Assessing Reproducibility and Methodological Rigor: A Case Study of "Unified mRNA Subcellular Localization Predictor Based on Machine Learning Techniques"

Prepared for Dr. Marcel Turcotte

by
Steven Wilson, University of Ottawa

April 17, 2025

Table of Contents

1.0 Introduction 3

2.0 Background Information.....4

3.0 Problem Definition.....5

3.0 Data Description.....6

3.1 Source of the Data.....6
3.2 File Formats Utilized 6
3.3 Data Encoding Strategies and Rationale.....6

4.0 Methods.....7

5.0 Results 8

5.1 CatBoost.....8
5.2 XGBoost.....9
5.2 LightGBM.....10

6.0 Conclusion.....12

8.0 References.....13

Introduction

Messenger RNA (mRNA) subcellular localization plays a crucial role in regulating gene expression, cellular migration, and adaptive responses within eukaryotic cells. There are traditional experimental approaches such as RNA-FISH, smFISH, and advanced high-throughput methods like APEX-RIP, which offer high-resolution images of individual transcripts while delivering both RNA copy and subcellular localization. These methods are often labor-intensive, expensive and limited to specific tissues. [1] To overcome these challenges, *in silico* machine learning models have emerged as an efficient alternative. Particularly, the Unified mRNA Subcellular Localization Predictor (UMSLP) developed by Musleh et al. integrates four complementary feature sets: k-mer counts, pseudo k-tuple nucleotide composition, physiochemical properties, and z-curve transformations to predict subcellular locales (nucleus, cytoplasm, endoplasmic reticulum, extracellular region, and mitochondria) with over 94% accuracy on independent data.

Building on this foundation, our project is designed to assess the methodologies and conclusions presented in Musleh et al., while aiming to reproduce a portion of their work. We will apply comprehensive data cleaning, normalization, and encoding techniques to generate a robust dataset. By implementing at least two machine learning algorithms, including XGBoost as used in the original study and an alternative such as LGBM, we will compare model performance, examine hyperparameter sensitivities, and evaluate classification metrics (precision, recall, accuracy, specificity, and F1-score) to determine the strengths and potential limitations of each approach.

A central focus of this work is assessing the reproducibility of the computational techniques implemented in the study. There is a growing concern in bioinformatics and other scientific

disciplines highlighting the issue of reproducibility even when code and data is shared. Programming workflows can be limited by gaps in documentation, environment discrepancies, versioning issues, missing files and data. [2] Motivated by the Repohackathon initiative at Université Paris-Saclay, our project will provide all the code as a Jupyter Notebook which was used on Google Colab.

By demonstrating best practices in both methodological rigor and reproducible research, our work aims to validate and assess mRNA localization predictors but also as a model framework for our future bioinformatics studies.

Background Information

Messenger RNA (mRNA) molecules carry the genetic instructions that tell cells how to build proteins. After being transcribed from DNA, mRNAs travel to different parts of the cell, such as the nucleus, cytoplasm, endoplasmic reticulum, and mitochondria, to carry out their roles. Where an mRNA goes can influence which proteins are made, when they are produced, and how a cell responds to its environment. Understanding where mRNAs reside within the cell is therefore essential for decoding gene expression regulation and its impact on cellular function. [1]

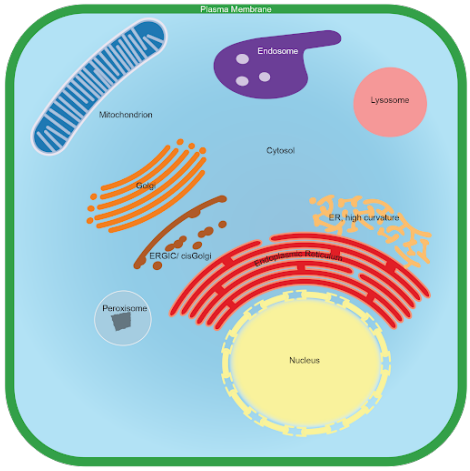

Figure I. Subcellular Localizations of a Eukaryotic Cell

Traditional experimental approaches to map mRNA localization rely on directly visualizing transcripts within cells. In fluorescence *in situ* hybridization (FISH) techniques probes complementary to target mRNAs are fluorescently labeled, allowing high resolution imaging. [3] Wet-lab methods provide rich information; however, they are costly, time-consuming, and often limited by throughput. By contrast, machine learning predictors can infer localization solely based on sequence information. First, RNA sequences are transformed into numeric feature vectors, for example, by counting k-mer frequencies. These features are then used to train classification models such as support vector machines (SVMs), gradient boosting machines (e.g. XGBoost, LightGBM), or ensemble methods combining multiple of these. The resulting models can achieve high accuracy across multiple locales and can be deployed rapidly.

Problem Definition

Our project addresses the challenge of reproducing a subset of the mRNA localization experiments described by Musleh et al. using their publicly released code and datasets. While the raw sequence data was provided, the authors' preprocessing scripts were not. To evaluate reproducibility and explore algorithmic choices, we trained both XGBoost (as in the original study), CatBoost (as in the original study), and LightGBM (our alternative) on our simpler reconstructed feature dataset. Complicating matters, the supplied codebase was fragmented and incomplete: key utility files were missing, dependencies were outdated, and scripts appeared to be stitched together from multiple files into one, therefore, needing extensive refactoring to create a cohesive and runnable workflow.

Data Description

The first analysis uses raw mRNA nucleotide sequences as input, capturing the distribution of mRNAs. Those sequences are converted into numerical features matrices for machine learning classification. The second analysis used the feature-engineered table provided by the authors for the machine learning classification.

Source of the Data

The authors sourced their mRNA sequences from the RNALocate v2 database and made it publicly available on GitHub (https://github.com/smusleh/UMSLP), including the combined FASTA file (mRNALoc_5_loca_master_fasta_files_combined_Cyto_Endo_Ext_Mit_Nuc.fasta). They also published a small cleaned, feature-engineered table as SampleData746.csv, derived from those sequences using their k-mer, PseKNC, and Z-curve preprocessing techniques.

File Formats Utilized

- FASTA (.fasta): plain-text format where each sequence record begins with a header line prefixed by >, followed by the nucleotide sequence.
- Comma Separated Values (.csv): Columns represent labels and numeric features (k-mer counts, PseKNC descriptors, Z-curve values)

Data Encoding Strategies and Rationale

In their study, Musleh et al. performed extensive feature engineering, computing k-mer sequences, pseudo k-tuple nucleotide compositions (PseKNC), physiochemical properties, and Z-curve descriptors, before applying dimensionality reduction. In our work, we did not replicate any of the advanced preprocessing steps, focusing solely on reconstructing the cleaned k-mer based feature matrix for each $k = 2, 3, 4, and\ 5$.

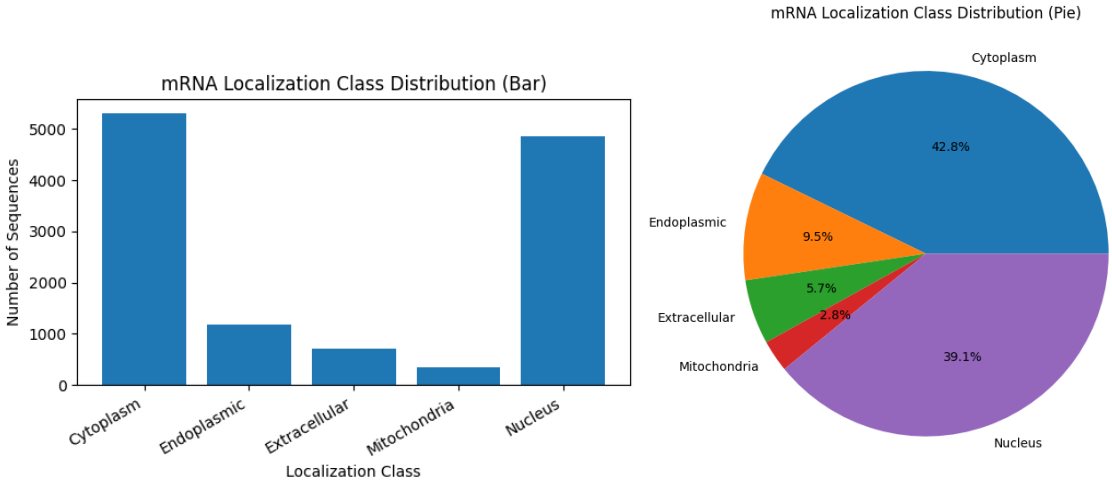

Figure II. Counts and Distributions of mRNAs in Different Subcellular Localizations

The mRNA distribution from the authors and ours convey the same overall distribution trends, showing that cytoplasm and nucleus account for the majority of mRNAs (approximately 43% and 39% respectively). However, the exact counts differ slightly (e.g. Cytoplasm: 6376 vs. 5400), reflecting the data cleaning the or other preprocessing steps the authors made.

Methods

We reproduced the dataset from the paper using the files provided on GitHub. The primary dataset used in our analysis was:

- mRNALoc_5_loca_master_fasta_files_combined_Cyto_Endo_Ext_Mit_Nuc.fasta

Which contains mRNA sequences annotated with their corresponding subcellular localization labels.

No preprocessing beyond parsing and formatting was applied to the raw sequences. Instead of replicating all the feature extraction techniques in the original study, we focused on reconstruing a clean, k-mer-based feature matrix. We extracted k-mer frequencies for *k = 1 through 5* using a Python script that normalized counts by sequence length and total possible k-mer combinations, resulting in a fixed length numerical representation of each sequence.

We used this feature matrix to train three classification models: CatBoost, XGBoost, LightGBM. Label encoding was applied to the localization categories, and an 80-20 stratified train-test split was used to ensure balanced class representation. Model performance was assessed using accuracy, precision, recall, and F1-score. All experiments were conducted in Python using sckit-learn, CGBoost, CatBoost libarires, with GPU acceleration enabled in Google Colab for faster training. In addition to XGBoost and CatBoost, we employed LightGBM due to its efficiency and scalability with high-dimensional, sparse datasets like our k-mer feature matrix.

Results

The results were obtained by training each model on the k-mer feature matrix and evaluating performance using metrics.

CatBoost

CatBoost achieved strong precision (0.9086 macro) and high accuracy (0.8066) on the test set, but showed lower recall (0.6671), especially for minority classes like Extracellular.

CatBoost (5-fold CV on TRAIN)		
Metric	***Score***	***SD***
Accuracy	0.7870	± 0.0041
Precision Macro	0.8718	± 0.0125
Recall Macro	0.6397	± 0.0094
F1 Macro	0.6789	± 0.0115

CatBoost evaluating on TEST				
Localization	***Precision***	***recall***	***f1-score***	***Support***
Cytoplasm	0.7717	0.8879	0.8257	1062
Endoplasmic	0.9573	0.4726	0.6328	237
Extracellular	1.0000	0.1620	0.2788	142
Mitochondria	1.0000	0.9286	0.9630	70
Nucleus	0.8142	0.8847	0.8480	971
Accuracy			0.8066	2482
Macro Avg	0.9086	0.6671	0.7096	2482
Weighted Avg	0.8255	0.8066	0.7886	2482

Table I. CatBoost Metric Evaluation on TRAIN and TEST

The feature importance plot highlights specific k-mers, such as "CCT" and "AAAAA", as key contributors to classification. The confusion matrix shows strong performance for major classes

like Cytoplasm and Nucleus, but considerable misclassifications for smaller classes like Extracellular.

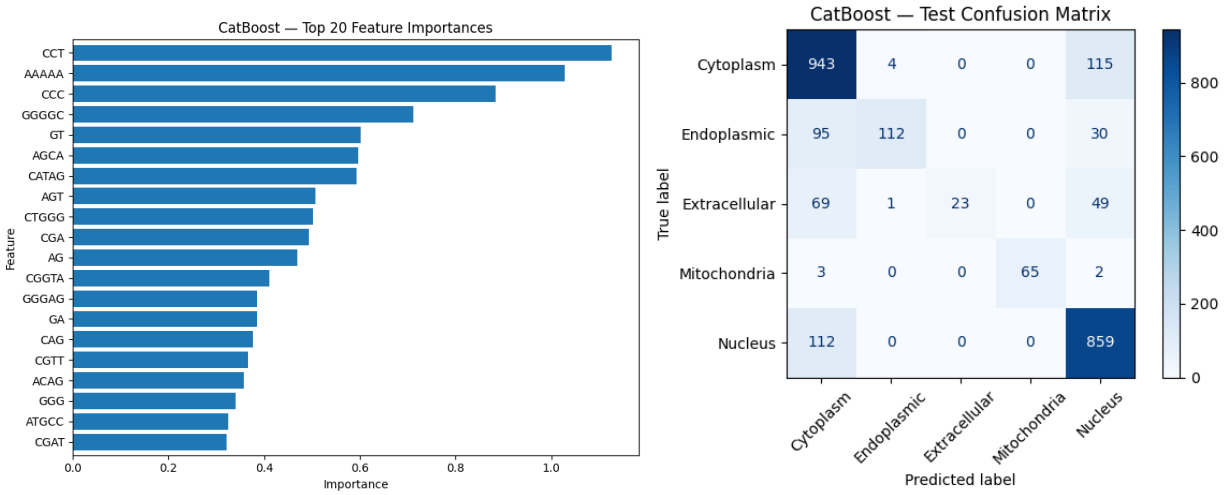

Figure III. CatBoost Feature Importances and Confusion Matrix

XGBoost

XGBoost performed comparably to CatBoost with a slightly higher test accuracy (0.8110) and similar macro precision (0.9059), but recall remained low (0.6752) for underrepresented classes. It showed strong classification for Cytoplasm, Nucleus, and Mitochondria, but struggled with Extracellular and Endoplasmic recalls.

XGBoost (5-fold CV on TRAIN)		
Metric	***Score***	***SD***
Accuracy	0.7841	± 0.0057
Precision Macro	0.8654	± 0.0150
Recall Macro	0.6351	± 0.0159
F1 Macro	0.6764	± 0.0174

XGBoost evaluating on TEST				
Localization	***Precision***	***recall***	***f1-score***	***Support***
Cytoplasm	0.7796	0.8927	0.8323	1062
Endoplasmic	0.9328	0.4684	0.6236	237
Extracellular	1.0000	0.1690	0.2892	142
Mitochondria	1.0000	0.9571	0.9781	70
Nucleus	0.8172	0.8888	0.8515	971
Accuracy			0.8110	2482
Macro Avg	0.9059	0.6752	0.7149	2482
Weighted Avg	0.8278	0.8110	0.7929	2482

Table II. XGBoost Metric Evaluation on TRAIN and TEST

The XGBoost feature importance plot highlights "CCT" and "CAG" as the most influential k-mers, though overall feature contributions are more evenly distributed than in CatBoost. The confusion matrix reveals strong predictions for dominant classes but continued misclassification in minor ones like Extracellular.

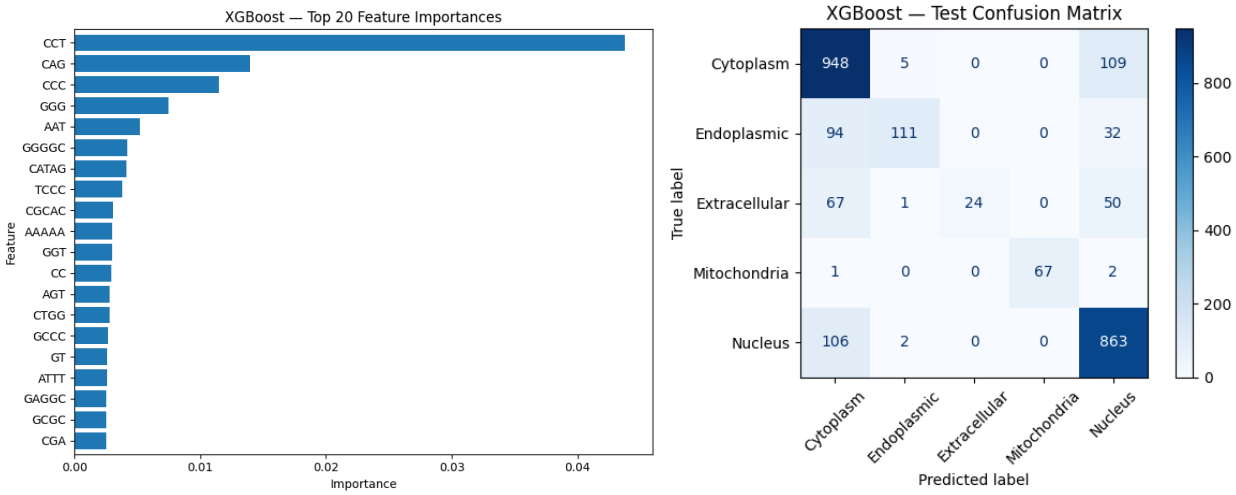

Figure IV. XGBoost Feature Importances and Confusion Matrix

LightGBM

LightGBM achieved the highest test accuracy (0.8207) among the models, along with strong macro precision (0.9090), but recall (0.6893) remained low for underrepresented classes. It performed particularly well on dominant classes like Nucleus and Mitochondria, but struggled with Extracellular recall, similar to the other models.

LightGBM (5-fold CV on TRAIN)		
Metric	***Score***	***SD***
Accuracy	0.8004	± 0.0032
Precision Macro	0.8779	± 0.0123
Recall Macro	0.6567	± 0.0078
F1 Macro	0.7002	± 0.0102

LightGBM evaluating on TEST				
Localization	***Precision***	***recall***	***f1-score***	***Support***
Cytoplasm	0.7961	0.8898	0.8404	1062
Endoplasmic	0.9302	0.5063	0.6557	237
Extracellular	1.0000	0.2042	0.3392	142
Mitochondria	1.0000	0.9429	0.9706	70
Nucleus	0.8189	0.9032	0.8590	971
Accuracy			0.8207	2482
Macro Avg	0.9090	0.6893	0.7330	2482
Weighted Avg	0.8352	0.8207	0.8050	2482

Table III. LightGBM Metric Evaluation on TRAIN and TEST

The LightGBM feature importance plot shows a strong reliance on specific k-mers like "AAAAA" and "GGTA", with overall higher importance values compared to the other models. The confusion matrix confirms solid classification of major classes but continued confusion between Endoplasmic, Extracellular, and Cytoplasm categories.

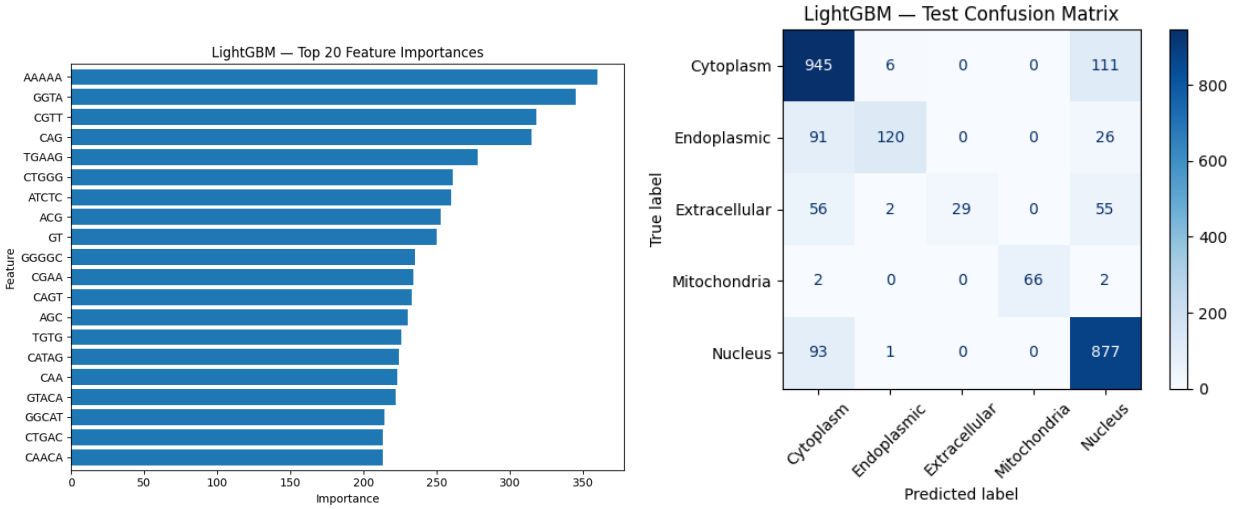

Figure V. LightGBM Feature Importances and Confusion Matrix

Conclusion

Our k-mer only implementation delivered strong macro-precision (~ 0.91) and moderate recall (~ 0.67 - 0.69) across CatBoost, XGBoost, and LightGBM, but overall test accuracy (0.81- 0.82) was roughly 10 percentage points short of the UMSLP's (~ 0.91 - 0.92) accuracy. This gap reflects the differences between our feature set and those of the authors', relying on normalized k-mer frequencies versus the authors' richer combination of PseKNC, physiochemical, and Z-curve descriptors plus ensemble stacking. The persistent misclassification of the localizations that have less sequences highlight that class imbalance is something that must be addressed more directly, and is very important to achieve high scores across all subcellular localizations.

In summary, our replication of the study demonstrates that, even when limited to basic k-mer features and an incomplete codebase, it is possible to reconstruct a functional mRNA localization machine learning model based on a scientific study. By training XGBoost, CatBoost, and LightGBM only on normalized k-mer counts, we validated the core premise of Musleh et al., that sequence composition alone carries significant localization signal, while also revealing the value added by their richer and more complex feature engineering and ensemble methods in boosting overall accuracy and minority class recall. The effort of refactoring fragmented code and independently implementing data cleaning, normalization, and feature selection demonstrated the challenges inherent to reproducibility in bioinformatics. Going forward, our study and notebooks provide a clear, reproducible starting point for others to build on, add new features, and improve handling of rare classes.

References

[1] Musleh S, Arif M, Alajez NM, Alam T. Unified mRNA Subcellular Localization Predictor based on machine learning techniques. BMC Genomics. 2024;25:151. doi:10.1186/s12864-024-10077-9.

[2] Cokelaer T, Cohen-Boulakia S, Lemoine F. Reprohackathons: Promoting reproducibility in bioinformatics through training. Bioinformatics. 2023 Jun;39(Suppl_1):i11–i20. doi:10.1093/bioinformatics/btad227.

[3] NanoString Technologies. Why oligonucleotides are critical research components [Internet]. NanoString; [cited 2025 Apr]. Available from: https://nanostring.com/blog/why-oligonucleotides-are-critical-research-components/

[4] Nature Education. Fluorescence in situ hybridization (FISH) [Internet]. Nature Publishing Group; [cited 2025 Apr]. Available from: https://www.nature.com/scitable/topicpage/fluorescence-in-situ-hybridization-fish-327/

[5] OpenAI. ChatGPT (o4-mini) code development assistance and assitive writing [Internet]. OpenAI; [cited 2025 Apr]. Available from: https://chat.openai.com/

[6] Cui T, Dou Y, Tan P, Ni Z, Liu T, Wang D, et al. RNALocate v2: an updated resource for RNA subcellular localization with increased coverage and annotation. Nucleic Acids Res. 2022;50(D1):D333–9. doi:10.1093/nar/gkab122.