Enhancing Genomic Data Interpretation through Natural Language Processing

Janani Karthikeyan
Aniruthan Swaminathan Arulmurugan

04.16.2025

Problem Addressed

What is the problem?

- Identifying promoter regions in DNA sequences.
- Promoters play a critical role in regulating gene expression.

Why is this important?

- Understanding promoter regions helps decode biological functions and disease mechanisms.
- Accurate promoter identification is key to genomic research and biotechnology applications.

Why NLP?

Challenges with traditional methods:

- Require extensive lab work and biological expertise.
- Time-consuming, costly, and not scalable for large datasets.

Need for a solution:

- A faster, automated, and scalable method using machine learning.
- Leverages DNA sequence patterns to predict promoter presence with high accuracy.

Existing Works

Traditional Methods

- DNase footprinting, ChIP-seq: accurate but slow & costly
- Tools like PromoterScan used motifs → poor generalization

Classical ML

- SVM, Random Forest with handcrafted features (e.g., k-mers)
- Effective but required heavy feature engineering

Existing Works

NLP-Inspired Approaches

- Modeled DNA as text → k-mers + TF-IDF improved representation
- Enabled ML models to better learn sequence patterns

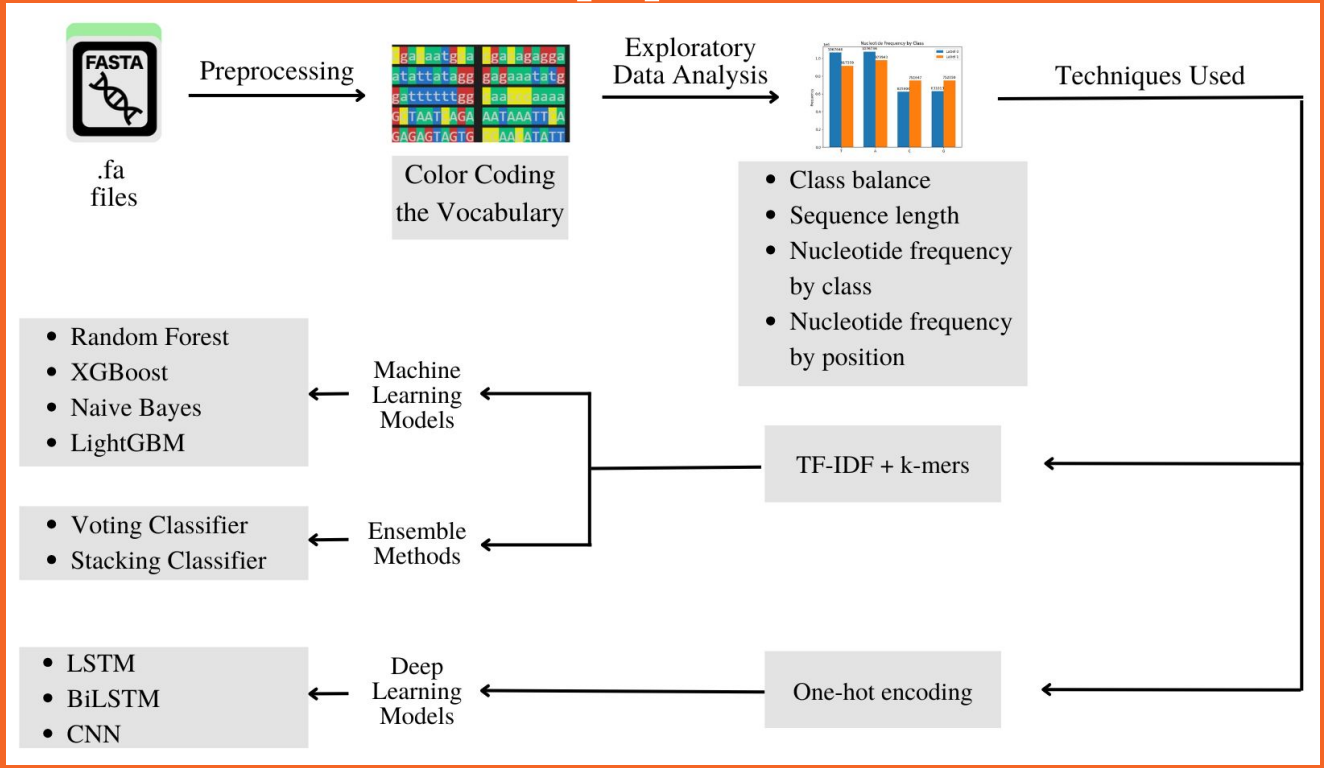
Deep Learning Advances

- CNNs, RNNs learned motifs & dependencies automatically
- Prior models: DeepBind, DeePromoter

Our Work

- Performing binary (2-class) classification:
  (0 -> non-promoter sequence, 1 -> promoter sequence)
- Predicting Promoter & Non-Promoter Sequence
- Combined ML + Ensemble Methods + NLP + Deep Learning
- Addressed class imbalance & motif variability

Our Approach



How?

Corpus Collection

- FASTA-format genome from Drosophila Melanogaster (22K sequences, balanced)

Preprocessing

- Cleaned, fixed-length sequences
- Removed unknowns and duplicates
- Labels mapped: Promoter = 1, Non-promoter = 0

Color-Coded Vocabulary

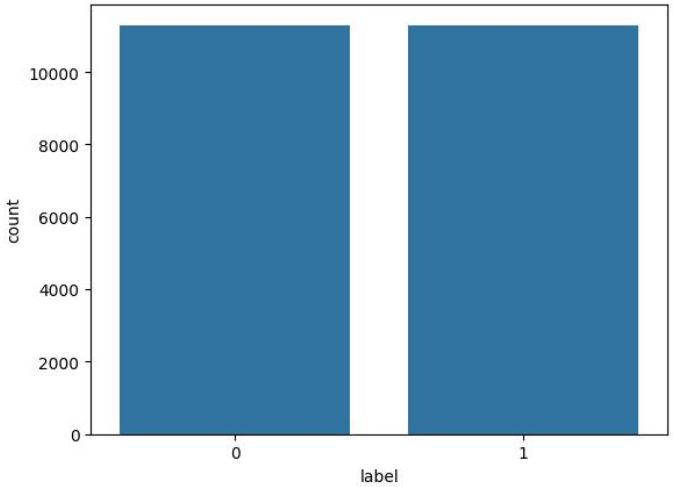
- A → Green (0), C → Yellow (1), G → Red (2), T → Blue (3)
- Used for EDA & k-mer simplification

Color-Coded Vocabulary

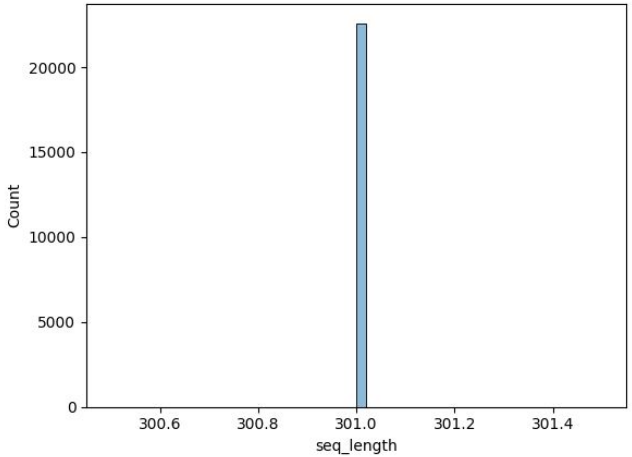
A → Green (0)
C → Yellow (1)
G → Red (2)
T → Blue (3)

EDA (Insights from genome patterns)

Class balance

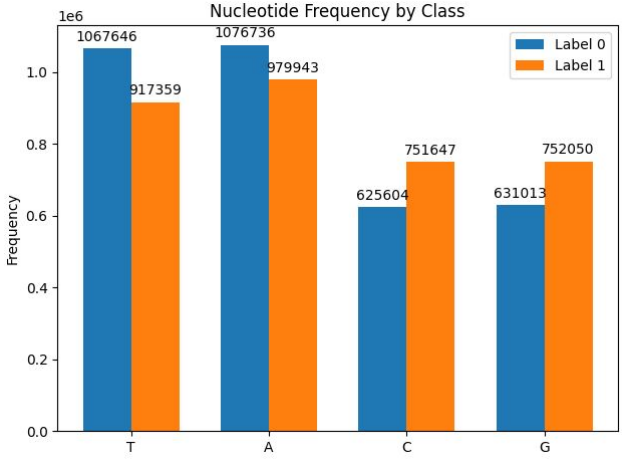


Constant sequence length

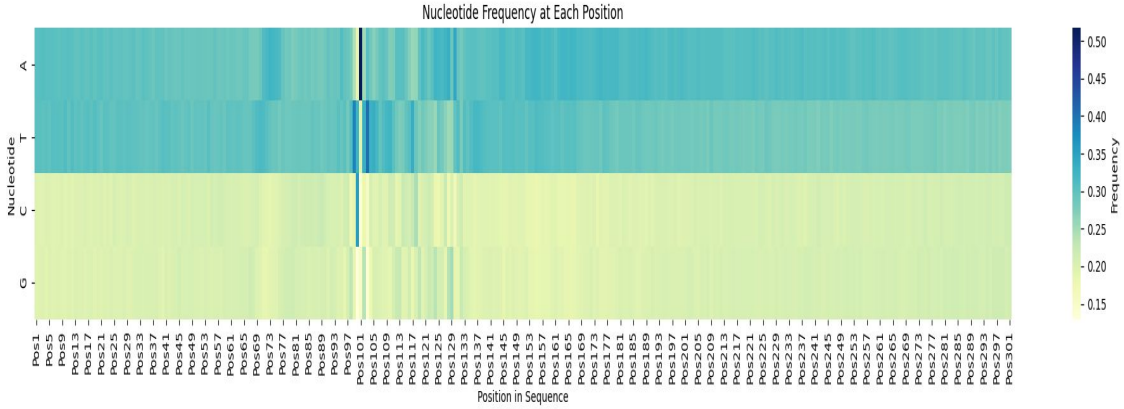


EDA (Insights from genome patterns)

Nucleotide frequency by class



Nucleotide frequency by position



How?

Techniques Used

- TF-IDF + k-mers for traditional ML (Random Forest, Naive Bayes, LightGBM, Logistic Regression, XGBoost)
- One-hot encoding for DL models (LSTM, BiLSTM, CNN)
- TF-IDF + k-mers for ensembles (Voting, Stacking Classifiers)

Modeling

- Compared 3 categories: ML, DL, Ensembles
- Best result from CNN with k-mer + dropout → 91.37% accuracy

One-hot encoding

Nucleotide	One-Hot Vector
A	[1, 0, 0, 0]
T	[0, 1, 0, 0]
C	[0, 0, 1, 0]
G	[0, 0, 0, 1]

Explanation - Why?

K-MERS

```
k=3 → Accuracy=0.7750
k=4 → Accuracy=0.7723
k=5 → Accuracy=0.7821
k=6 → Accuracy=0.7816
k=7 → Accuracy=0.7801
k=8 → Accuracy=0.7498
```

- Breaks DNA sequences into overlapping substrings of length k (e.g., ATGCGA → ATG, TGC, GCG, ...).
- Similar to tokenization in NLP, where sentences are broken into words or n-grams.
- Captures local patterns or motifs within sequences.
- Transforms raw DNA into a structured format suitable for ML/NLP models.
- Like n-grams in text, k-mers preserve sequence order and context.
- We have used k=5 as it provided the best results.

Explanation - Why?

TF-IDF

- TF-IDF (Term Frequency–Inverse Document Frequency) is used to numerically represent DNA sequences after k-mer tokenization, treating each k-mer like a word in NLP.

- **Reasons:**
  - Converts DNA into feature vectors usable by ML models
  - Highlights informative k-mers that are unique or rare
  - Reduces noise from common, non-discriminative patterns

Corpus Used

Drosophila Melanogaster Genome

(common fruit fly)

1. **Source:** IEEE DataPort
2. **Format:** Provided in FASTA format – standard for nucleotide sequences
3. **Content:** DNA sequences from the Drosophila Melanogaster genome. Each sequence is composed of A, T, G, C nucleotides
4. **Labels:** Binary classification: Promoter (1) vs. Non-Promoter (0)
5. **Size & Balance:** Contains 22,598 sequences. Dataset is balanced across both classes

Results

Model	Accuracy (%)
TF-IDF + Random Forest	72.2
TF-IDF + LightGBM	78.4
TF-IDF + Logistic Regression	78.2
TF-IDF + Naive Bayes	74.3
LSTM	69.1
BiLSTM	67.3
CNN	91.3

1. **CNN achieved the highest accuracy (91.3%)**
2. TF-IDF + LightGBM and Logistic Regression performed well (~78%)
3. Naive Bayes gave decent results (74.3%)
4. LSTM/BiLSTM underperformed (<70%)

Conclusion

1. Modeled DNA as text using TF-IDF + k-mers
2. CNN outperformed all models with 91.3% accuracy
3. Showed NLP + ML can improve genomic sequence classification

Future Work

1. Explore DNABERT and transformers
2. Add biological metadata
3. Expand to multi-class classification (e.g., enhancers, exons)

Thank you.