CMPE 549 - Assignment 1

Fall 2021

Due 20.12.2021, 20:00

Updates 16.12.2021

Description

In this assignment, we will be working on the genome of the SARS-CoV-2 virus. For this, we will be using the data set curated by Kunal Joshi & Phillip Compeau. For this task, you are expected to load and visualize the data, conduct a multiple sequence alignment on the data, and construct a phylogenetic tree based on the data.

In this assignment, in order to keep our sequences short, we will only be focusing on the gene that encodes the spike protein of SARS-CoV-2 virus (or a subset thereof). Throughout this assignment, when we refer to the *spike protein gene*, we refer to the nucleotide sequence between the positions 21563 and 25384 (inclusive, positions are 0-indexed). The positions are determined by the annotation of SARS-CoV-2 genome.

Requirements

- You are expected to use `Python` programming language for this assignment.
- Make sure you have installed the `Biopython` package (version 1.79) for your `Python` interpreter.
- Your submission should be in the form of a single Jupyter notebook (`.ipynb`) file. The answers to the questions, your code, and your visualizations should all be in the same notebook.
- Your file must be named according to the following template, all in lower case letters: `<last name>_<first name>_<student id>_assignment_1.ipynb`. If you have multiple first names, write your preferred name for the first name.
- Clearly separate answers to different questions and subquestions with headings.
- Remember to use the markdown feature of the Jupyter notebook for your headings and textual answers.
- Note that the deadlines are sharp due to time restrictions. Since you can make multiple submissions, make sure you make a first submission at least a couple hours before the deadline to avoid a potential system error.
- If for any point of clarification below, you do not have the time or resources to make the changes necessary in your assignment, you can keep your previous answer but make sure you let us know about this in your assignment document (e.g. a highlighted note at the start of your answer).

Questions

1. This part of the assignment concerns the loading and the visualization of the data. Download the data from this link and extract it in its current folder structure. The dataset includes the sequencing of SARS-CoV-2 taken from 100 randomly selected individuals at six different time points between 2020-11-03 and 2021-12-08 (the data for 2020-11-17 are corrupted). The dataset includes both the raw data as well as its aligned version in FASTA format. For all questions that include the aligned version of data, any characters in the sequences other than 'A', 'T', 'C', 'G', and '-' must be converted to '-' before doing any processing. This is not a standard practice, it is done in this case to make your work easier here.
    a. Refer to the documentation of `Biopython` to read the FASTA file for the aligned version of the November 3rd dataset: `HCOV19-ENGLAND-031120-A.fasta`.
    b. Take the first 20 samples from this dataset, and visualize the first 200 nucleotides of the spike protein gene in the form of a matrix, such that each nucleotide is represented by a different color and empty characters are left white.
    c. What are your observations regarding the resulting visualization? How similar are different nucleotide sequences? Are there any visible mutations?
    d. Using the same aligned data, create a matrix that includes all 100 samples from November 3rd, and all of the spike protein gene sequence. Compute an entropy score for this sequence, by implementing this function yourself.

2. For this part of the assignment we will ignore the original alignment provided by the curators of the data, and we will conduct our own multiple sequence alignment, by implementing the three main steps of the ClustalW algorithm. Therefore, for this part of the assignment, we will use the original (unaligned) data for November 3rd dataset: `HCOV19-ENGLAND-031120.fasta`.
    a. Again use the `Biopyhton` package to read the data. For the first 5 samples in the data set, extract the nucleotides between positions 21600 and 21799 (inclusive) for each patient. You should end up with a nucleotide matrix with a size of 5 x 200.
    b. Implement the Needleman-Wunsch global sequence alignment algorithm to compute pairwise alignment score between any two nucleotide sequences. For scoring, use 5 for a match, -4 for a mismatch or an indel to create a pairwise similarity matrix.
    c. Convert your pairwise similarity matrix to a pairwise distance matrix by subtracting every element of your pairwise similarity matrix from the largest value in it. Print the pairwise distance matrix.
    d. Create a guide tree using the neighbor joining algorithm implemented in `Phylo` module of `Biopython` package.
    e. Using your guide tree, conduct the progressive alignment of your 5 samples according to the Needleman-Wunsch algorithm, producing the final multiple sequence alignment.
    f. Visualize the results of your alignment algorithm similarly to the first question. Note: You can use `matplotlib.pyplot.imshow` for plotting or another function of your choice.

3. For this part of the assignment, we will go back to using the aligned versions of our data (with the preprocessing step that converts nonstandard characters to '-'). Using the aligned versions of the data, extract the spike protein gene from the first 4 patients of the samples from

November 3, November 27, and December 8. You should end up with a nucleotide matrix of size 12 x 3822.*

a. Construct a distance matrix for the given 12 aligned samples (use Needleman-Wunsch algorithm you implemented for scoring, again making the conversion from pairwise similarity matrix to pairwise distance matrix). Visualize your distance matrix.

b. Implement the UPGMA algorithm without using any third-party software (see footnote below) and using it construct a phylogenetic tree of the given 12 samples. Print the tree you constructed in the Newick format with distances and leaf names.

c. Using your Newick formatted tree as an input, visualize the phylogenetic tree for the 12 samples, using the `Phylo` module of `Biopython` package.

d. Use the UPGMA algorithm provided in the `Phylo` module to construct the tree, and visualize it. Compare the result to that of your own implementation, which you visualized in 3c. Is it different from your result? If so, how and why?

e. Use the `Phylo` module to construct and visualize the phylogenetic tree of the spike protein gene for all patients from November 3, November 27, and December 8.

* For 3a and 3b, you are free to use base data structures of `Phylo` such as `BaseTree.Clade` or `BaseTree.distance_matrix` instead of e.g. `numpy.array`. However, these can only be used passively, e.g. to store your data or interim results. Your implementation of the algorithm should still be manual without any core operations outsourced to `Phylo` and be observable in a step by step fashion. Your outputs should still conform to the formats described above also.