Daniel Ramirez
03/21/23
RBIF-109 Spring 2023
Final Assignment

Project Scope

Enzyme activity engineering is a process in which a biological enzyme is modified to gain/lose certain functions. While there are many different strategies to evolve enzymes, some strategies include looking at nature and exploring multiple homologs of the same protein across various species. The variations within enzyme homologs can yield information on adaptations incurred over evolution that may improve/attribute certain characteristics [1]. With the availability of tools such as BLASTP by the National Center of Biotechnology [2] and multitude of synthetic DNA vendors, such sequences are readily available and attainable.

I work for a synthetic biology company called Twist Bioscience that sells synthetic DNA products such as oligo pools, genes and DNA libraries. In my department, I work in a research and development team that seeks to engineer proteins of interest. While we employ multiple different techniques and practices, one of the initial analyses is similar to the above mentioned strategy, and we test many different homologs of the same enzyme. We leverage our company's DNA synthesis technologies to synthesize genes, create different versions of enzymes, test their activity and categorize various homologs in a high-throughput manner. We then use the collected homolog data to direct our next rounds of enzyme evolution.

Naturally, the first part in this process is collecting the homologous sequences in the first place, which is usually a tedious process. This process typically involves running multiple BLAST searches then selecting a handful of different homologs across a wide range of percent similarity to the queried sequence. From there, sequences from a wide range of genera and species could be chosen and ordered in a 96-well plate format.

For my final project, I sought to automate this process by writing a tool that could query BLASTP through NCBI, gather homologs, and then bin them by percent similarity, and export 96 randomly chosen sequences into a ready-to-order csv.

Final Project : Homolog Finder

I. WorkFlow

This tool accepts a protein sequence either in the form of a string or FASTA file. Using the given protein sequence, a BLASTP search will be conducted using the NCBI BLASTP tool. The BLASTP search will be parsed for homologous sequences and all sequences will be binned by percent similarity. Analysis of the spread of percent similarity with respect to variations in species and genus will be calculated and demonstrated in generating histogram plots. From the collected homologs, 95 homologs from each bin will be randomly chosen. The query sequence will be added to be the 96th sequence and the resulting csv file will be returned.

II. Calculation of Percent Similarity and Binning

From the BLASTP record, sequences will be given with a value called “Identities”. This corresponds to the number of matching residues compared to the queried sequence. Percent similarity will be calculated as the number of matching values divided by the overall queried sequence length then multiplied by a 100 to get a percentage.

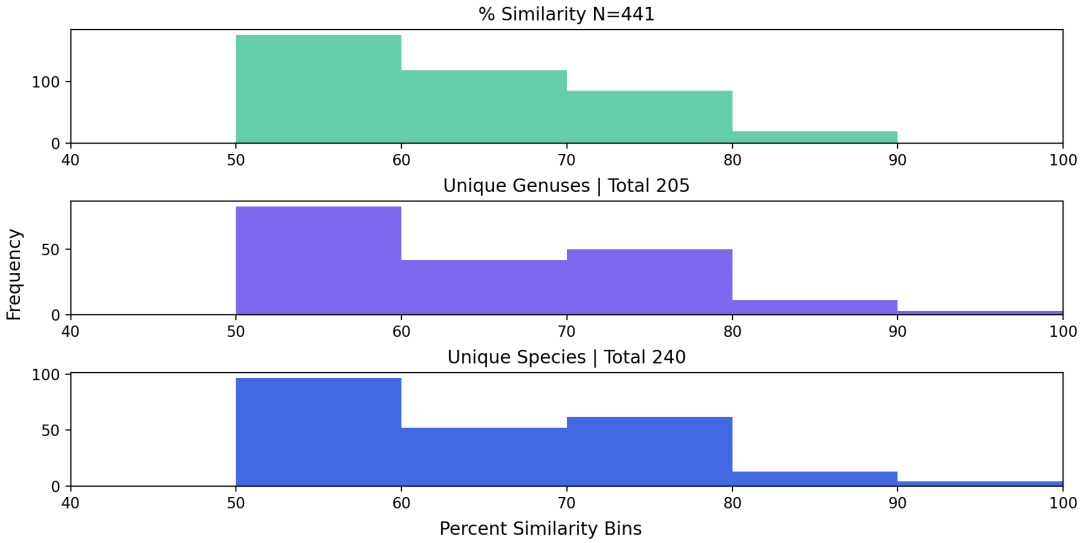
$$\%\, Similarity \;=\; (\#\ of\ identical\ residues\ /\ length\ of\ query\ sequence) \;*\; 100$$

After the percent similarity is calculated, it will be written as a row in the blastresults.csv as “%Similarity”. 5 bins of percent similarity will be created with each of the sequence homologs being added to each.

Bin 1	50% ≤ %Similarity < 60%
Bin 2	60% ≤ %Similarity < 70%
Bin 3	70% ≤ %Similarity < 80%
Bin 4	80% ≤ %Similarity < 90%
Bin 5	90% ≤ %Similarity < 100%

From each bin, 19 sequences will be chosen at random. If there are less than 19, then all sequences will be chosen.

III. Analysis of Plots



After the percent similarity of all sequences have been calculated, the top most plot will be generated detailing the spread of the different sequences across the 5 bins.The middle and bottom plots will show the number of unique genera and species respectively within each of the bins.

IV. Assumptions and Limitations

This program assumes that the amino acid sequence in question is found in nature with naturally found homologs that could be obtained with BLASTP. There is no filtering to avoid isoforms, or certain genera or species, however the blast_results.csv will be provided if the user seeks to add/remove homologs.

V. Program Requirements

Python 3.7 or higher
pandas 1.4 or higher
BioPython 1.7 or higher
Matplotlib 3.5 or higher

VI. Troubleshooting

If program initially terminates with error :

ValueError: What should we do with
CREATE_VIEW

Please ensure that Biopython 1.7 is installed then re-run the program a second time. On my machine, when running homolog_finder.py for the first time since starting my machine, I would receive this error. However, running a second time after the error cleared it.

Discussion

With this tool, I sought to streamline an initial but tedious process in my lab's enzyme engineering pipeline. It is designed to export a simple CSV for easy-ordering for DNA synthesis vendors (Twist Biosciences, IDT, GeneWiz). This tool is designed to be applicable to most analyses, however it is still the responsibility of the user to validate and ensure that the sequences fit the specific criteria for the screening.

I had initially intended to add additional sequence and domain annotation using hidden Markov models (HMM) [3], however it fell out of the scope of what could be completed by the deadline. If I am granted the appropriate resources at my workplace, I would like to set-up a server that could accommodate this functionality.

As I introduce this tool to the users of my lab, I plan to add additional functionality if ever requested.

README

Script for RBIF-109 Assignment 4 : Homolog Finder

Script will accept a peptide sequence either in the form of a string or FASTA file and conduct a BLASTP search to gather multiple sequence homologs. Homologs will be binned by percent similarity and 96 homologs across all bins will be chosen and exported to a ready-to-order csv.

Expected Usage:

python3 homolog_finder.py --fasta tdt.fasta

Expected Inputs (Only 1 required):

1) --fasta : path to a fasta file containing amino acid sequence
2) --aa_sequence : string of amino acid sequence

Expected Outputs:

1) 96-well-homologs.csv - CSV assortment of 96-homologs

2) blast_alignments.txt - TXT file containing the formatted alignments from the BLASTP query

3) blast_results.csv - CSV file containing the analyzed BLASTP records

4) blast_summary.png - PNG Plots describing the spread of percent similarity of homologs obtained

References

1. Donghyo Kim, Myung Hyun Noh, Minhyuk Park, Inhae Kim, Hyunsoo Ahn, Dae-yeol Ye, Gyoo Yeol Jung, Sanguk Kim, Enzyme activity engineering based on sequence co-evolution analysis, Metabolic Engineering, Volume 74, 2022, Pages 49-60, ISSN 1096-7176, https://doi.org/10.1016/j.ymben.2022.09.001.

2. Sayers EW, Bolton EE, Brister JR, Canese K, Chan J, Comeau DC, Connor R, Funk K, Kelly C, Kim S, Madej T, Marchler-Bauer A, Lanczycki C, Lathrop S, Lu Z, Thibaud-Nissen F, Murphy T, Phan L, Skripchenko Y, Tse T, Wang J, Williams R, Trawick BW, Pruitt KD, Sherry

ST. Database resources of the national center for biotechnology information. Nucleic Acids Res. 2022 Jan 7;50(D1):D20-D26. doi: 10.1093/nar/gkab1112. PMID: 34850941; PMCID: PMC8728269.

3. Yoon BJ. Hidden Markov Models and their Applications in Biological Sequence Analysis. Curr Genomics. 2009;10(6):402-415. doi:10.2174/138920209789177575