Evaluation and prediction of the efficiency of prime editor

Goosang Yu

The Graduate School
Yonsei University
Department of Medical Science

Evaluation and prediction of the efficiency of prime editor

A Dissertation Submitted
to the Department of Medical Science
and the Graduate School of Yonsei University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Medical Science

Goosang Yu

January 2025

This certifies that the Dissertation of Goosang Yu is approved

Thesis Supervisor	Hyongbum Henry Kim
Thesis Committee Member	Sung-Rae Cho
Thesis Committee Member	Dong Woo Chae
Thesis Committee Member	Hae-Jeong Park
Thesis Committee Member	Taeyoung Park

The Graduate School
Yonsei University

January 2025

ACKNOWLEDGEMENTS

I feel incredibly fortunate to have been surrounded by such wonderful and supportive people throughout my graduate studies. First and foremost, I am deeply grateful to Professor Hyongbum Kim for his guidance. Professor Kim not only provided invaluable mentorship in my development as a researcher but also encouraged me to grow as a better person. His relentless passion for research and curiosity has been a constant source of inspiration for me.

I would also like to express my sincere thanks to Dr. Hui Kwon Kim, who is now a professor. He has always been someone I aspire to emulate, and I am thankful for his collaboration on my research topic. Without his contribution, I would not have been able to complete this research. I am equally grateful to Jinman Park for significantly enhancing my analytical skills. I believe Jinman gave me the precious opportunity to take a step forward into a new field.

I would also like to thank all my lab colleagues, from whom I received immense help and inspiration. I am especially thankful to Hyunjong Kwak, Dongyoung Kim, and Yusang Jung, who worked with me late into the night, allowing us to collect and analyze a wealth of data. My sincere thanks go to Jihye Park, Hyewon Jang, Ramu Gopalappa, Sang Yeon Seo, Yoo Jin Chang, Sung-Ik Cho, Joo Hye Yeo, Young Gwang Kim, Myungjae Song, Hee Chan Yoo, Yunyoung Choi, Hanahrae Lee, Joongoo Min, and Jinyeong Yang who provided invaluable advice based on their experiences and helped me find my way through challenges. Thanks to Young-hye Kim and Seonmi Park, I was able to work in an excellent research environment that allowed me to focus fully on my studies. I am also grateful to all the other lab members with whom I shared both joyful and difficult times; your companionship has been a great source of strength.

Outside the lab, I would like to thank my parents and sister, who have always supported me and provided peace of mind. Finally, I dedicate this dissertation to my most precious friend and partner, Soyoung, who has supported me wholeheartedly in everything I do and guided me on the right path.

TABLE OF CONTENTS

LIST OF FIGURES iii
LIST OF TABLES iv
ABSTRACT v
1. Introduction........ 1
1.1. The Genetic Blueprint of Life 1
1.2. Technology for editing genomic information........ 1
1.3. Prime editing for precise genome editing 2
1.4. Content and Significance of This Study........ 3
2. Materials and Methods........ 4
2.1. Construction of vectors 4
2.1.1. General molecular techniques 4
2.1.2. Construction of prime editor 2 expressing vector 4
2.1.3. Preparation of PE variant expressing vector 4
2.1.4. Preparation of MLH1dn-GFP expressing vector........ 4
2.1.5. Preparation of MLH1dn expressing empty vector for library cloning 4
2.2. Library design 4
2.2.1. Design of Library-1 and Library-2........ 4
2.2.2. Design of Library-3........ 5
2.2.3. Design of Library-4........ 5
2.2.4. Design of Library-5........ 6
2.2.5. Design of Library-6........ 6
2.3. Plasmid library construction 6
2.3.1. Creation of initial plasmid libraries containing pegRNA-target pairs........ 6
2.3.2. Insertion of sgRNA scaffold........ 7
2.4. Culture and preparation of cell lines 7
2.4.1. Cell culture........ 7
2.4.2. Lentivirus production 7
2.4.3. Preparation of PE expressing cell lines 8
2.5. Evaluation of prime editing efficiency........ 8
2.5.1. High-throughput evaluation using Library1/2 in HEK293T cell line 8
2.5.2. Prime editing in HCT-116 and MDA-MB-231 cell lines 8
2.5.3. High-throughput evaluation of PE2 and its variants using Library-3/4, Library-5, and Library-6 8
2.5.4. High-throughput evaluation of PE4max and NRCH-PE4max using Library-5 and Library-6 9
2.5.5. Prime editing at endogenous sites 9
2.6. pegRNAs design for validation........ 9

2.6.1. Designing pegRNAs to Assess the Impact of Edit Type on Prime Editing Efficiency 9
2.6.2. Rational design of pegRNAs 10
2.7. Analysis of prime editing efficiencies 10
2.7.1. Deep sequencing 10
2.7.2. Calculate the prime editing efficiency 11
2.8. Development of computational models 11
2.8.1. Data preparation for machine learning 11
2.8.2. Generation of conventional machine learning-based models 12
2.8.3. Generation of DeepPE 12
2.8.4. Generation of DeepPrime 12
2.8.5. Addressing imbalance in data representation 13
2.8.6. Generation of DeepPrime-FT 14
2.8.7. Interpretation and feature analysis of tree-based machine learning models 14
2.9. Statistics 14
3. Results 20
3.1. Predicting prime editing efficiency in a limited form 20
3.1.1. High-throughput evaluation of PE2 efficiency 20
3.1.2. The correlation between SpCas9 and PE2 activities 24
3.1.3. Impact of PBS and RTT lengths on PE2 efficiency 26
3.1.4. Factors influencing PE2 efficiency 29
3.1.5. Influence of editing type and position on PE2 efficiency 32
3.1.6. Computational models that predict PE2 efficiencies........ 35
3.2. Predicting prime editing efficiency in various PE systems 41
3.2.1. High-throughput evaluation of PE2 efficiencies using four pairwise libraries . 41
3.2.2. Analyses of factors influencing prime editing efficiency 42
3.2.3. Development of DeepPrime 47
3.2.4. Enhancing prime editing efficiency through optimized pegRNA scaffolds, PAM co-editing, and PE variants 51
3.2.5. Development of DeepPrime-FT 55
3.2.6. Applications of DeepPrime 57
4. Discussion 60
5. Conclusion 61
References 62
Abstract in Korean 65
Publication List 66

LIST OF FIGURES

<Figure 1> Construction of libraries 1 and 2. 20
<Figure 2> Schematic of the position of pegRNA and target in this study. 21
<Figure 3> Schematic representation of the experimental procedure. 22
<Figure 4> High-throughput evaluation of PE2 activity 23
<Figure 5> The correlation between SpCas9 indel frequencies and PE2 efficiencies. 24
<Figure 6> Distribution of indel frequencies caused by Cas9 and efficiencies of PE2. 25
<Figure 7> The impact of PBS and RTT length on PE2 efficiency. 26
<Figure 8> The ideal pairing of PBS and RTT lengths. 27
<Figure 9> The most effective pegRNA for each target. 28
<Figure 10> Relationship between PE2 efficiencies across varying RTT lengths. 28
<Figure 11> Features influencing PE2 efficiency identified by Tree SHAP analysis. 29
<Figure 12> Effect of GC in PBS and RTT on PE2 efficiency. 30
<Figure 13> Effect of GC contents in PBS and PBS lengths on PE2 efficiency. 31
<Figure 14> Effect of the melting temperatures on PE2 efficiency. 32
<Figure 15> Prime editing efficiency depending on edit type 33
<Figure 16> Effect of the type of substitutions on prime editing efficiency. 34
<Figure 17> Impacts of editing type and location on PE2 efficiency. 35
<Figure 18> Development of computational models for predicting PE2 efficiencies. 36
<Figure 19> Benchmark of DeepPE using six datasets 37
<Figure 20> Evaluation of DeepPE using HCT-116 and MDA-MB-231 cells. 38
<Figure 21> Performance comparison of DeepPE and other approaches. 39
<Figure 22> Development of PE_type and PE_position. 40
<Figure 23> High-throughput assessment of prime editing efficiencies. 41
<Figure 24> Effect of PBS, RTT, edit position, and RHA 42
<Figure 25> Effect of PBS and RTT on prime editing efficiency. 43
<Figure 26> Effect of right homology arm length and the edit type on PE2 efficiency. 44
<Figure 27> Impact of edit length of prime editing efficiency. 45
<Figure 28> Impact of last templated nucleotide on prime editing efficiency. 45
<Figure 29> The features associated with various types of prime editing efficiencies. 46
<Figure 30> Factors influencing prime editing efficiency. 47
<Figure 31> Comparing training datasets for DeepPrime and DeepPE. 48
<Figure 32> Development of DeepPrime. 49
<Figure 33> Assessment of DeepPrime with CV-test as the evaluation set. 50
<Figure 34> Assessment of DeepPrime using independent datasets. 51
<Figure 35> Enhancing PE2 efficiencies through optimized scaffold and PAM co-editing. 51
<Figure 36> PAM compatibility analysis for PE2, NRCH-PE2, and NG-PE2. 53
<Figure 37> Comparison of prime editing types. 54
<Figure 38> Comparison of prime editing outcomes with pegRNAs versus epegRNAs. 55

<Figure 39> Development and performance of DeepPrime-FT. 56
<Figure 40> Applications of DeepPrime. 58
<Figure 41> Validation of DeepPrime performance for application. 59

LIST OF TABLES

<Table 1> Optimal hyperparameters for DeepPrime 13
<Table 2> Optimal hyperparameters for the DeepPrime-FT models 15
<Table 3> Primers for molecular cloning 16
<Table 4> Target sequence and pegRNA information 17
<Table 5> Error rates in the plasmid and cell library. 22

ABSTRACT

Evaluation and prediction of the efficiency of prime editor

The prime editor is a highly promising technology capable of inducing all forms of genomic corrections at desired locations, with vast potential for future applications. However, the complexity of designing prime editing guide RNAs (pegRNAs) and the sheer number of possible designs have posed challenges in selecting high-efficiency prime editors. In this study, we measured the efficiency of a large number of pegRNAs using high-throughput screening techniques and identified factors that influence prime editing efficiency. We compared not only the basic PE2 form but also various advanced prime editing systems, including PEmax, PE4, and epegRNA. Additionally, we developed a deep learning model to predict prime editing efficiency based on the data obtained from these experiments. The predictive model we developed demonstrated high correlation in predicting the efficiency of pegRNAs for inducing prime editing in cellular genomes, showing its potential utility in developing gene correction therapies for mutations that cause genetic diseases. This study is expected to play a crucial role in the selection of optimal prime editors for various applications and in advancing research and therapeutic development through gene editing.

Key words : prime editor, high-throughput screening, deep learning, genetic disease

1. Introduction

1.1. The Genetic Blueprint of Life

The information required for various functions essential to the survival of living organisms is encoded within their genomes as DNA sequences. DNA, composed of four types of nucleotides bases (A, T, G, C), carries the instructions for expressing a wide array of proteins through its specific sequences. Organisms rely on these proteins to perform vital functions that enable them to thrive in their respective environments. The genetic code is organized into codons, each consisting of three nucleotide bases, which are translated into amino acids. Humans, too, have a multitude of DNA sequences within the nucleus of each cell, forming a genome that encodes for approximately 20,000 proteins. Each protein plays a specific role, such as acting as an enzyme for phosphorylation, transporting ions across cell membranes, or ensuring proper muscle function. These functions are all dependent on the precise formation of amino acids according to the encoded genetic sequences, leading to proteins with specific molecular structures and functions. However, if a mutation occurs in the genetic sequence, it can alter the amino acids that are translated, affecting the protein's physical and chemical interactions, structure, and function. Certain mutations can have significant impacts on a protein's structure or function. Such mutations can manifest as genetic disorders in living organisms, which, in severe cases, can disrupt vital functions or lead to death.

Just a decade ago, mutations in the human genome were considered irreversible, and we had to accept the resulting diseases. If we were fortunate, we could find drugs that could replace the function of the damaged gene, but more often, that was not the case. It was believed that once a mutation occurred, it couldn't be changed. However, we now live in an era where we can correct the genes of living organisms into desired forms.

1.2. Technology for editing genomic information

To study and treat genetic disorders like those mentioned above, we need technology that allows for precise gene editing at specific genomic sites. Early gene editing techniques relied on DNA-binding proteins that could specifically attach to the target genomic site. For instance, the Zinc Finger (ZF) domain, found in transcription factors, can specifically recognize and bind to 3-nucleotide sequences of DNA. By combining multiple ZF modules, researchers could design systems to target specific DNA sequences that were 15-20 nucleotides long. Similarly, the later-discovered Transcription Activator-Like Effector (TALE) modules can specifically recognize 1-nucleotide sequences of DNA, allowing the creation of systems that target particular DNA sequences. Researchers connected these site-specific DNA binding modules to FokI, developing programmable nucleases known as ZFNs (Zinc Finger Nucleases) or TALENs (TALE Nucleases). These tools were utilized in many studies to elucidate gene functions. However, genome editing techniques using ZF and TALE had significant limitations: designing modules that worked well for each target site was extremely challenging. High-performance gene editing tools require validation under various experimental conditions, but creating precise protein combinations for each condition using ZF or TALE was a complex, time-consuming, and costly process. With the advent of the CRISPR (clustered regularly interspaced short palindromic repeats) system, nearly all genome editing research shifted to using this technology. Unlike complex proteins, the CRISPR gene-editing system uses a simple molecule called RNA. Since RNA is much easier to design and produce compared to proteins, the pace of research in gene correction significantly accelerated.

1.3. Prime editing for precise genome editing

Prime editing enables the introduction of all 12 possible substitutions, small insertions and deletions, as well as combinations of these changes into genomic DNA[1]. The prime editor system consists of a fusion protein made up of Cas9 nickase and reverse transcriptase, along with a prime editing guide RNA (pegRNA). A pegRNA contains a spacer sequence, the tracrRNA scaffold, a reverse transcription template (RTT), and a primer binding site (PBS). This prime editor-pegRNA complex identifies the target genomic site within cells, binds accurately to it, and synthesizes the new genetic information directly at that location. Unlike previous technologies, prime editing's ability to create new genetic information at the precise site of action makes it an exceptionally versatile and powerful genome editing tool.

To date, several prime editors have been developed from PE1 to PE5[1,2]. PE2, an improved version of PE1, is widely used due to its higher efficiency. PE3 combines PE2 with an additional nicking guide RNA (ngRNA) and generally achieves higher editing efficiency, though it also tends to cause more unintended indels compared to PE2. The PE4 and PE5 systems enhance the efficiency of prime editing by inhibiting the mismatch repair (MMR) system in host cells. When prime editing induces a new cDNA to replace the existing DNA and cause a mutation, the cell recognizes this as a type of DNA damage and initiates a repair mechanism. If this repair process is too efficient, the area targeted by prime editing may revert to its original state, preventing the intended prime editing from taking place. To inhibit this, genes such as *MLH1*, *MSH2*, and *MSH6* are knocked out or knocked down. A representative method is to deliver a dominant-negative form of MLH1 (MLH1dn) along with the prime editor. The PEmax is further enhancements of their respective predecessors[2].

Prime editing offers two major advantages over other existing CRISPR systems. The first advantage is its safety. If a genome editing tool malfunctions and damages healthy genes in our bodies, it could lead to unexpected side effects. Therefore, the safety of genome editing tools is of paramount importance. Early CRISPR genome editing tools completely cut the gene, which raised safety concerns among researchers. However, prime editing is a technology that can correct genes without fully cutting them. Due to its enhanced safety, it is anticipated that prime editing could be used in future gene therapies. The second advantage is its significantly higher versatility compared to existing genome editing tools. Until now, all genome editing tools had limitations on the types of changes they could make to genes. Some tools could only cut DNA, while others could only recognize a single type of nucleotide. In contrast, prime editing has no such limitations. It allows for virtually any type of gene correction using a single genome editing tool.

One of the primary challenges in prime editing is designing effective pegRNAs. The vast number of possible pegRNA designs makes it difficult to identify the most efficient one. For a single target edit, there could be hundreds or even thousands of potential pegRNA designs, necessitating extensive experimentation to determine the best option[1,3,4]. Moreover, there is a shortage of comprehensive data on prime editing efficiency across different cell lines, and there has been limited reporting on the effectiveness of newer prime editing systems like PEmax, PE4, and epegRNA.

1.4. Content and Significance of This Study

This study utilized high-throughput screening techniques to measure the efficiency of prime editing on a large scale. Through this approach, we were able to generate a dataset of prime editing efficiency for hundreds of thousands of pegRNAs. Through this approach, we were able to understand the factors that determine prime editing efficiency based on the characteristics of pegRNAs. This knowledge has revealed that the mechanism operates differently from the conventional CRISPR system using sgRNAs and has laid the foundation for effectively utilizing prime editing in the future. Using this dataset, we developed a deep learning model that can predict prime editing efficiency. This model accurately predicts the efficiency of a given pegRNA, allowing researchers to select the optimal pegRNA without manually experimenting with numerous options. Notably, the model provides various specialized versions tailored to different PE systems and cell lines, enabling researchers to use it according to their specific needs. This model, offering a range of options, is expected to become an essential tool for future applications of prime editing.

2. Materials and Methods

2.1. Construction of vectors

2.1.1. General molecular techniques

For obtaining plasmid vectors or PCR products cut by restriction enzymes at precise sizes, electrophoresis was performed on a 1-2% agarose gel, followed by purification using the MEGAquick-spin™ Plus Total Fragment DNA Purification Kit (iNtRON Biotechnology). In molecular cloning, the DNA fragments used as backbone vectors were treated with Quick CIP (NEB) at 37°C for 10 minutes after digestion with restriction enzymes.

2.1.2. Construction of prime editor 2 expressing vector

To create a lentiviral vector expressing PE2, LentiCas9-Blast (Addgene #52962) was digested with AgeI and BamHI and used as the backbone vector. The resulting linearized plasmid was then gel purified. The coding sequence of Prime Editor 2 (PE2) was amplified by PCR from pCMV-PE2 (Addgene #132775) and gel purified. The PCR products were then assembled with the linearized LentiCas9-Blast using the NEBuilder HiFi DNA assembly kit (NEB), resulting in pLenti-PE2-BSD (Addgene #161514).

2.1.3. Preparation of PE variant expressing vector

We constructed lentiviral plasmid vectors expressing PE variants using SpCas9, SpCas9-NG-PE2, and NRCH-Cas9. The cloning process was conducted similarly to the procedure used for creating pLenti-PE2-BSD. The primers used for cloning are described in **Table 3**.

2.1.4. Preparation of MLH1dn-GFP expressing vector

The pEGIP[5] was digested with EcoRV and used as the backbone vector. The coding sequences of MLH1dn and eGFP were amplified by PCR from pEF1a-hMLH1dn and pEGIP, respectively, and gel purified. The backbone vector and PCR products were assembled using Gibson assembly, resulting in the construction of the pLenti-EF1a-hMLH1dn-eGFP plasmid (Addgene #191104).

2.1.5. Preparation of MLH1dn expressing empty vector for library cloning

The pLenti-gRNA_Puro plasmid was linearized with the BsiWI restriction enzyme and subsequently assembled with the MLH1dn coding sequence, which was PCR-amplified from the pEF1a-hMLH1dn plasmid, using Gibson assembly. To prevent early transcription termination, a mutation was introduced at position I34 of hMLH1dn, disrupting an AATAAA signal sequence[6].

2.2. Library design

2.2.1. Design of Library-1 and Library-2

Library-1 was designed to examine the effect of PBS and RTT lengths on prime editing efficiency. It consists of pegRNAs with 24 PBS-RTT combinations for a single guide-target sequence. The 24 PBS-RTT combinations were made up of six PBS lengths (7, 9, 11, 13, 15, and

17nt) and four RTT lengths (10, 12, 15, and 20nt). The intended prime editing in the RTT was fixed as a +5 G to C substitution. To measure prime editing efficiency across various target sequences, we selected 2,000 guide-target sequences from a previous study where Cas9 efficiency had been measured[7], and for each target, we designed 24 pegRNA combinations, resulting in a total of 48,000 pegRNA-target pairs.

Additionally, we created a second library (Library-2) to evaluate how editing position, type, and length affect PE2 efficiencies. From the 2,000 sequences in Library-1, we randomly chose 200 target sequences and designed 34 distinct templates for each. The templates were structured as follows:

i) The effect of editing position (11 RTTs): RTTs were designed to introduce transversion mutations at positions +1, +2, ..., +8, +9, +11, and +14 from the nicking site, with PBS and RTT lengths fixed at 13 and 20 nucleotides, respectively.

ii) The effect of editing type and length (14 RTTs): RTTs were designed for insertions (sequences A, G, C, T, AG, AGGAA, and AGGAATCATG), deletions (1, 2, 5, and 10 nucleotides), and single nucleotide substitutions (all possible 1-nucleotide substitutions) at the +1 position from the nicking site. The PBS and right homology arm of the RTT were set at 13 and 14 nucleotides, respectively.

iii) The effect of PAM editing (9 RTTs): RTTs were designed to introduce 2-bp transversion mutations at various positions (e.g., +1 & +2, +1 & +5, +1 & +10, etc.), with PBS and RTT lengths fixed at 13 and 16 nucleotides, respectively.

Moreover, we included 36 pairs of pegRNAs and target sequences from the initial prime editing study[1], each tagged with five unique barcodes. This set was used to compare the prime editing efficiencies between integrated sequences and endogenous sites. In total, Libraries 1 and 2 encompassed 54,836 pegRNA-target pairs: 48,000 pairs from Library-1, 6,800 pairs from Library-2, and 36 pairs from the original prime editing study.

2.2.2. Design of Library-3

To assess the factors influencing prime editing efficiency, we created a library called Library-3, comprising 47,839 pegRNA and target sequence pairs. We selected 40 seed target sequences (20-nucleotide regions near the PAM) from Library-1 that showed high SpCas9-induced indel frequencies: 20 sequences where paired sgRNAs resulted in 70% - 75% indel frequencies, and another 20 sequences with 50% - 55% indel frequencies.

For each seed target sequence, we designed 74-nt target sequences (**Figure 2)** and pegRNAs with varying PBS and RTT lengths, as well as different editing positions, lengths, and types. We excluded 81 oligonucleotides containing the BsmBI cut site. The pegRNA-target pairs were categorized into seven groups as detailed below. Some pairs were assessed but omitted from the final analysis, with all details provided in Table S1.

2.2.3. Design of Library-4

To measure a wide range of prime editing efficiencies and obtain comprehensive data, we created a library containing 600,000 pegRNA-target pairs. We identified variants with 1-3 bp mutations from ClinVar and collected all possible spacers surrounding these mutations. Subsequently, we generated all possible pegRNAs for each mutation with RTT lengths up to 40 nucleotides. From these, we randomly selected eight pegRNAs per target. Detailed information can be found in **Figure 23**.

2.2.4. Design of Library-5

To test the editing efficiencies of PE variants and pegRNAs with standard or optimized[9] scaffolds across different cell lines, we created Library-5, comprising 6,000 pegRNA and target sequence pairs. First, we selected 2,990 pairs (1,495 for disease modeling and 1,495 for therapeutic purposes) from the CV-train dataset. Half were randomly chosen, while the other half were proportionally selected from editing efficiency ranges of 0%, 0 to 1%, 1 to 5%, and over 5%. Additionally, we generated 2,990 more pairs by randomly altering the NGG PAM sequence to NNN to assess the PAM compatibility of PE variants. Lastly, 20 pegRNAs with the highest editing efficiencies from our previous study were included at 5-fold redundancy (5 x 4 pegRNAs) as positive controls.

2.2.5. Design of Library-6

To evaluate the editing efficiencies of engineered pegRNAs (epegRNAs), we designed Library-6, consisting of 6,000 epegRNA sequences paired with their corresponding target sequences. Each epegRNA included an 8-nt linker and a tevopreQ1 structural motif at the 3' end. Aside from the added linkers, the 6,000 epegRNAs were identical to the pegRNAs in Library-5. The 8-nt linker sequences were specifically designed using the pegRNA Linker Identification Tool (pegLIT)[10].

2.3. Plasmid library construction

We constructed a plasmid library from an oligonucleotide pool containing pegRNA and target sequences, which was synthesized by Twist Bioscience (San Francisco, CA). Each oligonucleotide was designed with the following elements: a 19-nt guide sequence, a BsmBI restriction site #1, a 10-15 nt spacer sequence (barcode stuffer), a second BsmBI restriction site, the RTT sequence, the PBS sequence, a poly-T sequence, a 14-18 nt identification barcode, and a corresponding target sequence including a PAM and an RTT binding region. The spacer sequence was included to minimize the risk of template switching during PCR amplification, but was later removed by BsmBI digestion, while the identification barcode enabled precise identification of pegRNAs after deep sequencing[7,11]. Oligonucleotides containing unintended BsmBI restriction sites were excluded. The construction of the plasmid libraries, which contained paired pegRNA-encoding and corresponding target sequences, was performed using a two-step cloning process. This involved cutting with restriction enzymes followed by ligation and Gibson assembly. The method was adapted and modified from a previously published approach[12] to ensure the integrity of paired guide and target sequences during PCR amplification of oligonucleotides[13].

2.3.1. Creation of initial plasmid libraries containing pegRNA-target pairs

The oligonucleotide pool was first amplified by PCR for 15 cycles, followed by gel purification. Next, the Lenti_gRNA-Puro (Addgene #84752) and Lenti_gRNA-Puro-hMLHdn plasmids were digested with BsmBI at 55°C, and the resulting linearized vectors were treated with Quick CIP, then gel-purified. The amplified oligonucleotide pool was then inserted into the linearized vectors using Gibson assembly. The assembled plasmids were concentrated through isopropanol precipitation and transformed into electrocompetent cells (Lucigen) via a MicroPulser

(Bio-Rad). The transformation mixture was incubated with SOC medium at 37°C for an hour, followed by spreading onto LB agar plates with 50 μg/ml carbenicillin. The library coverage was assessed by plating small portions of the culture (0.1, 0.01, and 0.001 μl) separately. Colonies were harvested, and plasmids were extracted using a QIAGEN Plasmid Maxi kit (QIAGEN). The calculated coverages of these initial plasmid libraries were 113X, 986X, 2,210X, and 500X the number of oligonucleotides in each library for Library1/2, Library-3/4, Library-5-PE2, and Library-5-PE4, respectively.

2.3.2. Insertion of sgRNA scaffold

The plasmid libraries produced in Step I were digested with BsmBI for at least 6 hours, treated with Quick CIP at 37°C for 10 minutes, and then underwent size-selection on a 0.6% agarose gel followed by gel purification. Insert DNA fragments containing either a conventional or optimized scaffold sequence were PCR-amplified from the lentiGuide-Puro plasmid (Addgene #52963) or chemically synthesized oligonucleotides (IDT) using Phusion DNA polymerase with primers containing BsmBI recognition sites. The resulting amplicons were cloned into T-blunt vectors. The T-blunt vectors containing scaffold sequence were digested with BsmBI, and the conventional or optimized scaffold sequences with proper 5′- and 3′-overhangs were gel-purified. After purification, the scaffold insert sequence was ligated into the initial plasmid library vector using T4 ligase (Enzynomics) at 16°C for 3 hours at a 1:10 vector-to-insert ratio (w/w). The ligated products were purified via isopropanol precipitation and transformed into Endura electrocompetent cells (Lucigen). Colonies were collected, and the final plasmid library was obtained by plasmid extraction using a QIAGEN Plasmid Maxi kit.

2.4. Culture and preparation of cell lines

2.4.1. Cell culture

We cultured HEK293T, HeLa, HCT-116, DLD-1, A-549, and NIH-3T3 cells in DMEM, which was supplemented with 10% FBS. MDA-MB-231 cells were grown in RPMI 1640 medium with HEPES (Thermo Fisher Scientific) and 10% FBS. All cell cultures were kept at a temperature of 37°C with 5% CO_2, ensuring they remained below 80% confluency. The cells were passaged every 3-4 days. Cells that were transduced with lentivirus encoding PE2 or PE variants were selected using 10 μg/ml blasticidin S, while those transduced with pairwise libraries were selected using 1 μg/ml puromycin.

2.4.2. Lentivirus production

HEK293T cells were seeded onto 150-mm dishes in DMEM medium at a density of 55,000 cells/cm^2. After 15 hours, the medium was replaced with fresh medium containing 25 μM chloroquine diphosphate and incubated for up to 5 hours. For lentivirus production, a plasmid mix composed of the plasmid encoding protein of interest, psPAX2, and pMD2.G in a 4:3:1 weight ratio was transfected into the HEK293T cells using PEI MAX (Polysciences). After 15 hours, the medium was refreshed, and 48 hours later, the lentivirus-containing supernatant was harvested, filtered through a Millex-HV 0.45-μm low protein-binding membrane (Millipore), aliquoted, and stored at -80°C. The viral titer was estimated by transducing cells with serial dilutions of the viral aliquot in the presence of polybrene (8 μg/ml) and selecting transduced cells with puromycin. Selected cells were counted to calculate the titer, following a previously established method[14].

2.4.3. Preparation of PE expressing cell lines

To create cell lines expressing PE2 or its variants, HEK293T, HCT-116, MDA-MB-231, HeLa, DLD-1, A-549, and NIH-3T3 cells were transduced with lentivirus encoding PE2 or its variants at a multiplicity of infection (MOI) of 0.3, along with 0.8 µg/mL of polybrene. Within 24-48 hours post-transduction, untransduced cells were eliminated using 10 µg/mL blasticidin S. The successful lentiviral delivery of PE2- or PE variant-encoding sequences was verified by PCR and Sanger sequencing. All cell lines were maintained in culture with 10 µg/ml of blasticidin S.

2.5. Evaluation of prime editing efficiency

2.5.1. High-throughput evaluation using Library1/2 in HEK293T cell line

HEK293T cells were seeded in nine 150-mm dishes at a density of 1.6×10^7 cells per dish and incubated overnight in preparation for lentiviral transduction. The lentiviral library was transduced at a multiplicity of infection (MOI) of 0.3, ensuring over 500× coverage relative to the number of initial oligonucleotides. After overnight incubation, untransduced cells were eliminated by maintaining the culture in 2 µg/ml puromycin for 5 days. To maintain the diversity of the library, the cell count was kept above 3.0×10^7 cells throughout the experiment. Then, 3.0×10^7 cells (from three 150-mm culture dishes) expressing pegRNAs from Library1/2 were transfected with the pLenti-PE2-BSD plasmid (80 µg per dish) using 80 µl Lipofectamine 2000 (Thermo Fisher Scientific) according to the manufacturer's instructions. Six hours after transfection, the medium was replaced with DMEM supplemented with 10% fetal bovine serum and 20 µg/ml blasticidin S (InvivoGen). The cells were harvested 4.8 days post-transfection.

2.5.2. Prime editing in HCT-116 and MDA-MB-231 cell lines

Seventy-five plasmids, each carrying a pegRNA-target sequence pair, were randomly selected from plasmid Library-1 and identified by Sanger sequencing. From this pool, a small lentiviral library was generated as described earlier. PE2-expressing HCT-116 and MDA-MB-231 cells were seeded in 6-well plates at a density of 2.0×10^5 cells per well, incubated overnight, and then transduced with the lentiviral library. After overnight incubation, the medium was replaced with DMEM containing 1 µg/ml puromycin and 10 µg/ml blasticidin S for HCT-116 cells, or with RPMI containing 2 µg/ml puromycin and 10 µg/ml blasticidin S for MDA-MB-231 cells. The cells were harvested and analyzed 4.5 days post-transduction.

2.5.3. High-throughput evaluation of PE2 and its variants using Library-3/4, Library-5, and Library-6

Twenty-four hours before introducing the lentiviral libraries, PE-expressing cells were plated in 150-mm culture dishes. The cells were then transduced with a lentivirus containing one of the libraries at an MOI of 0.5 in the presence of 8 µg/ml polybrene. For the Library-3/4, a total of 6×10^8 cells were used to ensure 500× coverage of the pegRNA-target sequence pairs, while 2.4×10^6 cells were utilized to achieve 2,000× coverage for the Library-5 and Library-6. After transduction, the culture medium was replaced with DMEM containing 10% FBS and 2 µg/ml puromycin 12 hours later. The cells were harvested 7 days after transduction for Library-5 and Library-6 and 8 days after for Library-3/4.

2.5.4. High-throughput evaluation of PE4max and NRCH-PE4max using Library-5 and Library-6

For an in-depth analysis of the editing capabilities of PE4max and NRCH-PE4max, we carried out high-throughput evaluations in HEK293T cells utilizing Library-5. The hMLH1dn element was introduced via transient transfection using the pLenti-EF1a-hMLH1dn-eGFP plasmid (Addgene #191104). Specifically, 3.6×10^7 PE2max-expressing HEK293T cells were plated across three 150-mm dishes and transfected with 30 µg of the pLenti-hMLH1dn-eGFP plasmid using PEI. After 12 hours, these cells were transduced with Library-5 at an MOI of 0.5 in the presence of 8 µg/ml polybrene. Following puromycin selection, cells were collected 7 days after the library transduction. For the assessment of PE4max and NRCH-PE4max editing efficiencies in HEK293T, DLD-1, A-549, and NIH-3T3 cells, using either Library-5 or Library-6, hMLH1dn was delivered using a lentiviral vector. Cells were then transduced with either Library-5-hMLH1dn or Library-6-hMLH1dn at an MOI of 0.5 with 8 µg/ml polybrene. After 48 hours, the medium was refreshed with one containing puromycin, and the cells were harvested 7 days post-transduction for deep sequencing analysis. Details of the primers used for cloning can be found in **Table 3**.

2.5.5. Prime editing at endogenous sites

To confirm the results from high-throughput experiments conducted using Library-1/2, 33 pegRNA-encoding plasmids were randomly selected from the library. For transfection, HEK293T cells were plated on 48-well plates at a density of 5.0×10^4 or 1.0×10^5 cells per well, 16-18 hours before transfection. Cells were then transfected with a mixture containing 75 ng of PE2-encoding plasmid (pLenti-PE2-BSD) and 25 ng of pegRNA-encoding plasmid per 1.0×10^4 cells, using 1 µl of Lipofectamine 2000 or TransIT-2020 transfection reagent per 1,000 ng of DNA, following the manufacturer's guidelines. After overnight incubation, the medium was replaced with DMEM supplemented with 2 µg/ml puromycin. The cells were harvested either 4.5 days (for Endo-BR1 and Endo-BR2) or 7 days (for Endo-BR3) post-transfection. To evaluate prime editing efficiencies at endogenous genomic loci, sequences for 77 pegRNAs with an optimized scaffold were cloned into pU6-pegRNA-GG-acceptor (Addgene #132777, Table S4). For the analysis of the edit type (Figure S2E) and editing efficiencies in BRCA2 (Figure 6F), HEK293T cells were seeded on 48-well plates at a density of 5.0×10^4 cells per well, 22 hours prior to transfection. These cells were then transfected with a mix of 300 ng of PE2max-encoding plasmid (pLenti-EF1a-PE2max-BSD) and 100 ng of pegRNA-encoding plasmid, using 0.8 µl Lipofectamine 3000 and 0.6 µl P3000 reagent according to the manufacturer's instructions. The culture medium was replaced with DMEM containing 2 mg/ml puromycin after overnight incubation. Cells were harvested 7 days after transfection.

2.6. pegRNAs design for validation

2.6.1. Designing pegRNAs to Assess the Impact of Edit Type on Prime Editing Efficiency

To investigate how different types of edits affect prime editing efficiency at endogenous sites, we selected 13 pegRNAs known for their high efficiency (ranging from 15% to 35%) in inducing pathogenic or likely pathogenic 1-bp substitutions from the CV-train dataset. For each of these pegRNAs, two additional variants were designed with identical reverse homology arm (RHA) lengths to introduce either a 1-bp insertion or deletion at the same target site.

2.6.2. Rational design of pegRNAs

The rational design of pegRNA for creating pegRNAs to model ClinVar mutations began with identifying an NGG PAM sequence within a ±60-nucleotide range of the target edit site, followed by generating all possible pegRNAs using spacers derived from this PAM, with RTT lengths capped at 40 nucleotides. The next step involved evaluating these pegRNAs using the DeepSpCas9 scoring system, where those with scores of 30 or higher were selected. In cases where no pegRNAs met this criterion, the spacer with the highest DeepSpCas9 score among all options was used to generate pegRNAs. Following this, pegRNAs were filtered to include only those with intended edit positions at +5 or +6; pegRNAs not meeting this criterion were excluded. Among the filtered pegRNAs, the one with the shortest RTT ending with a 'C' nucleotide and an RHA length of at least 7 nucleotides was selected. If no pegRNAs had an RHA length of 7 or more, the one with the longest RHA was chosen instead. Finally, depending on the length of the RTT from the selected pegRNA, the PBS length was adjusted accordingly: if the RTT was 12 nucleotides or shorter, the PBS length was set to 11 nucleotides, while if the RTT was 13 nucleotides or longer, the PBS length was set to 12 nucleotides.

2.7. Analysis of prime editing efficiencies

2.7.1. Deep sequencing

To assess the efficiency of prime editing in high-throughput experiments, cells were collected, and their genomic DNA was extracted.

For Library-1/2, we initiated the process with 400 µg of genomic DNA, which, considering 10 µg of DNA per 10^6 cells, provided a coverage exceeding 700× over the library. We conducted 80 separate 50-µl PCR reactions, each containing 5 µg of genomic DNA. The resulting PCR products were pooled and gel-purified. Subsequently, 100 ng of this purified DNA was subjected to a second PCR using primers equipped with Illumina adaptors and barcode sequences (see Supplementary Table 6). For analyzing PE2 efficiencies at endogenous sites, we performed an initial PCR in a 40-µl volume containing 200 ng of genomic DNA per sample. A second PCR was carried out with 20 ng of the purified first PCR product in a 30-µl reaction to add the Illumina adaptors and barcodes. After gel purification, the amplicons were sequenced using Illumina HiSeq or MiniSeq platforms. The primers used are listed in Supplementary Table 6.

For the Library-3 and Library-4 experiments, we used 5,760 µg of genomic DNA from HEK293T cells for PCR, which provided coverage over 960×, assuming 10 mg of DNA per 10^6 cells. A total of 576 individual 50-µl PCR reactions were performed using 10 mg of genomic DNA, 200 nM of each primer set, and 25 µL of 2X Taq PCR Smart mix (SolGent) under the following conditions: initial denaturation at 95°C for 10 min, followed by 22 cycles of 95°C for 30 s, 60°C for 30 s, 72°C for 40 s, with a final extension at 72°C for 5 min. For achieving over 2,000X coverage in Library-5 and Library-6, PCR was conducted using at least 120 mg and 1 mg of genomic DNA for each sample, respectively. For Library-5, 24 independent 50-µl PCR reactions were performed using 5 mg of genomic DNA, while for Library-6, 200 independent reactions were conducted. Each reaction contained 500 nM of primers, 200 mM of dNTPs, DNA polymerase, and a reaction buffer. We used Q5 High-Fidelity DNA polymerase for Library-5 and Phusion DNA Polymerase for Library-6. After PCR amplification, products were pooled, gel-purified, and sequenced on Illumina NovaSeq.

For determining prime editing efficiencies at endogenous sites, cells were lysed in 100 µL of lysis buffer (10 mM Tris–HCl, 0.05% SDS, 025 mg/ml proteinase K, and pH 7) at 37°C for 1 hour. To deactivate the enzymes, the lysate was incubated at 80°C for 15 minutes. The first round of PCR was performed in a 50-µL volume containing 25 µL of 2X Taq PCR Smart mix, 5 µL of cell lysate, and 200 nM of the primer set under the following conditions: initial denaturation at 95°C for 1 min, followed by 35 cycles of 95°C for 30 s, 60°C for 30 s, and 72°C for 30 s, with a final extension at 72°C for 2 minutes. The second PCR, aimed at adding Illumina adaptors, was performed using 0.5 µL of the first PCR product in a 50-µL reaction under the same conditions, but with 12 cycles. Gel purification of the PCR products was followed by deep sequencing. The pegRNA-encoding regions, barcodes, and target sequences were amplified from genomic DNA using PCR primers that included Illumina adapters and unique i7 and i5 barcodes (as listed in Table S4).

2.7.2. Calculate the prime editing efficiency

To process and analyze the deep sequencing data, we modified existing Python scripts. The pegRNA and target sequence pairs were identified using a 22-nt sequence (comprising an 18-nt barcode and a 4-nt sequence upstream of the barcode) for Library-1/2, and a 36-nt sequence (consisting of a 12-nt PBS domain of the pegRNA, an 18-nt barcode, and a 6-nt sequence that includes 4-nt upstream and 2-nt of the target sequence) for other libraries. Reads containing the intended edits without unintended mutations within the target sequence were considered to represent PE2-induced mutations. To correct for background prime editing frequency, observed prime editing frequencies were normalized by subtracting the background frequency (determined in the absence of PE2) using the formula:

$$= \frac{\begin{gathered}\mathit{Read\ counts\ with\ intended\ edit\ and\ specified\ barcode} - \\ (\mathit{Total\ read\ counts\ with\ specified\ barcode} \times \mathit{background\ prime\ editing\ frequency}) \div 100\end{gathered}}{\begin{gathered}\mathit{Total\ read\ counts\ with\ specified\ barcode} - \\ (\mathit{Total\ read\ counts\ with\ specified\ barcode} \times \mathit{background\ prime\ editing\ frequency}) \div 100\end{gathered}} \times 100$$

To improve the accuracy of our analysis, we applied several filters to the deep sequencing data. PegRNA and target sequence pairs with deep sequencing read counts below 200 or with background editing frequencies above 5% were excluded. We implemented steps to reduce noise from random errors by first categorizing sequencing reads as wild type (WT) or edited based on the barcode. Reads with random errors in WT or variants other than the intended edit in edited sequences were removed. Pairs with fewer than 200 reads in total or those with background frequencies above 5% were also excluded. Finally, the read counts of replicates were pooled based on barcodes, and prime editing efficiency data were obtained, following previously established protocols[7,11,15-17] for Library-4, -Profiling, -Small, and -epegRNA.

2.8. Development of computational models

2.8.1. Data preparation for machine learning

To extract features from pegRNAs and their corresponding target sequences, we employed tools such as Biopython (version 1.79)[18], the ViennaRNA package (version 2.5.0) [19], and DeepSpCas9[7] to compute metrics like GC count, GC content, melting temperature (Tm), minimum free energy, and the DeepSpCas9 score. These metrics were combined with other sequence-derived characteristics, including the lengths of PBS, RTT, RT-PBS, and RHA, along with the type, position, and length of the intended edits, using custom Python scripts. We divided each dataset into training and testing subsets using stratified random sampling, ensuring no overlap of target sequences between the training and test datasets used for model development.

2.8.2. Generation of conventional machine learning-based models

For comparing machine learning models trained on the CV-test dataset, we utilized Scikit-learn[20], Pycaret[21], and other specialized machine learning pipeline packages to generate various regression models. During model training, we extracted both position-dependent and position-independent nucleotide and dinucleotide features from the wide target, PBS, and RTT sequences. Additionally, the z-score normalized Tm, GC count, GC content, minimum free energy, and DeepSpCas9 score were included. This comprehensive set of 2,956 features was used to train the conventional machine learning models. We optimized model parameters through a random grid search of 150 models. To evaluate performance, we used the Spearman correlation between the observed and predicted efficiencies, conducted 5-fold cross-validation, and compared Spearman's correlation across each fold.

2.8.3. Generation of DeepPE

DeepPE, a deep learning-based computational model, predicts the optimal combination of PBS and RTT lengths required to introduce a G-to-C transversion mutation at position +5 from the nicking site. The model was trained on a dataset that included prime editing efficiencies induced by PE2 and 38,692 pegRNAs. The training data encompassed 47-nt-wide target sequences, RTT plus PBS sequences (17-37 nt), and 20 additional features, such as melting temperature, GC count, GC content, and minimum self-folding free energy. Nucleotide sequences were encoded into four-dimensional binary matrices via one-hot encoding. DeepPE was developed using a convolutional layer coupled with a fully connected layer. The convolutional layer extracted two embedding vectors from the wide target sequences and RTT plus PBS sequences using ten filters of 3 nt in length. These embedding vectors were then combined with the 20 biological features. To maintain local information, the pooling layer was omitted, and a deep reinforcement learning algorithm was implemented[22]. The fully connected layer, containing 1,000 units, applied the ReLU activation function to multiply the vectors. The regression output layer performed a linear transformation of the outputs, predicting PE2 efficiency scores. After testing nine different models with varying hyperparameters (filters: 10, 20, 40; units: 200, 500, 1,000), we selected the model with the highest Spearman correlation between experimentally observed and predicted activity levels during 5-fold cross-validation. Dropout was used with a rate of 0.3 to prevent overfitting, and mean squared error served as the objective function. The model was optimized using the Adam optimizer with a learning rate of 0.001 and was implemented using TensorFlow[23]. For predicting PE2 efficiencies across various editing types and positions, we used a multilayer perceptron (MLP) instead of a convolutional neural network, as the latter yielded poor performance in initial trials. We conducted cross-validation to select from 18 MLP models with architectures and parameter counts similar to DeepPE but without convolutions. The hyperparameter configurations included the number of layers (2 or 3), the number of units per hidden layer (1,000, 200, 50 in the first layer, 50 in the second), the dropout rate, the learning rate (0.01, 0.001, 0.0001), and the ReLU activation function.

2.8.4. Generation of DeepPrime

DeepPrime is a deep learning-based model designed to predict the efficiencies of prime editing using pegRNAs with varying PBS and RTT lengths, aimed at introducing 1- to 3-bp at positions +1 to +30. Developed using PyTorch, DeepPrime processes input sequences consisting of unedited and prime-edited sequences. The input sequence processing module includes four convolutional layers followed by a gated recurrent unit (GRU) layer. Each convolutional layer used a kernel width of 3 and a stride of 1, with zero padding to preserve sequence length. The four convolutional layers contained 128, 108, 108, and 128 channels, respectively, with average pooling applied after the 2nd, 3rd, and 4th convolution operations. A pooling kernel size of 2 and a stride of 2 were utilized. Input sequences, one-hot encoded into four channels (A, T, G, C), were fed into the convolutional module. The output from the final convolutional layer was then passed through a bidirectional GRU to capture long-distance interactions and retain positional information of the input sequences[24]. The GRU's hidden state was 128-dimensional, and the final hidden state was projected linearly into a 12-dimensional vector. In addition, DeepPrime includes a separate four-layer perceptron module to analyze the physicochemical properties of pegRNAs and target sequences, referred to as "biofeatures," which includes metrics like Tm, GC count, GC content, the minimum self-folding free energy of the guide and RTT-PBS, and the DeepSpCas9 score. These biofeatures were transformed into a 128-dimensional latent vector, which was concatenated with the 12-dimensional GRU output to form a 140-dimensional vector. This vector was then projected linearly to produce a single regression output through softplus activation. We applied GELU activations[25] for the convolutional layers and ReLU for other layers, and batch normalization was used after each convolution and before the final linear projection to accelerate model training[26]. The model's hyperparameters—including hidden dimensions, the number of layers, kernel sizes, strides, channels, learning rate, and number of epochs—were optimized using Bayesian search with Optuna. The AdamW optimizer[27] with cosine annealing learning rate scheduling and warm restarts was employed for model training. We independently trained five models with different random seeds and averaged their predictions to achieve the final prediction. The optimal hyperparameters for DeepPrime are as follows:

Table 1. Optimal hyperparameters for DeepPrime

Optimizer				Scheduler		Model	
Batch size	Learning rate	Weight decay	Number of epochs	T_0	T_mult	Hidden size	Number of models
2048	5.00E-03	5.00E-02	10	10	1	128	5

2.8.5. Addressing imbalance in data representation

The training dataset exhibited a significant skew towards data points with low prime editing (PE) efficiencies, which hindered the model's ability to adequately represent cases with high PE efficiencies. To mitigate this imbalance, we employed a strategy that involved amplifying the loss for the underrepresented high-efficiency data. This was achieved by introducing a high offset, allowing the model to become more responsive to these rarer instances. The amplification (μ) was derived by simulating the reciprocal square root of the data distribution using a straightforward function, as shown below.

$$\mu = \min(\exp(6(\log(x+1) - 3) + 1), 5)$$
$$= \min\left(\frac{(x+1)^6}{\exp(18)} + 1, 5\right)$$

where, x indicates the measured prime editing efficiency (%).

Moreover, to specifically address the imbalance among different types of edits, we applied weight adjustments to the loss functions for insertions and deletions, which were less common than substitutions. Specifically, the losses associated with insertions and deletions were scaled by factors of 0.7 and 0.6, respectively. These weights were determined through analysis using 5-fold cross-validation.

2.8.6. Generation of DeepPrime-FT

For the enhancement of the base DeepPrime model, we utilized transfer learning techniques. We developed eighteen models by fine-tuning the base model with eighteen distinct datasets that represented prime editing efficiencies across eight different prime editing systems, including two scaffold sequence types in seven different cell types. The final weights from DeepPrime served as the starting point for these models. We maintained a batch size of 512 for all the fine-tuned models, while the optimal hyperparameters—including learning rates, weight decay, and epoch counts—were identified using Optuna. **Table 2** provides the optimal hyperparameters for the eighteen models.

2.8.7. Interpretation and feature analysis of tree-based machine learning models

To assess the impact of individual features on the prediction model for PE efficiency, we calculated Shapley values using the SHAP (Shapley Additive exPlanations, 0.40.0) Python package[28]. For data derived from Library-1/2, we extracted features and trained XGBoost models, employing the best hyperparameter configurations identified through 5-fold cross-validation as previously described. Similarly, for the Library-4 dataset, we trained LightGBM models, both with and without the exclusion of features with correlation of 0.7 or higher. To understand the global impact of each feature on the prediction models, we used SHAP values across the entire training dataset and compared Shapley values from each feature to assess local interaction effects.

2.9. Statistics

To evaluate the variation in prime editing efficiencies across different experiments utilizing distinct pegRNAs, a one-way ANOVA followed by a two-sided Tukey's post hoc test was performed. The Spearman correlation between prediction scores from various models was assessed using a two-sided Steiger's test, which is tailored for comparing two dependent correlation derived from the same dataset. We determined statistical significance using tools such as GraphPad Prism 8, PASW Statistics (version 17.0, IBM), and Microsoft Excel (version 2302, Microsoft Corporation). For the high-throughput assessment of PE2 efficiency using pairwise libraries, sequencing read counts from two independent replicates performed by different experimenters were pooled together. When presenting scatter plots, the number of data points (n), along with the Spearman's (R) and Pearson's (r) correlation, are also indicated for each graph. In the box plot, the top, middle, and bottom lines of each box represent the 25th, 50th, and 75th percentiles of the overall distribution, respectively. The upper and lower whiskers correspond to the 10th and 90th percentiles, while any outliers are indicated by dots. When comparing three or more groups, one-way ANOVA was performed followed by a two-sided Tukey's post hoc test, and groups with no statistical differences were indicated by letters such as a, b, c, etc.

Table 2. Optimal hyperparameters for the DeepPrime-FT models

cell line	PE system	Optimizer				Scheduler		Model	
		Batch size	Learning rate	Weight decay	Number of epochs	T_0 (if using scheduler)	T_mult	Hidden size	Number of models
A-549	PE2max	512	1.E-02	2.E-02	40	20	1	128	20
A-549	PE2max-e	512	2.E-03	1.E-02	100	-	-	128	20
A-549	PE4max	512	5.E-03	2.E-02	50	25	1	128	20
A-549	PE4max-e	512	1.E-02	2.E-02	100	50	1	128	20
DLD-1	NRCH-PE4max	512	4.E-03	2.E-02	100	-	-	128	20
DLD-1	PE2max	512	2.E-03	2.E-02	100	-	-	128	20
DLD-1	PE4max	512	1.E-03	0.E+00	100	-	-	128	20
HCT-116	PE2	512	8.E-03	1.E-02	50	-	-	128	20
HEK293T	NRCH-PE2	512	1.E-02	1.E-02	50	-	-	128	20
HEK293T	NRCH-PE2max	512	4.E-03	1.E-02	50	-	-	128	20
HEK293T	PE2	512	2.E-03	1.E-02	100	-	-	128	20
HEK293T	PE2max	512	1.E-03	0.E+00	100	-	-	128	20
HEK293T	PE2max-e	512	1.E-02	1.E-02	100	50	1	128	20
HEK293T	PE4max	512	5.E-03	1.E-02	100	-	-	128	20
HEK293T	PE4max-e	512	5.E-03	1.E-02	50	-	-	128	20
HeLa	PE2max	512	1.E-02	2.E-02	50	25	1	128	20
MDA-MB-231	PE2	512	5.E-03	1.E-02	100	-	-	128	20
NIH-3T3	NRCH-PE4max	512	2.E-03	2.E-02	100	-	-	128	20

Table 3. Primers for molecular cloning

Target construct	Template	Primer	Primer sequence (5' to 3') FP: forward primer, RP: reverse primer
pLenti-PE-BSD	CMV-PE2	FP	GAACACAGGACCGGTTCTAGAGCCACCATGAAACGGACAGC
		RP	TCTGAGGCACGATAGCGTCCACATCGTAGTCGGACAGCCGGTTGATG
	CMV-PE2	FP	TACGATGTGGACGCTATCGTGCCTCAGAGCTTTCTGAAGGACGACTCCA
		RP	AGAAGTTTGTTGCGCCGGATCCGACTTTCCTCTTCTTCTTGGGCTCGA
pLenti-PEmax-BSD	pCMV-PEmax-P2A-BSD	FP	GCAACGGGTTTGCCGCCAGAACACAGGACCGGTTCTAGAGCCACCATGAAACGGACAGC
		RP	GCTTCCGCCGCTAGAGCCTCCGCT
	pCMV-PEmax-P2A-BSD	FP	GATCCACCAGAGCATCACCG
		RP	TTTGTAATCCAGAGGTTGATTACCGATAAGCTTGATATCGACCGGTTAGCCCTCCCACAC
pLenti-NRCH-PEmax-BSD	pLenti-NRCH-PE2-BSD	FP	GCAACGGGTTTGCCGCCAGAACACAGGACCGGTTCTAGAGCCACCATGAAACGGACAGC
		RP	CCAGCTTTCTGCTCTTGCTCAGTC
	pLenti-NRCH-PE2-BSD	FP	GAGCAAGAGCAGAAAGCTGGAAAAT
		RP	GTCCTCTCTCTTCAGCTTCACGAGC
	pLenti-NRCH-PE2-BSD	FP	CTCGTGAAGCTGAAGAGAGAGGAC
		RP	GTCACCTCCCAGCTGAGA
	pCMV-PEmax-P2A-BSD	FP	TCTCAGCTGGGAGGTGAC
		RP	TTTGTAATCCAGAGGTTGATTACCGATAAGCTTGATATCGACCGGTTAGCCCTCCCACAC
pLenti_crRNA_hMLH1dn_Puro	pEF1a_hMLH1dn	FP	TCAAGCCTCAGACAGTGGTTC
		RP	GTTCTCGATCATCTCCTTGATTGCATTGGCAGGTCTCTGG
	pEF1a_hMLH1dn	FP	CCAGAGACCTGCCAATGCAATCAAGGAGATGATCGAGAAC
		RP	GAAAACCTTATAAAGGTCGGGCAGGTTG
	pLenti_FNLS_P2A_Puro	FP	CCTGCAGCTGGCCAACCTGCCCGACCTTTATAAGGTTTTCGCTAGCGGCAGCGGCGCCAC
		RP	GTTCTTGCAGCTCGGTGACC

Table 4. Target sequence and pegRNA information

Related Fig	Target sequence (4 bp + 20 bp Protospacer + PAM (3bp) + extended target context)	3' extension sequence of pegRNA
Figure 4c	TGGGTCCGCTGGTGCACTACGAGCTGGGAGCGCTGGCTGAAGCAGCT	GCTCCgAGCTCGTAGTGCACCAG
Figure 4c	AGTTCTACGTGAACAGCACACTTAAGGAACTGAGGAGGCTCTTAGCT	TCCTCAGTTCgTTAAGTGTGCTGTTC
Figure 4c	TGCTCAATTCTGAAGTCACTTTTTTGGTTGTAGAAATGACTGTAGCT	TTTCTACAACgAAAAAAGTGACTTCAGAAT
Figure 4c	GTAAATACCTGAAAATGCTTAAAGAGGAAAAGAGGGAAGAAGAAGCT	TTCTTCCCTCTTTTCgTCTTTAAGCATTTTCAGGT
Figure 4c	CTGCCATCTTCATTTTTGAACTTTTGGGTTCAAACTCAGAAGGAGCT	TTCTGAGTTTGAACCgAAAAGTTCAAA
Figure 4c	CAGGGGTGGCTCACCTCTAGGTACCGGCTTTCCCCTCTTAGAGAGCT	GGAAAGCgGGTACCTAGAG
Figure 4c	GTGGTGAGCCAAGTGAAAATAGCGAGGAACCCAAACTGTGGGGAGCT	AGTTTGGGTTggTCGCTATTTTCACTTGG
Figure 4c	CACAGCTACCCACTTGTTGCCACCAGGTAAGTAAGACTAAAGAAGCT	CTTACgTGGTGGCAACAAG
Figure 4c	CCTACCGTCTTGGAATCTAACTCATGGTGGGGCTGACCTAATAAGCT	TCAGCCCCACgATGAGTTAGAT
Figure 4c	AAGAACCTTCCTGTTAATGTACGTTGGCTACTTTGTGGCTGCCAGCT	GTAGCgAACGTACATTAAC
Figure 4c	AGCAGTTCCTTCTAGAGAAGCTGAAGGTAAGCCTGTTACTTGAAGCT	GGCTTACgTTCAGCTTCTCTAGA
Figure 4c	CGTTTGTCCCTCCAGCATCTGCTCTGGCTCCATGGCGGACCCTAGCT	ATGGAGCgAGAGCAGATGCTGGA
Figure 4c	ATGTGCGAGTTCAAGTGCTACCCGAGGTGCGAGGCCAGCTCGGAGCT	CTCGCACgTCGGGTAGCAC
Figure 4c	GAGTGCACTGTCACGGGTTTTCTGCGGGACAAGCTGCAGTACAAGCT	TACTGCAGCTTGTCCgGCAGAAAACCCGTGA
Figure 4c	AGCAGATTTGTAATCTCGATGTTTTGGTAGTTTCTCAATAAAGAGCT	AAACTACgAAAACATCGAGAT
Figure 4c	TCACATTGTTGTTCACCTCCTCGATGGCATTCACCCTCTTCGAAGCT	AGGGTGAATGCCAagGAGGAGGTGAACAA
Figure 4c	TACCCACCTGGCCGTTGGCCTTCGCGGGGGAAGCGCCTGCTGCAGCT	GCGCTTCCCCgGCGAAGGCCAACGGC
Figure 4c	CCCAAAGCGCATCCAGCACCGACATGGTGAACAAGGCCCGTGGCAGG	TGCCACGGGCCTTGCGGTGCTGGATGC
Figure 4c	CCTCAACATCTATAAAGACCGTCCTGGCAGATACATCTGGATCAGCT	TCCAGATGTATCTGCgAGGACGGTCTTTATAGATGTT
Figure 4c	CCTCTACGCCCTGCTTCGCGCATTCGGCTGGGCGCGCGTGGCCAGCT	CCAGCgGAATGCGCGAAGCAGGGCG
Figure 4c	TGAAGGGAGCCCTGACAAGGTGTATGGGTCGGTGTGTGGCCTTAGCT	ACCGACCgATACACCTTGTCAGGGCTCCC
Figure 4c	CATCTGCAGAGTTTCTGCCATTACTGGGTCGATTTGCAGTCGGAGCT	ATCGACCgAGTAATGGCAG
Figure 4c	TGTAGGCGCCGATGACCTCGTCCACGGTCACATTCTGGGCTGCAGGC	CTGCAGCCCAGAATACGAGGTCATCGG
Figure 4c	GGCCACACAATAGTGACAACTGTGAGGCATGTCTGCAGGGCTTAGCT	GACATGCgTCACAGTTGTCACTATTGTGT
Figure 4c	CAGGACTTGTGGAATGTGAGGATCAGGATCCACTTAATCCTGAAGCT	GGATCgTGATCCTCACA
Figure 4c	TGACCTAGACGGGTCGGGGGATCCTGGCGGCTTAGGGGACTGGAGCT	AAGCCGCgAGGATCCCCCGAC
Figure 4c	GATAACTTTAGCACCTTATTCCTTTGGCCCAGACAGCTTTGTTAGCT	CAAAGCTGTCTGGGCgAAAGGAATAAGGTGCTA
Figure 4c	GGTGGATTTGTGTGTACCCATAACAGGGTCATTTTCCATGTCAAGCT	AATGACCgTGTTATGGGTACACACAAA
Figure 4c	CCTAAGAAGGTCGACCTCATCATCTGGGAGATCGAGGTGCCAAAGCT	GGCACCTCGATCTCCgAGATGATGAGG
Figure 4c	TCCTGTATGAATCTTTCCACGCCCTGGCTCCGAGGGTCTTCAGAGCT	GGAGCgAGGGCGTGGAAAGATTCAT
Figure 4c	GCCCACATCACCCAGTCCCGAACATGGGTTTCTGTCCACTGCCAGCT	AAACCgATGTTCGGGACTGGGTGAT
Figure 4c	CTGAGTTTCAAAGATGCATAGCGTAGGGTTGCTCCTTCAAACTAGCT	AGCAACCgTACGCTATGCATC
Figure 4c	AGTCCCCTTTATGAAGAGTGTGTGAGGATGAATGATGGATTTTAGCT	ATTCATCgTCACACACTCTTC
Figure 4c	TCACTCTTCCAGCCGCGTTGTCCATGGTGAGGATGCGGTCCCCAGCT	CCTCACCATGcACAACGCGGCTGG
Figure 4c	AGTTGACAAGACATTATCAGCTACTGGTTGGCCGTTATTTATCAGCT	AACGGCCAACgAGTAGCTGATAATGT
Figure 4c	CTGGGCATCTGGAGGATTGTGATCAGGATCCAAAGAGGTAATGAGCT	TTACCTCTTTGGATCgTGATCACAATCCTCCAGATGC
Figure 4c	CTTCAATAATCAGCCAGCCTTCACTGGAGATGAACATGGCTCAAGCT	TCATCTCgAGTGAAGGCTG
Figure 4c	GCCTGGGCATCTTCGTCAGCAAAGTGGAGGAAGGCAGCAGTGCAGCT	TCCTCgACTTTGCTGACGAAGAT
Figure 4c	AAGGGGGATATGGTAGCTCCAACTGGGTGACCAGCTACCTCCTAGCT	GTCACgCAGTTGGAGCTACCATA
Figure 4c	GCTGGGGGTTACATGGCCGTAGCGTGGGCTGACGAATAAGGGTAGCT	GTCAGCCgACGCTACGGCC
Figure 4c	GCAGGAACTCAAAGCAGCCGCTGATGGCGTACTATCTGAAGTGAGCT	GATAGTACGCgATCAGCGGCTGCTTTGAGTTC
Figure 4c	ACCTGAAGAAAAGAGCAAGCGCACAGGGTTACTTCGTGGGGAGAGCT	TAACCgTGTGCGCTTGCTCTTTTCT
Figure 4c	CCTTCACTCGAACTTTGACTTGATCGGGTTCAAAACCGCATACAGCT	TTGAACCgGATCAAGTCAAAG
Figure 4c	TCATTGCCATAGGCACCATTGTCATGGTGACGGGCTTCCTCGGAGCT	AGCCCGTCACgATGACAATGGTGCCTATGGCA
Figure 4c	GGGCTGACACTAAGACGGTTAGTGTGGTTGTGGGCCGTGCTCTAGCT	CCACAACgACACTAACCGTCTTA
Figure 4c	GCTTCACTGCTACGTCCTCGCCGTTGGTGGTGGTGATGCCCAGAGCT	ACCACgAACGGCGAGGACG

Figure 4c	TTTCGAAGTTCTATCAGTTCTCGACGGAGATATTTCACCTGACAGCT	ATATCTCgGTCGAGAACTG
Figure 4c	TGGTCGGCCTGAAGAAAAAGAGTGCGGATGATGTGAAGAAGGTAGCT	TCATCgGCACTCTTTTTCTTCAG
Figure 4c	CCAAAGAATCTTTCATTTTCTTTCAGGGTCACGATCACCGGTGAGCT	TGACCgTGAAAGAAAATGAAA
Figure 4c	TGTTTTGATAGTTGCCTTATGCTTTGGATACAGAATGTTGACAAGCT	TCAACATTCTGTATCgAAAGCATAAGGCAAC
Figure 4c	TCATCAACACTAAGTTTATGTTTTTGGAACAGCTCCTATATAAAGCT	TGTTCgAAAAACATAAACTTAGT
Figure 4c	GATCAGGCAGCTAGAGACATGAAGAGGCTTGAAGAAAAGGACAAGCT	TTCAAGCgTCTTCATGTCTCTAGCTGC
Figure 4c	TGCTCAGGAACTACATCTCGATGCTGGTTTTGAGAAACTGACAAGCT	TTCTCAAAACgAGCATCGAGATGTAGTTCC
Figure 4c	AGCAATTCCGAAACACTGACATCCAGGAGTTCAAGCAACAGACAGCT	GCTTGAACTCgTGGATGTCAGTGT
Figure 4c	AGTTGATGACAGTCCAAGCACTAGTGGAGGAAGTTCCGATGGAAGCT	TCCTCgACTAGTGCTTGGACTGT
Figure 4c	CAGCTGACCCCACGTCAGAAAGCAAGGAAGATCAGTGGATGGGAGCT	TCTTCgTTGCTTTCTGACGTG
Figure 4c	TAAAGTCCCGTTACCGTGAATATCTGGTTACAGTTGTTTAAAGAGCT	GTAACgAGATATTCACG
Figure 4c	GGGCCCGCGCTCACCTGGGAGATGTGGGACTGGGTCCTCTCGCAGCT	AGTCCgACATCTCCCAG
Figure 4c	ACTAATTCGACATCAGTTTCATCGAGGAAAGGTAAGAAGAGCCAGCT	ACCTTTCgTCGATGAAACTGA
Figure 4c	CTCCTGAGCCCATTCGCCTCCTTTTGGATGGCTGGGTCCCATCAGCT	TCAGCgGGATGGGCTGG
Figure 26e	CCCTCCCCACCCCCAGGCTCAGAGTGGGTGGCTGCCCTGCAGGGGCTGACAGAGCACACATCGATACTGGTGAA	GCAGCCACCTACTCTGAGCCTGGGGGTG
Figure 26e	AGGCCCTGGTGGTCCAATGGGACCTGGATGTCCACTGGTTCCATCTTTGCCAGGAGGTCCACTGGGTCCAACAG	GACATCCAGATCCCATTGG
Figure 26e	CGCCCGCTGCCTGGGACCTCCACACGGTAAGCCCCCCAGACTGGGTCCTGGGAAAAACGCACCTGCTGGCTGTT	GGGGCTTACTGTGTGGAGGTCC
Figure 26e	GGATTGGAGCGGCGCTGGGCTCGGCGGGCACAGGCCCGGATGATGAGGTACACCACCTCGGCCACATTGAGGAT	GGGCCTGTGCCCGCTGAGCCCAGCG
Figure 26e	ACAAGCTGATGGTGGCCGTGTCGGAGGATGTCCTGCAGGTCTATGCGGACTGGCAGCGCTGGCTGTTTGGGGAG	CTGCCAGTCCGCATAGACCTACAGGACATCCTCCGACACGGCCACCATC
Figure 26e	TAAGTGAGTGCAAGAAGCGCGCCGAGGAGAAGAAACTGGAGAACGAGCGCATGGAGCGCAAGGTGGGCGCACCC	TTCTTCTCCTAGGCGCGCTTCTTGCA
Figure 26e	CTGGCAACACCTCCACGGCCGAGGAGGAGCTCTGTCGCCTTAAGCTGCTGGCCAAGCACCCCTGCCACATCAAG	AGAGCTCCTACTCGGCCGTGGAG
Figure 26e	TGAATGAGATGCGCTGCGCCTACGAGGTGACCCAGGCCAACGGAAAGTGGGAGGTGCTGATAGGTGAGTGGCCG	TGGCCTAGGTCACCTCGTAGGCGCAG
Figure 26e	TGGCCCTGAAGATCGACCTCACAGAGGCGCGAGTCCAGGTACGCGCGCCTGGAAACCGACCCCGCTCCGCCGCA	ACTCCCGCCTCTGTGAGGTCGATC
Figure 26e	GCATGGAGCTGCTTGTCCTGGTGGAGGGCATGGGCCAGCTGGTTCACAGTCTGATGGCAGAGGAAGGAAAGAAA	GGCCTATGCCCTCCACCAGGAC
Figure 26e	TTGTACGTCTTCTGCAGCCAGGAGGGGATGGCCGGCTGGAACCAGCGGTGGAAATTATCCAGGGCCAGGACTCC	GTTCTAGCCGGCCATCCCCTCCTGGCTGCAGAAG
Figure 26e	TTCAGACTGTGTCAGCTCTGCACAGGGTGAGTGGGACACAGCCCCAGCCAACTTGGCTCCCCATCTGCCCAACC	CTGCGCAGAGCTGA
Figure 26e	TGCGGGCGCGCCGGCGTGAGCTGCGGGTCCCACTGGTACTGCTGCTGGTAGAGCTGGTCACGGTGGAAGTGGCT	AGTAGGACCCGCAGCTCACGC
Figure 26e	CCCTCCCCACCCCCAGGCTCAGAGTGGGTGGCTGCCCTGCAGGGGCTGACAGAGCACACATCGATACTGGTGAA	CAGCCACCCGACTCTGAGCCTGGGGGTG
Figure 26e	AGGCCCTGGTGGTCCAATGGGACCTGGATGTCCACTGGTTCCATCTTTGCCAGGAGGTCCACTGGGTCCAACAG	ACATCCAGGCTCCCATTGG
Figure 26e	CGCCCGCTGCCTGGGACCTCCACACGGTAAGCCCCCCAGACTGGGTCCTGGGAAAAACGCACCTGCTGGCTGTT	GGGCTTACCGGTGTGGAGGTCC
Figure 26e	GGATTGGAGCGGCGCTGGGCTCGGCGGGCACAGGCCCGGATGATGAGGTACACCACCTCGGCCACATTGAGGAT	GGCCTGTGCCCGCCCGAGCCCAGCG
Figure 26e	ACAAGCTGATGGTGGCCGTGTCGGAGGATGTCCTGCAGGTCTATGCGGACTGGCAGCGCTGGCTGTTTGGGGAG	TGCCAGTCCGCATAGACCTGTCAGGACATCCTCCGACACGGCCACCATC
Figure 26e	TAAGTGAGTGCAAGAAGCGCGCCGAGGAGAAGAAACTGGAGAACGAGCGCATGGAGCGCAAGGTGGGCGCACCC	TCTTCTCCTCGGGCGCGCTTCTTGCA
Figure 26e	CTGGCAACACCTCCACGGCCGAGGAGGAGCTCTGTCGCCTTAAGCTGCTGGCCAAGCACCCCTGCCACATCAAG	GAGCTCCTCACTCGGCCGTGGAG
Figure 26e	TGAATGAGATGCGCTGCGCCTACGAGGTGACCCAGGCCAACGGAAAGTGGGAGGTGCTGATAGGTGAGTGGCCG	GGCCTGCGGTCACCTCGTAGGCGCAG
Figure 26e	TGGCCCTGAAGATCGACCTCACAGAGGCGCGAGTCCAGGTACGCGCGCCTGGAAACCGACCCCGCTCCGCCGCA	CTCGCCGCCTCTGTGAGGTCGATC
Figure 26e	GCATGGAGCTGCTTGTCCTGGTGGAGGGCATGGGCCAGCTGGTTCACAGTCTGATGGCAGAGGAAGGAAAGAAA	GCCCCATGCCCTCCACCAGGAC
Figure 26e	TTGTACGTCTTCTGCAGCCAGGAGGGGATGGCCGGCTGGAACCAGCGGTGGAAATTATCCAGGGCCAGGACTCC	TTCCGAGCCGGCCATCCCCTCCTGGCTGCAGAAG
Figure 26e	TTCAGACTGTGTCAGCTCTGCACAGGGTGAGTGGGACACAGCCCCAGCCAACTTGGCTCCCCATCTGCCCAACC	TGTTGCAGAGCTGA
Figure 26e	TGCGGGCGCGCCGGCGTGAGCTGCGGGTCCCACTGGTACTGCTGCTGGTAGAGCTGGTCACGGTGGAAGTGGCT	GTGAGGACCCGCAGCTCACGC
Figure 26e	CCCTCCCCACCCCCAGGCTCAGAGTGGGTGGCTGCCCTGCAGGGGCTGACAGAGCACACATCGATACTGGTGAA	GCAGCCACCACTCTGAGCCTGGGGGTG
Figure 26e	AGGCCCTGGTGGTCCAATGGGACCTGGATGTCCACTGGTTCCATCTTTGCCAGGAGGTCCACTGGGTCCAACAG	GACATCCAGTCCCATTGG
Figure 26e	CGCCCGCTGCCTGGGACCTCCACACGGTAAGCCCCCCAGACTGGGTCCTGGGAAAAACGCACCTGCTGGCTGTT	GGGGCTTACGTGTGGAGGTCC
Figure 26e	GGATTGGAGCGGCGCTGGGCTCGGCGGGCACAGGCCCGGATGATGAGGTACACCACCTCGGCCACATTGAGGAT	GGGCCTGTGCCCGCGAGCCCAGCG
Figure 26e	ACAAGCTGATGGTGGCCGTGTCGGAGGATGTCCTGCAGGTCTATGCGGACTGGCAGCGCTGGCTGTTTGGGGAG	CTGCCAGTCCGCATAGACCTCAGGACATCCTCCGACACGGCCACCATC
Figure 26e	TAAGTGAGTGCAAGAAGCGCGCCGAGGAGAAGAAACTGGAGAACGAGCGCATGGAGCGCAAGGTGGGCGCACCC	TTCTTCTCCTGGCGCGCTTCTTGCA
Figure 26e	CTGGCAACACCTCCACGGCCGAGGAGGAGCTCTGTCGCCTTAAGCTGCTGGCCAAGCACCCCTGCCACATCAAG	AGAGCTCCTCTCGGCCGTGGAG
Figure 26e	TGAATGAGATGCGCTGCGCCTACGAGGTGACCCAGGCCAACGGAAAGTGGGAGGTGCTGATAGGTGAGTGGCCG	TGGCCTGGTCACCTCGTAGGCGCAG
Figure 26e	TGGCCCTGAAGATCGACCTCACAGAGGCGCGAGTCCAGGTACGCGCGCCTGGAAACCGACCCCGCTCCGCCGCA	ACTCCGCCTCTGTGAGGTCGATC
Figure 26e	GCATGGAGCTGCTTGTCCTGGTGGAGGGCATGGGCCAGCTGGTTCACAGTCTGATGGCAGAGGAAGGAAAGAAA	GGCCATGCCCTCCACCAGGAC
Figure 26e	TTGTACGTCTTCTGCAGCCAGGAGGGGATGGCCGGCTGGAACCAGCGGTGGAAATTATCCAGGGCCAGGACTCC	GTTCAGCCGGCCATCCCCTCCTGGCTGCAGAAG
Figure 26e	TTCAGACTGTGTCAGCTCTGCACAGGGTGAGTGGGACACAGCCCCAGCCAACTTGGCTCCCCATCTGCCCAACC	CTGGCAGAGCTGA

Figure 26e	TGCGGGCGCGCCGGCGTGAGCTGCGGGTCCCACTGGTACTGCTGCTGGTAGAGCTGGTCACGGTGGAAGTGGCT	AGTGGACCCGCAGCTCACGC
Figure 41a-b	TTTAGTTGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGC	AATAGATGCCTAACCCCAGAAAGGGTGCTTCTT
Figure 41a-b	CAGCATCTCTGCATTCCTCAGAAGTGGTCTTTAAGATAGTCATCTGGTTTTCAGGCACTTCAAATGTACTCTTC	AAGGCCACTTCTGAGGAATGC
Figure 41a-b	TTTAGTTGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGC	AATAGATGTCTAAGCCCAGAAAGGGTGCTTCTT
Figure 41a-b	TTTAGTTGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGC	AATAGATGCATAAGCCCAGAAAGGGTGCTTCTT
Figure 41a-b	TTTAGTTGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGC	AATAGATGCGTAAGCCCAGAAAGGGTGCTTCTT
Figure 41a-b	CAGCATCTCTGCATTCCTCAGAAGTGGTCTTTAAGATAGTCATCTGGTTTTCAGGCACTTCAAATGTACTCTTC	AAGAGCACTTCTGAGGAATGC
Figure 41a-b	TTTTGTAATGAAGCATCTGATACCTGGACAGATTTTCCACTTGCTGTGCTAAAAATCCCACAAGTATTTGCAGA	AGTGGAAAATCAGTCCAGGTATCAGATGCTTCA
Figure 41a-b	CAGCTATGGAATGTGCCTTTCCTAAGGAATTTGCTAATAGATGCCTAAGCCCAGAAAGGGTGCTTCTTCAACTA	CAATTTCCTTAGGAAAGGCAC
Figure 41a-b	TTTAGTTGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGC	AATAGATGCCTAAGCTCAGAAAGGGTGCTTCTT
Figure 41a-b	TTTAGTTGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGC	AATAGATCCCTAAGCCCAGAAAGGGTGCTTCTT
Figure 41a-b	CAGCATCTCTGCATTCCTCAGAAGTGGTCTTTAAGATAGTCATCTGGTTTTCAGGCACTTCAAATGTACTCTTC	AAGACCACTGCTGAGGAATGC
Figure 41a-b	GCTAATAGATGCCTAAGCCCAGAAAGGGTGCTTCTTCAACTAAAATACAGGCAAGTTTAAAGCATTACATTACG	AAGAAGCCCCCTTTCTGGGCTTAGGC
Figure 41a-b	TGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGCTGCCAG	ATAGATGCCTAACCCCAGAAAGG
Figure 41a-b	CAGCATCTCTGCATTCCTCAGAAGTGGTCTTTAAGATAGTCATCTGGTTTTCAGGCACTTCAAATGTACTCTTC	TTAAAGGCCACTTCTGAGGAATGC
Figure 41a-b	TGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGCTGCCAG	ATAGATGTCTAAGCCCAGAAAGG
Figure 41a-b	TGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGCTGCCAG	ATAGATGCATAAGCCCAGAAAGG
Figure 41a-b	TGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGCTGCCAG	ATAGATGCGTAAGCCCAGAAAG
Figure 41a-b	CAGCATCTCTGCATTCCTCAGAAGTGGTCTTTAAGATAGTCATCTGGTTTTCAGGCACTTCAAATGTACTCTTC	TTAAAGAGCACTTCTGAGGAATGCA
Figure 41a-b	TTTTGTAATGAAGCATCTGATACCTGGACAGATTTTCCACTTGCTGTGCTAAAAATCCCACAAGTATTTGCAGA	GGAAAATCAGTCCAGGTATCAGATGCTT
Figure 41a-b	TGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGCTGCCAG	AAGGAAATTGCTAATAGATGCCTAAGCCCAGAAAGGGT
Figure 41a-b	CAGCTATGGAATGTGCCTTTCCTAAGGAATTTGCTAATAGATGCCTAAGCCCAGAAAGGGTGCTTCTTCAACTA	TTTCTGAGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATT
Figure 41a-b	TGAAGAAGCACCCTTTCTGGGCTTAGGCATCTATTAGCAAATTCCTTAGGAAAGGCACATTCCATAGCTGCCAG	ATAGATCCCTAAGCCCAGAAAGG
Figure 41a-b	CAGCATCTCTGCATTCCTCAGAAGTGGTCTTTAAGATAGTCATCTGGTTTTCAGGCACTTCAAATGTACTCTTC	TTAAAGACCACTGCTGAGGAATGCA
Figure 41a-b	CTAATAGATGCCTAAGCCCAGAAAGGGTGCTTCTTCAACTAAAATACAGGCAAGTTTAAAGCATTACATTACGT	AGAAGCCCCCTTTCTGGGCTTAGG

3. Results

3.1. Predicting prime editing efficiency in a limited form

3.1.1. High-throughput evaluation of PE2 efficiency

To conduct a comprehensive analysis of PE2 efficiencies on a large scale, we adapted and refined a paired library method that has been previously employed by our team and others to investigate the activities and outcomes of Cas12a and Cas9 across thousands of target sequences[7,11,16,29-32]. We constructed a lentiviral plasmid library, referred to as Library-1, derived from a collection of oligonucleotides. This library comprised 48,000 pegRNA-target sequence pairs, encompassing 2,000 distinct target sequences each paired with 24 different combinations of PBS and RTTs (**Figure**).

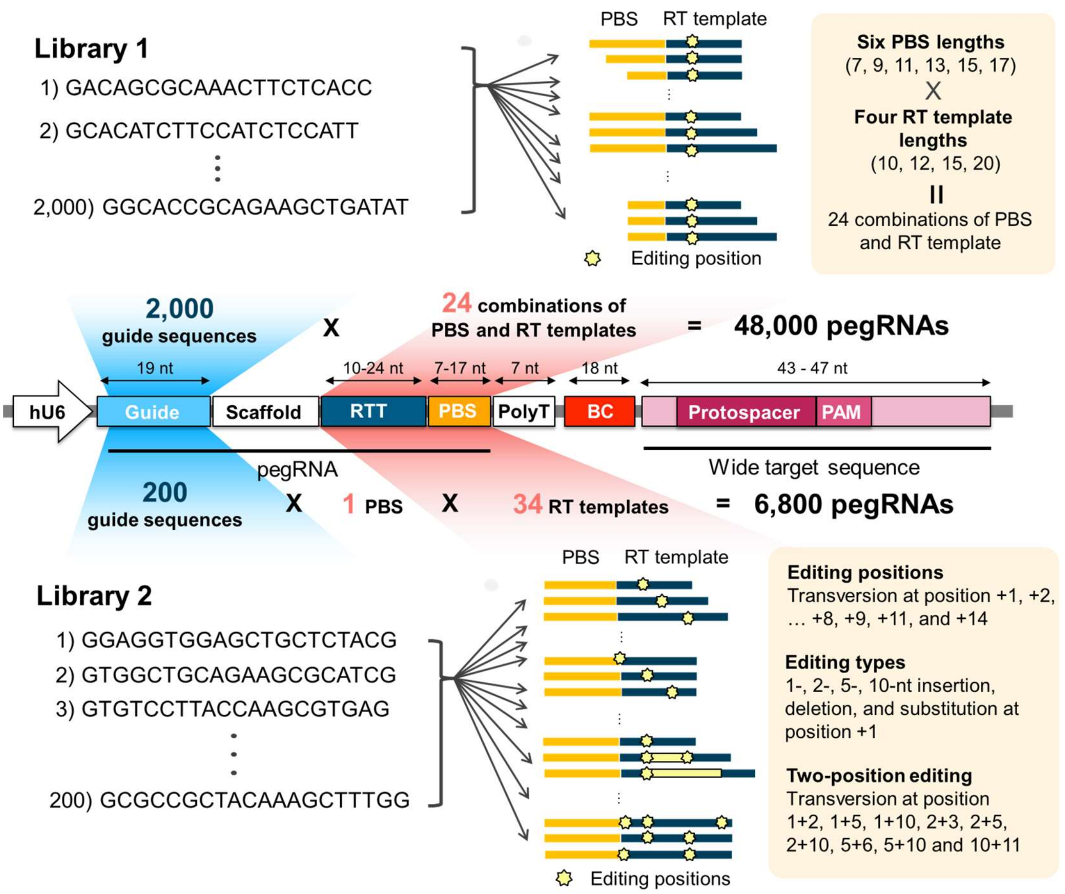

Figure 1. Construction of libraries 1 and 2. In Library-1, a total of 2,000 guide sequences were

paired with 24 distinct combinations of primer binding site (PBS) and reverse transcriptase (RT) template lengths, leading to the creation of 48,000 pegRNAs. Meanwhile, in Library-2, 200 guide sequences were combined with 34 unique PBS and RTT pairings, each designed to induce specific edits at various positions, resulting in a total of 6,800 pegRNAs.

To examine how variations in PBS and RTT lengths influence outcomes, the library was designed with 24 different combinations of these lengths—six PBS lengths (7, 9, 11, 13, 15, and 17 nucleotides) combined with four RTT lengths (10, 12, 15, and 20 nucleotides). This resulted in 24 distinct combinations for each of the 2,000 guide and target sequence pairs, aiming to induce a G-to-C transversion mutation at the +5 position from the nicking site (position 22 within the target sequence). This configuration produced 48,000 pegRNA-target sequence pairs in total (24 combinations x 2,000 pairs; see **Figure** , Library-1).

Additionally, to assess the impact of factors other than PBS and RTT lengths on PE2 efficiency, we created a second library, termed Library-2. This library contained 6,800 pairs of pegRNA sequences and their corresponding targets, specifically designed to evaluate variables such as editing positions, the type of edits (insertion, deletion, or substitution), and the locations for two-position edits (refer to **Figure** , Library-2). The numbering system used to describe the pegRNA and target sequences in this study is illustrated in **Figure 2**.

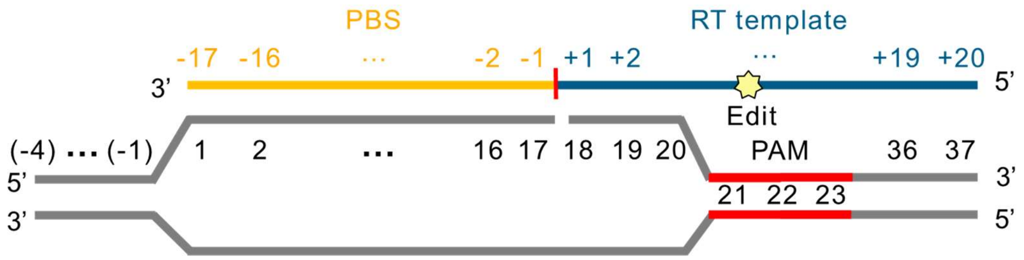

Figure 2. Schematic of the position of pegRNA and target in this study. The numbering of positions within the pegRNA and the cDNA produced from the pegRNA begins at the Cas9 nickase's nicking site. For the broader target sequence, positions are assigned starting from the 20th nucleotide upstream of the PAM, which is labeled as position 1, while the nucleotides in the NGG PAM are labeled as positions 21 through 23.

HEK293T cells were infected with lentivirus produced from the plasmid library to create a cell library at a multiplicity of infection (MOI) of 0.3, with non-infected cells eliminated via puromycin selection (refer to **Figure 3**). Each cell in this library harbored a pegRNA and the corresponding integrated target sequence. The cells were then transfected with a plasmid encoding PE2, and blasticidin was used to remove cells that were not successfully transfected. Four and a half days post-transfection, genomic DNA was extracted from the cells, and the target sequences were amplified through PCR.

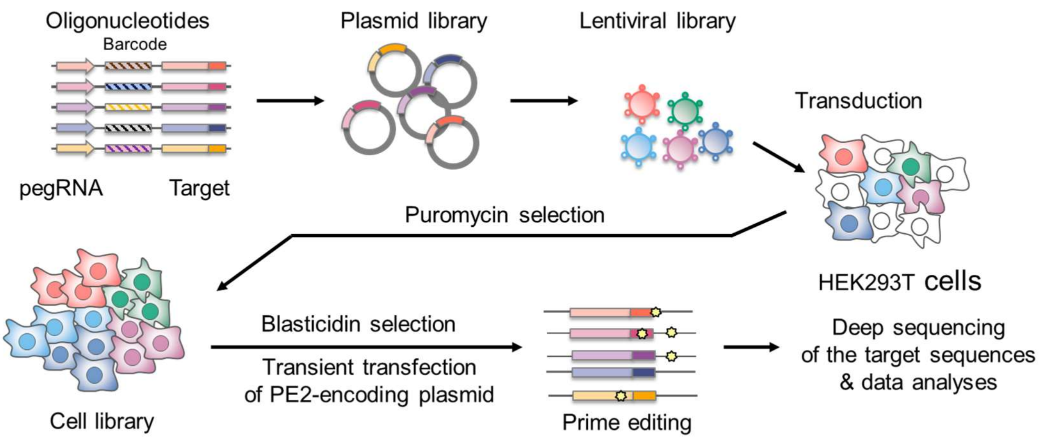

Figure 3. Schematic representation of the experimental procedure.

The amplicons underwent deep sequencing to assess the mutation rates triggered by PE2. Sanger sequencing revealed that 8.5% (12 out of 142) of the plasmid library copies contained at least one mutation within the guide sequence, scaffold, PBS, RTT, or target sequence regions (**Table 5a**). These mutations likely resulted from errors introduced during the synthesis of oligonucleotides and PCR amplification. Additionally, when using lentiviral vectors for high-throughput assessments, distant elements can sometimes rearrange or shuffle[33-36]. The uncoupling rate between the pegRNA-encoding and barcode-target sequences in the cell library was found to be 4.2% (**Table 5b**), which is consistent with previously reported rates[15,33-36]. Since prime editing is unlikely to occur with these mutated or uncoupled sequences, the observed PE2 efficiency would be approximately 87% of the actual PE2 efficiency (i.e., if the actual PE2 efficiency is 25%, the observed efficiency would be 25% x 87% = 22%).

Table 5. Error rates in the plasmid and cell library

(a) Error rate in the plasmid library	
Copies without any errors	130
Copies containing any error in the pegRNAs or target sequence regions	12
Copies with shuffling	0
Error rate in the plasmid library	12/142 = 8.5%

(b) Lentiviral vector-induced shuffling efficiency	
Initial template	Observed shuffling efficiency by deep sequencing
Plasmid library	11 / 1,101 (1.00%)
Genomic DNA from cell library	27 / 520 (5.19%)
Lentiviral vector-induced shuffling (%)	5.19% - 1.00% = 4.2%

(a) The error rates in the copies in the plasmid library were evaluated by Sanger sequencing. (b) The lentiviral vector-induced shuffling frequency was evaluated using deep sequencing. Given that the PCR used for the deep sequencing sample preparation induces shuffling, the shuffling efficiency observed in the plasmid library using deep sequencing was subtracted from the total observed shuffling frequencies in the cell library[15].

We identified a strong correlation between replicates transfected by two different researchers (**Figure 4a)**. This consistency led us to merge the data from these replicates for further analysis. These findings align with the robust correlation observed in our prior experiments involving Cas9 [7,11]. Subsequently, we assessed the relationship between editing efficiencies measured at integrated sequences using a high-throughput method and those at endogenous sites measured through individual tests. Our analysis of a previously published dataset of PE3 efficiencies from the initial study1 revealed a notable correlation (**Figure 4b**; Spearman's R = 0.59, Pearson's r = 0.69). Additionally, we created six new datasets evaluating PE2 efficiencies at 20 to 31 endogenous sites, selected randomly from the 54,836 pegRNAs in libraries 1 and 2. During these experiments, PE2 and pegRNA plasmids were transiently transfected. We found a consistent and high correlation between PE2 efficiencies at endogenous sites and their corresponding integrated target sequences (**Figure 4c**). The average PE2 efficiency with libraries 1 and 2 was 9.9%, which is comparable to the 9.5% efficiency recorded in the initial study[1] (**Figure 4d**).

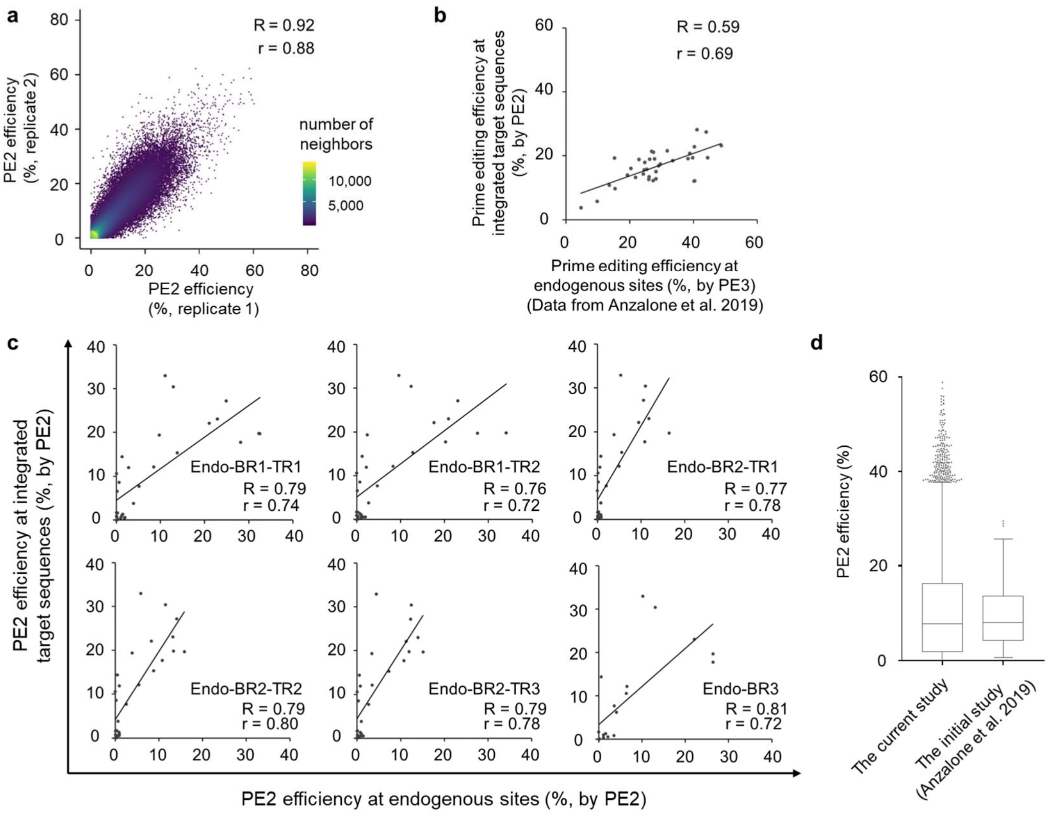

Figure 4. High-throughput evaluation of PE2 activity. (a) The correlation of PE2 efficiencies was evaluated by comparing results from separate transfections with the PE2-encoding plasmid. The color intensity of each dot reflects the density of neighboring dots (i.e., dots within a radius three times the dot size). The analysis included 49,301 pegRNA and target sequence pairs. (b-c)

Comparisons of PE2 efficiencies at endogenous sites versus corresponding integrated target sequences. Data were sourced from the initial study (b, ref.[1] , 36 pegRNA and target sequence pairs in HEK293T cells) and from newly generated PE2-Endo data in this study (c). (d) The distribution of PE2 efficiencies was compared to those from the initial study. The number of pegRNAs analyzed were 49,301 (from this study's libraries 1 and 2) and 186 (from the initial study[1]).

3.1.2. The correlation between SpCas9 and PE2 activities

In prime editing, SpCas9 must bind to the target site and create a single-strand break[1]. Given this requirement, a significant correlation between the activities of PE2-pegRNA and Cas9-sgRNA is anticipated. Our previous research assessed the indel frequencies linked to Cas9-sgRNA at 2,000 different target sites[7]. When we analyzed the correlation between PE2-pegRNA and Cas9-sgRNA activities at these same sites, we observed a moderate level of correlation, as expected (**Figure 5**).

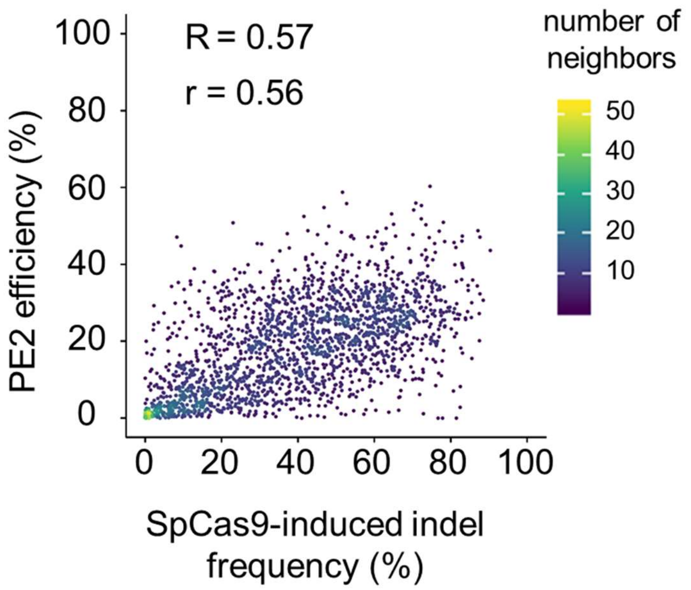

Figure 5. The correlation between SpCas9 indel frequencies and PE2 efficiencies. To reduce the impact of the length of PBS and RTT, the pegRNA with the highest efficiency out of 24 different PBS and RTT length combinations was selected for each target sequence. The color intensity of each dot represents the density of neighboring dots (i.e., dots within a radius three times the dot's size). This analysis involved 1,956 pegRNA and target sequence pairs.

The moderate correlation, rather than a strong one, can be attributed to the additional steps involved in prime editing that are not directly related to the indel formation facilitated by Cas9. These steps include the reverse transcription of pegRNA, cleavage of the 5' flap, and subsequent DNA repair processes. We will discuss these factors in more detail below. The efficiency of both PE2 and Cas9 nucleases generally exhibited uniform distributions, though high-efficiency instances were relatively rare (**Figure 6**). Additionally, while PE2 efficiencies exhibited a slight bimodal distribution—with one mode indicating very low editing activity (below 2%) and another around 25%—Cas9 efficiencies did not display this pattern. When considering all pegRNAs across the 24 different PBS and RTT length combinations, the frequency of pegRNAs tended to decline as PE2 efficiency increased.

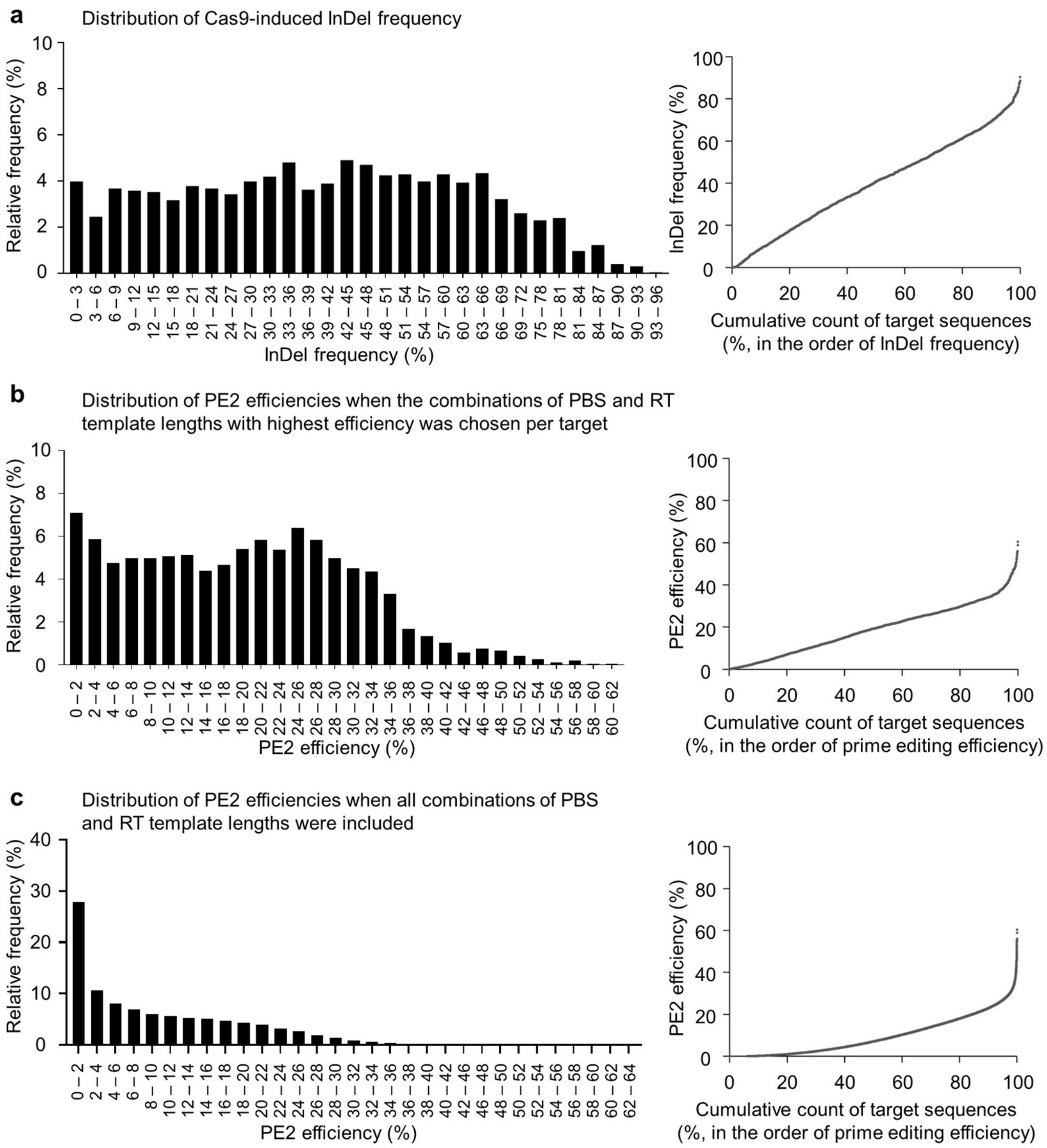

Figure 6. Distribution of indel frequencies caused by Cas9 and efficiencies of PE2. The histograms (left) and ranked scatter plots (right) display the distribution of SpCas9-induced indel frequencies (a) and PE2 efficiencies (b, c) across 1,956 target sequences. (b) The pegRNA with the

highest efficiency among the 24 different PBS and RTT length combinations was selected for each target sequence. (c) All pegRNAs, considering all 24 PBS and RTT length combinations, were included in the analysis.

3.1.3. Impact of PBS and RTT lengths on PE2 efficiency

The efficiency of prime editing at a specific target site is influenced by the lengths of the PBS and RTT regions in the pegRNA, with various combinations potentially affecting the overall efficiency[1]. To investigate this, we examined how different lengths of PBS and RTTs impacted PE2 efficiency across 2,000 target sequences. Our analysis revealed that the average editing efficiencies across the different PBS and RTT lengths followed a unimodal distribution. The highest average efficiency of 13.4% was achieved using pegRNAs with an 11- to 13-nt PBS and a 10- to 12-nt RTT (**Figure 7a-c**). When categorizing pegRNAs as poorly performing if their PE2 efficiency was below 5%, we found that 28% to 81% (with an average of 43%) of pegRNAs fell into this category depending on the PBS and RTT lengths (**Figure 7d**). Conversely, 19% to 72% (averaging 57%) of pegRNAs demonstrated PE2 efficiencies above 5% (**Figure 7e**).

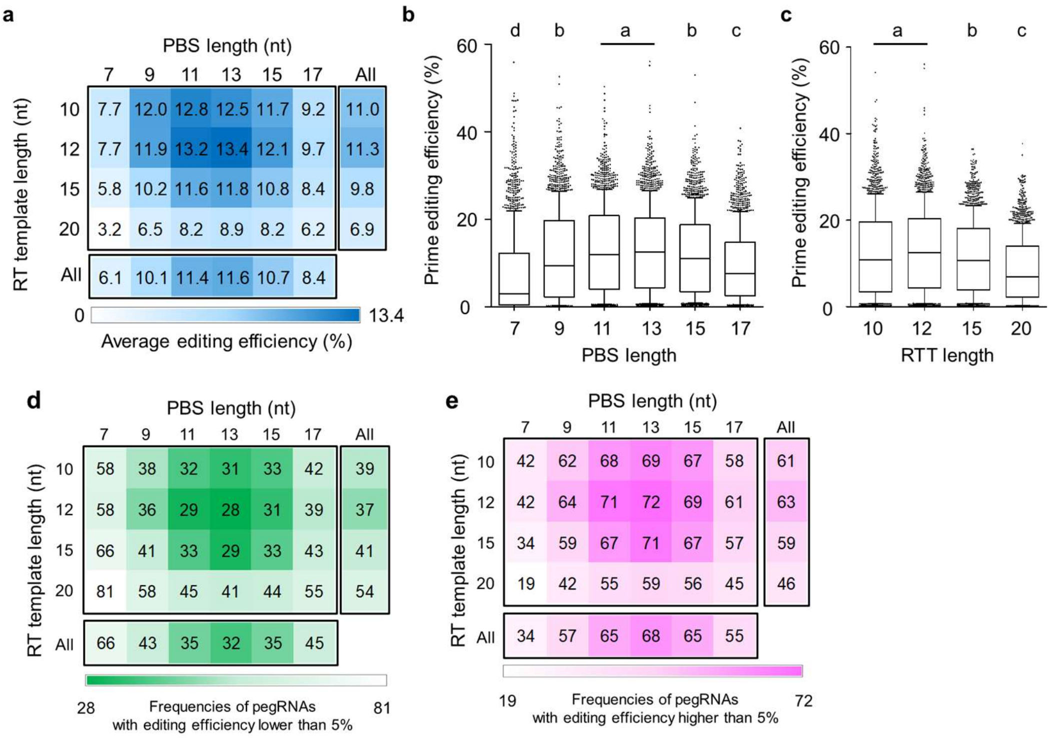

Figure 7. The impact of PBS and RTT length on PE2 efficiency. (a, d-e) Heat maps display the average editing efficiencies (a), and the frequencies of pegRNAs with PE2 efficiencies greater than (d) or less than (e) 5% for different lengths of PBS and RTTs. (b-c) PE efficiencies were assessed by varying the lengths of the PBS (b) or the RTT (c) while keeping the RTT (b) and PBS (c) lengths fixed at 12 nt and 13 nt, respectively.

Our findings indicate that the ideal combination of PBS and RTT lengths varies by target sequence, consistent with previous studies in human cells[1] and plants[37]. To further explore this, we assessed how often each PBS and RTT length combination resulted in the highest editing efficiencies for specific targets. These results also followed a unimodal distribution, with the most frequent high efficiencies observed using a 9- to 13-nt PBS and a 10- to 12-nt RTT (**Figure 8a**). Previous investigations into the effect of PBS and RTT lengths on editing efficiency had not clarified whether these factors operated independently. Our extensive dataset analysis revealed that these parameters are indeed independent ($p = 0.25$ by Chi-square test; **Figure 8b-c**).

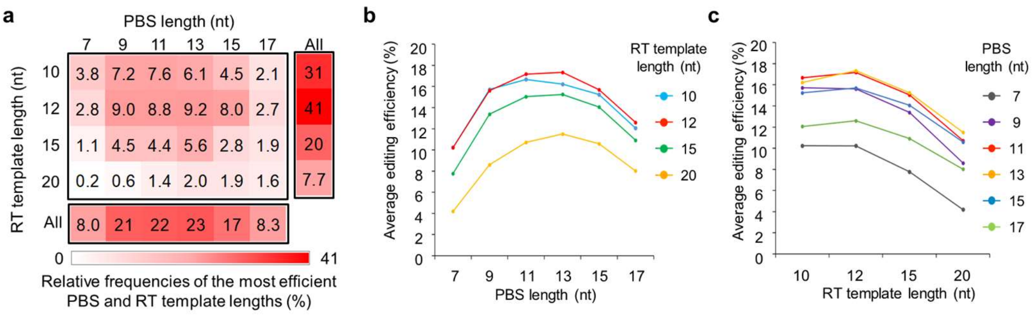

Figure 8. The ideal pairing of PBS and RTT lengths. (a) The heat maps illustrate the frequencies of different PBS and RTT length combinations that produced the highest editing efficiencies for each target sequence. (b-c) The correlation between PBS lengths and RTT lengths was analyzed by selecting the most efficient combination of these two parameters for each target sequence. Target sequences were excluded from the analysis if all 24 PE2 efficiencies (resulting from the 24 PBS and RTT length combinations) were below 10%, in order to reduce random errors and enhance comparison accuracy. No significant association was found between PBS and RTT lengths ($p = 0.25$, Chi-square test). Each dot represents $n = 651$ target sequences.

We also compared the average editing efficiencies of the most effective pegRNAs for each target sequence based on PBS and RTT lengths. Interestingly, the highest average efficiencies were achieved with shorter PBS and RTT lengths (e.g., a 7-nt PBS and a 10- to 12-nt RTT), with efficiency decreasing as the lengths increased (**Figure 9**). Based on these results, we recommend initially testing PE2 efficiencies with a 13-nt PBS and a 12-nt RTT, followed by further testing with a 9- to 15-nt PBS and a 10- to 15-nt RTT. This approach aligns with the length recommendations from the initial study[1], which suggested approximately a 13-nt PBS and a 10- to 16-nt RTT based on evaluations from five target sequences.

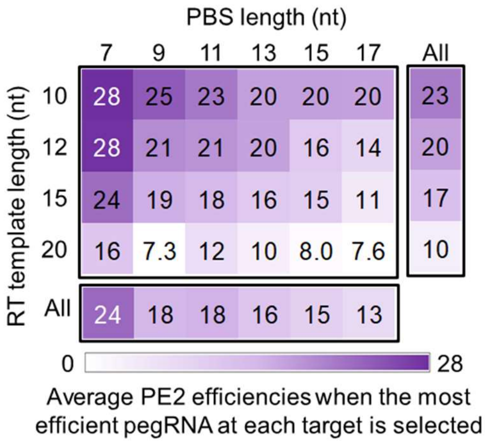

Figure 9. The most effective pegRNA for each target. The heat map display the average editing efficiencies based on selecting the combination of PBS and RTT lengths that yielded the highest editing efficiency for each specific target.

In our analysis of pegRNAs featuring a 13 nt PBS and identical target sequences, we noted that variations in RTT lengths yielded relatively high correlations in their efficiencies (**Figure 10**). This indicates that additional variables may influence the performance of PE2 beyond just the template length.

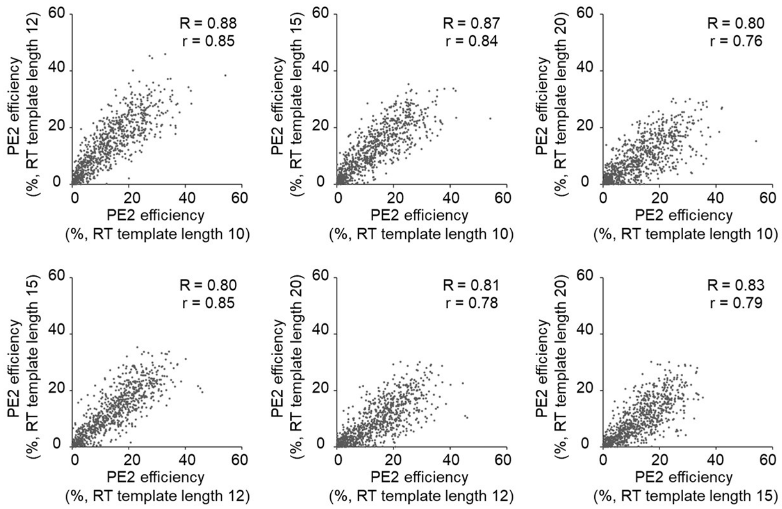

Figure 10. Relationship between PE2 efficiencies across varying RTT lengths. The dataset includes 887 pairs of pegRNAs and target sequences. The RTT lengths are represented on both the x and y axes.

3.1.4. Factors influencing PE2 efficiency

To systematically analyze factors affecting PE2 efficiency, we employed Tree SHAP[28], using 1,766 features. These included metrics such as melting temperature, GC counts, GC content, and minimum self-folding free energy of various pegRNA regions, along with PBS and RTT lengths, DeepSpCas9 scores (which predict Cas9 nuclease activity at specific target sequences)[7], and direct sequence information like mono- and dinucleotides (**Figure 11**). Features associated with high and low prime editing efficiencies were categorized as advantageous and disadvantageous, respectively.

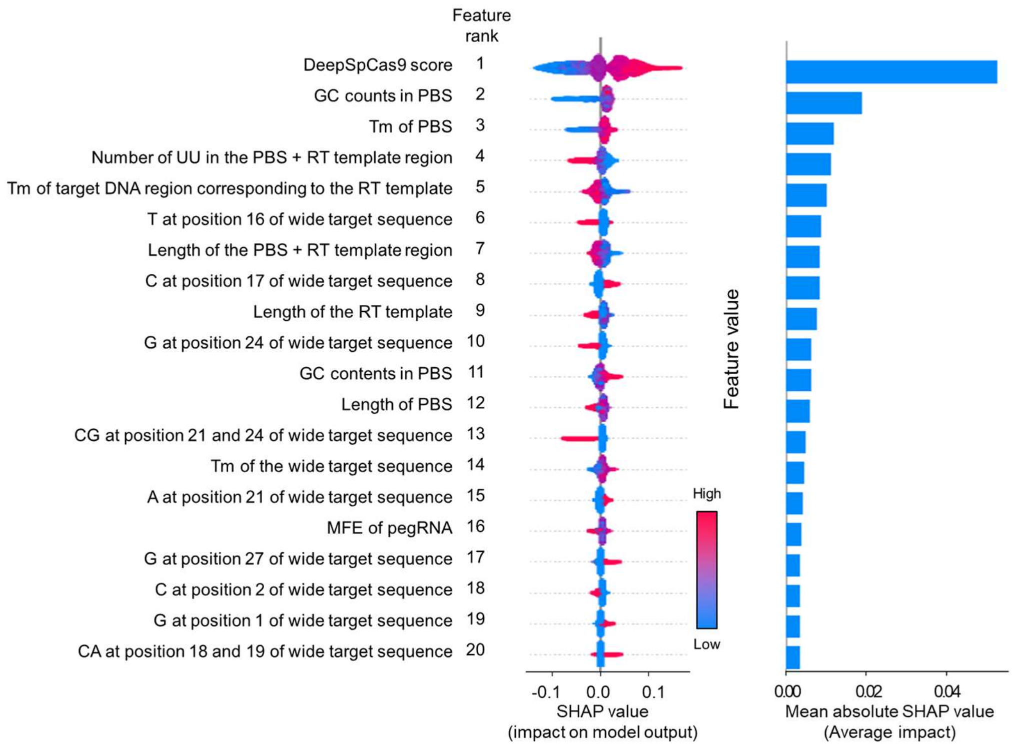

Figure 11. Features influencing PE2 efficiency identified by Tree SHAP analysis. In the summary violin plot (left), each target sequence is depicted as a dot on the x-axis, where the position reflects its SHAP value. High SHAP values are associated with elevated prime editing efficiencies, while low SHAP values correspond to reduced efficiencies. The color of each dot indicates the magnitude of the relevant feature for that target sequence, with red and blue denoting high and low values, respectively. The term "Tm" refers to the melting temperature. The dataset includes 38,692 pegRNA and target sequence pairs, represented by the number of dots in the summary plot.

The most influential feature was the DeepSpCas9 score (beneficial), consistent with the observed correlation between SpCas9-induced indel frequencies and PE2 efficiencies. GC counts within the PBS region (beneficial) ranked second in importance, and GC content in PBS (beneficial) was the 11th most significant feature. Higher GC content in the PBS contributes to stronger binding of the pegRNA to the target DNA, essential for effective reverse transcription. Systematic evaluation revealed that increased GC content and GC counts in the PBS positively correlated with higher PE2 efficiencies (**Figure 12**).

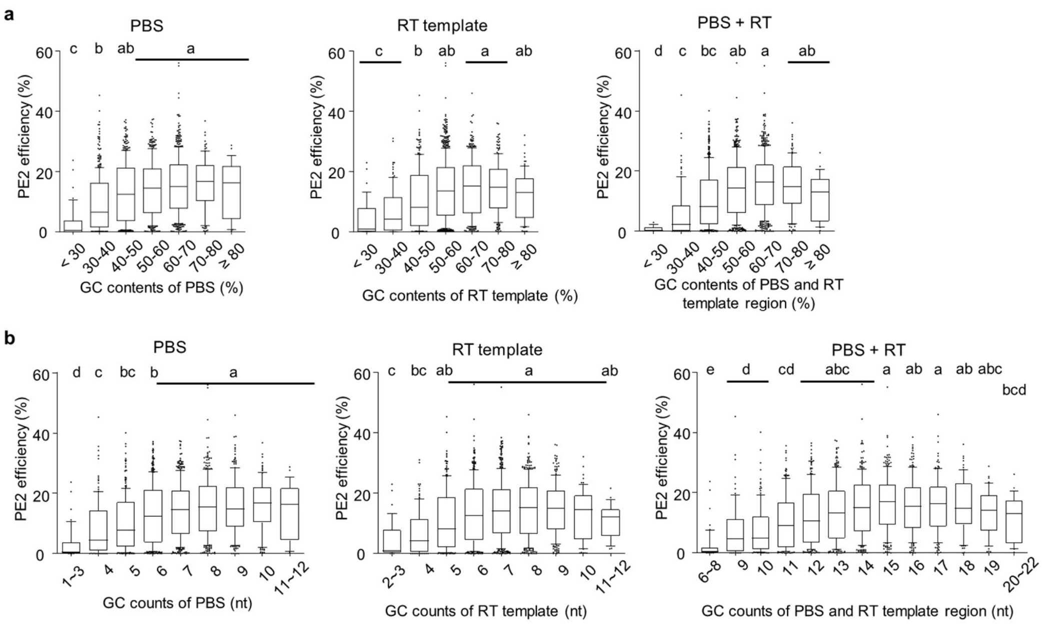

Figure 12. Effect of GC in PBS and RTT on PE2 efficiency. The pegRNAs were categorized based on the GC content (a) and GC count (b) within the PBS and RTT regions. For all groups, the PBS length was fixed at 13 nt, and the RTT length was 12 nt.

When GC content in the PBS was below 30%, PE2 efficiencies were generally low across all PBS lengths, though longer PBS sequences (e.g., 15 nt) showed relatively better efficiency. Conversely, when GC content exceeded 60%, shorter PBS sequences (7 to 11 nt) yielded higher PE2 efficiency. Based on these observations, we recommend using a 15-nt PBS for GC contents below 40% and a 9-nt PBS for GC contents above 60% (**Figure 13**). The impact of GC content and counts in the RTT on PE2 efficiency was minimal, with extreme GC values generally leading to lower efficiencies. Accordingly, these features were not among the top 40 most significant.

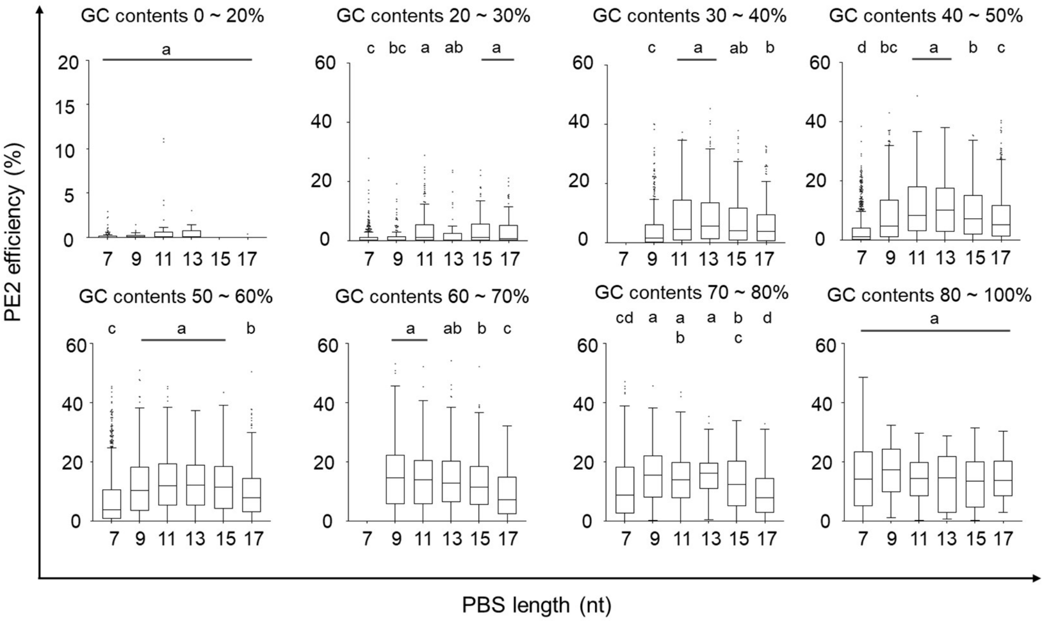

Figure 13. Effect of GC contents in PBS and PBS lengths on PE2 efficiency. The pegRNAs were categorized based on their GC content and PBS lengths. The efficiency of PE2 was then assessed for these categories across all RTT lengths tested.

Other notable features included the melting temperature of the PBS (beneficial) and the melting temperature of the target DNA region corresponding to the RTT (detrimental if above 35˚C). A higher melting temperature of the PBS likely correlates with a higher GC count, enhancing the pegRNA's binding to the target DNA and facilitating reverse transcription. We observed that higher PBS melting temperatures corresponded with increased PE2 efficiencies (**Figure 14a**). However, excessively high melting temperatures of the target DNA region could impede the conversion of the 3' flap into a 5' flap, crucial for integrating the reverse-transcribed DNA into the genome. We found that PE2 efficiency decreased slightly when the melting temperature of this region exceeded 35°C, though the difference was not statistically significant (**Figure 14b**). The number of UUs in the RT + PBS region (detrimental) was another important factor. High numbers of Ts in pegRNA-encoding sequences result in many Us in the pegRNAs, potentially reducing RNA polymerase III transcription efficiency[9,38] and leading to lower intracellular pegRNA levels.

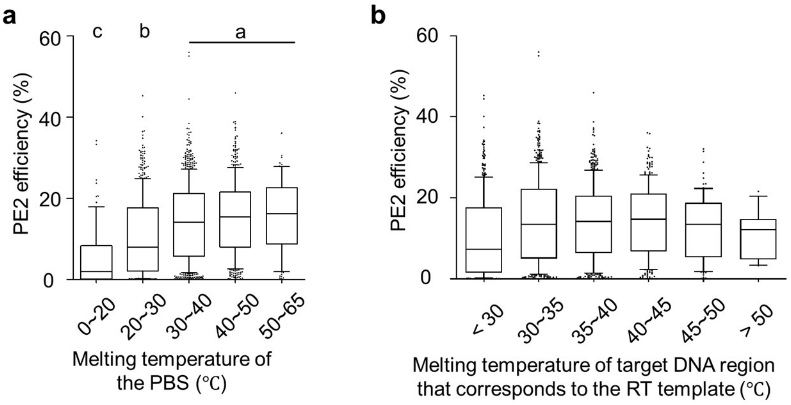

Figure 14. Effect of the melting temperatures on PE2 efficiency. The pegRNAs were sorted based on the melting temperatures of the PBS (a) and the target sequence regions associated with the RTT (b). In all cases, the PBS was 13 nt long and the RTT was 12 nt long.

Additional significant features included the presence of T at position 16 (detrimental) and C at position 17 (beneficial) in the target sequence (where Position 1 is the 20th nucleotide from the NGG PAM). A T at position 16 has been associated with reduced Cas9 nuclease activity[32,39,40] and lowers GC counts in the PBS, which impairs reverse transcription, especially with shorter PBS lengths. Conversely, a C at position 17 has been linked to increased Cas9 activity and higher GC counts in the PBS, enhancing reverse transcription. The RT-PBS lengths, as well as the RTT length (only detrimental when long), were also significant factors, as discussed earlier. The fixed type of prime editing (+5 G to C) would replace a G at position 22, which would alter the PAM sequence and block the re-binding of Cas9 to the target DNA. However, if the nucleotide at position 24 is a G, it could create a GG PAM sequence across positions 23 and 24. This change could enable Cas9 to re-bind to the target site, potentially resulting in the nicking of the reverse-transcribed DNA strand prior to the repair of the complementary strand[1,11,41-43].

3.1.5. Influence of editing type and position on PE2 efficiency

We previously assessed PE2 efficiency for G to C conversions at a fixed position (+5 from the nicking site) across 2,000 target sequences using Library-1. In our extended analysis, we explored a broader range of genome edits using Library-2, which included 6,800 pegRNA-target sequence pairs (200 target sequences × 1 PBS/target sequence × 34 RTTs/target sequence). This analysis aimed to evaluate how different types of editing (such as insertions versus deletions versus substitutions), the position of the edit, and the number of nucleotides inserted or deleted affect PE2 efficiency. Our initial findings revealed that, generally, the efficiency of generating 1-bp insertions, deletions, and substitutions could be ranked as follows: insertions ≥ deletions ≥ substitutions, with the difference between insertion and substitution efficiencies being statistically significant (**Figure 15**). We then examined the impact of varying the number of inserted nucleotides and discovered that while 1-bp and 2-bp insertions yielded similar efficiencies, the efficiency dropped for 5-bp insertions and decreased even further for 10-bp insertions (**Figure 15**). Similarly, for deletions, the efficiency for 1-, 2-, and 5-bp deletions was comparable, whereas 10-bp deletions showed a marked reduction in efficiency.

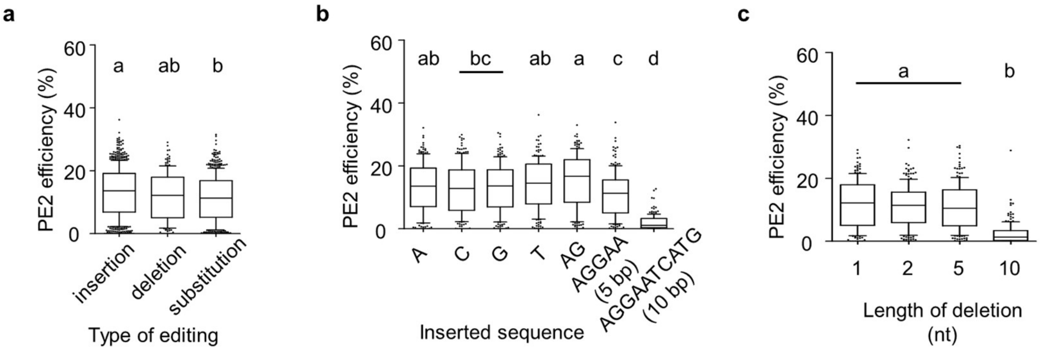

Figure 15. Prime editing efficiency depending on edit type. (a) PE2 efficiencies for single-base insertions, deletions, and substitutions. (b) Impact of the type and quantity of inserted nucleotides on PE2 efficiency. (c) Influence of deletion length on PE2 efficiency. Editing was targeted at position +1 relative to the Cas9 nickase's nicking site, with a PBS length of 13 nt. The total length of the RTT's left and right homology arms was 14 nt.

We also investigated how the identity of substituted nucleotides influenced PE2 efficiency. Testing all possible 1-bp substitutions at position +1 from the nicking site (which is between positions 17 and 18 in the target sequence) showed that PE2 efficiencies varied slightly depending on the substitution type, with C to T conversions achieving the highest efficiency and T to G conversions the lowest (**Figure 16**).

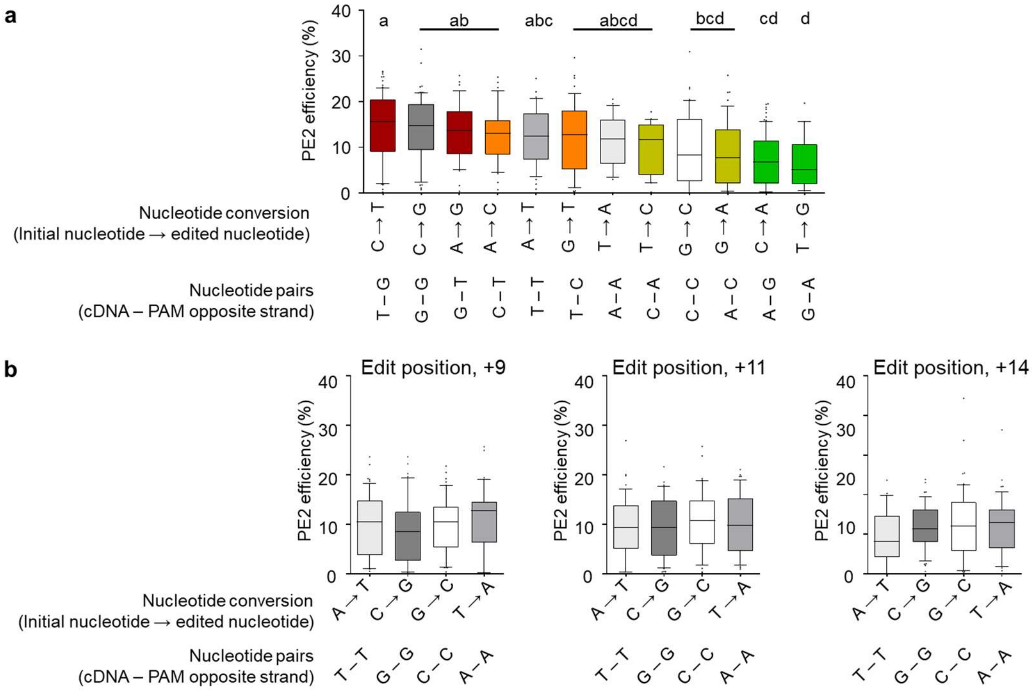

Figure 16. Effect of the type of substitutions on prime editing efficiency.

To understand these effects mechanistically, we analyzed the temporary base pairing between the cDNA from the RTT and the complementary nucleotide in the PAM-opposite strand. The efficiency rankings for various nucleotide pairings were as follows: T (cDNA) – G (PAM-opposite) and G–T pairings ≥ C–T and T–C pairings ≥ C–A and A–C pairings ≥ A–G and G–A pairings. The significant differences between T–G/G–T pairings and A–G/G–A pairings suggest that temporary base pairing between cDNA and the PAM-opposite strand affects PE2 efficiency. Pairings involving identical nucleotides, such as T–T, G–G, C–C, and A–A, produced comparable efficiencies for conversions like A to T and C to G (**Figure 16a**). Additionally, when examining these conversions at different positions (+9, +11, and +14 from the nicking site), efficiencies remained consistent across all tested positions (**Figure 16b**), supporting our findings at position +1.

We also studied the effect of the edit position on 1-bp substitution efficiencies. Editing efficiencies were generally similar across all positions from +1 to +14, with the exception of +3, +5, and +6 (**Figure 17a**). The lowest efficiency was noted at position +3, though the cause is unclear. The highest efficiencies were observed at positions +5 and +6, where the GG PAM is located. If the PAM is not altered, Cas9 can re-bind to the target and nick the reverse-transcribed DNA strand, which reduces PE2 efficiency. This PAM editing effect was also evident in 2-bp substitution efficiencies. Editing efficiencies were higher when one or both nucleotides in the PAM (positions +5 and +6) were edited (e.g., at positions +1 and +5, +2 and +5, +5 and +6) compared to when the PAM remained intact (e.g., at positions +1 and +2, +1 and +10) (**Figure 17b**). These results suggest that employing SpCas9 variants with different PAM specificities could enhance PE2 efficiency for certain targets.

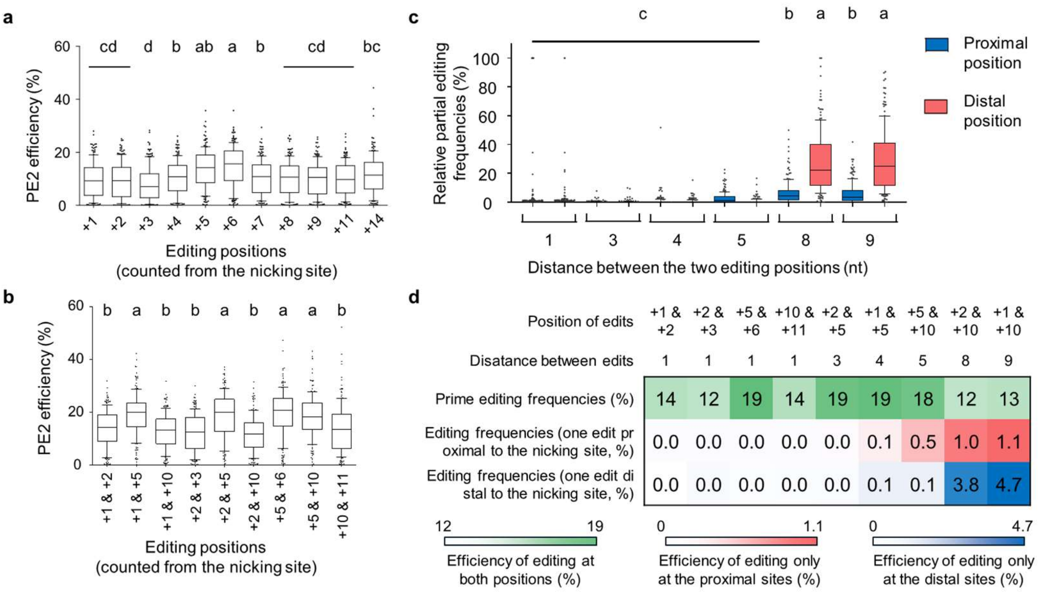

Figure 17. Impacts of editing type and location on PE2 efficiency. (a) The influence of editing location on PE2 efficiency for single-base transversion substitutions. The positions on the x-axis represent distances from the nicking site. (b) The effect of editing position on prime editing efficiency for single-base transversion substitutions at two distinct locations. (c) Frequency distribution of partial editing based on the separation between the two editing sites mentioned in (b). (d) A heatmap illustrating the average occurrence rates of partial (1 nt) and full (2 nt) edits as outlined in (b).

Interestingly, up to a median of 20% of sequences with at least one intended edit contained only a single edit (**Figure 17c-d**). Partial editing was more common at positions further from the nicking site and increased as the distance between the two editing sites grew.

3.1.6. Computational models that predict PE2 efficiencies

We aimed to create a computational model to predict PE2 efficiencies for various target sequences using 24 distinct pegRNAs with differing PBS and RTT lengths. Building on our previous success with deep learning models for predicting Cas12a and Cas9 efficiencies[7,11,30], we applied similar techniques to develop a model for PE2. For this purpose, we utilized data from Library-1, which contained 48,000 pegRNA-target sequence pairs. This data was divided into two subsets: HT-training (38,692 pairs) and HT-test (4,457 pairs), ensuring no overlap of target sequences between the two groups. Using the HT-training subset, we trained our model to predict

PE2 efficiencies for 24 pegRNAs with varying PBS and RTT lengths, focusing on G to C conversions at position +5 (**Figure 18**).

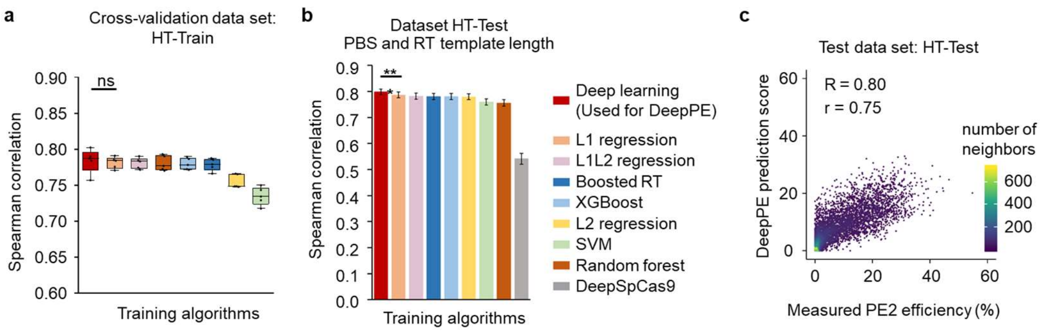

Figure 18. Development of computational models for predicting PE2 efficiencies. (a) Cross-validation of predictive models based on different machine learning approaches. Each point represents the Spearman's correlation derived from five-fold cross-validation of predicted PE2 efficiencies versus observed values (total, n = 5). (b) Comparison of DeepPE's performance with other predictive models using the HT-Test dataset. The bar graph displays Spearman's correlation between observed PE2 efficiencies and predicted scores. Error bars indicate 95% confidence intervals, with the analysis based on n = 4,457 pegRNA and target sequence pairs. For brevity, statistical comparisons are shown only between the top-performing model and the next best model; NS indicates no significant difference; statistical significance was tested using two-sided Steiger's test. (c) Assessment of DeepPE with the HT-Test dataset (n = 4,457 pegRNA and target sequence pairs). Dot colors reflect the density of neighboring points (i.e., points within a radius three times the size of the dot).

Our cross-validation results indicated that the deep learning approach performed the best, though its advantage over L1 regression, the second most effective method, was not statistically significant (**Figure 18a**). Testing on the HT-test dataset showed that DeepPE, our deep learning model, slightly outperformed other conventional machine learning models (**Figure 18b-c**), consistent with our earlier findings for Cas12a and Cas9[7,30]. Evaluation using six independent replicates at endogenous sites yielded Spearman and Pearson correlation of R = 0.67 to 0.77 (average 0.73) and r = 0.63 to 0.74 (average 0.69), respectively (**Figure 19**), demonstrating DeepPE's reliability in predicting PE2 efficiencies in endogenous site.

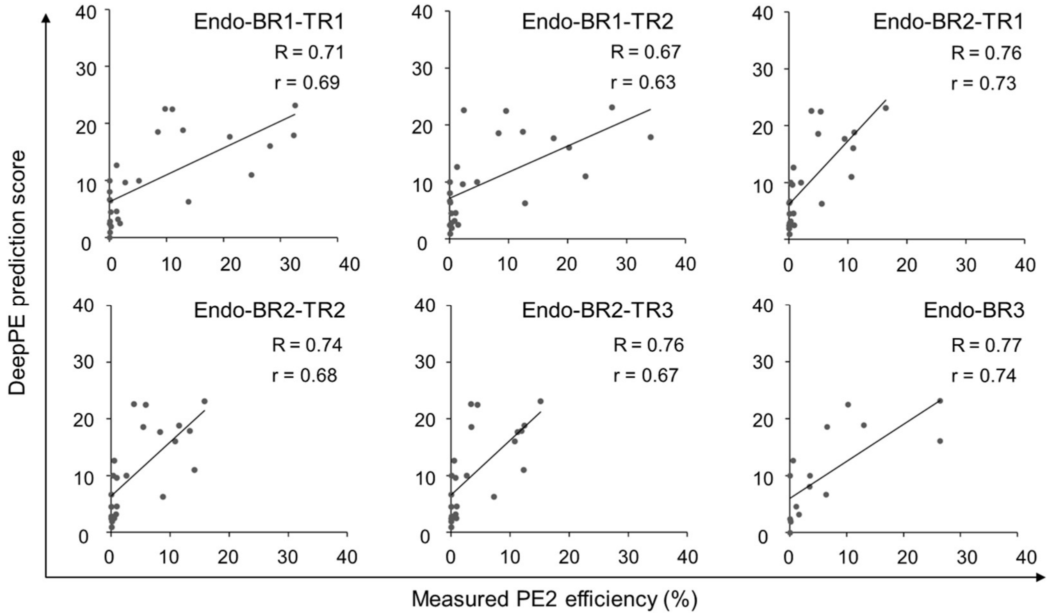

Figure 19. Benchmark of DeepPE using six datasets. Three biological replicates (BR1, BR2, and BR3) were evaluated in HEK293T cell and each biological replicate had one, two, or three technical replicates (TRs).

Further testing of DeepPE in HCT-116 (a colorectal carcinoma cell line) and MDA-MB-231 (a human breast adenocarcinoma cell line) on novel target sequences confirmed its robust performance across biological and technical replicates (HCT-116: R = 0.70 to 0.77 (average 0.74), r = 0.57 to 0.61 (average 0.59); MDA-MB-231: R = 0.76 to 0.81 (average 0.79), r = 0.62 to 0.65 (average 0.64)) (**Figure 20**).

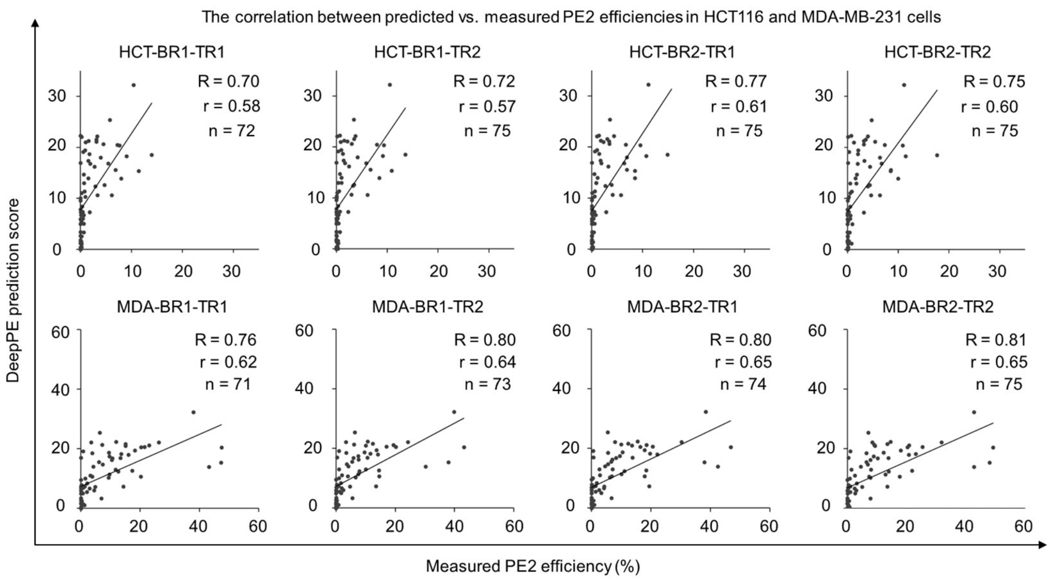

Figure 20. Evaluation of DeepPE using HCT-116 and MDA-MB-231 cells. Eight datasets of PE2 efficiencies were generated using HCT-116 (abbreviated as HCT) and MDA-MB-231 (abbreviated as MDA) cell lines at lentivirally integrated target sequences that were never used for the training of DeepPE. Two biological replicates (BR1 and BR2) per cell line were evaluated and each biological replicate had two technical replicates (TR1 and TR2).

We also evaluated the effectiveness of DeepPE in selecting the optimal PBS and RTT lengths out of 24 possible combinations for any given target sequence. The use of DeepPE led to an average absolute PE2 efficiency improvement of 1.2% and a relative efficiency improvement of 8.3% compared to initial recommendations based on a 13-nt PBS and a 12-nt RTT (**Figure 21**). DeepPE's advantage extends to selecting the most efficient target sequence among multiple options for a given edit.

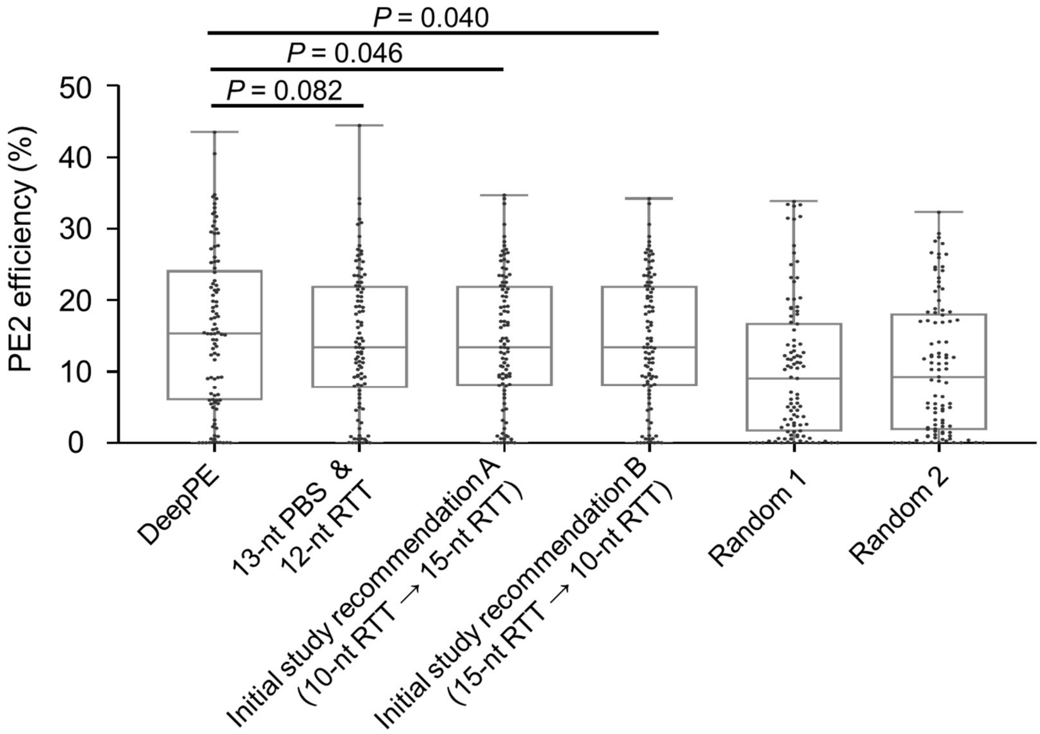

Figure 21. Performance comparison of DeepPE and other approaches. "13-nt PBS & 12-nt RTT" refers to choosing this combination of lengths regardless of the target sequence. Initial study recommendations A and B are based on using a 13-nt PBS and a 12-nt RTT (RTT) and avoiding a G as the last templated nucleotide by changing the RTT length as necessary. In recommendation A, if the last templated nucleotide is a G, then a 10-nt, rather than a 12-nt, RTT is chosen. If after this change the last templated nucleotide is again a G, then a 15-nt RTT is chosen. In recommendation B, if the last templated nucleotide is a G, then a 15-nt, rather than a 12-nt, RTT is chosen. If after this change the last templated nucleotide is again a G, then a 10-nt RTT is chosen. As controls, we also selected pegRNAs randomly (Random 1 and Random 2). Statistical significances determined by using the two-sided paired t-test are shown. The number of target sequences n = 97 per group.

Additionally, we developed two more computational models using data from Library-2 to predict PE2 efficiencies for various editing types and positions. The dataset was divided into Type-training, Type-test, Position-training, and Position-test sets, ensuring no overlap of target sequences (**Methods**). Random forest emerged as the most effective model for both type-based and position-based predictions, though the differences from other frameworks were not statistically significant (**Figure 22a-b**). Deep learning models showed limited performance, likely due to the smaller dataset size. The random forest-based models PE_type and PE_position demonstrated useful performance in predicting efficiencies (PE_type: R = 0.47, r = 0.48;

PE_position: R = 0.56, r = 0.56) (**Figure 22c-d**). Expanding the dataset to include more target sequences and diverse pegRNAs may improve model performance.

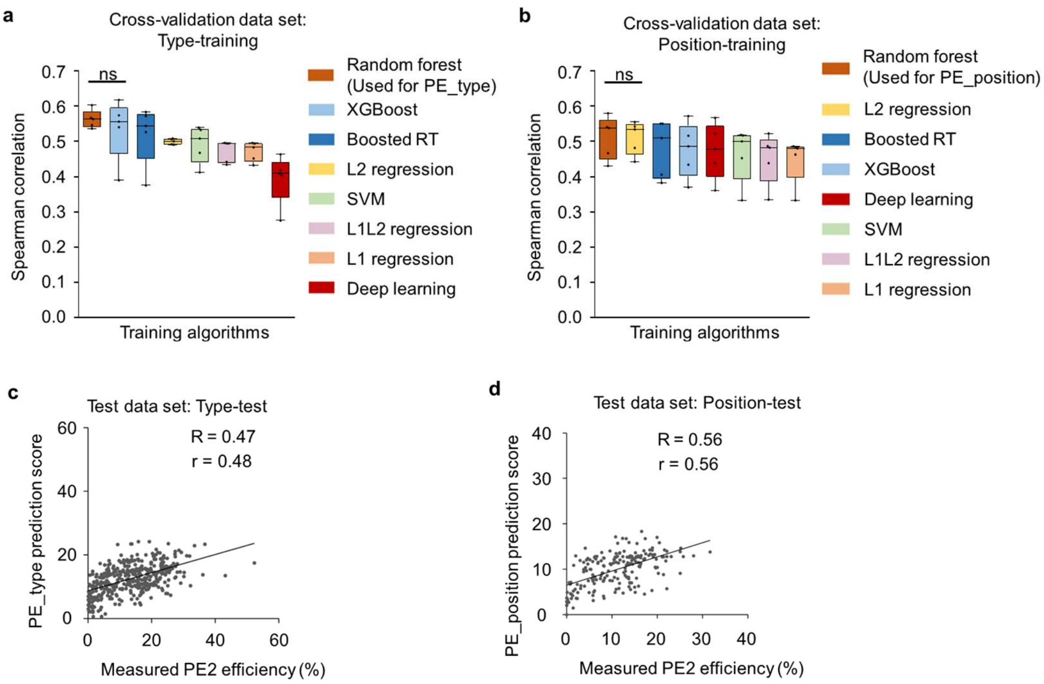

Figure 22. Development of PE_type and PE_position. (a-b) Cross-validation of predictive models based on different machine learning methodologies. Each point in the plot corresponds to the Spearman's correlation comparing the actual PE2 efficiency with predictions obtained from five-fold cross-validation (total, n = 5). Statistical comparisons are shown only between the top-performing model and the second-best one; NS indicates no significant difference; comparisons were made using a two-sided Steiger's test. (c-d) Assessment of models for PE_type (c) and PE_position (d).

We offer a web tool at http://deepcrispr.info/DeepPE that provides predictions from DeepPE, PE_type, and PE_position for any given target sequence. By entering a target sequence, users receive efficiency predictions for 57 pegRNAs, including 24 from DeepPE, 23 from PE_type (covering various deletions, insertions, and substitutions), and 10 from PE_position (covering substitutions at multiple positions).

3.2. Predicting prime editing efficiency in various PE systems

3.2.1. High-throughput evaluation of PE2 efficiencies using four pairwise libraries

To assess PE2 efficiencies on a large scale, we used lentiviral libraries that combined pegRNA sequences with their corresponding target sequences (**Figure 2**) and introduced them into HEK293T cells expressing PE2. We prepared four distinct libraries—Library-3, Library-4, Library-5, and Library-6 (see **Methods** and **Figure 23a**). These libraries were transduced into HEK293T cells that also expressed prime editor proteins, and editing efficiencies were evaluated through deep sequencing. The results from two independent experiments showed a high level of correlation, with Pearson correlation of 0.90 and 0.92 for Library-3 and Library-4, respectively, and Spearman correlation of 0.94 and 0.92 (**Figure 23b**).

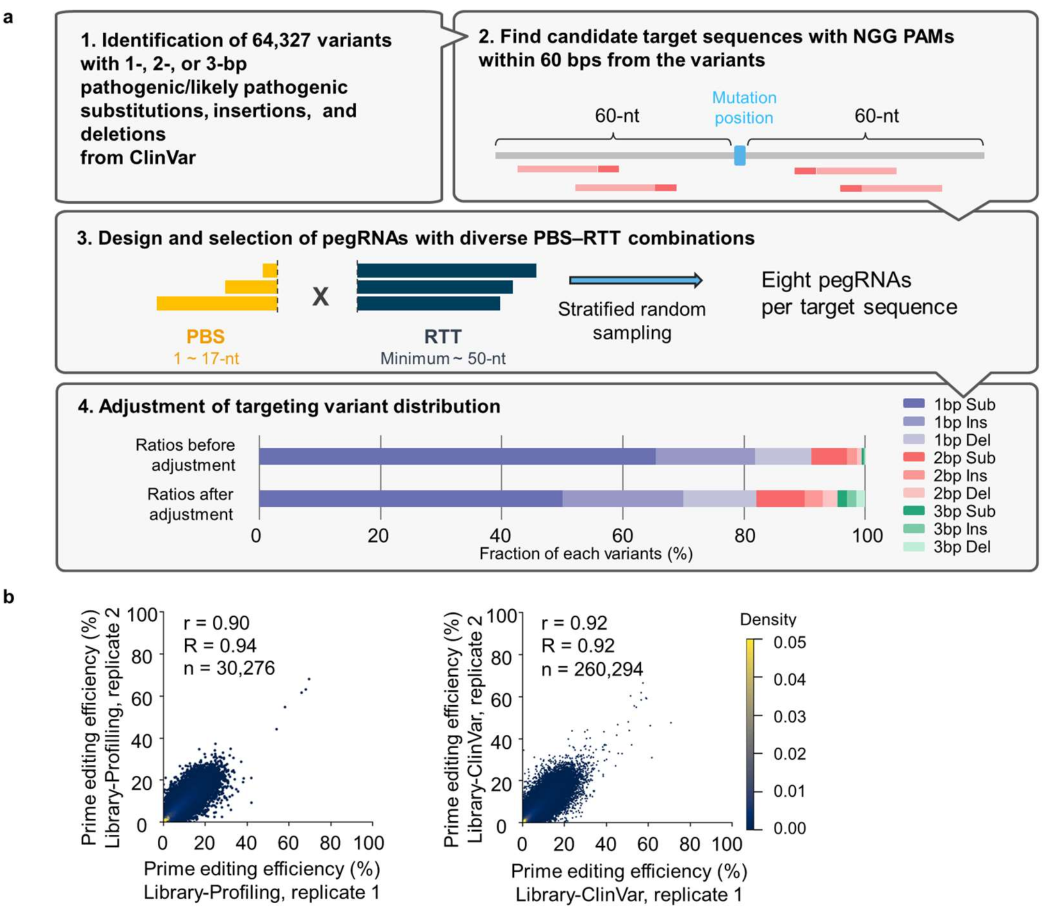

Figure 23. High-throughput assessment of prime editing efficiencies. (a) An overview of the process for selecting pegRNAs in Library-4 is illustrated. Pathogenic or likely pathogenic mutations, including substitutions, insertions, and deletions ranging from 1 to 3 base pairs, were extracted from the ClinVar database. Target sequences were selected, and pegRNAs were designed

with randomly assigned PBS and RTT lengths, ensuring a minimum RTT length based on the edit position. Since the ClinVar database predominantly contains single nucleotide variants, the number of pegRNAs designed for 2- and 3-bp edits was increased by decreasing the number of target sequences for 1-bp edits and incorporating randomly generated 3-bp variants. (b) The correlations between prime editing efficiencies observed in high-throughput experiments with Library-3 (left) and Library-4 (right) are shown. The color of each point reflects the density of surrounding points.

3.2.2. Analyses of factors influencing prime editing efficiency

We explored how the length of the PBS affects editing efficiency using pegRNAs with RTTs of 12 and 20 nucleotides. Our findings revealed that the highest average efficiencies were achieved with 11-nt (13%) and 12-nt (8.5%) long PBSs (**Figure 24a**), supporting previous results, though 12-nt PBSs had not been previously tested.

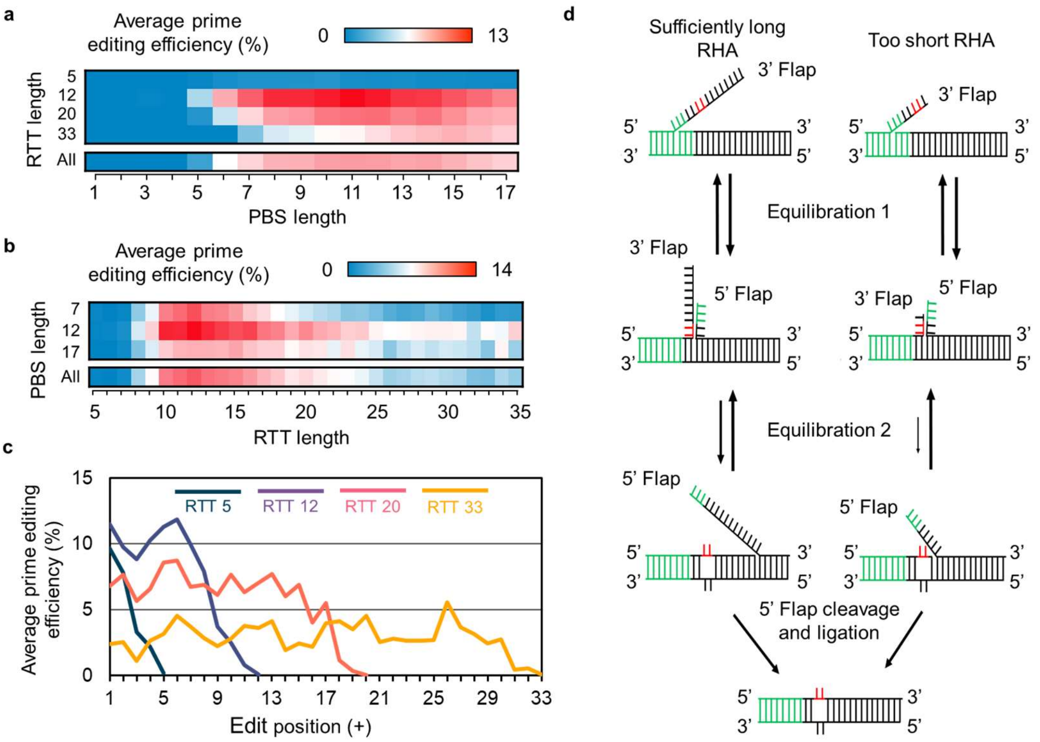

Figure 24. Effect of PBS, RTT, edit position, and RHA. Heatmaps illustrating the mean efficiencies for (a) PBSs with lengths ranging from 1 to 17 nucleotides and (b) RTTs spanning from 5 to 35 nucleotides.

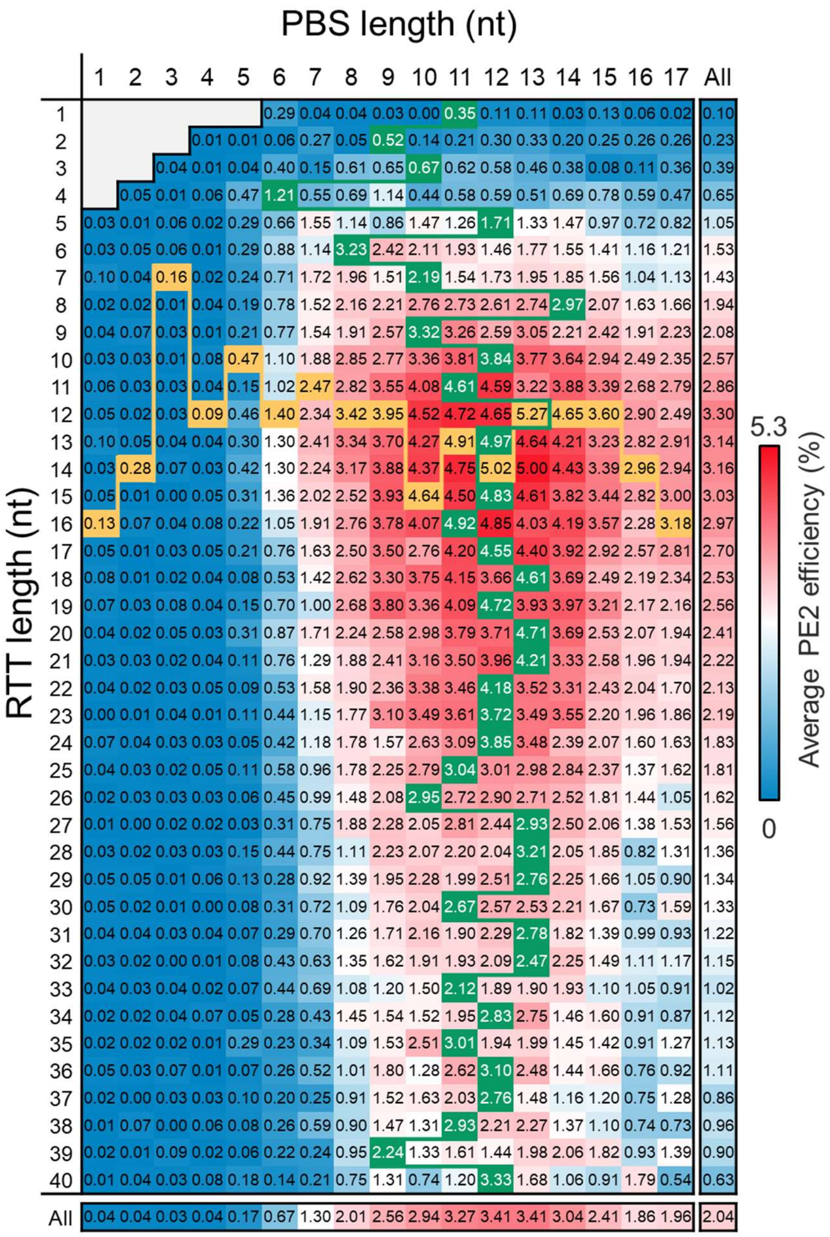

Figure 25. Effect of PBS and RTT on prime editing efficiency. A heatmap depicting the average prime editing efficiencies for various PBS and RTT lengths. In this map, yellow and green boxes represent the maximum average efficiencies observed for each PBS length when evaluated across all RTT lengths, and for each RTT length when tested with all PBS lengths, respectively.

A similar trend was observed in the analysis with Library-4 (**Figure 25**). Based on these observations, we recommend using an 11-nt PBS for RTTs of 12 nt or shorter and a 12-nt PBS for longer RTTs. Further analysis of RTT length indicated that the highest prime editing efficiencies were achieved with RTTs ranging from 10 to 14 nt in length (**Figure 24b**). This pattern was also seen in the Library-4 data (**Figure 25**) and aligns with earlier studies. We also examined the impact of editing position, noting that efficiency dropped sharply starting approximately 5 nucleotides before the end of the RTT (e.g., with a 12-nt RTT, a significant drop began around position +7) (**Figure 24c**). This suggests a need for a minimum length of the right homology arm (RHA). Unlike PBS and RTT lengths, the influence of RHA length has not been widely studied [1]. Short RHAs might favor the formation of a 3' flap over the necessary 5' flap for integrating edited sequences into the genome (**Figure 24d**)[1].

When investigating RHA length requirements across different editing types and positions, we found that 5-nt, 7-nt, and 9-nt RHAs were necessary for substitutions, insertions, and deletions, respectively (**Figure 26a-c**). Similar requirements were noted in Library-4 (**Figure 26d**). Based on these findings, we suggest using a minimum of 9-nt RHAs for all editing types and positions, although 7-nt RHAs could suffice in some cases. Additionally, we observed that overall prime editing efficiencies for substitutions were marginally higher compared to insertions and deletions, though this difference was not statistically significant (Figure 26d). A comparable trend was noted in prime editing at endogenous sites (**Figure 26e**).

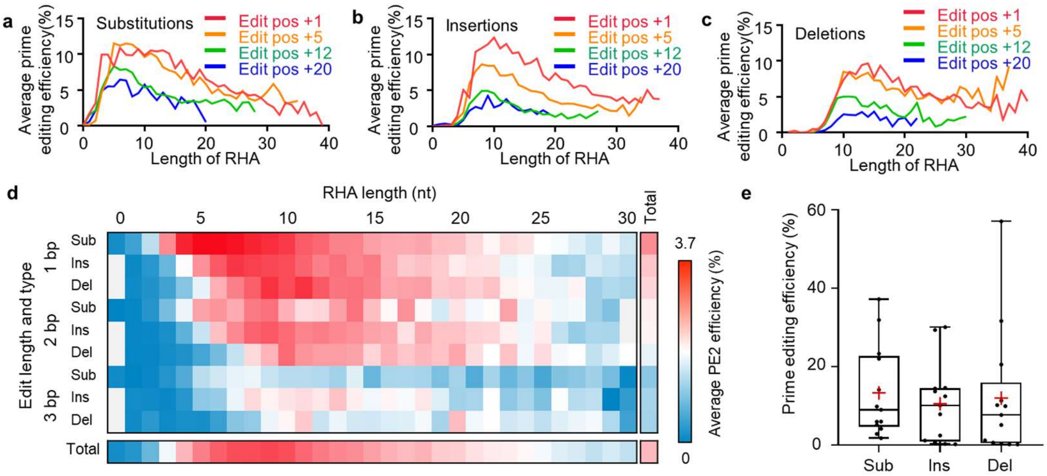

Figure 26. Effect of right homology arm length and the edit type on PE2 efficiency. (a-c) Average prime editing efficiency as a function of RHA length for different types of edits: substitution (a), insertion (b), and deletion (c). Editing positions are categorized as +1, +5, +12, and +20. (d) A heatmap illustrating the average prime editing efficiencies for 1- to 3-bp substitutions (Sub), insertions (Ins), and deletions (Del) across various RHA lengths. (e) Impact of the edit type on prime editing efficiency at endogenous sites. Each point represents the efficiency observed at a specific target sequence, with the total number of target sequences (n) = 13 for each type of edit.

We also evaluated how the number of edited nucleotides affects efficiency. Substitutions up to 3 nucleotides in length showed similar efficiencies, while those involving 4 to 10 nucleotides had reduced efficiencies as the number of edited nucleotides increased (**Figure 27a**). Similarly, efficiencies for insertions and deletions up to about 3 to 5 nucleotides were comparable, but decreased for longer sequences (**Figure 27b-c**). This indicates that PE2 efficiencies generally decline with an increase in the number of edited nucleotides, especially beyond three nucleotides.

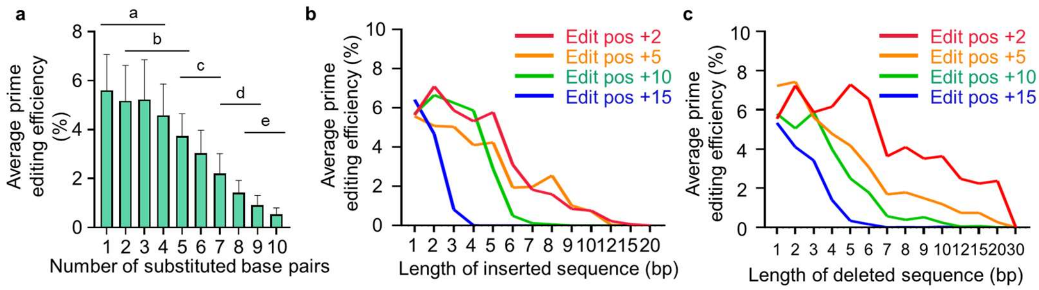

Figure 27. Impact of edit length of prime editing efficiency. (a) Influence of the number of base pairs edited on prime editing efficiency, with error bars representing the 95% confidence intervals. (b-c) Line graphs depicting the average prime editing efficiency for insertions (b) and deletions (c) of varying lengths.

The optimal nucleotides at the final templated position for efficient prime editing were found to be in the order: C > T > A > G, irrespective of editing type or RTT length (**Figure 28**).

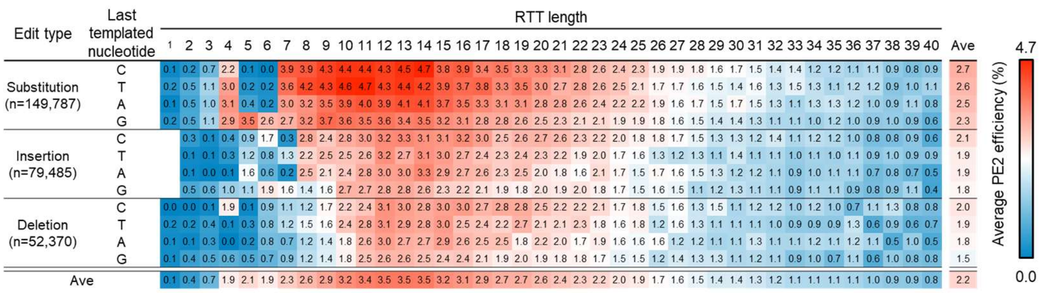

Figure 28. Impact of last templated nucleotide on prime editing efficiency. The heatmap illustrates how the nucleotide at the last templated position influences the average prime editing efficiencies. The pegRNAs from Library-4 were categorized based on RTT lengths and the type of edits they encode. Each cell in the heatmap reflects the average efficiency of editing.

We conducted SHAP analysis to identify key determinants of prime editing efficiency, considering both with and without multicollinearity (**Figure 29**). The GC content of the PBS was identified as the most critical factor, favoring higher GC counts and influencing the PBS length, melting temperature (Tm), and the number of C and G nucleotides. The RHA length was the second most significant feature, favoring longer RHAs (without multicollinearity) and higher Tm values (with multicollinearity), related to the GC content and count of the RHA. The DeepSpCas9 score at the target sequence was the third most important feature, consistent with previous studies (**Figure 11**). Although we determined the optimal range for each feature, the values often varied depending on other factors, complicating the manual design of efficient pegRNAs (**Figure 30**).

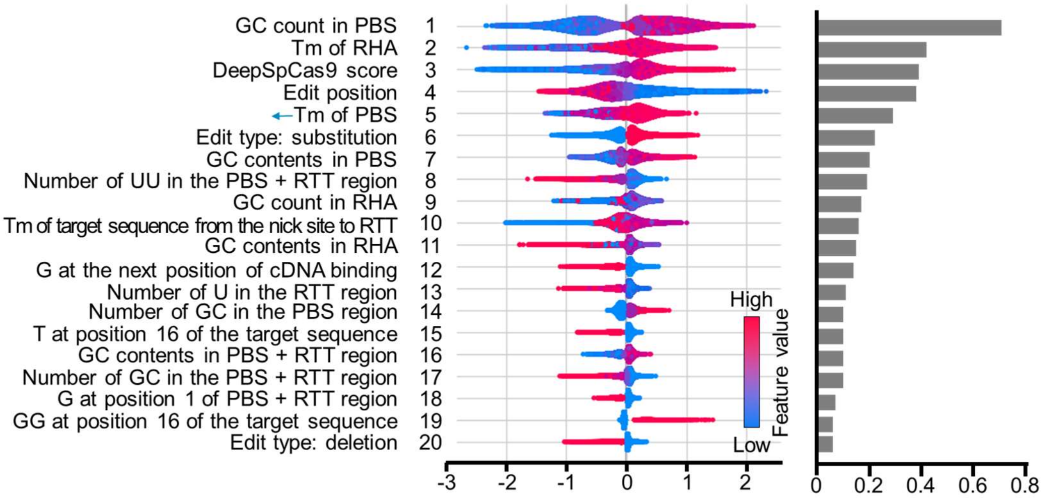

Figure 29. The features associated with various types of prime editing efficiencies. The features are assessed using Tree SHAP, with multicollinearity addressed by applying a correlation threshold of 0.7. Factors excluded because of multicollinearity are indicated in blue next to their related variables.

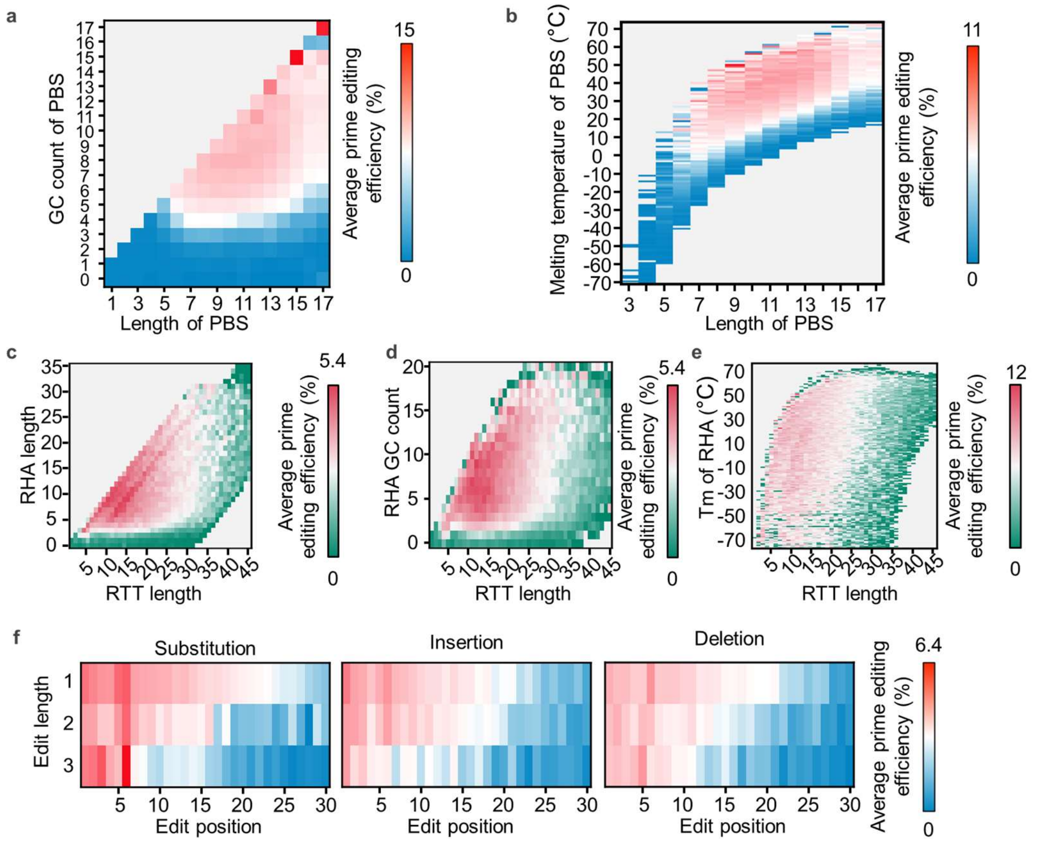

Figure 30. Factors influencing prime editing efficiency. (a-e) The impact of PBS-related characteristics (a and b) and RHA-related attributes (c-e) on the average efficiency of prime editing. Tm denotes the melting temperature. (f) The influence of editing position and length on average prime editing efficiencies for substitutions (left), insertions (center), and deletions (right).

3.2.3. Development of DeepPrime

Previously, our PE2-activity prediction models—DeepPE, PE_type, and PE_position (**Figure 18** and **Figure 22**)—were constrained by their training datasets, which limited their ability to predict various types of editing, positions, and lengths of PBS and RTT. To address these limitations, we utilized a comprehensive dataset from Library-4, which includes 288,793 pegRNA-target sequence pairs covering 850 different PBS and RTT length combinations. This dataset spans all edit types (substitutions, insertions, and deletions) across 1, 2, or 3 base pairs, and encompasses editing positions from +1 to +30. This data was divided into two subsets: CV-train (n = 259,910) and CV-test (n = 28,883), with no overlap in target sequences between the two sets. CV-train is 6.7 times larger than the dataset used for DeepPE and features a 35-fold broader range of PBS-RTT length combinations (**Figure 31a**), 582-fold more editing type and position combinations (**Figure 31b**), and 30 times more target sequences.

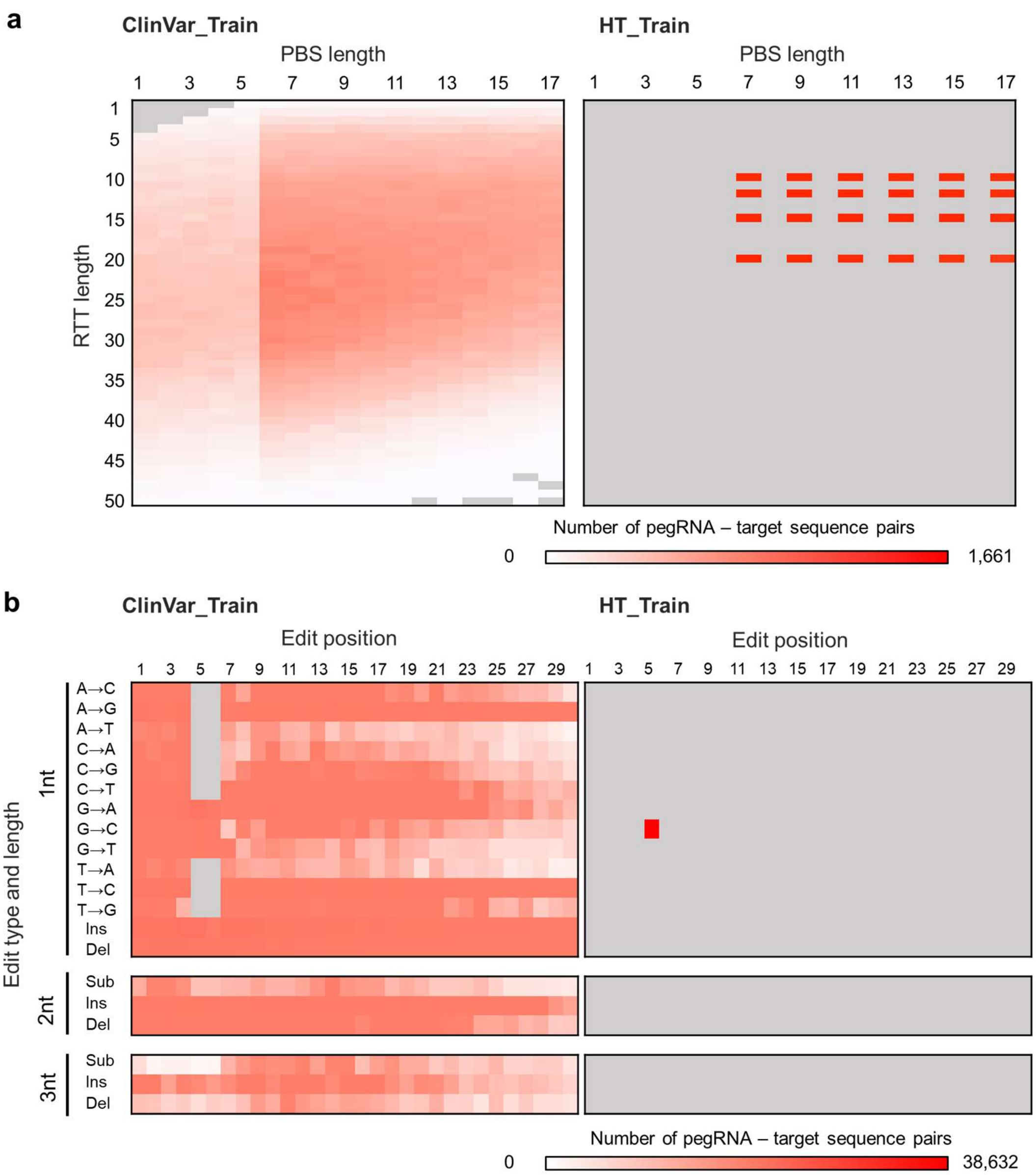

Figure 31. Comparing training datasets for DeepPrime and DeepPE. (a) Analysis of the variation in PBS-RTT combinations between CV-train and DPE_train datasets. (b) Examination of the ranges of edit types, lengths, and positions in both CV-train and DPE_train. (a-b) Heatmaps illustrate the quantity of pegRNA and target sequence pairs associated with each intended edit. Edits not included in the analysis are marked in gray.

We trained various machine learning and deep learning models using CV-train to predict PE2 efficiencies for different target sequences. Among the 15 algorithms tested, a convolutional neural network (CNN) combined with a gated recurrent unit (GRU)(**Figure 32a**) outperformed all other models, including the next best CNN with an attention module (P ≤ 9.3 x 10^-3 and 8.8 x 10^-3; Steiger's tests of Pearson's and Spearman's correlations) (**Figure 32b-c**). This model was named DeepPrime. Analysis of how training dataset size affected model performance revealed that accuracy stabilized once the dataset reached around 180,000 data points (**Figure 32d-e**). Nonetheless, prediction accuracies for deletions and 3-bp edits were relatively lower, likely due to their limited representation in the dataset despite our efforts to enrich it (**Figure 23a**).

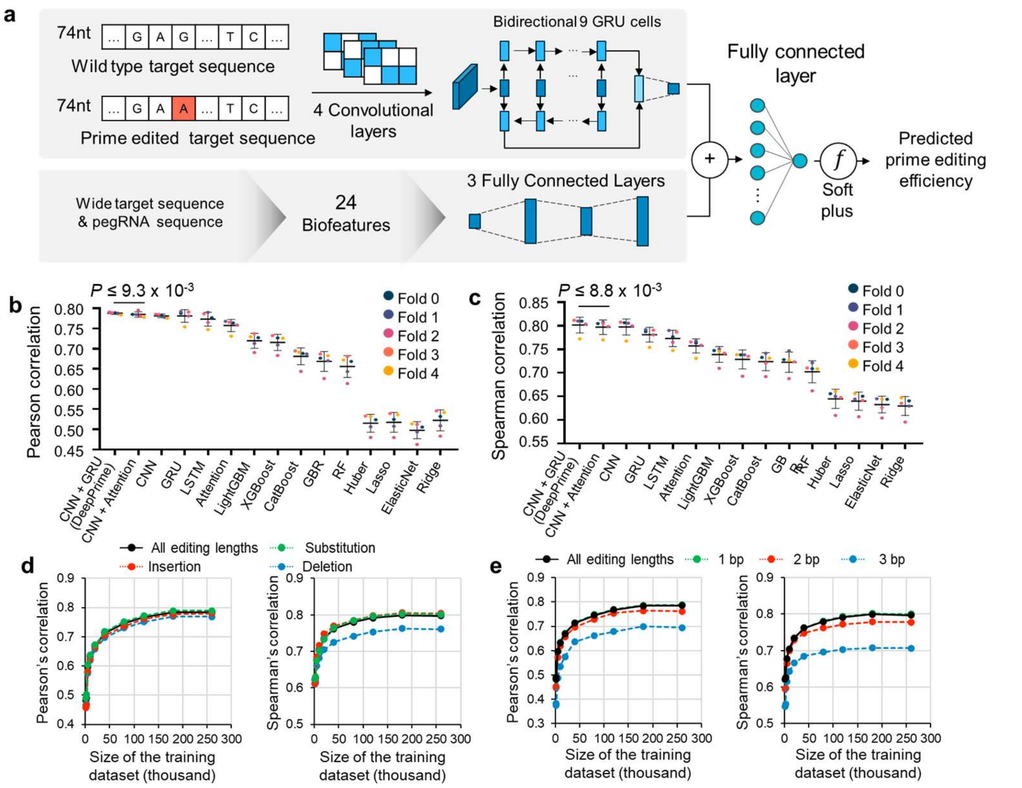

Figure 32. Development of DeepPrime. (a) Diagram illustrating the deep learning architecture utilized for creating DeepPrime. GRU, gated recurrent unit. (b-c) Evaluation of machine learning models for predicting prime editing efficiencies through cross-validation. Each point denotes the Pearson's (b) or Spearman's correlation (c) between actual and predicted efficiencies from five-fold cross-validation, with a total of 5 per analysis. Statistical comparisons between the top two algorithms are presented using two-sided Steiger's test. The bars and error bars represent the mean correlation and their standard deviations, respectively. Algorithms include CNN, GRU, LSTM, LightGBM, XGBoost, GBR, and RT. (d-e) The impact of model training on performance was assessed using smaller training datasets created by random subsampling. The results are shown for different edit types (d) and edit lengths (e), with y-axis values representing the averages from five-fold cross-validation.

To evaluate DeepPrime's effectiveness across different editing types, we analyzed its performance on nine subsets of CV-test, each representing a specific edit type (1-, 2-, or 3-bp substitutions, insertions, or deletions). The model demonstrated strong Pearson (r) and Spearman (R) correlation, ranging from 0.76 to 0.89 for Pearson and from 0.70 to 0.88 for Spearman (**Figure 33a**). Overall, DeepPrime achieved a Pearson correlation of 0.84 and a Spearman correlation of 0.86 when applied to CV-test across all editing types (**Figure 33b**). The model's performance varied by editing position, with Pearson correlations from 0.68 to 0.85 (average 0.79, median 0.80) for positions +1 to +30 and Spearman correlations from 0.63 to 0.88 for positions +1 to +27, with lower values from 0.47 to 0.58 for positions +28 to +30 (average 0.75, median 0.78) (**Figure 33c**). These findings suggest that DeepPrime provides accurate predictions of prime editing efficiencies across diverse editing types, lengths, and positions.

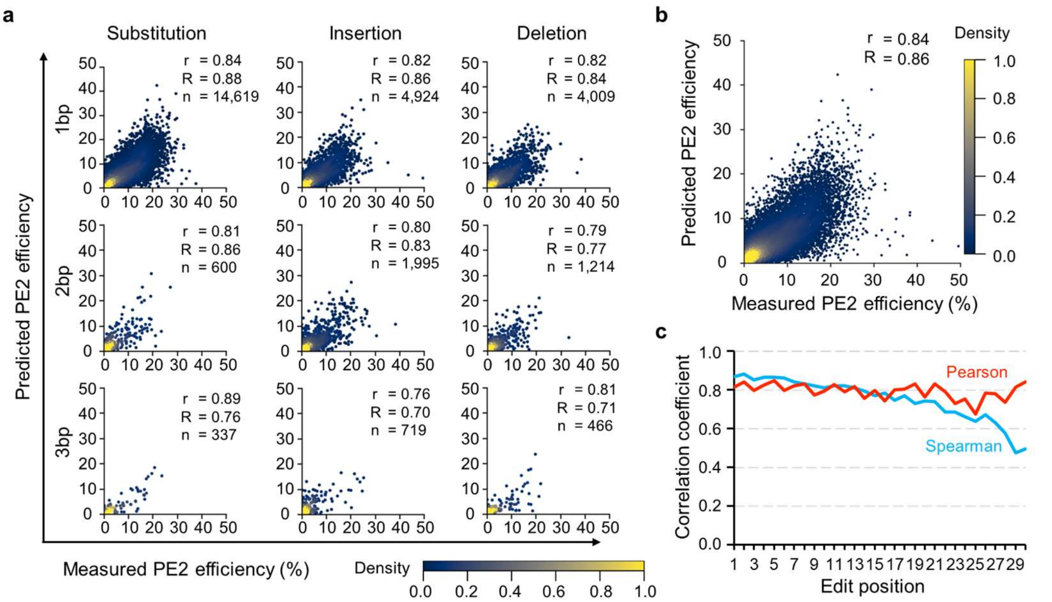

Figure 33. Assessment of DeepPrime with CV-test as the evaluation set. (a-b) Dot colors were assigned based on Kernel density estimation utilizing a Gaussian kernel. The CV-test dataset was divided into either 9 subsets based on edit type and length (b) or 30 subsets according to edit position (c) for model assessment. (c) The Pearson's and Spearman's correlation between observed and predicted PE2 efficiencies are represented by red and blue lines, respectively.

In previous high-throughput evaluations to identify the most efficient pegRNA for specific edits[3], DeepPrime's predictions correlated highly with experimental results (**Figure 34a**). Furthermore, for 100 possible pegRNAs targeting a given edit, DeepPrime recommended the same pegRNA that was experimentally validated. Additionally, when tested with PE2 efficiency data from an independent prime editing study at endogenous sites [1], DeepPrime showed high Pearson and Spearman correlation of r = 0.74 and R = 0.74, respectively, demonstrating its effectiveness in predicting PE2 efficiencies in endogenous contexts (**Figure 34b**).

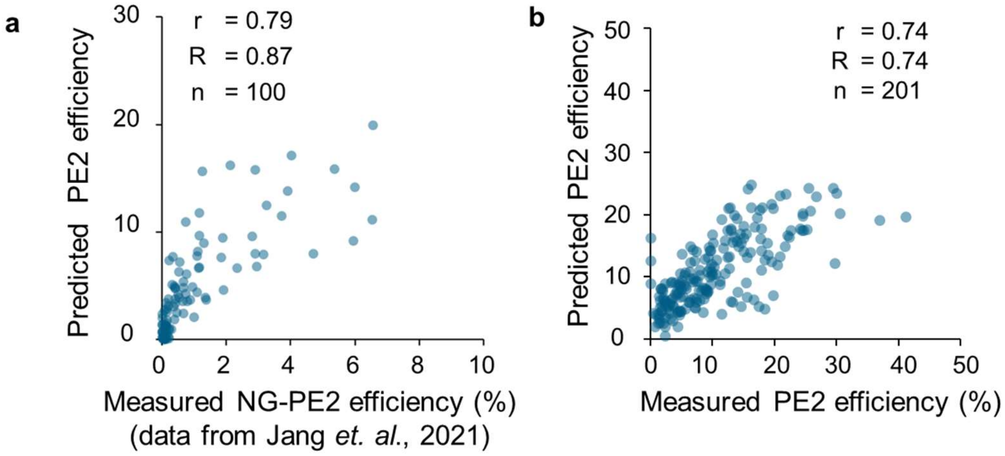

Figure 34. Assessment of DeepPrime using independent datasets. Previously recorded prime editing efficiencies were utilized for evaluation, including data from integrated target sequences (a, Jang et al.[3]) and endogenous sites (b, initial study by the Liu group on prime editing[1]).

3.2.4. Enhancing prime editing efficiency through optimized pegRNA scaffolds, PAM co-editing, and PE variants

We investigated the impact of an enhanced sgRNA scaffold—a 5-nt longer loop with TTTC instead of TTTT—on prime editing efficiency, drawing from research indicating improved Cas9 activity with such modifications[9]. Using Library-5, we compared the efficiency of pegRNAs with the standard scaffold versus those with the optimized scaffold. Results showed that optimized pegRNAs outperformed conventional ones in 79% of tested pegRNA-target pairs (1,674 out of 2,132), yielding a significant 1.25-fold increase in average efficiency (**Figure 35a**). This suggests that employing an optimized pegRNA scaffold can often boost prime editing performance. Additionally, co-editing of the NGG PAM site along with the intended edit has been shown to enhance editing efficiency [1]. We assessed the effects of such co-editing by utilizing pegRNAs with all 15 possible types of PAM co-editing. This approach resulted in a 1.7 to 1.2-fold increase in average prime editing efficiency, with the most substantial improvement observed when the NGG PAM was altered to NAT (**Figure 35b**).

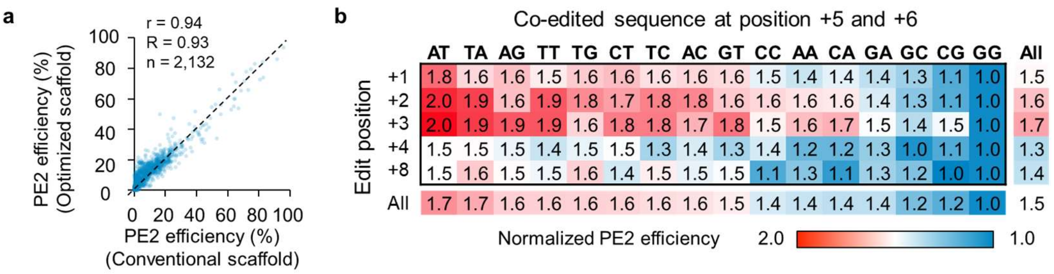

Figure 35. Enhancing PE2 efficiencies through optimized scaffold and PAM co-editing. (a)

Evaluation of PE2 efficiencies, comparing pegRNAs with conventional scaffolds versus those with optimized scaffolds. (b) Impact of PAM co-editing on PE2 efficiency, where average prime editing efficiencies with PAM co-editing are shown relative to those without it.

To extend the applicability of prime editing where NGG PAM sequences are scarce, we developed two alternative PE2 variants: SpCas9-NG (NG-PE2)[3] and SpCas9-NRCH (NRCH-PE2)[44]. Comparison of average prime editing efficiencies and nuclease-induced indel generation across these variants revealed high correlations (Pearson's r from 0.85 to 0.97 and Spearman's R from 0.75 to 0.89) (**Figure 36a-e**), indicating that the Cas9 variants and their associated PE systems have compatible PAM recognition profiles. We further evaluated the performance of PE2, NRCH-PE2, and NG-PE2 at target sequences containing 3-nt PAMs (NXXX, where X varies). We defined a PAM as effective if it resulted in an average prime editing efficiency greater than 1% seven days post-transduction of the pegRNA-target library (with optimized scaffold). The analysis showed that 12 of 64 (19%) PAM sequences were effective for PE2, 30 of 64 (47%) for NRCH-PE2, and 26 of 64 (41%) for NG-PE2 (**Figure 36f-h**). Overall, 35 out of 64 (55%) potential 3-nt PAM sequences were usable by at least one of the PE2 variants.

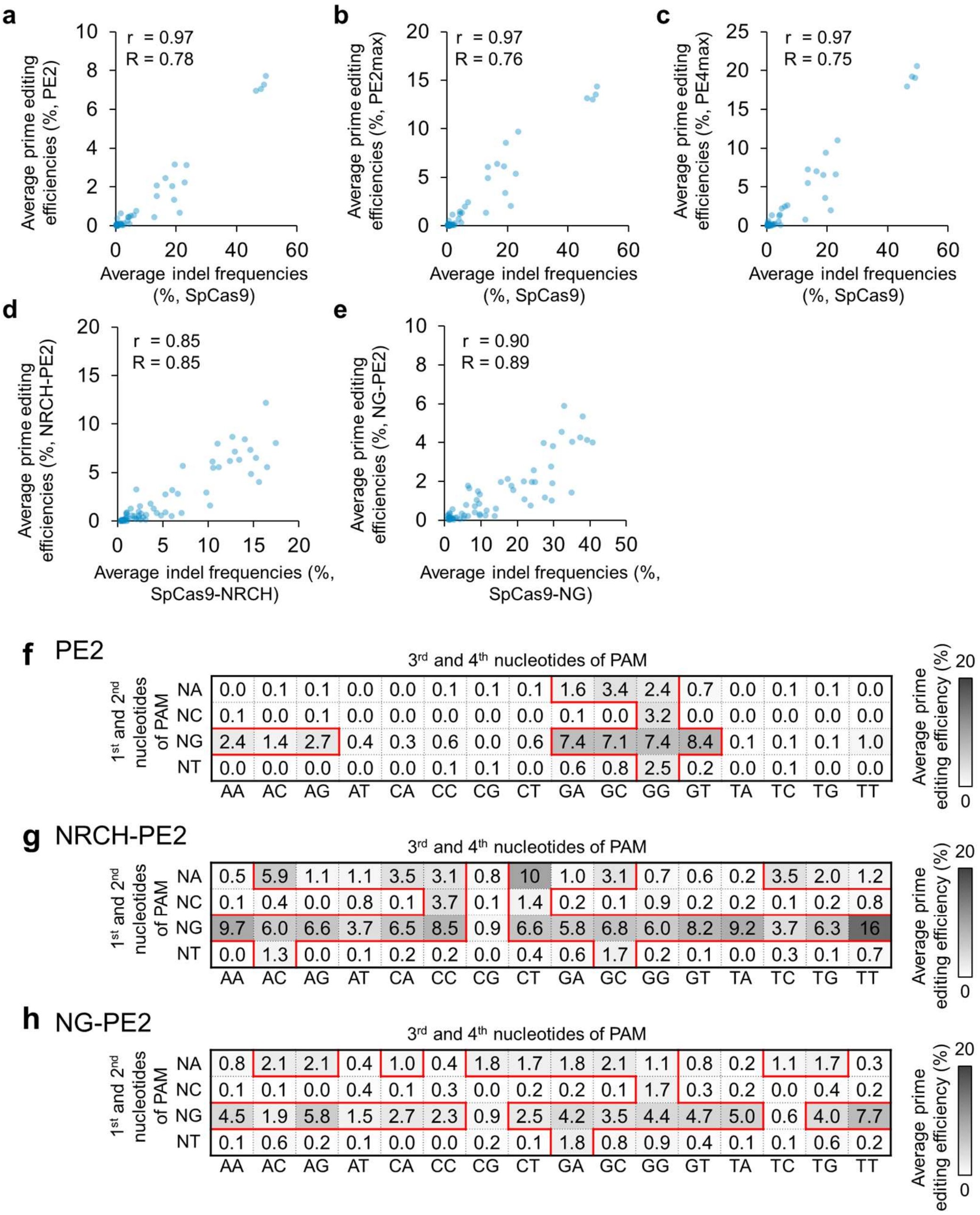

Figure 36. PAM compatibility analysis for PE2, NRCH-PE2, and NG-PE2. (a–e) The relationships between the activities of SpCas9 variants and PE2 modifications at target sequences with identical PAM motifs are examined. This includes average indel frequencies induced by SpCas9 variants and average prime editing efficiencies by PE2 variants across 64 different PAM

sequences (NXXX, with X varying). The total number of PAM sequences analyzed is n = 64. (f–h) Heatmaps displaying the average prime editing efficiencies achieved by PE2 (f), NRCH-PE2 (g), and NG-PE2 (h) at target sequences with 64 distinct PAM sequences. PAM sequences yielding average prime editing efficiencies exceeding 1% are highlighted in red.

The efficiency of prime editing can also be significantly improved by utilizing advanced prime editors such as PE2max and PE4max[2]. PE4max combines the enhanced PE2max with MLH1dn, an inhibitor of mismatch repair (MMR). Comparative analysis revealed strong correlations between PE2 and PE2max (Pearson's r = 0.91, Spearman's R = 0.96) and between PE2 and PE4max (r = 0.88, R = 0.96). PE2max and PE4max demonstrated 1.9-fold and 2.7-fold increases in efficiency, respectively, compared to PE2 in HEK293T cells (**Figure 37a-b**). Prime editing efficiency is cell-type dependent, likely due to variations in MMR component expression levels[2]. A comparison of efficiencies in HEK293T cells versus other cell types—HCT-116, MDA-MB-231, HeLa, A-549, DLD-1, and NIH-3T3—showed variable correlations (Pearson's r from 0.63 to 0.89, Spearman's R from 0.80 to 0.93) (**Figure 37c-i**), highlighting the influence of cell type on editing efficiency.

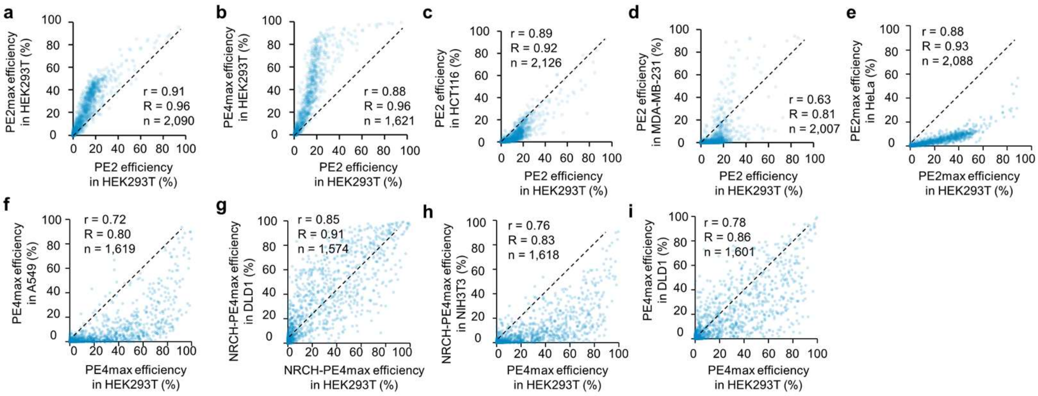

Figure 37. Comparison of prime editing types. (a-b) The relationship between prime editing efficiencies achieved with PE2 and PE2max (a), as well as PE4max (b), at target sites with NGG PAM sequences. (c–i) Comparisons of prime editing efficiencies between PE2, PE2max, PE4max, and NRCH-PE4max in HEK293T cells versus other cell types, specifically at target sites with NGG PAM sequences.

Moreover, while epegRNAs have been shown to enhance prime editing efficiency, the magnitude of improvement varies with cell type and prime editor[10]. In HEK293T cells and A-549 cells, the use of epegRNAs increased average PE2max efficiencies by 1.5-fold in HEK293T cells and by 4.1-fold and 0.89-fold for PE2max and PE4max, respectively, in A-549 cells (**Figure 38**).

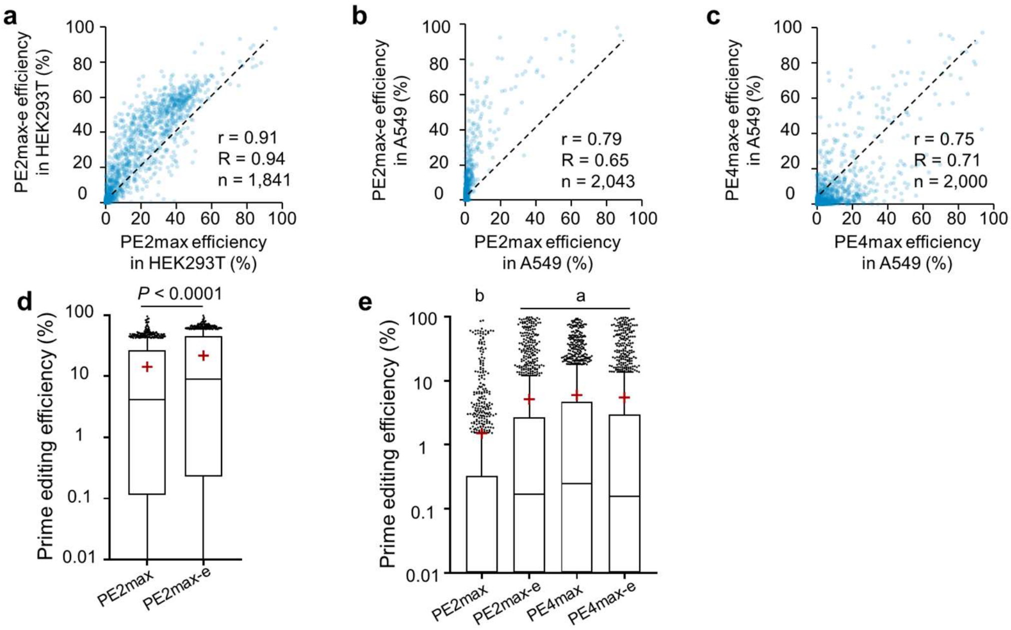

Figure 38. Comparison of prime editing outcomes with pegRNAs versus epegRNAs. (a–c) Correlations between prime editing efficiencies achieved with pegRNAs compared to epegRNAs. (d-e) Average prime editing efficiencies are marked with red plus signs. (d) Statistical significance is indicated through paired t-tests. Prime editing conducted with epegRNAs is denoted by appending "-e" to the prime editor's name.

3.2.5. Development of DeepPrime-FT

Accurately predicting the efficiency of prime editing across different prime editors and cell types can significantly aid in selecting the most suitable PE variant and pegRNA for diverse experimental scenarios. To address this need, we have developed a set of computational models designed to estimate prime editing efficiencies under various conditions. These models leverage eighteen datasets encompassing prime editing efficiencies for PE2, PE2max, PE2max-e (PE2 with epegRNAs), PE4max, PE4max-e (PE4max with epegRNAs), NRCH-PE2, NRCH-PE2max, and NRCH-PE4max, across seven distinct cell lines (HEK293T, HCT-116, DLD-1, MDA-MB-231, A-549, HeLa, and NIH-3T3). These datasets were created using Library-5 and were divided into training and test sets. We then fine-tuned the DeepPrime model to create a set of 18 distinct computational models, collectively referred to as DeepPrime-FT. Comparative performance analysis between the original DeepPrime and the fine-tuned DeepPrime-FT showed that while DeepPrime demonstrated reliable performance across various cell types, scaffolds, and prime editors (Pearson's r, average 0.58, median 0.57; Spearman's R, average 0.73, median 0.75), fine-tuning resulted in improved accuracy (Pearson's r, average 0.71, median 0.74; Spearman's R, average 0.80, median 0.82) (**Figure 39**).

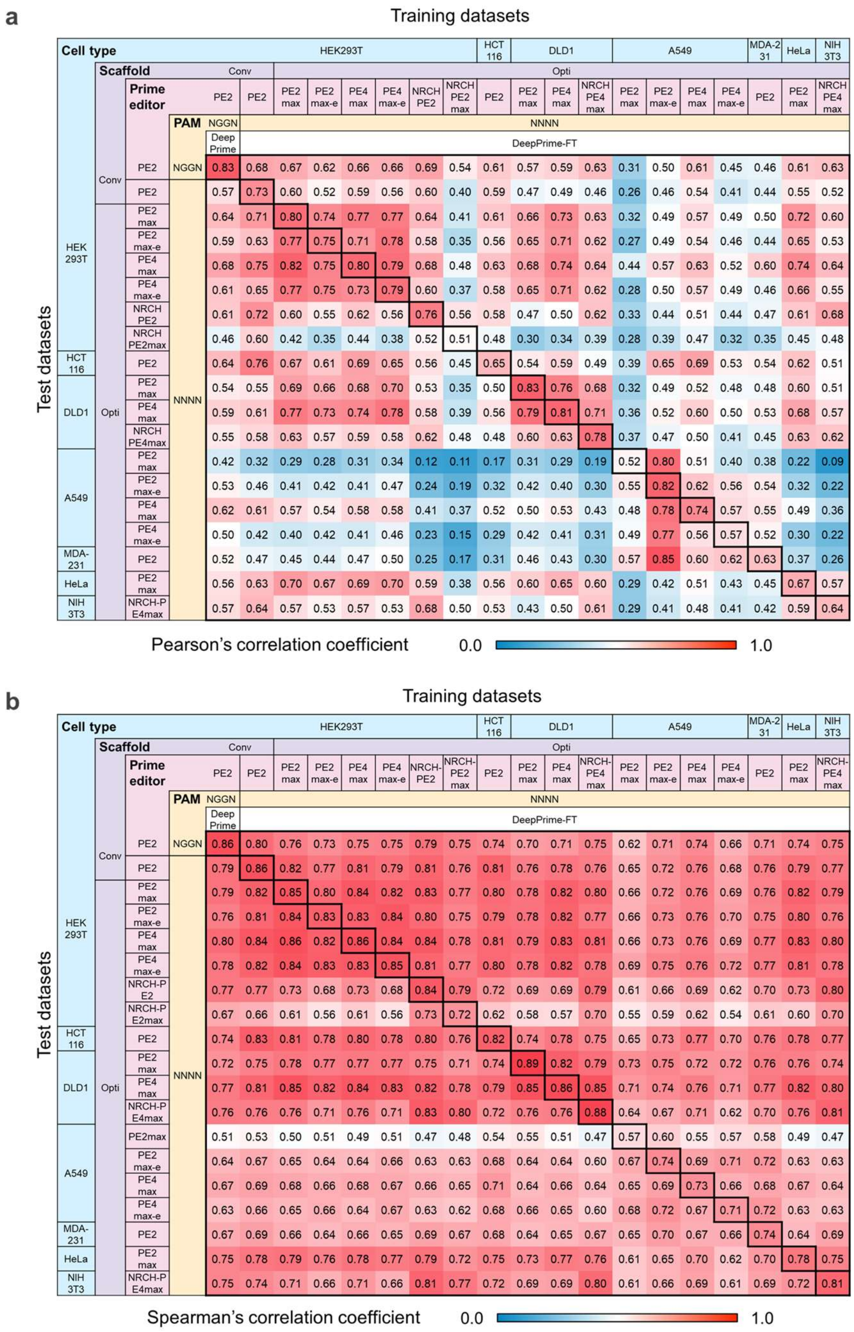

Figure 39. Development and performance of DeepPrime-FT. (a-b) Heatmaps displaying Pearson's (a) and Spearman's (b) correlation, illustrating how predicted versus actual prime editing efficiencies vary based on experimental factors such as cell type, pegRNA scaffold, prime editor,

and PAM sequence. Prime editing with epegRNAs is denoted by appending "-e" to the prime editor's name. Each highlighted box represents the correlation for a model assessed using a test dataset that mirrors the experimental conditions of the training dataset.

Our analysis revealed that the Spearman correlation for DeepPrime were generally higher than the Pearson across different cell types, scaffolds, and prime editor systems. This suggests that DeepPrime is better at preserving the rank order of pegRNA efficiencies than capturing their precise relative values. SHAP analysis indicated that key factors influencing prime editing efficiency were consistent across cell types and prime editor systems, although the specific effects varied slightly (data not shown). This consistency supports the effectiveness of DeepPrime across different conditions. Researchers can thus rely on DeepPrime for general predictions of pegRNA efficiencies across various experimental settings. However, for enhanced accuracy, particularly when specific experimental conditions are critical, using the fine-tuned DeepPrime-FT models tailored to particular cell types, scaffolds, and prime editor versions is recommended.

3.2.6. Applications of DeepPrime

To evaluate the utility of DeepPrime and DeepPrime-FT in generating and correcting pathogenic mutations documented in ClinVar, we employed these tools alongside other pegRNA design strategies. Specifically, we designed pegRNAs using three different methods: (i) predictions from DeepPrime, (ii) rational design based on known characteristics of highly efficient pegRNAs, and (iii) design using DeepSpCas9 scores[7], which predict Cas9 activity and correlate with prime editing efficiency. Random pegRNA designs were used as negative controls. When comparing these approaches for correcting mutations, DeepPrime significantly outperformed the other methods, with mean and median expected prime editing efficiencies of 9.0% and 7.6%, respectively. This was markedly higher than the efficiencies from rational design (4.6% and 2.4%), DeepSpCas9 score-based design (3.2% and 1.1%), and random designs (both 1.2% and 0.3%) (**Figure 40a**). Similar trends were observed for generating mutations, where DeepPrime again led with mean and median efficiencies of 10% and 9.2% (**Figure 40b**). The efficiency boost offered by DeepPrime—up to 3.2-fold higher for correcting mutations and up to 2.9-fold higher for generating mutations compared to rational designs—demonstrates its potential for designing highly efficient pegRNAs.

In a recent study by Erwood et al., prime editing was used to introduce variants in NPC1 and BRCA2[45]. We found that using DeepPrime-designed pegRNAs could have increased the average efficiency of variant generation by 2.1-fold compared to their rationally designed counterparts (**Figure 40c**). Furthermore, if NRCH-PE2 combined with DeepPrime-FT or PE2max with DeepPrime-FT had been utilized, the efficiency could have been increased by 4.5-fold or 7.7-fold, respectively, relative to the rational design using PE2. These findings suggest that combining DeepPrime or DeepPrime-FT with optimized prime editors can significantly enhance the functional study of genetic variants. We also examined the efficiency ranking of DeepPrime-selected pegRNAs among eight possible designs for specific edits in CV-test. Notably, 50% of the DeepPrime-selected pegRNAs ranked first, and 25% ranked second (**Figure 40d**). In comparison, when applying similar analysis to previously conducted experiments at endogenous sites, 56% of DeepPrime-selected pegRNAs ranked within the top 10%, and 11% ranked within the top 20% (**Figure 40e**).

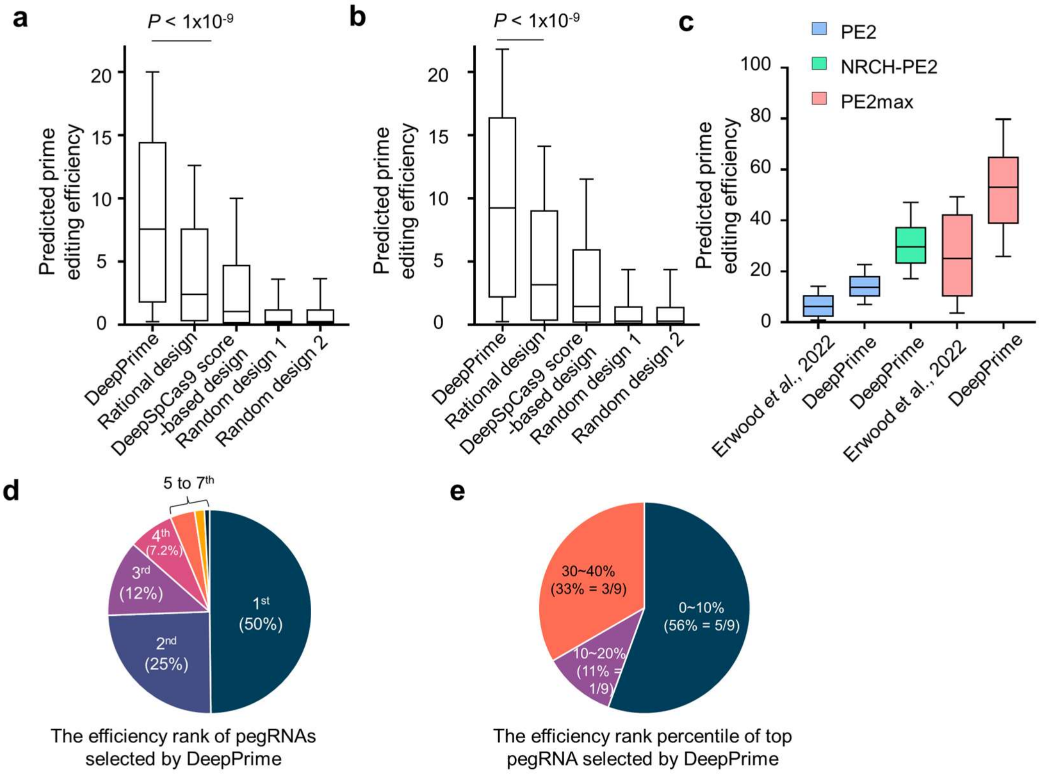

Figure 40. Applications of DeepPrime. (a-b) Projected efficiencies of pegRNAs designed through DeepPrime, a method leveraging the characteristics of highly effective pegRNAs, the DeepSpCas9 score, or random selection, aimed at correcting (a) and generating (b) pathogenic or likely pathogenic mutations documented in ClinVar. The dataset includes n = 64,327 pegRNAs per design. (c) Forecasted editing efficiencies of pegRNAs selected via DeepPrime, DeepPrime-FT, and a previously employed rational design approach (Erwood et al.[45]), for introducing alterations into NPC1 and BRCA2. This includes n = 426 pegRNAs per design. (d) The distribution of measured prime editing efficiency ranks for pegRNAs with the highest DeepPrime scores among the top eight pegRNAs per target, with a total of n = 845 pegRNA sets. (e) The distribution of measured prime editing efficiency percentile ranks for pegRNAs with the highest DeepPrime scores across all tested pegRNAs per target, with n = 9 pegRNA sets.

Further, we compared the prime editing efficiencies at endogenous sites predicted by DeepPrime-FT with those from earlier designs. DeepPrime-FT predicted a 2.3-fold increase in average PE2max efficiencies compared to previously published designs. Experimentally, pegRNAs designed using DeepPrime in HEK293T cells achieved a 3.5-fold higher average efficiency than those from earlier designs (**Figure 41a**). The correlation between observed prime editing efficiencies at endogenous sites and DeepPrime-FT predictions was robust (r = 0.82 and R

= 0.83) (**Figure 41b**), underscoring the effectiveness of DeepPrime-FT in designing efficient prime editing strategies.

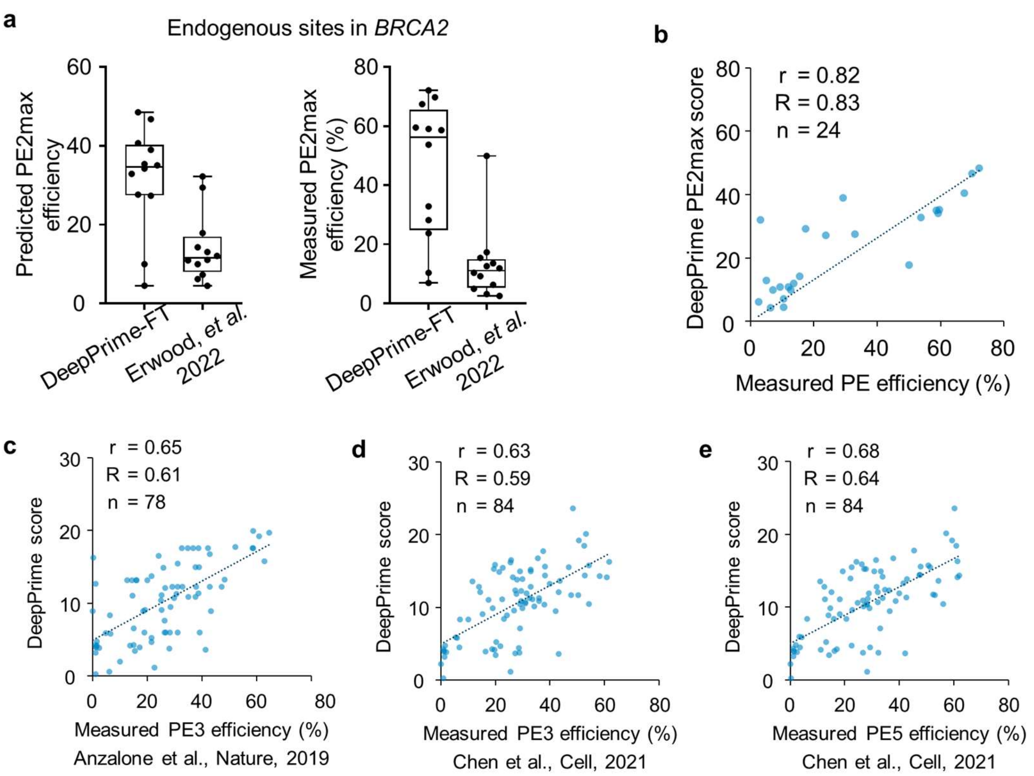

Figure 41. Validation of DeepPrime performance for application. (a-b) Comparison of forecasted versus experimentally observed prime editing efficiencies at endogenous BRCA2 sites. The data include n = 12 pairs of pegRNA and target sequences for both DeepPrime-FT and Erwood et al., 2022. Each value represents the average of duplicate measurements. (c-e) Analysis of the correlation between predicted and actual prime editing efficiencies for PE3 and PE5 at endogenous sites.

Finally, we explored whether DeepPrime could predict the efficiencies of untested prime editing systems like PE3 and PE5 at endogenous sites using previously published data . While DeepPrime demonstrated good predictive performance for PE3 (r = 0.65, 0.63; R = 0.61, 0.59) and PE5 (r = 0.68; R = 0.64), its accuracy was slightly lower than for systems it had been explicitly trained on (**Figure 41c-e**). This may be due to the additional influence of sgRNA activities, which are not predicted by DeepPrime, on the efficiency of these systems.

4. Discussion

When targeting a specific genetic edit, there are often multiple possible target sequences located near the intended edit site. For each of these sequences, it is theoretically possible to design over 850 pegRNAs, considering the combination of 17 PBS lengths and 50 RTT lengths. If four such potential target sequences exist, the total number of possible pegRNAs increases to 3,400. With DeepPrime and DeepPrime-FT, it is feasible to predict the efficiency of thousands of these pegRNAs, allowing researchers to identify the most effective ones within minutes, without the need for time-consuming laboratory experiments.

In this study, we generated a comprehensive dataset of prime editing efficiencies with high accuracy and precision. We systematically analyzed the factors influencing prime editing success and developed computational models capable of predicting editing efficiencies across different cell types and prime editors. Additionally, we evaluated prime editing efficiencies at sequences with mismatches and created a predictive model for these scenarios.

While our research offers tailored predictions for eight prime editing systems in seven different cell lines, it does not cover many other cell types, such as primary cultured cells and pluripotent stem cells. Furthermore, we did not assess prime editing efficiencies in living animals or embryos. The use of SpCas9-NRCH instead of wild-type SpCas9 expanded the range of targetable sequences; employing other Cas9 variants with different PAM preferences could further extend this range. Although we primarily used lentiviral vectors to deliver pegRNAs and prime editing proteins, other delivery methods—such as adeno-associated viruses, lipid nanoparticles, virus-like particles[46,47], or the electroporation of ribonucleoprotein complexes[48]—could influence editing efficiency at both matched and mismatched target sequences.

Moreover, our predictions did not extend to prime editing systems that use dual pegRNAs (like TwinPE[49], Prime-Del[50], and HOPE[51]) or those that combine a pegRNA with an sgRNA (like PE3 and PE5). However, it is possible that the efficiency of these systems could be estimated by separately predicting the performance of each pegRNA or the combination of sgRNA and pegRNA. Finally, while our computational models are robust, their accuracy could be further enhanced with larger training datasets, particularly those enriched with specific types of edits, such as 3-bp modifications or deletions.

5. Conclusion

The prime editor can be considered a "universal genome-editing tool" when it comes to its ability to correct genes. However, with the improvement in its capabilities, the complexity of the tool has also significantly increased, offering a wide range of options that users can choose from. While the traditional CRISPR-Cas9 genome-editing tool only required determining the location of the genome to cut, the prime editor allows for precise control over the type, location, and length of the desired genetic edits.

This study has enabled a deeper understanding of the complex prime editing system by leveraging extensive experimental data. We have gained insights into the biochemical factors that influence prime editing, providing clues on how to design new prime editors to enhance editing efficiency. Moreover, the AI models developed in this research have significantly lowered the barrier for researchers, allowing them to utilize prime editing without the need for extensive experimental work or background knowledge. Our research has demonstrated that it is possible to design a prime editor that is much more efficient than those used in previous studies, suggesting that prime editing could be effectively applied in research and treatment of mutations related to genetic diseases.

The development of genome-editing technology has progressed at an astonishing pace. It has been about a decade since the invention of CRISPR genome-editing tools, and the first therapy based on this technology has already received FDA approval and is being marketed. Although the next-generation genome-editing tool, the prime editor, has not yet been clinically applied, we expect numerous attempts to bring this technology into therapeutic use in the near future. Given that it is a new technology, many aspects of its efficiency and safety still need to be validated. This will likely be a challenge for all researchers to tackle together moving forward.

References

1. Anzalone, A.V., Randolph, P.B., Davis, J.R., Sousa, A.A., Koblan, L.W., Levy, J.M., et al. (2019). Search-and-replace genome editing without double-strand breaks or donor DNA. Nature *576*, 149-157.
2. Chen, P.J., Hussmann, J.A., Yan, J., Knipping, F., Ravisankar, P., Chen, P.F., et al. (2021). Enhanced prime editing systems by manipulating cellular determinants of editing outcomes. Cell *184*, 5635-5652 e5629.
3. Jang, H., Jo, D.H., Cho, C.S., Shin, J.H., Seo, J.H., Yu, G., et al. (2022). Application of prime editing to the correction of mutations and phenotypes in adult mice with liver and eye diseases. Nat Biomed Eng *6*, 181-194.
4. Doman, J.L., Sousa, A.A., Randolph, P.B., Chen, P.J., and Liu, D.R. (2022). Designing and executing prime editing experiments in mammalian cells. Nat Protoc *17*, 2431-2468.
5. Zou, J., Maeder, M.L., Mali, P., Pruett-Miller, S.M., Thibodeau-Beganny, S., Chou, B.K., et al. (2009). Gene targeting of a disease-related gene in human induced pluripotent stem and embryonic stem cells. Cell Stem Cell *5*, 97-110.
6. Zafra, M.P., Schatoff, E.M., Katti, A., Foronda, M., Breinig, M., Schweitzer, A.Y., et al. (2018). Optimized base editors enable efficient editing in cells, organoids and mice. Nat Biotechnol *36*, 888-893.
7. Kim, H.K., Kim, Y., Lee, S., Min, S., Bae, J.Y., Choi, J.W., et al. (2019). SpCas9 activity prediction by DeepSpCas9, a deep learning-based model with high generalization performance. Sci Adv *5*, eaax9249.
8. Landrum, M.J., Lee, J.M., Benson, M., Brown, G., Chao, C., Chitipiralla, S., et al. (2016). ClinVar: public archive of interpretations of clinically relevant variants. Nucleic Acids Res *44*, D862-868.
9. Dang, Y., Jia, G., Choi, J., Ma, H., Anaya, E., Ye, C., et al. (2015). Optimizing sgRNA structure to improve CRISPR-Cas9 knockout efficiency. Genome Biol *16*, 280.
10. Nelson, J.W., Randolph, P.B., Shen, S.P., Everette, K.A., Chen, P.J., Anzalone, A.V., et al. (2022). Engineered pegRNAs improve prime editing efficiency. Nat Biotechnol *40*, 402-410.
11. Kim, H.K., Lee, S., Kim, Y., Park, J., Min, S., Choi, J.W., et al. (2020). High-throughput analysis of the activities of xCas9, SpCas9-NG and SpCas9 at matched and mismatched target sequences in human cells. Nat Biomed Eng *4*, 111-124.
12. Shen, J.P., Zhao, D., Sasik, R., Luebeck, J., Birmingham, A., Bojorquez-Gomez, A., et al. (2017). Combinatorial CRISPR–Cas9 screens for de novo mapping of genetic interactions. Nature methods *14*, 573-576.
13. Du, D., Roguev, A., Gordon, D.E., Chen, M., Chen, S.H., Shales, M., et al. (2017). Genetic interaction mapping in mammalian cells using CRISPR interference. Nat Methods *14*, 577-580.
14. Shalem, O., Sanjana, N.E., Hartenian, E., Shi, X., Scott, D.A., Mikkelson, T., et al. (2014). Genome-scale CRISPR-Cas9 knockout screening in human cells. Science *343*, 84-87.
15. Kim, N., Kim, H.K., Lee, S., Seo, J.H., Choi, J.W., Park, J., et al. (2020). Prediction of the sequence-specific cleavage activity of Cas9 variants. Nat Biotechnol *38*, 1328-1336.
16. Kim, H.K., Song, M., Lee, J., Menon, A.V., Jung, S., Kang, Y.M., et al. (2017). In vivo high-throughput profiling of CRISPR-Cpf1 activity. Nat Methods *14*, 153-159.
17. Song, M., Kim, H.K., Lee, S., Kim, Y., Seo, S.Y., Park, J., et al. (2020). Sequence-specific

prediction of the efficiencies of adenine and cytosine base editors. Nat Biotechnol *38*, 1037-1043.
18. Cock, P.J., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., et al. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics *25*, 1422-1423.
19. Lorenz, R., Bernhart, S.H., Honer Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P.F., et al. (2011). ViennaRNA Package 2.0. Algorithms Mol Biol *6*, 26.
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research *12*, 2825-2830.
21. Ali, M. (2020). PyCaret: An open source, low-code machine learning library in Python.
22. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., et al. (2015). Human-level control through deep reinforcement learning. nature *518*, 529-533.
23. Abadi, M. (2016). TensorFlow: learning functions at scale. pp. 1-1.
24. Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
25. Hendrycks, D., and Gimpel, K. (2016). Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.
26. Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. (pmlr), pp. 448-456.
27. Loshchilov, I., and Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.
28. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., et al. (2020). From Local Explanations to Global Understanding with Explainable AI for Trees. Nat Mach Intell *2*, 56-67.
29. Allen, F., Crepaldi, L., Alsinet, C., Strong, A.J., Kleshchevnikov, V., De Angeli, P., et al. (2018). Predicting the mutations generated by repair of Cas9-induced double-strand breaks. Nat Biotechnol.
30. Kim, H.K., Min, S., Song, M., Jung, S., Choi, J.W., Kim, Y., et al. (2018). Deep learning improves prediction of CRISPR-Cpf1 guide RNA activity. Nat Biotechnol *36*, 239-241.
31. Shen, M.W., Arbab, M., Hsu, J.Y., Worstell, D., Culbertson, S.J., Krabbe, O., et al. (2018). Predictable and precise template-free CRISPR editing of pathogenic variants. Nature *563*, 646-651.
32. Wang, D., Zhang, C., Wang, B., Li, B., Wang, Q., Liu, D., et al. (2019). Optimized CRISPR guide RNA design for two high-fidelity Cas9 variants by deep learning. Nat Commun *10*, 4284.
33. Schlub, T.E., Smyth, R.P., Grimm, A.J., Mak, J., and Davenport, M.P. (2010). Accurately measuring recombination between closely related HIV-1 genomes. PLoS Comput Biol *6*, e1000766.
34. Sack, L.M., Davoli, T., Xu, Q., Li, M.Z., and Elledge, S.J. (2016). Sources of Error in Mammalian Genetic Screens. G3 (Bethesda) *6*, 2781-2790.
35. Feldman, D., Singh, A., Garrity, A.J., and Blainey, P.C. (2018). Lentiviral co-packaging mitigates the effects of intermolecular recombination and multiple integrations in pooled genetic screens. bioRxiv, 262121.
36. Hill, A.J., McFaline-Figueroa, J.L., Starita, L.M., Gasperini, M.J., Matreyek, K.A., Packer, J., et al. (2018). On the design of CRISPR-based single-cell molecular screens. Nat Methods *15*, 271-274.

37. Lin, Q., Zong, Y., Xue, C., Wang, S., Jin, S., Zhu, Z., et al. (2020). Prime genome editing in rice and wheat. Nat Biotechnol *38*, 582-585.
38. Nielsen, S., Yuzenkova, Y., and Zenkin, N. (2013). Mechanism of eukaryotic RNA polymerase III transcription termination. Science *340*, 1577-1580.
39. Doench, J.G., Hartenian, E., Graham, D.B., Tothova, Z., Hegde, M., Smith, I., et al. (2014). Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. Nat Biotechnol *32*, 1262-1267.
40. Doench, J.G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E.W., Donovan, K.F., et al. (2016). Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat Biotechnol *34*, 184-191.
41. Lin, Y., Cradick, T.J., Brown, M.T., Deshmukh, H., Ranjan, P., Sarode, N., et al. (2014). CRISPR/Cas9 systems have off-target activity with insertions or deletions between target DNA and guide RNA sequences. Nucleic acids research *42*, 7473-7485.
42. Chen, H., Choi, J., and Bailey, S. (2014). Cut site selection by the two nuclease domains of the Cas9 RNA-guided endonuclease. J Biol Chem *289*, 13284-13294.
43. Zeng, Y., Cui, Y., Zhang, Y., Zhang, Y., Liang, M., Chen, H., et al. (2018). The initiation, propagation and dynamics of CRISPR-SpyCas9 R-loop complex. Nucleic Acids Res *46*, 350-361.
44. Miller, S.M., Wang, T., Randolph, P.B., Arbab, M., Shen, M.W., Huang, T.P., et al. (2020). Continuous evolution of SpCas9 variants compatible with non-G PAMs. Nat Biotechnol *38*, 471-481.
45. Erwood, S., Bily, T.M.I., Lequyer, J., Yan, J., Gulati, N., Brewer, R.A., et al. (2022). Saturation variant interpretation using CRISPR prime editing. Nat Biotechnol *40*, 885-895.
46. Banskota, S., Raguram, A., Suh, S., Du, S.W., Davis, J.R., Choi, E.H., et al. (2022). Engineered virus-like particles for efficient in vivo delivery of therapeutic proteins. Cell *185*, 250-265 e216.
47. Mangeot, P.E., Risson, V., Fusil, F., Marnef, A., Laurent, E., Blin, J., et al. (2019). Genome editing in primary cells and in vivo using viral-derived Nanoblades loaded with Cas9-sgRNA ribonucleoproteins. Nat Commun *10*, 45.
48. Petri, K., Zhang, W., Ma, J., Schmidts, A., Lee, H., Horng, J.E., et al. (2022). CRISPR prime editing with ribonucleoprotein complexes in zebrafish and primary human cells. Nat Biotechnol *40*, 189-193.
49. Anzalone, A.V., Gao, X.D., Podracky, C.J., Nelson, A.T., Koblan, L.W., Raguram, A., et al. (2022). Programmable deletion, replacement, integration and inversion of large DNA sequences with twin prime editing. Nat Biotechnol *40*, 731-740.
50. Choi, J., Chen, W., Suiter, C.C., Lee, C., Chardon, F.M., Yang, W., et al. (2022). Precise genomic deletions using paired prime editing. Nat Biotechnol *40*, 218-226.
51. Zhuang, Y., Liu, J., Wu, H., Zhu, Q., Yan, Y., Meng, H., et al. (2022). Increasing the efficiency and precision of prime editing with guide RNA pairs. Nat Chem Biol *18*, 29-37.

Abstract in Korean

프라임 에디터 효율의 평가와 예측

프라임 에디터 (Prime editor)는 원하는 위치에 모든 형태의 유전체 교정을 일으킬 수 있기에 앞으로 그 활용도가 매우 높을 것으로 예상되는 기술이다. 하지만 특정 형태의 프라임 에디팅을 유도하기 위한 프라임 에디팅 가이드 RNA (pegRNA)의 설계가 매우 복잡하고, 그 종류가 매우 많기 때문에 높은 효율의 프라임 에디터를 선정하기 어렵다는 한계가 있어왔다. 본 연구에서 우리는 대량의 pegRNA 의 효율을 고효율 스크리닝 기법을 이용해 측정하고 프라임 에디팅의 효율에 영향을 미치는 요소들을 밝혀냈다. 가장 기본적인 형태인 PE2 뿐만 아니라, PEmax, PE4, epegRNA 등 다양한 형태의 프라임 에디팅에 대해서도 동일선상에서 비교하였으며, 또한 위 실험에서 얻은 데이터를 활용하여 프라임 에디팅 효율을 예측하는 딥러닝 모델을 개발하였다. 본 연구에서 개발한 예측 모델은 세포 내 유전체에 직접 프라임 에디팅을 유도하는 pegRNA 들의 효율을 높은 성능으로 예측하였으며, 이는 유전질환을 유도하는 돌연변이에 대한 유전자 교정 치료제를 개발할 때 효과적으로 활용될 수 있음을 보여주었다. 본 연구는 앞으로 프라임 에디터가 활용될 수 있는 다양한 분야에서 최적의 프라임 에디터를 선정하는 것에 사용되고, 유전자 교정을 통한 연구 및 치료제 개발에 중요한 역할을 할 것으로 기대된다.

핵심되는 말 : 프라임 에디터, 고효율 스크리닝, 딥러닝, 유전질환

Publication List

Kim, H.K., Yu, G., Park, J., Min, S., Lee, S., Yoon, S., et al. (2021). Predicting the efficiency of prime editing guide RNAs in human cells. Nat. Biotechnol. 39, 198–206.

Jang, H., Jo, D.H., Cho, C.S., Shin, J.H., Seo, J.H., Yu, G., et al. (2022). Application of prime editing to the correction of mutations and phenotypes in adult mice with liver and eye diseases. Nat. Biomed. Eng. 6, 181–194.

Yu, G., Kim, H.K., Park, J., Kwak, H., Cheong, Y., Kim, D., et al. (2023). Prediction of efficiencies for diverse prime editing systems in multiple cell types. Cell 186, 2256-2272.e2223.

Park, J., Yu, G., Seo, S.Y., Yang, J., and Kim, H.H. (2024). SynDesign: web-based prime editing guide RNA design and evaluation tool for saturation genome editing. Nucleic Acids Res 52, W121-W125.

* This dissertation is based on the works of Kim *et al.* (2021) and Yu *et al.* (2023).