BINF200 Assignment 2

Tom Michoel

2023-09-29

1 Deadline, grading and report

The assignment is due **13 October 2023**.

The assignment is scored on **20 points** and counts towards **10% of the final grade**.

Your report should be a **single PDF file** that contains your report text, code, and figures in a single document. The easiest workflow is probably to run your analyses in a Jupyter or similar notebook, and save the final notebook as a PDF file.

You *may* work together, but you *must* declare it in your report.

Any use of ChatGPT or other generative AI tools *must* be declared in your report.

2 Background

In this compulsory assignment, you will perform bioinformatics analyses covering multiple sequence alignment, phylogenetic tree construction and motif finding. The assignment will cover practical use of state-of-the-art tools, and also questions requiring programming. You may use any programming language: Python, Julia, R, …

3 Data

Download all files in the following OneDrive folder:

https://universityofbergen-my.sharepoint.com/:f:/r/personal/tom_michoel_uib_no/Documents/public/BINF200/Coronavirus?csf=1&web=1&e=D5umem

You should find the following files:

- **protein_N_data.fasts** - Sequences of the gene coding for coronavirus nucleocapsid (N) protein in a number of coronaviruses
- **GCA_011537005.1_partial_genomic.fasta** - Part of the BetaCoV/Wuhan/IPBCAMS-WH-02/2019 genome

- **motifCountMatrix.csv** - Count matrix of a sequence motif

4 Tasks

4.1 Multiple sequence alignment and phylogenetic tree construction for the coronavirus nucleocapsid protein

We will focus on different genera of corona viruses namely: alpha, beta, gamma and delta. Their genomes, gene and protein sequences, together with annotations and data reports are available from NCBI:

- Alpha coronavirus: https://www.ncbi.nlm.nih.gov/datasets/taxonomy/693996/
- Beta coronavirus: https://www.ncbi.nlm.nih.gov/datasets/taxonomy/694002/
- Gamma coronavirus: https://www.ncbi.nlm.nih.gov/datasets/taxonomy/694013/
- Delta coronavirus: https://www.ncbi.nlm.nih.gov/datasets/taxonomy/1159901/

For simplicity, we will use data from only one important gene, that encodes the coronavirus nucleocapsid (N) protein. This is a structural protein that forms complexes with genomic RNA, interacts with the viral membrane protein during virion assembly and plays a critical role in enhancing the efficiency of virus transcription and assembly. You can read more about it in the paper *The SARS-CoV-2 Nucleocapsid Protein and Its Role in Viral Structure, Biological Functions, and a Potential Target for Drug or Vaccine Mitigation*.

The datasets listed in Figure 1 were used to create **protein_N_data.fasta**.

4.1.1 Parse the fasta file

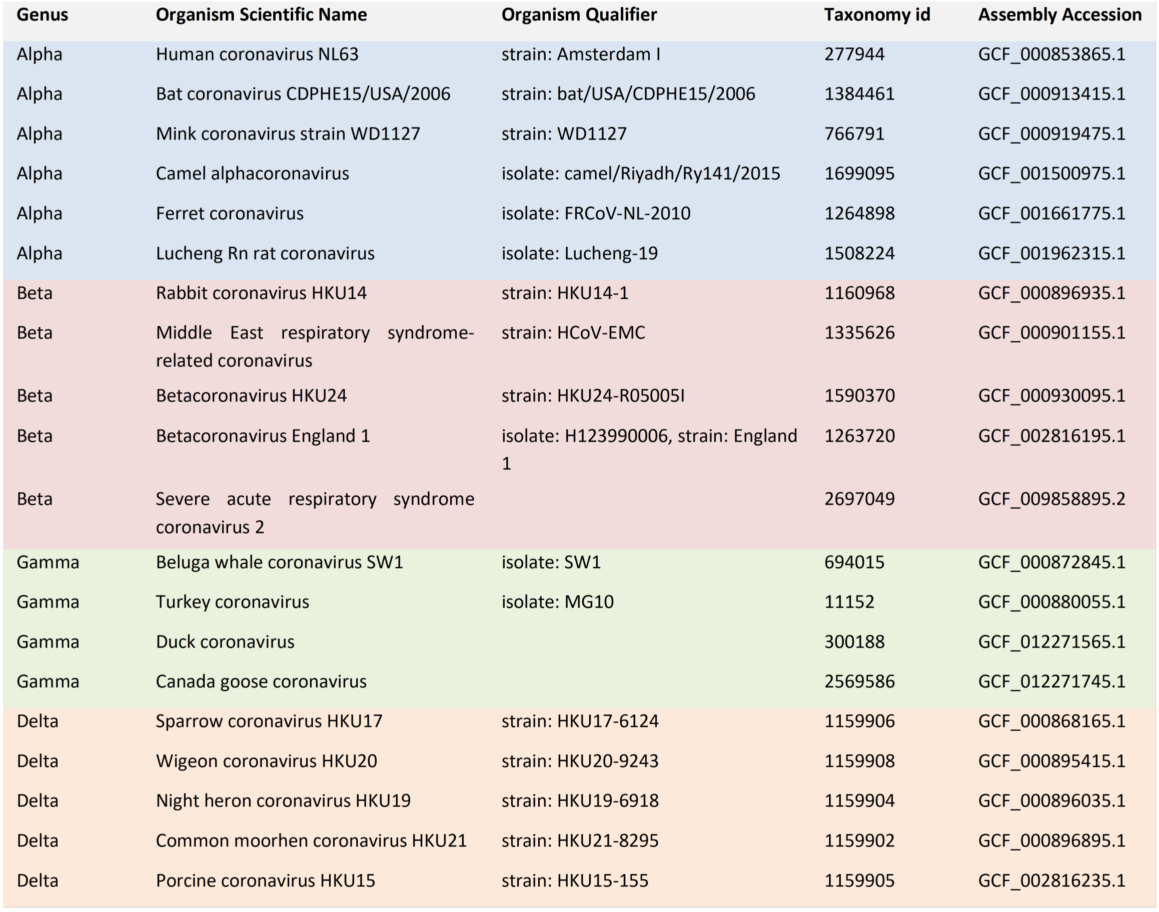
How many sequences are contained in the file **protein_N_data.fasta**?

List the names of the sequences.

4.1.2 Find protein N in a specific coronavirus genome

From the sequences in **protein_N_data.fasta**, find the sequence for which the first letter in its name is closest in the alphabet to the **first letter in your first name**. If there are multiple sequences starting with the same letter, pick one arbitrarily. For the selected sequence:

- Find the assembly accession ID in the table above.
- Go to the NCBI website (cf. links above) and find the corresponding genome assembly.
- What are the genomic coordinates (start and end position) of gene N in this genome? (Hint: follow the RefSeq link)



Genus	Organism Scientific Name	Organism Qualifier	Taxonomy id	Assembly Accession
Alpha	Human coronavirus NL63	strain: Amsterdam I	277944	GCF_000853865.1
Alpha	Bat coronavirus CDPHE15/USA/2006	strain: bat/USA/CDPHE15/2006	1384461	GCF_000913415.1
Alpha	Mink coronavirus strain WD1127	strain: WD1127	766791	GCF_000919475.1
Alpha	Camel alphacoronavirus	isolate: camel/Riyadh/Ry141/2015	1699095	GCF_001500975.1
Alpha	Ferret coronavirus	isolate: FRCoV-NL-2010	1264898	GCF_001661775.1
Alpha	Lucheng Rn rat coronavirus	isolate: Lucheng-19	1508224	GCF_001962315.1
Beta	Rabbit coronavirus HKU14	strain: HKU14-1	1160968	GCF_000896935.1
Beta	Middle East respiratory syndrome-related coronavirus	strain: HCoV-EMC	1335626	GCF_000901155.1
Beta	Betacoronavirus HKU24	strain: HKU24-R05005I	1590370	GCF_000930095.1
Beta	Betacoronavirus England 1	isolate: H123990006, strain: England 1	1263720	GCF_002816195.1
Beta	Severe acute respiratory syndrome coronavirus 2		2697049	GCF_009858895.2
Gamma	Beluga whale coronavirus SW1	isolate: SW1	694015	GCF_000872845.1
Gamma	Turkey coronavirus	isolate: MG10	11152	GCF_000880055.1
Gamma	Duck coronavirus		300188	GCF_012271565.1
Gamma	Canada goose coronavirus		2569586	GCF_012271745.1
Delta	Sparrow coronavirus HKU17	strain: HKU17-6124	1159906	GCF_000868165.1
Delta	Wigeon coronavirus HKU20	strain: HKU20-9243	1159908	GCF_000895415.1
Delta	Night heron coronavirus HKU19	strain: HKU19-6918	1159904	GCF_000896035.1
Delta	Common moorhen coronavirus HKU21	strain: HKU21-8295	1159902	GCF_000896895.1
Delta	Porcine coronavirus HKU15	strain: HKU15-155	1159905	GCF_002816235.1

Figure 1: Table of coronavirus source datasets

4.1.3 Multiple sequence alignment

Build a multiple sequence alignment for the **protein_N_data** using a multiple sequence alignment tool of your choice. (Hint: check out the services provided by EMBL's European Bioinformatics Institute (EMBL-EBI).)

4.1.4 Phylogenetic tree reconstruction

Based on the results from the previous step, build a phylogenetic tree. (Hint: at this stage it is not required to make an "advanced tree", providing a simple tree is enough). Save the image of the phylogenetic/guide tree.

4.1.5 Interpretation

Based on the results from the previous two steps, what do you see? Elaborate with a small text (3-4 lines): Explain what you observe from the multiple sequence alignment itself (hint: check the number of conserved sites), and give a short interpretation of the phylogenetic tree you have constructed.

4.2 Step-by-step multiple sequence alignment and phylogenetic tree construction using UPGMA

4.2.1 Compute pairwise similarities

Use the Needleman-Wunsch (dynamic programming) pairwise alignment algorithm to build a matrix of global alignment scores for each pair of sequences in **protein_N_data.fasta**. You can choose between multiple options:

- Implement the Needleman-Wunsch algorithm yourself. (Hint: You have probably done this already in BINF100)
- Use an existing implementation of the algorithm. (Hint: Check biopython, biojulia)
- Use the *needleall* command line program from the EMBOSS suite. (Hint: You installed the whole EMBOSS suite for Assignment 1.)
- Use a webserver such as EMBL-EBI's EMBOSS Needle service. (Hint: Manually inputting every pair of sequences will be extremely tedious, though they do provide APIs.)

4.2.2 Generate a pairwise distance matrix

Generate a distance matrix from the score matrix you have created in the previous step. For this task we will use Feng & Doolittle's formulation, and we will compute the distance D using formula:

$$D = -\log S_{eff} = -\log \frac{S_{obs} - S_{rand}}{S_{max} - S_{rand}}$$

where

- S_{obs} is the observed pairwise alignment score
- S_{max} is the best alignment score for both sequences, obtained by taking the average of the score of aligning either sequence to itself
- S_{rand} is the expected (average) score for aligning two random sequences of the same length and residue composition, obtained by random shuffling the nucleotide composition of the two sequences. (Hint: more info about the Feng & Doolittle can be found at this URL: https://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Feng-Doolittle)

Compute S_{rand} by taking the average score of **10** pairwise alignments between random sequences with the same sequence compositions as the original sequences.

4.2.3 Generate a "guide tree" of phylogenetic relationships

Generate a "guide tree" of phylogenetic relationships from the pairwise distance matrix you have created in the previous step using the UPGMA method. You can choose between multiple options:

- Implement the UPGMA hierarchical clustering algorithm yourself. (Hint: You can represent the tree as a binary tree, either implementing a tree class yourself, or using an existing data structure.)
- Use an existing implementation of the algorithm. (Hint: UPGMA is more commonly known as hierarchical clustering with average linkage. Check SciPy or similar packages for other languages.)

4.2.4 Interpret your results

Visualize your guide tree and compare it to the phylogenetic tree constructed in Section 4.1.4. Elaborate with a small text (3-4 lines) to explain what you observe.

4.3 Sequence motifs

Do simple motif searching on corona virus sequences using the input dataset (**protein_N_data.fasta**) we have already analysed.

4.3.1 MEME analysis

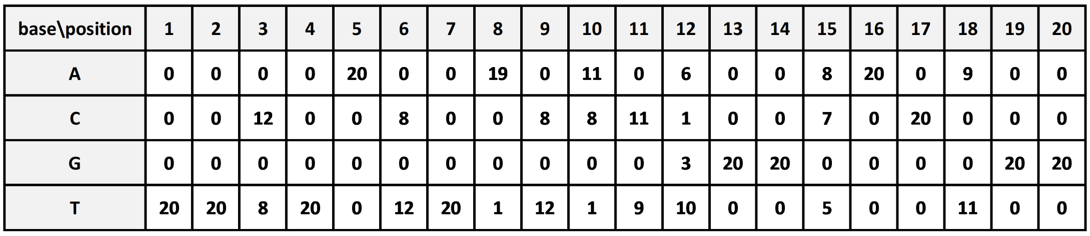
Connect to the MEME platform at https://meme-suite.org/.

- Find the MEME motif discovery tool.
- Input **protein__N__data.fasta** to discover enriched motifs in this set of sequences, allowing for zero or one motif occurrence per sequence and finding upto 5 motifs. Which discovery mode, sequence alphabet, and site distribution options do you select?

Open and download the **MEME HTML output file** and include the sequence logos of the motifs found in your report.

4.3.2 Convert count matrix to PWM

We will work with a 20-nucleotide subset of the first motif found by the MEME software, given by the count matrix:



base\position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	0	0	0	0	20	0	0	19	0	11	0	6	0	0	8	20	0	9	0	0
C	0	0	12	0	0	8	0	0	8	8	11	1	0	0	7	0	20	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	3	20	20	0	0	0	0	20	20
T	20	20	8	20	0	12	20	1	12	1	9	10	0	0	5	0	0	11	0	0

Figure 2: Motif count matrix

The count matrix is also available as a file **motifCountMatrix.csv**.

1. Compare the count matrix against your sequence logos and mark the 20-nucleotide window corresponding to this count matrix in the right logo.

2. Convert the count matrix to a position-specific probability matrix (PPM) P. To avoid zeros in the PPM, we add *pseudo-counts* and define

$$P_{k,i} = \frac{\text{Count}_{k,i} + 0.25 * \sqrt{N}}{N + \sqrt{N}},$$

   where $\text{Count}_{k,i}$ is the value of the count matrix for nucleotide k in motif position i, and N is the number of sequences in **protein__N__data.fasta** (Hint: Count the totals in each column of the count matrix).

3. Convert the PPM matrix to a position-specific weight matrix (PWM) W using the formula

$$W_{k,i} = \log_2 \frac{P_{k,i}}{0.25}$$

What would be the value of W for a random background site with equal counts for all nucleotides and using the pseudo-count formula above to compute the random probabilities?

4.3.3 Scan a coronavirus genome for motif occurrences

Scan part of the BetaCoV/Wuhan/IPBCAMS-WH-02/2019 genome (the sequence in the file **GCA_011537005.1_partial_genomic.fasta**) and score all possible motif occurrences. Use the sliding window approach presented in the lecture and report (table and figure) both the log-odds score and the odds of each possible motif starting position in the genome sequence.

Elaborate with a small text (3-4 lines) to explain what you observe.