Homework 2

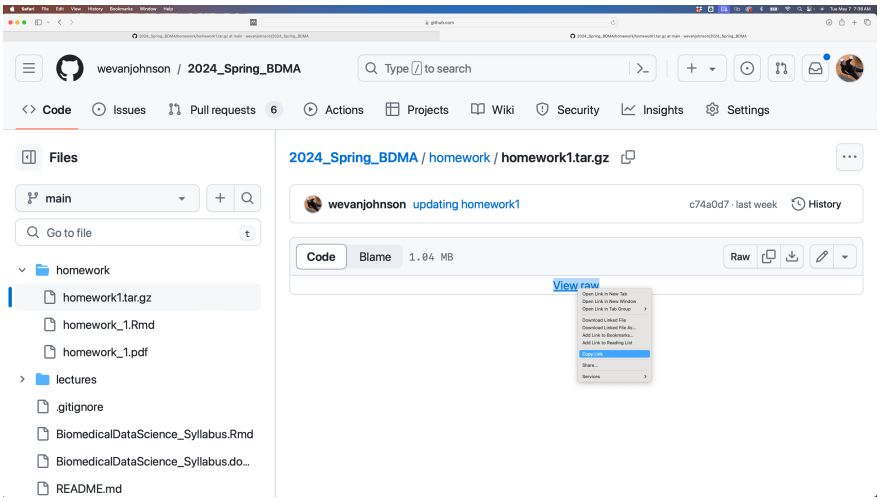
GSND 5345Q, Fundamentals of Data Science

Due Monday, January 27th, 2025

Advanced Unix Tools

Most Unix implementations include a large number of powerful tools and utilities. (Unix has been in development for more than 50 years!). We were only able to scratch the surface in our class time. It will take time to become comfortable with Unix, but as you struggle, you will find yourself learning just by looking at `man` files and finding solutions on the internet. For this Homework, you will explore several more advanced Unix functions. You can use any resource available to you–classmates, the internet (Google and ChatPGT!), and Dr. Johnson. Ask all the questions you want, just make sure you do the work and you learn!

1. Learn more about tools for downloading files from external servers (e.g., `scp`, `ftp`, `sftp`, `rsync`), and for to downloading data from webpages (e.g., `curl`, `wget`, `mget`). Use an appropriate function to download the `homework2.tar.gz` from the homework folder on course GitHub page. Give the code you used to download these data.(**Hint:** To download the `homework2.tar.gz` from GitHub, control/right click on the "View raw" link and copy the location (see image). If you use the URL in the address bar it downloads the .html for the website. Note this picture might be from a different homework and class, but the action is still the same.)



2. Learn about the `tar` function. What is a tarball? How is it different from a .zip file? Download the `homework2.tar.gz` file from GitHub and unzip the contents, and report that code you used. How effective is the compression for this tarball? After you complete this homework, add your homework files directory and generate a gzipped tarball for all the Homework 1 data plus your answers. Make sure to provide the code you used to generate the tarball for your homework.

3. Research the `chmod` function. Give short explanation of what this function does, its syntax, and examples when you would use it. Practice `chmod` by changing the permissions on the 'TB_microbiome_data.txt' file in the Homework 1 directory from the previous questions. Give examples of the code you used and show that the code works (e.g., use `ls -l`).

4. The `grep` function is an extremely powerful tool for search (potentially large) files for patterns and strings. One advantage is that you don't have to open the file to conduct a search! Using the internet, find a short tutorial on the basics of `grep`, and give the code and results for the following tasks:

   (a) How many FC receptor genes are present in the 'TB_nanostring.txt' file? (hint: search for 'FC' in the file)

   (b) How many samples (rows) in the 'nanostring_annotation.txt' do not have a co-morbid condition or other risk factor?(i.e., inverse search – how many rows *do not* have a "Yes")

   (c) How many coronavirus genomes are present in the 'viral.fasta' file? How many of these are SARS-COV-2?

   (d) How many times does the letter 'A' (capital or lowercase) appear in all the files from the homework2 tar file? (i.e., ignore case).

   (e) What *Staphylococcus* species are present in the 'TB_microbiome_data.txt' file? (hint: each separate microbe has its own row in the file). Print out the counts for *Mycobacterium tuberculosis*. How many *Streptococcus* species are present?

5. Learn how to use `less` to display large text files in the terminal using the `man` help page. Using the "OPTIONS" section of the `man` page, open the 'viral.fasta' file to display so that it does not wrap long lines (default), displays line numbers, and opens at the first occurrence of 'coronavirus'. Provide the command you used to open the file in this way. Within `less`, learn and practice how to scroll forward/backward, scroll forward/backward n lines, jump to the middle or end of the file, and search for text in the document. When would it be advantageous to use `less` over a tool like Microsoft Word? Ask Dr. Johnson why in Unix `more` is less and `less` is more :-).

6. Open a text file in `vim` and change the file. How do you move to the beginning/end of a line, insert text, copy and paste, delete text and lines? How do you save your file or exit `vim` with/without saving your result? What are the advantages and disadvantages of `vim` versus `less`? In which scenarios would you use each of these?

7. Learn about `pipes` and `redirects` in Unix. In which scenarios would you use them, and why are they helpful? describe what the following commands do:

   (a) `ls -l | less`

   (b) `ls -l > directory_contents.txt`

   (c) `ls -l » directory_contents.txt`

   (d) `cat directory_contents.txt | head -3 | tail -2`

   (e) `ls | grep -c html`

   (f) `ls | wc -l`

   (g) `cat file1.txt file2.txt > file3.txt`

   You can also us pipes in R! Investigate how to do this and give the code for a great example.

8. Learn about another Unix command that we have not discussed. Give a short description of this function, when you would use it, its syntax, and give some examples of its use.