A1.1B  
  
You have sequenced DNA extracted from (what you think is) Mycobacterium Leprae. You received two fastq files with reads that are 10bp and 30bp long each. You will align both of these files and investigate the effect of read length on the resulting mapped reads.

You can find the reference genome for M. Leprae here:

/home/abigail/Assignment2025\_Part1B\_Data/Mycobacterium\_leprae.fa.gz

And the fastq files here:

/home/abigail/Assignment2025\_Part1B\_Data/L10.fq.gz  
/home/abigail/Assignment2025\_Part1B\_Data/L30.fq.gz

Note1: We assume that adapters have already been filtered out from the reads.   
Note2: In your answers, please supply the command(s) you used in each question.

Some inspiration and hints can be found in   
https://github.com/ANGSD/adv\_binf\_2025\_week1/tree/main/day1  
and  
<https://github.com/ANGSD/adv_binf_2025_week1/tree/main/day2>

Before starting I move all of the files to my ~directory with

cp /home/abigail/Assignment2025\_Part1B\_Data/Mycobacterium\_leprae.fa.gz ~/M.leprae.fa.gz  
…

* 1. What is the filesize of the reference file? How large (in bases) is the bacterial reference genome  
     ls -l ~/M.leprae.fa.gz  
     Filesize is 999660 bytes

cat M.leprae.fa.gz.fai  
3 187 112 BP

* 1. Calculate the GC content of the reference genome.

zcat M.leprae.fa.gz | grep -v "^>" | tr -d '\n' | awk '{gc+=gsub(/[GCgc]/,""); at+=gsub(/[ATat]/,"")} END{print (gc/(gc+at))\*100"%"}'  
57.8103%

1. For each fastq file, make a histogram of the GC content of the reads. Are these histograms consistent with the GC content of the reference genome? How could you explain this consistency/inconsistency?  
   I decided to make the histogram in a bash instance of python and then export the histograms as pdfs to my computer, this way do do not have to download any files and I can make good looking python histograms.

Make python file on server

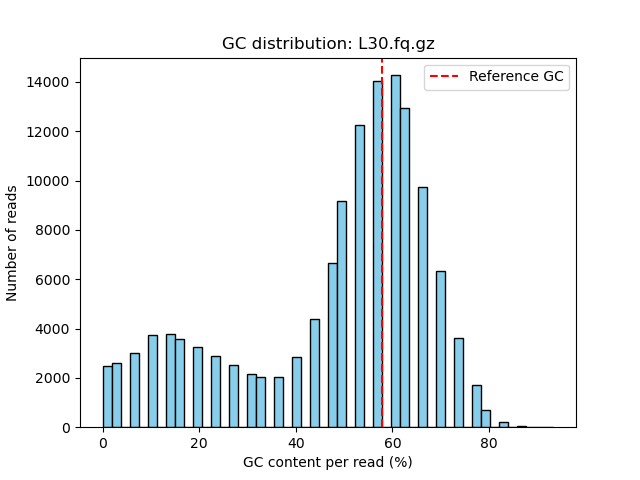
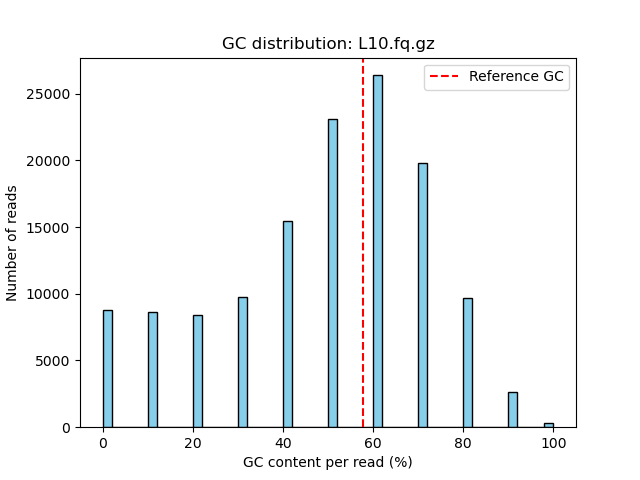
nano gc\_plots.py

Writing the code for histogram with python in that file.

I ran into issues: had to install biopython and downgrade numpy into older version….

run file

python3 gc\_plots.py -it uses the files on the server and creates pdf files which I can then copy to my computer using:  
scp advbinf9@emily.popgen.dk:~/A1/\*.png

This copies the files safely to my computer:

This seems very consistent to the reference, the peak is at about the center of the gc content per read. There seems to be a wider tail of gc\_poor reads. The gc poor reads are likely part of repetitive AT-rich regions, promoters. It could partly be explained by sequencing errors. But I will not be investigating the confidence of the reads.

1. Use bwa aln (and bwa samse) to align the two fastq files to the bacterial reference (Hint: look at the exercises from day 2).

advbinf9@emily:~/A1$ bwa index M.leprae.fa.gz

advbinf9@emily:~/A1$ bwa aln M.leprae.fa.gz L30.fq.gz > L30.sai

advbinf9@emily:~/A1$ bwa aln M.leprae.fa.gz L10.fq.gz > L10.sai

advbinf9@emily:~/A1$ bwa samse M.leprae.fa.gz L30.sai L30.fq.gz > L30.sam

advbinf9@emily:~/A1$ bwa samse M.leprae.fa.gz L10.sai L10.fq.gz > L10.sam

Sort and index the resulting files using samtools, and make sure they are saved in bam format.

samtools view -bS L30.sam | samtools sort -o L30.sorted.bam

samtools view -bS L10.sam | samtools sort -o L10.sorted.bam

samtools index L30.sorted.bam

samtools index L10.sorted.bam

Filter out the unaligned reads and create new bam files that contain only the mapped reads. Identify which flag to filter out on https://broadinstitute.github.io/picard/explain-flags. html   
Flag: -F 4 to keep the only aligned reads.

samtools view -b -F 4 L30.sorted.bam > L30.mapped.bam

samtools view -b -F 4 L10.sorted.bam > L10.mapped.bam

For each fastq files, what is the proportion of reads that could be mapped? Why do you think these were the mapping proportions?

samtools flagstat L30.sorted.bam

133000 + 0 in total (QC-passed reads + QC-failed reads)

100036 + 0 mapped (75.22% : N/A)

Mapping proportion of L30: 75.22%

samtools flagstat L10.sorted.bam

133000 + 0 in total (QC-passed reads + QC-failed reads)

133000 + 0 mapped (100.00% : N/A)

Mapping proportion of L10: 100%

Mapping in the shorter reads are 100% because the reads are short. They can be aligned by chance because the reference genome is long enough and the reads are short enough.

The mapping of the 30 bp reads is much lower, around 30, they are still able to be aligned randomly but the probability is far lower. The aligned reads of 30 bp will actually provide some information because they will be less ambiguous ( they can not be aligned all over the place so one can therefore say potentially draw conclutions that the source of the reads is related to the reference.

1. For each of the 2 bam files from 3.3, make a histogram of the GC content of the mapped reads. Compare these with the results from 2. Do you notice any change? Why?

A graph with blue bars

AI-generated content may be incorrect.

GC-content is the same for L10 since no reads was removed in the filtering step. Nothing to discuss here.

A graph of a number of blue bars

AI-generated content may be incorrect.

Here we can see changes because about 30 percent of the reads were removed. We ca also see that most of the reads that were removed were the ones that are of lower GC content. From this I draw the conclusion that these removed reads are mostly from repeat regions with either high mutation rates and or high sequencing error rates. Repeat regions are usually non-coding and will not have any evolutionary effect when they are mutated this is why they become mutational hotspots.

1. In addition to the mapping quality, bwa reports some information about the number of ’best’ and ’suboptimal’ hits for each read. These are the X0 and X1 tags in the bamfile (you can read more about them in the bwa reporting style in the online manual https://bio-bwa.sourceforge. net/bwa.shtml).

advbinf9@emily:~/A1$ samtools view L10.mapped.bam | head

advbinf9@emily:~/A1$ samtools view L30.mapped.bam | head  
I ran this to inspect the file from the top.  
here I see ex:

for L10

T3\_RID4\_S0\_NZ\_CP029543.1:1777512-1777521\_length:10\_R1 0 NZ\_CP029543.1 77 0 10M \* 0 0

TAGGTTTCAC CC=GGG=GGG XT:A:R NM:i:0 X0:i:5 X1:i:93 XM:i:0 XO:i:0 XG:i:0 MD:Z:10

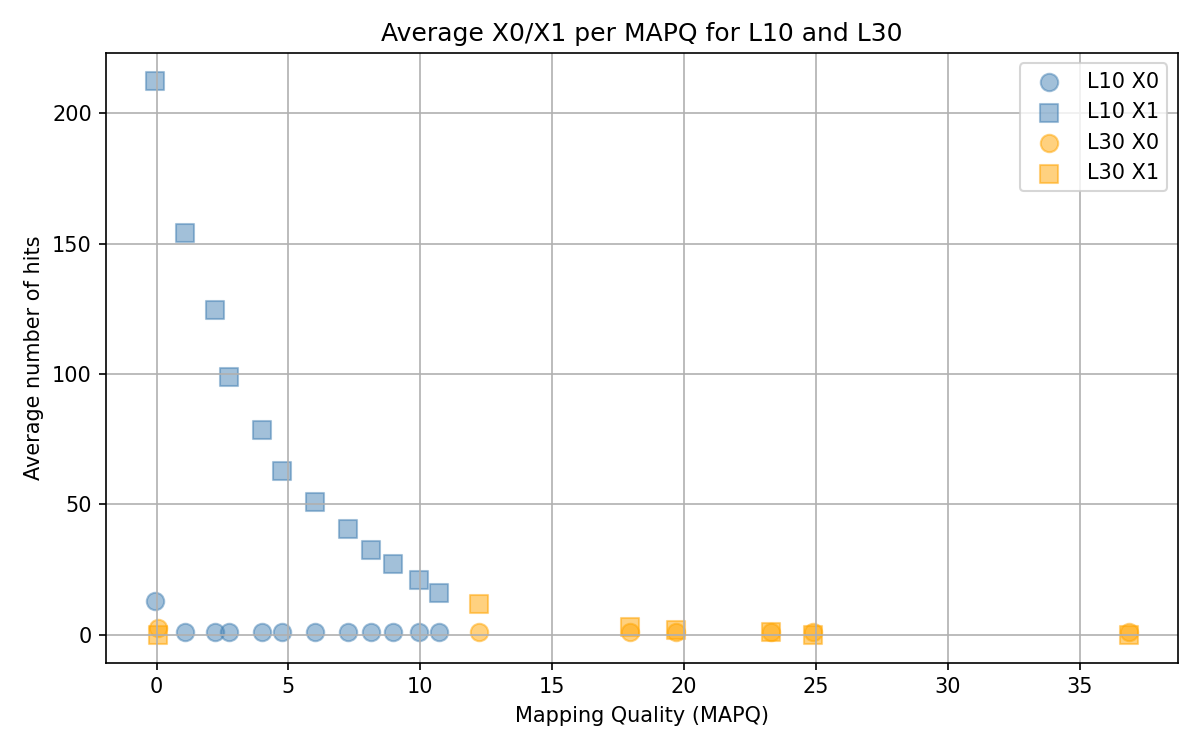
Due to the read being short it is being aligned all over the reference genome.

For L30

advbinf9@emily:~/A1$ samtools view L30.mapped.bam | head

T2\_RID19\_S0\_NZ\_CP029543.1:36-65\_length:30\_R1 0 NZ\_CP029543.1 36 37 30M \* 0 0 CGAGAACCAGAGAGATACGTCGTTGGCCGA =C=GG=GGGGGGGJJJGJ=JGJJJJJJJJJ XT:A:U NM:i:0 X0:i:1 X1:i:0 XM:i:0 XO:i:0 XG:i:0 MD:Z:30

This read is far longer and is only being aligned perfectly at one location in the reference. This is good since we can be quite certain that this is a true alignment.

For each of the files (L10 and L30), for each mapping quality value, plot the average number of ’best’ hits and the average number of ’suboptimal’ hits for the reads in each mapping quality category.

In general, which one of the files has more ’suboptimal’ hits? Why is this? How does the mapping quality reflect the number of ’suboptimal’ hits? Does the number of ’best’ hits also have to do with the mapping quality?

The L10 file has more suboptimal hits, however these hits are useless since they are ambiguous. They get more suboptimal hits because they happen by chance because the reads are too short. More suboptimal hits leads to lower mapping quality. Yes when there are many best hits the mapping quality also decreases. The best situation is when there is only one best hit or one suboptimal hit. Because then one can be sure that it is a true alignment.

1. Now create two new bam files where you filter the reads so we only retain reads with a mapping quality greater than or equal to 1.

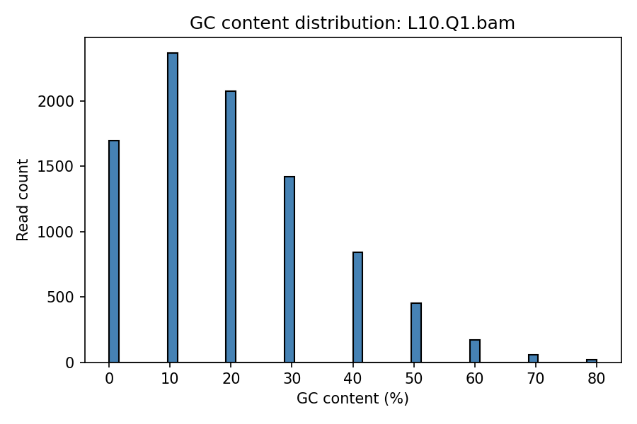
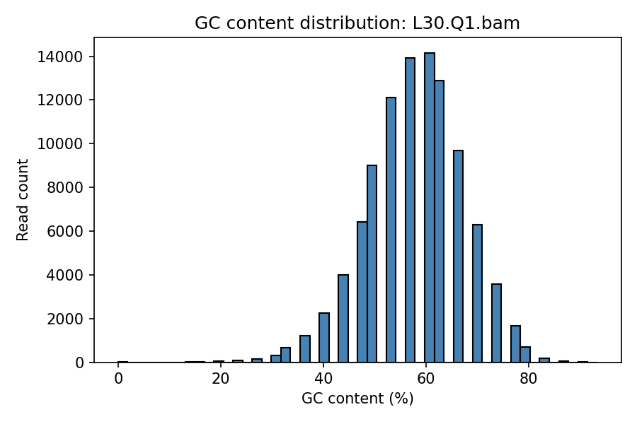
Filtering L30:

samtools view -b -q 1 -F 4 L30.mapped.bam > L30.Q1.bam

Filtering L10

samtools view -b -q 1 -F 4 L10.mapped.bam > L10.Q1.bam

Using these mapped and filtered reads, make a GC content histogram like the one in 4.



How do these new histograms compare with the histograms you made for the data before mapping (1.2) and after mapping but without a mapping quality filter (4.)? Do you notice any change? Why?

The L10.Q1 plot is very different from the previous ones. I suspect that the reads that are left are either there due to pure chance or perhaps due to mutations in those regions gc poor regions leading to unlikely sequences which are less likely to me mapped anywhere. Overall, I do not think that this plot is very interesting other than that most reads have been filtered out.

Samtools view -c L10.mapped.bam:

133000

Samtools view -c L10.Q1.bam:

9101

*This tells me that only 9101 of the 133000 reads has MAPQ > 1, is less than 10% remaining after filtering. This was consistent with previous findings.*

The L30.Q1 plot is like the previous one afte7r first filtering. I did not expect this one to be very different since we could see in the mapping quality plot that most reads in this file had high MAPQ.

Samtools view -c L30.mapped.bam:

100036

Samtools view -c L30.mapped.bam:

99485

*This tells me that 99485 of the 100036 reads has MAPQ > 1, about than 99% remaining after filtering. This was expected.*

1. For the M. Leprae genome we are using, what is the expected number of random hits with UP TO 2 mismatches for reads that are 10 and 30 bp long?

How many variants there are of a 10 bp read. And there are 4^n since there are four alternatives at each n position.

For the mismatches I ask myself how many alternatives of this string there are with mismatches. This would be 10 over 2 combinations with 3 alternatives at each, so total 405 ways, same calculation with 0 and 1 missmatch gives total 436 ways. So, the probability that a random n bp sequence matches with 2 missmatches is:

436/(4^n) where n is the seq len.

This number can be multiplied by the length of the sequence which we are aligning onto to get the expected number of random hits assuming independence between sequences and that sequences are random. Using the length of the sequence from (1.).

For L30:

Number of expected random matches: 1.2 \*10^-9

For L10:

Number of expected random matches: 1325

With these estimates and the results from 6., what can you conclude about the reliability of the data in the two different fastq files (L10.fq.gz, L30.fq.gz). If you were to produce new data, what would be your strategy?

Based on this the data from the L10 file is not reliable since it will align randomly all over the reference genome. The reads of length are 30 reliable since the reads will not be aligned randomly. If I were to produce new data, I would make sure that the reads are of appropriate length. In some cases, one might have to consider including shorter reads (20-24 bp) but that is only when it is not possible to get longer and more reliable ones. It would always be the goal to keep reads as long as possible.