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:
https://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"*.

First thing to do is to find this article on NCBI or PubMed. This can be done with a quick Google search. In the article, find the section *Data availability*. There will be an accession number for a BioProject.

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 a lot of different features, while we are only interested in the SRA (sequence read archive) accession numbers. We can extract the first column with `cut` and skip the first line with the column name with `tail`:

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

- `-n 2` tells `tail` to display last 2 lines, while `+2` tells it 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`.

Sequencing data is obtained from the SRA database with the SRA Toolkit. It can be installed with `conda install -c bioconda sta-tools`.
A quick look into our csv shows us that the `Sample` column contains only `SINGLE` entries, meaning single-end sequencing was performed and not paired-end. This is important since we will use `fasterq-dump` from `sra-tools` tool next (difference from `fastq-dump` is that it uses `--split-3` flag by default.. and it's faster).

With this we can obtain our data with:

```
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.

The prefetched runs can then 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.

Checking quality of sequences

Install multiqc and fastqc tool with `sudo apt install multiqc` and `sudo apt install fastqc`.

Run `fastqc` on our dataset.

```
fastqc SRR30833097.fastq
```

Then we run `multiqc` on FastQC results folder. The `.` pulls all the reports in the current directory.

```
multiqc .
```