> Lines starting with `$` are commands that should be run in the terminal. You are meant to run everything that comes after `$`.

Setting up your environment

Virtual environments are great, because they let you have separate environments for separate proejcts. This is advantageous, since one project could rely on a certain package version 3, while some other may require version 4.

Conda

An advantage that **Conda** provides is not only for managing Python libraries, but also command line tools. This can make the tool's instalation process uniform and more generalized for users that don't work on the same systems.

I recommend installing Conda with these instructions: docs.conda.io/projects/conda/en/latest/user-guide/install/index.html.
Essentially the difference between Miniconda and Anaconda is that with Miniconda you have to install many tools manually. Install whichever you like.

Bioconda

To use certain bioinformatics tools, we need to use the **Bioconda** channel. No installation is needed, only this 3 commands that alter your `.condarc` configuration:

```
$ conda config --add channels bioconda
$ conda config --add channels conda-forge
$ conda config --set channel_priority strict
```

Virtual environment

Finally, we can create an environment for this project using:

```
$ conda create --name ans
```

Answer the prompt with yes to create an environment and then **activate** the environment with `$ conda activate ans`. Activation of the environment is local to a terminal session, so you will have to use this command again if you open another terminal instance.
To deactivate it simply run `$ conda deactivate`.

Finding our data

During this course we were tasked to work on the RNA-seq dataset linked to the article titled *"Wolbachia pipientis modulates germline stem cells and gene expression associated with ubiquitination and histone lysine trimethylation to rescue fertility defects in Drosophila"*.

BioProject accession number

First thing to do is to find this article on **NCBI** or **PubMed**. This can be done with a quick Google search: https://academic.oup.com/genetics/article/229/3/iyae220/7934994#510949052.

In the article, find the section *Data availability*. There will be an accession number for a **BioProject** (a BioProject is a collection of biological data related to a large-scale research effort).

SRA accession number

We can search for sequencing data related to this BioProject through the command line using **NCBI Entrez Direct tool**. First we install it in our environment with `$ conda install bioconda::entrez-direct`. Then we make a query and save the output in `csv` format.

```
$ esearch -db pubmed -query PRJNA1166928 | efetch -format runinfo > runinfo.csv
```

If we take a quick look at this file, it has many different columns, while we are only interested in the SRA (sequence read archive) accession numbers. We can extract the first column with `cut` on the output of `tail` that will skip the first line (the header):

```
$ tail -n +2 runinfo.csv | cut -d ',' -f1 > SraAccList.txt
```

- `-n +2` tells `tail` to start displaying from the second line
- The `-d` flag is used to specify a delimiter and the `-f1` tells `cut` to extract the first field.

Downloading data

Each of us students had to choose one accession number. I choose **SRR30833097**. Save this into a file with `$ echo "SRR30833097" > OurAcc.txt`.

Prefetch

Sequencing data is obtained from the SRA database with the SRA Toolkit. It can be installed with `$ conda install -c bioconda sra-tools`.

```
$ prefetch --option-file OurAcc.txt
```

Prefetch is used to obtain *Runs* (sequence files in compressed SRA format). The `--output-file` flag is used to use a file with a list of accession numbers as input. The command creates a directory named after the given accession number, where the downloaded files reside.

FastQ files

FastQ files are text files that combine **FASTA formatted sequences** with their **quality scores** and is the standard format for storing the output of high-throughput sequencing instruments.

The prefetched runs can be converted into FastQ format using `fasterq-dump`, that takes the created directory as input:

```
$ fasterq-dump --skip-technical SRR30833097/
```

`--skip-technical` returns only biological reads.

Since we have only single-end sequences, it should output a single `.fastq` file in the current directory. You can check that the line count matches 18149460 with:

```
$ wc -l SRR30833097.fastq
```

Generating a quality report

Since FastQ files bundle quality scores of sequences, we can take a quick look at the first stored sequence with `$ head -n 4 SRR30833097.fastq`. This will output

```
@SRR30833097.1 NB500947:1144:HM5GMBGXH:1:11101:4455:1091 length=86
GGCGGTCGAGTGCCTCACAGTGTATCAAGGGTNGGCCACGNTCCTNACTAATNGNGGCTNNTTGCGCCATCGTC
TCANGCAATGTT
+SRR30833097.1 NB500947:1144:HM5GMBGXH:1:11101:4455:1091 length=86
AAAAAEEEEEEEEEEEEEEEEEE<E//EEE#EEEEEE<#EEEE#E/AEEE#/#EEEE##EEE/EEAEEEEAE
6EE#/EEEAAEA
```

Each entry starts with `@` and is followed by a sequence identifier and some other information about the sequence. The line directly below it shows the raw sequence. The `+` line again can contain information about the sequence and below are the quality values for each nucleotide.

Command line FastQC

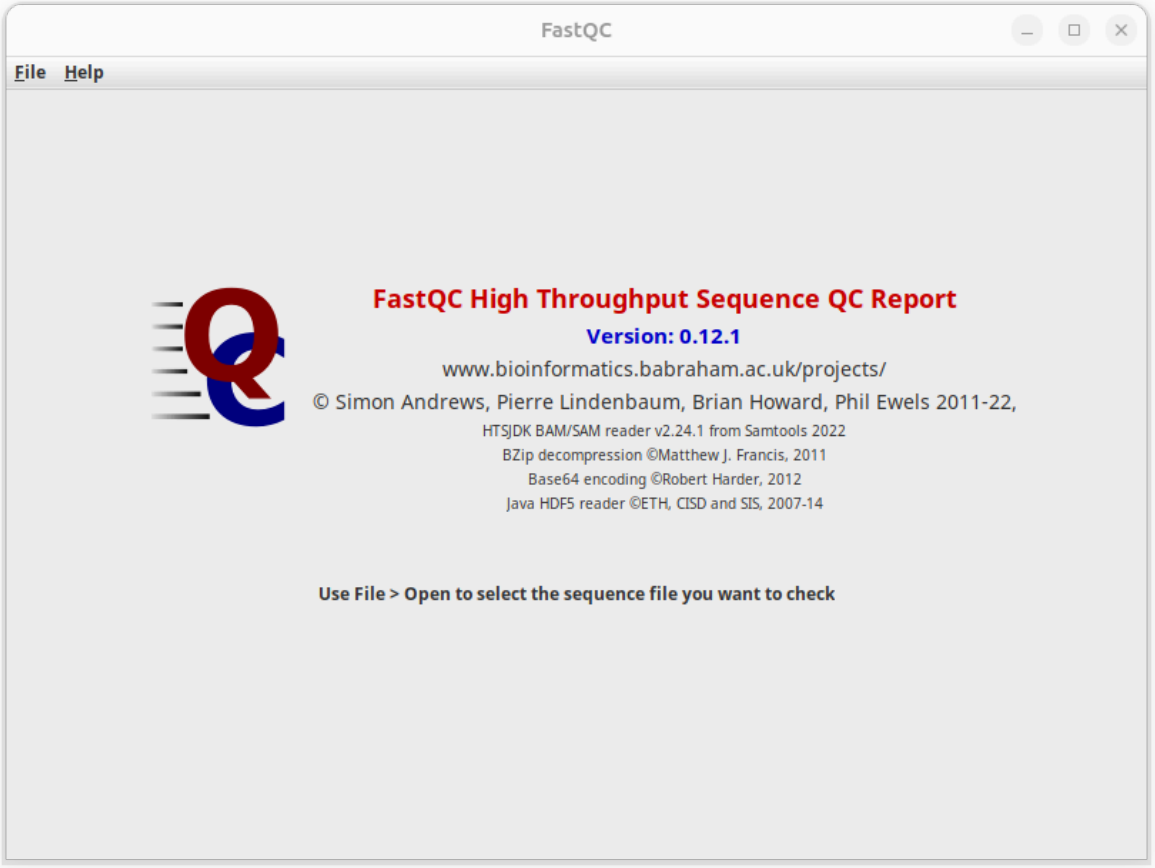
To get a **full report** on all sequences in our dataset, we can use **FastQC tool**. Install it with `$ conda install -c bioconda fastqc` and run it on the `.fastq` file.

```
$ fastqc SRR30833097.fastq
```

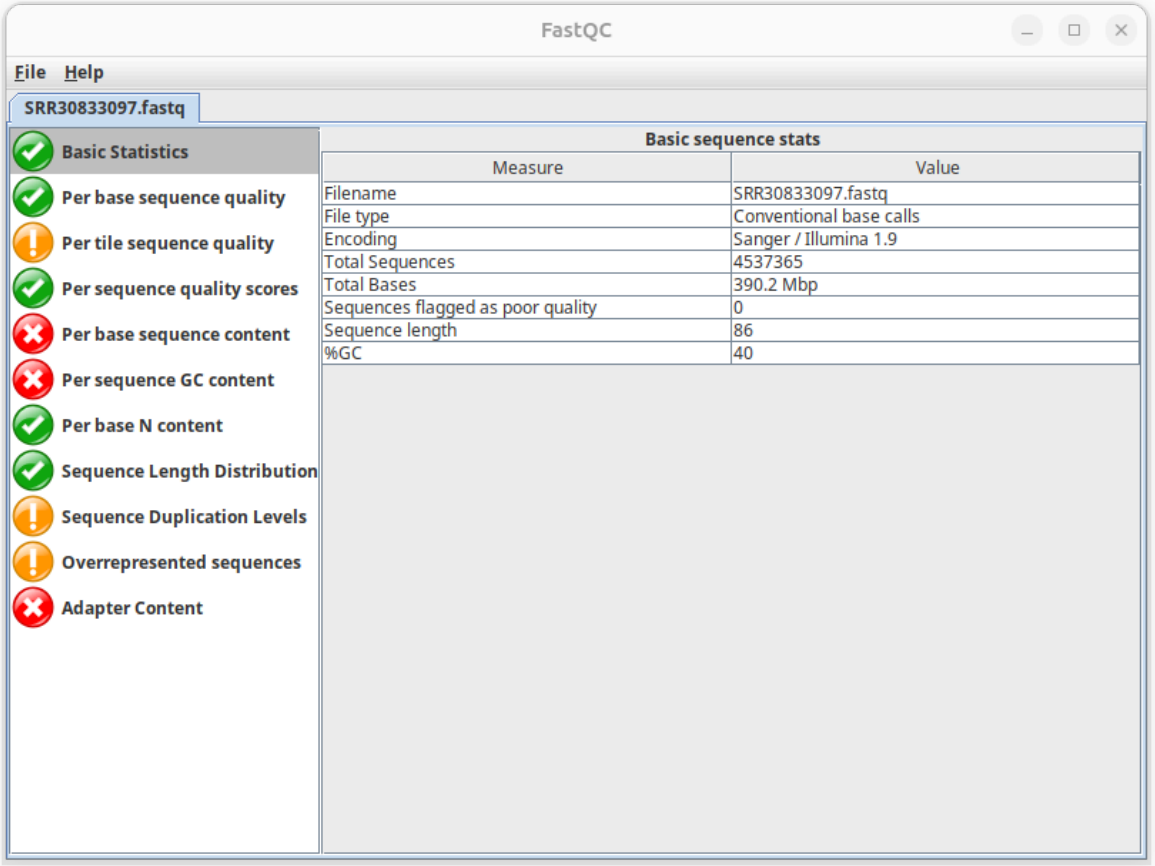
The quality report is the generated `.html` file. To view the rendered file, you can open it with your browser (e.g. `opera *.html`).

GUI FastQC

Alternatively, you can open a **graphical user interface** of FastQC tool by executing only `$ fastqc`. This will open a new window where you can open your `.fastq` file and generate and view the report.



Click on `File > Open` and select your file. When it's finished, you should see the report.



Checking quality of sequences