REVIEW ARTICLE

Prediction of genome base-editing efficiency and outcomes based on machine learning: A deep review

Haichang Yao[1, 2, *], Guangyong Hu[1], Peihao Zhang[1]

[1]School of Computer and Software, Nanjing Vocational University of Industry Technology, Nanjing, Jiangsu, China. [2]School of Computer Science and Technology, Tiangong University, Tianjin, China.

Received: May 30, 2024; **accepted:** August 19, 2024.

Base editing, a revolutionary genome editing technology, has risen to prominence for its distinguished features such as high fidelity, precision, and targeted specificity. It has found broad applications across the spectrum of gene therapy, precise breeding, and in-depth gene function studies. The efficiency of base editing and the integrity of resultant genotypic products are the most important performances of base editing technology, which determine whether it can ultimately be suitable for clinical utilization. Because the underlying determinants that influence base editing efficiency and genotypic output remain elusive, the optimization of base editing presently is predominantly dependent on empirical knowledge and iterative experimental attempts. Machine-learning-based prediction for editing efficiency and genotypic outputs can guide base editing applications and optimize base editors in silico, helping researchers improving experimental efficiency and saving experimental costs, which positions it as a significant research direction within this field. This research systematically reviewed the development trajectory of prediction methodologies from CRISPR/Cas9 to base editing, highlighted the intrinsic differences between predictions for base editing and those for CRISPR/Cas9, and then provided a detailed review of all outstanding base editing prediction methods for the first time. The key issues and future directions were also provided for upcoming researchers.

Keywords: genome; base editing; machine learning; efficiency prediction; outcome prediction.

[*]**Corresponding author:** Haichang Yao, School of Computer and Software, Nanjing Vocational University of Industry Technology, Nanjing 210023, Jiangsu, China. Emails: yaohc@niit.edu.cn.

Introduction

Genome editing has become an important technology with applications spanning gene therapy [1, 2], veterinary and agricultural biotechnology [3], foundational genomics research [4], *etc*., which facilitates precise genetic manipulations including the insertion, deletion, and alteration of DNA sequences both *in vivo* and *in vitro* [5]. Presently available genome editing tools such as zinc finger nuclease (ZFN) [6], transcription activator-like effector nuclease (TALEN) [7], and the transformative clustered regularly interspaced short palindromic repeats/Cas (CRISPR/Cas) system [8] elicit targeted double-strand breaks (DSBs) in the DNA, which activate intracellular non-homologous end joining (NHEJ) or homologous directed repair (HDR) pathways [9] to achieve various genetic modifications [10]. NHEJ-mediated cellular repairing processes of DSBs may introduce random nucleotide insertions and deletions (indels), leading to disruptive frameshift mutations [11]. In contrast, HDR-mediated

repairing requires a donor template, allowing for genome editing using artificially designed foreign DNA templates [12]. Generally, NHEJ and HDR coexist and compete with NHEJ being more efficient than HDR. Consequently, most editing outcomes typically involve both insertion and deletion outside the target locus [13], posing significant latent risks. Numerous studies have observed substantial off-target effects associated with these genome editing methods, some of which even carry a carcinogenic risk [14].

Base editing is a genome editing technology that has been evolving since 2016. Different from traditional genome editing technology, base editing does not rely on DNA double-strand breaks but enables single nucleotide editing in DNA or RNA with high efficiency and precision [15]. The development of base editing has significantly affected the basic and clinical research [16]. About 58% of the known genetic variants that cause disease in humans are point mutations, also referred as single nucleotide variants (SNVs) [17]. In addition, SNVs are the main genetic variants affecting livestock traits such as growth, development, fertility, *etc*. [18]. This is why base editing has been seen as a promising way to overcome the challenges of treating many diseases and biological breeding [19]. Base editing achieves gene correction through a complex of deaminase enzyme, Cas9 variants, and single guide RNA (sgRNA). The sgRNA serves as a targeting mechanism that guides the Cas9 variants to unpair the desired DNA sequence, while the deaminase enzyme catalyzes the conversion of specific nucleotide bases [15, 20]. The base editing systems are classified into two categories based on the deaminase enzyme utilized including the cytosine base editor (CBE) and the adenine base editor (ABE), which complete the conversion of cytosine (C) to thymine (T) and adenine (A) to guanine (G), respectively (Figure 1). Base editing systems conjugate either a catalytically inactive Cas9 (dCas9), which has undergone mutations that render it devoid of nuclease activity, or a nickase Cas9 (nCas9), which can only cleave single DNA backbone with a deaminase enzyme that mediates the conversion of the target nucleotide [21, 22]. The sgRNA is used to anchor the Cas9 variants-deaminase complex to the genomic target. Upon targeting, an 'R-loop' is formed through base pairing between the sgRNA and the DNA, resulting in a localized single-strand DNA (ssDNA). Within a certain range, known as the 'activity window', the deaminase enzyme converts C to uracil (U) or A to inosine (I). The modified nucleotides are then repaired or replicated to achieve precise base substitutions [23]. They differ in that CBE converts C to U, which pairs as T during DNA replication, resulting in C to T and G to A conversion [24]. ABE converts A to I, which is treated as G during replication, resulting in A to G and T to C conversion [25]. Following the inception of ABE and CBE, many researchers have witnessed significant advancements in the optimization of these tools. A multitude of base editors has been introduced, of which ABE has developed to the 7th generation [26]. Additionally, CGBE that can convert C to G has also been developed [27].

Although base editing is more efficient and has fewer by-products than traditional genome editing approaches, the off-target problem still exists. The off-target of base editing means that the base editor converts the non-targeted nucleotides. Once the off-target occurs, it may cause genetic disorders, disrupt normal cellular functions, and even induce cancers [28]. Therefore, precision is the prerequisite and guarantee for the successful therapeutic application of base editing [29]. Motivated by this need, the prediction of base editing efficiency and outcomes has become important research to guide the therapeutic application of base editing and the development of new base editors [30]. A lot of prediction methods and tools have been proposed for CRISPR/Cas9 systems. Notably, experimental biotechnology-driven tools such as GUIDE-Seq [31], Digenome-Seq [32], SITE-Seq [33], CIRCLE-Seq [34], HTGT [35], and BLISS [36], *etc*. have been instrumental in identifying off-target events across the genome. These methods offer unbiased, genome-wide detection capabilities. Conversely, computational-based

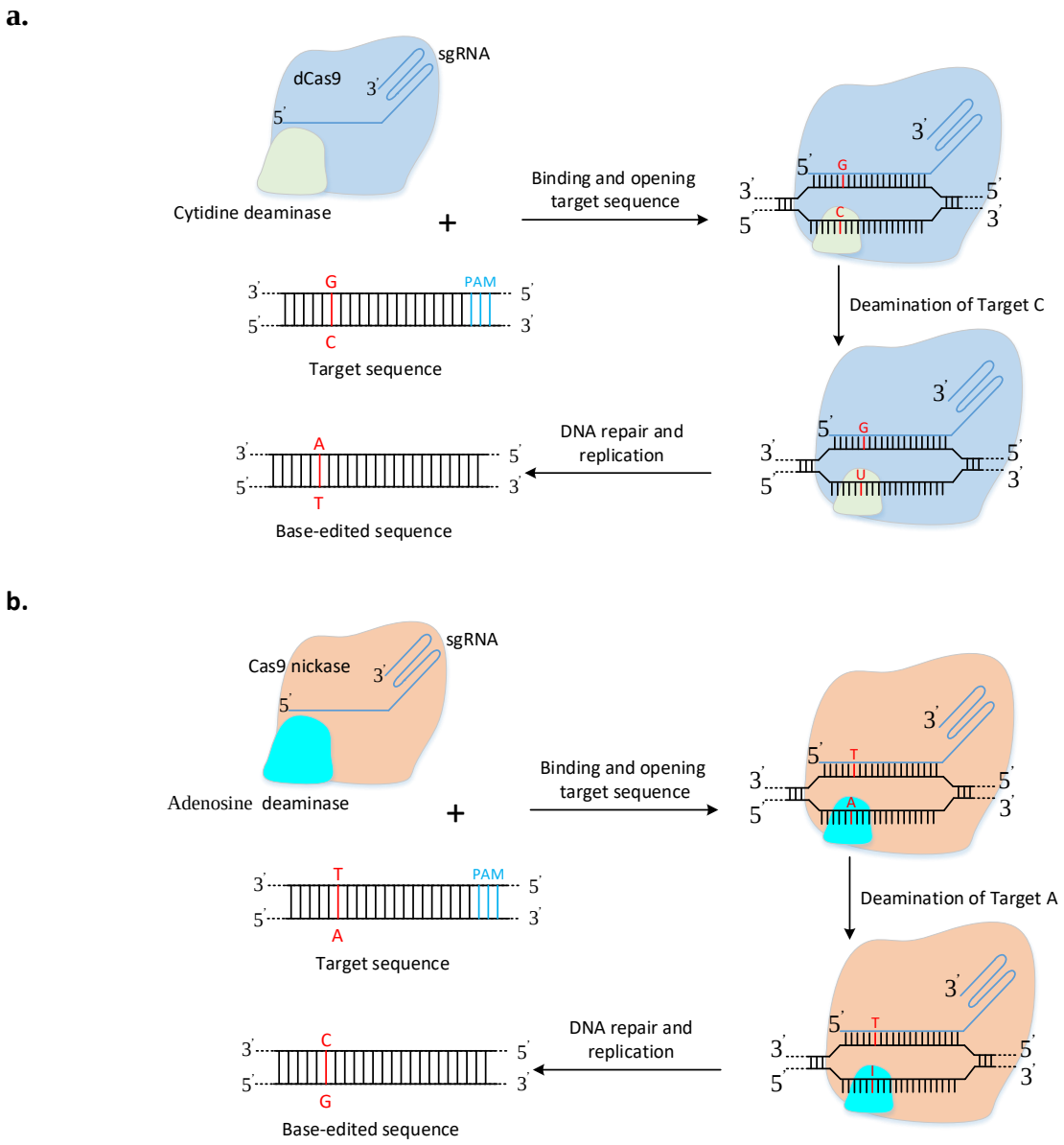


Figure 1. Schematic representation of CBE (**a**) [15] and ABE (**b**) [20].

methods offer a swift, economical, and straightforward alternative by quantifying off-target effects without the requirement for experimental design [37]. Pioneering computational-based off-target detection methods like CCTOP [38], MIT-score [39], and CROP-IT [40] are proposed based on empirical knowledge to formulate detection rules. With the evolution of machine learning, many researchers have applied machine learning techniques such as naive Bayesian networks, enhanced regression trees, support vector machines (SVM), and convolutional neural networks (CNN) to off-target effect prediction. The first machine-learning-based off-target detection algorithm is the cutting frequency determination (CFD) score [41], which already outperformed empirical rule-based off-target detection methods [42]. As the corpus of genome editing knowledge expands and validation datasets become more comprehensive, a succession of machine-learning-based off-target

detection methods has emerged, each surpassing the CFD score in Receiver Operating Characteristic (ROC) curve performance. Listgarten *et al*. proposed Elevation, an approach that included two machine learning models, of which one for assessing individual guide-target interactions, another amalgamated these scores into a collective guide score [43]. Alkan *et al*. proposed CRISPRoff, a model predicated on the approximate binding energy of the Cas9-gRNA-DNA complex [44]. ROC analysis showed that CRISPRoff outperformed all the tools mentioned above. The ability of deep learning in predicting off-target effects has been underscored by several studies. Chuai *et al*. proposed DeepCRISPR, a hybrid neural network model that employed unsupervised deep learning for sgRNA feature representation, followed by a CNN classifier for off-target prediction [45]. Another deep learning model CNN_std leveraged multiple convolutional kernels to learn sequence characteristics of sgRNAs and target sequences, combined with a batch normalization layer to avoid over-fitting and a drop-out layer to enhance generalization ability, and consequently further improved the prediction accuracy [46]. In 2020, the authors of CNN_std developed another prediction model CRISPR-Net [37], a recurrent convolutional network for scoring the off-target activities with mismatches and indels, addressing a notable gap in the field and enhancing predictive performance over CNN_std.

While the aforementioned prediction approaches for CRISPR/Cas9 systems are constantly being optimized, it is important to recognize that the off-target factors in base editing systems differ. The dynamic interplay between the base editor and its target sequence affects the editing efficiency and outcomes in a more complex and sometimes unintuitive way [47]. Scenarios where the activity window encompasses multiple target nucleotides may result in a broader spectrum of potential genotypic outcomes. To date, the factors impacting the base editing off-target effects have not been explored deeply enough, and there is a lack of consensus in the field [48]. Generally speaking, base editing systems exhibit two primary forms of off-target activities including genome-wide off-target activities mainly caused by mismatches of sgRNAs [49] and off-target activities within the editing window due to incorrect editing of bystander nucleotides. The former is also observed in CRISPR/Cas9 systems, allowing for the adaptation of existing off-target detection methodologies, while the latter cannot be predicted by current CRISPR-based methods. Currently, the methods of improving the purity of base editing outcomes are usually done by empirically optimizing the base editors as well as sgRNAs [50]. Certain feasible off-target activities that do not match empirical guidelines are easily overlooked as simple empirical guidelines cannot fully cover all possibilities of base editing systems. There is an imperative need to design specialized approaches that are tailored to the intrinsic workings of base editing systems, which can be very helpful not only in instructing the applications of base editing systems but also in developing new base editors. With the rapid development of artificial intelligence (AI) in recent years, prediction technologies have been significantly developed. Several distinguished approaches specialized for base editing prediction have been proposed. However, up to now, there has been no literature that systematically reviews the evolution of prediction technologies for base editing. This research first systematically reviewed the development trajectory of genome editing outcome prediction technologies for CRISPR/Cas9 and highlighted the intrinsic differences between predictions for base editing and those for CRISPR/Cas9. The study provided a detailed review of all outstanding base editing prediction methods from the beginning to the most recent including the architecture, training data construction, and training methods along with the analysis of their performance. The challenges and directions for the future development of base editing prediction technologies were also identified. This study will be helpful in attracting more AI experts to apply the latest AI technologies to assist in the development of base editing prediction methods.

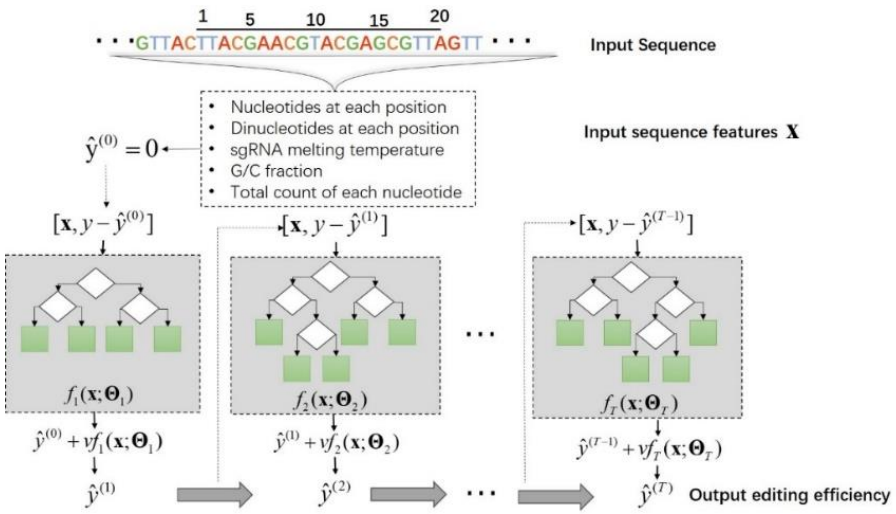
Methods for base editing prediction

The efficiency and outcomes distribution are important performance indicators for base editing systems. The total efficiency of base editing systems is defined as the number of all short reads that yield an editing effect divided by the number of valid short reads after high throughput sequencing. Specifically, the base editing efficiency at position *i* is calculated as total reads that contain nucleotide transition at position *i* divided by total valid reads. The outcomes distribution is the probability distribution of each editing outcome across all editing outcomes. Efficiency measures the effectiveness of the base editing systems, and the outcomes distribution measures the off-target risk of the base editing systems. The efficiency and off-target rates varying greatly in different base editing systems and target sequences are major constraints for base editing applications [51]. The optimization of base editing systems is aimed at improving efficiency and reducing off-target rates. However, they are more of a mutually restrictive relationship. For example, enhancing the concentration and expression time of deaminase and Cas9 complex in cells may improve the efficiency to a certain degree, but at the same time, that will also lead to an increase of the off-target rate. Modifying deaminase can narrow the activity window of the deaminase, thus reduce the off-target rate, but that may reduce the efficiency. Therefore, accurate prediction of editing efficiency and outcomes under specific conditions is essential for the strategic selection of sgRNAs, base editors, and Cas9 variants, thereby maximizing the desired efficiency and outcomes. Since the inception of base editing in 2016, along with the development of machine learning technology, several distinguished methods have been proposed by researchers, which leverage machine learning technology to predict the efficiency and outcomes distribution of base editing systems.

(1) BE-Hive

Komor *et al*. published the first base editing system and researched the off-target effects of base editing earlier [15]. In 2020, Arbab *et al*. proposed BE-Hive, the first machine-learning-based model for the prediction of base editing outcomes [47]. BE-Hive incorporated two models including one for the prediction of editing efficiency and another for bystander editing. The editing efficiency model, a logistic regression (LR) model, was first designed to extract the correlation between editing efficiency and sequence motif. The prediction results indicated that the sequence motif was an important factor affecting the editing efficiency for all base editors with a weight of 15% to 32%. To further exploit higher-order interaction and more features, a gradient-boosted regression tree (GBRT) model was subsequently utilized. The architecture of GBRT is shown in Figure 2a. The GBRT model expanded upon the LR model by incorporating more base editing factors including the single nucleotide and dinucleotide motifs at each position, sgRNA melting temperature, the G/C fraction, the total count of each nucleotide, the activity window, and so on. The GBRT model demonstrated enhanced predictive accuracy over the traditional LR model. To predict the bystander editing outcomes, BE-Hive designed and implemented a deep conditional autoregressive (DCAR) model (Figure 2b). The model was implemented by a pair of neural networks including an encoder with two hidden layers and a decoder with five hidden layers. Each hidden layer comprised 64 neurons. The networks were fully connected with Rectified Linear Unit (ReLU) as activations along with residual connections that linked neighboring layers. DCAR processed each substrate nucleotide and its immediate sequence context through the shared encoder to create a deep representation, which was then utilized by the autoregressive decoder. The decoder systematically produced a distribution of potential base editing outcomes for each nucleotide loci, taking into account the context of previously determined outcomes. The model considered four factors including 50 base pair (bp) target sequence and protospacer adjacent motif (PAM), sgRNA, base editor, and cell type. The findings of BE-Hive inspired the

a.



b.

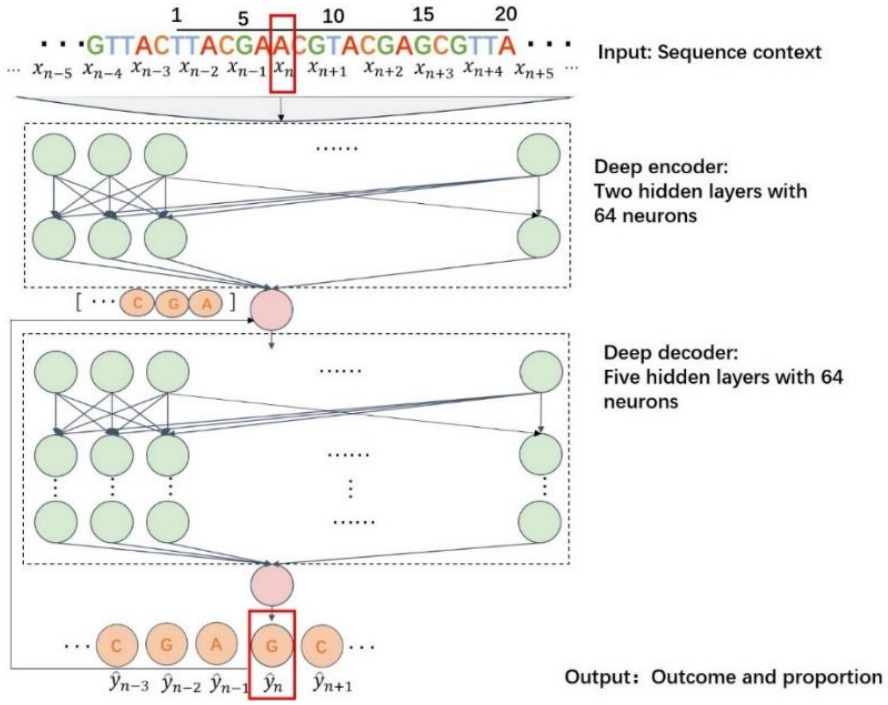


Figure 2. The architecture of GBRT (**a**) and DCAR (**b**) [47].

improvements of base editors and achieved previously unattainable editing goals. The results demonstrated the method's performance by accurately correcting 3,388 disease-associated SNVs with a success rate of 90% or higher. Furthermore, BE-Hive also identified previously unpredictable determinants of C to G or C to A transitions, facilitating the precise correction of 174 pathogenic SNVs with comparable accuracy. In the following year, a CGBE system was developed and retrained the BE-Hive model using the CGBE datasets, expanded its capabilities [52]. Regarding the datasets, sequence-activity correlations of 11 base editors across 38,538 integrated sequences was constructed, and these data sets was used to train BE-Hive. These

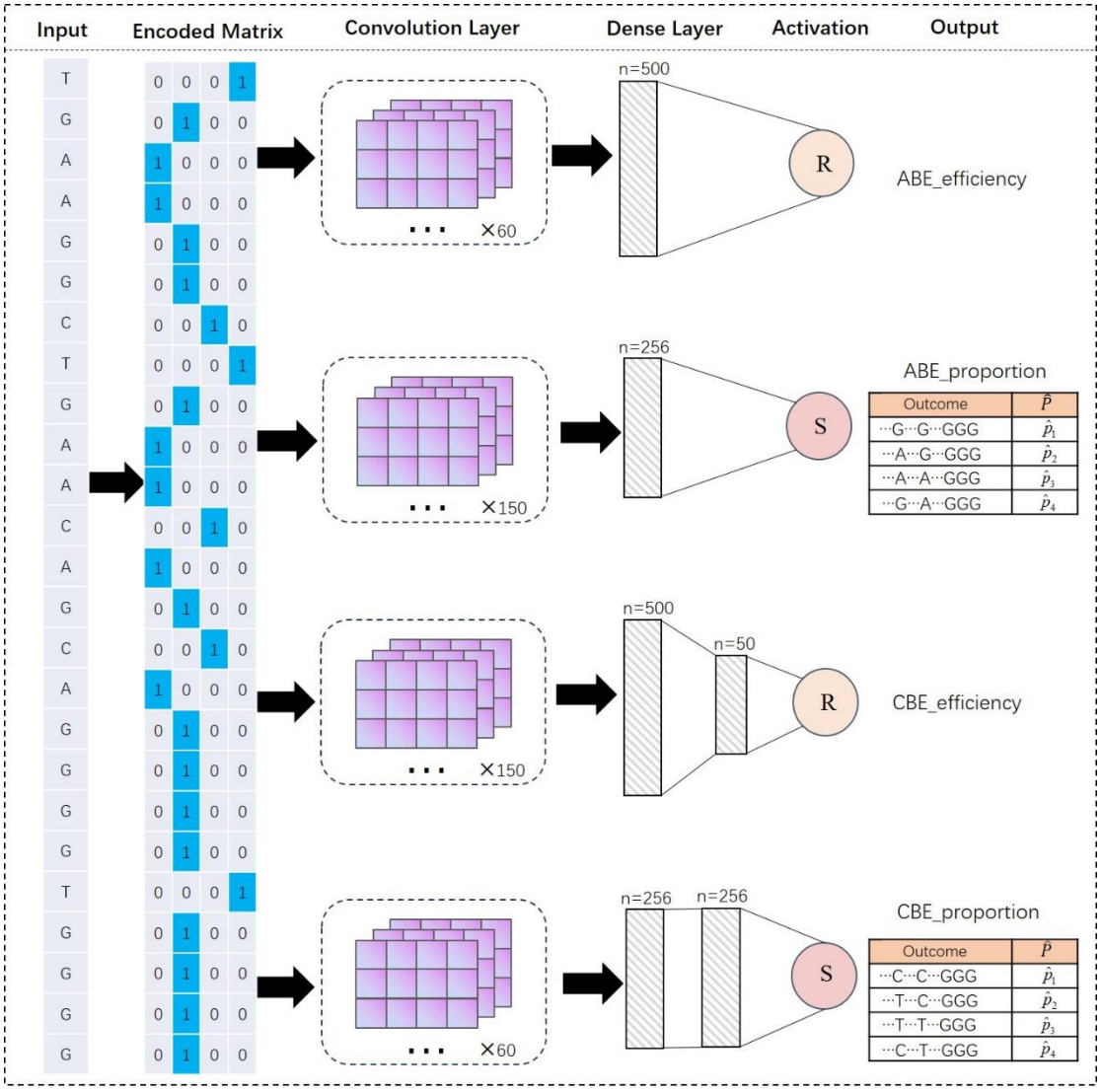


Figure 3. The architecture of DeepBase [53].

sequences covered all base arrangement combinations within the editing window, and then were integrated into three types of cell lines including mouse embryonic stem cells (mESCs), human embryonic kidney cells (HEK293T), and human osteosarcoma cells (U2OS) for base editing.

(2) DeepBase

DeepBase is a CNN-based prediction model developed for the base editing system by Song *et al*. in 2020 [53]. According to the types of base editors ABE and CBE and the purpose of prediction for efficiency and proportion predictions, DeepBase designed and implemented four specialized models including ABE_efficiency, CBE_efficiency, ABE_proportion, and CBE_proportion. Each model was designed with a similar structure yet distinct in some parameters. A schematic representation of their architectures and important parameters was shown in Figure 3. The input of each model was uniform, one-hot encoded matrix of the target sequence with PAM. The convolution layer of each model shared the parameters of kernel size 3, channel dimension 4, stride 1, and no padding. The variation across models laid in the number of filters employed. Notably, the exclusion of

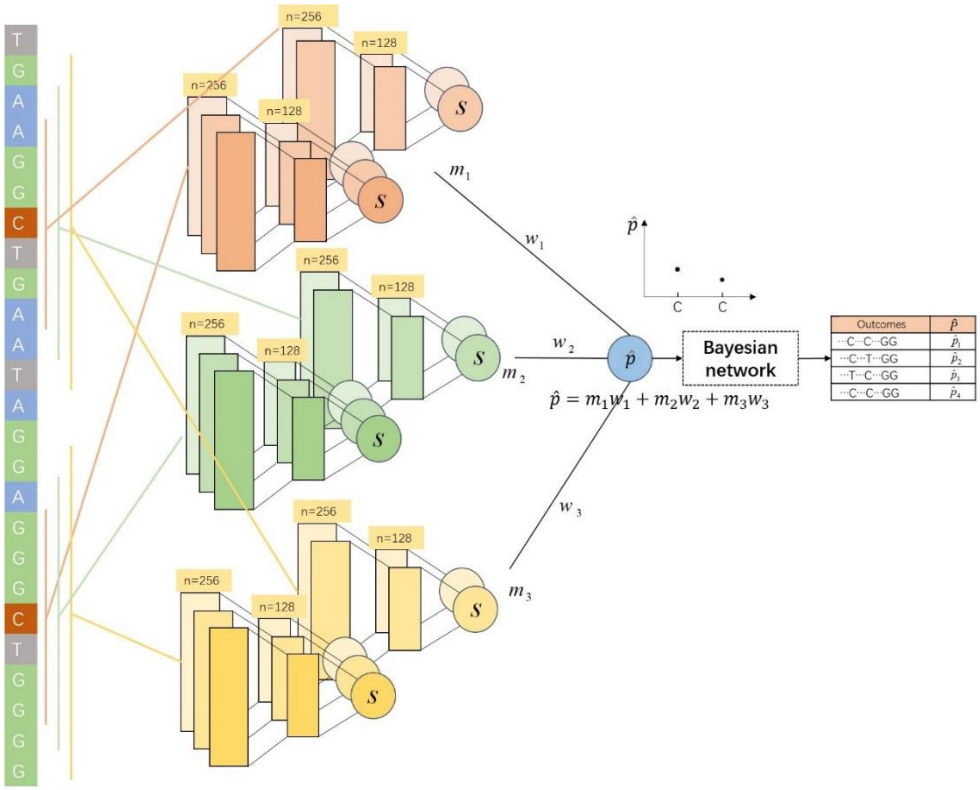


Figure 4. A schematic representation of CGBE-SMART [55].

pooling layers was a deliberate design choice, stemming from the observation that models performed more effectively without them. The count of dense layers and the associated nodes also varied between models. ReLU activation functions were utilized throughout all layers with the exception of the last layer of the proportion models, which employed a Softmax activation function. During the training process, the model adopted two strategies to prevent overfitting, which were early stop strategy based on the predictive performance of the validation datasets and a 0.3 dropout rate across all layers. Especially, when the datasets were not large enough, the use of the dropout strategy was a common method to prevent overfitting by avoiding the co-adaptation problem of nodes. In addition, seven traditional machine learning models were constructed for performance comparison. The comparative analysis demonstrated that, with an adequate volume of data, the CNN-based model consistently outperformed these traditional models, even though DeepBase contained only 2-3 hidden layers. This finding aligned with previous observations in CRISPR genome editing applications. Concerning the datasets, the efficiencies and outcomes at 13,504 and 14,157 integrated sequences edited by ABEs and CBEs were first generated followed by 95 and 102 endogenous target sites edited by ABEs and CBEs, respectively, in HEK293T cells to evaluate the prediction accuracy of DeepBase at endogenous sites.

(3) CGBE-SMART

The traditional CBE is capable of editing C to T, but editing C to G or A to T is capable of addressing approximately 11% of the 32,044 pathogenic point mutations [54]. In 2021, researchers developed cytosine-to-guanine base editor (CGBE) and proposed the predictive methods for CGBE [52, 55]. The ability of CGBE was achieved by removing uracil DNA glycosylase

inhibitor (UGI) or increasing uracil DNA N-glycosylase (UNG). Compared to CBEs, CGBEs exhibited a reduced specificity, which meant that the outcomes were more diverse. The purity of CGBE could only reach 30-50% [56]. In some CGBEs, the amount of C to T by-products might even exceed the target C to G target products, possibly due to CGBE's increased reliance on the cellular DNA repair mechanisms to convert AP sites to guanines [57]. Yuan *et al*. developed a total of 8 CGBE variants and trained the CGBE-SMART model based on the editing datasets of these variants [55] (Figure 4). The CGBE-SMART used a deep convolutional neural network inspired by Google inception networks that employed a parallel processing approach using filters of varying sizes to capture features across multiple scales [58]. CGBE-SMART comprised nine foundational models, each with window sizes of 7, 9, and 11 nucleotides, triplicated for every nucleotide position. Each model was assigned a learned weight, which contributed to the final output. The input was the one-hot encoded matrix of the target sequence from protospacer position 1-20. For a base model, the first layer comprised 256 neurons followed by the second layer with 128 neurons, both employing ReLU activation. The final layer, consisting of a single neuron, employed a Sigmoid activation to produce the output predictions. A dropout rate of 0.3 was integrated during training to prevent overfitting, and the mean square error (MSE) was utilized as the loss function. The efficiency prediction model's output provided the editing probability for each individual target nucleotide, which was more precise than the editing probability of the entire target sequence output by BE-HIVE and DeepBase. CGBE-SMART incorporated a Markov network to account for the interdependencies between nucleotide positions. For the sake of simplicity, the model primarily focused on the relationships between adjacent editing sites, simplifying the Markov network to a Bayesian network equivalent. By inputting the editing probability of each position as determined by the efficiency model, the Bayesian network deduced the distribution of all possible editing outcomes. On the datasets, 41,388 integrated target sequences and 100 HEK293T endogenous target sites were generated based on eight CGBEs. CGBE_SMART could also be used for training and prediction of other base editing systems such as ABE and CBE. When compared with BE-Hive and DeepCBE models, CGBE-SMART demonstrated superior predictive accuracy in 7 out of 8 CGBE datasets.

(4) BE-DICT

BE-DICT represented an innovative methodology to predict per-base editing efficiency and outcomes through an attention-based deep learning algorithm [59]. The inspiration from the transformer architecture was drawn [60]. Each nucleotide of BE-DICT models was as a linguistic entity, employing a multi-head self-attention mechanism to process the genetic sequence (Figure 5). The model took a 23 bp target sequence containing PAM as input and translated each nucleotide and its positional context into a dense vector format through one-hot encoding. This vector was then channeled into the encoder block, a core component comprising a self-attention layer, normalization and residual connection layers, and a feed-forward network. The self-attention layer operated through a multi-head mechanism, where a series of single-head self-attention layers worked in tandem to refine the input vector. The outputs from these layers were concatenated and processed to produce a vector of fixed dimensions. Residual connections were strategically integrated to enhance the gradient propagation during the training phase, while layer normalization was applied to address the "covariate-shift" phenomenon by re-standardizing the vector representations. The feed-forward network further refined the vector representation derived from the preceding layers. Post the cascade of N encoders, the edit probability for each base was determined *via* a linear transformation followed by a Softmax activation. The outcomes prediction module adhered to an encoder-decoder framework, mirroring the encoder's complexity in the decoder's structure. The decoder included a mask self-attention layer and an encoding-decoding attention layer with the former

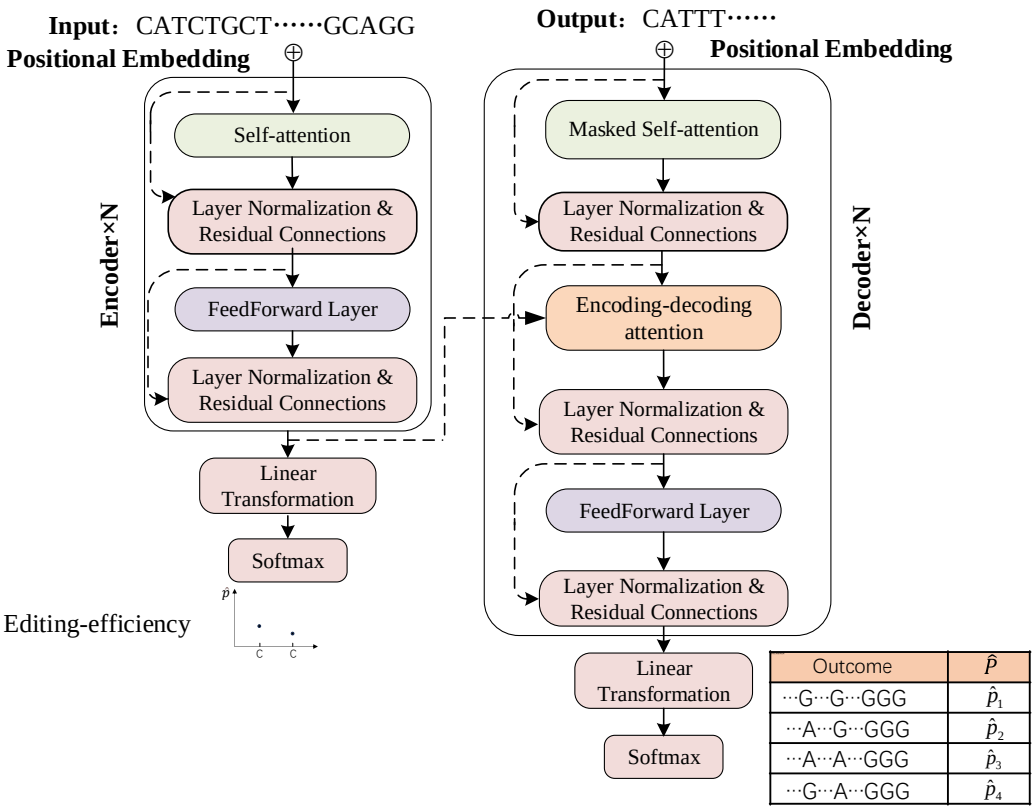


Figure 5. The architecture of BE-DICT [59].

functioning as an "autoregressive layer" that utilized only preceding information to ascertain the output probability. The encoding-decoding attention layer was instrumental in discerning the significance of each base in the input sequence for the output base vector [61], enabling the model to precisely determine the likelihood of each potential outcome. For training, the model was fed with a dataset of 23,123 random target sequences and 5,171 SNVs related to diseases. To mitigate bias towards edited sequences, an additional 25% of unedited target sequences were integrated into the dataset for each base editor. Furthermore, the model used 18 ABE and 16 CBE endogenous loci editing data from HEK293T cells for validation.

(5) FORECasT-BE

Ananth *et al*. constructed a suite of prediction models for each target nucleotide within editing window 3-10 [62]. These models were constructed based on linear regression with L1 and L2 regularization, gradient boosting trees, and neural networks, respectively. For the neural network models, the evaluation spanned a range of hidden layers of 1 ~ 5 and channel counts per layer of 10 ~ 500. The gradient boosting tree models were assessed based on the number of decision trees of 10 ~ 1,000, tree depth of 1 ~ 10, minimum leaf count of 1 ~ 50, and learning rates of 0.001 ~ 1. The final evaluation on the datasets generated revealed that the three types of models had similar performance, and finally, the gradient boosted tree model was chosen because it had better feature extraction capability than that of linear regression and better interpretability than that of neural networks. Therefore, the model architecture of FORECasT-BE was similar to Figure 2a. Notably, the method could only predict the editing efficiency by base but not the probability distribution of editing outcomes. On the datasets, the CBE datasets were generated in K562 cells, and the ABE

datasets were generated in HEK293T cells. The datasets contained target base editing data for each position within the editing window of 3 ~ 10 totaling 14,409 target sequences. In addition, the FORECasT-BE was jointly trained by fusing the datasets of BE-Hive and DeepBase to improve the generalization ability.

(6) DeepBE

The applications of the base editing systems are frequently restricted by specific PAM, base editing types, and sgRNA. Cas9 is susceptible to PAMs. Different Cas9s recognize different PAMs, e.g. the canonical PAM required by SpCas9 is NGG. Because humans have an average of one NGG every eight bases, this restriction can be easily overcome. However, there are still cases where some of the target loci lack the required PAM. In these cases, variants or homologues of Cas9 that can recognize different PAMs such as SpCas9-VQR, SpCas9-VRQR, *etc*. are developed. Until now, more than a dozen different Cas9 variants have been developed. Base editing can be categorized as ABE, CBE, and CGBE, however, each base editing type also contains numerous variants. The selection of sgRNA can affect the positioning of the target nucleotide within the editing window, and different Cas9 and base editing variants affect the efficiency of base editing. How to determine the most efficient combination of editing systems among different sgRNA, Cas9 variants, and base editing types is very difficult for base editors. To address this problem, Kim *et al*. firstly developed the deep-learning-based model DeepCas9variants trained on the edited datasets of nine Cas9 variants for the purpose of predicting which Cas9 variants induced the most efficient editing to the target nucleotide [63]. Taking the prediction scores of DeepCas9variants as input, another deep-learning-based model DeepBE was developed for 63 base editors which were combinations of nine Cas9 variants and seven base editing variants. The architecture of DeepCas9variants and DeepBE was shown in Figure 6. The input of DeepCas9variants was the 30 bp target sequence and specific Cas9 variant. After one-hot encoding, the base editing efficiency guided by this Cas9 variant was obtained after a series of layers. The architecture within the dense layers was tailored for each Cas9 variant with variations in both the number of layers and their associated hyperparameters. For the SpCas9, SpRy, and Sc++ variants, a three-layer dense structure was implemented, comprising 1,500 nodes for SpCas9 and 1,000 for SpRy and Sc++ in the first two layers and 100 nodes in the third layer. In contrast, other Cas9 variants utilized a two-layer setup with 1,500 and 100 nodes, respectively. The convolution kernels were all 4 × 10 in size with either 1,000 or 2,000 channels. The DeepBE model was also divided into DeepBE_efficiency and DeepBE_proportion. Different from the existing efficiency prediction model, DeepBE_efficiency used the editing window ±1 bp as input information. In addition, it leveraged the DeepCas9variant scores as an indicator of Cas9's activity level. The input to DeepBE_proportion was a 20 bp target sequence deliberately excluding the PAM to eliminate the influence of PAM compatibility on the prediction. Within the convolutional layer, a variable number of nodes of 256, 512, or 1,024 were engaged based on the specific base editing system, and the extracted features were flattened. The final prediction scores for the base editing systems incorporating Cas9 variants were derived from the multiplication of the output layers from both the efficiency and proportion models [63]. Although DeepBE was developed to predict efficiency and outcomes for 63 base editors, it’s difficult to generate datasets of so many base editors for training. The dataset utilized for DeepBE's training encompassed seven base editors containing the SpCas9-NG variant and an additional seven base editors containing a diverse array of Cas9 nickase variants. It was noteworthy that the authors also deployed five traditional machine-learning-based models for comparison with the results demonstrating that the deep-learning-based models outperformed other conventional models, corroborating the superior predictive capabilities of deep-learning-based models observed in the DeepBase models.

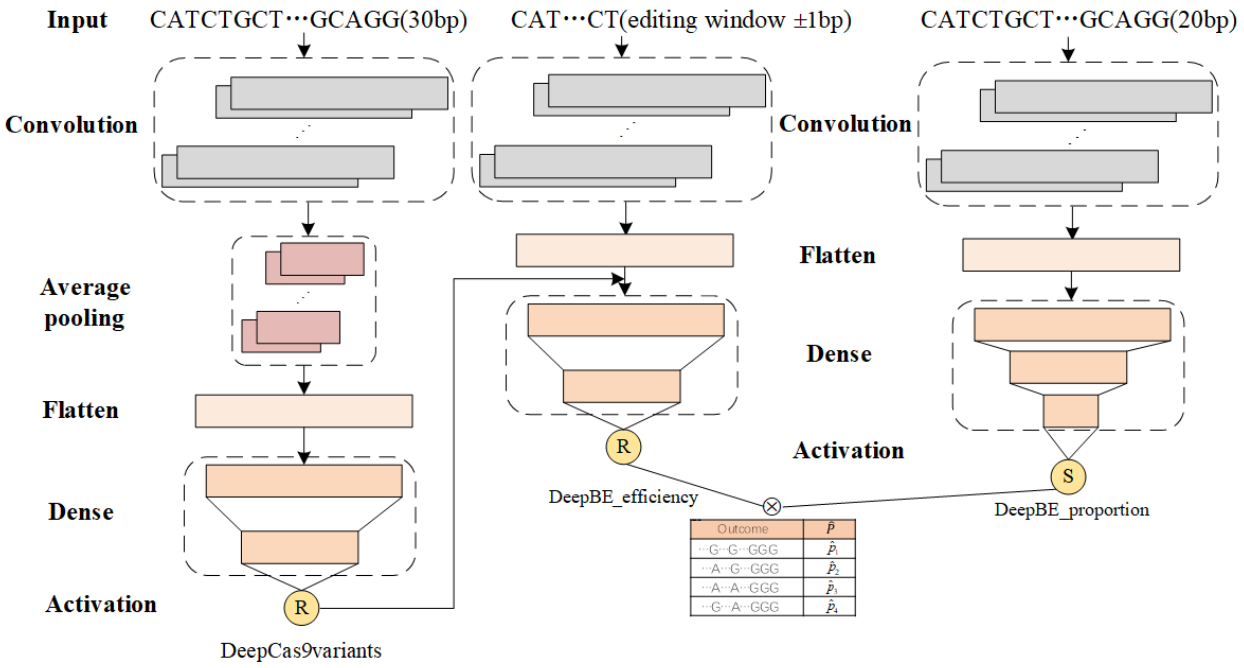


Figure 6. The architecture of DeepCas9variants and DeepBE [63].

(7) BE_Endo

The above studies developed computational models mainly based on integrated datasets, offering advantages in optimizing sgRNA for genome editing and holding potential for future applications. However, a critical distinction exists between the synthetic measurements of editing efficiency and outcomes in integrated datasets and those observed in endogenous genome editing. Although the aforementioned studies also developed some endogenous datasets, these datasets were mainly used for testing, not for training. Because the scale of these datasets was small with only a few hundred at most. Yuan *et al*. constructed a comprehensive endogenous dataset encompassing over 5,000 target sites [64]. Comparative analysis demonstrated significant variations in editing efficiency and outcomes between endogenous and integrated targets. Factors influencing endogenous editing efficiency included, but were not limited to, transcriptional activity, Pol II and CTCF binding sites, chromatin accessibility, and histone modification patterns. By incorporating analyzed endogenous factors and sequence features, the authors proposed a deep learning model similar to Google Inception named BE_Endo with the architecture depicted in Figure 7. The efficiency model of BE_Endo was structured with five convolutional layers featuring varying kernel sizes, followed by a concatenation layer, a max pooling layer, and a 20-size vector output layer. The input matrix was a one-hot encoded representation of the 40 bp target sequence, spanning 10 bp upstream, the 20 bp protospacer, the 3 bp PAM sequence, and 7 bp downstream, complemented by additional channels for one or more endogenous factors. The factors incorporated were selected by influence according to the analysis of endogenous datasets. The ABE_Endo model only incorporated the H3K27ac modification factor, and the CBE_Endo incorporated all factors listed in Figure 7. The proportion model succeeded the efficiency model, employing a Bayesian network to ascertain the combined probabilities of all potential editing outcomes for each target site. The results showed that, compared with models only incorporating sequence features, the models incorporating endogenous factors improved the prediction accuracy at endogenous target sites.

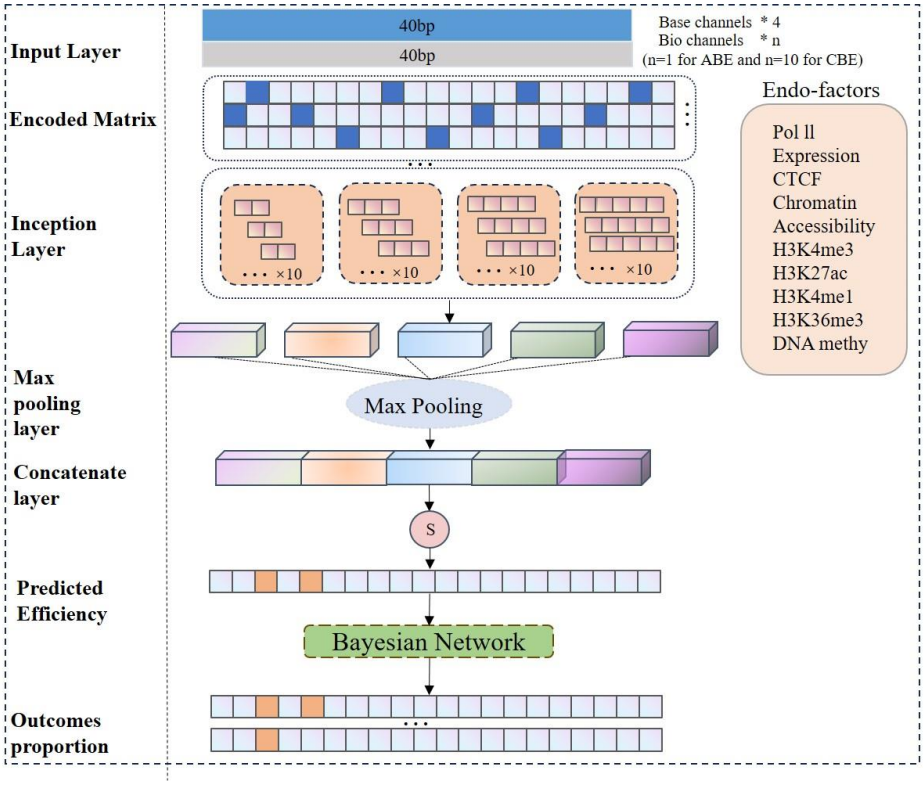


Figure 7. The architecture of BE_Endo [64].

(8) Comparison of proposed methods

A comparative analysis of the key aspects of the aforementioned methods was shown in Table 1. In terms of core technologies, BE-HIVE and FORECasT-BE adopted conventional machine learning models, while others adopted deep learning models, in which DeepBase and DeepBE applied simple CNN, CGBE-SMART, and BE-Endo applied Inception architecture, BE-DICT applied Transformer architecture. Every method implemented efficiency prediction and outcomes prediction, except for FORECasT-BE which only incorporated the efficiency prediction. The types of researched base editors were getting richer from ABE and CBE to CBGE. The number of researched base editors was increasing from several to dozens. The datasets for training models were more comprehensive from considering only base editing types to considering Cas9 variants and endogenous factors. Collectively, the prediction models were constantly evolving with a gradual increase in prediction ability and generalization. The most universal metric for evaluating the performance of base editing efficiency and outcomes prediction is the Pearson correlation coefficient, followed by the Spearman correlation coefficient [65]. Additionally, some research has also incorporated the KL divergence as a metric for gauging the success of predictive models. The Pearson correlation coefficient was calculated as equation (1).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\ \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

where x_i was the *i*-th predicted outcome. $\bar{x}$ signified the average of all predicted results. Correspondingly, y_i and $\bar{y}$ were the measured and the mean values of the sample set, respectively. The total count of samples was denoted by *n*. The Pearson correlation coefficient, *r*, varied from -1 to 1, where a positive *r* indicated a positive association, and a

Table 1. The comparative analysis of proposed methods.

Method	Model-based	Sub-models	Ability	BE Types and counts	*Inte data sets	*Endo data sets
BE-Hive [47]	Machine learning	LR	Efficiency prediction under different Motifs	ABE: 2 CBE: 6	38,538	3,388
		GBRT	Efficiency prediction			
		DCAR	Outcomes prediction			
DeepBase [53]	CNN	ABE_efficiency	ABE efficiency prediction	ABE: 1 CBE: 1	15,656	197
		CBE_efficiency	CBE efficiency prediction			
		ABE_proportion	ABE outcomes prediction			
		CBE_proportion	CBE outcomes prediction			
CGBE-SMART [55]	Inception	CGBE_SMART-Efficiency	Efficiency prediction per-base	CGBE: 8	41,388	100
		CGBE_SMART-Proportion	Outcomes prediction			
BE-DICT [59]	Transformer	per-base model	Efficiency prediction per-base	ABE: 2 CBE: 2	28,294	34
		bystander model	Outcomes prediction			
FORECasT-BE [62]	Machine learning	Gradient Boosting Trees	Efficiency prediction	ABE: 2 CBE: 2	14,409	/
DeepBE [63]	CNN	DeepCas9variants	Efficiency prediction of Cas9 variants	Cas9 variants: 9	25,628	/
		DeepNG-BE_efficiency	Efficiency prediction of BE Guided by SpCas9-NG	SpCas9-NG-BE：7	17,280	/
		DeepNG-BE_proportion	Outcomes prediction of BE Guided by SpCas9-NG			
		DeepBE_efficiency	Efficiency prediction of 63 BEs	Cas9-BE：7	11,015	515
		DeepBE_proportion	Outcomes prediction of 63 BEs			
BE_Endo [64]	Inception	BE_Endo_efficiency	Efficiency prediction	ABE: 1 CBE: 1	11,868	5,012
		BE_Endo_proportion	Outcomes prediction			

* 'Inte' meant Integrated, and 'Endo' meant Endogenous. '/' indicated that no endogenous datasets were generated.

negative *r* suggested an inverse relationship. An *r* value approaching 1 signified a robust positive correlation, which was indicative of the model's enhanced predictive capabilities. Spearman

correlation coefficient was a non-parametric statistical correlation and was calculated as equation (2).

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \quad (2)$$

where $d_i = x_i - y_i$ indicated the difference in ranks between the measured and the predicted values. The Spearman correlation coefficient operated on the principle of determining the Pearson correlation coefficient for the ranks of the two vectors, thereby assessing the extent to which a monotonic relationship could characterize the two vectors. When a monotonic correlation existed between the variables, the Spearman coefficient reached its extremities of +1 or -1. Similar to the Pearson correlation coefficient, an increase in the Spearman value towards 1 denoted a more pronounced monotonic relationship, which correlated with superior predictive efficacy of the model. The KL Divergence served as a measure of the resemblance between two probability distributions [66], making it a valuable tool for assessing the accuracy of outcome predictions. BE-Hive integrated the KL divergence into its bystander editing framework as a loss function, while DeepBase had employed the symmetric KL divergence to quantify the efficacy of its predictive modeling. The calculation of the symmetric KL Divergence was shown in equation (3).

$$KL = \sum_{i=1}^{n} (P_i \log \frac{P_i}{Q_i} + Q_i \log \frac{Q_i}{P_i}) \quad (3)$$

where P_i and Q_i represented the predicted and observed proportion for the *i*-th sample, respectively. A lower KL value suggested superior predictive capabilities. In evaluating the predictive efficacy of proposed methodologies, the performance of LR-based models was inferior to the machine-learning-based models. Both BE-HIVE and BE-Endo implemented the LR-based models for the purpose of predicting the preference of sequence context adjacent to target sites. The comparison of BE-HIVE showed that LR achieved a Pearson *r* ranging from 0.50 to 0.57, whereas DCAR improved this to 0.69 ~ 0.80 for ABEs and 0.53 ~ 0.74 for CBEs. BE_Endo also improved the accuracy from 0.41 ~ 0.57 for ABEs and 0.17 ~ 0.64 for CBEs using LR-based models, to 0.64 ~ 0.78 for ABEs and 0.57 ~ 0.82 for CBEs with BE_Endo efficiency model. This was due to machine-learning-based models considering higher-order interactions and additional sequence features compared to logistic regression models. Broadly speaking, deep-learning-based models tended to surpass conventional machine-learning or shallow-learning models in performance when ample training data was available [53]. Otherwise, they had similar performance [62]. Amongst the deep-learning-based models, Inception-based models were able to extract multiple sequence features through various kernels with different sizes, Transformer-based models exceled by assigning attention scores to each position within the protospacer, reflecting the position's impact on editing outcomes, thus outperforming straightforward CNN-based models. Besides, training datasets are also an important factor for the performance of models. For deep-learning-based models, considering more factors and providing adequate datasets will result in better predictive performance. For example, the model trained by enough endogenous datasets performed better at endogenous target sites prediction than other models only trained by integrated datasets [64].

Challenges and future directions

Although various models for predicting the base editing efficiency and outcomes have been proposed, compared to the achievements of CRISPR/Cas9 systems, prediction research on base editing systems is still in its early stage and there is still a lot of research space. At present, all the proposed machine-learning-based models almost directly use the existing models in the

Python package, and deep-learning-based models also draw on general models in the image or natural language processing field. There is no specific model for base editing systems yet. One of the reasons is that the multitude of influencing factors and their relationships that affect the base editing systems are currently unclear, which can be reflected in the diversity of proposed models. The machine learning models used in the proposed methods include logistic regression, gradient boosting trees, and autoregressive models. The deep learning models used in the proposed methods include CNN architecture, Inception architecture, and Transformer architecture. In terms of input information, almost no model is identical. BE-DICT, CGBE-SMART, and Deep-BE only require the target sequence, but the length of the input sequence is different. BE-DICT and CGBE-SMART require the input length to be 20 bp and 40 bp, respectively. Deep_BE_Efficiency only requires inputting the sequence of editing window length. In addition to the 20 bp target sequence, FORECasT-BE also requires inputting the information of sgRNA melting temperature. The input of the BE-HIVE also includes the G/C content, the total number of each nucleotide, and the dinucleotide motif. Different models require different inputs, which also indicates that the affecting factors of base editing efficiency and outcomes are not yet clear. In the future, with the development of base editing technology, the factors will gradually become clearer, and it is possible and necessary to design specific machine learning models for base editing systems. The datasets are also pivotal in the optimization of prediction models. Currently, almost all proposed methods are trained and validated on their own generated datasets, but some methods differed greatly in their performance on other datasets. For example, DeepCBE had Pearson r = 0.76 on its own HEK293T datasets. However, on the mESC datasets constructed by Arbab *et al.*, the Pearson r decreased to 0.38 [47]. The final model selected by FORECast-BE was simple, but it still achieved an excellent performance, mainly because it integrated the BE-Hive and DeepBase datasets during training. At present, the base editing datasets are limited in size and dispersed, necessitating the creation of comprehensive benchmark datasets. The benchmark datasets should encompass genome-wide base editing across various base editors, cell types, sequence motifs, and target nucleotide counts within diverse editing windows. Besides, having standardized datasets could attract more AI experts into the field and further promote the development of base editing technology.

Conclusion

Since the emergence of base editing in 2016, it has evolved from ABE and CBE to CGBE, and has now developed dozens of variants with rich applications. Although base editors have been greatly optimized in various aspects during upgrading, there remains substantial scope for refining their editing efficiency and reducing off-target effects, which is currently the main factor restricting their therapeutic application. The prediction research of base editing can be used to guide the selection of base editors, sgRNAs, and Cas9 variants, recognize the preferred motif of specified base editors, and predict the affecting factors and their weights of base editing systems. The findings can contribute to the optimization and improvement of base editors. However, due to the fact that base editing has not been invented for a long time, the prior knowledge of base editing is not rich enough, and the size of the datasets is small, which makes the prediction of the editing efficiency and off-target rate is still in its infancy with a few methods having been proposed. Both in terms of the AI models and the standardization of datasets, there is still a lot of work that needs to be done. Designing specific models based on the characteristics of base editing systems and standardizing comprehensive datasets are important research directions to improve predictive performance in the future.

Acknowledgements

This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 22KJB520001), in part by Youth Foundation for Humanities and Social Sciences Research of the Ministry of Education: Research on the Personalized and Intellectual Adapted Teaching Model and Technology for Vocational Undergraduate Education (Grant No. 23YJC880132), in part by Modern Educational Technology Research Program of Jiangsu Province in 2022 (Grant No. 2022-R-98629), in part by Scientific Research Start-up Foundation of Nanjing Vocational University of Industry Technology (Grant No. YK21-05-04).

References

1. Adli M. 2018. The CRISPR tool kit for genome editing and beyond. Nat Commun. 9:1911.
2. Alhakamy NA, Curiel DT, Berkland CJ. 2021. The era of gene therapy: From preclinical development to clinical application. Drug Discovery Today. 26(7):1602-1619.
3. Shillito RD, Whitt S, Ross M, Ghavami F, De Vleesschauwer D, D'Halluin K et al. 2021. Detection of genome edits in plants-from editing to seed. In Vitro Cell Dev Biol Plant. 57(4):595-608.
4. Kim D, Alptekin B, Budak H. 2018. CRISPR/Cas9 genome editing in wheat. Funct Integr Genomics. 18(1):31-41.
5. Prasad K, George A, Ravi NS, Mohankumar KM. 2021. CRISPR/Cas based gene editing: marking a new era in medical science. Mol Biol Rep. 48(5):4879-4895.
6. Bibikova M, Beumer K, Trautman JK, Carroll D. 2003. Enhancing gene targeting with designed zinc finger nucleases. Science. 300(5620):764..
7. Christian M, Cermak T, Doyle EL, Schmidt C, Zhang F, Hummel A *et al*. 2019. Targeting DNA double-strand breaks with TAL effector nucleases. Genetics. 186(2):757-761.
8. Jinek M, Chylinski K, Fonfara I, Hauer M, Doudna JA, Charpentier E. 2012. A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. Science. 337(6096):816-821.
9. Ceccaldi R, Rondinelli B, D'Andrea AD. 2016. Repair pathway choices and consequences at the double-strand break. Trends Cell Biol. 26(1):52-64.
10. Sioud M: RNA and CRISPR Interferences: Past, Present, and Future Perspectives. In: Sioud M. (eds) RNA Interference and CRISPR Technologies. Methods in Molecular Biology. Volume 2115. New York: Humana; 2020:1-22.
11. Yan AL, Du SW, Palczewski K. 2023. Genome editing, a superior therapy for inherited retinal diseases. Vision Res. 206:108192.
12. Kabadi AM, Ousterout DG, Hilton IB, Gersbach CA. 2014. Multiplex CRISPR/Cas9-based genome engineering from a single lentiviral vector. Nucleic Acids Res. 42(19):e147.
13. Paquet D, Kwart D, Chen A, Sproul A, Jacob S, Teo S *et al*. 2016. Efficient introduction of specific homozygous and heterozygous mutations using CRISPR/Cas9. Nature. 533:125-129.
14. Haapaniemi E, Botla S, Persson J, Schmierer B, Taipale J. 2018. CRISPR-Cas9 genome editing induces a p53-mediated DNA damage response. Nat Med. 24(7):927-930.
15. Komor AC, Kim YB, Packer MS, Zuris JA, Liu DR. 2016. Programmable editing of a target base in genomic DNA without double-stranded DNA cleavage. Nature. 533:420-424.
16. Li JY, Zhang C, He YB, Li SY, Yan L, Li YC, *et al*. 2023. Plant base editing and prime editing: The current status and future perspectives. J Integr Plant Biol. 65(2):444-467.
17. Rees HA, Liu DR. 2018. Base editing: precision chemistry on the genome and transcriptome of living cells. Nat Rev Genet. 19(12):770-788.
18. Chen J, Wu Z, Chen R, Huang Z, Han X, Qiao R, *et al*. 2022. Identification of genomic regions and candidate genes for litter traits in French large white pigs using genome-wide association studies. Animals. 12:1584.
19. Sichani AS, Ranjbar M, Baneshi M, Zadeh FT, Fallahi J. 2023. A review on advanced CRISPR-based genome-editing tools: Base editing and prime editing. Mol Biotechnol. 65(6):849-860.
20. Gaudelli NM, Komor AC, Rees HA, Packer MS, Badran AH, Bryson DI, *et al*. 2017. Programmable base editing of AT to GC in genomic DNA without DNA cleavage. Nature. 554:464-471.
21. Guo MX, Chen HY, Dong ST, Zhang Z, Luo HM. 2022. CRISPR-Cas gene editing technology and its application prospect in medicinal plants. Chin Med. 17(1):33.
22. Harbottle JA. 2021. Immunotherapy to get on point with base editing. Drug Discovery Today. 26(10):2350-2357.
23. Nishimasu H, Ran FA, Hsu Patrick D, Konermann S, Shehata Soraya I, Dohmae N, *et al*. 2014. Crystal structure of Cas9 in complex with guide RNA and target DNA. Cell. 156(5):935-949.
24. Liang YH, Chen FB, Wang KP, Lai LX. 2023. Base editors: development and applications in biomedicine. Front Med. 17(3):359-387.
25. Singh K, Kaur R, Deshmukh R: Biotechnological advances for disease tolerance in plants. Singapore: Spinger. 2024:293-316
26. Liu N, Zhou LF, Qu JY, Yao SH. 2021. Recent advances in CRISPR/Cas9 directed base editing. Curr Gene Ther. 21(4):327-337.
27. Kurt IC, Zhou RH, Iyer S, Garcia SP, Miller BR, Langner LM, *et al*. 2021. CRISPR C-to-G base editors for inducing targeted DNA transversions in human cells. Nat Biotechnol. 39(1):41-46.
28. Kantor A, McClements ME, MacLaren RE. 2020. CRISPR-Cas9 DNA base-editing and prime-editing. Int J Mol Sci. 21(17):6240.
29. Xu X, Yang S, Hao H, Du W, Pang Y, Zhao S, *et al*. 2021. Research progress on application of single base editing technology. China Anim Husb Vet Med. 48(12):4403-4411.
30. Zheng JT, Yan NN, Zuo EW. 2021. Research advancement in CRISPR-based genome editing technologies. J Fujian Med Univ. 55(03):187-191.
31. Tsai SQ, Zheng Z, Nguyen NT, Liebers M, Topkar VV, Thapar V, *et al*. 2015. GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. Nat Biotechnol. 33(2):187-197.

32. Kim D, Bae S, Park J, Kim E, Kim S, Yu HR, *et al*. 2015. Digenome-seq: genome-wide profiling of CRISPR-Cas9 off-target effects in human cells. Nat Methods. 12(3):237-243.
33. Cameron P, Fuller CK, Donohoue PD, Jones BN, Thompson MS, Carter MM, *et al*. 2017. Mapping the genomic landscape of CRISPR-Cas9 cleavage. Nat Methods. 14(6):600-606.
34. Tsai SQ, Nguyen NT, Malagon-Lopez J, Topkar VV, Aryee MJ, Joung JK. 2017. CIRCLE-seq: a highly sensitive *in vitro* screen for genome-wide CRISPR-Cas9 nuclease off-targets. Nat Methods. 14(6):607-614.
35. Frock RL, Hu J, Meyers RM, Ho YJ, Kii E, Alt FW. 2015. Genome-wide detection of DNA double-stranded breaks induced by engineered nucleases. Nat Biotechnol. 33(2):179-186.
36. Yan WX, Mirzazadeh R, Garnerone S, Scott D, Schneider MW, Kallas T, *et al*. 2017. BLISS is a versatile and quantitative method for genome-wide profiling of DNA double-strand breaks. Nat Commun. 8:15058.
37. Lin J, Zhang Z, Zhang S, Chen J, Wong K. 2020. CRISPR-Net: A recurrent convolutional network quantifies CRISPR off-target activities with mismatches and indels. Adv Sci. 7(13):1-17.
38. Stemmer M, Thumberger T, Del Sol Keyer M, Wittbrodt J, Mateo JL. 2015. CCTop: An intuitive, flexible and reliable CRISPR/Cas9 target prediction tool. PLoS One. 10(4):e0124633.
39. Hsu PD, Scott DA, Weinstein JA, Ran FA, Konermann S, Agarwala V, *et al*. 2013. DNA targeting specificity of RNA-guided Cas9 nucleases. Nat Biotechnol. 31(9):827-832.
40. Singh R, Kuscu C, Quinlan A, Qi Y, Adli M. 2015. Cas9-chromatin binding information enables more accurate CRISPR off-target prediction. Nucleic Acids Res. 43(18):e118.
41. Doench JG, Fusi N, Sullender M, Hegde M, Vaimberg EW, Donovan KF, *et al*. 2016. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat Biotechnol. 34(2):184-191.
42. Kim D, Luk K, Wolfe SA, Kim JS. 2019. Evaluating and enhancing target specificity of gene-editing nucleases and deaminases. Annu Rev Biochem. 88:191-220.
43. Listgarten J, Weinstein M, Kleinstiver BP, Sousa AA, Joung JK, Crawford J, *et al*. 2018. Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. Nat Biomed Eng. 2(1):38-47.
44. Alkan F, Wenzel A, Anthon C, Havgaard JH, Gorodkin J. 2018. CRISPR-Cas9 off-targeting assessment with nucleic acid duplex energy parameters. Genome Biol. 19(1):177-190.
45. Chuai G, Ma H, Yan J, Chen M, Hong N, Xue D, *et al*. 2018. DeepCRISPR: optimized CRISPR guide RNA design by deep learning. Genome Biol. 19(1):80-98.
46. Jiecong L, Ka-Chun W. 2018. Off-target predictions in CRISPR-Cas9 gene editing using deep learning. Bioinformatics. 34(17):i656-i663.
47. Arbab M, Shen MW, Mok B, Wilson C, Matuszek Z, Cassa CA, *et al*. 2020. Determinants of base editing outcomes from target library analysis and machine learning. Cell. 182(2):463-480.
48. Sapoval N, Aghazadeh A, Nute MG. 2022. Current progress and open challenges for applying deep learning across the biosciences. Nat Commun. 13:1728.
49. Wang ZL, Liang JB, Wang XY. 2022. BE-dot: a tool for sgRNA design and off-target profile prediction of base editing. Prog Biochem Biophys. 50(02):397-404.
50. Gehrke JM, Cervantes O, Clement MK, Wu Y, Zeng J, Bauer DE, *et al*. 2018. An APOBEC3A-Cas9 base editor with minimized bystander and off-target activities. Nat Biotechnol. 36(10):977-982.
51. Tang JL, Lee T, Sun T. 2019. Single-nucleotide editing: From principle, optimization to application. Hum Mutat. 40(12):2171-2183.
52. Koblan LW, Arbab M, Shen MW, Hussmann JA, Liu DR. 2021. Efficient CG-to-GC base editors developed using CRISPRi screens, target-library analysis, and machine learning. Nat Biotechnol. 39:1414-1425.
53. Song M, Kim HK, Lee S, Kim Y, Seo SY, Park J, *et al*. 2020. Sequence-specific prediction of the efficiencies of adenine and cytosine base editors. Nat Biotechnol. 38(9):1037-1043.
54. Li B, Li YQ, Zhao D, Yang J, Ma YH, Bi CH, *et al*. 2022. Sequence motifs and prediction model of GBE editing outcomes based on target library analysis and machine learning. J Genet Genomics. 3:254-257.
55. Yuan TL, Yan NN, Fei TY, Zheng JT. 2021. Optimization of C-to-G base editors with sequence context preference predictable by machine learning methods. Nat Commun. 12(1):4902.
56. Zhao DD, Li J, Li SW, Xin XQ, Hu MZ, Price MA, *et al*. 2021. Glycosylase base editors enable C-to-A and C-to-G base changes. Nat Biotechnol. 39(1):35-40.
57. Chen LW, Park JE, Paa P, Rajakumar PD, Prekop HT, Chew YT, *et al*. 2021. Programmable C:G to G:C genome editing with CRISPR-Cas9-directed base excision repair proteins. Nat Commun. 12(1):1384.
58. Szegedy C, Liu W, Jia Y, Sermanet P, Rabinovich A. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. IEEE:1-9.
59. Marquart KF, Allam A, Janjuha S, Sintsova A, Villiger L, Frey N, *et al*. 2021. Predicting base editing outcomes with an attention-based deep learning algorithm trained on high-throughput target library screens. Nat Commun. 12(1):5114-5123.
60. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, *et al*. Attention is all you need. In: NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017. ACM:6000–6010.
61. Min B, Zhang X, Zhang CP, Zhang X. 2024. Prediction of bearing capacity of cracked asymmetrical double-arch tunnels using the artificial neural networks. Eng Fail Anal. 156:107805.
62. Ananth P, Elinmadli P, Jonas K, Juliane W, Thomas V, Uyenlinh H, *et al*. 2022. Predicting base editing outcomes using position-specific sequence determinants. Nucleic Acids Res. 50(6):3551-3564.
63. Kim N, Choi S, Kim S, Song M, Seo JH, Min S, *et al*. 2024. Deep learning models to predict the editing efficiencies and outcomes of diverse base editors. Nat Biotechnol. 42:484–497.
64. Yuan TL, Wu LL, Li SY, Zheng JT, Li NA, Xiao X, *et al*. 2024. Deep learning models incorporating endogenous factors beyond DNA sequences improve the prediction accuracy of base editing outcomes. Cell Discovery. 10(20):1-19.

65. Liu QY, Cheng X, Liu G, Li BH, Liu XQ. 2020. Deep learning improves the ability of sgRNA off-target propensity prediction. BMC Bioinf. 21(1):1-22.
66. Zhang YF, Pan JL, Liu WW, Chen ZB, Li KL, Wang J, *et al*. 2024. Kullback-Leibler divergence-based out-of-distribution detection with flow-based generative models. IEEE Trans Knowl Data Eng. 36(4):1683-1697.