CSSD 1101 Group Project: BRCA1 Gene Mutation Detection

Project Overview

In this project, your team will implement a software to analyze and compare DNA sequences from patients with that of a healthy BRCA1 gene sequence, referred to as the WildType (WT) sequence. The goal is to identify mutations in patient DNA, understand the consequences of these mutations at the protein level, and compare them to the WT at both the DNA and protein levels.

Learning Outcomes

1. By completing this project, you will:
2. Understand basic biological concepts such as DNA sequences, RNA transcription, and protein translation.
3. Develop computational skills, including reading DNA sequences from files, cleaning up data, and verifying nucleotide composition.
4. Implement algorithms to detect mutations (e.g., insertion, deletion, substitution) and evaluate their impact on both the DNA and amino acid sequences.
5. Visualize genetic mutations by developing graphical/text-based representations of DNA and RNA changes.
6. Collaborate effectively in a team environment to design, code, and test a software program.
7. Improve problem-solving abilities by applying programming techniques to solve biological challenges.

Project Tasks

Retrieve information about the WT sequence

- Read the WT sequence from a text file. You may need to exclude irrelevant data such as headers
- Verify that it is a DNA sequence as opposed to RNA or other invalid content or content in inappropriate format etc.
- Find the distribution of the nucleotides
- Transcribe the DNA into an RNA sequence

- Find the open reading frame (ORF) of the sequence (start to stop codon). Please note that there are multiple start codons and end codons when the sequence is translated to an RNA. Please choose the longest sequence as the ORF.
- Find the translated protein based on the ORF

Retrieve information about the patient sequence

- Read the patient sequence from a text file.
- Verify that it is a DNA sequence, and it is a BRAC1 variant sequence (as opposed to some other unrelated gene sequences). You can compare that the nucleotide counts are similar between the WT type and the patient sequence. The two sequences are similar if the count of each nucleotide differs by less than 1.
- On this DNA sequence (i.e., the patient sequence), find the ORF sequence that corresponds to the ORF of the wild type.
- Find the type of DNA mutation (i.e., substitution, deletion, insertion). You can assume the mutation only involves one nucleotide, and the mutation will only occur in the ORF and will not occur on the start codon and stop codon.
- Transcribe the DNA into an RNA sequence and translate the sequence into protein
- Identify the consequences on the patient protein (silent, missense, nonsense, frameshift, frameshift with early or later stop)

Report the findings

- Generate a visual aligning the WT DNA with that of the patient ensuring that the mutation is displayed clearly. Feel free to only include the relevant subset of the sequence where the mutation is located. For example, this can be 20 nucleotides followed by the mutation and another 20 downstream nucleotides. You may take a screen shot of your output. You can be creative!
- Similarly, generate a visual aligning the WT protein sequence with that of the patient ensuring that the change is displayed clearly. This can also be a subset of the protein sequence. You may take a screen shot of your output.
- Generate a report for this comparison of WT to patient sequence including all the results you have generated in the questions above (if your code works, the program should automatically generate a report). Consider the example report content below.

Teamwork

Teamwork is a critical skill in computer science, as most real-world software projects involve collaboration. Working in teams allows you to tackle complex problems more effectively, combine diverse ideas, and learn from each other's strengths. It fosters better problem-solving, encourages innovation, and helps you practice communication skills—key attributes in any tech career. Additionally, teamwork helps manage workloads efficiently and builds a support network for learning.

Suggestions for Effective Teamwork:

- **Set clear goals**: Define the tasks and divide responsibilities based on individual strengths.
- **Communicate regularly**: Consistent communication is key. You can use any preferred platform (e.g., Discord, Slack), but make sure to check in regularly to stay on track and share updates.
- **Be open to feedback**: Constructive feedback is essential for improving your work.
    - When giving suggestions, do so professionally and supportively. Expressing frustration or anger toward teammates is not acceptable and will not be tolerated in this course.
    - If a teammate behaves in an aggressive or inappropriate manner, prioritize your safety and avoid engaging in arguments. You are encouraged to contact York Security for assistance:
        - Urgent matters: 416-736-5333 x33333
        - General Inquiry: 416-650-8000 x58000
    - When receiving feedback, don't take it personally. Remember that input from others is meant to strengthen your team's overall performance.
- **Manage conflicts positively**: If disagreements arise, focus on finding solutions rather than dwelling on the problem.
- **Stay accountable**: Follow through on your assigned tasks and assist your teammates when needed.
- If you're unable to resolve an issue within the team, please reach out for assistance. Use the designated method under "Contact Us" on eClass (this count as a logistic question), and we will intervene as needed.

Starter Code Description

The provided starter code is a foundation for implementing the BRCA1 gene mutation detection program. It provides structure and guidelines for comparing patient DNA sequences against a healthy WT BRCA1 gene sequence. The main logic is in place, and you

are expected to complete the provided functions to fully implement the mutation detection system.

What is provided:

- Enumerations: Classes to represent different mutation types at the DNA and amino acid levels.
    - Do not worry if you do not know what enumeration is. This will be used to indicate the type of the mutation. For example, for an insertion at the DNA level, you can use: `DNAMutation.INSERTION`, you can take handle it the same way as other values we talked about in class (e.g., int), and assign it to a variable and/or return it.
- Codon Map: A pre-defined dictionary mapping RNA codons to their corresponding amino acids.
- Main Program Structure: The `main` function orchestrates the analysis process, calling each of the provided functions to read sequences, detect mutations, and generate a mutation report.
- Function Documentation: All functions in the starter code include descriptions, input/output specifications, the types of parameters, and example usage to guide the implementation.

You are required to implement the following functions:

1. DNA Sequence Processing:
    - `read_sequence_from_file`: Reads a sequence from a file.
    - `verify_dna_sequence`: Ensures that the sequence is a valid DNA sequence.
    - `get_dna_distribution`: Counts the occurrences of each nucleotide base in the DNA sequence.
    - `transcribe_dna`: Transcribe a DNA sequence (this is the template sequence) into RNA.
2. ORF Detection:
    - `get_orf`: Identifies the longest ORF in an RNA sequence.
    - `get_corresponding_orf`: Extracts the ORF from the patient sequence that corresponds to the WT ORF.
3. Mutation Analysis:
    - `is_brac_mutation`: Determines if the patient sequence is similar to the wild-type sequence.
    - `identify_dna_mutation`: Identifies mutation in the patient DNA sequence.
    - `identify_aa_mutation`: Identifies mutations at the amino acid level.
4. Protein Translation:

    - `translate_rna`: Converts the RNA sequence into a protein sequence.
5. Visualization:
    - `visualize_dna_mutation`: Provides a visual comparison of DNA sequences.
    - `visualize_aa_mutation`: Generates a similar visual for protein sequences.

Upon completion, running the `main` function will automatically generate a detailed mutation report based on the comparison between WT and patient sequences, including visualizations of DNA and protein changes.

Suggested Workflow:

1. We already break down the procedure for you as it may be a relatively large project for your current stage.
2. For each function:
    a. Think about how they fit into the entire process
    b. Identify the required input and output
    c. You can add helper functions if needed
    d. Please note that you can submit the code to receive immediate feedback, and you can submit as many times as you want
    e. Please note that we drop the requirement for you to complete all the pre-condition check.
3. After finishing each function, please run the entire script which runs your scripts with the given WT and patient DNAs. Please note that those are real examples and your can use it to help you answer the three questions below in the rubrics. You can also create your own sequence to test things out.

Rubrics

You are allowed to reuse your code in previous labs. However, you are NOT allowed to copy code for labs from students outside of your team.

CLO3 participation (1 pt in total)

- (0.25 pt) Coding: visualize DNA mutation
- (0.25 pt) Coding: visualize RNA mutation
- (0.5 pt) Submit reports generated by your program

CLO5 evaluation (5 pts in total)

- Function implementation (3.5 pts in total)
    - read_dna_sequence_from_file: 0.25 pt

    - verify_dna_sequence: 0.25 pt
    - get_orf: 0.5 pt
    - get_dna_distribution: 0.25 pt
    - is_brac_mutation: 0.25 pt
    - transcribe_dna: 0.25 pt
    - translate_rna: 0.25 pt
    - get_corresponding_orf: 0.5 pt
    - identify_dna_mutation: 0.5 pt
    - identify_aa_mutation: 0.5 pt
- Answer questions
    1. (0.5 pt) When comparing the DNA sequence of a patient to a Wild-type sequence, one can compare the number of As, Cs, Ts, and Gs. Describe the biological limitation of this approach. Is this a feasible way of comparing the functionality of both sequences? In other words, if two sequences have the same # of As, Cs, Ts, and Gs, would that imply that their proteins are identical?

    2. (0.5 pt) Consider a situation where you are given a DNA sequence from a patient. Upon comparing the patient's DNA sequence to the Wildtype sequence, you detect one base pair change. However, when you compare the patient and wilt-type translated amino acid sequences, you do not detect any differences. What could be the reason for this?

    3. (0.5 pt) Consider a situation where you are given a DNA sequence from a patient. Upon comparing the patient's DNA sequence to the Wildtype sequence, you detect a very early stop codon. Describe the consequences of this mutation on the size and function of the protein produced in the patient.

CLO7 evaluation (1 pt in total)

- Teamwork

Submission

Please submit all documents to PrairieLearn:
https://ca.prairielearn.com/pl/course_instance/9139/assessment/60437

Please make sure you save the files for the questions that need to be manually graded!!!!!

Please follow the instructions on PriairLearn to submit your files (e.g., what to submit, file name, file type)