Week 3 Exercise BIOL 7200

1.

To acquire a dataset for a later question, I downloaded all the histidine kinase domains of proteins found in Escherichia coli strain K-12 from the Uniprot database. This dataset is dicsussed in the relevant question below. The sequences came with long headers including lots of extranious information. Fortunately, we can quickly tidy them up using regex so you can use them later.

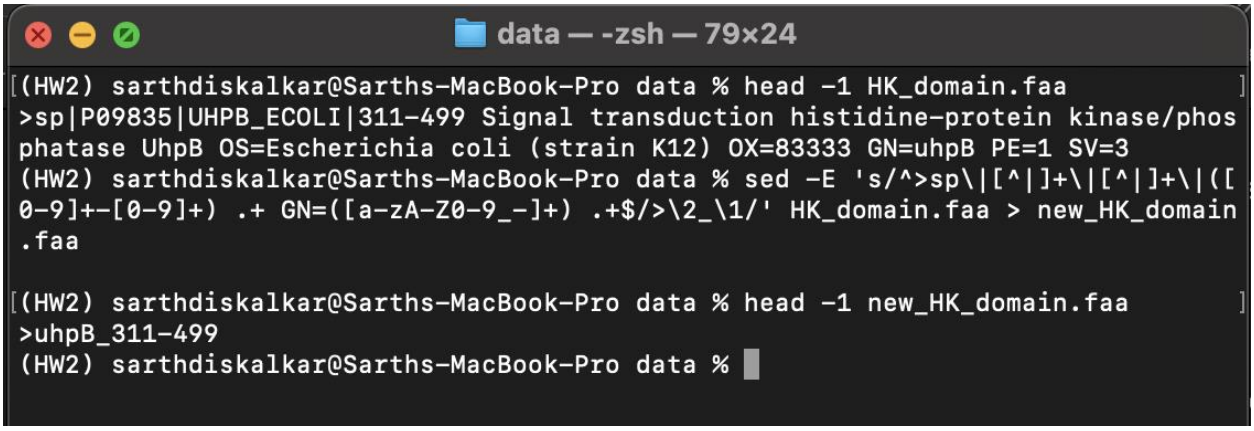
Using a regex with `sed`, create a copy of HK_domains.faa in which you replace fasta headers of HK_domain.faa file with just the gene name after GN= in the existing headers and the position ranges.

E.g., the first header is currently

>sp|P09835|UHPB_ECOLI|311-499 Signal transduction histidine-protein kinase/phosphatase UhpB OS=Escherichia coli (strain K12) OX=83333 GN=uhpB PE=1 SV=3

the new header should be
>uhpB_311-499
Provide a screenshot showing the following commands executed in the same terminal window:

- `head -1 <original file>`
- Your command that converts the headers and outputs to a new file
- `head -1 <new file>`



2.

While the new headers are better than what we had before, the above headers are not good enough because HK_domains.faa contains protein sequence. Bacterial protein names start with an uppercase letter, while gene names start with a lowercase letter. Create a new file the fasta

headers with the protein names instead. Protein names are present once on each line and bacterial protein names are of the format AbcD (1 uppercase letter, two lowercase letters, one upppercase letter).

Create a new copy of HK_domains.faa in which you replace the headers with just the protein name and amino acid position range.

e.g., the first header is currently

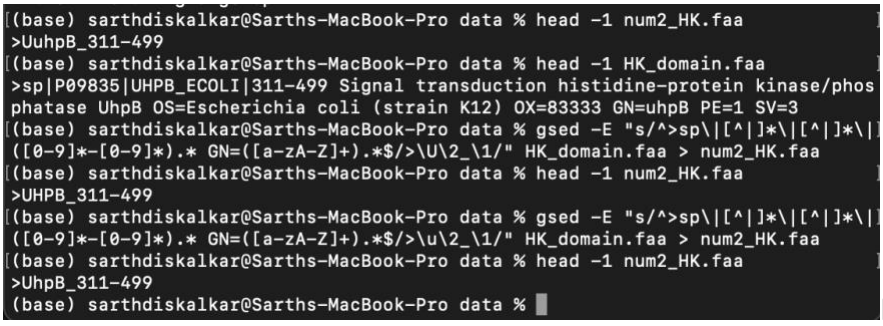
>sp|P09835|UHPB_ECOLI|311-499 Signal transduction histidine-protein kinase/phosphatase UhpB OS=Escherichia coli (strain K12) OX=83333 GN=uhpB PE=1 SV=3

the new header should be
>UhpB_311-499
Provide a screenshot showing the following executed in the same terminal window:

```
head -1 <original file>
```

Your command that converts the headers and outputs to a new file

```
head -1 <new file>
```



located at end

3.

Make another copy of HK_domains.faa where you remake the headers as in question 1 (i.e., get the gene name from next to "GN="), but modify the regex replace command so that it converts the first character of the gene name to uppercase. i.e., the output of the command should be identical to that of question 2, but you should do it with a regex that matches the same thing that you matched in question 1.

Provide a screenshot showing the following executed in the same terminal window:

`head -1 <original file>`

Your command that converts the headers and outputs to a new file

`head -1 <new file>`

```
(base) sarthdiskalkar@Sarths-MacBook-Pro data % head -1 HK_domain.faa
>sp|P09835|UHPB_ECOLI|311-499 Signal transduction histidine-protein kinase/phos
phatase UhpB OS=Escherichia coli (strain K12) OX=83333 GN=uhpB PE=1 SV=3
(base) sarthdiskalkar@Sarths-MacBook-Pro data % gsed -E 's/^>sp\|[^|]+\|[^|]+\|
([0-9]+-[0-9]+).* GN=([a-zA-Z]+).*$/>\u\2_\1/' HK_domain.faa > modified_HK_doma
in_with_uppercase.faa

(base) sarthdiskalkar@Sarths-MacBook-Pro data % head -1 modified_HK_domain_with
_uppercase.faa
>UhpB_311-499
(base) sarthdiskalkar@Sarths-MacBook-Pro data %
```

4.

Task :Newick leaf names can contain any characters except "(", ")", ",", ":", and ";". Write a regex that would match **ANY** allowed leaf name in a newick string. Use your regex and `grep` to extract all leaf names from the Tree_of_life.nwk in the data directory.

Note that leaf names can be numbers so consider what about newick format is used to distinguish leaf names from other elements. Your regex should be able to identify the leaf names "0.5" and "0.000123" in the newick string "(0.5:0.4,0.000123:0.123);"

Provide a screenshot showing the following executed in the same terminal window:

Your command to identify all leaves piped to `wc -l` to count the results (i.e., `<your command> | wc -l`)
Your command to identify all leaves piped to `head` to show a snippet of the results (i.e., `<your command> | head`)

```
data — -zsh — 79×24
(base) sarthdiskalkar@Sarths-MacBook-Pro data % ggrep -Po "(?<=[,(])[^(),:;]+(?
=:)" Tree_of_life.nwk | wc -l
     191
(base) sarthdiskalkar@Sarths-MacBook-Pro data % ggrep -Po "(?<=[,(])[^(),:;]+(?
=:)" Tree_of_life.nwk | head
Nanoarchaeum_equitans
Pyrobaculum_aerophilum
Aeropyrum_pernix
Sulfolobus_solfataricus
Sulfolobus_tokodaii
Thermoplasma_volcanium
Thermoplasma_acidophilum
Archaeoglobus_fulgidus
Halobacterium_sp._NRC-1
Methanosarcina_acetivorans
(base) sarthdiskalkar@Sarths-MacBook-Pro data %
```

5. Use a regex and `grep` to extract all branch lengths from the Tree_of_life.nwk file and then calculate the sum of all branch lengths using `awk`. Note again that leaf names can be numbers. Your regex should match only branch lengths and not node support or leaf names. Provide a screenshot showing the command and its result in a terminal window

```
(base) sarthdiskalkar@Sarths-MacBook-Pro data % ggrep -oP ':\d+\.\d+' Tree_of_l
ife.nwk | gawk '{sum += substr($0, 2)} END {print sum}'
45.54
(base) sarthdiskalkar@Sarths-MacBook-Pro data % 89
```

6. Use `find` with a regular expression and `wc -l` to determine how many gene files are in the directory "data/find_data/". Note use "egrep" regextype as that refers to the GNU ERE covered in class.

```
(HW2) sarthdiskalkar@lawn-143-215-107-230 data % gfind find_data/ -type f -regextype egrep -regex ".*/[a-z].*[.]
[fa][sta]+" | wc -l

    4334
(HW2) sarthdiskalkar@lawn-143-215-107-230 data %
```

7. Use `find` with a regular expression and `wc -l` to determine how many protein files are in the directory "data/find_data/".

   Provide a screenshot of a terminal window showing your command and the output

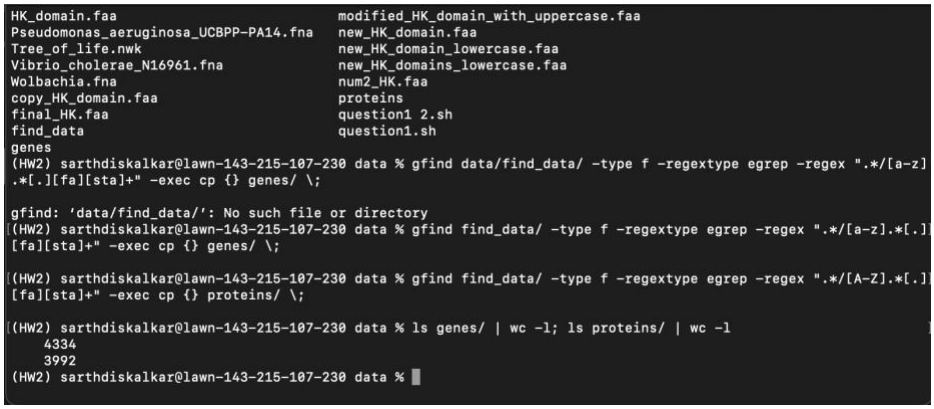
   Total files are 8326

```
(HW2) sarthdiskalkar@lawn-143-215-107-230 data % gfind find_data/ -type f -regextype egrep -regex ".*/[a-z].*[.]
[fa][sta]+" | wc -l

    4334
(HW2) sarthdiskalkar@lawn-143-215-107-230 data % gfind find_data -type f -regextype egrep -regex ".*/[A-Z].*[.][
fa][sta]+" | wc -l

    3992
(HW2) sarthdiskalkar@lawn-143-215-107-230 data %
```

8. Use `find` with a regular expression and `-exec` to copy all genes to a new directory called "genes/" and another `find` command to copy all proteins to a new directory called "proteins/"

   Provide a screenshot showing the following executed in the same terminal window:

   Your command to copy all gene files to "genes/"
   Your command to copy all protein files to "proteins/"
   The output of the command `ls genes/ | wc -l; ls proteins/ | wc -l`



9. Setting up and forking a git repo (30 points)

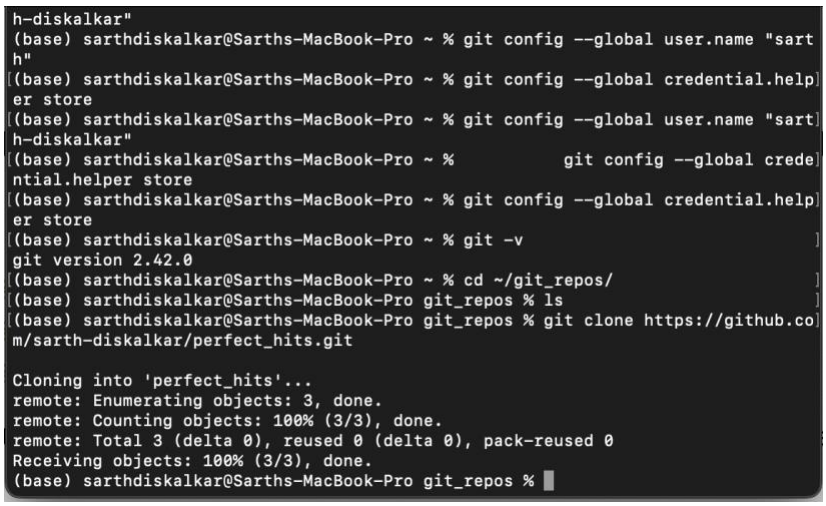
Tasks for number 9

1. (6 points)

   Make a git repo on your github account called "perfect_hits". OPTIONAL: You may wish to create a README.md when prompted in which you can document the usage of the script in this repo, but your README will not be assessed.

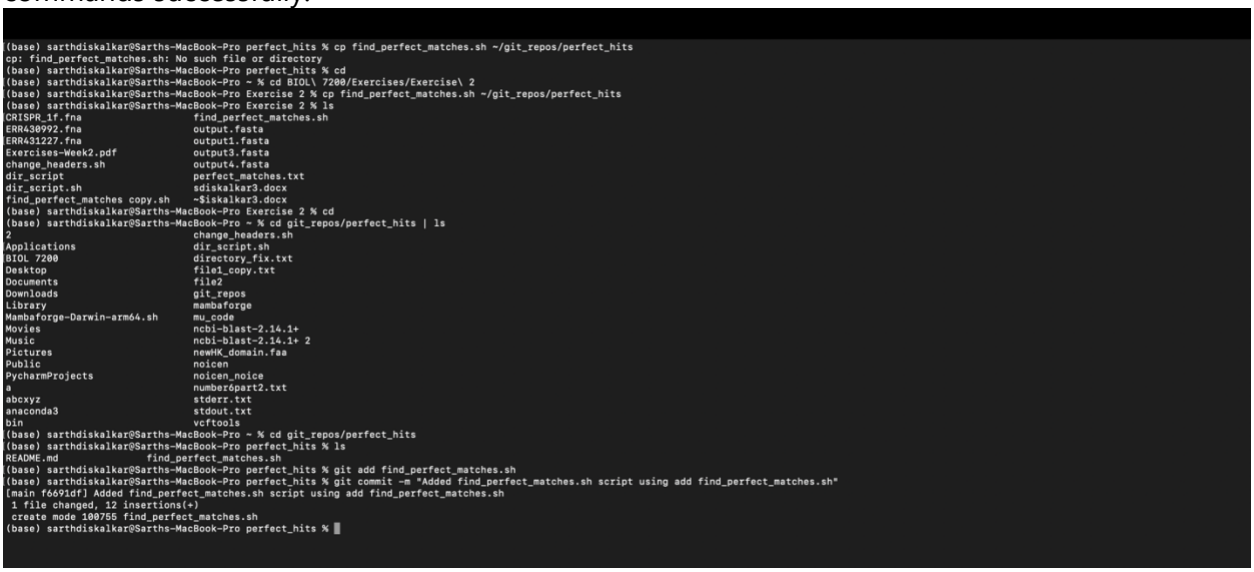
   Clone the repo to a location on your computer (perhaps ~/git_repos/perfect_hits/).

   Provide a screenshot of your terminal in which you execute the clone command.



2. **Copy** your `find_perfect_matches.sh` script from last week (Question 4.3) into the new repo. Add and commit your script so it is tracked by git.

Provide a screenshot of your terminal in which you executed the copy, add, and commit commands successfully.

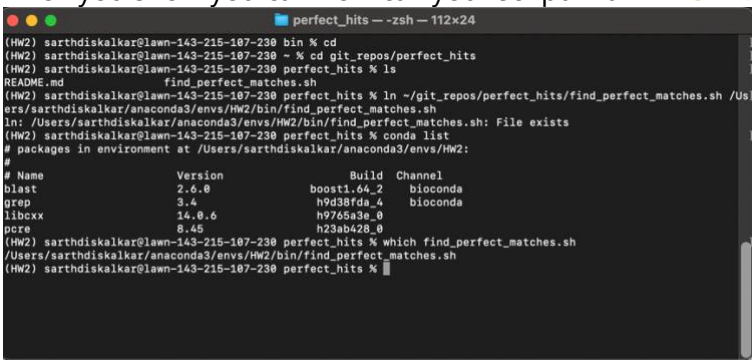


3.

Make and activate a conda/mamba environment with blast installed. Activate an existing one if you already have one.

Make a hardlink of your `find_perfect_matches.sh` script in the mamba env bin folder for the activated environment.

Provide a screenshot in which you show you can now call your script with `which <script>` while in the mamba env



Same image right below

```
(base) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % ls
README.md               find_perfect_matches.sh
(base) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % conda activate HW2
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % conda list
# packages in environment at /Users/sarthdiskalkar/anaconda3/envs/HW2:
#
# Name                    Version                   Build  Channel
blast                     2.6.0              boost1.64_2    bioconda
grep                      3.4                 h9d38fda_4    bioconda
libcxx                    14.0.6              h9765a3e_0
pcre                      8.45                h23ab428_0
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % ln find_perfect_matches.sh $(conda info --base)/envs/blast_env/bin/find_perfect_matches
ln: /Users/sarthdiskalkar/anaconda3/envs/blast_env/bin/find_perfect_matches: No such file or directory
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % conda env list
# conda environments:
#
base                  /Users/sarthdiskalkar/anaconda3
HW2                 *  /Users/sarthdiskalkar/anaconda3/envs/HW2
                      /Users/sarthdiskalkar/mambaforge/envs/newenv

(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % ln ~/git_repos/perfect_hits/ find_perfect_matches.sh /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches

ln: /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches: No such file or directory
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % ln ~/git_repos/perfect_hits/ find_perfect_matches.sh /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh

ln: /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh: No such file or directory
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % ln ~/git_repos/perfect_hits/find_perfect_matches.sh /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh

(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % ln ~/git_repos/perfect_hits/find_perfect_matches.sh /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh

ln: /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh: File exists
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits % which find_perfect_matches.sh
/Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh
(HW2) sarthdiskalkar@Sarths-MacBook-Pro perfect_hits %
```

4. Replace the contents of the `find_perfect_matches.sh` script in your repo with `echo "oops!"`. Save the file.

   Run the `find_perfect_matches.sh` that is on your path (not the file that is in your git repo). Provide a screenshot of the output of running `find_perfect_matches.sh` from your PATH. Is the hardlinked file that you put in the mamba env bin directory also modified? Explain?

```
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % echo $PATH
/Users/sarthdiskalkar/anaconda3/envs/HW2/bin:/Users/sarthdiskalkar/anaconda3/condabin:/opt/homebrew/bin:/opt/hom
ebrew/sbin:/Library/Frameworks/Python.framework/Versions/3.11/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
bin — -zsh — 112×24
ers/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh
ln: /Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh: File exists
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % conda list
# packages in environment at /Users/sarthdiskalkar/anaconda3/envs/HW2:
#
# Name                    Version                   Build  Channel
blast                     2.6.0              boost1.64_2    bioconda
grep                      3.4                 h9d38fda_4    bioconda
libcxx                    14.0.6              h9765a3e_0
pcre                      8.45                h23ab428_0
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % which find_perfect_matches.sh
/Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % nano find_perfect_matches.sh
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % ./find_perfect_matches.sh
oops!
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % which find_perfect_matches.sh
/Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh
(HW2) sarthdiskalkar@lawn-143-215-107-230 perfect_hits % cd
(HW2) sarthdiskalkar@lawn-143-215-107-230 ~ % cd /Users/sarthdiskalkar/anaconda3/envs/HW2/bin
(HW2) sarthdiskalkar@lawn-143-215-107-230 bin % ./find_perfect_matches.sh
oops!
(HW2) sarthdiskalkar@lawn-143-215-107-230 bin % which find_perfect_matches.sh
/Users/sarthdiskalkar/anaconda3/envs/HW2/bin/find_perfect_matches.sh
(HW2) sarthdiskalkar@lawn-143-215-107-230 bin %
```

Yes, the hardlinked file in the conda environment's envs/HW2/bin directory will be modified when you modify the original script in your Git repository. This is because a hardlink was created which means that both the original file and the hardlink file were pointing to the same inode number/data blocks.

5. Revert the changes to the script in your git repo by running `git checkout .`.

The above command changed every file in the repo to the version at the HEAD of your git history. What does HEAD refer to in this context? HEAD refers to the latest, most recent commit in your repo. HEAD is updated whenever a new commit is created.

If you wanted to revert to a different point in your git history (e.g., a specific commit) how would you do that? You would have to type in "git log" then look at the specific commit hash/number. To revert to a different point, use the command "git checkout" and then type in or copy the <commit number> right afterwards with a space in between. You will then revert back to the state of your repo when that commit was made.

10. find_homologs.sh (40 points)

Tasks

1. Create a copy on github of your perfect_hits repo created in question 9.1. To do this you can click the "+" at the top right of the page, choose import repository, and then paste the URL of your perfect_hits repo in the "old repository clone URL" box. Name your copy something like "find_homologs"
2. Clone your new repo to a location on your computer
3. rename the "find_perfect_matches.sh" script to be "find_homologs.sh" using `git mv` and commit the change
4. Using git to track changes to your script (by adding and committing them), modify the script so that it performs the following function:

   Given a protein sequence query, perform a blast search against a nucleotide subject (This website should help determine which BLAST program to use)
   Filter hits to keep only hits with >30% sequence identity and >90% match length (90% of the query sequence length)

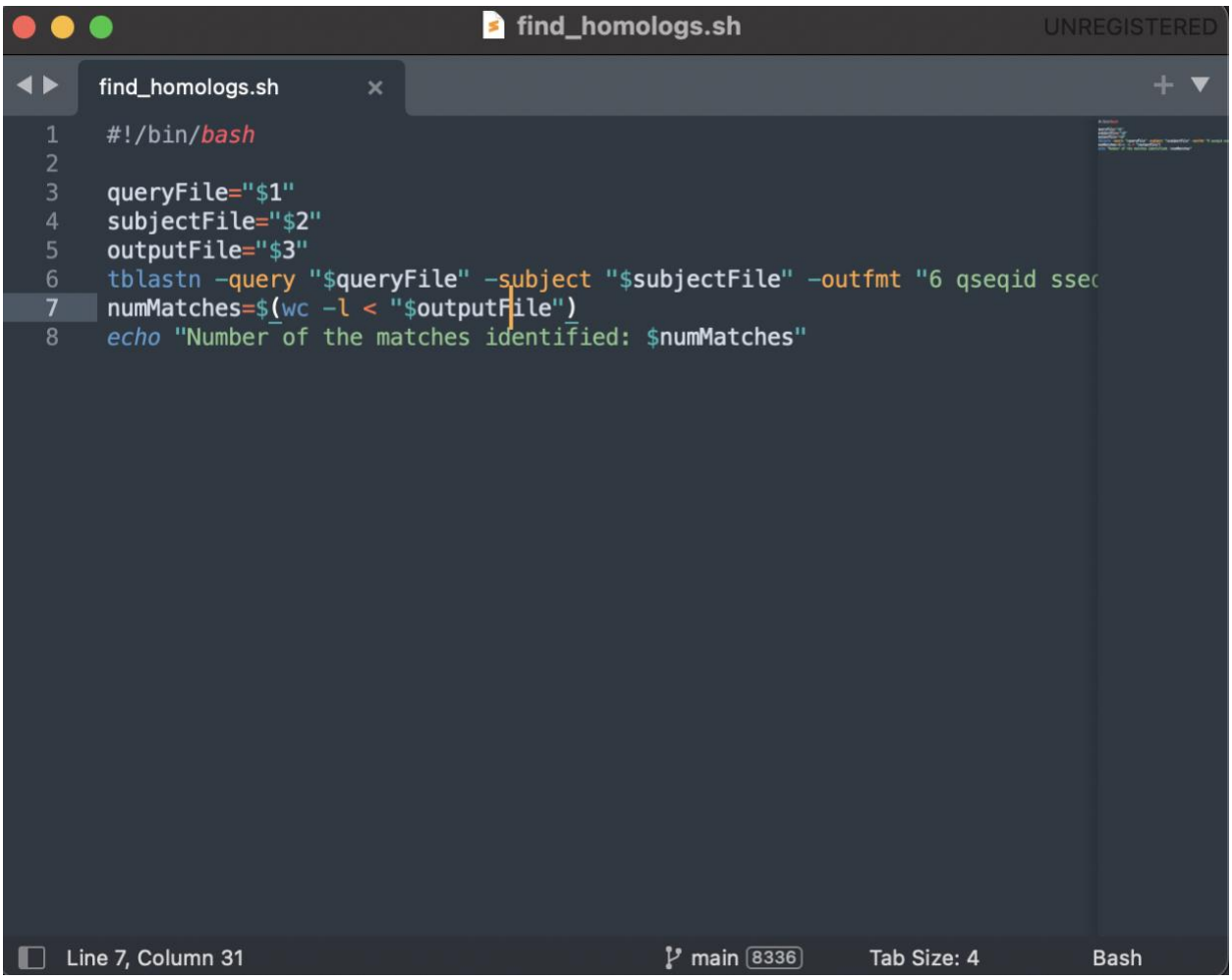
   Output matches to a specified file

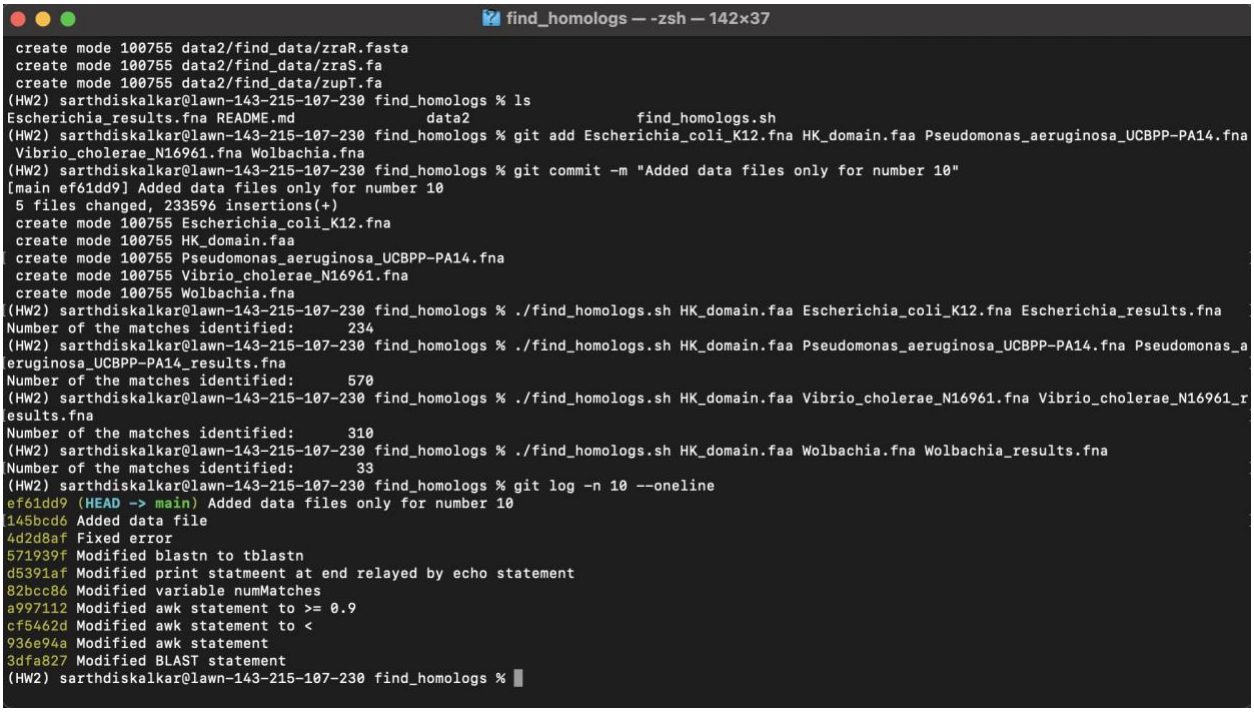
Print the number of matches identified
Usage of your script should be `./find_homologs.sh <query file> <subject file> <output file>`

In your submission sheet include the following:

- The content of your find_homologs.sh script
  - A screenshot of your terminal showing the last 10 commits with one commit per line (i.e., not the default output) using the command `git log -n 10 --oneline`. Note if you made fewer than 10 commits that is acceptable. You need only show that you made commits.
  - A screenshot of your terminal showing the output of your script when used to identify homologs of the "HK_domains.faa" sequences in each of the four bacterial assemblies provided in the "data/" dir of the "week3_data.tar.gz" archive ("Escherichia_coli_K12.fna", "Pseudomonas_aeruginosa_UCBPP-PA14.fna", "Vibrio_cholerae_N16961.fna", and "Wolbachia.fna")

```
#!/bin/bash

queryFile="$1"

subjectFile="$2"

outputFile="$3"

tblastn -query "$queryFile" -subject "$subjectFile" -outfmt "6 qseqid sseqid pident length" | awk '$3 > 30 && $4 >= 0.9 * length($1)' > "$outputFile"

numMatches=$(wc -l < "$outputFile")

echo "Number of the matches identified: $numMatches"
```


find_homologs.sh
UNREGISTERED
find_homologs.sh
1 #!/bin/bash
2
3 queryFile="$1"
4 subjectFile="$2"
5 outputFile="$3"
6 tblastn -query "$queryFile" -subject "$subjectFile" -outfmt "6 qseqid ssec
7 numMatches=$(wc -l < "$outputFile")
8 echo "Number of the matches identified: $numMatches"
Line 7, Column 31
main 8336
Tab Size: 4
Bash


find_homologs — -zsh — 142×37
create mode 100755 data2/find_data/zraR.fasta
create mode 100755 data2/find_data/zraS.fa
create mode 100755 data2/find_data/zupT.fa
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % ls
Escherichia_results.fna README.md data2 find_homologs.sh
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % git add Escherichia_coli_K12.fna HK_domain.faa Pseudomonas_aeruginosa_UCBPP-PA14.fna
Vibrio_cholerae_N16961.fna Wolbachia.fna
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % git commit -m "Added data files only for number 10"
[main ef61dd9] Added data files only for number 10
5 files changed, 233596 insertions(+)
create mode 100755 Escherichia_coli_K12.fna
create mode 100755 HK_domain.faa
create mode 100755 Pseudomonas_aeruginosa_UCBPP-PA14.fna
create mode 100755 Vibrio_cholerae_N16961.fna
create mode 100755 Wolbachia.fna
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % ./find_homologs.sh HK_domain.faa Escherichia_coli_K12.fna Escherichia_results.fna
Number of the matches identified: 234
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % ./find_homologs.sh HK_domain.faa Pseudomonas_aeruginosa_UCBPP-PA14.fna Pseudomonas_a
eruginosa_UCBPP-PA14_results.fna
Number of the matches identified: 570
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % ./find_homologs.sh HK_domain.faa Vibrio_cholerae_N16961.fna Vibrio_cholerae_N16961_r
esults.fna
Number of the matches identified: 310
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % ./find_homologs.sh HK_domain.faa Wolbachia.fna Wolbachia_results.fna
Number of the matches identified: 33
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs % git log -n 10 --oneline
ef61dd9 (HEAD -> main) Added data files only for number 10
145bcd6 Added data file
4d2d8af Fixed error
571939f Modified blastn to tblastn
d5391af Modified print statmeent at end relayed by echo statement
82bcc86 Modified variable numMatches
a997112 Modified awk statement to >= 0.9
cf5462d Modified awk statement to <
936e94a Modified awk statement
3dfa827 Modified BLAST statement
(HW2) sarthdiskalkar@lawn-143-215-107-230 find_homologs %