Predicting Gene Function and Mutation Impact Using GenBank Data

December 2025

By Praneel P., Angris K., Hakim A.

I. Defining the Project

What problem are we solving?

This project focuses on addressing the computational challenge of predicting gene function categories and how specific mutations might affect them using sequence-derived features extracted from GenBank data. We developed a machine learning pipeline that classifies genes into fifteen distinct functional categories based on their nucleotide and protein sequence characteristics. We also simulated point mutations and used our models to estimate how these changes might impact protein structure, using both classification and regression methods.

The problem is fundamentally a supervised learning task operating on high-dimensional biological sequence data. The gene function task is a multi-class classification problem with fifteen target classes (e.g., DNA Repair, Tumor Suppressor, Ion Channel, Metabolism, etc.). The mutation impact tasks include both binary classification (high-impact vs. low-impact) and continuous regression (predicting an impact score between 0 and 1).

What Strategic Aspects are involved?

The strategic importance of this project lies in its contribution to computational genomics and personalized medicine. Understanding the functional role of genes and the potential harmful effects of mutations is critical for three main focus areas:

- **Disease Diagnosis:** Identifying whether a novel gene variant is likely to disrupt protein function will help clinical interpretation of genomic data
- **Drug Target Discovery:** Predicting gene function from sequence features can accelerate the identification of potential targets for new treatments
- **Evolutionary Biology:** Understanding sequence-function relationships provides insights into how organisms adapt and diverge over time

By automating feature extraction from publicly available GenBank records and training interpretable machine learning models, this project establishes a reproducible framework for genomic data science applications.

Relation to Course Lectures and Research

This project directly applies and integrates multiple foundational concepts from CS 439, as each component of the project directly reflects a major unit from the course:

- **Data Acquisition, Cleaning, and Validation:** Performs large scale data retrieval, using NCBI E-utilities followed by cleaning and validation using pandas API. We apply this to DataFrame operations, filtering, merging, and restructuring, GroupBy summaries, and memory-efficient transformations. Through Biopython to programmatically retrieve mRNA sequences, we effectively pipeline this data through a pre-processing sequence that includes several text-processing and standardization techniques. The pipeline also includes comprehensive validation logic to detect and remove malformed records, ensuring data integrity before model training.
- **Feature Engineering:** Extracting 115 numeric features from raw sequence data (including GC content, codon usage frequencies, k-mer distributions, and amino acid composition) exemplifies the feature engineering process. Although we ran the risk of potentially overfitting the data due to the volume of features, we believed it was necessary due to the information density each gene held.
- **Machine Learning Models:** The project implements multiple classification algorithms (Logistic Regression, Naive Bayes, Random Forest) and regression (Linear Regression), enabling comparative evaluation of model performance.
- **Model Interpretability:** Integration of SHAP (SHapley Additive exPlanations) values provides explainability for Random Forest predictions, aligning with course discussions on the importance of interpretable machine learning.
- **Graphical Representation:** Visualization components include PCA-based dimensionality reduction, correlation heatmaps, confusion matrices, and interactive 3D scatter plots.

II. Importance of the Project

Understanding gene function and the impact of mutations sits at the core of modern genomics and clinical genetics. As whole-genome sequencing becomes cheaper and more common, we can now generate huge amounts of sequence data much faster than we can interpret it. Being able to *computationally* predict how a genetic variant might affect gene function is therefore essential.

Our project focuses on bridging the gap between raw DNA sequence and biological function using machine learning. Instead of relying only on expensive and time-consuming lab experiments, we use features derived directly from gene sequences to train predictive models. This lets us:

- **Bridge sequence and function:** We infer gene function from sequence characteristics (such as nucleotide composition, codon usage, and amino acid composition) using machine learning models. This makes it possible to rapidly screen newly sequenced genes before any wet-lab validation.

- **Enable mutation triage:** Not every mutation is harmful. Our models aim to estimate the potential impact of mutations so that clinicians and researchers can prioritize high-risk

variants for deeper investigation, helping to streamline variant interpretation pipelines.

- **Democratize genomic analysis:** By using open-source tools (Biopython, scikit-learn, SHAP) and freely available data (NCBI GenBank), we show that meaningful genomic analysis can be done without proprietary software or specialized commercial databases. This lowers the barrier of entry for students, researchers, and small labs.

Why are we excited about it? This project sits at the intersection of biology and machine learning, which is one of the most exciting and fast-moving areas in data science. For us, it was a chance to take concepts we learned in class data cleaning, feature engineering, model training, and interpretability and apply them to real genomic data with direct relevance to human health. At the same time, working with biological sequences forced us out of our comfort zone. The "language" of our data is made up of nucleotides and codons instead of words or standard tabular features. As a team whose primary background is in Computer Science and Data Science rather than bioinformatics, we had to learn how to represent and interpret this kind of data effectively. That learning curve is part of what made the project challenging, but also very rewarding.

Existing Issues in Data Science Practices? Current approaches to gene function prediction and mutation impact assessment face several limitations:

- **Annotation lag** : Many genes in public databases lack functional annotations and manual curation cannot keep pace with the rate of new sequence data generation.
- **Data Heterogeneity:** Genomic data comes from diverse sources with varying quality and annotation standards, requiring robust preprocessing pipelines.

Related Works? Our project builds on open-source tools including Biopython for data acquisition, scikit-learn for model training, and SHAP for interpretability. We combined multiple feature types (nucleotide composition, codon usage, amino acid composition,etc.) into a unified classification framework.

III. Techniques Used

Data Science Component

Gene Selection: Fifteen genes were selected representing diverse functional categories:

- Disease-associated genes: BRCA1, BRCA2, TP53, CFTR, HBB, APOE, FBN1, DMD, HTT
- Housekeeping genes: ACTB, GAPDH, RPLP0, HPRT1, B2M, COX1

Data Source: We utilized the NCBI GenBank database, accessing mRNA sequence records rather than utilizing full shotgun genome sequences because mRNA provides a clean, spliced, gene-focused representation that includes the CDS and protein translation without massive

non-coding regions. This makes the data far easier to preprocess, compare across species, and use in machine learning, whereas full genomes are extremely large, noisy, and computationally impractical for our project scope.

Data Collection: Retrieval of this information was performed using the NCBI Entrez E-utilities API, where we implemented a search-and-fetch workflow: First querying the database with defined gene and RefSeq filters, then fetching each record by its returned ID. Lastly, for record consistency, each record was individually parsed into Biopython sequence objects which would allow for easier data accession. All outputs - raw GenBank files, mRNA FASTA, CDS FASTA, and protein FASTA - are stored in a well-organized local filesystem to support clean, reproducible data access.

Data Construction:

- **Sequence Metadata:** Sequence length, CDS start position, CDS end position, CDS length, protein length
- **Nucleotide Composition:** Percentage of each nucleotide (A, T, C, G) and overall GC content
- **Codon Usage:** Relative frequency of all 64 codons, capturing codon usage bias that reflects translational selection and species-specific preferences
- **K-mer Frequencies:** Distribution of all 16 dinucleotides (k=2), providing local sequence context information
- **Amino Acid Composition:** Percentage of each of the 20 standard amino acids in the translated protein sequence

The final cleaned dataset comprises **1,375 sequence records** with **115 numeric features per record**.

Machine Learning Component

Experiment 1 - Gene Function Classification

We implemented multiple classification algorithms to predict gene function:

- **Logistic Regression:** Used as a baseline model because it provides a clear linear decision boundary and allows us to evaluate how separable the gene classes are using only simple linear relationships among our engineered sequence features.
- **Gaussian Naive Bayes**: Handles high-dimensional biological features efficiently, makes strong independence assumptions that often hold reasonably well for compositional sequence data, and provides a fast performance baseline for comparison.
- **Random Forest:** We chose Random Forest as our primary classifier because it captures nonlinear interactions within nucleotide, amino-acid, and K-mer features, is robust to noise and irrelevant variables, and provides interpretable feature importance scores that help highlight biologically meaningful patterns across genes.

Setup: 80/20 train-test split with stratification to maintain class balance

Preprocessing: Standard scaling of all numeric features

Metrics: Accuracy, macro-averaged F1 score, per-class precision/recall/F1

MODEL PERFORMANCES

Model	Accuracy	Macro F1
Logistic Regression	99.27%	0.991
Naive Bayes	96.36%	0.949
Random Forest	98.55%	0.983

The performance of all three models was strong, with Logistic Regression(99.27% accuracy) slightly outperforming Random Forest(98.55%) on this dataset.

These results validate our modeling choices, showing that the selected classifiers were well-suited for gene function prediction and successfully captured the underlying biological patterns encoded in our engineered features. Ultimately, the Random Forest model was selected as the primary model due to its interpretability through feature importance and SHAP values.

Part 2 - Mutation Impact Prediction

We focused on predicting the functional impact of simulation mutations on coding sequences. This task examines whether a mutation is likely to cause a high-impact change to gene properties, such as GC content and protein length, enabling early detection of potentially disruptive sequence alterations. Considered both a classification and regression formulation which is Random Forest Classification, and Linear Regression. Initially experimented with Logistic Regression for the classification task but it under-performed on minority-class recall, whereas Random Forest handled the imbalance and nonlinear relationships more effectively.

- **Setup:** We simulated 5 mutations per CDS sequence, resulting in 75 mutation instances per gene.
- **Features:** We calculated original GC content, mutated GC content, delta GC, delta protein length, and original protein length.

- **Target:** We used a binary label (high impact = top 20% of impact scores) for classification.
- **Train/Test Split & Model:** We used 75% of the data for training, and 25% of the data for testing with stratification on high impact.
- **Metrics:** We report precision, recall, and F1 for both classes, and visualize performances using a confusion matrix

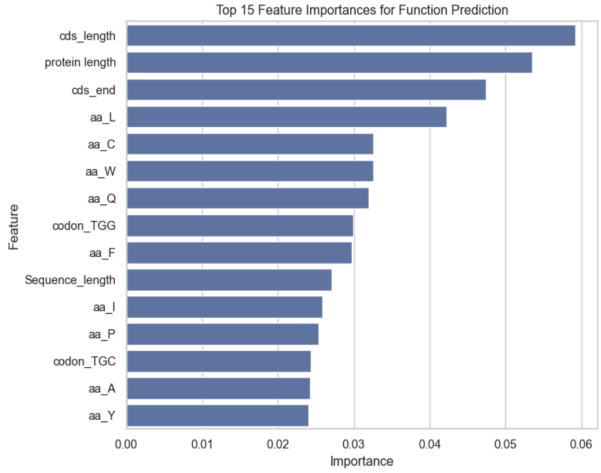
Results:

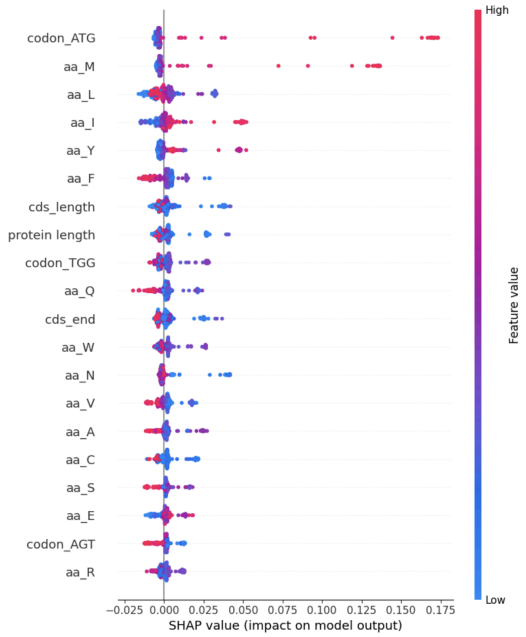
Model	**Accuracy**	**Macro F1**
Random Forest	100.00%	1.00

The Random Forest classifier achieved perfect performance on the test set which correctly identifies all high-impact and low-impact mutations. While the dataset is relatively small, these results demonstrate that the engineered mutation features provide a highly separable representation of mutation severity. The classifier effectively captured nonlinear relationships such as how simultaneous GC changes and protein truncation drive high-impact outcomes.

Information Visualization / Analysis Component

- **Gene Function is Predictable from Sequence Features:** All three models achieved excellent performance, this demonstrates that genes with different biological functions have distinct sequence signatures that machine learning models can detect.
- **Simulation Mutations have Distinct, Learnable Biological Impact Signatures:** In the mutation impact classification experiment, we simulated mutations which generate distinct feature changes that the Random Forest model achieved 100% accuracy correctly identified all high-impact mutations in the test set (100% recall for class 1).
- **Feature Importance Aligns with Biology:** Through SHAP value integration, we visualized feature importance, providing comprehensive interpretability of the model's decision-making process.



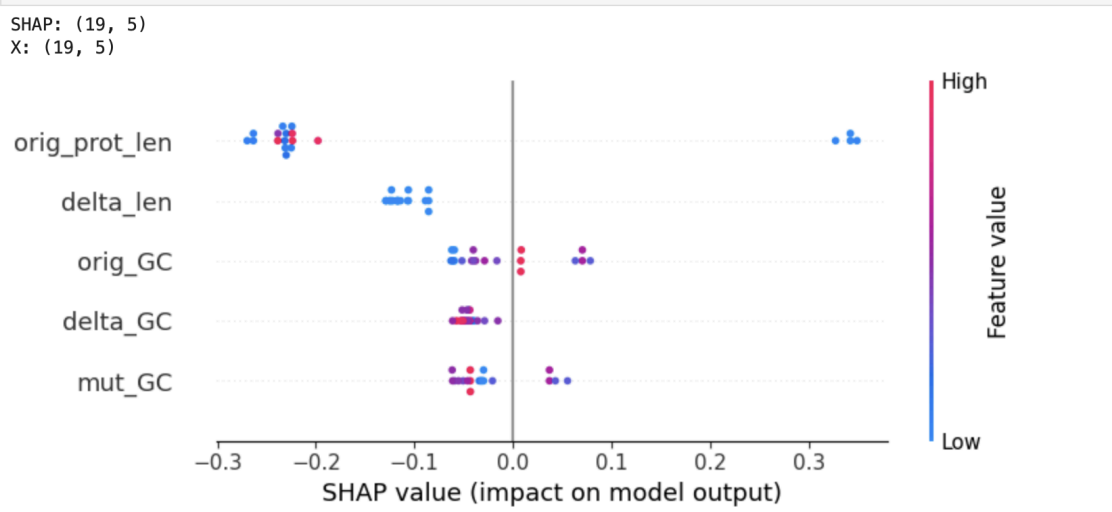


The first figure (Top Feature Importances for Gene Function Prediction) represents a bar plot which shows the top 15 most influential sequence features used by the Random Forest classifier to predict gene function. Length-based features and specific amino acid frequencies play the largest roles which shows that gene function is strongly associated with measurable sequence characteristics.

The second figure (SHAP Summary Plot for Gene Function Model) represents a SHAP beeswarm visualization which highlights how individual features contribute to the Random Forest model's gene function predictions. Codon usage, amino acid composition, and sequence length show strong directional influence which align with biological expectations and validating the interpretability of the model.

These findings make biological sense: different functional categories (e.g., structural proteins vs. enzymes) have characteristic size distributions and amino acid compositions.

- **Mutation Impact Can Be Estimated:** Both classification and regression approaches successfully predicted mutation impact. The low RMSE of the Linear Regression model suggests that even a simple linear relationship between sequence features and impact score captures the underlying biology.
- **SHAP Values Provide Biological Insights:** SHAP analysis revealed which features most influenced predictions for individual genes, enabling hypothesis generation about the sequence determinants of gene function.



The third figure (SHAP Summary Plot for Mutation Impact Model) represents a bar plot that shows the contribution of engineered mutation features, such as protein length changes and GC content changes, to the prediction of high-impact mutations. The clear separation of SHAP values shows that the simulated mutations generate consistent, learnable biological patterns that the model uses to identify severe functional impacts.

Advantages and Limitations

Our project offers several advantages, beginning with a fully automated data acquisition and feature extraction pipeline that ensures reproducibility and scalability with expansion to additional genes. The models we implemented achieved exceptionally high accuracy, which could pose a positive and negative sign in our discoveries. Potential explanations for this could be the overfitting issue, since we do use excessive features to classify genes. However, this could be justified with the intricate architecture of genomic data, meaning although we have attributed several features, we have also not included even more features from each record.

Additionally, our mutation-impact modeling offers the advantage of generating controlled simulated mutations which allows us to systematically analyze how small sequence changes affect protein structure and measurable features like GC content and length. The Random Forest model performed exceptionally well on this dataset as it shows these engineered features capture meaningful biological patterns. The main limitations are that simulated mutations may oversimplify real biological processes and that the small, high structures dataset can lead to overfitting and overly optimistic performance estimates.

IV. Changes After Proposal

In our proposal, we initially planned to incorporate exon count as a genomic structural feature. However, this became infeasible once we began working with mRNA RefSeq records, which do not preserve exon-level annotations. Because mRNA sequences represent fully spliced transcripts, every record consistently reported an exon count of 1, making the feature biologically uninformative and statistically meaningless for our machine learning pipeline. Extracting true exon structure would have required downloading full genomic or whole-genome shotgun (WGS) sequences, which are extremely large and would have introduced substantial storage overhead, processing latency, and pipeline complexity, far beyond the scope of our project. To address this constraint, we compensated by engineering a richer set of alternative sequence-based features, including k-mer frequencies, amino acid composition, GC content, and codon-derived attributes, which collectively captured meaningful evolutionary and functional variation across genes. These enhancements ensured that the predictive performance of our models was not hindered by the absence of exon count, while keeping the system computationally efficient and aligned with our project scope.