Antibiotic Resistance Gene Prediction (Deep Learning)

Antibiotic resistance is the process exhibited by bacterial species (mainly); this allows the bacteria to grow even in the presence of an antibiotic exhibiting resistance and no sensitivity to the antibiotic. Such resistances are incorporated due to the production of specific resistance mechanisms or proteins which either cleave the antibiotic (when enters the bacteria) or inactivates it by destroying the Beta-Lactam rings (penicillin-based antibiotics).

The prediction of antibiotic resistance exhibited by a particular microbe is extremely crucial for healthcare and medicine as it would ensure appropriate dosage and incorporation of antibiotic (to which microbe is not resistant) at the early cycles of a particular infection.

In this project CNN (Convolution Neural Network) based learning and prediction models would be employed to predict AMR (Anti-Microbial Resistance). Appropriate AMR +ve and AMR -ve dataset would be used to train, test and validate the models.

The aim of this project is to make a robust prediction model which distinguish ARG (Antibiotic Resistance Gene) and Non-ARG from a data which is unknown or novel.

1. **Preparing ARG (Antibiotic Resistance Gene) and Non ARG sequences**

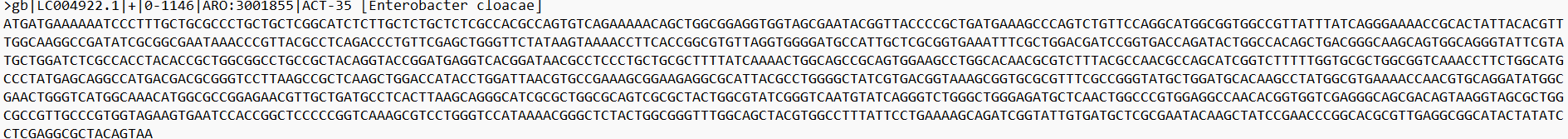
Positive and Negative samples are needed for a deep learning model like CNN so, ARG would serve as positive samples and Non ARG sequences as negative ones.

1. CARD data has been downloaded from: <https://card.mcmaster.ca/latest/data>

This contains several fasta files which give nucleotide and protein sequences using various methods of analysis (like overexpression, variant etc.) but for this project a particular file titled: “nucleotide\_fasta\_protein\_homolog\_model.fasta” serves as a positive sample data.

This file encompasses various SampleIDs (wherein each SampleID denotes a new sample) with nitrogen bases sequences. These nitrogeneous base sequences when transcribed would form RNA sequences which when further grouped by codon (3 RNA sequences = 1Amino Acid) mechanism would form the protein. This protein is responsible for the generation of antibiotic resistance in that particular organism. The data file also contains the Name of the organism in which this particular sequence has been found.

Protein homology model has been used to predict if the sequence would create AMR or not meaning as stated earlier that the 3AA would form 1 protein which would be responsible for antibiotic resistance, this protein (target) is compared to another protein (template) which is homologous to itself. If both the proteins achieve higher levels of matching scores then the target protein is considered as the one which induces the antibiotic resistance.



The image above depicts one of the samples from the fasta file.

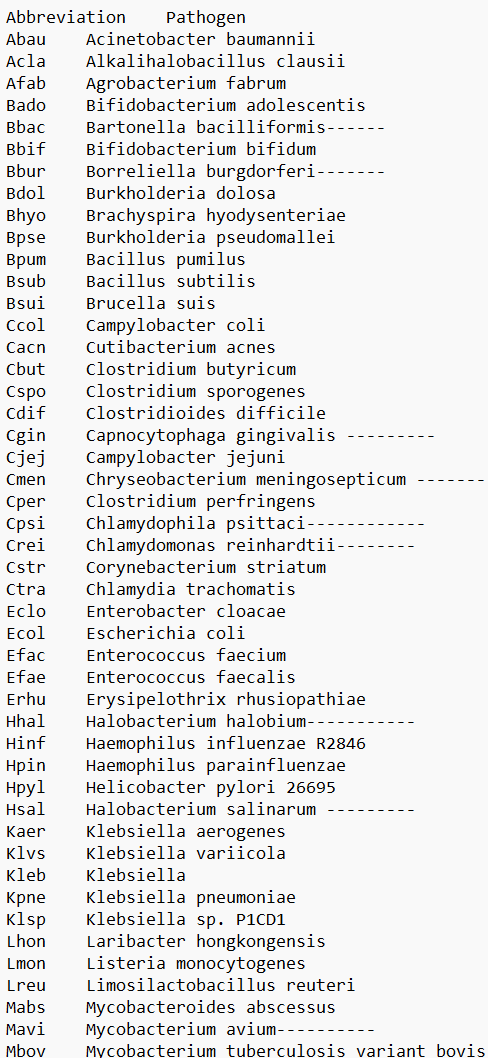
gb|LC004922.1|+|0-1146|ARO:3001855|ACT-35 is the ID

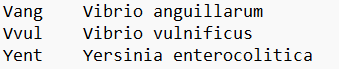
*Enterobacter Cloacae* is the organism in which these sequences have been found to induce antimicrobial resistance.

1. Code to prepare Positive ARG (Antibiotic Resistance Genes):
2. *# Collecting ARG Sequences (Positive Samples)*
3. import **os**
4. from Bio import SeqIO
5. import **numpy** as **np**
6. def **load\_fasta\_sequences**(filepath):
7. sequences = []
8. ids = []
9. **print**(f"Loading sequences from: {filepath}")
10. try:
11. for record in SeqIO.parse(filepath, "fasta"):
12. sequences.**append**(**str**(record.seq).**upper**()) *#All sequences in uppercase*
13. ids.**append**(record.id)
14. **print**(f"Successfully loaded {**len**(sequences)} sequences.")
15. except **FileNotFoundError**:
16. **print**(f"Error: File not found at {filepath}.")
17. except **Exception** as e:
18. **print**(f"An error occurred while parsing {filepath}: {e}")
19. return ids, sequences
20. *#File path*
21. arg\_fasta\_path = 'C:/Users/mohak/Desktop/card-data/nucleotide\_fasta\_protein\_homolog\_model.fasta'
22. arg\_ids, arg\_sequences = **load\_fasta\_sequences**(arg\_fasta\_path)
23. if not arg\_sequences:
24. **print**("No ARG sequences loaded.")
25. avg\_arg\_len = 1000
26. std\_arg\_len = 500
27. else:
28. **print**(f"First 3 ARG IDs: {arg\_ids[:3]}")
29. **print**(f"Length of first ARG sequence: {**len**(arg\_sequences[0])}")
30. **print**(f"First 50 bases of first ARG sequence: {arg\_sequences[0][:50]}...")
31. arg\_lengths = [**len**(s) for s in arg\_sequences]
32. avg\_arg\_len = **np**.**mean**(arg\_lengths)
33. std\_arg\_len = **np**.**std**(arg\_lengths)
34. **print**(f"Approximate Average ARG Length: {avg\_arg\_len:.2f} bp")
35. **print**(f"Approximate Std Dev ARG Length: {std\_arg\_len:.2f} bp")

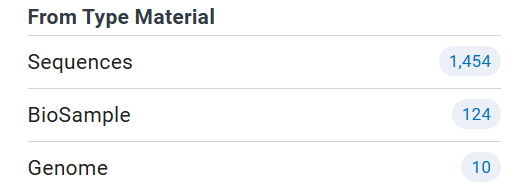
B. Non-ARG:

We have harvested positive samples but we also need negative samples which would help improve the overall robustness of the model. Now, a list of organisms has been given by the card-data file and these are given below:

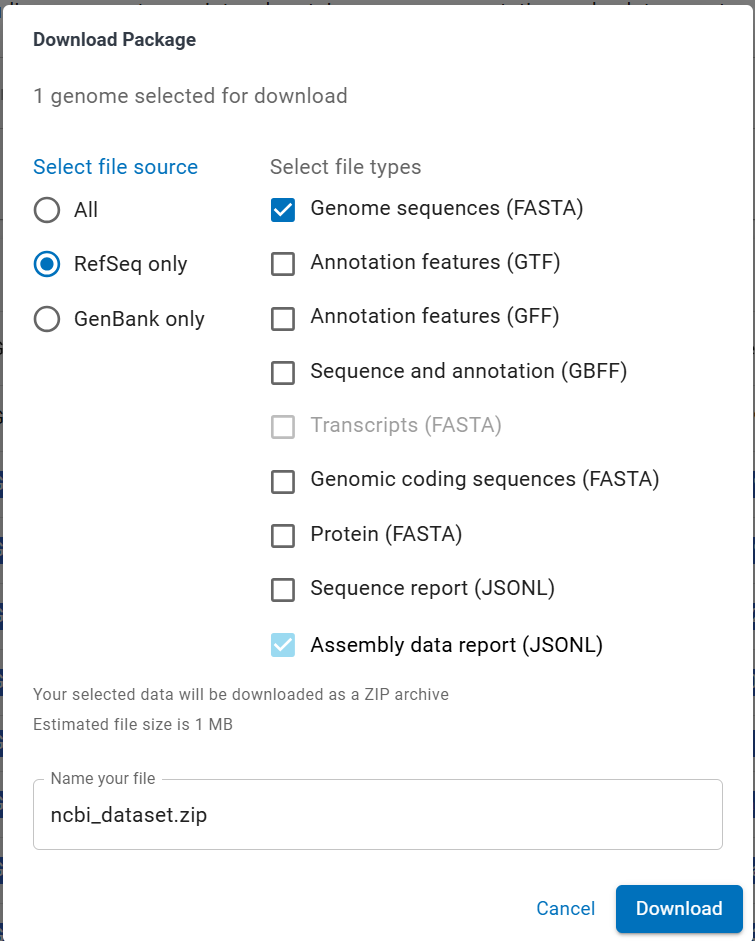




The genomic data for all of the mentioned organisms have been downloaded from <https://www.ncbi.nlm.nih.gov/datasets/>

* Consider the first one: *Acinetobacter baumannii*
* Steps to download the genomic data:
* Search *Acinetobacter baumannii* in the search box.
* 

Navigate to “Genome”

* 
* A green tick dataset has been downloaded. This particular tick denotes that extreme measures have been taken to ensure that the sequence is correct and also the full sequence for that particular strain might be recorded.
* Click on the  icon and “Download” option.
* 
* All the genomic data for the succeeding organisms must be downloaded using the same process.