# Homework 1

CSCI 5481, Computational Techniques for Genomics
University of Minnesota
Instructor: Dan Knights

## Instructions

- Please turn this assignment in on the course Canvas page.
- If there are multiple files to turn in, all text and code should be placed into a single folder with a name like *lastname_homework1*. The folder should then be compressed and submitted as a single archive (.zip or .tgz)
- You are encouraged to discuss this project with others, but you need to write your own code.
- Please write the names of anyone with whom you discussed this assignment at the top of your assignment.

## Background

The purpose of this exercise is to get you familiar with downloading and processing genomic data, and with thinking about differences between coding sequencing and non-coding sequences and differences between nucleotide (DNA) and amino acid (protein) sequences.

## Tasks

1. Download the whole-genome and separate-gene DNA sequences of SARS-CoV-2, the virus that causes COVID-19, from the Homework01 directory in the Files section of the course Canvas page.

2. (20 points) Write a program that counts how many times each 3-character *codon* (substring of 3 characters) appears in the whole-genome file, starting with characters 1-3, then characters 4-6, and so on till the end of the genome. If there are extra characters at the end because it is not perfectly divisible by 3, just ignore the last one or two. Your program should run something like this, OK to invoke python/R whatever interpreter too:

   ```
   count_codons input.fna output.csv
   ```

   Your program should output a comma-separated (CSV) file with the codon string in column 1 and the total count in column 2, like this:

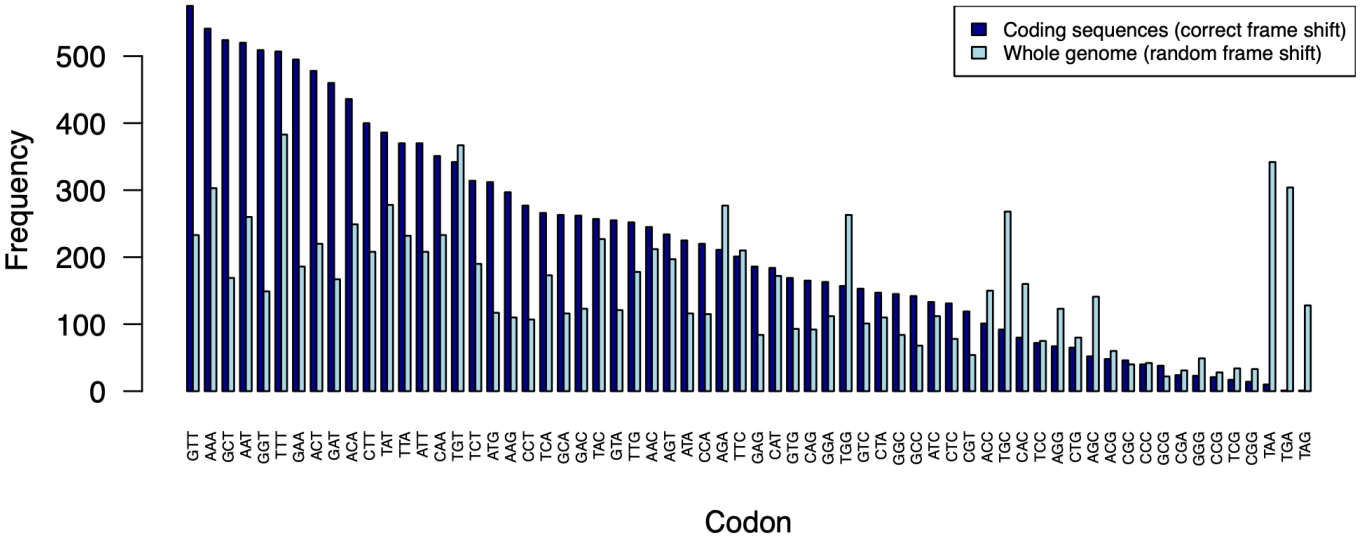
   ```
   TTT,383
   TGT,367
   ...
   ```

   Include this code in your submission.

3. (20 points) Create a small fake genome file where you know the expected answer, and test your program for correctness. Include this test genome file and the expected answer (in a separate CSV file). You don't need to turn in your output since it should be exactly the same as the expected output.

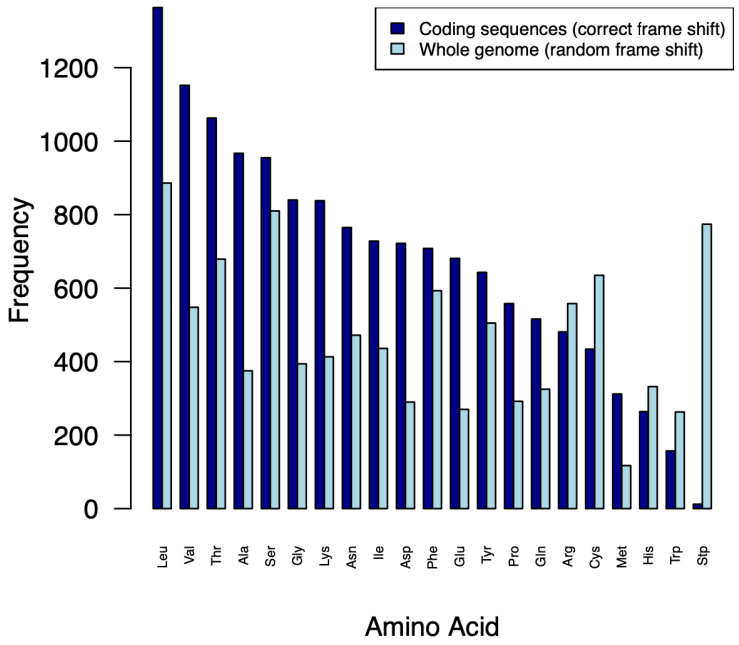
4. (20) Run the program on both input DNA files: the whole-genome file, and the file with the genome split into separate genes. Include the output CSV files with your submission.

5. (20) Make a barplot comparing side-by-side the counts of each codon in the two different files, sorted by count in the separate-gene file. You can use any program for visualization such as R, Python, or Excel. Your plot should look approximately like this:




You do not need to turn in your code for this step, only the figure.

6. (10 points) Convert the codon counts from your two CSV files to amino acid counts using the genetic code. You can use this table or an external package. Then make a similar barplot comparing amino acid counts between the two files. Your plot should look approximately like this:




Turn in the figure for this step.

7. (10 points) Where is the largest discrepancy in amino acid counts between the coding sequences and the whole genome sequence, and why?

Added Tuesday 9/26:

Note: your answer needs to be very clear to receive full credit. There may be partial credit if you describe part of the answer correctly.

Here are two hints for question 7:

Hint 1. You can see at what genomic position each gene starts and ends in the separate_genes file, which I have pulled out for your convenience here. Note that the second gene is a little weird because is actually part of the first gene, so you can just ignore the second gene for this purpose. Knowing that the whole genome is 29,904 bases long, how many bases are not contained in a gene? Is this enough to generate the ~800 extra stop codons shown in problem 6? (recall that you need 3 bases for every 1 codon)

```
location=join(266..13468,13468..21555)     gene=ORF1ab
location=266..13483   gene=ORF1ab
location=21563..25384 gene=S
location=25393..26220 gene=ORF3a
location=26245..26472 gene=E
location=26523..27191 gene=M
location=27202..27387 gene=ORF6
location=27394..27759 gene=ORF7a
location=27756..27887 gene=ORF7b
```

```
location=27894..28259 gene=ORF8
location=28274..29533 gene=N
location=29558..29674 gene=ORF10
```

Hint 2: In your script for problem 2 to count codons, try printing out the locations of the first 10 stop codons (TAA, TAG, or TGA). What positions are they? Compare this to the start/end locations of the genes. Why don't these stop codons show up when you count codons in the separate genes file?

Hint 3: What position does the first start codon, ATG, show up in the whole genome file? Why didn't it show up at position 266 (265 if zero-indexed), where the first gene starts?

## Bonus

1. (3 points) Download an additional SARS-CoV-2 virus from NCBI and design a simple computational approach method to list all mutations between the two viruses that occur inside the known coding sequencing/genes. See the instructions in the Appendix below. The easiest way to do this is to use separate-gene versions of the reference and the new genome. Otherwise you will have more insertions/deletions that will mess up the frame shift. For each mutation, list:
    a. the nucleotide in the reference genome
    b. the different nucleotide in the other genome
    c. the amino acid for that nucleotide's codon in the reference genome
    d. the amino acid for that nucleotide's codon in the other genome.

    The output format should be a 6-column CSV with these columns:

    - gene name (or gene start position, derived from sequence header)
    - position within gene
    - reference nucleotide
    - new nucleotide
    - reference amino acid
    - new amino acid

    How many mutations were there? How many of them changed the amino acid being produced?

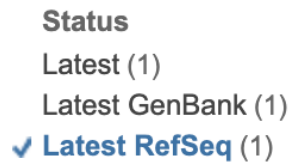
## Deliverables

1. Please turn in, via Canvas:
    a. Your well-commented code for step 2 above. Note in your code the names of any people with whom you discussed the assignment.
    b. The fake genome, expected result (csv), and your program's output in step 3 (csv).
    c. The output CSV files in step 4 above.
    d. The output figures for steps 5 and 6 above.
    e. Your answer to question 7 above in txt/doc/pdf.

## Appendix

For your reference, here is one way to find the raw genome files on the internet. This is provided only in case you are curious. You do not need to do these steps because the DNA files are already provided for you.

1. Google “NCBI” and go to the main NCBI webpage.
2. Select “Assembly” from the dropdown menu next to the search bar, and search for “SARS-CoV-2”
3. On the left, select only “Latest RefSeq” to find the latest reference version, or select “Latest” to see other genomes.




4. Click on the link to the latest RefSeq assembly of the genome. Note: this version is normally also the first result in the list if you don’t filter the results.
5. Click the blue "Download" button near the top. If there isn't one, look for a link (normally on the right side) for “FTP directory for RefSeq assembly”.
6. If it asks you what file types, click "Genomic coding sequences (FASTA)". Or if you are in an FTP directory, download a file with a name like, “GCF_009858895.2_ASM985889v3_cds_from_genomic.fna.gz” for the separate-gene sequences.
7. Unzip the file.
8. Note: FASTA-formatted files from NCBI usually have line wraps on the DNA sequences after every 80 characters, which makes the files annoying to parse. You can remove the extra line wraps with:

```
awk '/^>/ {printf("\n%s\n",$0);next; } { printf("%s",$0);}  END
{printf("\n");}' < input.fna | sed '/^$/d' > output.fna
```