Team Project: Targeted Mutation Calculator for Specific Protein Production

Team Members: Lujain Khusheim, Rana Boustany, Zakiah Tcheifa, and Angela Abrego Chavez

Our project is based on recombinant protein expression. Recombinant protein expression in bacteria requires the insertion of a DNA fragment into an expression vector to utilize the natural protein production path for the production of a specific protein. We aim to design a program that can take in DNA sequences from many bacteria and amino acids and a desired protein and output the most suitable bacteria based on the least number of required mutations to produce the specific protein.

We designed this project because the creation of recombinant DNA for protein production can be a tedious process, and more specifically locating a proper vector to use for the production. After that, it can take a long time to identify the mutations in each possible vector and compare them. This program will be able to combine finding the proper vector and also the mutations needed to make that vector appropriate for protein production into one place for the user.

The algorithm is a free program and requires the user to have a python running platform and the ability to install python packages pandas and itertools. The user will be able to respond to the specific commands in the command line to obtain results. In the instructions, the user will be required to download and input the bacteria and amino acid databases as well as prepare a text file containing the target protein. The user will run the algorithm and results are displayed in little to no time.

In order to properly allow the code to run, if working with files other than those in the github, the user needs to manually add or remove elements in the product function (line 135).

The element indexes inside should go up to 1 less than the number of amino acids in your desired sequence. Simply copy one of the items in the input and change the index to a larger number to increase, or remove it to decrease.

The format of the input files are important to ensure that the code can function properly. The format of the desired protein file must be a .txt file, which contains the names of the amino acids. The names of the amino acids must be written in full, with the first letter(s) capitalizeded. If the amino acid has multiple words, the space in between is removed. Each amino acid needs to be in its own line, with their order start to end moves from top to bottom. Currently we are only supporting up to 17 amino acids using our bacterial database. The user can also choose to provide their own bacterial database to accommodate a larger number of amino acids. The format of the bacterial database needs to be a .csv file, where the first column has a header called "ID" and the second column has a header called "sequence". Underneath those two headers, the user can input their desired bacterial IDs and corresponding sequences using capital letters. Lastly, the user will also need to input the amino acid database, that contains each amino acid and their corresponding codons. This is not customizable and needs to be downloaded and used as-is from the Github repository.

The algorithm will then proceed to create an intermediate file (.csv) that contains the possible sequences for the desired protein sequence and will be used in the comparison calculation. The program uses predefined functions to locate the number of mismatches between each bacteria and sequence and output the bacteria names with the smallest number of mutations.

The algorithm will run and display and download the general report that shows the number of minimum mutations for each bacteria as well as the most suitable bacteria based on the desired protein sequence that was inputted. It will also give an option to download a more

detailed report which shows the protein sequence for each suitable bacteria and the location of the mutations.

We also included MATLAB code for the Graphical User Interface which is a more aesthetic way for the user to use our program, gives information about the program, and lets the user run the program to find the specific protein output.

In this project, Lujain and Rana worked on the python code. Lujain worked on prompting the user and the workflow through the code, as well as creating all the possible DNA sequences for the desired amino acid sequence. Rana worked on coding the functions that calculate the mutations and generate the output files. Lujain and Rana also created the powerpoint, and Zakiah and Angela collaborated on editing it. Zakiah worked on creating a Mock-Up GUI in MATLAB which we hope to implement into our python code in the future. Angela worked on the visualization part of our code which we hope to implement into our python code in the future as well. Angela and Lujain worked on the final report and the GitHub link.