<u>Final Project Report</u>
<u>Project Title:</u> Examining Bacterial Genome using Python

Introduction (Goal and Problem Addressed):

The primary goal of this project is to develop a Python-based tool that can efficiently analyze bacterial genomes to identify potential open reading frames (ORFs), translate these ORFs into protein sequences, calculate their molecular mass, and perform BLAST analysis. This tool aims to streamline the process of gene identification and characterization, by contributing to our understanding of bacterial genetics.

The sequence used in this project is the complete synthetic genome of *Mycoplasma mycoides JCVI-syn1.0*, which is a milestone project by the Craig Venter Institute. This work aimed to construct a minimized version of *Mycoplasma mycoides* bacterium's genome, focussing on essential life-sustaining genes. Studying such genomes is crucial for understanding the minimal genetic requirements of life, paving the way for advancements in biotechnology and medicine. Performing BLAST on such genomes is crucial for understanding the biological functions, evolutionary context, and safety of the synthetic constructs, bridging the gap between theoretical design and practical application, and ensuring that synthetic biology advances in a safe, ethical, and scientifically sound manner.

Approach:

1. Scanning both strands of a bacterial genome to find ORFs.
2. Translating the identified ORFs into their corresponding protein sequences.
3. Calculating the predicted molecular mass of each translated protein sequence.
4. Performing BLAST analysis for up to 5 protein sequences against the NCBI database.

Tools Used:

1. ***Python***
2. ***Installing necessary Python libraries:***
   (This will enable working with the genomic data, performing analysis for finding ORFs (open reading frame), translating nucleotide sequences to protein sequences, and interacting with the NCBI BLAST).
   - <u>biopython</u> - Collection of Python tools for computational biology and bioinformatics, it helps in parsing bioinformatics files into Python structures, and also performs several bioinformatics tasks, like finding ORF, sequence alignment, and interacting with NCBI.
   - <u>requests</u> - HTTP library for Python, which makes sending HTTP/1.1 requests easy.
3. ***Importing packages from biopython:***
   - <u>SeqIO</u> - For parsing bacterial genome sequence files.
   - <u>Seq</u> - For handling sequences extracted from the files for further analysis.
   - <u>SeqRecord</u> - To keep track of the metadata for each sequence.
   - <u>SeqUtils</u> - To calculate the predicted molecular mass of each protein.
   - <u>SeqFeature</u> - For handling sequence features.
   - <u>SearchIO</u> - For paring BLAST results while analyzing protein sequences against the NCBI database.
   - <u>NCBIWWW</u> - For submitting BLAST queries to the NCBI BLAST server.
   - <u>NCBIXML</u> - For parsing the XML results returned by BLAST searches.
4. ***Sequence used:***
   - CP002027.1 (Synthetic Mycoplasma mycoides JCVI-syn1.0 clone sMmYCp235-1, complete sequence)

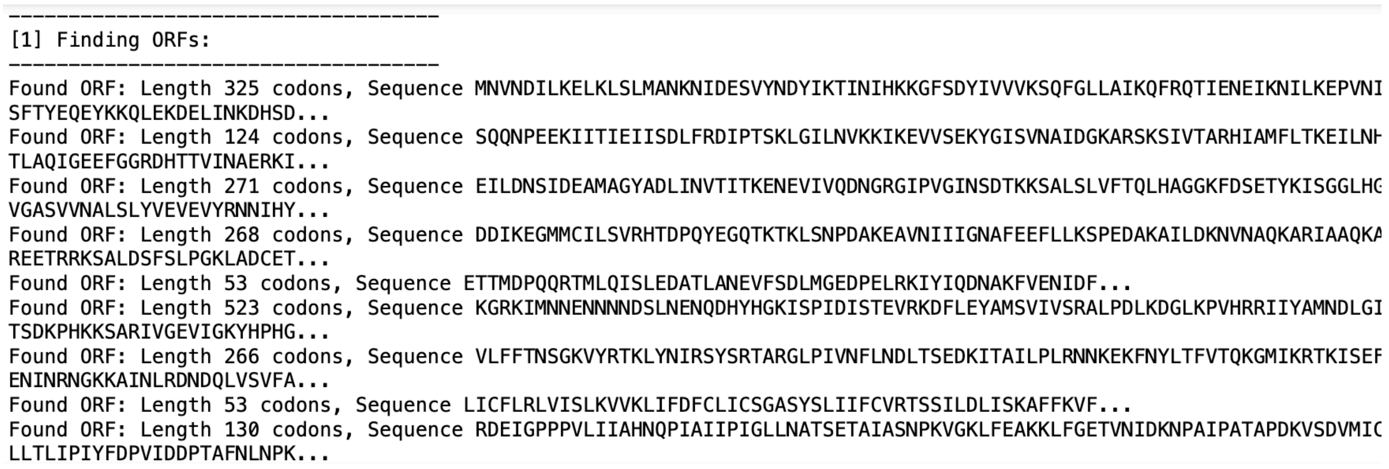
Code Description:

1. ***Reading and preparing Genomic Data* - def read_genome(filename):**
    - This function takes a single parameter called 'filename', in the form of a string representing the path to a FASTA file containing genomic sequences.
    - The function is a utility for loading genomic data from FASTA files, using Biopython for parsing.
    - It simplifies the process of reading and handling genomic sequences, making it a fundamental step in bioinformatics workflow that involves sequence analysis.
2. ***Scanning for ORFs and Translation ORFs to Protein Sequences* - def find_orfs(sequence, min_protein_length=50):**
    - This function takes two parameters, sequence, which takes in the DNA sequence as a 'Seq' object from Biopython, in which ORFs are to be found, and min_protein_length, which specifies the minimum length of the protein for an ORF to be considered significant, it defaults to 50 amino acids, if not specified otherwise.
    - This function identifies ORFs across both strands of DNA sequence in all possible reading frames, based on a specified minimum protein length.
    - By examining both the forward sequence and its reverse compliment, and considering all three reading frames, the function ensures that no potential ORF is overlooked.

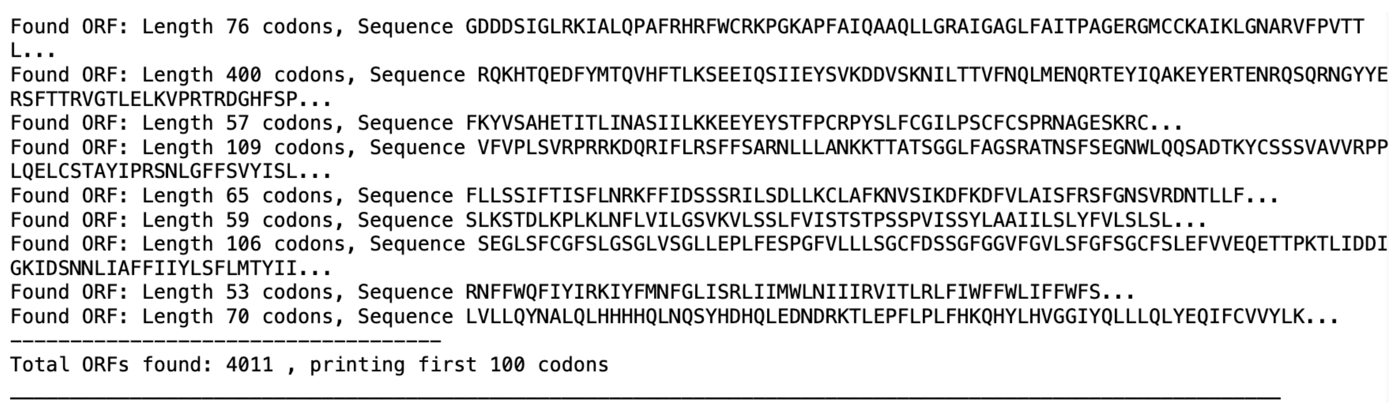
5. ***Calculating Molecular Mass* - def calculate_molecular_mass(proteins):**
    - A protein's molecular mass is an important biochemical property that provides insights into protein structure, function, and interactions within the cell.
    - This function accepts one parameter, 'proteins', which is expected to be a list of protein sequences.
    - The function returns a list of molecular mass (in kDa) corresponding to each protein sequence provided.
6. ***Selecting protein sequences for BLAST analysis* - def filter_sequences_for_blast(proteins, max_length=1000):**
    - This function takes a list of protein sequences for BLAST analysis by filtering out any sequences that exceed a specified maximum length, which is important for ensuring that the BLAST search is manageable and compiles with the limitations of the BLAST database or to optimize the search for performance reasons.
    - Providing detailed logging also helps in quickly understanding the makeup of the sequences moving forward in the analysis pipeline.
7. ***Performing BLAST analysis* - def perform_blast(protein_sequences):**
    - This function is for executing a BLASTp search against the NCBI's non-redundant (nr) database for a list of protein sequences.
    - It returns the top 5 hits for each sequence, providing insights into the possible function, structure, or evolutionary relationships for the proteins under investigation.
    - In this step, we will use Biopython's NCBIWWW module to submit selected protein sequences for BLAST analysis and parse the results to extract information about most similar hits.
8. ***Compilation of the results* - def save_results_to_csv(results, filename= "results.csv"):**
    - This step is for organizing all the necessary data into a CSV file, which includes ORF sequence, translated protein sequence, molecular mass of the protein, and BLAST results (most similar hits, accession numbers, match identity, and e-values).
    - A CSV format ensures that the result can be easily accessed and manipulated with a wide range of tools.

Results:

1. ***Finding ORFs:***
    - The program found a total of 4011 ORFs and translated them into proteins.

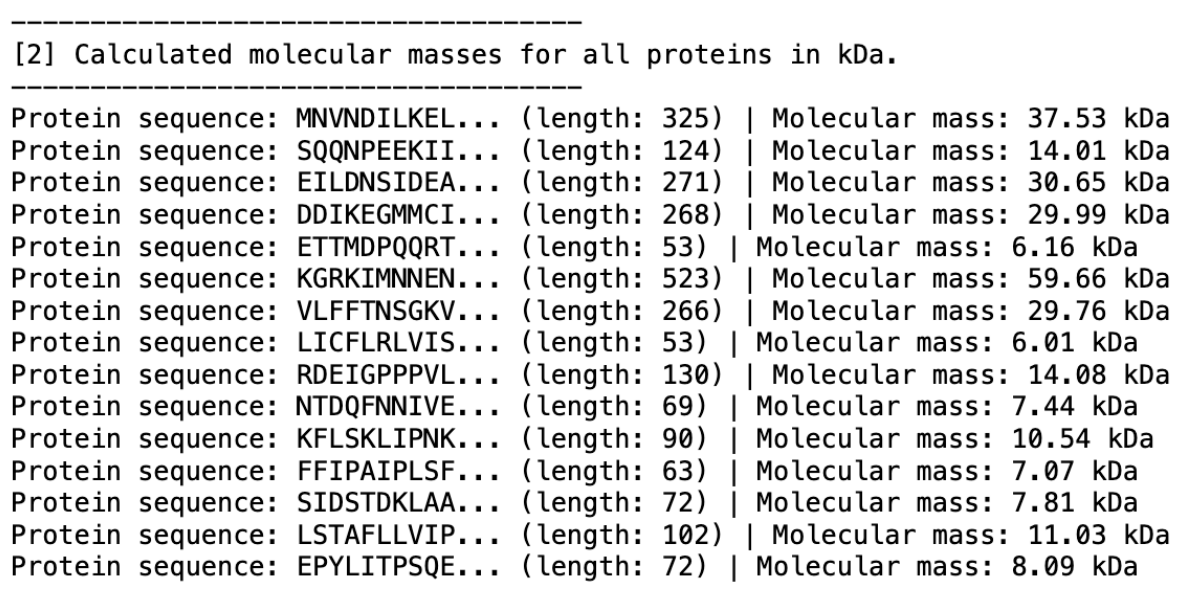


```
-----------------------------------
[1] Finding ORFs:
-----------------------------------
Found ORF: Length 325 codons, Sequence MNVNDILKELKLSLMANKNIDESVYNDYIKTINIHKKGFSDYIVVVKSQFGLLAIKQFRQTIENEIKNILKEPVNI
SFTYEQEYKKQLEKDELINKDHSD...
Found ORF: Length 124 codons, Sequence SQQNPEEKIITIEIISDLFRDIPTSKLGILNVKKIKEVVSEKYGISVNAIDGKARSKSIVTARHIAMFLTKEILNH
TLAQIGEEFGGRDHTTVINAERKI...
Found ORF: Length 271 codons, Sequence EILDNSIDEAMAGYADLINVTITKENEVIVQDNGRGIPVGINSDTKKSALSLVFTQLHAGGKFDSETYKISGGLHG
VGASVVNALSLYVEVEVYRNNIHY...
Found ORF: Length 268 codons, Sequence DDIKEGMMCILSVRHTDPQYEGQTKTKLSNPDAKEAVNIIIGNAFEEFLLKSPEDAKAILDKNVNAQKARIAAQKA
REETRRKSALDSFSLPGKLADCET...
Found ORF: Length 53 codons, Sequence ETTMDPQQRTMLQISLEDATLANEVFSDLMGEDPELRKIYIQDNAKFVENIDF...
Found ORF: Length 523 codons, Sequence KGRKIMNNENNNNDSLNENQDHYHGKISPIDISTEVRKDFLEYAMSVIVSRALPDLKDGLKPVHRRIIYAMNDLGI
TSDKPHKKSARIVGEVIGKYHPHG...
Found ORF: Length 266 codons, Sequence VLFFTNSGKVYRTKLYNIRSYSRTARGLPIVNFLNDLTSEDKITAILPLRNNKEKFNYLTFVTQKGMIKRTKISEF
ENINRNGKKAINLRDNDQLVSVFA...
Found ORF: Length 53 codons, Sequence LICFLRLVISLKVVKLIFDFCLICSGASYSLIIFCVRTSSILDLISKAFFKVF...
Found ORF: Length 130 codons, Sequence RDEIGPPPVLIIAHNQPIAIIPIGLLNATSETAIASNPKVGKLFEAKKLFGETVNIDKNPAIPATAPDKVSDVMIC
LLTLIPIYFDPVIDDPTAFNLNPK...
```



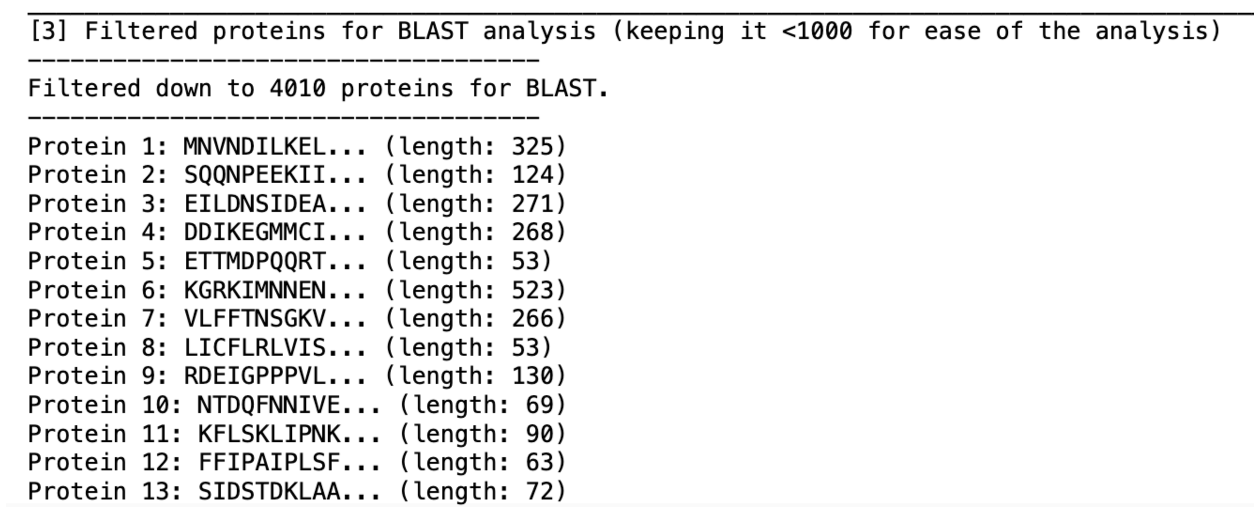
```
Found ORF: Length 76 codons, Sequence GDDDSIGLRKIALQPAFRHRFWCRKPGKAPFAIQAAQLLGRAIGAGLFAITPAGERGMCCKAIKLGNARVFPVTT
L...
Found ORF: Length 400 codons, Sequence RQKHTQEDFYMTQVHFTLKSEEIQSIIEYSVKDDVSKNILTTVFNQLMENQRTEYIQAKEYERTENRQSQRNGYYE
RSFTTRVGTLELKVPRTRDGHFSP...
Found ORF: Length 57 codons, Sequence FKYVSAHETITLINASIILKKEEYEYSTFPCRPYSLFCGILPSCFCSPRNAGESKRC...
Found ORF: Length 109 codons, Sequence VFVPLSVRPRRKDQRIFLRSFFSARNLLLANKKTTATSGGLFAGSRATNSFSEGNWLQQSADTKYCSSSVAVVRPP
LQELCSTAYIPRSNLGFFSVYISL...
Found ORF: Length 65 codons, Sequence FLLSSIFTISFLNRKFFIDSSSRILSDLLKCLAFKNVSIKDFKDFVLAISFRSFGNSVRDNTLLF...
Found ORF: Length 59 codons, Sequence SLKSTDLKPLKLNFLVILGSVKVLSSLFVISTSTPSSPVISSYLAAIILSLYFVLSLSL...
Found ORF: Length 106 codons, Sequence SEGLSFCGFSLGSGLVSGLLEPLFESPGFVLLLSGCFDSSGFGGVFGVLSFGFSGCFSLEFVVEQETTPKTLIDDI
GKIDSNNLIAFFIIYLSFLMTYII...
Found ORF: Length 53 codons, Sequence RNFFWQFIYIRKIYFMNFGLISRLIIMWLNIIIRVITLRLFIWFFWLIFFWFS...
Found ORF: Length 70 codons, Sequence LVLLQYNALQLHHHHQLNQSYHDHQLEDNDRKTLEPFLPLFHKQHYLHVGGIYQLLLQLYEQIFCVVYLK...
-----------------------------------
Total ORFs found: 4011 , printing first 100 codons
```

2. ***Calculate the molecular mass in kDa:***
    - The molecular mass of each protein was calculated in kDa and printed.



```
-----------------------------------
[2] Calculated molecular masses for all proteins in kDa.
-----------------------------------
Protein sequence: MNVNDILKEL... (length: 325) | Molecular mass: 37.53 kDa
Protein sequence: SQQNPEEKII... (length: 124) | Molecular mass: 14.01 kDa
Protein sequence: EILDNSIDEA... (length: 271) | Molecular mass: 30.65 kDa
Protein sequence: DDIKEGMMCI... (length: 268) | Molecular mass: 29.99 kDa
Protein sequence: ETTMDPQQRT... (length: 53) | Molecular mass: 6.16 kDa
Protein sequence: KGRKIMNNEN... (length: 523) | Molecular mass: 59.66 kDa
Protein sequence: VLFFTNSGKV... (length: 266) | Molecular mass: 29.76 kDa
Protein sequence: LICFLRLVIS... (length: 53) | Molecular mass: 6.01 kDa
Protein sequence: RDEIGPPPVL... (length: 130) | Molecular mass: 14.08 kDa
Protein sequence: NTDQFNNIVE... (length: 69) | Molecular mass: 7.44 kDa
Protein sequence: KFLSKLIPNK... (length: 90) | Molecular mass: 10.54 kDa
Protein sequence: FFIPAIPLSF... (length: 63) | Molecular mass: 7.07 kDa
Protein sequence: SIDSTDKLAA... (length: 72) | Molecular mass: 7.81 kDa
Protein sequence: LSTAFLLVIP... (length: 102) | Molecular mass: 11.03 kDa
Protein sequence: EPYLITPSQE... (length: 72) | Molecular mass: 8.09 kDa
```

3. ***Filtering down proteins for BLAST:***
    - Filtered down to 4010 proteins for BLAST.



```
[3] Filtered proteins for BLAST analysis (keeping it <1000 for ease of the analysis)
-----------------------------------
Filtered down to 4010 proteins for BLAST.
-----------------------------------
Protein 1: MNVNDILKEL... (length: 325)
Protein 2: SQQNPEEKII... (length: 124)
Protein 3: EILDNSIDEA... (length: 271)
Protein 4: DDIKEGMMCI... (length: 268)
Protein 5: ETTMDPQQRT... (length: 53)
Protein 6: KGRKIMNNEN... (length: 523)
Protein 7: VLFFTNSGKV... (length: 266)
Protein 8: LICFLRLVIS... (length: 53)
Protein 9: RDEIGPPPVL... (length: 130)
Protein 10: NTDQFNNIVE... (length: 69)
Protein 11: KFLSKLIPNK... (length: 90)
Protein 12: FFIPAIPLSF... (length: 63)
Protein 13: SIDSTDKLAA... (length: 72)
```

4. ***CSV file:***
    - The BLAST results are saved in a .csv format for ease of review.

Discussion and Conclusion:

The project successfully identified several Open Reading Frames(ORFs) within the given genome sequence, calculated their molecular masses, performed BLAST analysis against the 'nr' database, and documented the findings.

The identification of the ORFs is a crucial step in genomics, as some of them are protein-coding sequences. Calculating molecular mass further characterizes these sequences, providing insights into the physical properties of the potential proteins.

The BLAST analysis provided crucial links to known proteins in existing databases, revealing potential functions or evolutionary relationships for some of the ORFs. This connection between novel and known sequences is fundamental in genomics and bioinformatics, as it can lead to discoveries about mechanisms, and gene functions.

Future Prospects:

1. By identifying ORFs and analyzing sequences, this project can contribute to functional genomics studies, helping in the understanding of gene function and regulation.
2. The BLAST analysis component can help in predicting the function of newly discovered proteins based on similarity to known proteins, aiding in the annotation of genomes.
3. The techniques used in this project can be applied to identify genes associated with diseases, especially in genomes that have not been fully explored.

Future Improvements:

1. Improving the automation of the analysis pipeline and its scalability to handle large datasets efficiently.
2. Implementing more sophisticated algorithms for ORF prediction and functional annotation to improve accuracy and reduce false positives.
3. Expanding the project to include integration with other genomic databases and analysis tools to enhance the depth and breadth of the analysis.

Citations:

1. Stefano Allesina, Wilmes M. Computing Skills for Biologists. Princeton University Press; 2019.
2. OpenAI. ChatGPT [Internet]. San Francisco: OpenAI; [March 8, Friday]. Available from: https://openai.com/chatgpt/
3. Synthetic Mycoplasma mycoides JCVI-syn1.0 clone sMmYCp235-1, complete sequence. NCBI Nucleotide [Internet]. 2010 Sep 29 [cited 2024 Mar 8]; Available from: https://www.ncbi.nlm.nih.gov/nuccore/CP002027.1?report=fasta