Worksheet 2: Reproducible research

To ensure reproducibility in genomics task (i.e. getting the same result every time), bioinformaticians use a variety of tools including:

- **Docker**: containerisation of a program to ensure it will always run even if your computer is updated or dependencies change
- **Workflow managers**: Lists the inputs, outputs and tasks required in a bioinformatic pipeline. Here, we will be using **WDL** to script our workflows.
- **Version control**: All changes to a script are documented (e.g. from python or R). **Github** or **gitlab** is commonly used for this

Task 4: Using samtools in docker

- Sign up and download!
- Go to dockerhub and find an image of "samtools"
- Download samtools. You will need to use terminal command:

```
docker pull <user/samtools>:<tag>
```

  Note that by default the latest version is downloaded, but it is good practice to specify the tag e.g. docker pull acct/image:v1.0
- To execute the docker image you be running the following options:

```
# runs <command> using the docker image
docker run <samtools>:<tag> <command>
# -v to mount a local folder in the folder /mnt
docker run -v ${path}:/mnt <samtools>:<tag> <command>
# -it in interactive mode
docker run ${path}:/mnt -it <samtools>:<tag> <command to run>
```

- Have a go exploring the files within the docker image.
  - Can you see your local files in the "mnt" folder?
  - Where is the "samtools" application located?

```
Hint: find . -name "samtools"
```

- Using samtools we can explore sequencing statistics from the bam file used in Task 3.
  - What genome is it aligned to and what is the aligner used?
  - What is the mapping rate?
  - What is the average depth?

```
Hint: samtools view -H ; samtools flagstat ; samtools depth
```

Task 5: Formalising the above in a WDL script

We are going to create a WDL workflow which does the following:

- Inputs:
  - Bam File
  - Bam Index
- Outputs:
  - File containing the header of the bam file
  - File indicating the mapping rate
  - File indicating the average depth

WDL is a workflow description language to link your inputs, outputs and tasks together. To get started:

- Make sure you have a (non-microsoft word) text editor installed! Some options include SublimeText, BBedit, Textwrangler, Vscode

- Install womtool (syntax checker) and cromwell (workflow executor). Make sure you also have java installed. Type `java` in terminal (command prompt in windows) to check.

Components of a WDL document:

- To get started view the example wdl file provided and look at the inputs, outputs and tasks
- The runtime attributes are used to specify the docker container used, memory, cpu and space requirements. Whilst the docker container is important for a local run, the other parameters are ignored here (but are VERY IMPORTANT in the cloud setting)
- For more help https://support.terra.bio/hc/en-us/articles/360037117492-Getting-started-with-WDL

Validating syntax:

- Once you're happy with the structure of your workflow, check the syntax

```
java -jar womtool-71.jar validate myworkflow.wdl
```

- Create a .json file to store your input variables

```
java -jar womtool-71.jar inputs myworkflow.wdl > inputs.json
```

- Fill in the inputs .json and then run the workflow!

```
java -jar cromwell-71.jar run myworkflow.wdl --inputs inputs.json
```

- And debug the errors which appear!

Task 6: Github

Github is a useful to store plain text files (GOOD: .txt, .csv, .R, .wdl, .json, .yaml) but NOT binary files (BAD: .doc, .xls, .ppt). It can be used to backup progress in scripts, and revert to previous versions if necessary

Setup a new repository:

- Create a github account. If you already have an existing bitbucket/gitlab, it's fine to use this too
- Use your university affiliation to sign up for unlimited free private repositories
- Create a new repository within github (e.g. summerProject2021). Add some text on what the repository is about in the `README` file.

Download this repository:

- Install git locally. Optional: install git desktop. If you're perfectly happy using command line for git, that's fine too.
- In terminal create a folder where you will store all your work.

```
mkdir <MyGitHubRepositories>
cd <MyGitHubRepositories>
```

- The link to your github repository is located on the website under "code" – copy this to clone your repository

```
git clone https://github.com/<username>/<repository>
```

- Open the README within this directory and check if it matches what you just typed on the website!

Upload your .wdl file

- Copy the .wdl and .json file created in Task 5 to your local repository. Also copy a random word or pdf file
- Check the status of your files using

```
git status
```

  Note that some hidden files are present and staged to be uploaded if we don't exclude them
- Create a `.gitignore` file: This is also useful to store files/file extensions which are not to be uploaded to github e.g. .DS_Store files, word documents and pdfs

```
.DS_Store
.pdf
.word
```

- Copy a random pdf or word file to the directory and check again with `git status`
- To upload this file, you will need the following 3 commands:

```
# Select the files to add
git add myworkflow.wdl ## adds specifically myworkflow.wdl
git add -A ## adds everything which is not listed in the .gitignore file
# Create a message for the commit
git commit -m "my first upload"
# push the file to your online repo
git push
```

- At this point terminal will prompt you to enter your github credentials. If you run into an error by using your password, you may need to create an authentication using a token
- Now visit your github repository and check whether the file has uploaded!

Task 7: Getting started with Terra and Google Cloud Computing

Terra is a cloud-based data and workflow management system. It uses WDL and Cromwell and relies on docker images in the execution.

- Once garvan gives you GCP access, begin reading through this user guide to set up a Terra account. Let me know once this has been established and I can share the workspace with you
- Upload your WDL script to firecloud (https://portal.firecloud.org/) and test it on example bam files present in the workspace.
    - How many reads does each "subsample" of the NA12878 bam file have?

https://docs.google.com/document/d/1pZVTxjRJfAyWYiFWmmF_o31lORQ8a-xmqdK1zE3RDyk/edit#