

BIOL550 - Assignment #1 (5 questions, 45 points)

Always read through assignment/project/exam instructions carefully; they are provided for a reason, and a lot of time can be saved by taking an extra second reading through them!

All commands necessary to answer the following questions should be stored in a text file named `assignment1_[lastname]_[firstname].txt`, which in turn will be submitted on Blackboard as part of the assignment. The format of your assignment file should match the following format:

```
# Question 1
My command line 1
My command line 2

# Question 2
My command line 1...
```

NOTE: Blackboard will not accept the submission of files with the .sh extension likely in attempts to prevent nefarious activities; to submit files with .sh extensions, you can change the extension to .txt or compress your files into a single tar archive.

Please **refrain from using a word processor like Microsoft Word** to write commands and code for assignments/projects/exams, as word processors have tendencies to corrupt commands and code via formatting. Instead, utilize a code editor such as Visual Studio Code (<https://code.visualstudio.com/>) or Notepad++ (<https://notepad-plus-plus.org/>).

This assignment is composed of five (5) multi-part questions. Some questions may be dependent on previous questions. If you are unable to complete a particular question, and its result is needed in another question, provide the command you would have used.

Question 1 (12 points) Making command lines do repetitive tasks for you.

At the beginning of a project, it is wise to set up an environment that will help you stay organized. The goal of this question is to create an organized work environment for this and future assignments, while also practicing some common file manipulation commands. Here we will start off by performing repetitive tasks manually, however, our time is better spent doing more exciting things, so we will transition to smart and more efficient command lines that help reduce workload.

- a) (1pts) In your home directory (~), create a directory named **Assignments** using the **absolute path**.
- b) (1pts) Within ~/Assignments, create directories **Assignment_1** to **Assignment_5** using **relative paths**, one command per directory.
- c) (3pts) Using a single bash 'for' loop, create directories **Q01** to **Q25** in ~/Assignments/Assignment_1. ## In your 'for' loop, utilize a starting range of 01 to add leading zeros!
- d) (1pts) Remove directories **Q20** to **Q22**, one command line per directory, using **absolute paths**.
- e) (1pts) Remove directories **Q23** to **Q25**, one command line per directory, using **relative paths**.
- f) (3pts) Using a single bash 'for' loop make empty files **Q01_data.file** to **Q19_data.file** in the respective directories (**Q01_data.file** in ~/Assignments/Assignment_1/Q01, **Q02_data.file** in ~/Assignments/Assignment_2/Q02,).
- g) (2pts) Using a single command line, remove directories **Q10** to **Q19** and their contents.

Question 2 (5 points) Utilizing command manuals and help messages.

Sometimes we will need to download content from various online sources, whether a tool or a database. In class we discussed how **wget** is one of several tools that can be used to grab content from online, however, we did not look at all the functionality the tool has to offer. The goal of the question is to become comfortable with inspecting either the **manpages** or **--help message** for Linux commands.

- a) (1pts) Copy and paste the tar archive link (<https://github.com/PombertLab/3DFI/archive/refs/tags/1.0.1a.tar.gz>) for our pipeline, 3DFI, into a file called **3DFI.url** in `~/Assignments/Assignment_1/Q02`.
- b) (4pts) With **wget** download the tar archive specifying **3DFI.url** as an input file and save it as `~/Assignments/Assignment_1/Q02/3DFI-1.0.1a.tar.gz`.

Question 3 (10 points) Working with tar archives.

Compressed-tar archives are not only great for saving space, but for increasing portability as well. Often, when downloading tools or databases, the standard format they are obtained in is a compressed-tar archive, and thus it is important to know how to create and decompress them. The goal of this question is to become familiar with decompressing and creating tar archives.

- a) (2pts) Change directories into `~/Assignments/Assignment_1/Q02` and decompress the tar archive `3DFI-1.0.1a.tar.gz`.
- b) (1pts) Create a copy of the decompressed archive `3DFI-1.0.1a.tar.gz` in `~/Assignments/Assignment_1/Q03`.
- c) (1pts) Change directories into `~/Assignments/Assignment_1/Q03` and move the `Visualization` directory from `3DFI-1.0.1a` into the current directory (`~/Assignments/Assignment_1/Q03`).
- d) (3pts) Create a `BZIP2`-compressed tar archive of the directory `3DFI-1.0.1a` and name it `3DFI_no_vis.tar.bz2`.
- e) (3pts) Create a `GZIP`-compressed tar archive of the directory `Visualization` and name it `3DFI_vis_scripts.tar.gz`.

Question 4 (12 points) Tracking your steps for reproducibility.

Keeping track of the commands you use is essential housekeeping for a bioinformatician. There may come a time where an analysis you performed is going to be included in a publication and written methods are needed, or perhaps you are asked to run the exact same analysis, just on a different dataset. You could try to dig through your history file and piece the process back together, or you could be proactive and keep a list of commands you used for the analysis. Having these commands handy in a bash script allows you to replace the original dataset with the new one and run the bash script instead of each command individually. What a time saver! It is also wise to include comments reminding you why exactly you ran this command, or why a certain database was used. The goal of this question is to familiarize yourself with setting up a bash script and good commenting habits, as well as a few other common Linux commands.

Create a bash script named `my_process.sh` in `~/Assignments/Assignment_1/Q04` that includes the following:

- (1pts) The **proper shebang** (`#!/`) so the bash shell knows what interpreter to use.
- (1pts) Thoughtful comments providing short detail on what each command in the script is doing. Comments in bash start with `#` to tell the interpreter that these are not commands. Example:

```
## This is a comment in Bash!
```

The script, when ran, should do the following:

- (1pts) Creates a directory `SequencingData` in `~/Assignments/Assignment_1`
- (1pts) Copies all files ending with `.fastq.gz` from `/media/Data_2/BIOL550/tmp/` to `~/Assignments/Assignment_1/SequencingData`.
- (2pts) Decompresses all `.fastq.gz` files in `~/Assignments/Assignment_1/SequencingData`
- (2pts) Concatenates the contents of all `.fastq` files into a single file name `PacBio_reads.fastq` in `~/Assignments/Assignment_1/SequencingData`.
- (2pts) Changes the permission of `PacBio_reads.fastq` to 644 (`-rw-r--r--`)
- (2pts) Creates a symbolic link for `Pacbio_reads.fastq` in `~/Assignments/Assignment_1/Q04`

Be sure that your bash script runs all the way through! You must make your bash script executable to run it without needing to specify the bash interpreter. **Submit your script to Blackboard** (remember, files with `.sh` extensions cannot be submitted to Blackboard, so change the file extension or create a tar archive of your answer file and this script).

Question 5 (6 points) Customizing your Bash environment.

Customizing your environment is a common task when working as a bioinformatician. Installing a new tool? Add it to \$PATH. Create a new BLAST database? Add it to \$blastdb. Maybe you're a Linux administrator who has to modify the global configuration file frequently but hates that it takes two commands to edit and source the file. Good news, you can create an alias that opens the file and sources it after you close it to save you the trouble! The goal of this question is to familiarize you with creating aliases and environmental variables and adding those variables to your local configuration files to make the changes persistent for future logins.

- a) (1pts) Create an alias **hello**, when invoked, prints "**Greetings, [your first name here]!**" to the shell using **echo**.
- b) (2pts) Add this alias to your `~/.bashrc` and source it. Check that it works. Copy the contents of your `~/.bashrc` and paste it at the end of `assignment1_[lastname]_[firstname].txt`.
- c) (1pts) Create an environment variable **TDFI** that points to the **3DFI-1.0.1a** directory in `~/Assignments/Assignment_1/Q02`.
- d) (2pts) Add 3DFI to your `~/.bash_profile` and source it. Check that it works. Copy the contents of your `~/.bash_profile` and paste it at the end of `assignment1_[lastname]_[firstname].txt`.