Exercise 1

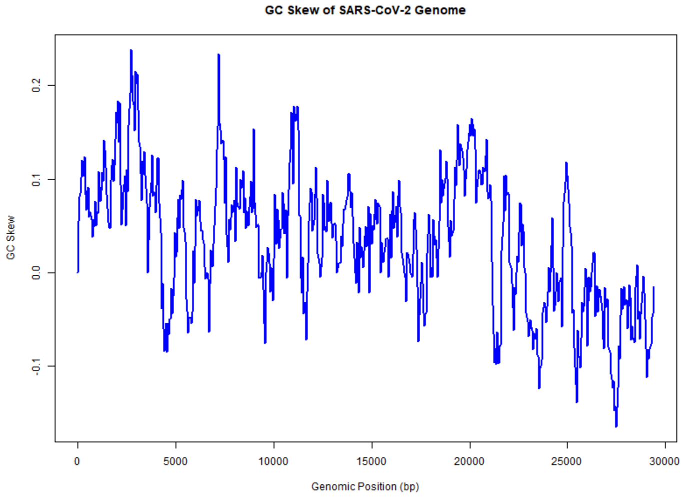
Author: Paolo Fabbri
Date: 19 January 2026

Introduction

In this analysis, we examine the genomic sequence of SARS-CoV-2 relative to other coronaviruses using some of the statistical and computational techniques introduced during the first lectures of the course, which provide foundational tools for the study of biological sequences.

1. Analyzing sequence composition and open reading frames

First, we retrieved the SARS-CoV-2 genomic sequence from GenBank and computed the GC skew values using a sliding window approach, with a window size of 500 base pairs and a step size of 50 base pairs. The GC skew value represents the relative imbalance between Guanine (G) and Cytosine (C) nucleotides within a sequence and helps to analyze local variations in nucleotide composition.
We then plotted the GC skew against the genomic position. As shown in the graph, the GC skew tends to decrease as the genomic position increases, which suggests compositional heterogeneity across different regions of the genome.



In the preparation of this report, we used AI tools to support the analysis and presentation of results. Specifically, ChatGPT was used to review R code, clarify the function of commands, and understand parameter settings. Gemini was used to improve the styling and readability of plots and figure legends. All interpretations, conclusions, and final decisions remain the authors' own.

Afterwards, we applied the findORFs() function from the Biostrings library to extract all ORFs from the SARS-CoV-2 genome. ORFs are special DNA sequences that have the potential to encode proteins.
In this analysis, we focused on identifying ORFs that start with a start codon (ATG) and terminate at the first in-frame stop codon (TAA, TAG, or TGA). As required from the point 2 of the exercise, we report the position and length of the longest ORF:

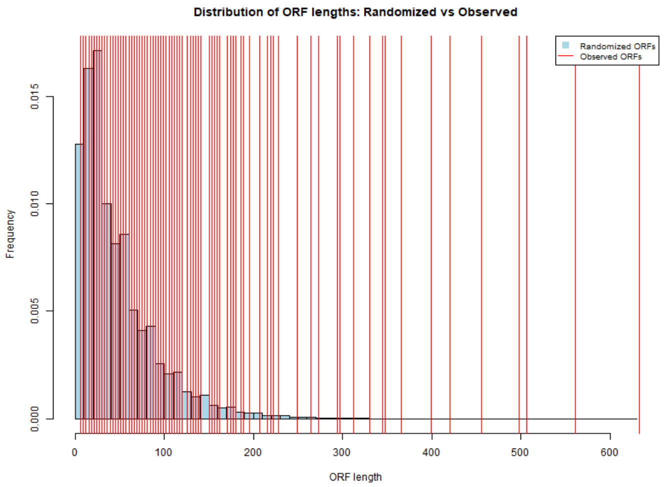
```
Longest ORF in SARS-CoV-2 genome:
Start position:  266
End position:    13483
Length:          13218 bp
```

After obtaining the observed ORFs, the next step in a proper analysis is to perform a permutation test to assess whether the ORFs are generated randomly or if their occurrence is dependent, rather than independent, across the sequence.
We apply a randomization test through a loop of 1000 iterations where the original sequence was shuffled, and the findORFs() function was applied in each iteration to identify the ORFs for that specific permutation. All the randomly generated ORFs were then used to construct a null distribution (H0), which was compared to the observed ORFs in order to calculate empirical p-values.
This process is the hypothesis test, which creates our null hypothesis, against which we set our alternative hypothesis: ORFs are longer than expected by chance. This allows us to answer the question: *Is this ORF longer than we would expect if nucleotides were randomly arranged?*

Once we obtained the distribution of ORF lengths from the randomized sequences, we calculated the empirical p-value for each observed ORF.
We then counted how many ORFs had a p-value below <0.01, which correspond to ORFs that are longer than we would expect if nucleotides were randomly arranged.

Number of significant orfs: 200

In the graph we can see the distribution of ORF lengths from the randomized sequences at which we overlay the lengths of the observed ORFs from the original sequence.
The graph confirms our statistical approach idea: the distribution of randomized ORF lengths is concentrated almost entirely on the left side of the graph. This indicates that, in a random sequence, it is extremely unlikely for long coding regions to form spontaneously.
In contrast, the red lines represent the length of observed ORFs. Several of these are positioned on the right tail of the random distribution; these are the significant values according to our p-value threshold (0.01). telling us that this is very unlikely that they can be generated randomly. Their position demonstrates that it is highly improbable for such lengths to be generated by chance, suggesting that they are biologically relevant sequences.



As we have studied, when we perform hypothesis tests a very large number of times, we need more reliable measures to control for false positives. Two widely used approaches are the Bonferroni and Benjamini-Hochberg (BH) correction methods. These are “correction” techniques that make the p-value threshold more stringent and reduce the risk of false positives:

1. Bonferroni correction: This is the most conservative method. The significance level α is divided by the total number of tests (n). It is very safe against false positives, but it increases the risk of false negatives.

2. Benjamini-Hochberg (BH) correction: Instead of controlling the risk of a *single* false positive, it controls the *proportion* of false positives among all significant results. This method is less conservative than Bonferroni and allows for more discoveries while still limiting false positives.

These are the results of our tests, and as expected, the number of significant ORFs is lower after applying both the Bonferroni and Benjamini-Hochberg corrections compared to the number of significant ORFs calculated previously. Additionally, the Bonferroni correction is more stringent than the Benjamini-Hochberg method.

```
Multiple testing correction results:

Bonferroni correction:
Number of significant ORFs (p < 0.01):  176

Benjamini-Hochberg correction:
Number of significant ORFs (p < 0.01):  192
```

---

2. Identifying the spike protein encoding ORF

Now the task is to identify the SARS-CoV-2 ORF that codes for the spike protein using a probabilistic model trained on spike protein coding nucleotide sequences from other human coronaviruses (HCoV-229E, HCoV-NL63, HCoVOC43, HCoV-HKU1, SARS-CoV, and MERS-CoV). The spike proteins are critical for viral entry into host cells, making them important targets for therapeutics.

We create a function to train a first-order Markov model by estimating the nucleotide transition probabilities from the sequences of spike protein coding regions from the other human coronaviruses. A first-order Markov model allows us to assess the probability of observing a nucleotide given only the previous one (P(Xi+1 | Xi)). By computing these transition probabilities across all training sequences, we constructed a 4×4 nucleotide transition matrix representing the characteristic nucleotide patterns of spike protein coding regions.

The transition matrix was constructed by counting how many times each nucleotide follows another nucleotide across the training sequences. These counts were then normalized by the length of the sequence in order to obtain probabilities.

Subsequently, we built a function to compute the log-likelihood of each significant ORF. We used the previously trained first-order Markov model to calculate the conditional probabilities of observing each nucleotide given the previous one.
As highlighted in the exercise text, when dealing with multiplication of small probabilities, it is better to use Log-likelihood in order to prevent numerical underflow. Therefore, we set the initial probability P(x0) to 1 and then sum the logarithms of the conditional transition probabilities along the sequence. Finally, we normalize the log-likelihood values to account for differences in ORF lengths.

In the final step we extract the nucleotide sequences of significant ORFs identified using the Bonferroni correction in Question 3. We loop these ORF sequences and compute the loglikelihood of each one of them under the first-order Markov derived from the spike protein coding nucleotide sequences of the other human coronaviruses. Finally, we identify the ORF with the maximum normalized log-likelihood, which is therefore the most likely candidate to encode the spike protein in SARS-CoV-2.

Most likely sequence to be the one coding for the spike protein in SARS-CoV-2:
Start position: 6551
End position: 13483
Length: 6933
Normalized Log-Likelihood: -1.322286

---

3. Identifying the spike protein encoding ORF

The next task is to explore how SARS-CoV-2 relates to other human coronaviruses by comparing their spike protein sequences. To keep the computational complexity reasonable, we focus on the spike proteins rather than the entire viral genome.
We loaded the nucleotide sequences coding for the spike proteins from "*data/coronaviruses spike.fasta*" and "*data/SARS-CoV-2 spike.fasta*". Then, we translate the nucleotide sequences into amino acid sequences verifying if each one of them starts with methionine (M), which is indicative of a correct start codon (ATG).

To assess the overall similarity between the SARS-CoV-2 spike protein and the spike proteins of other human coronaviruses, we performed global pairwise alignments at the amino acid level. We used the Needleman-Wunsch algorithm with the BLOSUM62 substitution matrix and we set the opening gap (cost to start a new gap. A high penalty discourages opening too many gaps) penalty to 10 and the extension gap (additional cost to extend an already opened gap. A lower penalty allows long gaps without making the score too negative.) penalty to 2.

- Global alignment aims to maximize the alignment quality over the whole sequences

---

```
Human coronavirus most similar globally to SARS-CoV-2 spike protein:
Name: Human-SARS
Global alignment score: 5156
```

For each coronavirus spike protein, we computed the global alignment score against the SARS-CoV-2 spike protein. The sequence with the highest global alignment score was identified as the most globally similar spike protein to SARS-CoV-2.

We then repeated the same experiment but computing the local alignment.

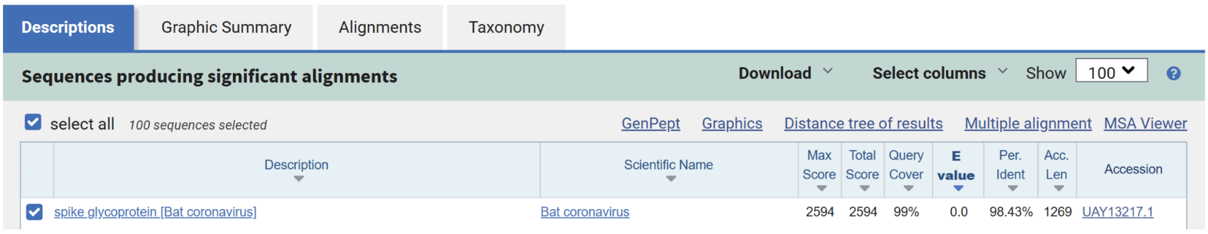
- Local alignment, on the other hand, searches for the most similar regions between two sequences

---

```
Human coronavirus most similar to SARS-CoV-2 spike protein:
Name: Human-SARS
Local alignment score: 5260
```

BLAST Analysis of SARS-CoV-2 Spike Protein

At the end, we conduct a BLAST search online by uploading the SARSCoV-2 spike protein sequence and searching against a database of bat coronaviruses. We identify and report the top bat coronavirus with the highest similarity to SARS-CoV-2 based on the spike protein sequence.
These are the results:



Sequences producing significant alignments

☑ select all 100 sequences selected

☐	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
☑	spike glycoprotein [Bat coronavirus]	Bat coronavirus	2594	2594	99%	0.0	98.43%	1269	UAY13217.1

BLASTp analysis of SARS-CoV-2 spike protein against bat coronaviruses
Top BLAST hit:
Description: Spike glycoprotein
Accession code: UAY13217.1
Alignment score:
E-value: 0.0
Percentage identity: 98.43%
Host species: Rhinolophus malayanus

```
FEATURES             Location/Qualifiers
     source          1..1269
                     /organism="Bat coronavirus"
                     /isolate="BANAL-20-52/Laos/2020"
                     /isolation_source="rectal swabs"
                     /host="Rhinolophus malayanus"
                     /db_xref="taxon:1508220"
                     /geo_loc_name="Laos"
                     /collection_date="05-Jul-2020"
```