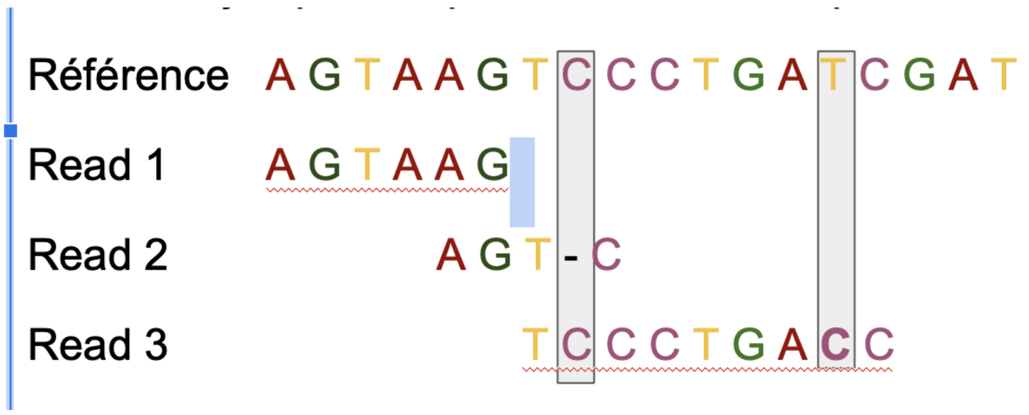
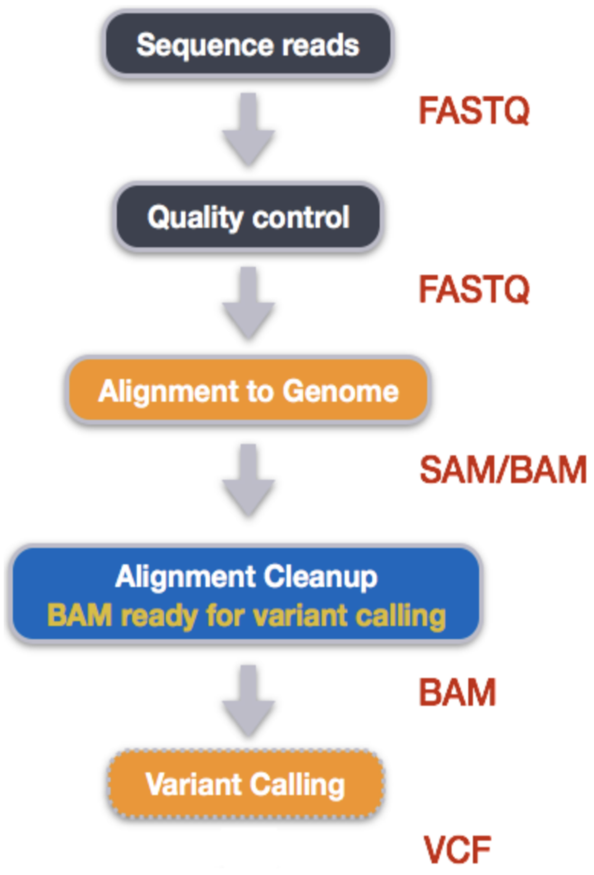
Introduction

The alignment makes it possible to determine the origin of our reads on the reference genome.
Naively, it consists of comparing a read at each position of the reference genome until a good match is found.

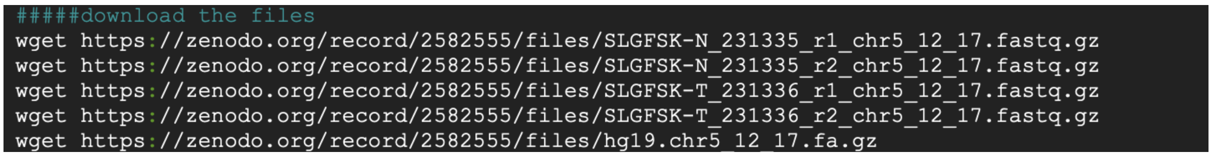


To do so, we will follow the pipeline below



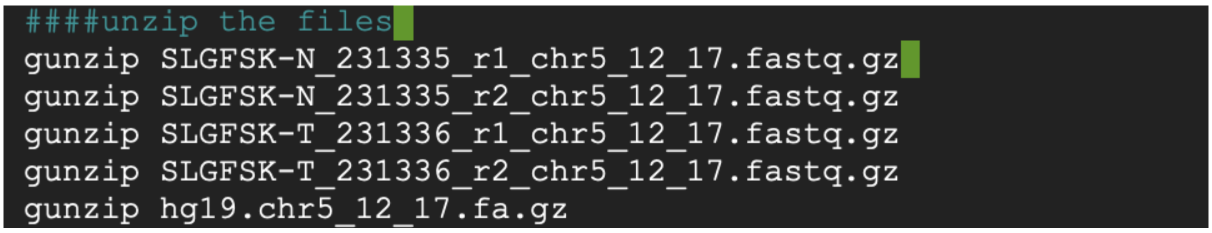
Materials and methods

To perform our analysis, we have a reference genome and reads which are already trimmed.
The quality contrôle was done.
However, this data must be downloaded.
So we will download them using the wget command



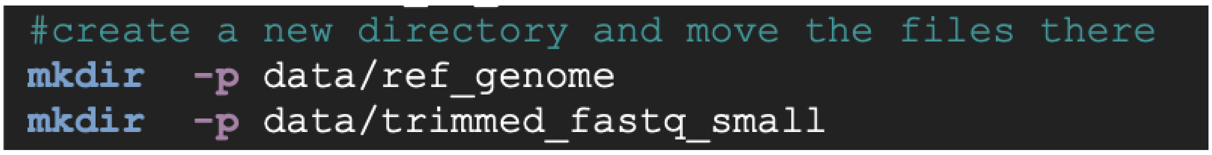
```
#####download the files
wget https://zenodo.org/record/2582555/files/SLGFSK-N_231335_r1_chr5_12_17.fastq.gz
wget https://zenodo.org/record/2582555/files/SLGFSK-N_231335_r2_chr5_12_17.fastq.gz
wget https://zenodo.org/record/2582555/files/SLGFSK-T_231336_r1_chr5_12_17.fastq.gz
wget https://zenodo.org/record/2582555/files/SLGFSK-T_231336_r2_chr5_12_17.fastq.gz
wget https://zenodo.org/record/2582555/files/hg19.chr5_12_17.fa.gz
```

As we can see, this data is in zip format. We will then unzip them.



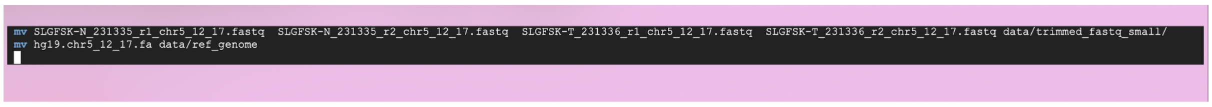
```
####unzip the files
gunzip SLGFSK-N_231335_r1_chr5_12_17.fastq.gz
gunzip SLGFSK-N_231335_r2_chr5_12_17.fastq.gz
gunzip SLGFSK-T_231336_r1_chr5_12_17.fastq.gz
gunzip SLGFSK-T_231336_r2_chr5_12_17.fastq.gz
gunzip hg19.chr5_12_17.fa.gz
```

We will create two directories. One for the reference genome and another for the reads to be mapped.



```
#create a new directory and move the files there
mkdir  -p data/ref_genome
mkdir  -p data/trimmed_fastq_small
```

We will then organize our directory by moving our data into these subdirectories.

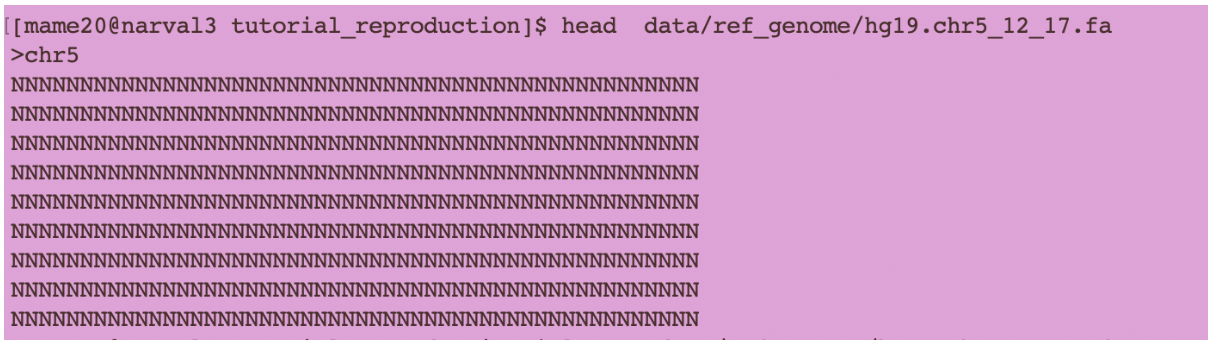


```
mv SLGFSK-N_231335_r1_chr5_12_17.fastq  SLGFSK-N_231335_r2_chr5_12_17.fastq  SLGFSK-T_231336_r1_chr5_12_17.fastq  SLGFSK-T_231336_r2_chr5_12_17.fastq data/trimmed_fastq_small/
mv hg19.chr5_12_17.fa data/ref_genome
```

We will create a directory for the results
####create directories for results
mkdir -p results/sam results/bam results/bcf results/vcf

Now that we have our fasta reads and our reference genome, we will move on to analysis. We will first create a directory for the results and a subdirectory for each extension. We have the bam, sam and bcf ,vcf extensions.

When looking at the reference genome, we saw that we have NNNNN instead of the nucleotides. So we will remove them before going further.



```
[[mame20@narval3 tutorial_reproduction]$ head  data/ref_genome/hg19.chr5_12_17.fa
>chr5
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

We will use this command:

```
awk '/^>/ {printf("%s%s\t",(N>0?"\n":""),$0);N++;next;} {printf("%s",$0);} END
{printf("\n");}' < data/ref_genome/hg19.chr5_12_17.fa | sed 's/N//g' | tr "\t" "\n" >
hg19.chr5_12_17_new.fa
```

Now we have the following as a result:

```
head  data/ref_genome/hg19.chr5_12_17_new.fa
```

```
>chr5
taaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccc
taaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccc
taaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaa
```

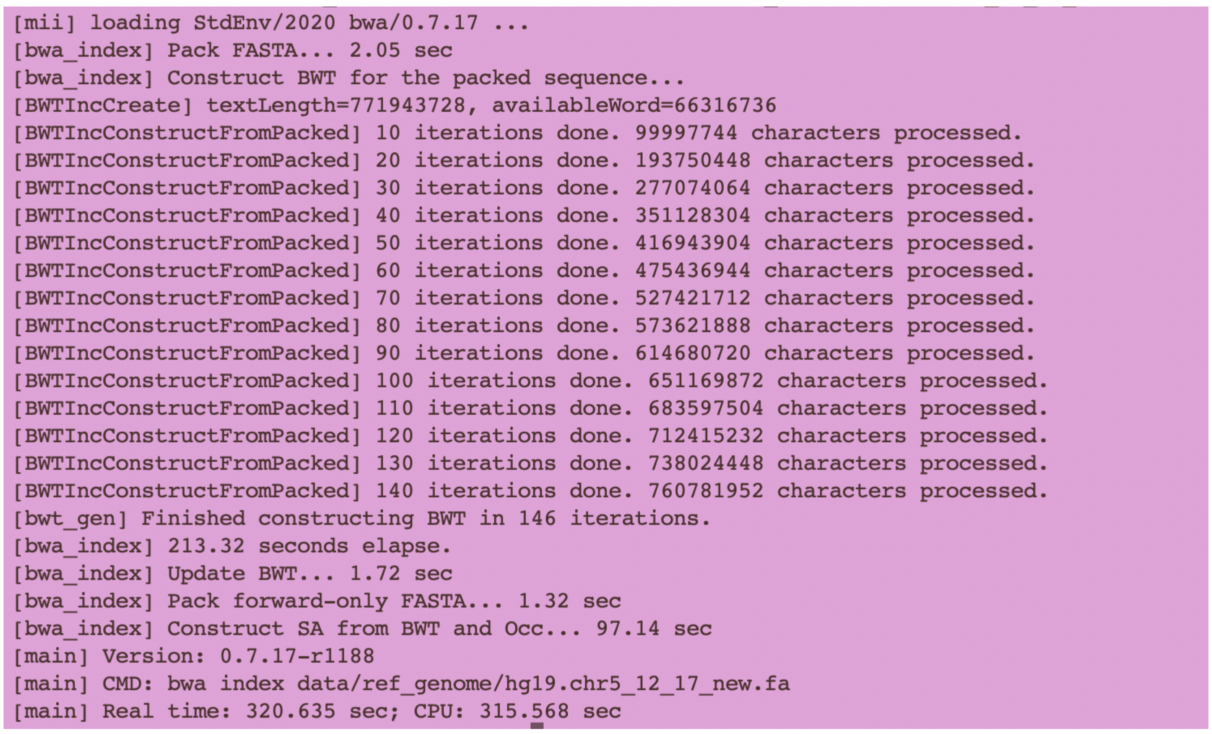
We can continue our analysis

Index the reference genome

outil bwa

```
bwa index data/ref_genome/hg19.chr5_12_17_new.fa
```

After running this command, this is what appears on the command line. Let's have a look at this output.



```
[mii] loading StdEnv/2020 bwa/0.7.17 ...
[bwa_index] Pack FASTA... 2.05 sec
[bwa_index] Construct BWT for the packed sequence...
[BWTIncCreate] textLength=771943728, availableWord=66316736
[BWTIncConstructFromPacked] 10 iterations done. 99997744 characters processed.
[BWTIncConstructFromPacked] 20 iterations done. 193750448 characters processed.
[BWTIncConstructFromPacked] 30 iterations done. 277074064 characters processed.
[BWTIncConstructFromPacked] 40 iterations done. 351128304 characters processed.
[BWTIncConstructFromPacked] 50 iterations done. 416943904 characters processed.
[BWTIncConstructFromPacked] 60 iterations done. 475436944 characters processed.
[BWTIncConstructFromPacked] 70 iterations done. 527421712 characters processed.
[BWTIncConstructFromPacked] 80 iterations done. 573621888 characters processed.
[BWTIncConstructFromPacked] 90 iterations done. 614680720 characters processed.
[BWTIncConstructFromPacked] 100 iterations done. 651169872 characters processed.
[BWTIncConstructFromPacked] 110 iterations done. 683597504 characters processed.
[BWTIncConstructFromPacked] 120 iterations done. 712415232 characters processed.
[BWTIncConstructFromPacked] 130 iterations done. 738024448 characters processed.
[BWTIncConstructFromPacked] 140 iterations done. 760781952 characters processed.
[bwt_gen] Finished constructing BWT in 146 iterations.
[bwa_index] 213.32 seconds elapse.
[bwa_index] Update BWT... 1.72 sec
[bwa_index] Pack forward-only FASTA... 1.32 sec
[bwa_index] Construct SA from BWT and Occ... 97.14 sec
[main] Version: 0.7.17-r1188
[main] CMD: bwa index data/ref_genome/hg19.chr5_12_17_new.fa
[main] Real time: 320.635 sec; CPU: 315.568 sec
```

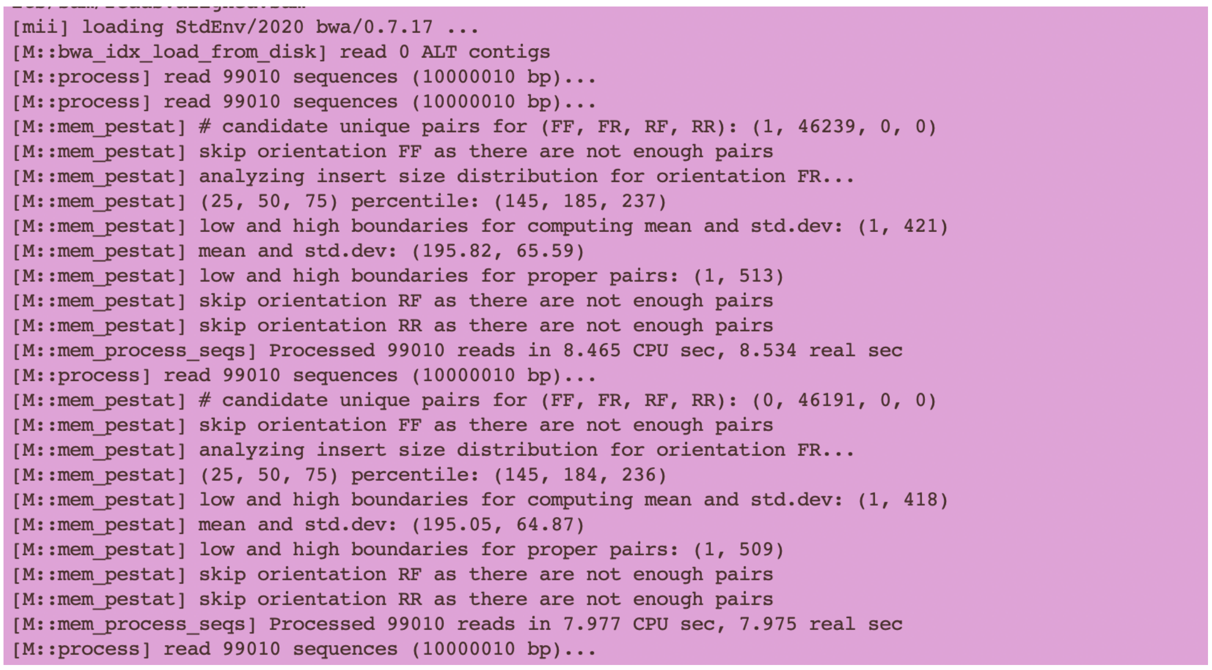
Align reads to reference genome

To align the reads, we will use the bwa tool

So we are going to align our reads at the same time and to do this we are going to use the -M tag. The command is as follows.

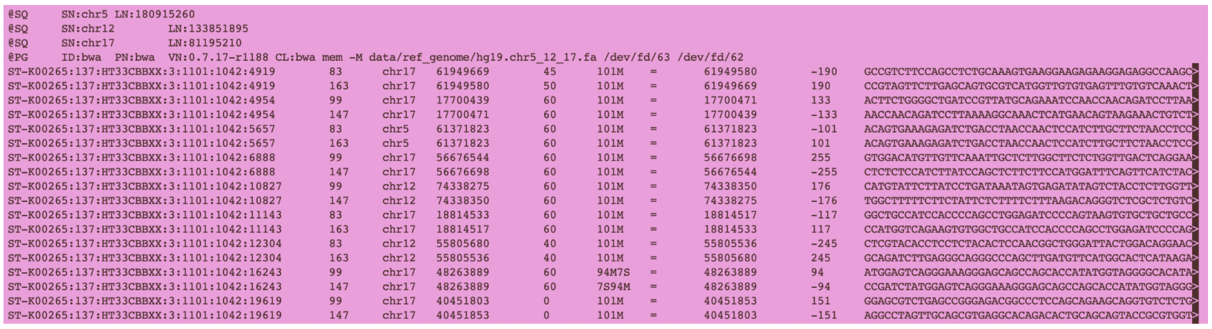
```
bwa mem -M data/ref_genome/hg19.chr5_12_17_new.fa <(cat
data/trimmed_fastq_small/SLGFSK-N_231335_r1_chr5_12_17.fastq
data/trimmed_fastq_small/SLGFSK-T_231336_r1_chr5_12_17.fastq) <(cat
data/trimmed_fastq_small/SLGFSK-N_231335_r2_chr5_12_17.fastq
data/trimmed_fastq_small/SLGFSK-T_231336_r2_chr5_12_17.fastq) >
results/sam/reads.aligned.sam
```

The output:



```
[mii] loading StdEnv/2020 bwa/0.7.17 ...
[M::bwa_idx_load_from_disk] read 0 ALT contigs
[M::process] read 99010 sequences (10000010 bp)...
[M::process] read 99010 sequences (10000010 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (1, 46239, 0, 0)
[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, 237)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 421)
[M::mem_pestat] mean and std.dev: (195.82, 65.59)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 513)
[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 99010 reads in 8.465 CPU sec, 8.534 real sec
[M::process] read 99010 sequences (10000010 bp)...
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (0, 46191, 0, 0)
[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, 184, 236)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 418)
[M::mem_pestat] mean and std.dev: (195.05, 64.87)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 509)
[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 99010 reads in 7.977 CPU sec, 7.975 real sec
[M::process] read 99010 sequences (10000010 bp)...
```

This analysis took nearly 2 hours of time
The resulting sam file looks like this



```
@SQ     SN:chr5 LN:180915260
@SQ     SN:chr12        LN:133851895
@SQ     SN:chr17        LN:81195210
@PG     ID:bwa  PN:bwa  VN:0.7.17-r1188 CL:bwa mem -M data/ref_genome/hg19.chr5_12_17.fa /dev/fd/63 /dev/fd/62
ST-K00265:137:HT33CBBXX:3:1101:1042:4919        83      chr17   61949669        45      101M    =       61949580        -190    GCCGTCTTCCAGCCTCTGCAAAGTGAAGGAAGAGAAGGAGAGGCCAAGC>
ST-K00265:137:HT33CBBXX:3:1101:1042:4919        163     chr17   61949580        50      101M    =       61949669        190     CCGTAGTTCTTGAGCAGTGCGTCATGGTTGTGTGAGTTTGTGTCAAACT>
ST-K00265:137:HT33CBBXX:3:1101:1042:4954        99      chr17   17700439        60      101M    =       17700471        133     ACTTCTGGGGCTGATCCGTTATGCAGAAATCCAACCAACAGATCCTTAA>
ST-K00265:137:HT33CBBXX:3:1101:1042:4954        147     chr17   17700471        60      101M    =       17700439        -133    AACCAACAGATCCTTAAAAGGCAAACTCATGAACAGTAAGAAACTGTCT>
ST-K00265:137:HT33CBBXX:3:1101:1042:5657        83      chr5    61371823        60      101M    =       61371823        -101    ACAGTGAAAGAGATCTGACCTAACCAACTCCATCTTGCTTCTAACCTCC>
ST-K00265:137:HT33CBBXX:3:1101:1042:5657        163     chr5    61371823        60      101M    =       61371823        101     ACAGTGAAAGAGATCTGACCTAACCAACTCCATCTTGCTTCTAACCTCC>
ST-K00265:137:HT33CBBXX:3:1101:1042:6888        99      chr17   56676544        60      101M    =       56676698        255     GTGGACATGTTGTTCAAATTGCTCTTGGCTTCTCTGGTTGACTCAGGAA>
ST-K00265:137:HT33CBBXX:3:1101:1042:6888        147     chr17   56676698        60      101M    =       56676544        -255    CTCTCTCCATCTTATCCAGCTCTTCTTCCATGGATTTCAGTTCATCTAC>
ST-K00265:137:HT33CBBXX:3:1101:1042:10827       99      chr12   74338275        60      101M    =       74338350        176     CATGTATTCTTATCCTGATAAATAGTGAGATATAGTCTACCTCTTGGTT>
ST-K00265:137:HT33CBBXX:3:1101:1042:10827       147     chr12   74338350        60      101M    =       74338275        -176    TGGCTTTTTCTTCTATTCTCTTTTCTTTAAGACAGGGTCTCGCTCTGTC>
ST-K00265:137:HT33CBBXX:3:1101:1042:11143       83      chr17   18814533        60      101M    =       18814517        -117    GGCTGCCATCCACCCCAGCCTGGAGATCCCCAGTAAGTGTGCTGCTGCC>
ST-K00265:137:HT33CBBXX:3:1101:1042:11143       163     chr17   18814517        60      101M    =       18814533        117     CCATGGTCAGAAGTGTGGCTGCCATCCACCCCAGCCTGGAGATCCCCAG>
ST-K00265:137:HT33CBBXX:3:1101:1042:12304       83      chr12   55805680        40      101M    =       55805536        -245    CTCGTACACCTCCTCTACACTCCAACGGCTGGGATTACTGGACAGGAAC>
ST-K00265:137:HT33CBBXX:3:1101:1042:12304       163     chr12   55805536        40      101M    =       55805680        245     GCAGATCTTGAGGGCAGGGCCCAGCTTGATGTTCATGGCACTCATAAGA>
ST-K00265:137:HT33CBBXX:3:1101:1042:16243       99      chr17   48263889        60      94M7S   =       48263889        94      ATGGAGTCAGGGAAAGGGAGCAGCCAGCACCATATGGTAGGGGCACATA>
ST-K00265:137:HT33CBBXX:3:1101:1042:16243       147     chr17   48263889        60      7S94M   =       48263889        -94     CCGATCTATGGAGTCAGGGAAAGGGAGCAGCCAGCACCATATGGTAGGG>
ST-K00265:137:HT33CBBXX:3:1101:1042:19619       99      chr17   40451803        0       101M    =       40451853        151     GGAGCGTCTGAGCCGGGAGACGGCCCTCCAGCAGAAGCAGGTGTCTCTG>
ST-K00265:137:HT33CBBXX:3:1101:1042:19619       147     chr17   40451853        0       101M    =       40451803        -151    AGGCCTAGTTGCAGCGTGAGGCACAGACACTGCAGCAGTACCGCGTGGT>
```

Now that we have our sam files, we will change it to bam as it is lighter for next analysis.
To do so, we will use samtool. samtool is …. we will use the tag view

```
samtools view -S -b results/sam/reads.aligned.sam > results/bam/reads.aligned.bam
```

Sort BAM file by coordinates

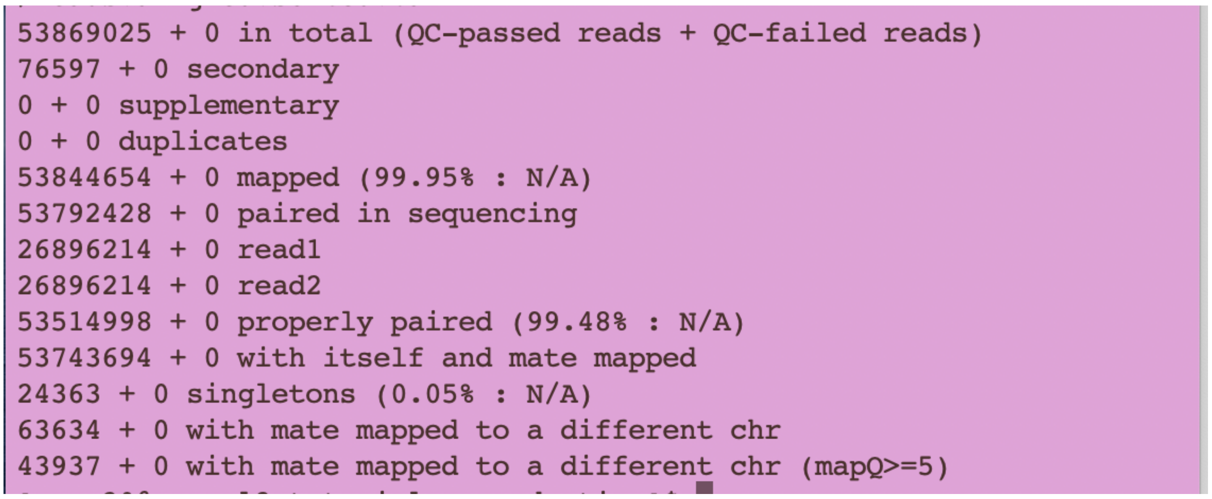
```
samtools sort -o results/bam/reads.aligned.sorted.bam results/bam/reads.aligned.bam
```

This command will output the following:

```
merging from 20 files and 1 in-memory blocks...
```

Let's have a look to that outputed file

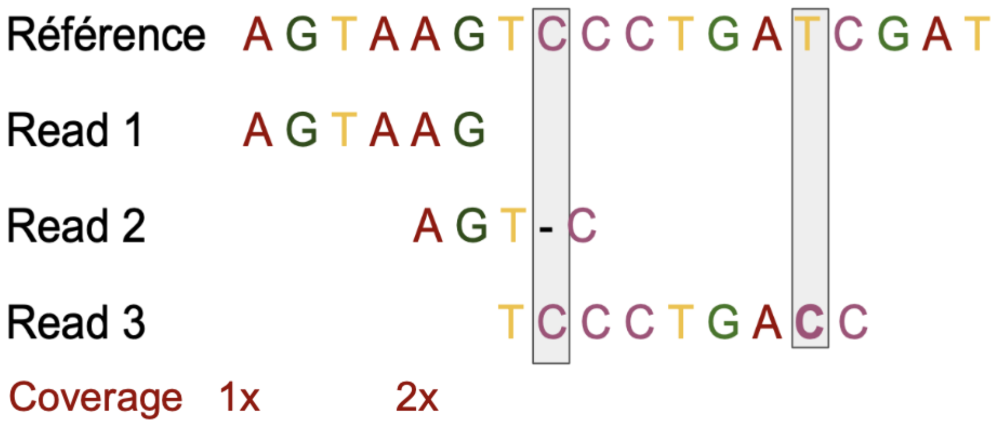
```
samtools flagstat results/bam/reads.aligned.sorted.bam
```



```
53869025 + 0 in total (QC-passed reads + QC-failed reads)
76597 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
53844654 + 0 mapped (99.95% : N/A)
53792428 + 0 paired in sequencing
26896214 + 0 read1
26896214 + 0 read2
53514998 + 0 properly paired (99.48% : N/A)
53743694 + 0 with itself and mate mapped
24363 + 0 singletons (0.05% : N/A)
63634 + 0 with mate mapped to a different chr
43937 + 0 with mate mapped to a different chr (mapQ>=5)
```

Calculate the read coverage of positions in the genome

With the following image we can have an idea on what coverage means. After that we will calculate it using bfctools.



```
bcftools mpileup -O b -o results/bcf/reads_raw.bcf -f
data/ref_genome/hg19.chr5_12_17_new.fa  results/bam/reads.aligned.sorted.bam
```

Detect the single nucleotide variants

We will try to detect the SNV using bfctools and by giving the ploidy (regarding the organism)

```
bcftools call --ploidy 1 -m -v -o results/vcf/reads_variants.vcf results/bcf/reads_raw.bcf
```

Let’s have a look to the number of variants in this file before filtering

```
 228363  2283524 27282461 results/vcf/reads_variants.vcf
```

Filter and report the SNV variants in variant calling format

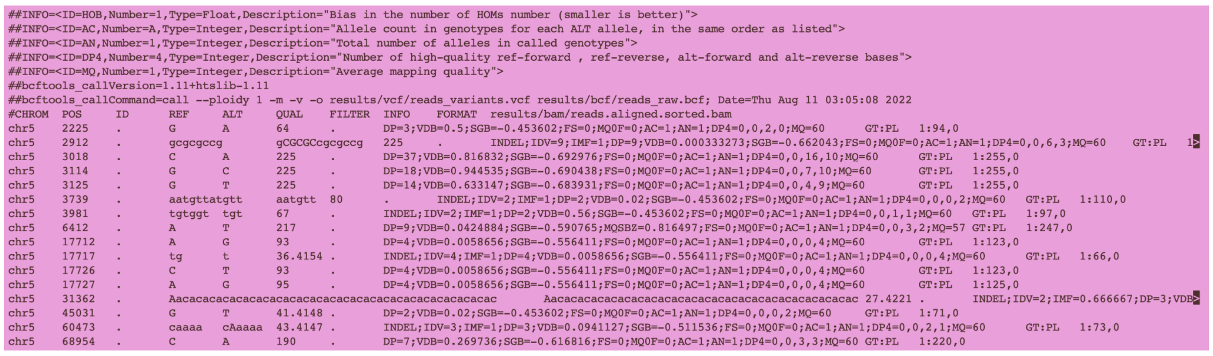
```
vcfutils.pl varFilter results/vcf/reads_variants.vcf  > results/vcf/reads_final_variants.vcf
```

After filtering, we have this:

```
 140167  1401564 18143976 results/vcf/reads_final_variants.vcf
```

Let’s have a look to this file. We have the chromosome, the position, the id, the reference allele, the alternate allele …

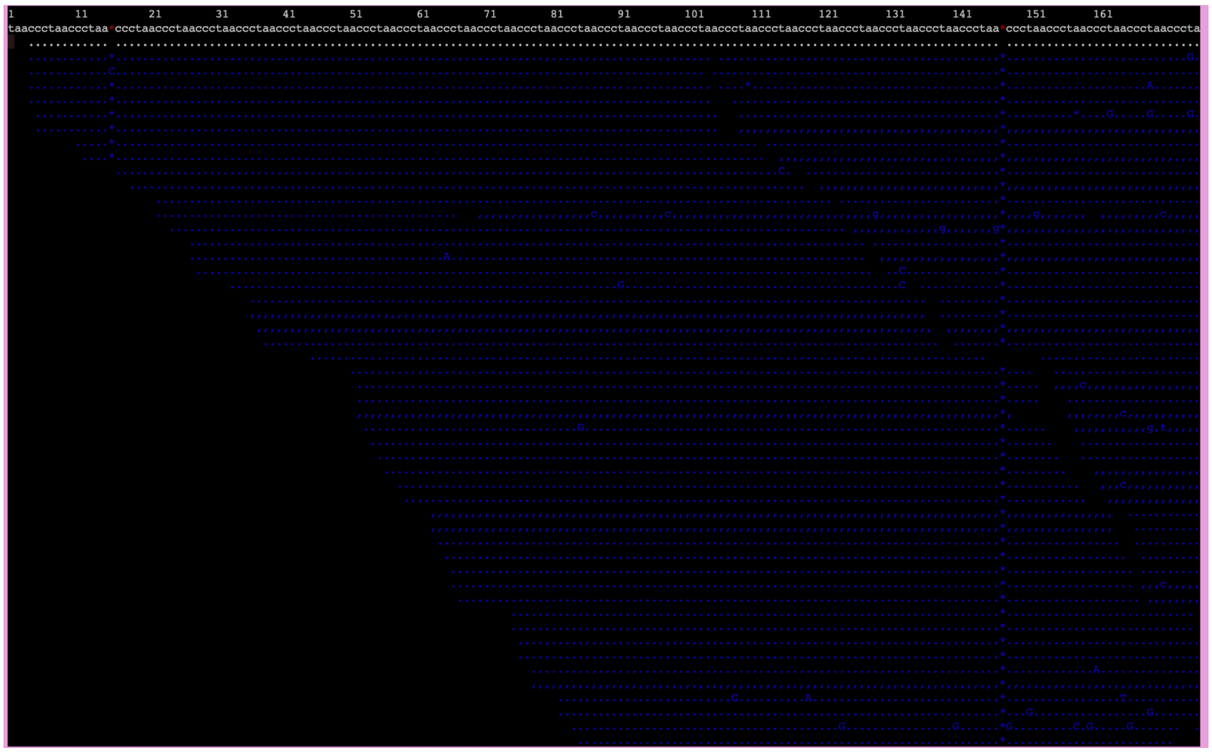
```
less -S results/vcf/reads_final_variants.vcf
```



```
##INFO=<ID=HOB,Number=1,Type=Float,Description="Bias in the number of HOMs number (smaller is better)">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward and alt-reverse bases">
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callVersion=1.11+htslib-1.11
##bcftools_callCommand=call --ploidy 1 -m -v -o results/vcf/reads_variants.vcf results/bcf/reads_raw.bcf; Date=Thu Aug 11 03:05:08 2022
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  results/bam/reads.aligned.sorted.bam
chr5    2225    .       G       A       64      .       DP=3;VDB=0.5;SGB=-0.453602;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,2,0;MQ=60      GT:PL   1:94,0
chr5    2912    .       gcgcgccg        gCGCGCcgcgccg   225     .       INDEL;IDV=9;IMF=1;DP=9;VDB=0.000333273;SGB=-0.662043;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,6,3;MQ=60    GT:PL   1>
chr5    3018    .       C       A       225     .       DP=37;VDB=0.816832;SGB=-0.692976;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,16,10;MQ=60      GT:PL   1:255,0
chr5    3114    .       G       C       225     .       DP=18;VDB=0.944535;SGB=-0.690438;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,7,10;MQ=60       GT:PL   1:255,0
chr5    3125    .       G       T       225     .       DP=14;VDB=0.633147;SGB=-0.683931;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,4,9;MQ=60        GT:PL   1:255,0
chr5    3739    .       aatgttatgtt     aatgtt  80      .       INDEL;IDV=2;IMF=1;DP=2;VDB=0.02;SGB=-0.453602;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,2;MQ=60   GT:PL   1:110,0
chr5    3981    .       tgtggt  tgt     67      .       INDEL;IDV=2;IMF=1;DP=2;VDB=0.56;SGB=-0.453602;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,1,1;MQ=60   GT:PL   1:97,0
chr5    6412    .       A       T       217     .       DP=9;VDB=0.0424884;SGB=-0.590765;MQSBZ=0.816497;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,3,2;MQ=57 GT:PL   1:247,0
chr5    17712   .       A       G       93      .       DP=4;VDB=0.0058656;SGB=-0.556411;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,4;MQ=60        GT:PL   1:123,0
chr5    17717   .       tg      t       36.4154 .       INDEL;IDV=4;IMF=1;DP=4;VDB=0.0058656;SGB=-0.556411;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,4;MQ=60      GT:PL   1:66,0
chr5    17726   .       C       T       93      .       DP=4;VDB=0.0058656;SGB=-0.556411;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,4;MQ=60        GT:PL   1:123,0
chr5    17727   .       A       G       95      .       DP=4;VDB=0.0058656;SGB=-0.556411;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,4;MQ=60        GT:PL   1:125,0
chr5    31362   .       Aacacacacacacacacacacacacacacacacacacacacacac       Aacacacacacacacacacacacacacacacacacacacacacacac 27.4221 .       INDEL;IDV=2;IMF=0.666667;DP=3;VDB>
chr5    45031   .       G       T       41.4148 .       DP=2;VDB=0.02;SGB=-0.453602;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,2;MQ=60     GT:PL   1:71,0
chr5    60473   .       caaaa   cAaaaa  43.4147 .       INDEL;IDV=3;IMF=1;DP=3;VDB=0.0941127;SGB=-0.511536;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,2,1;MQ=60      GT:PL   1:73,0
chr5    68954   .       C       A       190     .       DP=7;VDB=0.269736;SGB=-0.616816;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,3,3;MQ=60 GT:PL   1:220,0
```

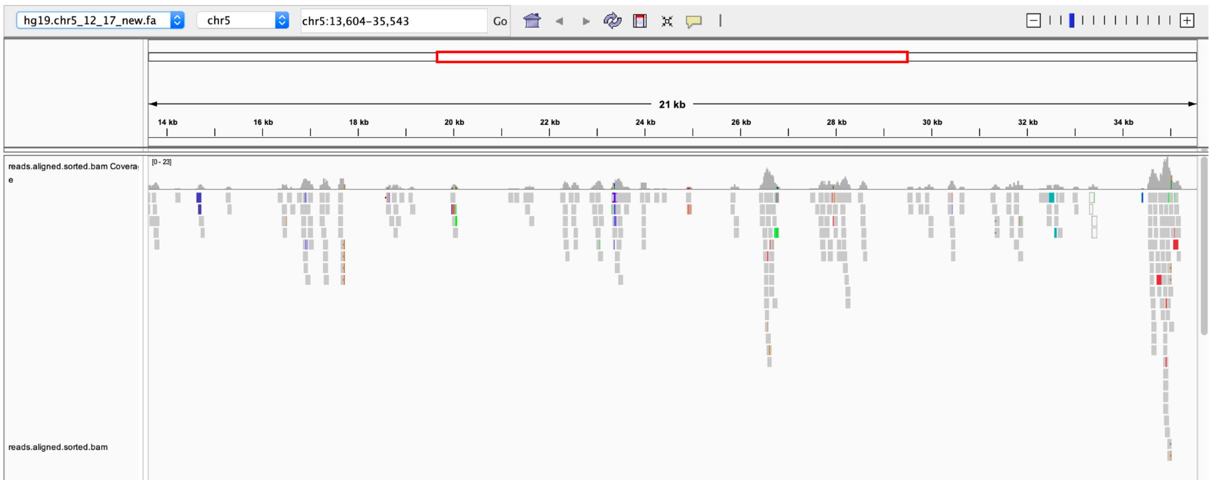
Assess the alignment (visualization) - optional step

Here we can see our alignment. We can see the variation with the letters modified. When we have , or . it's aligned to the forward or the reverse.



```
1        11        21       31       41       51       61       71       81       91       101      111      121      131      141       151      161
taaccctaaccctaa*ccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaaccctaa*ccctaaccctaaccctaaccctaaccctaaccta
```

Here we can visualize using the IGV tool. The gray one refers to the genome.



Conclusion

We performed variant calling using several tools. The process allows us to understand the variant calling pipeline and to push our limits.