Predicting Bacterial Pathogeny

Random Forest with k-mers and Chaos Game Representation Features

Zeyad Ahmed

Introduction

Overview

- In this project, we aim to predict the pathogenicity of bacterial species using machine learning techniques applied to DNA sequences.
- Pathogenic bacteria are microorganisms that have the capability to cause disease in humans, animals, or plants.

Introduction

Importance

- Accurate prediction of bacterial pathogenicity is crucial for various fields including medicine, agriculture, and environmental science.

Introduction

Methods

- We employ the Random Forest algorithm along with k-mers and frequency Chaos Game Representation (fCGR) as feature extraction techniques for DNA sequence analysis.

Data Wrangling

Scraping Process

- Bacteria species names were scraped from Wikipedia and PubMed.
- Pathogenic bacteria species labels were scraped from Wikipedia.

- Libraries used:
  - BeautifulSoup

Data Wrangling

Data Collection

- We instantiated a bot that collects NCBI accession ID's of the respective bacterial species from GTDB.
- Only GTDB representative sequences with < 5% contamination and > 95% completeness were considered.
- We downloaded the sequences using command-line scripts.

- Libraries used:
  - Selenium
  - Curl

Data Engineering

Data Cleaning

- We cleaned the sequences by removing N’s and converted DNA sequences into appropriate formats for further analysis.

Data Engineering

Data Preprocessing

- For each species, we created a representative fasta file which is ~500,000 bp long.
- This was done by either sampling a portion from the longest contig, or by concatenating shorter contigs until reaching the desired length.

Data Engineering

K-mers

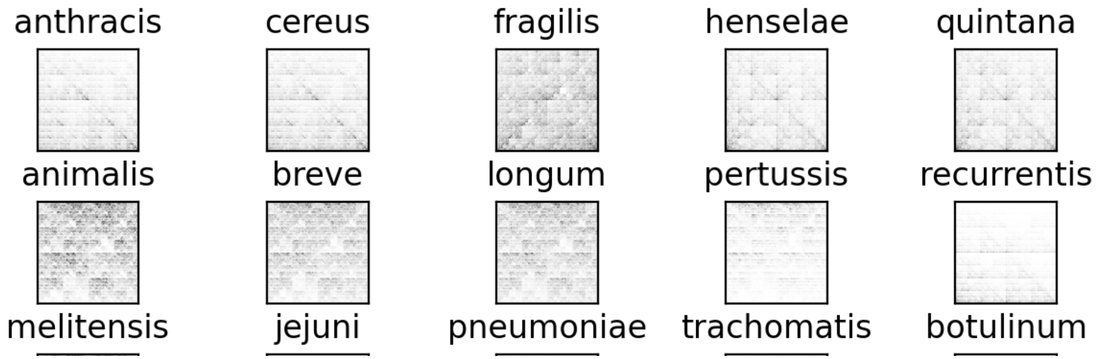
- K-mers are subsequences of length k extracted from DNA sequences, providing insights into genetic patterns.
- Bacterial DNA sequences were preprocessed to extract k-mers (k=7) as features for classification.

Exploratory Data Analysis

CGR plots

- We generated CGR plots to explore patterns in each species.
- We found that species within a genus share very similar CGR patterns.
- Below is a portion of the CGR plots:

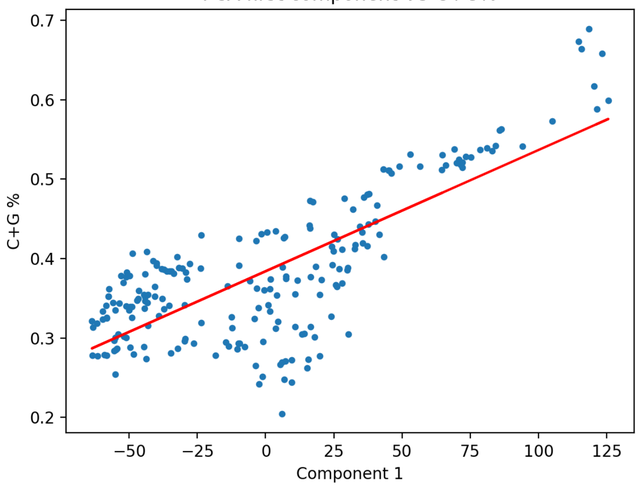
- Libraries used:
  - NumPy
  - Matplotlib

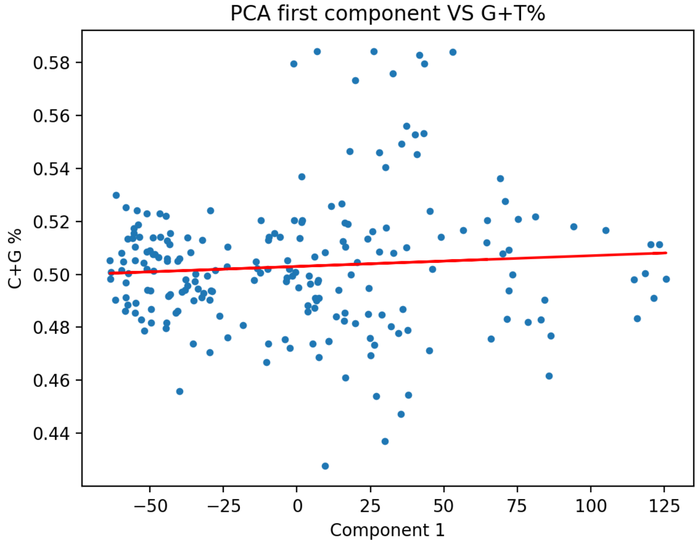


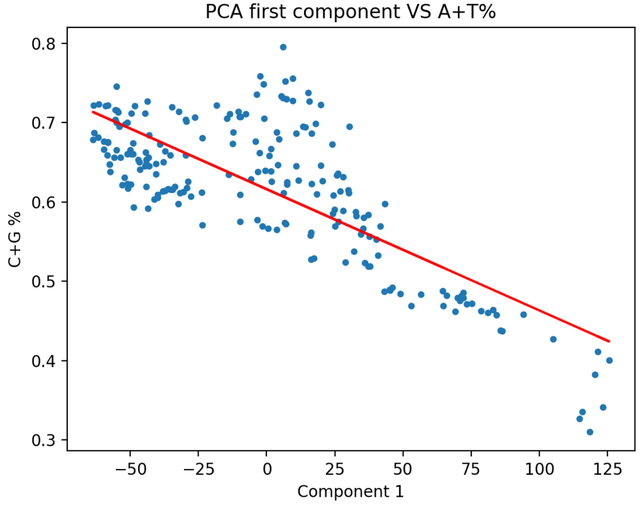
Exploratory Data Analysis

PCA

- We applied Principal Component Analysis (PCA) to explore interesting findings in literature.



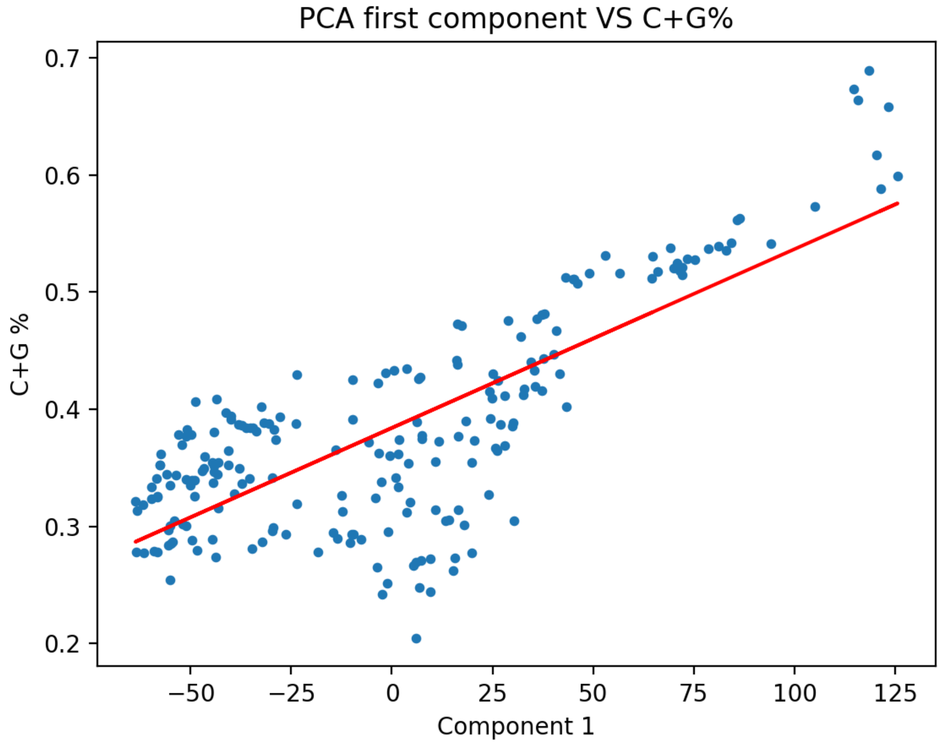




Exploratory Data Analysis

C+G%

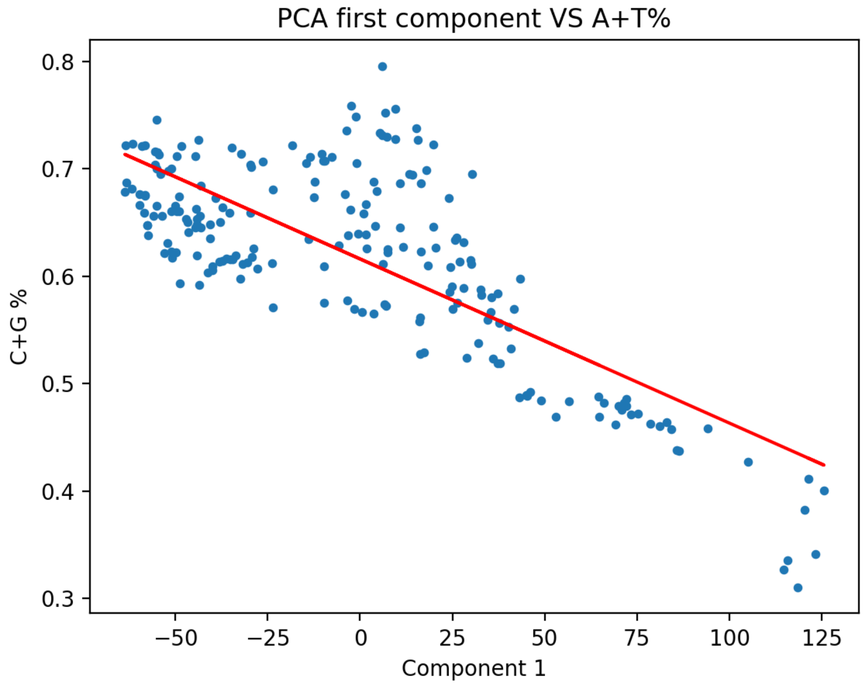
- We found that there was a correlation between the first component of the PCA and the C+G% content.



Exploratory Data Analysis

A+T%

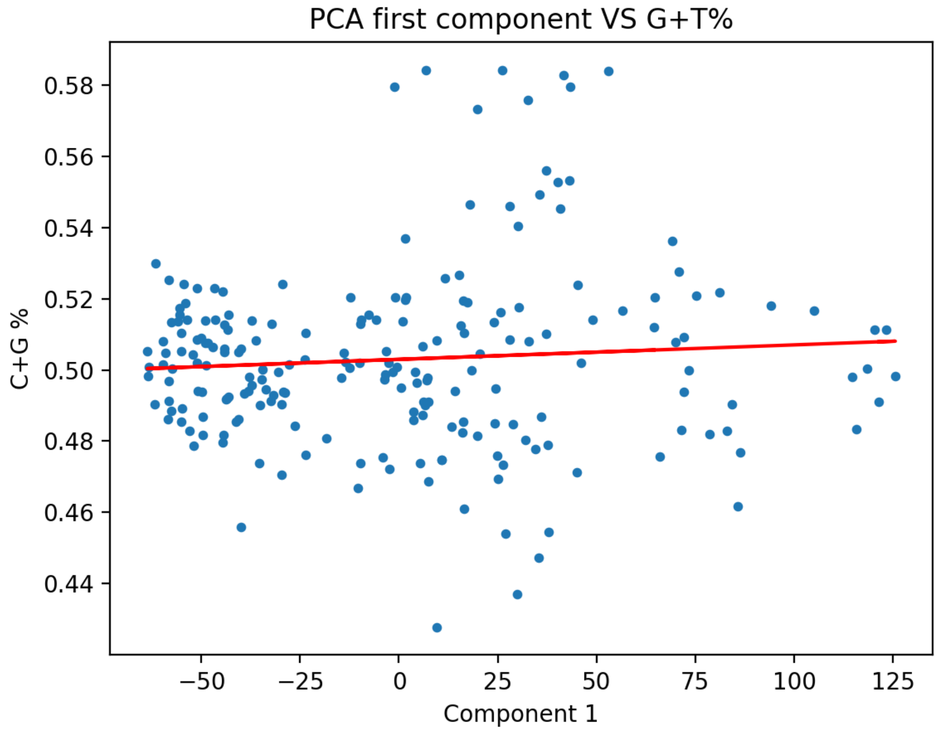
- Similarly, we found that there was a negative correlation between the first component of the PCA and the A+T% content.



Exploratory Data Analysis

G+T%

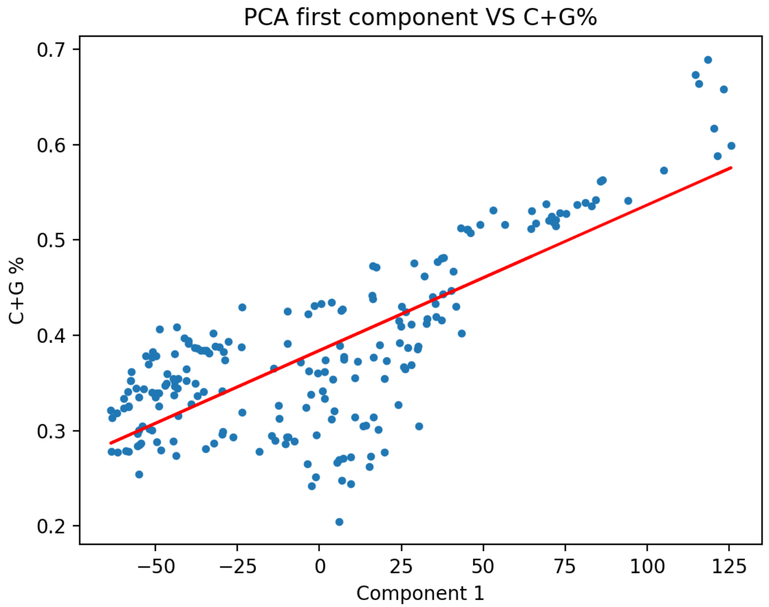
- On the contrary, there was no correlation between the other combinations of letters such as G+T.

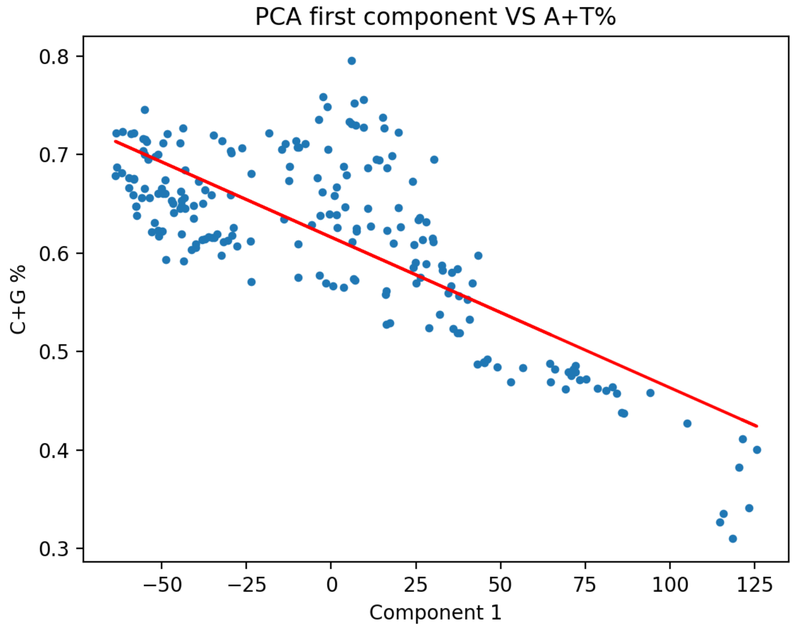


Exploratory Data Analysis

Further Questions

- What causes the C+G% plot and the A+T% plot to appear as mirror images of each other?
- Reverse complement (A—>T and C—>G) ?



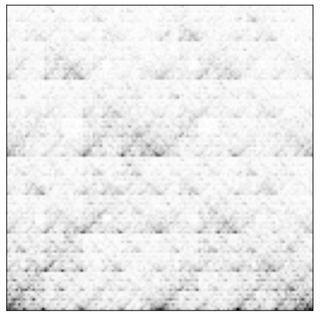


Machine Learning Model

Overview

- We employed the Random Forest Classifier for binary classification.
- Generated fCGR plots (images) were used as feature vectors.

- Libraries Used:
  - Scikit-learn
  - OpenCV



fCGR

```
array([[255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       [253, 253, 253, ..., 255, 254, 253],
       ...,
       [105, 105, 101, ...,  65,  75,  81],
       [ 96,  96,  92, ...,  36,  53,  63],
       [ 96,  96,  92, ...,  36,  53,  63]], dtype=uint8)
```

Matrix representation

```
array([[1.        , 1.        , 1.        , ..., 1.        , 1.        ,
        1.        ],
       [1.        , 1.        , 1.        , ..., 1.        , 1.        ,
        1.        ],
       [0.99215686, 0.99215686, 0.99215686, ..., 1.        , 0.99607843,
        0.99215686],
       ...,
       [0.41176471, 0.41176471, 0.39607843, ..., 0.25490196, 0.29411765,
        0.31764706],
       [0.37647059, 0.37647059, 0.36078431, ..., 0.14117647, 0.20784314,
        0.24705882],
       [0.37647059, 0.37647059, 0.36078431, ..., 0.14117647, 0.20784314,
        0.24705882]])
```

Normalized matrix

Machine Learning Model

10 Folds

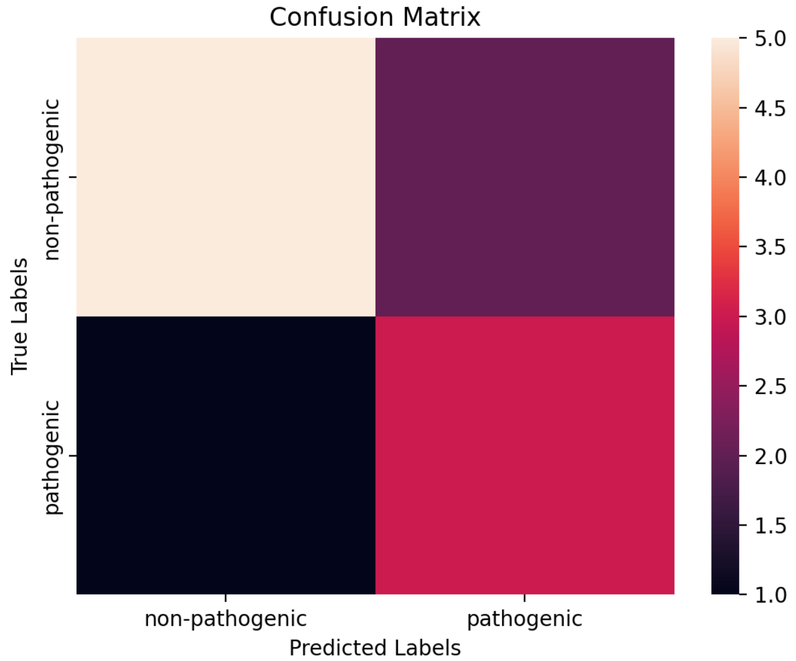
- We can see that the model accuracy ranges between 75% - 100%.
- The mean accuracy is 88%
- Further questions:
  - Why is the accuracy **range** so high?

Fold 1	75%
Fold 2	83%
Fold 3	92%
Fold 4	92%
Fold 5	100%
Fold 6	83%
Fold 7	92%
Fold 8	92%
Fold 9	83%
Fold 10	92%

Machine Learning Model

Training

- After evaluation, we split the model into Training (80%) and Testing (20%) datasets.
- The model correctly identified pathogenic bacteria 88% of the time.



Conclusion

Summary

- Our model predicted the pathogeny of bacteria using its fCGR with ~88% accuracy.
- This may suggest that pathogeny of bacteria may be associated with their genomic fingerprint — fCGR in this project.

Limitations

Dataset

- In this project, the size of the pathogenic dataset is roughly three times larger than the non-pathogenic dataset, which might introduce bias in the model.
- The model is trained on a subset of bacteria, including more genera could increase the reliability of the model.
- There is no agreement among scientists on the pathogeny of some species included in the dataset.

Future Work

Improvement

- Using a larger, more representative dataset to verify the model results.
- Exploring other significant features, such as k-mer variability within each pathogeny class.
- Exploring other numerical methods, such as Discrete Fourier Transform.