Chek de quality of fastq files:

FASTQC

```
< fastqc input_file.fastq >
```

- HTML file with the interactive report;
- Zip file containing the detailed data of the report.
- Source code available at: **https://github.com/s-andrews/FastQC.git**

Remove low-quality reads:

TRIMMOMATIC

```
< java -jar trimmomatic.jar PE -phred30 input_forward.fastq input\_reverse.fastq
output_forward_paired.fastq ILLUMINACLIP:adapters.fa:2:30:10 LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:36 >
```

- PE: Mode for paired-end data;
- phred30: Specifies the quality encoding;
- ILLUMINACLIP:adapters.fa:2:30:10: Removes sequence adapters:

  adapters.fa: File with adapter sequences;

  2:Number of allowed mismatches;

  30: Threshold score for trimming adapters;

  10: Minimum fragment length after adapter removal;
- LEADING:3: Removes low-quality bases (score below 3) at the start of the read;
- TRAILING:3: Removes low-quality bases at the end of the read;
- SLIDINGWINDOW:4:15: Checks the average quality within a sliding window (4 bases). Trims when the average is below 15;
- MINLEN:36: Discards reads shorter than 36 base Source code available at: **https://github.com/usadellab/Trimmomatic.git**

Removing duplicates:

PICARDTOOLS

```
< java -jar picard.jar MarkDuplicates INPUT=input.bam OUTPUT=deduplicated.bam
METRICS_FILE=metrics.txt >
```

- MarkDuplicates: Call to exclude duplicate regions;
- INPUT: Specifies the input BAM file;
- OUTPUT: Specifies the output BAM file with duplicates;
- METRICS_FILE: Output file containing duplication metrics.

Source code available at: **https://github.com/broadinstitute/picard.git**

Mapping against the reference genome - H37Rv

BWA-MEM

```
< bwa mem -t 8 -M reference.fa reads_1.fq reads_2.fq > aligned.sam >
```

- -t: Number of threads for parallelization;
- -M: Marks secondary reads as secondary alignments.

Source code available at: **https://github.com/bwa-mem2/bwa-mem2.git**

Variant calling (SNPs and indels):

SAMTOOLS

```
< samtools mpileup -uf reference.fasta input.bam | bcftools call -mv -Ov > variants.vcf >
```

- -u: Generates output in uncompressed format (to be processed by other tools);
- -f reference.fasta: Specifies the reference genome;
- input.bam: Input BAM file;
- -m: Performs variant calling using the multiallelic caller;
- -v: Reports only variants (excludes unchanged regions);
- -Ov: Specifies output in text-based VCF format.

Source code available at: **https://github.com/samtools/samtools.git**

Variant calling (SNPs and indels):

GATK

First part: Variant Calling:

```
< gatk HaplotypeCaller -R reference.fasta -I sorted_output.bam -O raw_variants.vcf >
```

- HaplotypeCaller: Identifies and calls all variants;
- -R reference.fasta: FASTA file containing the reference genome;

Second part: Variant Filtering:

```
< gatk VariantFiltration -R reference.fasta -V raw_variants.vcf -O filtered_variants.vcf >
```

- VariantFiltration: Applies filtering criteria to variant calls;
- -R reference.fasta: Same reference used in the previous step;
- -V raw_variants.vcf: Input VCF file containing raw variants;
- -O filtered_variants.vcf: Output VCF file.
- Source code available at: **https://github.com/broadinstitute/gatk.git**

Perform mutation prediction and describe the lineage:

TB-PROFILER

```
< tb-profiler profile -1 input_forward.fq -2 input_reverse.fq -o tbprofiler_output >
```

- -1 and -2: Specify the input reads;
- -o: Defines the prefix for the output files.

Source code available at: **https://github.com/jodyphelan/TBProfiler.git**