✅

Quality Control of Raw read data - Downloadable

Assumed prerequisite: Exercises are executed on a Unix operating system, with Micromamba installed - If Conda is installed and you rather want to use this, replace `micromamba` in commands with `conda`

Goal

There are no clear-cut answers for what makes up a great sequencing dataset, as this is entirely dependant on what the sequencing data are being used for. With that being said, there are certain quality parameters one can aim to fulfil to ensure that the quality of the sequencing data does not interfere with analysis and results interpretation. These exercises aims to introduce you to some of quality parameters one can extract from raw sequencing data.

Difficulties

Some tasks are designed with varying levels of difficulties.

Tasks with multiple levels of difficulties **all lead to the same outcome**, yet the approach to solving these differs.

While the easier tasks are faster to solve, you can learn more from solving the harder difficulties, and in some instances to solve issues smarter. Therefore, whenever you face a task with multiple levels of difficulties, you **only** select a **single level** of difficulty and solve those corresponding tasks.

The levels are designed to help you progress through the exercises, so feel free to switch difficulties.

Tasks with varying difficulties are always phrased in a separate coloured bullet point, followed by sub bullets on the approach to solve the task. It is the approach to solving the tasks which presents various levels of difficulties. Difficulty levels in the assignment are designed with the following principles in mind:

- A question with multiple difficulties:
  - Beginner: Tasks at this level of difficulty should not be challenging to complete. If any available, the required information can be found entirely in a handout document.
  - Advanced: Tasks at this level of difficulty require a bit more individual thinking, but most information required can be found in the handout document. This level is intended for participants with limited command line experiences, or fast learners.
  - Expert: Tasks at this level of difficulty require some degree of experimentation in the terminal. When solving expert tasks, attempt to avoid using the handout document until you get stuck for a while. This level is intended for participants who are comfortable learning from trial and error.

Setup

To ensure consistency between training sessions, make sure to follow the setup guide.

Before the exercises, make a folder called `QC/` inside the `BTG/` folder.

```
mkdir -p ~/BTG/QC
```

- Inside the `QC/` folder make two folders, one called `FastQC/` and the other called `fastp/`

Software used

- FastQC

- fastp
- MultiQC

Dataset

The dataset for this exercise consists of compressed raw read sequencing data (in the .fastq.gz format). It is not relevant to run quality assessment **for all samples**. Two or a few samples is sufficient. Moreover, the QC images included in these exercises are not necessarily representative to bacterial isolate data, they are included for demonstrative purposes. You will generate QC parameters on a selection of the included data, but whether the quality is good or not, is up for you and your co-participants to determine.

Introduction

The lab you are affiliated with have received a batch of raw read data for bacterial isolates collected in a neighboring region. You wish to assess whether there are specific resistance genes present in their local samples. However, before you start analysing, you want to ensure that you do not draw false conclusions on the basis of low quality data. Therefore you undertake the task of assessing the quality parameters for determining the presence or absence of known resistance genes in your population. Since you know that your lab routinely generate sequencing libraries from bacterial isolates, you trust their capabilities, and **do not** impose any filtration criteria on the output raw sequencing data.

Preparing environments

Before starting the analysis, lets ensure we have the correct environment loaded into our session.

- Before doing anything head into the following folder: `
- Ensuring you have the required environment
  - Ensure that you can find the relevant environment
  - Create an environment file with the relevant software, then make the environment.
    - Inspect the mamba create help page (Hint: you need the `--file` parameter and at least one other parameter)
    - Copy paste the following into a simple text file called QC.yaml

```
channels:
 - conda-forge
 - bioconda
dependencies:
 - fastqc
 - fastp
 - multiqc
```

    - Create an environment from the QC.yaml and name the environment QC
    - Ensure that you can find your environment

What it takes to search for genes

Usually resistance genes span from hundreds to thousands of nucleotides, this makes detection of your feature genes less sensitive to low quality reads. Thus, searching for overall read quality in this case can be more of a parameter for benchmarking the performance in the lab, which can be usable for assessing issues occurring during sequencing library preparation or in the sequencer itself.

Generating QC reports

We are starting out with the software called FastQC for generating a QC report, so in order to keep things straight lets make a couple of folders for organisation:

- Navigate into the FastQC folder
- Evoke the `fastqc` help page and look under the `SYNOPSIS` section
  - Activate the `QC` environment and call the help page from `fastqc`

- Inspect the FastQC help page by envoking the fastqc command without loading the environment
  - Inspect the help page for `micromamba run`
  - Run `fastqc --help` through `micromamba run` (!Remember to state the environment name)

☐ Based on the `SYNOPSIS` section, how is FastQC supposed to be executed?

- Navigate to the `QC/` folder and execute the `fastqc` command for your selected samples in `~/BTG/SequenceData/`. Don't forget to point the output (`-o`) to the `~/BTG/QC/FastQC/` folder.

▼ Solution

```
micromamba run -n QC fastqc -o FastQC/ /path/2/reads/R1.fastq.gz /path/2/reads/R2.fastq.gz
```

☐ List the files within the `FastQC` folder, what output are generated by FastQC?

☐ How many reports do you gain per single isolate with FastQC?

Report inspection

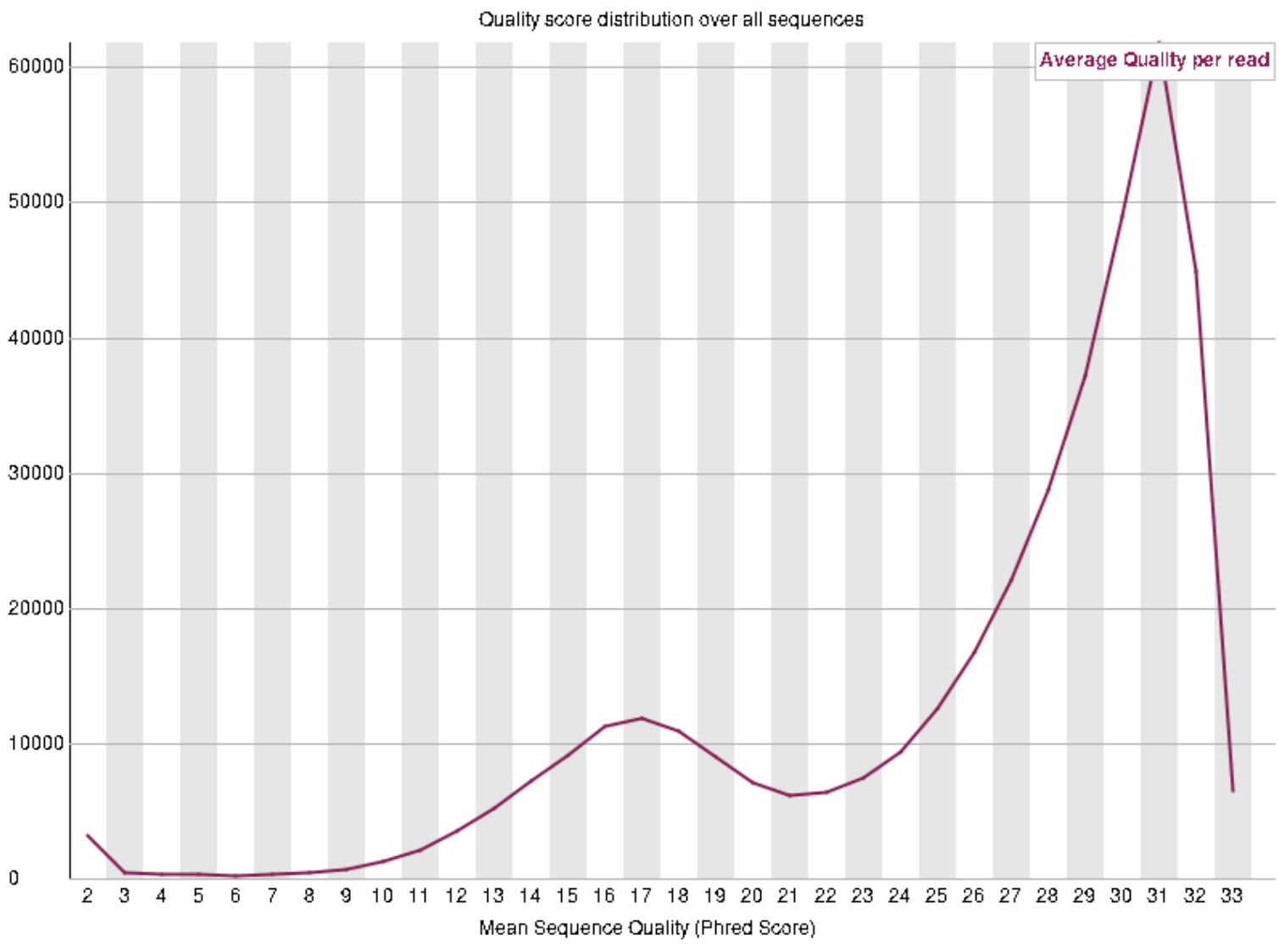
Having generated FastQC reports for our sample collection, lets inspect some over random quality plots and assess their quality.

Discuss

Average read quality

☐ What would be your arguments for using / discarding the following data for detecting presence of resistance genes?

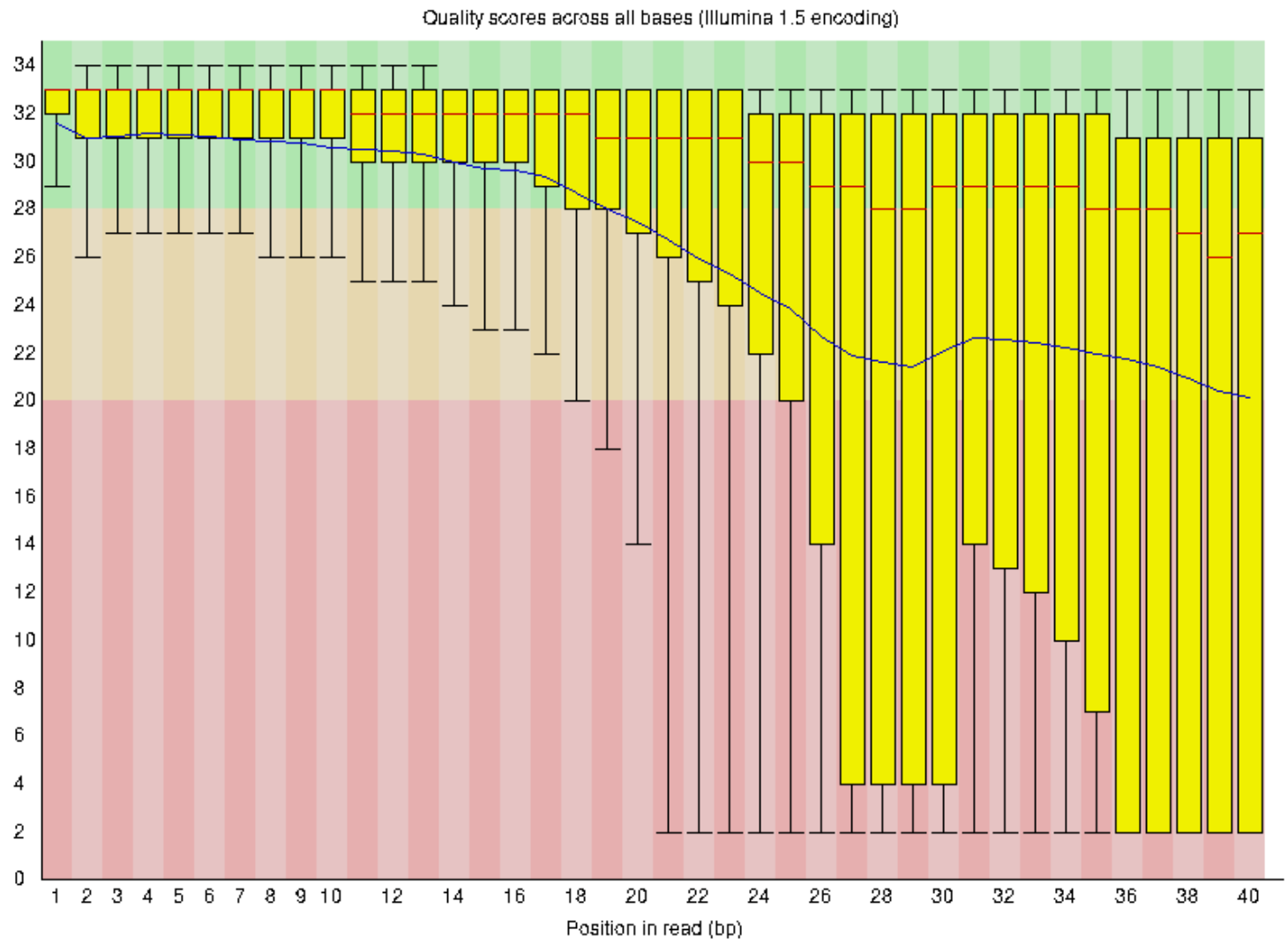
☐ How about determining absence of resistance genes?



Per sequence quality scores plot generated from FastQC

Base calls

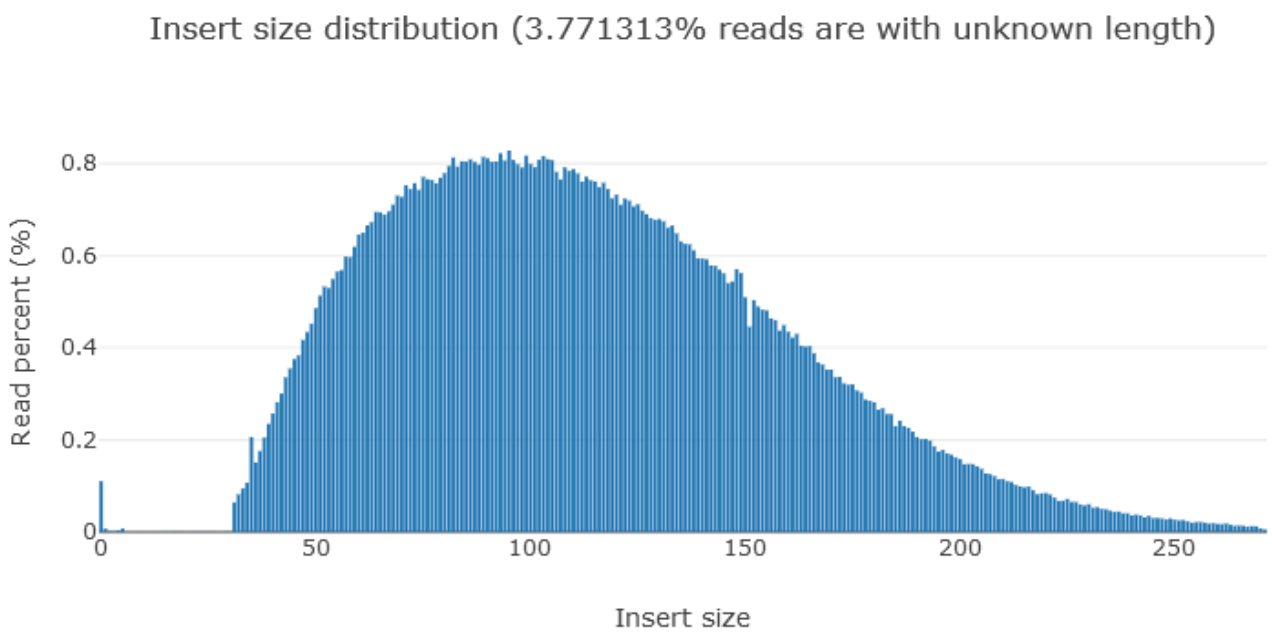
☐ What would be your arguments for using / discarding the following data for doing a accurate de-novo assembly?



Per base sequence quality plot generated from FastQC

Read size

- [ ] Expecting an average read size of 150 bp, what minimal **insert size** would you expect from your sample?
    - [ ] Does the following **insert size** plot live up to your expectations, and what does the plot indicate? (Note image below does NOT come from FastQC)



Insert size estimation plot generated from ***fastp***

Report aggregation

Having inspected some quality parameter plots for a single sample (above plots), it's time to look at the entire dataset as a whole. One major lacking feature in FastQC, is report aggregation, where a single report can be generated across both read mates and throuhgout an entire dataset. Luckily, there exist a tool for aggregating the results into a single report. This tool is named MultiQC, luckily for us, it is placed within the same envrionment as FastQC. Lets inspect the tool help page:

- If not there already, navigate to the `BTG/QC/` folder.
- Evoke the `multiqc` help page

    - If not allready loaded, activate the `QC` environment and then call the `multiqc` help page
    - Without loading any environment, evoke the help page through `micromamba`

- [ ] Look for the line stating "**usage"**. How can MultiQC be executed with fewest possible options??
- [ ] Execute MultiQC and inspect the aggregated report for your samples, consider the following:
    - [ ] What is the average read size throughout your samples (Sequence length distribution)
    - [ ] Are there any read positions with a quality lower than Q30 (Per base Sequence quality)?
    - [ ] Are there any issues with duplication (Sequence Duplication Levels)?

How about a bit of trimming?

So by now, you may have experienced that you can't necessarily skip read trimming, as there may be some reads that are low quality. This is very difficult to avoid even at the best of the labs.

There exists several tools for performing read trimming, for the sake of simplicity we will go with fastp, although `bbduk` is at least just as great a tool.

Lets inspect the help page of `fastp` and consult some of the default settings.

- Evoke the `fastp` help page (it's in the same environment)

- [ ] What option are used to set the average read quality score?
    - [ ] Considering you wish an average read score of Q30, would you need to apply quality filters?
- [ ] Does fastp automatically do read deduplication? (Filter out identical reads)
    - [ ] Considering your duplication rates, would you benefit from deduplication?
- [ ] What are the differences between the `cut_front` and `trim_front`?
    - [ ] What does `cut_right` and `cut_left` do?
    - [ ] If you wish to remove the the leading and trailing bases (e.g. due to tagmentation), would you use `cut_front` or `trim_front`?
- [ ] Do `fastp` perform adapter trimming by default?

Time for some trimming

Now its time to perform some trimming on our raw read data. In some cases DNA fragments has an insert size of less than 2xRead length, which means that there have been read through. Since bcl2fastq only trims adapters from one end of the read, read through into adapter sequences would not be identified. In these cases manual intervention MUST be taken. THUS, we enable adapter trimming for this exercise, and since we are working with paired end reads, make sure to enable the `--detect_adapter_for_pe` argument!

- Without loading any environment, evoke the help page through `micromamba`

- [ ] What arguments are required to determine input files and output files?

- Decide on trimming parameters from your previous considerations, then select one or two samples for trimming.
    - Run `fastp` for the selected samples with the selected trimming parameters. Make sure to point the output files to the `fastp/` folder, and a consider adding "_trimmed_" to the filenames.

    ▼ **Hint**

    ```
    micromamba run -n QC fastp -i /path/2/read/R1.fastq.gz -I /path/2/read/R2.fastq.gz -o /path/2/read/fastp
    ```

- [ ] How many reports does `fastp` create per sample?

- Rerun `multiqc` with the following command and inspect the aggregated report

    ```
    multiqc -f .
    ```

- [ ] Have the trimming solved the potential quality issues of your samples?