PE Ensemble: An Interactive Review of In-silico Prime Editing Guide Design Tools

Abstract

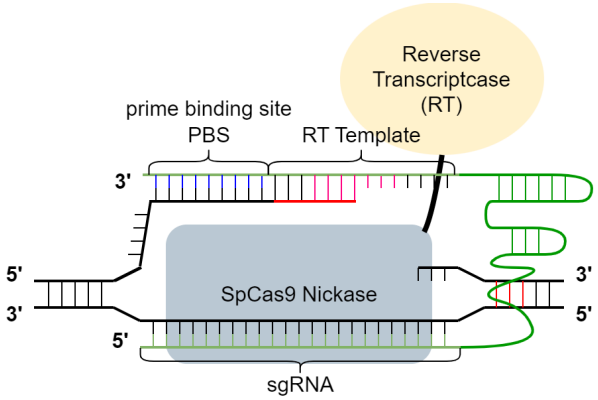
Prime editing is a novel genome editing technology that enables precise base editing without the need for double-strand breaks. The design of prime editing guides is a critical step in the prime editing workflow. In this review, we evaluate the performance of several in-silico prime editing guide design tools. We compare the quality of the guide designed by these tools, and to improve the usage of these state of art tools, they were reimplemented and integrated into a single web base application. Additionally, we provided the ability to aggregate the results from multiple tools using ensemble learning to improve the overall guide design quality. Thus, with the on and off target activity of prime editing quantified, we can provide a complete overview of the outcome of using a specific pegRNA sequence on a specific target loci in a specific cell line. This should noticeably improve the safety and efficiency of prime editing, and thus accelerate its clinical application. Thus, with the on and off target activity of prime editing quantified, we can provide a complete overview of the outcome of using a specific pegRNA sequence on a specific target loci in a specific cell line. This should noticeably improve the safety and efficiency of prime editing, and thus accelerate its clinical application.

Keywords: Prime editing, Machine Learning, in-silico tools, Ensemble Learning

1 Background

Prime editing is a versatile and precise genome editing technology that enables the introduction of all 12 possible base-to-base conversions as well as insertions and deletions without the need for double-strand breaks[1].

The versatility of prime editors comes from the fusion of a reverse transcriptase (RT) and a Cas9 nickase (nCas9) to a prime editing guide RNA (pegRNA) (Figure 1a). After the guide RNA binds to the protospacer, the nCas9 creates a single-strand break in the complementary strand, which allows the RT to copy the edited sequence from the pegRNA into the target DNA. This mechanism enables theoretically any types of edits, as RT can be an arbitrary sequence of nucleotides[1].



(a) Prime editing mechanism

Figure 1: (a) shows the mechanism of prime editing. The prime editing guide RNA (pegRNA) binds to the target DNA, and the nCas9 creates a single-strand break in the complementary strand. The reverse transcriptase (RT) then copies the edited sequence from the pegRNA into the target DNA.

More than 6,000 disorders are known to be caused by various types of mutations in the genome, with around 300 new genetic disorders being discovered each year[2]. Up to 90% of these disorder-inducing mutations can be corrected using prime editing[3]. However, its clinical application is significantly limited by its relative low editing efficiency at certain target loci. Empirical methods could be used to identify prime editing guides with high editing efficiency, but they are time-consuming and expensive. Therefore, in-silico prediction tools have garnered significant interest in the scientific community.

In-silico Prime Editing Guide Design Tools

A number of in silico on target prediction tools have been developed to predict the efficiency of prime editing guides.

PRIDICT 2 makes a further step towards improving the prediction accuracy by updating the data preprocessing and model training step. By implementing multitask learning sharing the embedding and bidirectional RNN layers, PRIDICT 2 is able to predict the editing efficiency of prime editing guides with higher accuracy than its predecessor[4].

2 Methods

Data Acquisition and Preprocessing

The dataset used in this study was obtained from the DeepPrime and PRIDICT studies[5, 4, 6], which contains around 220,000 and 110,000 prime editing guides-target pairs, respectively.

A set of functions were implemented to handle the conversion between formats required by different models. For datasets with fold information recorded, the corresponding trained models were preserved in the ensemble. While for datasets without fold information, a 5-fold cross-validation split used by DeepPrime and PRIDICT 2.0 was applied, and the models were retrained on the new folds.

A standardized format was devised

Since the DeepPrime dataset does not contain a wide enough flank sequence of the target site for the PRIDICT model, padding was applied when converting the DeepPrime dataset to the PRIDICT format.

Ensemble Learning

Three ensemble learning approaches were investigated in this study: weighted average, bagging and AdaBoost. The algorithms were implemented in Python, but without the use of Scikit-learn ensemble library, as it does not support having different types of base learners in the ensemble.

However, no significant difference in performance was observed among the three ensemble learning methods ($p > 0.1$, paired t-test across corresponding folds, subsection A.1), possibly due to the high correlation in error between the base models (Add figure here). The weighted average method was chosen for the final implementation due to its simplicity and ease of interpretation.

3 Results

We present PE Ensemble, a web-based application that integrates multiple in-silico prime editing guide design tools. PE Ensemble provides a user-friendly interface for users to design prime editing guides and evaluate their on-target efficiencies. The application also allows users to aggregate the results from multiple tools using weighted mean ensemble learning to improve the overall guide design quality.

4 Discussion

A Appendix

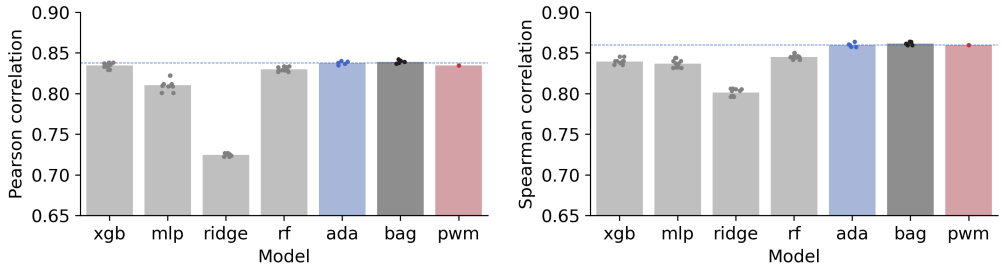
A.1 Ensemble Learning Methods

Three ensemble models in total was investigated in this study: Weighted Average, Bagging and AdaBoost.

As the name suggests, weighted average produces the final prediction by taking the weighted mean of the predictions from the base models, using the performance as measured by the Spearman correlation between the predicted and observed on-target activity as the weight.

Bagging is a method that trains multiple models on different subsets of the training data, hoping to

AdaBoost is a boosting ensemble method that trains multiple models sequentially.



(a) Pearson correlation (b) Spearman correlation

Figure 2: Comparison of ensemble learning methods' performance using (a) Pearson correlation and (b) Spearman correlation.

A.2 Features used by DeepPrime, PRIDICT 1.0/2.0 and conventional ML methods

References

[1] Liu David R. et al. "Search-and-Replace Genome Editing without Double-Strand Breaks or Donor DNA". In: *Nature* 576.7785 (Dec. 5, 2019), pp. 149–157. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-019-1711-4. URL: https://www.nature.com/articles/s41586-019-1711-4 (visited on 02/08/2024).

[2] Gunda Petraitytė, Eglė Preikšaitienė, and Violeta Mikštienė. "Genome Editing in Medicine: Tools and Challenges". In: *Acta Medica Lituanica* 28.2 (2021), pp. 205–219. ISSN: 1392-0138. DOI: 10.15388/Amed.2021.28.2.8. pmid: 35637939. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9133615/ (visited on 06/11/2024).

[3] Ariel Kantor, Michelle McClements, and Robert MacLaren. "CRISPR-Cas9 DNA Base-Editing and Prime-Editing". In: *International Journal of Molecular Sciences* 21.17 (Aug. 28, 2020), p. 6240. ISSN: 1422-0067. DOI: 10.3390/ijms21176240. URL: https://www.mdpi.com/1422-0067/21/17/6240 (visited on 02/08/2024).

[4] Nicolas Mathis et al. "Machine Learning Prediction of Prime Editing Efficiency across Diverse Chromatin Contexts". In: *Nature Biotechnology* (June 21, 2024), pp. 1–8. ISSN: 1546-1696. DOI: 10.1038/s41587-024-02268-2. URL: https://www.nature.com/articles/s41587-024-02268-2 (visited on 06/23/2024).

[5] Nicolas Mathis et al. "Predicting Prime Editing Efficiency and Product Purity by Deep Learning". In: *Nature Biotechnology* 41.8 (Aug. 2023), pp. 1151–1159. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/s41587-022-01613-7. URL: https://www.nature.com/articles/s41587-022-01613-7 (visited on 04/24/2024).

[6] Goosang Yu et al. "Prediction of Efficiencies for Diverse Prime Editing Systems in Multiple Cell Types". In: *Cell* 186.10 (May 2023), 2256–2272.e23. ISSN: 00928674. DOI: 10.1016/j.cell.2023.03.034. URL: https://linkinghub.elsevier.com/retrieve/pii/S0092867423003318 (visited on 05/03/2024).