Predicting Functional Dropout Outcomes in Prime Editing Screens Using Machine Learning

Daivik Patel, Shrenik Patel

Abstract

Prime editing is a versatile and precise genome-editing technique that enables targeted mutations without inducing double-stranded breaks. In their recent study, Deshpande et al. (2024) presented a high-throughput platform for evaluating the functional consequences of prime editing across essential genes in K562 cells. By monitoring epegRNA dropout, they identified mutations likely to result in loss-of-function phenotypes. In this work, we build upon their dataset and methodology to develop predictive models capable of anticipating dropout outcomes based on sequence and design features. We show that classical and ensemble machine learning approaches can moderately predict dropout events using minimal input features. These predictive models, paired with biological interpretation, offer an initial path toward functional variant prioritization and more scalable experimental design.

1 Introduction

Prime editing has emerged as a next-generation genome editing tool capable of introducing targeted substitutions, insertions, and deletions [2]. The study by Deshpande et al. [1] presents a scalable platform using epegRNAs to introduce designed mutations in essential genes. The platform measures cellular dropout effects by calculating Z-scores over time, identifying mutations that negatively impact proliferation.

While this approach is experimentally robust, it relies on resource-intensive pooled screens. We hypothesize that machine learning (ML) models can complement this framework by providing predictive capabilities based on simple, design-accessible features. Our goal is to evaluate whether dropout events can be predicted using features such as GC content, edit length, and epegRNA subtype. If successful, this would enhance experiment design by prioritizing mutations with high likelihood of causing functional impact.

2 Methods

2.1 Dataset and Preprocessing

We used the supplementary dataset from Deshpande et al. (2024), specifically Supplemental Table 2. We filtered rows marked as `included_in_analysis` and extracted key features:

- `Edit_length`: length of the intended genomic edit
- `gc_content`: calculated GC fraction of the protospacer sequence
- `epegRNA_type`: one-hot encoded categorical subtype

The target variable, `dropout`, was defined as a Z-score below -2 at day 28 (`Z_score_d28_avg < -2`). We addressed class imbalance by upsampling dropout instances to match the number of non-dropouts, ensuring balanced model training.

2.2 Model Training

We trained the following models to predict dropout:

- Random Forest Classifier (baseline ensemble model)
- Logistic Regression (interpretable linear baseline)
- Gradient Boosting Classifier (tree-based boosting model)

All models were trained on a 70/30 train-test split of the balanced dataset. We used default hyperparameters for initial evaluation and reported classification metrics and ROC AUC scores to assess performance.

2.3 Evaluation Metrics

We computed:

- Precision, Recall, F1-score (per class)
- ROC AUC (Area Under the Curve)
- Confusion matrices (numeric and visual)

Performance comparison was visualized using a grouped bar chart summarizing all metrics.

3 Results

3.1 Random Forest Classifier (Baseline)

Class	Precision	Recall	F1-score	Support
False	0.56	0.72	0.63	22558
True	0.61	0.44	0.51	22666
Accuracy		0.58 (Total: 45224)		

Table 1: Random Forest classification performance

	Pred No	Pred Yes
Actual No	16287	6271
Actual Yes	12717	9949

Table 2: Random Forest Confusion Matrix (numeric format)

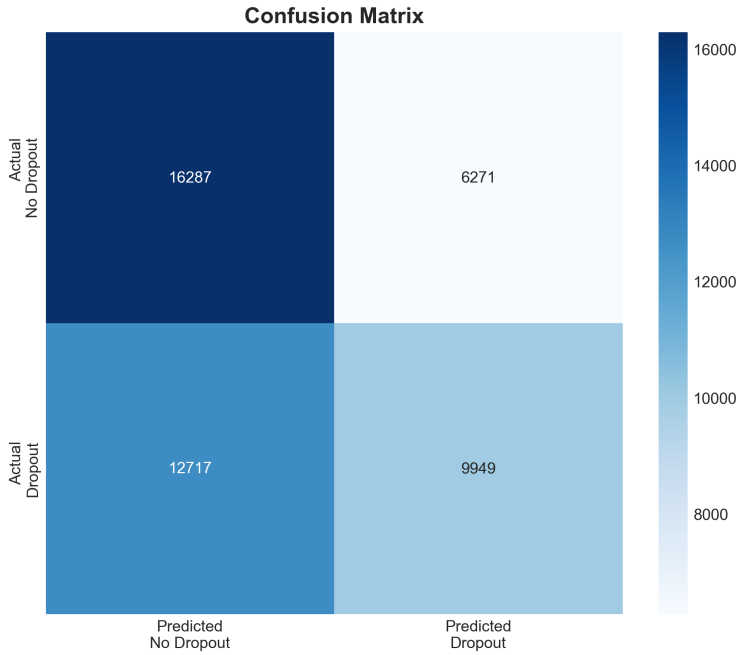

Figure 1: Visual representation of the confusion matrix for the Random Forest classifier. The heatmap highlights class imbalance and model bias, where false negatives (bottom left quadrant) are relatively high. This suggests that while the model identifies many non-dropouts correctly, it frequently misses true dropout cases.

3.2 Logistic Regression

Class	Precision	Recall	F1-score	Support
False	0.58	0.54	0.56	22558
True	0.57	0.61	0.59	22666
Accuracy		0.57 (Total: 45224)		

Table 3: Logistic Regression classification performance

AUC: 0.604

3.3 Gradient Boosting

Class	Precision	Recall	F1-score	Support
False	0.56	0.72	0.63	22558
True	0.61	0.44	0.51	22666
Accuracy		0.58 (Total: 45224)		

Table 4: Gradient Boosting classification performance

AUC: 0.610

3.4 Performance Summary

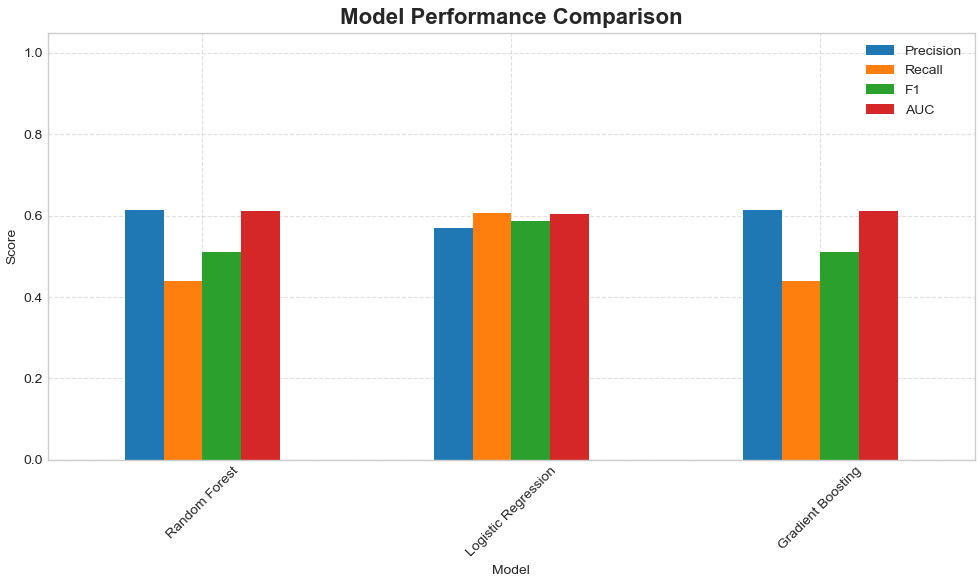

Figure 2: Bar chart comparing Precision, Recall, F1, and AUC across models. Gradient Boosting and Random Forest show similar behavior, with Logistic Regression offering a slightly better recall but lower AUC.

4 Discussion

Our analysis demonstrates that dropout events in high-throughput prime editing screens can be predicted with moderate accuracy using minimal, design-derived features. The Random Forest and Gradient Boosting models achieved the highest F1 and AUC scores, while Logistic Regression provided a consistent linear baseline.

The confusion matrix visualization (Figure 2) highlights key model limitations. Specifically, the Random Forest model produces a substantial number of false negatives (12,717), indicating difficulty in identifying all dropout-causing edits. However, the true positive count (9,949) also reflects that the model captures a meaningful subset of impactful mutations. These patterns underscore the potential value—and limitations—of using ML models as prioritization tools rather than definitive classifiers.

The predictive power, although modest (AUC ~ 0.60), is statistically above random guessing (AUC $= 0.5$), suggesting the presence of underlying biological signals. This extends the findings of Deshpande et al. by showing that dropout likelihood can be approximated from edit features alone, offering a computational layer to support functional genomics.

Future improvements could include:

- Integrating mutation type annotations (e.g., synonymous, nonsense)
- Using gene essentiality or evolutionary conservation scores
- Leveraging deep learning for sequence embedding and structured context

5 Conclusion

We show that dropout outcomes in prime editing screens, as defined by Deshpande et al. (2024), can be partially predicted using simple ML models and basic sequence-derived features. While not a substitute for experimental validation, this approach offers a lightweight computational complement to large-scale CRISPR-based functional screens. These findings suggest opportunities to scale prime editing designs more efficiently and inform follow-up prioritization.

References

[1] Deshpande, A., Vlaming, H., Lei, L., Marin, M., Vu, J. T., Wong, J. C., ... Bassik, M. C. (2024). Functional evaluation of a scalable prime editing platform for introducing mutations across essential genes. *Nature Methods*, 21, 391–401. `https://doi.org/10.1038/s41592-024-02502-4`

[2] Anzalone, A. V., et al. (2019). *Search-and-replace genome editing without double-strand breaks or donor DNA*. Nature, 576(7785), 149–157.