quality deleting bad reads.

The app will be used for this: Trimmomatic Tutorial

java -jar PE [-threads] ...

- PE - paired mode (there is also single mode)
- [-threads \< threads] - количество используемых потоков для повышения производительности на многоядерных компьютерах. Если не указан, он будет выбран автоматически.
- [-phred33 | -phred64] - Starting in Illumina 1.8, the quality scores have basically returned to the use of the Sanger format (Phred+33).

- LEADING: x - Remove leading (с начала) low quality or N bases (below quantity x)
- TRAILING:x - Remove trailing (с конца) low quality or N bases (below quantity x)
- SLIDINGWINDOW:4:15 - Scan the read with a 4-base wide sliding window, cutting when the average quality per base drops below 15

- MINLEN: x - Drop reads which are less than 36 bases long after these steps

Let's try to run trimmomatic in PE with different quality score (q = 20, 30):

- Cut bases off the start of a read if quality below q
- Cut bases off the end of a read if quality below q
- Trim reads using a sliding window approach, with window size 10 and average quality within the window q.
- Drop the read if it is below length q.

The output of these runs are 4 files

- amp_1P_20.fastq, amp_2P_20.fastq - trimmed forward and reverse reads, where both read passed the trimming filter
- amp_1U_20.fastq, amp_2U_20.fastq - only one of forward nd reverse reads passed the trimming filter

q=20

In [4]:
```
! java -jar /usr/share/java/trimmomatic.jar PE -phred33 amp_res_1.fastq amp_res_2.fastq amp_1P_20.fastq amp_1U_20.fastq amp_2P_20.fastq amp_2U_20.fastq
```

```
TrimmomaticPE: Started with arguments:
 -phred33 amp_res_1.fastq amp_res_2.fastq amp_1P_20.fastq amp_1U_20.fastq amp_2P_20.fastq amp_2U_20.fastq LEADING:20 TRAILING:20 SLIDINGWINDOW:10:20 MIN
LEN:20
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
```

In [7]:
```
! wc -l amp_1P_20.fastq
```

```
1785036 amp_1P_20.fastq
```

In [8]:
```
#percent of reads where both read passed the trimming filter
1785036/1823504 * 100
```

Out[8]:
```
97.89043511832165
```

In [9]:
```
! wc -l amp_1U_20.fastq
```

```
36864 amp_1U_20.fastq
```

In [10]:
```
#only one of forward nd reverse reads passed the trimming filter
36864/1823504 * 100
```

Out[10]:
```
2.0216023655555455
```

So, all the numbers matched with the Trimmomatic output

q=30

In [5]:
```
! java -jar /usr/share/java/trimmomatic.jar PE -phred33 amp_res_1.fastq amp_res_2.fastq amp_1P_30.fastq amp_1U_30.fastq amp_2P_30.fastq amp_2U_30.fastq
```

```
TrimmomaticPE: Started with arguments:
 -phred33 amp_res_1.fastq amp_res_2.fastq amp_1P_30.fastq amp_1U_30.fastq amp_2P_30.fastq amp_2U_30.fastq LEADING:30 TRAILING:30 SLIDINGWINDOW:10:30 MIN
LEN:30
Input Read Pairs: 455876 Both Surviving: 360209 (79.01%) Forward Only Surviving: 36716 (8.05%) Reverse Only Surviving: 27291 (5.99%) Dropped: 31660 (6.9
4%)
TrimmomaticPE: Completed successfully
```

In [11]:
```
! wc -l amp_1P_30.fastq
```

```
1440836 amp_1P_30.fastq
```

In [12]:
```
#percent of reads where both read passed the trimming filter
1440836/1823504 * 100
```

Out[12]:
```
79.01468820468725
```

In [13]:
```
! wc -l amp_1U_30.fastq
```

```
146864 amp_1U_30.fastq
```

In [14]:
```
#only one of forward nd reverse reads passed the trimming filter
146864/1823504 * 100
```

Out[14]:
```
8.053944493678106
```

So, all the numbers matched with the Trimmomatic output

FastQC inspecting of trimmed data

q=20

In [17]:
```
! fastqc -o . /home/gressy/Current\ work/Bioinf\ 2021-2022/Projects/Project1/amp_1P_20.fastq /home/gressy/Current\ work/Bioinf\ 2021-2022/Projects/Proje
```

```
Started analysis of amp_1P_20.fastq
Approx 5% complete for amp_1P_20.fastq
Approx 10% complete for amp_1P_20.fastq
Approx 15% complete for amp_1P_20.fastq
Approx 20% complete for amp_1P_20.fastq
Approx 25% complete for amp_1P_20.fastq
Approx 30% complete for amp_1P_20.fastq
Approx 35% complete for amp_1P_20.fastq
Approx 40% complete for amp_1P_20.fastq
Approx 45% complete for amp_1P_20.fastq
Approx 50% complete for amp_1P_20.fastq
Approx 55% complete for amp_1P_20.fastq
Approx 60% complete for amp_1P_20.fastq
Approx 65% complete for amp_1P_20.fastq
Approx 70% complete for amp_1P_20.fastq
Approx 75% complete for amp_1P_20.fastq
Approx 80% complete for amp_1P_20.fastq
Approx 85% complete for amp_1P_20.fastq
Approx 90% complete for amp_1P_20.fastq
```

```
Approx 95% complete for amp_1P_20.fastq
Analysis complete for amp_1P_20.fastq
Started analysis of amp_2P_20.fastq
Approx 5% complete for amp_2P_20.fastq
Approx 10% complete for amp_2P_20.fastq
Approx 15% complete for amp_2P_20.fastq
Approx 20% complete for amp_2P_20.fastq
Approx 25% complete for amp_2P_20.fastq
Approx 30% complete for amp_2P_20.fastq
Approx 35% complete for amp_2P_20.fastq
Approx 40% complete for amp_2P_20.fastq
Approx 45% complete for amp_2P_20.fastq
Approx 50% complete for amp_2P_20.fastq
Approx 55% complete for amp_2P_20.fastq
Approx 60% complete for amp_2P_20.fastq
Approx 65% complete for amp_2P_20.fastq
Approx 70% complete for amp_2P_20.fastq
Approx 75% complete for amp_2P_20.fastq
Approx 80% complete for amp_2P_20.fastq
Approx 85% complete for amp_2P_20.fastq
Approx 90% complete for amp_2P_20.fastq
Approx 95% complete for amp_2P_20.fastq
Analysis complete for amp_2P_20.fastq
```

FastQC Report

Thu 14 Oct 2021
amp_1P_20.fastq

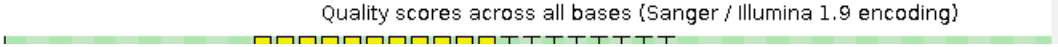
Summary

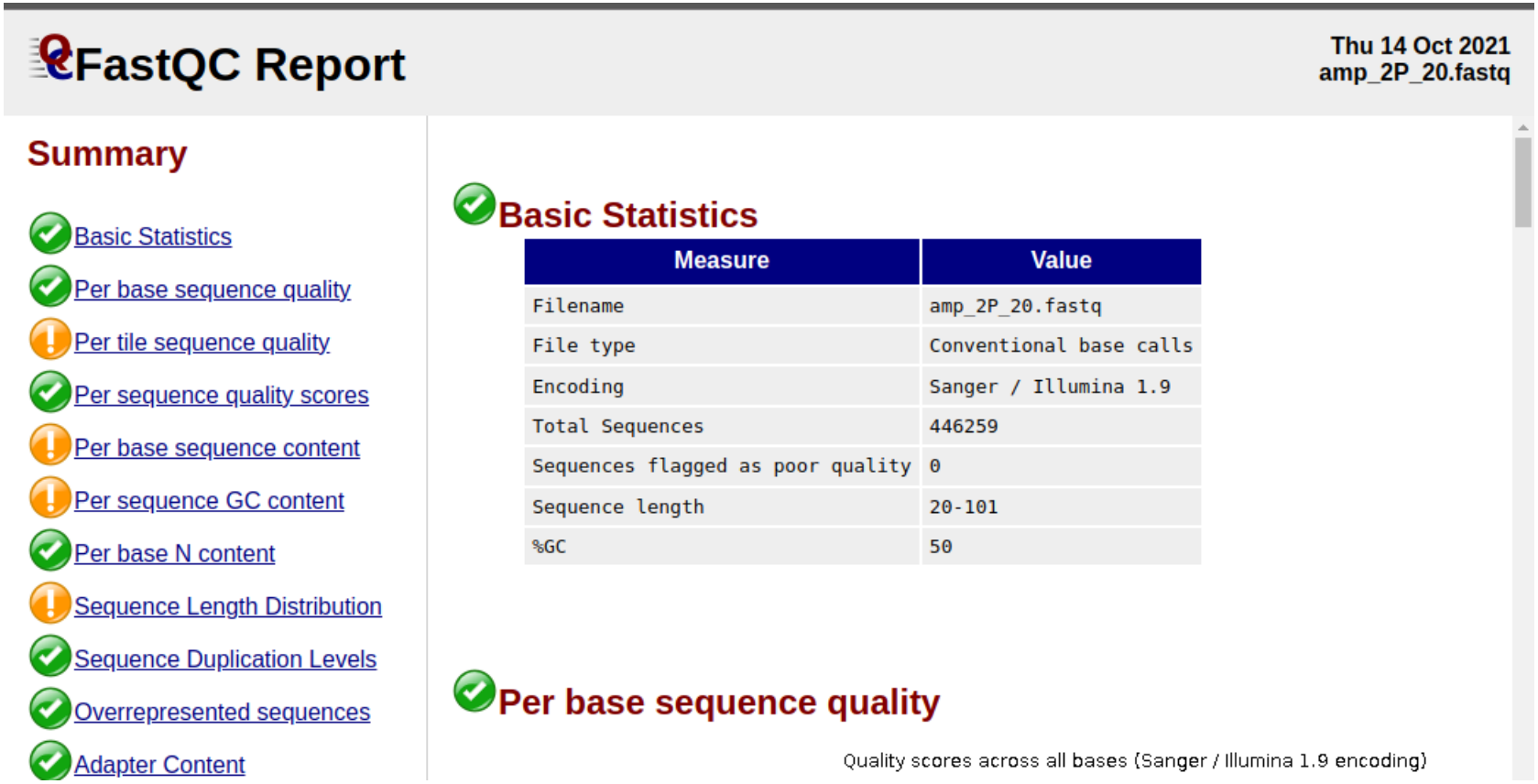
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	amp_1P_20.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	446259
Sequences flagged as poor quality	0
Sequence length	20-101
%GC	50

Per base sequence quality





FastQC Report

Thu 14 Oct 2021
amp_2P_20.fastq

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	amp_2P_20.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	446259
Sequences flagged as poor quality	0
Sequence length	20-101
%GC	50

Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)

q=30

In [18]:
```
! fastqc -o . /home/gressy/Current\ work/Bioinf\ 2021-2022/Projects/Project1/amp_1P_30.fastq /home/gressy/Current\ work/Bioinf\ 2021-2022/Projects/Proje
```

```
Started analysis of amp_1P_30.fastq
Approx 5% complete for amp_1P_30.fastq
Approx 10% complete for amp_1P_30.fastq
Approx 15% complete for amp_1P_30.fastq
Approx 20% complete for amp_1P_30.fastq
Approx 25% complete for amp_1P_30.fastq
Approx 30% complete for amp_1P_30.fastq
Approx 35% complete for amp_1P_30.fastq
Approx 40% complete for amp_1P_30.fastq
Approx 45% complete for amp_1P_30.fastq
Approx 50% complete for amp_1P_30.fastq
Approx 55% complete for amp_1P_30.fastq
Approx 60% complete for amp_1P_30.fastq
Approx 65% complete for amp_1P_30.fastq
Approx 70% complete for amp_1P_30.fastq
Approx 75% complete for amp_1P_30.fastq
Approx 80% complete for amp_1P_30.fastq
```

```
Approx 85% complete for amp_1P_30.fastq
Approx 90% complete for amp_1P_30.fastq
Approx 95% complete for amp_1P_30.fastq
Analysis complete for amp_1P_30.fastq
Started analysis of amp_2P_30.fastq
Approx 5% complete for amp_2P_30.fastq
Approx 10% complete for amp_2P_30.fastq
Approx 15% complete for amp_2P_30.fastq
Approx 20% complete for amp_2P_30.fastq
Approx 25% complete for amp_2P_30.fastq
Approx 30% complete for amp_2P_30.fastq
Approx 35% complete for amp_2P_30.fastq
Approx 40% complete for amp_2P_30.fastq
Approx 45% complete for amp_2P_30.fastq
Approx 50% complete for amp_2P_30.fastq
Approx 55% complete for amp_2P_30.fastq
Approx 60% complete for amp_2P_30.fastq
Approx 65% complete for amp_2P_30.fastq
Approx 70% complete for amp_2P_30.fastq
Approx 75% complete for amp_2P_30.fastq
Approx 80% complete for amp_2P_30.fastq
Approx 85% complete for amp_2P_30.fastq
Approx 90% complete for amp_2P_30.fastq
Approx 95% complete for amp_2P_30.fastq
Analysis complete for amp_2P_30.fastq
```

FastQC Report

Thu 14 Oct 2021
amp_1P_30.fastq

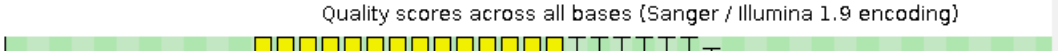
Summary

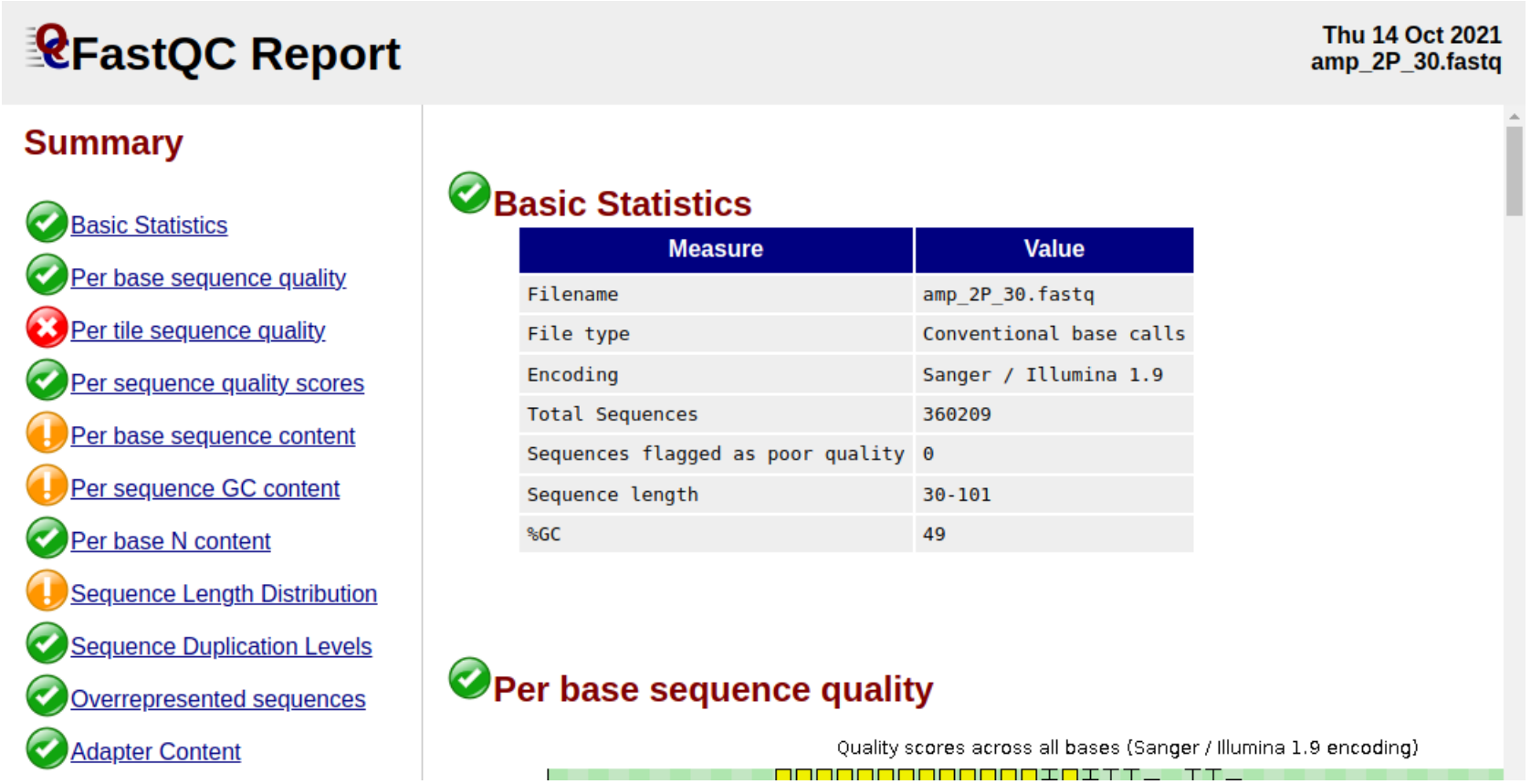
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Basic Statistics

Measure	Value
Filename	amp_1P_30.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	360209
Sequences flagged as poor quality	0
Sequence length	30-101
%GC	49

Per base sequence quality





Thus, the best quality is obtained by using Trimmomatic with quality score=20. So further we will use this data.

5. Aligning sequences to reference

For alignment will be used the app BWA-MEM (it makes use of the Burrows-Wheeler Transform (BWT))

5.1. Indexing the reference file

In [39]:
```
! bwa index GCA_000005845.2_ASM584v2_genomic.fna
```

```
[bwa_index] Pack FASTA... 0.04 sec
[bwa_index] Construct BWT for the packed sequence...
[bwa_index] 1.86 seconds elapse.
[bwa_index] Update BWT... 0.03 sec
[bwa_index] Pack forward-only FASTA... 0.02 sec
[bwa_index] Construct SA from BWT and Occ... 0.86 sec
[main] Version: 0.7.17-r1188
[main] CMD: bwa index GCA_000005845.2_ASM584v2_genomic.fna
[main] Real time: 2.838 sec; CPU: 2.807 sec
```

The result of command is 5 files:.sa, .pac, .bwt, .ann, .amb.

5.2 Aligning reads

In [41]:
```
! bwa mem GCA_000005845.2_ASM584v2_genomic.fna amp_1P_20.fastq amp_2P_20.fastq > alignment_20.sam
```

```
[M::bwa_idx_load_from_disk] read 0 ALT contigs
[M::process] read 106336 sequences (10000178 bp)...
[M::process] read 108264 sequences (10000141 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (9, 51236, 0, 22)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (145, 185, 230)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 400)
[M::mem_pestat] mean and std.dev: (189.31, 63.57)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 485)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (114, 158, 263)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 561)
[M::mem_pestat] mean and std.dev: (131.82, 63.27)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 710)
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 106336 reads in 4.662 CPU sec, 4.627 real sec
[M::process] read 108866 sequences (10000007 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (20, 51941, 0, 17)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (80, 107, 210)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 470)
[M::mem_pestat] mean and std.dev: (119.00, 69.53)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 600)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (143, 182, 228)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 398)
[M::mem_pestat] mean and std.dev: (187.08, 63.02)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 483)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (89, 176, 483)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 1271)
[M::mem_pestat] mean and std.dev: (169.14, 114.54)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 1665)
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 108264 reads in 4.120 CPU sec, 4.095 real sec
[M::process] read 106080 sequences (10000082 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (12, 52355, 0, 9)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (100, 144, 178)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 334)
[M::mem_pestat] mean and std.dev: (144.00, 68.68)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 419)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (142, 181, 226)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 394)
[M::mem_pestat] mean and std.dev: (185.62, 62.36)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 478)
```

```
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[M::mem_pestat] skip orientation FF
[M::mem_process_seqs] Processed 108866 reads in 3.172 CPU sec, 3.081 real sec
[M::process] read 105898 sequences (10000056 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (12, 51082, 0, 13)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (84, 104, 198)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 426)
[M::mem_pestat] mean and std.dev: (130.17, 58.69)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 540)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (144, 184, 231)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 405)
[M::mem_pestat] mean and std.dev: (189.11, 64.13)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 492)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (110, 130, 1006)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 2798)
[M::mem_pestat] mean and std.dev: (475.38, 646.65)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 3694)
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 106080 reads in 2.759 CPU sec, 2.696 real sec
[M::process] read 106462 sequences (10000139 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (13, 50847, 0, 11)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (54, 73, 174)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 414)
[M::mem_pestat] mean and std.dev: (115.38, 97.82)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 534)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (143, 182, 227)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 395)
[M::mem_pestat] mean and std.dev: (186.61, 62.43)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 479)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (83, 125, 196)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 422)
[M::mem_pestat] mean and std.dev: (118.30, 58.38)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 535)
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 105898 reads in 2.847 CPU sec, 2.780 real sec
[M::process] read 107332 sequences (10000102 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (18, 51182, 0, 16)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (89, 126, 175)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 347)
[M::mem_pestat] mean and std.dev: (122.25, 49.00)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 433)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (144, 183, 229)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 399)
[M::mem_pestat] mean and std.dev: (188.25, 63.49)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 484)
[M::mem_pestat] skip orientation RF as there are not enough pairs
```

```
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (52, 105, 220)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 556)
[M::mem_pestat] mean and std.dev: (120.33, 80.63)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 724)
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 106462 reads in 2.962 CPU sec, 2.883 real sec
[M::process] read 104428 sequences (10000060 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (11, 51264, 0, 13)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (54, 90, 172)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 408)
[M::mem_pestat] mean and std.dev: (108.73, 60.01)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 526)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (143, 183, 228)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 398)
[M::mem_pestat] mean and std.dev: (187.11, 62.86)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 483)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (58, 108, 277)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 715)
[M::mem_pestat] mean and std.dev: (144.67, 98.99)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 934)
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 107332 reads in 3.129 CPU sec, 3.049 real sec
[M::process] read 38852 sequences (3652257 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (12, 50337, 0, 10)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (68, 135, 213)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 503)
[M::mem_pestat] mean and std.dev: (120.09, 68.06)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 648)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (143, 183, 229)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 401)
[M::mem_pestat] mean and std.dev: (188.26, 63.56)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 487)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation RR...
[M::mem_pestat] (25, 50, 75) percentile: (102, 140, 152)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (2, 252)
[M::mem_pestat] mean and std.dev: (116.11, 42.39)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 302)
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RR
[M::mem_process_seqs] Processed 104428 reads in 3.005 CPU sec, 2.943 real sec
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (2, 18569, 0, 4)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (141, 179, 224)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 390)
[M::mem_pestat] mean and std.dev: (184.02, 61.49)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 473)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
```

```
[M::mem_process_seqs] Processed 38852 reads in 1.149 CPU sec, 1.094 real sec
[main] Version: 0.7.17-r1188
[main] CMD: bwa mem GCA_000005845.2_ASM584v2_genomic.fna amp_1P_20.fastq amp_2P_20.fastq
[main] Real time: 27.490 sec; CPU: 27.986 sec
```

The result of the command is alignment_20.sam file,in which the alignment result is written.

5.3 Compressing SAM file

working with files is performed using the program Samtools

In [47]:
```
#A compressed sam file is called a bam file
! samtools view -S -b alignment_20.sam > alignment_20.bam
```

In [48]:
```
#some statistics
! samtools flagstat alignment_20.bam
```

```
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)
```

So, mapped 99.87% of reads

5.4 Sorting and indexing BAM file

In [49]:
```
#sorting by sequence coordinate on reference
! samtools sort alignment_20.bam -o alignment_20_sorted.bam
```

In [50]:
```
#indexing file for faster search
! samtools index alignment_20_sorted.bam
```

So, now we have reference aligned reads!

6. Variant calling

The goal of work is to find mutations, responsible for ampicillin resistance.

And this section is about find vaiants in reads comparing with reference genom.

In [51]:

```
! samtools mpileup -f GCA_000005845.2_ASM584v2_genomic.fna alignment_20_sorted.bam >  my0.mpileup
```

```
[mpileup] 1 samples in 1 input files
```

In [52]:

```
#this command doesn't work without "" and pull path to VarScan
! java -jar "/home/gressy/Programs/VarScan.v2.3.9.jar" mpileup2snp -h
```

```
Only SNPs will be reported
Warning: No p-value threshold provided, so p-values will not be calculated
Min coverage:   8
Min reads2:     2
Min var freq:   0.2
Min avg qual:   15
P-value thresh: 0.01
USAGE: java -jar VarScan.jar mpileup2cns [pileup file] OPTIONS
        mpileup file - The SAMtools mpileup file

        OPTIONS:
        --min-coverage  Minimum read depth at a position to make a call [8]
        --min-reads2    Minimum supporting reads at a position to call variants [2]
        --min-avg-qual  Minimum base quality at a position to count a read [15]
        --min-var-freq  Minimum variant allele frequency threshold [0.01]
        --min-freq-for-hom      Minimum frequency to call homozygote [0.75]
        --p-value       Default p-value threshold for calling variants [99e-02]
        --strand-filter Ignore variants with >90% support on one strand [1]
        --output-vcf    If set to 1, outputs in VCF format
        --vcf-sample-list       For VCF output, a list of sample names in order, one per line
        --variants      Report only variant (SNP/indel) positions [0]
```

We need to choose mutations, the percentage of which is more than a certain threshold. To choose a threshold, we can try to set it to different values and look at the differences.

In [59]:

```
#threshold=10%
! java -jar "/home/gressy/Programs/VarScan.v2.3.9.jar"  mpileup2snp my0.mpileup --min-var-freq 0.1 --variants --output-vcf 1 > VarScan_results_0.1.vcf
```

```
Only SNPs will be reported
Warning: No p-value threshold provided, so p-values will not be calculated
Min coverage:   8
Min reads2:     2
Min var freq:   0.1
Min avg qual:   15
P-value thresh: 0.01
Reading input from my0.mpileup
4641343 bases in pileup file
9 variant positions (6 SNP, 3 indel)
1 were failed by the strand-filter
5 variant positions reported (5 SNP, 0 indel)
```

In [66]:

```
#threshold=1%
! java -jar "/home/gressy/Programs/VarScan.v2.3.9.jar"  mpileup2snp my0.mpileup --min-var-freq 0.01 --variants --output-vcf 1 > VarScan_results_0.01.vcf
```

```
Only SNPs will be reported
Warning: No p-value threshold provided, so p-values will not be calculated
Min coverage:   8
Min reads2:     2
Min var freq:   0.01
Min avg qual:   15
```

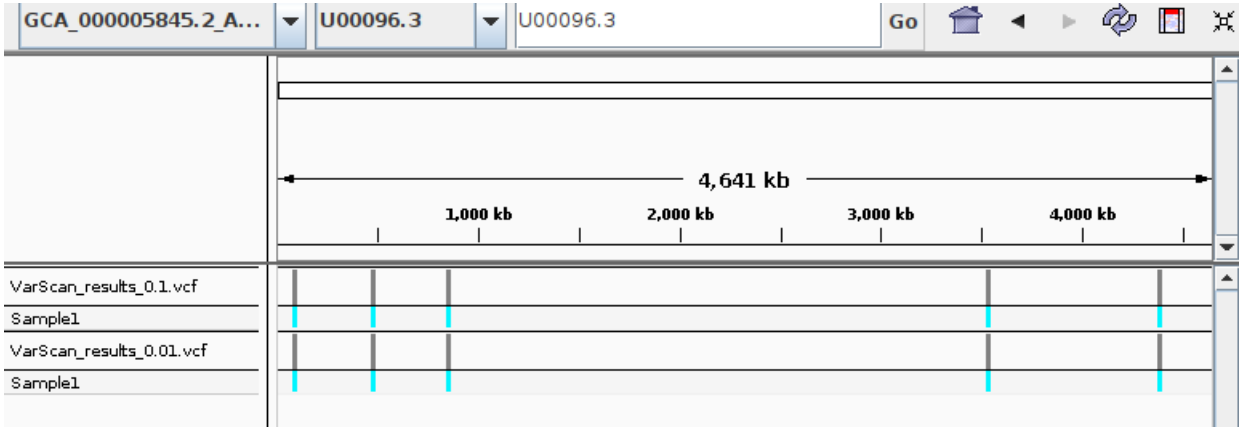
```
P-value thresh: 0.01
Reading input from my0.mpileup
4641343 bases in pileup file
9 variant positions (6 SNP, 3 indel)
1 were failed by the strand-filter
5 variant positions reported (5 SNP, 0 indel)
```

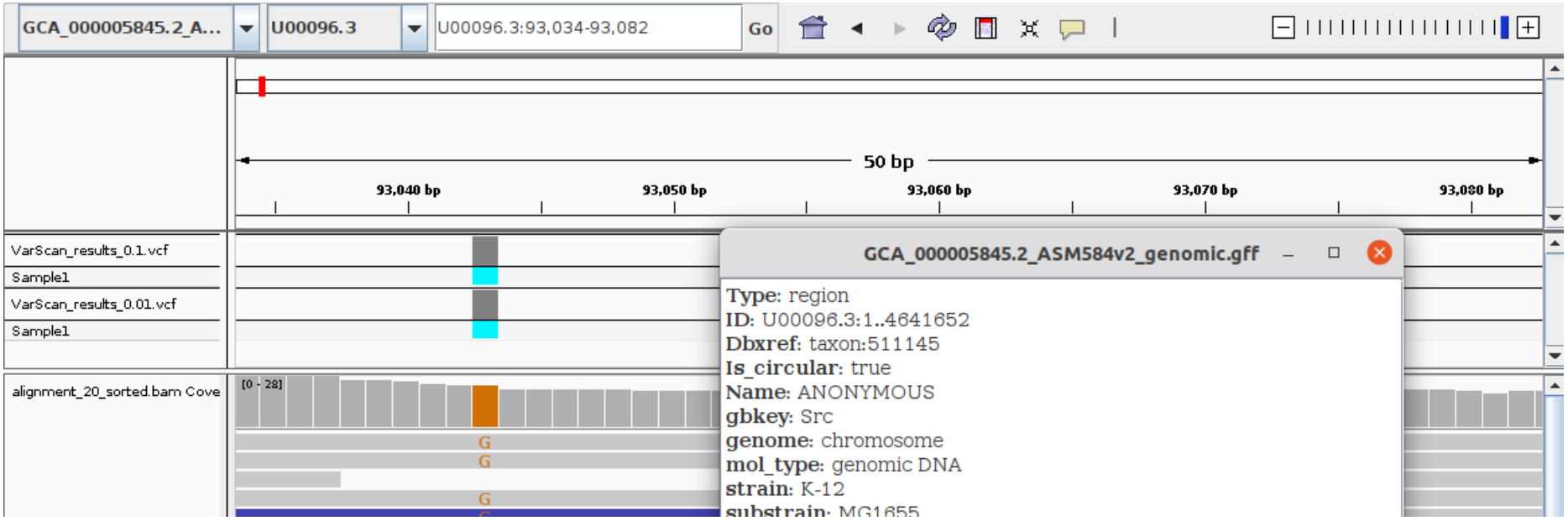
Here we have 4 files with variants

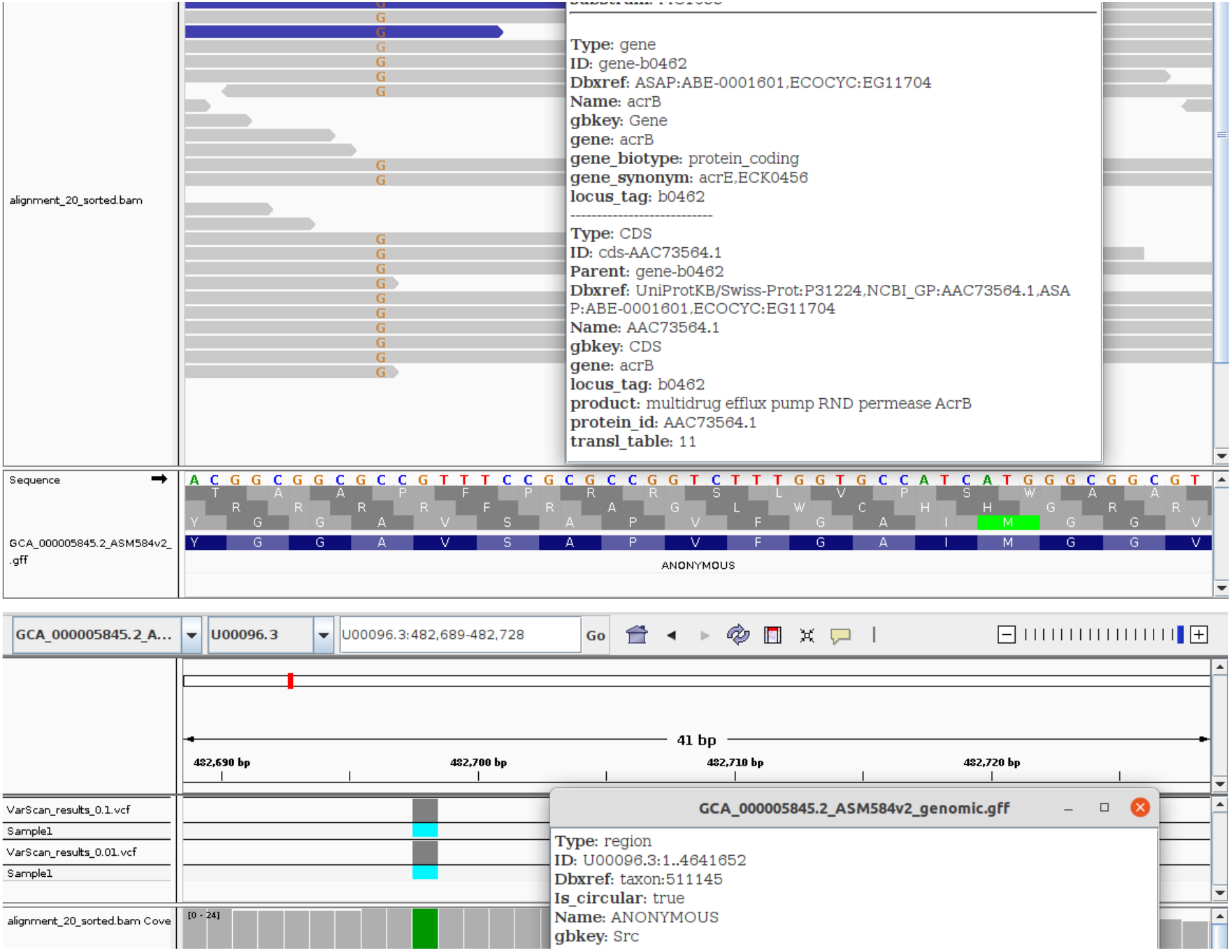
7. Variant effect prediction

There we will look at variants using IGV genome browser.

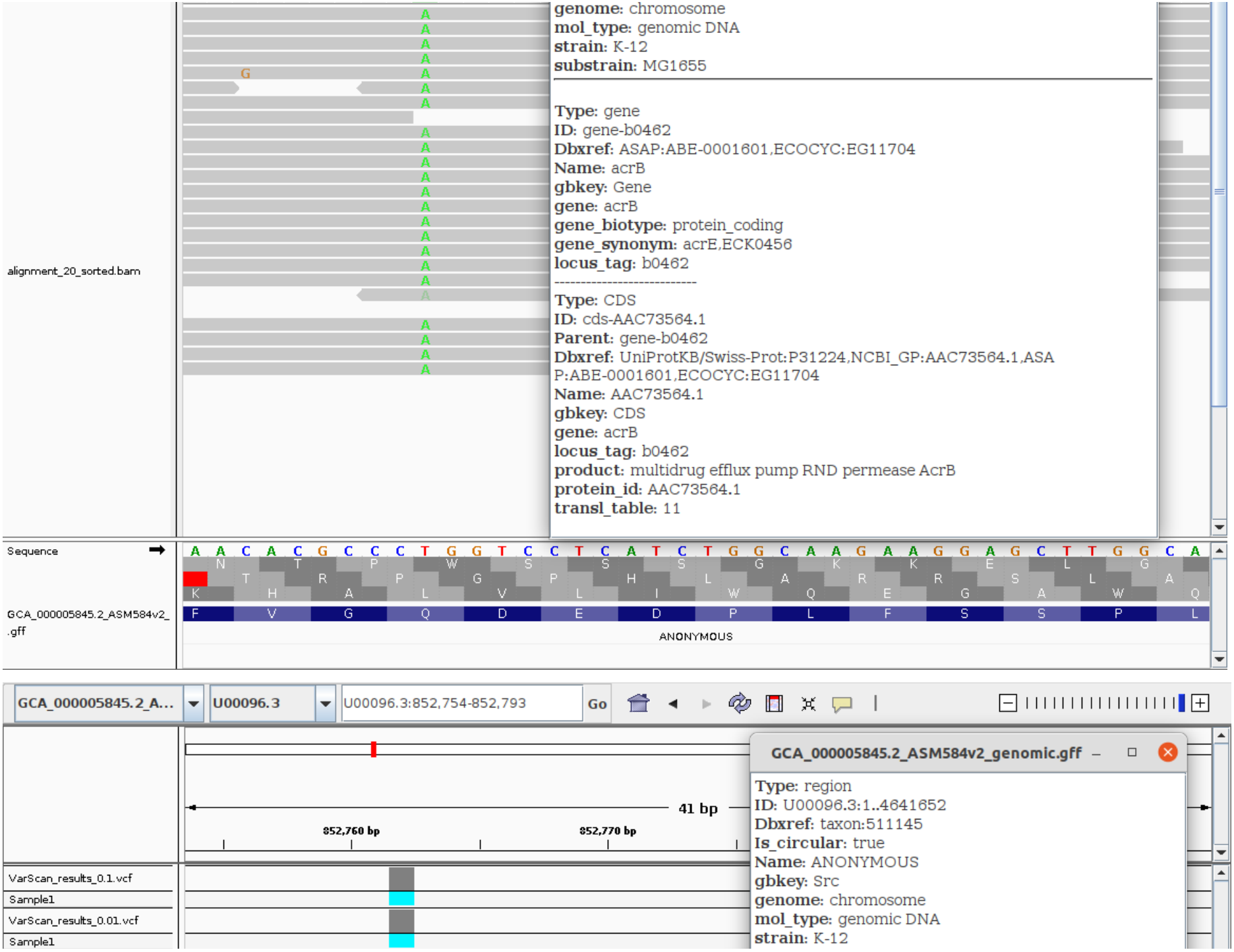
The results of work are presented below

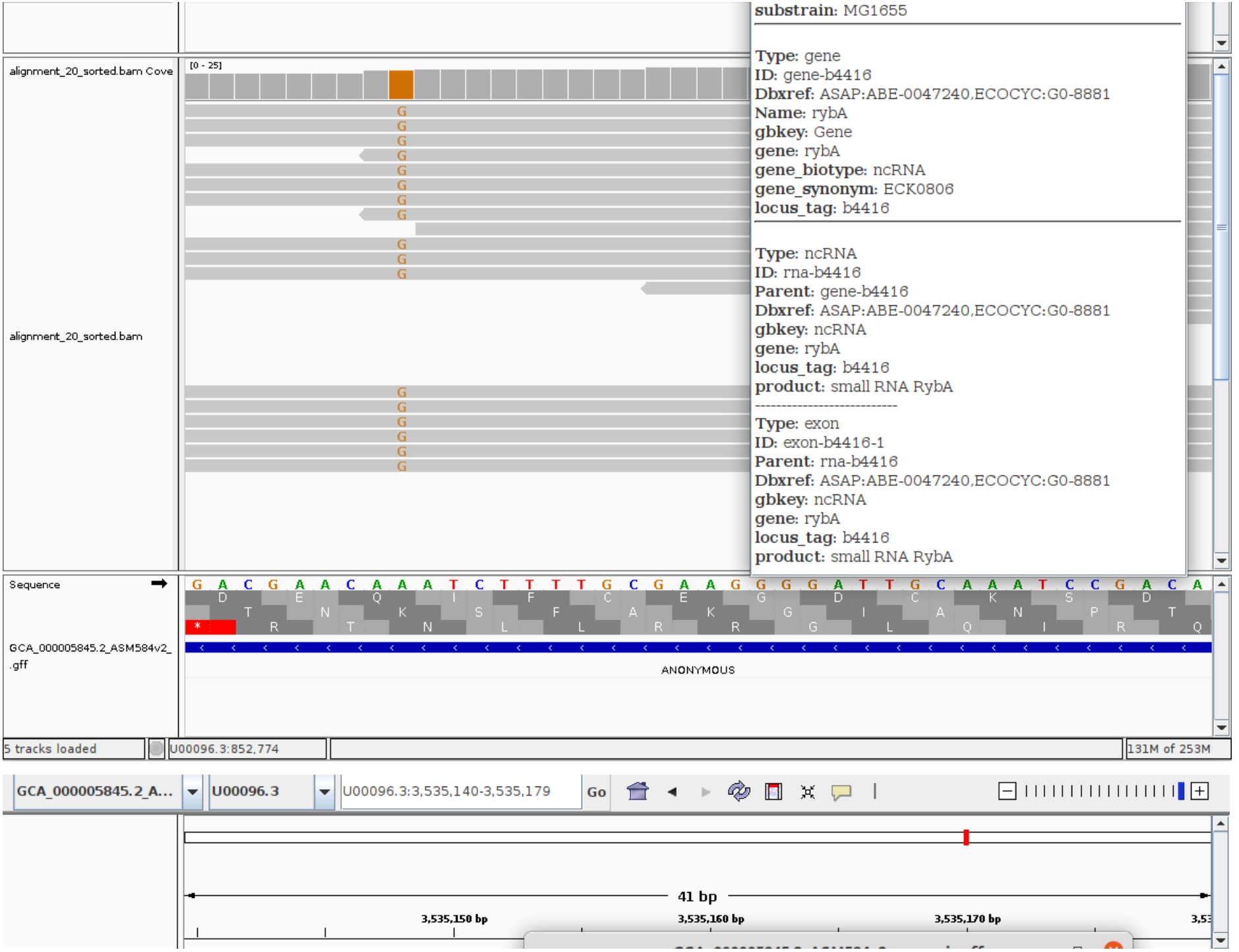


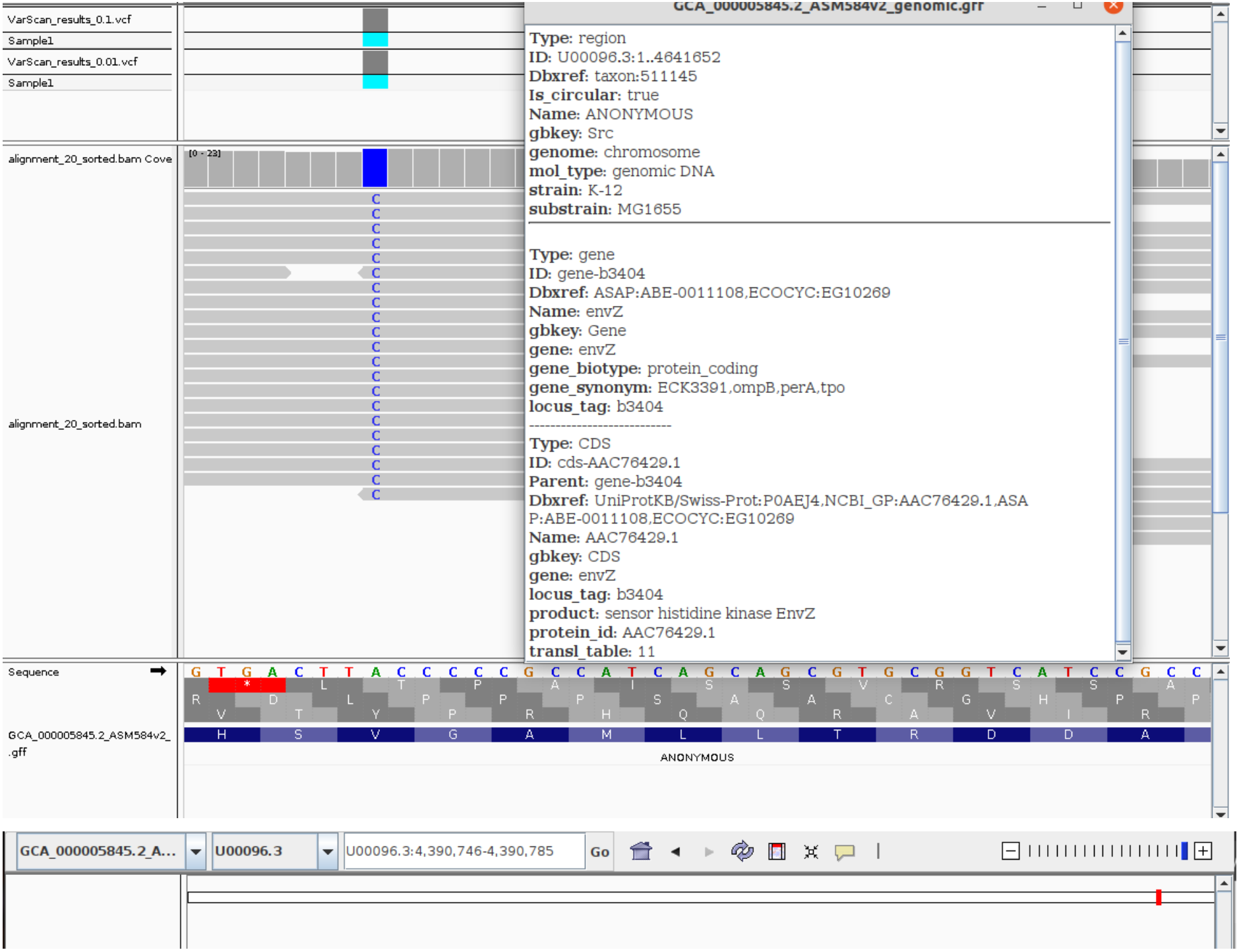


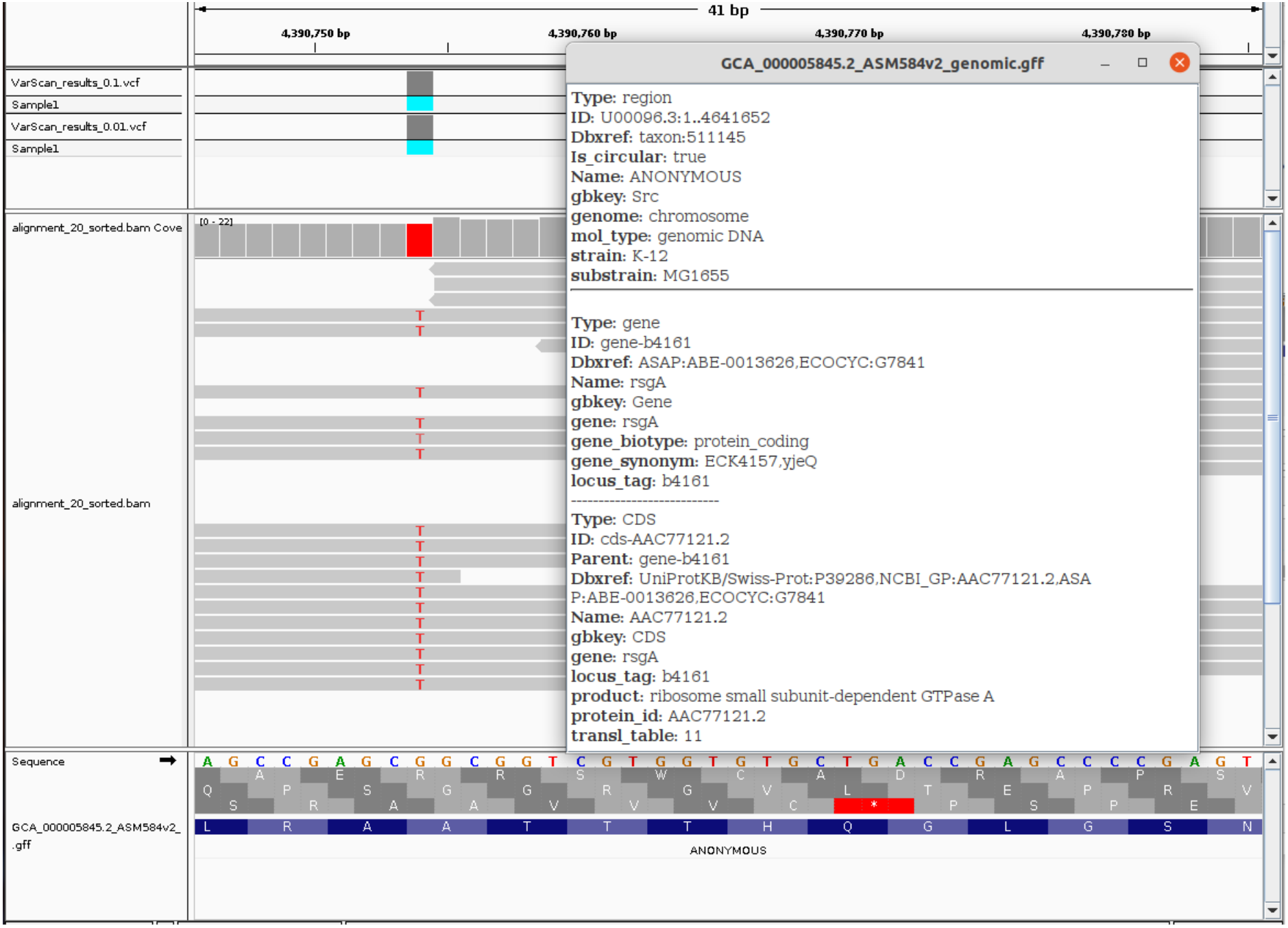

alignment_20_sorted.bam
Type: gene
ID: gene-b0462
Dbxref: ASAP:ABE-0001601,ECOCYC:EG11704
Name: acrB
gbkey: Gene
gene: acrB
gene_biotype: protein_coding
gene_synonym: acrE,ECK0456
locus_tag: b0462

Type: CDS
ID: cds-AAC73564.1
Parent: gene-b0462
Dbxref: UniProtKB/Swiss-Prot:P31224,NCBI_GP:AAC73564.1,ASAP:ABE-0001601,ECOCYC:EG11704
Name: AAC73564.1
gbkey: CDS
gene: acrB
locus_tag: b0462
product: multidrug efflux pump RND permease AcrB
protein_id: AAC73564.1
transl_table: 11
Sequence
GCA_000005845.2_ASM584v2_.gff
ANONYMOUS
GCA_000005845.2_A...
U00096.3
U00096.3:482,689-482,728
Go
41 bp
482,690 bp
482,700 bp
482,710 bp
482,720 bp
VarScan_results_0.1.vcf
Sample1
VarScan_results_0.01.vcf
Sample1
alignment_20_sorted.bam Cove
[0 - 24]
GCA_000005845.2_ASM584v2_genomic.gff
Type: region
ID: U00096.3:1..4641652
Dbxref: taxon:511145
Is_circular: true
Name: ANONYMOUS
gbkey: Src


alignment_20_sorted.bam
genome: chromosome
mol_type: genomic DNA
strain: K-12
substrain: MG1655
Type: gene
ID: gene-b0462
Dbxref: ASAP:ABE-0001601,ECOCYC:EG11704
Name: acrB
gbkey: Gene
gene: acrB
gene_biotype: protein_coding
gene_synonym: acrE,ECK0456
locus_tag: b0462
Type: CDS
ID: cds-AAC73564.1
Parent: gene-b0462
Dbxref: UniProtKB/Swiss-Prot:P31224,NCBI_GP:AAC73564.1,ASAP:ABE-0001601,ECOCYC:EG11704
Name: AAC73564.1
gbkey: CDS
gene: acrB
locus_tag: b0462
product: multidrug efflux pump RND permease AcrB
protein_id: AAC73564.1
transl_table: 11
Sequence
GCA_000005845.2_ASM584v2_.gff
ANONYMOUS
GCA_000005845.2_A...
U00096.3
U00096.3:852,754-852,793
Go
41 bp
852,760 bp
852,770 bp
VarScan_results_0.1.vcf
Sample1
VarScan_results_0.01.vcf
Sample1
GCA_000005845.2_ASM584v2_genomic.gff
Type: region
ID: U00096.3:1..4641652
Dbxref: taxon:511145
Is_circular: true
Name: ANONYMOUS
gbkey: Src
genome: chromosome
mol_type: genomic DNA
strain: K-12


substrain: MG1655
Type: gene
ID: gene-b4416
Dbxref: ASAP:ABE-0047240,ECOCYC:G0-8881
Name: rybA
gbkey: Gene
gene: rybA
gene_biotype: ncRNA
gene_synonym: ECK0806
locus_tag: b4416
Type: ncRNA
ID: rna-b4416
Parent: gene-b4416
Dbxref: ASAP:ABE-0047240,ECOCYC:G0-8881
gbkey: ncRNA
gene: rybA
locus_tag: b4416
product: small RNA RybA
Type: exon
ID: exon-b4416-1
Parent: rna-b4416
Dbxref: ASAP:ABE-0047240,ECOCYC:G0-8881
gbkey: ncRNA
gene: rybA
locus_tag: b4416
product: small RNA RybA
alignment_20_sorted.bam Cove
[0 - 25]
alignment_20_sorted.bam
Sequence
GCA_000005845.2_ASM584v2_.gff
ANONYMOUS
5 tracks loaded
U00096.3:852,774
131M of 253M
GCA_000005845.2_A...
U00096.3
U00096.3:3,535,140-3,535,179
Go
41 bp
3,535,150 bp
3,535,160 bp
3,535,170 bp


VarScan_results_0.1.vcf
Sample1
VarScan_results_0.01.vcf
Sample1
alignment_20_sorted.bam Cove
alignment_20_sorted.bam
Sequence
GCA_000005845.2_ASM584v2_.gff
ANONYMOUS
GCA_000005845.2_genomic.gff
Type: region
ID: U00096.3:1..4641652
Dbxref: taxon:511145
Is_circular: true
Name: ANONYMOUS
gbkey: Src
genome: chromosome
mol_type: genomic DNA
strain: K-12
substrain: MG1655
Type: gene
ID: gene-b3404
Dbxref: ASAP:ABE-0011108,ECOCYC:EG10269
Name: envZ
gbkey: Gene
gene: envZ
gene_biotype: protein_coding
gene_synonym: ECK3391,ompB,perA,tpo
locus_tag: b3404
Type: CDS
ID: cds-AAC76429.1
Parent: gene-b3404
Dbxref: UniProtKB/Swiss-Prot:P0AEJ4,NCBI_GP:AAC76429.1,ASAP:ABE-0011108,ECOCYC:EG10269
Name: AAC76429.1
gbkey: CDS
gene: envZ
locus_tag: b3404
product: sensor histidine kinase EnvZ
protein_id: AAC76429.1
transl_table: 11
GCA_000005845.2_A...
U00096.3
U00096.3:4,390,746-4,390,785
Go

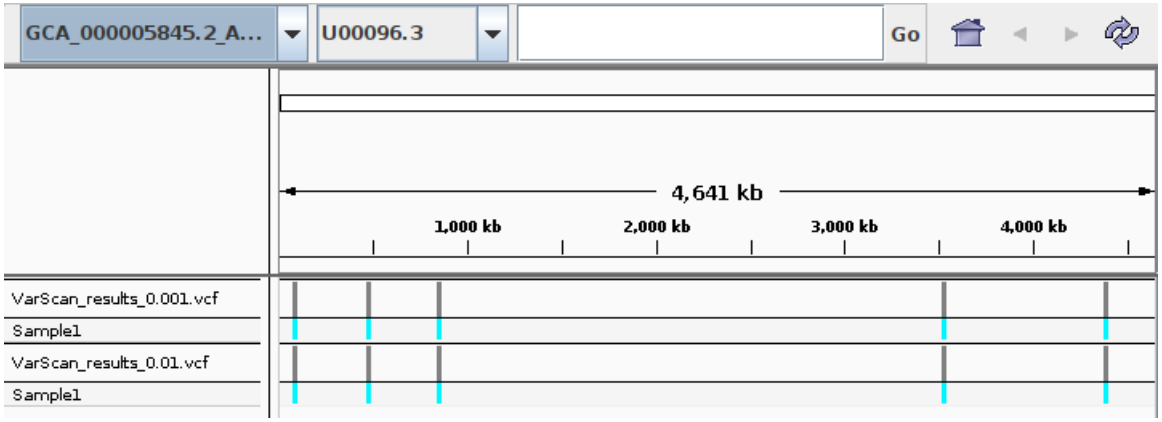


So, there are 5 variants with 100% mismatch.

To find mismatches with less threshold we run command with threshold=0.1%.

In [69]:
```
#threshold=0.1%
! java -jar "/home/gressy/Programs/VarScan.v2.3.9.jar"  mpileup2snp my0.mpileup --min-var-freq 0.001 --variants --output-vcf 1 > VarScan_results_0.001.v
```

```
Only SNPs will be reported
Warning: No p-value threshold provided, so p-values will not be calculated
Min coverage:   8
Min reads2:     2
Min var freq:   0.001
Min avg qual:   15
P-value thresh: 0.01
Reading input from my0.mpileup
4641343 bases in pileup file
9 variant positions (6 SNP, 3 indel)
1 were failed by the strand-filter
5 variant positions reported (5 SNP, 0 indel)
```



This is no new mutation. If the percentage of mutations is less, these may already be random errors comparing with 100% mismatches. So, there are 5 variants to research in lab report.