Machine Learning-Based Prediction of Prime Editing Efficiency

Yousef Jan

Abstract

Prime editing is a novel genome editing technique which improves on some of the shortcomings of earlier CRISPR-based editing systems. However, multiple optimization trials are necessary to determine maximally efficient prime editing guide RNAs (pegRNAs) for a desired edit. There is great potential for predictive tools in overcoming this barrier. To this end, I have made use of editing outcomes of 447,321 pegRNAs gathered from published data collected via high-throughput prime editing experiments to train and test a number of machine learning algorithms. A broad variety of features previously established in affecting efficiency were factored into the models including biophysical and genetic information pertaining to the pegRNA and targeted DNA sequence. The best performing model, utilizing the convolutional neural network architecture, yielded a Pearson's r of and Spearman's R of . This model outperforms existing predictive tools and illustrates the ability of machine learning methods in the field of genome engineering.

Dr. Evgeni (Zhenya) Ivakine conceived the original idea for this research project and provided guidance throughout the project's evolution. Yousef Jan was responsible for the execution of all aspects of the project including data collection and processing, model design, statistical analysis, and the write-up.

Introduction

The emergence of CRISPR-based gene editing techniques has transformed the field of genetic engineering, in large part as they have eliminated the need for complex design processes for every edit as in their predecessors (Jinek et al., 2012). The latest of them, prime editing, is the most sophisticated technology to date, enabling all possible substitutions, insertions up to 44bp, deletions up to 80bp in a targeted DNA sequence as well as combinations of the aforementioned (Anzalone et al., 2019).

Prime editing involves the use of a prime editor - a fusion protein combining the Cas9 nickase (from Streptococcus pyogenes) and reverse transcriptase - and a prime editing guide RNA (pegRNA) (figure 1). The pegRNA comprises a spacer sequence (which base pairs with the protospacer), a scaffold sequence, and a 3' extension encoding the primer binding site (PBS) and reverse transcriptase template (RTT) carrying the desired edit to be made (figure 1). Guided by the spacer sequence, Cas9 nicks the targeted strand three base pairs upstream of the protospacer adjacent motif (PAM in figure 1). Following, the PBS on the 3' end of the pegRNA hybridizes with the forward strand and reverse transcriptase synthesizes an edited DNA flap containing the desired edit using the reverse transcriptase template (RTT). The edited and unedited flaps remain in equilibration until DNA repair machinery excises the unedited flap, successfully installing the edit (Anzalone et al., 2019; Chen et al., 2021). Many prime editing systems exist, their editors coined PE1, PE2, PE3, PE4 and PE5 with 'max' variants of each. Described above is the scheme for PE1. PE2 differs in that reverse transcriptase is mutated for greater thermostability

(Anzalone et al., 2019; Arezi & Hogrefe, 2009). Remaining systems have further modifications.

In all systems, a significant challenge is optimizing editing efficiency (the ratio between the number of edited allele and number of edited + nonedited alleles). Many design decisions affecting it must be made when selecting the pegRNA. It is possible that thousands of pegRNAs could be used for a single edit. Most all of the design decisions affect the pegRNA's efficiency by way of but not limited to its constituent lengths, nucleotide contents, melting temperatures and folding energies. As a consequence, multiple optimization trials are necessary to determine those that are maximally efficient.

Thus far, there have been some efforts for automating this design process, particularly by statistical learning. In recent years, Mathis (2023a); Mathis (2023b); Kim (2021); Yu (2023) have developed machine learning models trained using data generated from high-throughput experiments. In this study, I have leveraged the aforementioneds' published datasets along with the convolutional neural network architecture in building an improved model for predicting prime editing efficiency (Kim, Mathis 2023a,b, Yu).

Methods

To build machine learning models via supervised learning, a large amount of labelled data is needed. That is - data consisting of pegRNAs and their efficiencies. I kept my scope to data consisting of the outcomes of editing using the PE2 system in HEK293T cells. The first of the data sources were from which was used to train DeepPE (Kim et al., 2021). This study was largely limited to G to C edits at the +5 position (from the nicking site) and included 43,149 pegRNAs. The second set was used to train DeepPrime and consists of 288,793 pegRNAs for substitution, deletion, and insertion edits (Yu et al., 2023). Thirdly was 92,423 pegRNAs from Mathis et al. (2023) used to train PRIDICT, and lastly was 22,956 to train PRIDICT2.0 (Mathis, 2023b). Combined, this data contains the efficiencies of 447,321 pegRNAs representing a diversity of edit contexts, types, and lengths. For use in machine learning, the data was randomly sampled and split into training, validation, and testing datasets in the ratio 70:15:15.

Within some datasets and in the combined data, there was overrepresentation of low efficiency and underrepresentation of medium-high efficiency pegRNAs. For the purpose of training a deep learning model, it is imperative that a range of target values are observed in order to ensure the model is generalizable. To rectify this, a weighted cost function $\mathcal{E}'$ was employed which assigned weights to the mean squared loss (defined as squared difference between estimate $\hat{y}^{(i)}$ and target $y^{(i)}$ efficiency vectors) in accordance with their level of representation in the data of size N. This expression was found by estimating the inverse distribution of efficiency range frequencies using a polynomial function, μ.

$$\mu\left(x\right) = \frac{5}{3 \times 10^{20}} x^{10} - \frac{8}{10^{5}} x^{2} + 0.006x$$

$$\mathcal{E}' = \frac{1}{N} \sum_{i=1}^{N} \mu(y^{(i)}) \left(y^{(i)} - \hat{y}^{(i)}\right)^{2}$$

I have considered the effects of twenty-eight features including those of the pegRNA and wide target sequence in my models (supplementary table 1), all of which have been established as efficiency affecting factors (cite). These can be broadly sorted into two groups: genetic, and biophysical. Genetic features encompassing effects of edit type, position, length were found using Python scripts when not directly available in the data. Target sequence and pegRNA GC content, and poly-U content were found using the python Biopython package version 1.81 (cite). Biophysical features including melting temperature and minimum free energy of folding of the oligo were found with python package ViennaRNA version 2.6.4 (cite). DeepSpCas9 score predicts nicking efficiency of the Cas9-spacer sequence complex and can naturally be extended for prediction of PE2-pegRNA activity (Kim et al.,

2019). This feature was provided in all datasets.

A number of linear regression-based and tree-based models were trained and tested. These include lasso, random forest (RF), gradient boosting (GB), extreme gradient boosting (XGBoost), CatBoost and LightGBM regressors (cite). A variety of neural network architectures were also tried including feed-forward, long short-term memory and convolutional neural networks. Hyperparameter optimization was done via randomized grid search using the validation data. Following previous studies, Spearman's r was the criterion for hyperparameter tuning and final model evaluation (Yu et al., 2023). This was performed using the Python package scikit-learn version 1.3.2. In the final selected model, feature importance was assessed by Shapley additive explanations (SHAP), a generalizable tool for assessing feature importance in machine learning models (Lundbeg & Lee, 2017).

Results

Exploratory data analysis revealed...

Among all published models, GC content of the pegRNA, emerged as a highly important feature positively associated with efficiency (cite). A possible explanation proposed by Kim et al. (2021) is that a higher proportion of GC nucleotides, particularly in the PBS, results in tighter base pairing with the target DNA allowing more time for reverse transcriptase to act. Further, the choice of poly-U nucleotide stretch length as a feature in prime editing efficiency (first identified by Mathis et al. (2023)) was chosen since they have been shown to cause stalling of RNA polymerase III (Nielson et al., 2013). Carrying these trends, my final model also shows importance of these features (figure 7).

The reported effects of edit type, position, and length on efficiency predictions in the literature has been somewhat variable. Yu et al. (2023) observed slightly lower efficiencies for insertion and deletion edits compared to substitutions, but this trend was not statistically significant at endogenous loci. Mathis et al., 2023 found clear pattern in single nucleotide substitutions being more efficient than other edits. In aggregate, the combined data I used reflects this. Mean editing efficiency for substitution edits were calculated to be 19% vs 17% for insertions and 9% for deletions. Coming to edit length, there is significant experimental evidence to support a negative relationship between it and efficiency (cite). This trend in deletion and insertion edits is clear in my data (figure 2). Edit position has a similar negative relation with efficiency as can be seen by the gradient in figure 3. A notable spike is observed at positions +5 and +6 due to PAM disruption, preventing PE2 from rebinding increasing probability of WT sequence being restored.

Another feature previously shown to have very strong correlation with editing efficiency is the DeepSpCas9 score generated from the deep learning model, DeepSpCas9. Although Kim et al. (2021) report only "modest" correlation between the two measures, their group as well as Li et al., 2021 (also using the DeepPE dataset) observed high importance of this feature in their developed models (Yu et al., 2023). In contrast, Mathis et al., 2023 did not observe high importance of DeepSpCas9 score in their XGBoost model and did not include it in their final neural network. Correlations between DeepSpCas9 score and efficiency among each of the four datasets I used are shown in figure 4. For my model, I have opted to include this feature due to its theoretical feasibility and as it is a strong predictor for 74% of the collated data. My SHAP analysis showed this feature was important but ranked lower than in DeepPE and DeepPrime (figure 7) (Yu et al., 2023; Kim et al., 2021).

Biophysical features take into account the physical environment of the prime editor and pegRNA. They also, to an extent, represent details about the sequence lengths and contents that are important for prime editing efficiency. Previous studies' SHAP analyses show very high importance of the melting temperature of the RTT overhang sequence (figure 1) and PBS (cite). Notably, when PRIDICT was tested using data used to train DeepPE, a low

RTT overhang melting temperature was associated with higher efficiency as opposed to lower efficiency in the PRIDICT dataset, likely attributable to lack of variation in DeepPE data.

The derived features were then used to train a number of machine learning models as previously described. Model performance, measured by Spearman's r, in each of the models developed is shown in figure 6. SHAP analysis was performed on the best performing model shown in figure 7.

Discussion

As described, I have developed a machine learning model for predicting prime editing efficiency for the purpose of pegRNA design automation. I made use of the most all large publicly available datasets of prime editing outcomes. My model outperforms existing ones

References

Anzalone, A. V., Randolph, P. B., Davis, J. R., Sousa, A. A., Koblan, L. W., Levy, J. M., & Liu, D. R. (2019). Search-and-replace genome editing without double-strand breaks or donor DNA. Nature, 576(7785), 149-157.

Arezi, B., & Hogrefe, H. (2009). Novel mutations in Moloney Murine Leukemia Virus reverse transcriptase increase thermostability through tighter binding to template-primer. Nucleic acids research, 37(2), 473-481.

Chen, P. J., Hussmann, J. A., Yan, J., Knipping, F., Ravisankar, P., Chen, P. F., ... & Liu, D. R. (2021). Enhanced prime editing systems by manipulating cellular determinants of editing outcomes. Cell, 184(22), 5635-5652.

Costa, B. L. D., Levi, S. R., Eulau, E., Tsai, Y. T., & Quinn, P. M. (2021). Prime editing for inherited retinal diseases. Frontiers in Genome Editing, 35.

Fleuret, F. (2023). The Little Book of Deep Learning. A lovely concise introduction. Gaudelli, N. M., Komor, A. C., Rees, H. A., Packer, M. S., Badran, A. H., Bryson, D. I., & Liu, D. R. (2017). Programmable base editing of A• T to G• C in genomic DNA without DNA cleavage. Nature, 551(7681), 464-471.

Kim, H. K., Yu, G., Park, J., Min, S., Lee, S., Yoon, S., & Kim, H. H. (2021). Predicting the efficiency of prime editing guide RNAs in human cells. Nature Biotechnology, 39(2), 198-206.

Li, Y., Chen, J., Tsai, S. Q., & Cheng, Y. (2021). Easy-Prime: a machine learning–based prime editor design tool. Genome biology, 22, 1-11.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

Mathis, N., Allam, A., Kissling, L., Marquart, K. F., Schmidheini, L., Solari, C., & Schwank, G. (2023a). Predicting prime editing efficiency and product purity by deep learning. Nature Biotechnology, 1-9.

Mathis, N., Allam, A., Tálas, A., Benvenuto, E., Schep, R., Damodharan, T., Balázs, Z., Janjuha, S., Schmidheini, L., Böck, D., van Steensel, B., Krauthammer, M., & Schwank, G. (2023b). Predicting prime editing efficiency across diverse edit types and chromatin contexts with machine learning. BioRxiv. https://doi.org/10.1101/2023.10.09.561414

Nielsen, S., Yuzenkova, Y., & Zenkin, N. (2013). Mechanism of eukaryotic RNA polymerase III transcription termination. Science, 340(6140), 1577-1580.

Yu, G., Kim, H. K., Park, J., Kwak, H., Cheong, Y., Kim, D., & Kim, H. H. (2023). Prediction of efficiencies for diverse prime editing systems in multiple cell types. Cell, 186(10), 2256-2272.

Supplementary Information

Supplementary Table 1: List of features used in the model.