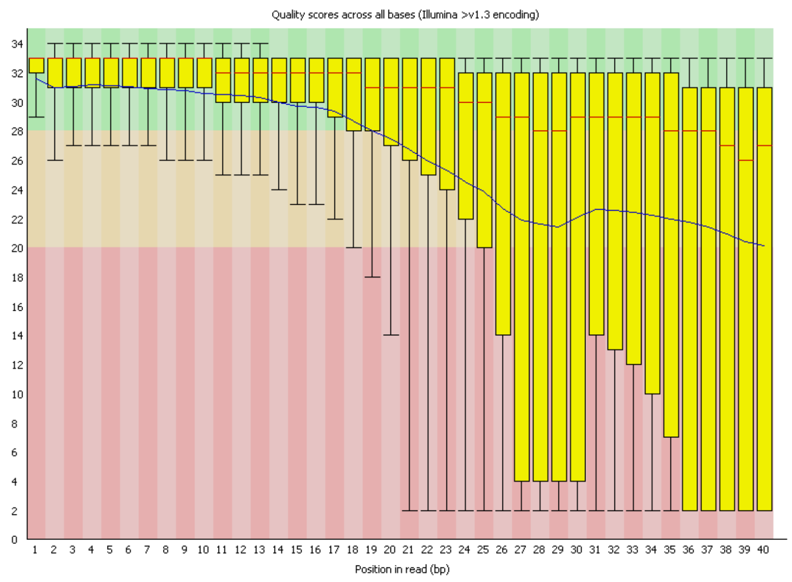
Computational Practical 2: Accessing Data and Quality Control
Module Developers: Dr. Stanford Kwenda and Collins Kigen

2.0 Learning outcomes

2.1 Introduction
A typical whole genome sequencing process involves genomic DNA isolation, library preparation and sequencing. Errors can be introduced during the library preparation and sequencing steps which may cause inaccurate representations of the original DNA sequences.

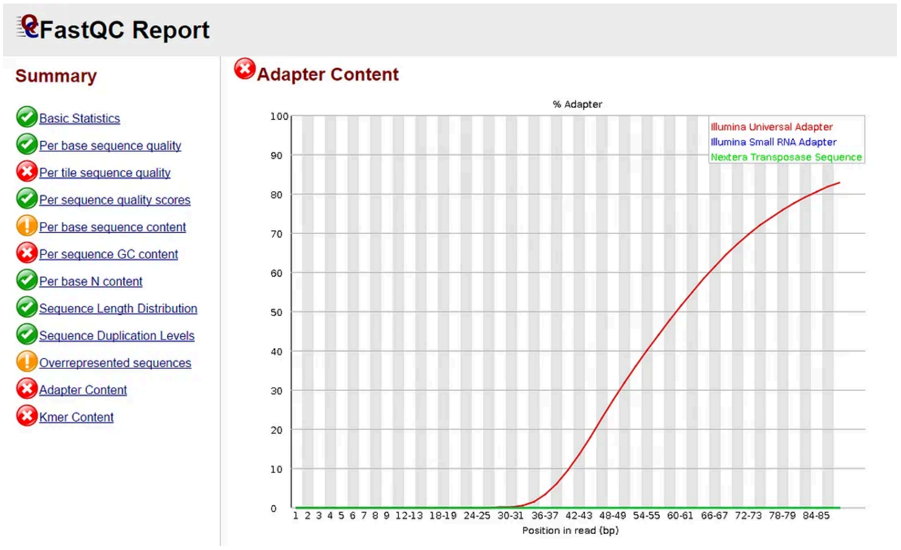
Potential problems or quality related issues with raw next generation sequencing (NGS) data include:

1. Low base quality - sequencing technologies can produce reads with varying quality



https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/2%20Per%20Base%20Sequence%20Quality.html

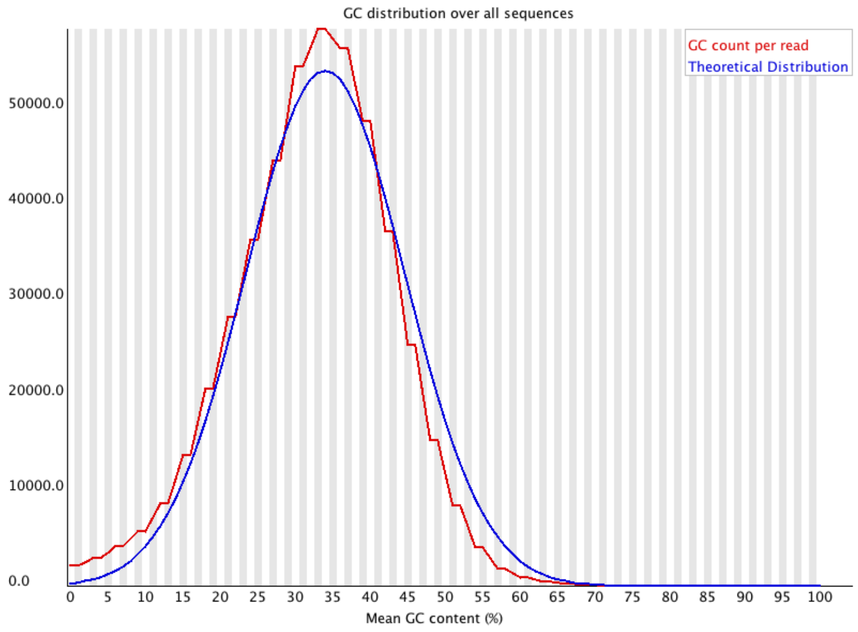
2. Contamination with adapter sequences


FastQC Report
Summary
Adapter Content
% Adapter
Illumina Universal Adapter
Illumina Small RNA Adapter
Nextera Transposase Sequence
Position in read (bp)

✓ 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
✗ Kmer Content

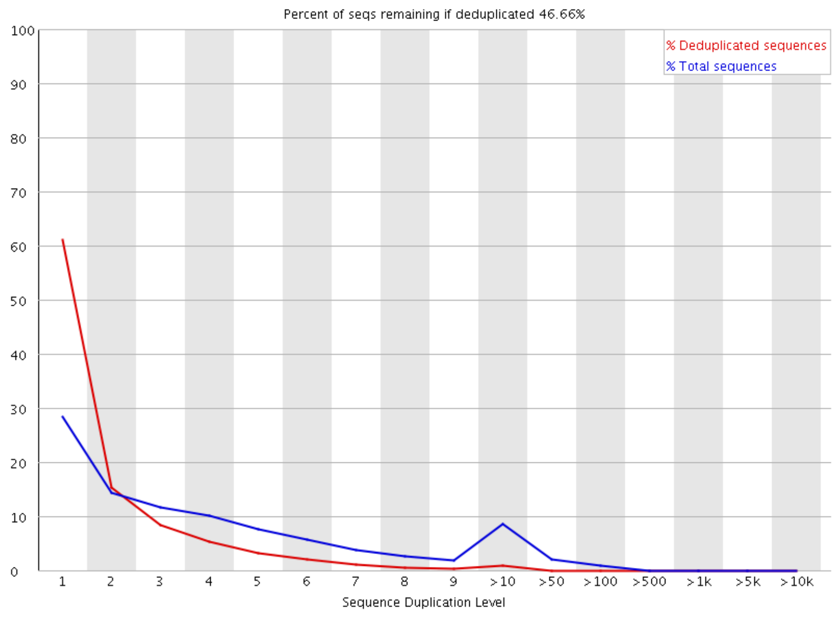
https://medium.com/@shilparaopradeep/merge-paired-end-reads-of-ancient-dna-with-bbmerge-improves-read-quality-ddc146187064

3. Base composition biases
4. GC content distribution



https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/5%20Per%20Sequence%20GC%20Content.html

5. Sequence duplication



https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/8%20Duplicate%20Sequences.html

Several tools are available for performing quality control and preprocessing of fastq data. Some popular standalone tools tools include:

1. FASTQC - quality control tool for visualizing key quality profiling metrics
2. Cutadapt - adapter removal and read filtering
3. Trimmomatic - adapter removal, read pruning and filtering e.g. using sliding window cutting
4. TrimGalore - a wrapper tool around cutadapt and FASTQC

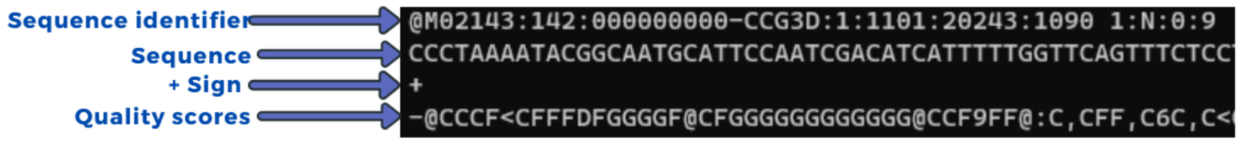
In recent years tools that integrate both quality control and read filtering have been developed, such as TrimGalore, AfterQC, fastp, etc. In this practical we will introduce and use fastp for raw data quality control, filtering, adapter removal and base correction for fastq data. Fastp is an ultra-fast multithreaded tool that incorporates most features of fastqc, cutadapt, trimmomatic and AfterQC, and is best suited for short reads, but with basic support for long read data QC.

For visualization of quality metrics we will use MultiQC

The preprocessing of FASTQ data, which means removing adapter contamination, filtering low-quality reads, and correcting wrongly represented bases, is an indispensable but resource intensive part of sequencing data analysis.

2.1 Fastq format

In NGS, fastq data format is broadly adopted as the standard for the raw output data regardless of the sequencing platform and the sequencing throughput.



```
@M02143:142:000000000-CCG3D:1:1101:20243:1090 1:N:0:9
CCCTAAAATACGGCAATGCATTCCAATCGACATCATTTTTGGTTCAGTTTCTCC
+
-@CCCF<CFFFDFGGGGF@CFGGGGGGGGGGGG@CCF9FF@:C,CFF,C6C,C<
```

2.2. (Phred) Quality Scores Interpretation

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%

FASTQ data need to go through quality control and a series of preprocessing steps before they can enter the downstream analysis to ensure that clean and accurate reads are processed downstream.

2.3 Common QC metrics for raw data

Removal of adapter contamination

- Sequence-matching based adapter trimming
  - Tools: Cutadapt, Trimmomatic etc.
- Adapter trimming based on finding overlaps of read-pairs
  - Tools: Fastp, AfterQC
  - Automatic adapter detection

Base correction

- Mainly applied to paired-end data

- Only base pairs with an imbalance quality score are corrected

Improving read quality

- Sliding window quality pruning method
    - Drops low quality bases in each reads 5` or 3` region
- Filtering reads using a low-quality base percentage, N base number and read length

Trimming of polyG and polyX tails

Some Illumina sequencing platforms such as the NextSeq series can produce reads with polyG tails due to their fluorescence chemistry resulting in some T and C bases being interpreted as Gs in read tails when signal strength of each cluster becomes weaker in subsequent cycles. The polyG tail issue can result in a serious base content separation problem, meaning that A and T or C and G have substantially different base content ratios.

Overrepresented sequence analysis

Some sequences, or even entire reads, can be overrepresented in FASTQ data. Analysis of these overrepresented sequences provides an overview of certain sequencing artifacts such as PCR over-duplication, polyG tails and adapter contamination.

Hands-on exercise 1: (20 min)

We will now apply some of the concepts that were introduced above and perform QC on a set of raw data from Klebsiella pneumoniae and Staphylococcus aureus.

1. Using fastp and commands provided below perform QC and preprocessing for K. pneumoniae. The output data should be saved in a directory called kpn_qc.

a. First let’s create the output directory

```
kpn_res=~/ACORN_course/cp2/kpn_qc
mkdir -p $kpn_res
```

b. Activate the appropriate conda environment for read QC tools

```
conda activate readQC
```

c. Now create an output directory for fastp results as a sub-directory of the directory we created in step (a) above.

```
clean_reads=$kpn_res/fastp
mkdir -p $clean_reads
```

d. Provide path to the raw reads (define path as is below)

```
raw_reads=/data/acorn_training_course_2024/RawRead/kleb
```

e. Execute the command below to perform QC on all samples in the raw_reads directory

f. Otherwise we can use a handy for loop to process all the raw fastq using a single command

# Execute the for loop to perform QC on all samples in the raw_reads directory

```
threads=6
for fq in $(find $raw_reads -name "*R1*.f*q.gz"); do
      sampleid=$(basename $fq | cut -d_ -f1)
      read1=$(find $raw_reads -name "${sampleid}*R1*f*q.gz")
      read2=$(find $raw_reads -name "${sampleid}*R2*f*q.gz")

      fastp -i "$read1" -I "$read2" \
      -q 20 -l 36 --cut_front -M 10 -W 4 \
      -R "$sampleid" -j $clean_reads/${sampleid}.fastp.json \
      -h $clean_reads/${sampleid}.fastp.html \
     --correction --overrepresentation_analysis --thread $threads \
     -o $clean_reads/${sampleid}.R1.fq.gz -O
$clean_reads/${sampleid}.R2.fq.gz
done >> $clean_reads/qc_step1.log 2>&1
```

g. For QC visualization we will use a tool called multiqc

```
qc_reports=$kpn_res/multiqc
multiqc -f --no-data-dir $clean_reads --outdir $qc_reports
```

Group activity 1: Read QC and filtering (10min):
Navigate to the multiqc output folder and open the report in your browser. In groups of 2 or 3, discuss the following:

1. Number of reads in each sample?
2. Percentage of reads which passed filters in each sample?
3. What was the duplication rate before filtering?
4. The average quality after filtering?
5. What is the average length of the reads?

Hands-on exercise 2: Repeat the above steps for *S. aureus* QC and preprocessing (15 min)

1. You should name the *S. aureus* output directory as staa_qc. All the other paths should remain the same. See below:

```
raw_reads=/data/acorn_training_course_2024/RawRead/saures
sta_res=~/ACORN_course/cp2/staa_qc
```

Any questions?