BINF200 Assignment 2

Tom Michoel

2023-09-26

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 Tasks

3.1 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*.

Download a dataset (**protein_N_data.fasta**) with the sequences of interest from OneDrive or copy this URL:

https://universityofbergen-my.sharepoint.com/:f:/r/personal/tom_michoel_uib_no/Documents/public/BINF200/Coronavirus-protein-N?csf=1&web=1&e=RINJI2

The following datasets were used to create **protein_N_data.fasta**:

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

3.1.1 Parse the fasta file

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

List the names of the sequences.

3.1.2 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 on 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)

3.1.3 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).)

3.1.4 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.

3.1.5 Interpretation

Based on the results from the previous two steps, what do you see? Please elaborate with a small text (3-4 lines), and 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.

3.2 Implementation of multiple sequence alignment and phylogenetic tree construction using UPGMA

3.2.1 Implement the Needleman-Wunsch alignment algorithm

Write a function to build a matrix of global alignment scores of each pair of sequences using Needleman-Wunsch alignment. A python sample code (**Needleman_Wunsch.py**) is provided for the Needleman-Wunsch alignment using two input sequences, but you are free to use any other implementation or pseudo code you like.

3.2.2 Implement the Feng-Doolittle progressive alignment algorithm

Write a function to generate a distance matrix from the score matrix you have created in the previous step. For this task we will use Feng & Doolittle 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} provides the best alignment score for both sequences, by taking the average of the score of aligning either sequence to itself; and S_{rand} is the expected score for aligning two random sequences of the same length and residue composition. S(rand) is calculated by random shuffling the nucleotide composition of the two sequences. (Hint: more info about the Feng & Doolittle can be found here https://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Feng-Doolittle)