## Learning outcomes

**After having completed this chapter you will be able to:**

- Find information about a sequence run on the Sequence Read Archive
- Run `fastqc` on sequence reads and interpret the results
- Trim adapters and low quality bases using `fastp`

## Material

:fontawesome-solid-file-pdf: Download the presentation{: .md-button }

- `fastqc` command line documentation
- `cutadapt` manual
- Unix command line E-utilities documentation

## Exercises

### Download and evaluate an E. coli dataset

Check out the dataset at SRA.

**Exercise:** Browse around the SRA entry and answer these questions:

**A.** Is the dataset paired-end or single end?

**B.** Which instrument was used for sequencing?

**C.** What is the read length?

**D.** How many reads do we have?

??? done "Answers" A. paired-end

```
B. Illumina MiSeq

C. 2 x 251 bp

D. 400596
```

Now we will use some bioinformatics tools to do download reads and perform quality control. The tools are pre-installed in a conda environment called `ngs-tools`. Every time you open a new terminal, you will have to load the environment:

```
conda activate ngs-tools
```

Make a directory `reads` in `~/project` and download the reads from the SRA database using `prefetch` and `fastq-dump` from SRA-Tools into the `reads` directory. Use the code snippet below to create a scripts called `01_download_reads.sh`. Store it in `~/project/scripts/`, and run it.

```
#!/usr/bin/env bash

cd ~/project
mkdir reads
cd reads
prefetch SRR519926
fastq-dump --split-files SRR519926
```

**Exercise:** Check whether the download was successful by counting the number of reads in the fastq files and compare it to the SRA entry.

!!! tip "Tip" A read in a fastq file consists of four lines (more on that at file types). Use Google to figure out how to count the number of reads in a fastq file.

??? done "Answer" e.g. from this thread on Biostars:

~~~~
```sh
## forward read
echo $(cat SRR519926_1.fastq | wc -l)/4 | bc

## reverse read
echo $(cat SRR519926_2.fastq | wc -l)/4 | bc
```
~~~~

### Run fastqc

**Exercise:** Create a script to run `fastqc` and call it `02_run_fastqc.sh`. After that, run it.

!!! tip "Tip" `fastqc` accepts multiple files as input, so you can use a wildcard to run `fastqc` on all the files in one line of code. Use it like this: `*.fastq`.

??? done "Answer" Your script `~/project/scripts/02_run_fastqc.sh` should look like:

~~~
```sh title="02_run_fastqc.sh"
#!/usr/bin/env bash
cd ~/project/reads

fastqc *.fastq
```
~~~

**Exercise:** Download the html files to your local computer, and view the results. How is the quality? Where are the problems?

!!! info "Downloading files" You can download files by right-click the file and after that select **Download**:

```
<figure>
  <img src="../../assets/images/download_file.gif" width="500"/>
</figure>
```

??? done "Answer" There seems to be:

```
* Low quality towards the 3' end (per base sequence quality)
* Full sequence reads with low quality (per sequence quality scores)
* Adapters in the sequences (adapter content)

We can probably fix most of these issues by trimming.
```

### Trim the reads

We will use fastp for trimming adapters and low quality bases from our reads. The most used adapters for Illumina are TruSeq adapters, and `fastp` will use those by default. A reference for the adapter sequences can be found here.

**Exercise:** Check out the documentation of fastp, and the option defaults by running `fastp --help`.

- What is the default for the minimum base quality for a qualified base? ( option `--qualified_quality_phred`)
- What is the default for the maximum percentage of unqualified bases in a read? (option `--unqualified_percent_limit`)
- What is the default for the minimum required read length? (option `--length_required`)
- What happens if one read in the pair does not meet the required length after trimming? (it can be specified with the options `--unpaired1` and `--unpaired2`)

??? done "Answer"

```
- The minimum base quality is 15: `Default 15 means phred quality >=Q15 is
qualified. (int [=15])`
- The minimum required length is also 15: `reads shorter than
length_required will be discarded, default is 15. (int [=15])`
- If one of the reads does not meet the required length, the pair is
discarded if `--unpaired1` and/or `--unpaired2` are not specified: `for PE
input, if read1 passed QC but read2 not, it will be written to unpaired1.
Default is to discard it. (string [=])`.
```

**Exercise:** Complete the script below called `03_trim_reads.sh` (replace everything in between brackets `[]`) to run `fastp` to trim the data. The quality of our dataset is not great, so we will overwrite the defaults. Use a a minimum qualified base quality of 10, set the maximum percentage of unqalified bases to 80% and a minimum read length of 25. Note that a new directory called `~/project/results/trimmed/` is created to write the trimmed reads.

```
#!/usr/bin/env bash

TRIMMED_DIR=~/project/results/trimmed
READS_DIR=~/project/reads

mkdir -p $TRIMMED_DIR

cd $TRIMMED_DIR

fastp \
-i $READS_DIR/SRR519926_1.fastq \
-I $READS_DIR/SRR519926_2.fastq \
-o $TRIMMED_DIR/trimmed_SRR519926_1.fastq \
-O $TRIMMED_DIR/trimmed_SRR519926_2.fastq \
[QUALIFIED BASE THRESHOLD] \
[MINIMUM LENGTH THRESHOLD] \
[UNQUALIFIED PERCENTAGE LIMIT] \
--cut_front \
--cut_tail \
--detect_adapter_for_pe
```

!!! note "Additional options" Note that we have set the options `--cut_front` and `--cut_tail` that will ensure low quality bases are trimmed in a sliding window from both the 5' and 3' ends. Also `--detect_adapter_for_pe` is set, which ensures that adapters are detected automatically for both R1 and R2.

??? done "Answer" Your script (`~/project/scripts/03_trim_reads.sh`) should look like this:

~~~
```sh title="03_trim_reads.sh"
#!/usr/bin/env bash
~~~

~~~
TRIMMED_DIR=~/project/results/trimmed
READS_DIR=~/project/reads

mkdir -p $TRIMMED_DIR

cd $TRIMMED_DIR

fastp \
-i $READS_DIR/SRR519926_1.fastq \
-I $READS_DIR/SRR519926_2.fastq \
-o $TRIMMED_DIR/trimmed_SRR519926_1.fastq \
-O $TRIMMED_DIR/trimmed_SRR519926_2.fastq \
--qualified_quality_phred 10 \
--length_required 25 \
--unqualified_percent_limit 80 \
--cut_front \
--cut_tail \
--detect_adapter_for_pe
```
~~~

!!! note "The use of `\`" In the script above you see that we're using `\` at the end of many lines. We use it to tell bash to ignore the newlines. If we would not do it, the `fastp` command would become a very long line, and the script would become very difficult to read. It is in general good practice to put every option of a long command on a newline in your script and use `\` to ignore the newlines when executing.

**Exercise:** Check out the report in `fastp.html`.

- Has the quality improved?
- How many reads do we have left?
- *Bonus*: Although there were adapters in R2 according to `fastqc`, `fastp` has trouble finding adapters in R2. Also, after running `fastp` there doesn't seem to be much adapter left (you can double check by running `fastqc` on `trimmed_SRR519926_2.fastq`). How could that be?

??? done "Answers" - Yes, low quality 3' end, per sequence quality and adapter sequences have improved. Also the percentages >20 and >30 are higher. - 624724 reads, so 312362 pairs (78.0%) - The 3' end of R2 has very low quality on average, this means that trimming for low quality removes almost all bases from the original 3' end, including any adapter.