Supplementary information

Deep learning models simultaneously trained on multiple datasets improve base-editing activity prediction

Ying Sun[1*], Kunli Qu[2*], Giulia I. Corsi[1*], Christian Anthon[1*], Xiaoguang Pan[2*], Xi Xiang[3], Lars Juhl Jensen[4], Lin Lin[5,6], Yonglun Luo[5,6†], Jan Gorodkin[1†]

[1] Center for non-coding RNA in Technology and Health, Department of Veterinary and Animal Sciences, Faculty of Health and Medical Sciences, University of Copenhagen, Frederiksberg, Denmark

[2] Department of Biology, University of Copenhagen, Copenhagen N, Denmark

[3] Scientific Research Center, The Seventh Affiliated Hospital of Sun Yat-sen University, Shenzhen, Guangdong 518107, China

[4] Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark

[5] Department of Biomedicine, Aarhus University, Aarhus C, Denmark

[6] Steno Diabetes Center Aarhus, Aarhus University Hospital, Aarhus N, Denmark

[*]These authors contributed equally.
[†]To whom corresponding should be addressed: Yonglun Luo, alun@biomed.au.dk, Jan Gorodkin, gorodkin@rth.dk

Supplementary Note 1. Deep learning model for base editor efficiency prediction

Accurate prediction of designed gRNA editing efficiency and edited outcomes from base editors accelerates their applications. Considering the bystander edits from base editors, the dependency between outcome frequencies from one gRNA is considered. Currently, several methods have already been designed for base editor efficiency prediction, such as DeepABE/CBE[1], BE-HIVE[2], BE-DICT[3], BE_Endo[4], BEDICT2.0[5] and DeepBE[6]. Our approach for predicting base editor efficiency and outcomes differs from existing methods in that: 1) We directly predict the both the gRNA efficiency and outcome frequency instead of predicting editing efficiency and outcome proportion separately and multiplying these values as the final outcome frequency, as proposed in DeepABE/CBE[1], BE-HIVE[2], BEDICT2.0[5] and DeepBE[6]. According to the definition of outcome proportion, which is the number of outcome reads divided by the total edited reads, the outcome proportion might be close to 100% for an inefficient gRNA, making outcome proportion less meaningful and informative than outcome frequency. Therefore, we chose to predict outcome frequency directly in our models. 2) Considering the bystander edits in base editors resulting in the dependency between outcome frequencies for a single gRNA, we applied a convolutional neural network (CNN)-based model with gRNA editing efficiency and outcome frequency predicted simultaneously (using two outputs), rather than the autoregressive neural network-based, transformer-based or Markov network-based model proposed by Arbab *et al.,*[2], Marquart *et al.,*[3], Kissling *et al*.,[5] and Yuan *et al.,*[4]. Compared with these models considering joint distribution from outcomes, the output with predicted gRNA editing efficiency into consideration also addresses the challenges presented by these dependencies.

Considering that gRNA editing efficiency from the gRNAs with multiple outcomes is predicted multiple times in our model, we calculated the range of predicted gRNA editing efficiency by calculating the maximum predicted gRNA editing efficiency minus the minimum predicted gRNA editing efficiency for each gRNA. The range of predicted gRNA editing efficiency is less than 3% for both ABE and CBE (**Supplementary Fig. 10**). This demonstrates that our model setup has a robust gRNA editing efficiency prediction. To ensure full consistency between predicted efficiencies and corresponding predicted outcome frequencies, we first averaged the predicted values of a gRNA for the cases where the same gRNA is in an output pair with different edit outcomes. Then each predicted outcome frequency was further normalized by multiplying the obtained average of the predicted gRNA editing efficiency to the fraction of the predicted outcome frequency divided by sum of predicted edited outcome frequencies for the given gRNA. These values after normalization are used for the final model performance evaluation, e.g. two-dimensional R_2 and ρ_2 (see detailed calculation in **Supplementary Note 4).**

Supplementary Note 2. Incorporating CRISPRon prediction for base editor efficiency prediction

To enhance the performance of base editor efficiency prediction, we, in light of the correlation between CRISPR/Cas9-gRNA indel frequency and base editor transition efficiency[1] (**Supplementary Fig. 1**), integrated the predicted Cas9-gRNA efficiencies into our model. We employed the CRISPRon method[7], a deep learning-based model predicting Cas9-gRNA indel frequency, to add predicted gRNA efficiencies as a feature into our model. Two versions of CRISPRon are available: CRISPRon_V0 (trained with five validation sets) and CRISPRon_V1

(trained with all six validation sets). To prevent data leakage from CRISPRon, we carefully selected CRISPRon_V0 for predicting Cas9 efficiency. This leaves 868 gRNAs of ABE and 1,297 gRNAs of CBE from our SURRO-seq Test and 430 gRNAs of ABE and 474 gRNAs of CBE from Song Test that have never been used for CRISPRon_V0 training. Hence, they are further selected as independent tests.

Supplementary Note 3. Analysis of features important for base editor efficiency prediction

To characterize the most important features for predicting base editor gRNA editing efficiency, we employed the Shapley Additive exPlanations (SHAP) analysis[8] based on an XGBoost regression model[9] on SURRO-seq datasets. Initially, we extracted sequence context features following the approach in CRISPRon (CRISPRon-GBRT V0 and V1)[7], including: one-hot encoded positional-specific single and di-nucleotides for 30nt target DNA sequence, one-hot encoded nucleotides surrounding the "GG" Cas9 binding sites, *k*-mer (*k* = 1 or 2) counts from 30nt target DNA sequence and GC content of 20nt gRNA sequences. Additionally, we extracted molecular features such as the predicted Cas9 efficiency from CRISPRon (CRISPRon_V0)[7], the minimum free energy from the 20nt gRNA (MFE_spacer), the minimum free energy from the 20nt gRNA and scaffold (MFE_extension) calculated by RNAfold (Version 2.5.1)[10], the RNA-DNA binding energy ΔG_B calculated by CRISPRoff (Version 1.1.2)[11], and the melting temperatures at positions within the editing window calculated using the Biopython 1.79 Tm_NN formula[12]. The XGBoost regression model underwent training using five-fold cross-validation on five partitions of the dataset, following the same procedure as in the deep learning model. Subsequently, SHAP values were calculated, and the important features for ABE and CBE were visualized in **Supplementary Fig. 3**. Notably, the predicted Cas9 efficiency from CRISPRon emerged as the most important feature for both ABE and CBE. We also carried out the Gini importance analysis[13], but that did not provide anything new in addition to the SHAP analysis.

Supplementary Note 4. *K*-dimensional correlation coefficient for jointly evaluation of gRNA editing efficiency and outcome frequency

To extend Pearson correlation coefficient and Spearman rank correlation coefficient, we consider two datasets *X* and *Y* with *N* rows and *K* columns, represented by a *N* * *K* matrix. The covariance between *X* and *Y* was defined by Gorodkin[14] as:

$$COV\left(\underline{X},\underline{Y}\right) = \sum_{k=1}^{K} w_k COV\left(\underline{X}_k,\underline{Y}_k\right) = \frac{1}{K}\sum_{n=1}^{N}\sum_{k=1}^{K}\left(X_{nk}-\overline{X}_k\right)\left(Y_{nk}-\overline{Y}_k\right), \quad (1)$$

where $\bar{X}_k = \frac{1}{N}\sum_{n=1}^{N} X_{nk}$, $\bar{Y}_k = \frac{1}{N}\sum_{n=1}^{N} Y_{nk}$ and $w_k = \frac{1}{K}$. Hence X_{nk} is the *kth* coordinate of the *nth* datapoint.

The correlation coefficient is defined as:

$$R_K = \frac{COV(\underline{X},\underline{Y})}{\sqrt{COV(\underline{X},\underline{X})COV(\underline{Y},\underline{Y})}}. \quad (2)$$

To calculate *K*-dimensional Pearson correlation coefficient (R_K), *X* and *Y* are used as the input for calculations.

To calculate K-dimensional Spearman rank correlation coefficient (ρ_K), the formula is different depending on the tie (identical number) in the dataset or not. If ties are detected in the dataset, the X and Y are the rank for X and Y and ρ_K can be calculated using Formula 2. If no ties are detected in the dataset, ρ_K can be shown to take the similar form for $K = 1$, the standard Spearman rank correlation coefficient:

$$\rho_K = 1 - \frac{6\sum_{k=1}^{K}\sum_{n=1}^{N} d_{nk}^2}{KN(N^2-1)}, \quad (3)$$

where d_{nk} is the rank of the *kth* coordinate of the *nth* data point.

In the context of base editor efficiency prediction model, we proposed to evaluate its performance by comparing ground truth and prediction results using combined gRNA editing efficiency and outcome frequency with the $K = 2$. Given the gRNA editing efficiency is displayed multiple times as the outcomes and outcomes without reads supported with a frequency equal to 0, ties would be detected in X, leading to the use of Formula 2 for K-dimensional Spearman rank correlation coefficient calculation. Finally, for the evaluation of base editor efficiency prediction, we utilized two-dimensional R_2 and ρ_2 (Formula 1, 2) instead of the one-dimensional Pearson and Spearman rank correlation coefficients.

Supplementary Note 5. Diverse dataset from different studies

The ABE and CBE datasets from separate studies exhibit distinct editing efficiency distributions (**Fig. 2A, 2B**). Beyond the variations in paired gRNA-target sequences, the observed distinctions can be elucidated through the following considerations (see summary in **Fig. 2C** and **Supplementary Fig. 6**):

(1) Arbab *et al*. and Kissling *et al*. enhanced the stability of base editors by implementing a modified single guide RNA structure with an extended stem and modified nucleotide[2]. Growing evidence suggests that different scaffold sequences with diverse self-folding or interacting patterns with gRNAs can influence gRNA efficiencies[15]. The modified sgRNA in Arbab and Kissling datasets may introduce variations in editing efficiency and resulting edited products.

(2) Despite utilizing the same deaminase types (ABE7.10 and BE4) for dataset generation, the differences exist across datasets. For CBE, we utilized the full-length CBE with Gam protein connected (Addgene plasmid ID # 100806), while Song *et al.* employed the split CBE editor without Gam (Addgene plasmid ID # 100802). The inclusion of Gam protein (a bacteriophage Mu protein) has been verified to reduce the indel formation of base editing and improve the product purity[16], introducing a potential source of divergence in the datasets. Moreover, Marquart *et al.* and Arbab *et al.* chose a different CBEmax expression plasmid (Addgene plasmid ID # 152991) for CBE editing experiments. For ABE, Song *et al.* and we selected Addgene plasmid ID # 102919, while Arbab *et al.*, Marquart *et al.* and Kissling *et al.* employed Addgene plasmid ID # 152989. Notably, a split ABE was applied in Song's experiments.

(3) Variations also exist in the number of bystander edits. Marquart Test exhibited a single target nucleotide edit in the editing window, while other datasets revealed multiple bystander edits. In ABE, one to three edited adenines were frequently observed, while in CBE, one to four neighboring cytosines could be edited simultaneously (**Supplementary Fig. 6**).

(4) Deaminases present different motif preferences and can lead to different outcome products[2, 6]. Compared with ABE7.10, ABE8e presents a high bystander edit activity[5] (**Supplementary Fig. 6).**

(5) There are other experimental factors contributing to the potential divergence in datasets, such as library delivery methods[5], expression levels of the base editors, cell harvesting time to allow CRISPR base editing happening, and multiplicity of infection (MOI) when using viral libraries.

Understanding these factors is crucial for contextualizing the observed variations in the editing efficiency across datasets and providing the insights for choosing the suitable models for efficiency prediction.

Supplementary Table 1. Description of dataset feature in the fused model. The column "Editor" shows the base editor ABE or CBE. The column "Dataset feature" shows the input features in the model. The column "Name" shows the corresponding name for our model. For example, if we employ feature [1,0,0] in our CBE model for prediction, the model is called CRISPRon-CBE(100). The column "Usage" shows this feature is used for training model or testing on independent test sets. The column "Description" explains how to employ this dataset feature to train a fused dataset from different sources and how to provide tailormade prediction on new dataset.

Editor	Dataset feature	Name	Usage	Description
ABE	[1, 0, 0, 0, 0]	N.a	Training	Training using SURRO-seq dataset
	[0, 1, 0, 0, 0]	N.a	Training	Training using Song Train
	[0, 0, 1, 0, 0]	N.a	Training	Training using Arbab dataset
	[0, 0, 0, 1, 0]	N.a	Training	Training using Kissling ABE7.10 dataset
	[0, 0, 0, 0, 1]	N.a	Training	Training using Kissling ABE8e dataset
	[1, 0, 0, 0, 0]	10000	Test	Model tuned to SURRO-seq dataset
	[0, 1, 0, 0, 0]	01000	Test	Model tuned to Song dataset
	[0, 0, 1, 0, 0]	00100	Test	Model tuned to Arbab dataset
	[0, 0, 0, 1, 0]	00010	Test	Model tuned to Kissling ABE7.10 dataset
	[0, 0, 0, 0, 1]	00001	Test	Model tuned to Kissling ABE8e dataset
	[0.5, 0.5, 0, 0, 0]	22000	Test	Model with equal weight to SURRO-seq and Song datasets
	[0, 0, 0.5, 0.5, 0]	00220	Test	Model with equal weight to Arbab and Kissling ABE7.10 datasets
	[0.25, 0.25, 0.25, 0.25, 0]	44440	Test	Model with equal weight to four ABE7.10 datasets
CBE	[1, 0, 0]	N.a	Training	Training using SURRO-seq dataset
	[0, 1, 0]	N.a	Training	Training using Song Train
	[0, 0, 1]	N.a	Training	Training using Arbab dataset
	[1, 0, 0]	100	Test	Model tuned to SURRO-seq dataset
	[0, 1, 0]	010	Test	Model tuned to Song dataset
	[0, 0, 1]	001	Test	Model tuned to Arbab dataset
	[0.5, 0.5, 0]	220	Test	Model with equal weight to SURRO-seq and Song datasets
	[0.33, 0.33, 0.33]	333	Test	Model with equal weight to all three datasets

N.a: not applicable for training

Supplementary Table 2. Primers designed for targeted deep sequencing. The genotypings of ABE and CBE are amplified using the paired "ABE7.10-iden" and "BE4-iden" primers, respectively. The synthetic oligos are amplified using paired "TRAP-oligo (BsmBI GGA)" primers. The surrogate target sites from both plasmids and genomic DNA are amplified using paired "TRAP-NGS" primers.

Primer Name	**Primer sequence(5'-3')**
ABE7.10-iden-F	CAGTACTCGTGCTCAACAATCG
ABE7.10-iden-R	GGCGTTGCGAACACCGAATACA
BE4-iden-F	TTCTTCGATCCGAGAGAGCTCC
BE4-iden-R	CTGCACCTTGTGTTCGGACAG
TRAP-oligo (BsmBI GGA)-F	TACAGCTaccacgtctcaCACC
TRAP-oligo (BsmBI GGA)-R	AGCACAAccgtcgtctccAAAC
TRAP-NGS-F1	GGACTATCATATGCTTACCGTA
TRAP-NGS-R1	ACTCCTTTCAAGACCTAGCTAG

Supplementary Table 3. Description of dataset for machine learning. The column "Editor" shows the base editor ABE or CBE. The column "Dataset" shows the dataset used in this study. The column "#gRNAs" shows the number of gRNAs in this raw dataset. The column "File/sheet name" and "Reference" illustrate the original source of the raw dataset. The column "Purpose" points out this dataset is used for training or test in this study.

Editor	Dataset	#gRNAs	File/sheet name	Purpose	Reference
ABE	SURRO-seq	11,484	Supplementary Data 1	Training/Test	This study
	Song Train	10,209	HT_ABE_Train	Training	Song *et al.*[1]
	Song Test	438	HT_ABE_Test	Test	Song *et al.*[1]
	Arbab dataset	6,633	HEK293T_12kChar_ABE	Training/Test	Arbab *et al.*[2]
	Marquart Test*	1,667	BE-HIVE_marquart_freq_ABE	Test	Marquart *et al.*[3]
	Kissling ABE7.10	11,911	Provided by authors	Training/Test	Kissling *et al*.[5]
	Kissling ABE8e	11,906	Provided by authors	Training/Test	Kissling *et al*.[5]
	Arbab mESC Test	7,585	mES_12kChar_ABE	Test	Arbab *et al.*[2]
	Arbab U2OS Test	1,400	U2OS_12kChar_ABE	Test	Arbab *et al.*[2]
CBE	SURRO-seq	11,406	Supplementary Data 1	Training/Test	This study
	Song Train	9,743	HT_CBE_Train	Training	Song *et al*.[1]
	Song Test	483	HT_CBE_Test	Test	Song *et al.*[1]
	Arbab dataset	7,156	HEK293T_12kChar_BE4	Training/Test	Arbab *et al.*[2]
	Marquart Test*	1,675	BE-HIVE_marquart_freq_CBE	Test	Marquart *et al*.[3]
	Arbab mESC Test	6,258	mES_12kChar_BE4	Test	Arbab *et al.*[2]
	Arbab U2OS Test	1,270	U2OS_12kChar_BE4	Test	Arbab *et al.*[2]

*The training sets for ABEmax and CBE and all the dataset for ABE8e from Marquart *et al.*[3] were not collected because their 20nt sequences were not appliable for our model and their flanking sequences were not provided.

Supplementary Table 4. Number of gRNAs before and after processing. The column "Editor" shows the base editor ABE or CBE. The column "Preprocessing steps" shows the detailed processing steps to filter the dataset. The other columns show that how many gRNAs are left after these preprocessing steps. The dataset is preprocessing by the following steps: 1). Keeping the gRNAs with "NGG" PAM at target sites; 2). Keeping the gRNAs with at least 100 total reads; 3). Keeping the gRNAs with less than 100,000 total reads; 4). Keeping the gRNAs with at least one target nucleotide ("A" for ABE and "C" for CBE) in the editing window; 5). Keeping the gRNAs in the training set with at least 100 edited reads.

Editor	Preprocessing steps	SURRO-seq	Song Train	Song Test*	Arbab dataset†	Marquart Test	Kissling ABE7.10	Kissling ABE8e
ABE	All gRNAs	11,484	10,209	430	6,633	1,667	11,911	11,906
	"NGG" PAM	11,484	10,209	430	6,633	1,667	1,170	1,170
	Total reads ⩾ 100	11,484	10,209	430	6,631	1,667	1,162	1,151
	Total reads ⩽ 100,000	11,484	10,207	430	6,631	1,667	1,162	1,151
	⩾1 target nucleotide in editing window	11,474	10,207	430	6,607	1,659	1,103	1,092
	Edited reads ⩾ 100 (training set only)	5,314	4,983	N.a	6,607	N.a	966	1,023
CBE	All gRNAs	11,406	9,743	475	7,156	1,675	/	/
	"NGG" PAM	11,406	9,743	475	7,156	1,675	/	/
	Total reads ⩾ 100	11,406	9,743	475	7,156	1,675	/	/
	Total reads ⩽ 100,000	11,406	9,741	474	7,155	1,675	/	/
	⩾ 1 target nucleotide in editing window	11,353	9,741	474	7,095	1,612	/	/
	Edited reads ⩾ 100 (training set only)	7,962	3,975	N.a	7,095	N.a	/	/

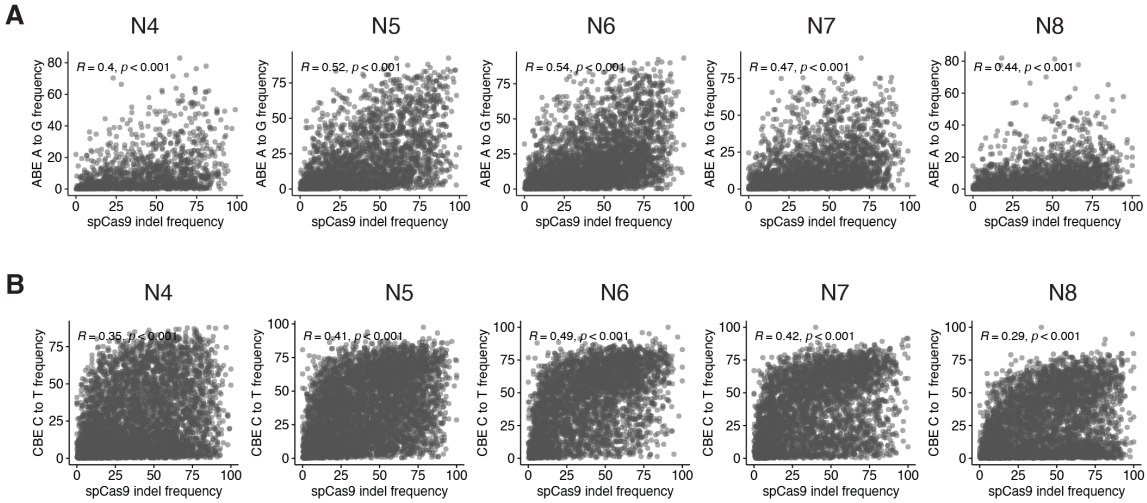
*8 identical gRNAs between Song Train and Song Test for both ABE and CBE were removed
†The initial Arbab dataset was already preprocessed using the same 100 edited reads threshold applied in this study.
N.a: not applicable for test sets.
/: No available dataset

Supplementary Table 5. Composition of datasets for model development. This table illustrates the number of data points among datasets used in this study. The column "#gRNAs" shows the number of gRNAs in this dataset after above preprocessing steps. The column "#outcomes" shows the number of all the potential outcomes in this dataset, including the outcome sequence identical to gRNA sequence and the outcomes not measured by experiments. The column "#outcomes measured by experiments" shows the number of outcomes measured by experiments.

Editor	Dataset	#gRNAs	#outcomes	#outcomes measured by experiments
ABE	SURRO-seq Dataset	5,314	34,100	21,848
	Song Train	4,983	39,196	28,880
	Song Test	430	2,626	2,162
	Arbab Dataset	6,607	46,530	32,055
	Kissling ABE7.10 Dataset	966	7,772	5,472
	Kissling ABE8e Dataset	1,023	7,982	4,645
	Marquart Test	1,659	10,314	4,296
CBE	SURRO-seq Dataset	7,962	77,660	52,917
	Song Train	3,975	43,728	27,300
	Song Test	474	3,884	2,736
	Arbab Dataset	7,095	57,270	34,555
	Marquart Test	1,612	10,964	4,049

Supplementary Table 6. Counts of gRNAs from various datasets and distribution among six partitions. Each dataset (SURRO-seq, Song, Arbab, Kissling ABE7.10 and Kissling ABE8e dataset) is split into six partitions, among which five partitions used for training and the 6th partition for test. For each dataset, the row "#gRNA" shows the number of gRNAs in this partition. The row "#outcome" shows the number of all the potential outcomes in this partition.

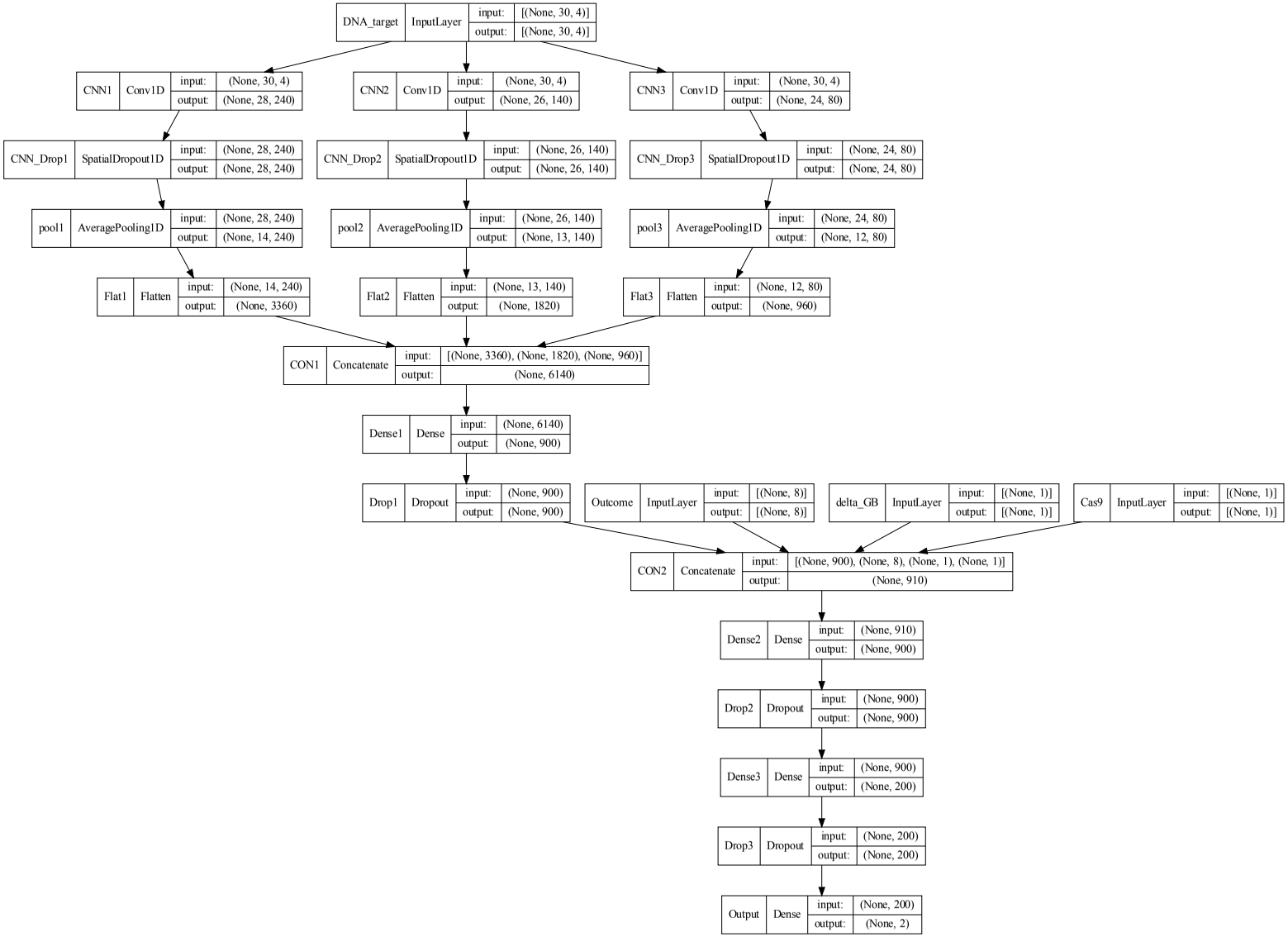
Editor	Dataset source	Dataset type	Partition1 (training)	Partition2 (training)	Partition3 (training)	Partition4 (training)	Partition5 (training)	Partition6 (test)
ABE	SURRO-seq	#gRNA	886	886	886	886	885	885
		#outcome	5,750	5,508	5,424	5,798	5,582	6,038
	Song	#gRNA	997	997	997	996	996	430
		#outcome	7,736	7,752	8,140	7,768	7,800	2,626
	Arbab	#gRNA	1,102	1,101	1,101	1,101	1,101	1,101
		#outcome	7,868	7,558	7,882	7,746	8,054	7,422
	Kissling ABE7.10	#gRNA	161	161	161	161	161	161
		#outcome	1,360	1,200	1,412	1,204	1,232	1,364
	Kissling ABE8e	#gRNA	171	171	171	170	170	170
		#outcome	1,384	1,238	1,356	1,308	1,254	1,442
CBE	SURRO-seq	#gRNA	1,327	1,327	1,327	1,327	1,327	1,327
		#outcome	13,202	12,524	13,576	12,674	13,024	12,660
	Song	#gRNA	795	795	795	795	795	474
		#outcome	8,496	8,756	8,920	8,420	9,136	3,884
	Arbab	#gRNA	1,183	1,183	1,183	1,182	1,182	1,182
		#outcome	9,482	9,906	9,594	9,410	9,396	9,482



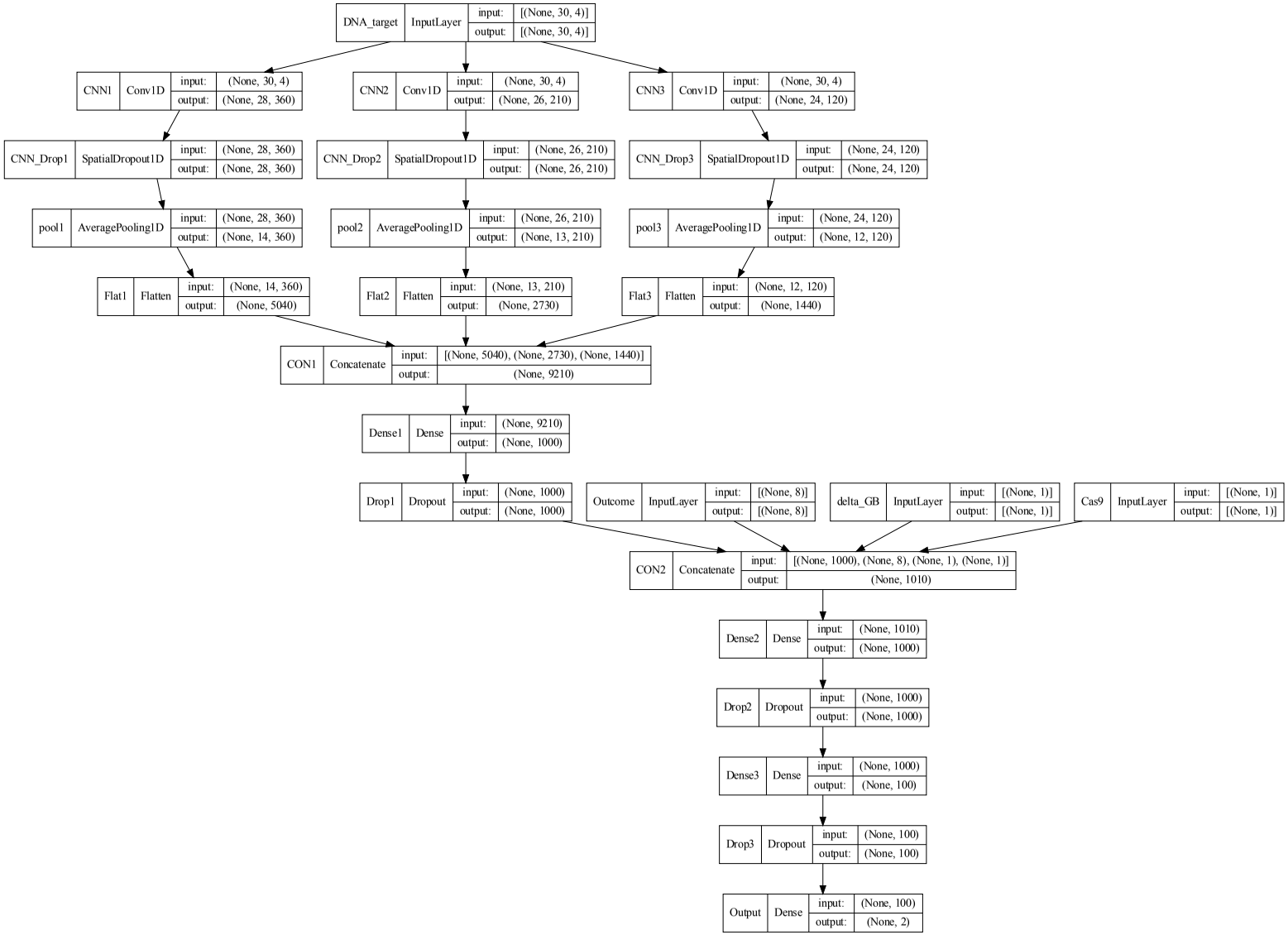
Supplementary Fig. 1: **Correlation between Cas9-induced indel frequency and base editor-induced nucleotide conversion efficiency using datasets generated by SURRO-seq technology**[17].

The experimental Cas9-induced indel frequency dataset was collected from CRISPRon study[7]. N represents the target nucleotide position in the gRNA where N1 is defined as the distal base pair of the target site from the PAM. (A) ABE; (B) CBE. Source data are provided as a Source Data file.

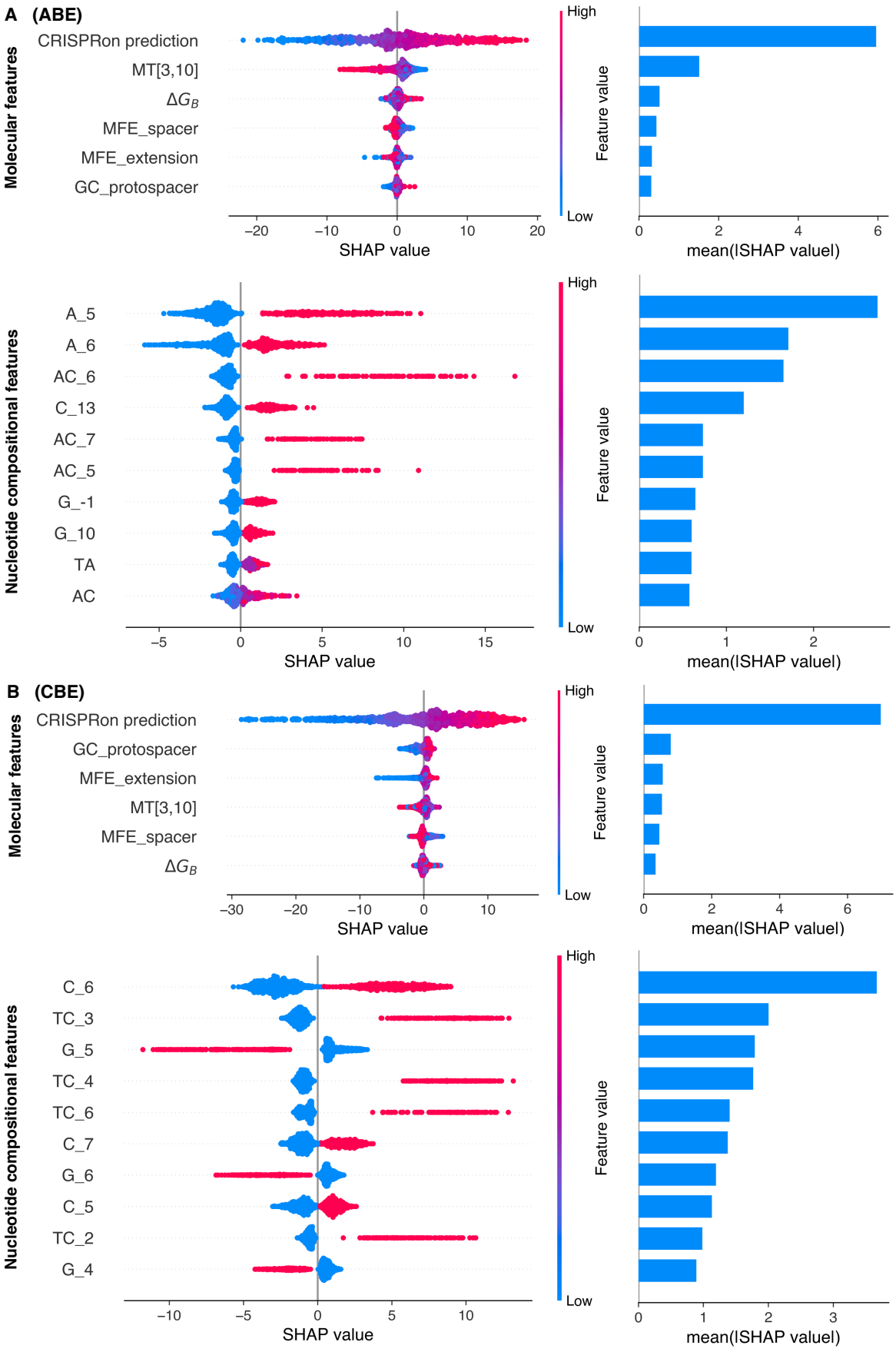
(A) ABE


DNA_target | InputLayer | input: [(None, 30, 4)] | output: [(None, 30, 4)]
CNN1 | Conv1D | input: (None, 30, 4) | output: (None, 28, 240)
CNN2 | Conv1D | input: (None, 30, 4) | output: (None, 26, 140)
CNN3 | Conv1D | input: (None, 30, 4) | output: (None, 24, 80)
CNN_Drop1 | SpatialDropout1D | input: (None, 28, 240) | output: (None, 28, 240)
CNN_Drop2 | SpatialDropout1D | input: (None, 26, 140) | output: (None, 26, 140)
CNN_Drop3 | SpatialDropout1D | input: (None, 24, 80) | output: (None, 24, 80)
pool1 | AveragePooling1D | input: (None, 28, 240) | output: (None, 14, 240)
pool2 | AveragePooling1D | input: (None, 26, 140) | output: (None, 13, 140)
pool3 | AveragePooling1D | input: (None, 24, 80) | output: (None, 12, 80)
Flat1 | Flatten | input: (None, 14, 240) | output: (None, 3360)
Flat2 | Flatten | input: (None, 13, 140) | output: (None, 1820)
Flat3 | Flatten | input: (None, 12, 80) | output: (None, 960)
CON1 | Concatenate | input: [(None, 3360), (None, 1820), (None, 960)] | output: (None, 6140)
Dense1 | Dense | input: (None, 6140) | output: (None, 900)
Drop1 | Dropout | input: (None, 900) | output: (None, 900)
Outcome | InputLayer | input: [(None, 8)] | output: [(None, 8)]
delta_GB | InputLayer | input: [(None, 1)] | output: [(None, 1)]
Cas9 | InputLayer | input: [(None, 1)] | output: [(None, 1)]
CON2 | Concatenate | input: [(None, 900), (None, 8), (None, 1), (None, 1)] | output: (None, 910)
Dense2 | Dense | input: (None, 910) | output: (None, 900)
Drop2 | Dropout | input: (None, 900) | output: (None, 900)
Dense3 | Dense | input: (None, 900) | output: (None, 200)
Drop3 | Dropout | input: (None, 200) | output: (None, 200)
Output | Dense | input: (None, 200) | output: (None, 2)

(B) CBE

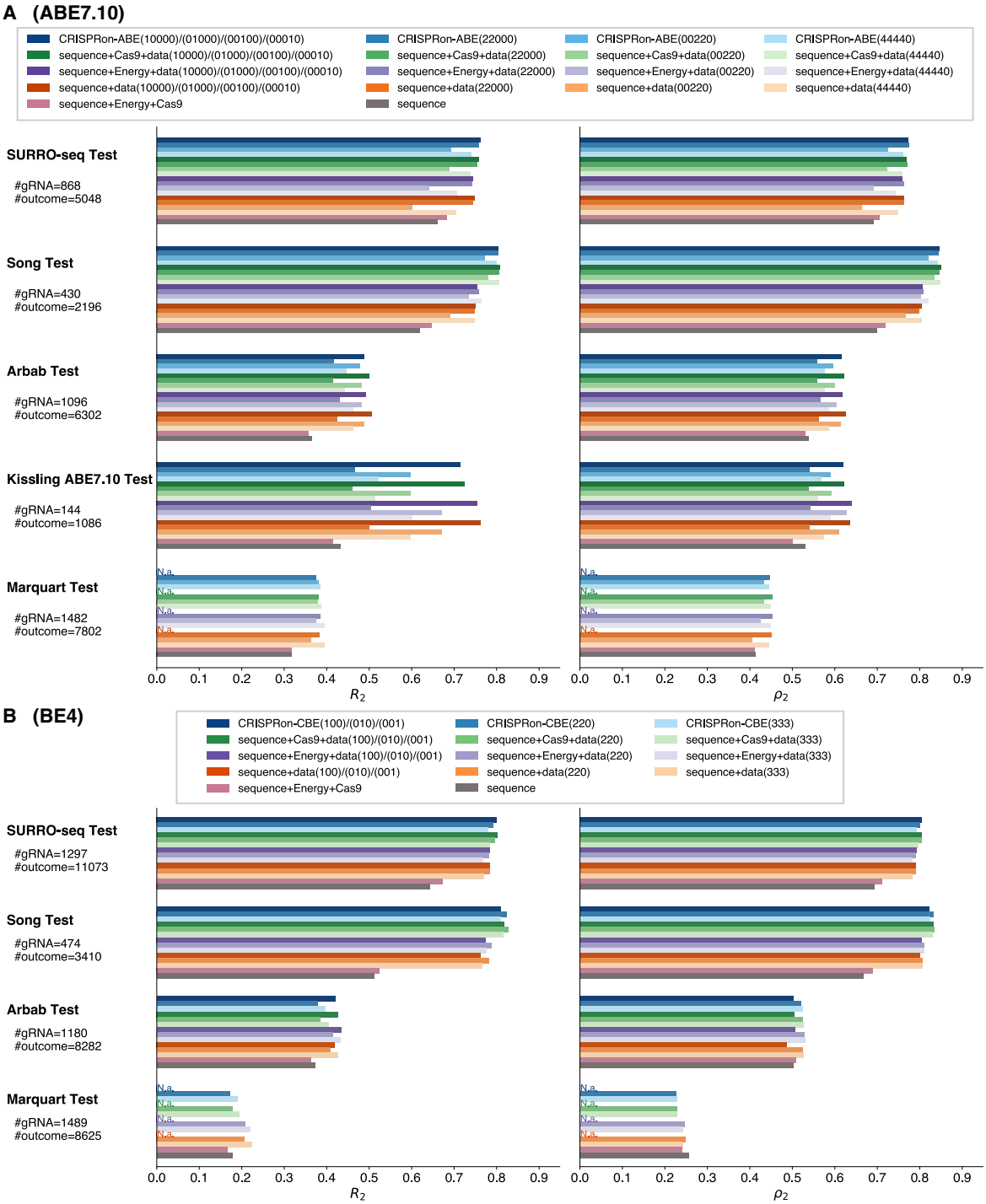

DNA_target | InputLayer | input: [(None, 30, 4)] | output: [(None, 30, 4)]
CNN1 | Conv1D | input: (None, 30, 4) | output: (None, 28, 360)
CNN2 | Conv1D | input: (None, 30, 4) | output: (None, 26, 210)
CNN3 | Conv1D | input: (None, 30, 4) | output: (None, 24, 120)
CNN_Drop1 | SpatialDropout1D | input: (None, 28, 360) | output: (None, 28, 360)
CNN_Drop2 | SpatialDropout1D | input: (None, 26, 210) | output: (None, 26, 210)
CNN_Drop3 | SpatialDropout1D | input: (None, 24, 120) | output: (None, 24, 120)
pool1 | AveragePooling1D | input: (None, 28, 360) | output: (None, 14, 360)
pool2 | AveragePooling1D | input: (None, 26, 210) | output: (None, 13, 210)
pool3 | AveragePooling1D | input: (None, 24, 120) | output: (None, 12, 120)
Flat1 | Flatten | input: (None, 14, 360) | output: (None, 5040)
Flat2 | Flatten | input: (None, 13, 210) | output: (None, 2730)
Flat3 | Flatten | input: (None, 12, 120) | output: (None, 1440)
CON1 | Concatenate | input: [(None, 5040), (None, 2730), (None, 1440)] | output: (None, 9210)
Dense1 | Dense | input: (None, 9210) | output: (None, 1000)
Drop1 | Dropout | input: (None, 1000) | output: (None, 1000)
Outcome | InputLayer | input: [(None, 8)] | output: [(None, 8)]
delta_GB | InputLayer | input: [(None, 1)] | output: [(None, 1)]
Cas9 | InputLayer | input: [(None, 1)] | output: [(None, 1)]
CON2 | Concatenate | input: [(None, 1000), (None, 8), (None, 1), (None, 1)] | output: (None, 1010)
Dense2 | Dense | input: (None, 1010) | output: (None, 1000)
Drop2 | Dropout | input: (None, 1000) | output: (None, 1000)
Dense3 | Dense | input: (None, 1000) | output: (None, 100)
Drop3 | Dropout | input: (None, 100) | output: (None, 100)
Output | Dense | input: (None, 100) | output: (None, 2)

Supplementary Fig. 2: Model architecture for CRISPRon-ABE (A) and CRISPRon-CBE (B) without dataset source flagging in the input.
The figure was generated by TensorFlow[18]. The model requires four inputs: (1) one-hot encoded 30nt DNA sequence (named "DNA_target", with a size of [None, 30, 4]); (2) one-hot encoded outcome sequence (named "Outcome", with a size of [None, 8]); (3) gRNA-DNA binding energy ΔG_B (named "delta_GB", with a size of [None, 1]); (4) predicted Cas9 indel frequency (named "Cas9", with a size of [None, 1]), where "None" represents the variable number of samples for training or evaluation.


A (ABE)
Molecular features
CRISPRon prediction
MT[3,10]
ΔG_B
MFE_spacer
MFE_extension
GC_protospacer
SHAP value
High
Feature value
Low
mean(|SHAP value|)
Nucleotide compositional features
A_5
A_6
AC_6
C_13
AC_7
AC_5
G_-1
G_10
TA
AC
B (CBE)
C_6
TC_3
G_5
TC_4
TC_6
C_7
G_6
C_5
TC_2
G_4

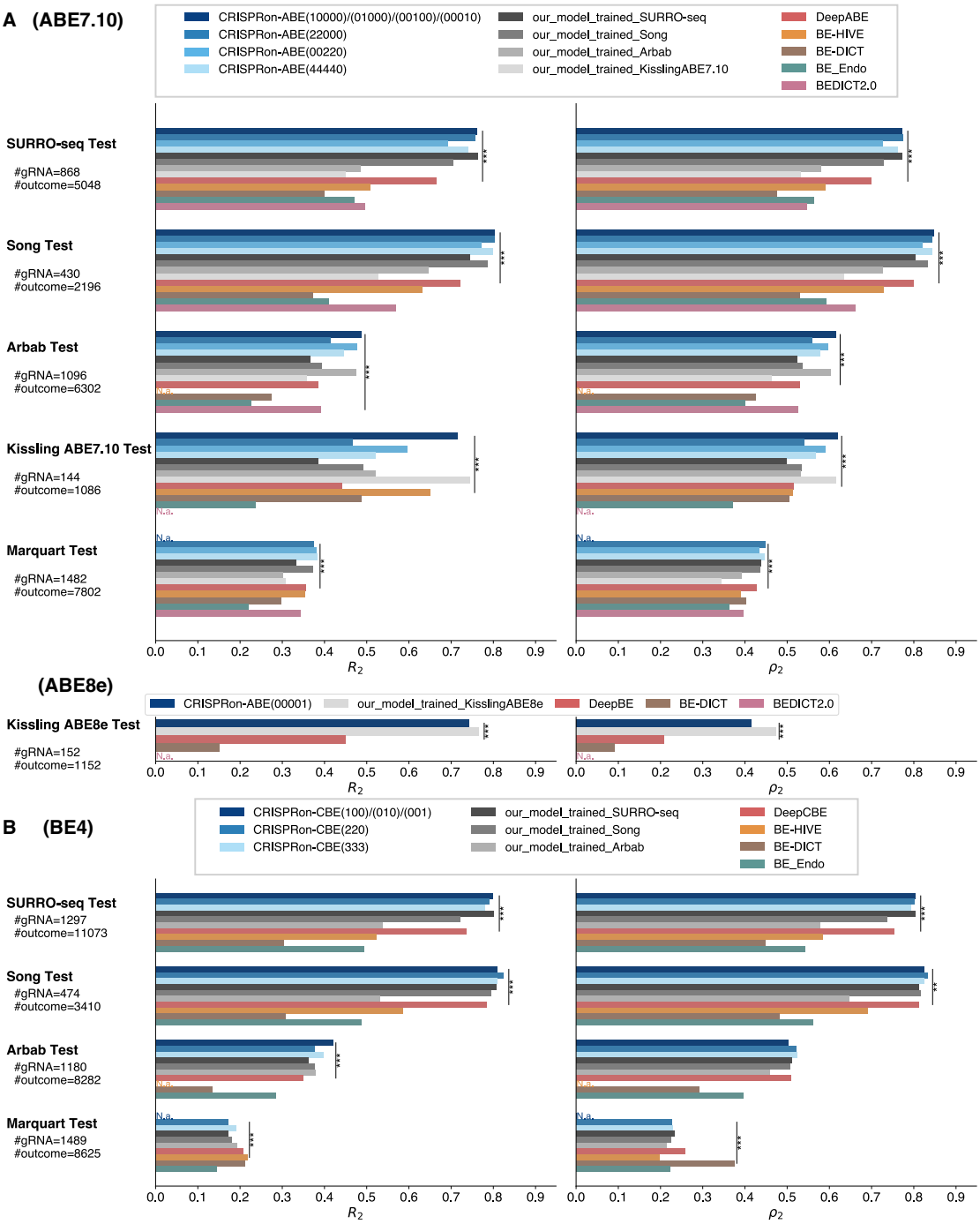
Supplementary Fig. 3: Important features associated with base editor efficiency prediction using SURRO-seq datasets[17].

The 30nt target DNA sequence, thermodynamic features and predicted Cas9 activity by CRISPRon were used for important feature extraction by calculating SHapley Additive exPlanations "SHAP" values on XGBoost model. A high SHAP value indicates a high contribution to the prediction output. In the violin plot (left), each point represents a 30nt target DNA sequence, whose color reflects the value of the specific feature (red for high and blue for low). All the molecular features and the top 10 important nucleotide features are presented for both ABE (A) and CBE (B). Positions on the 30nt DNA input are labeled as follows: left context: -4 to -1; gRNA spacer: 1 to 20, PAM: N, G1, G2; right context: +1 to +3. The spacer interval used to compute melting temperatures (MT) is indicated in square brackets. The Minimum Free Energies (MFE) from spacer and spacer + scaffold are labeled as "MFE_spacer" and "MFE_extension", respectively. (A) ABE; (B) CBE.



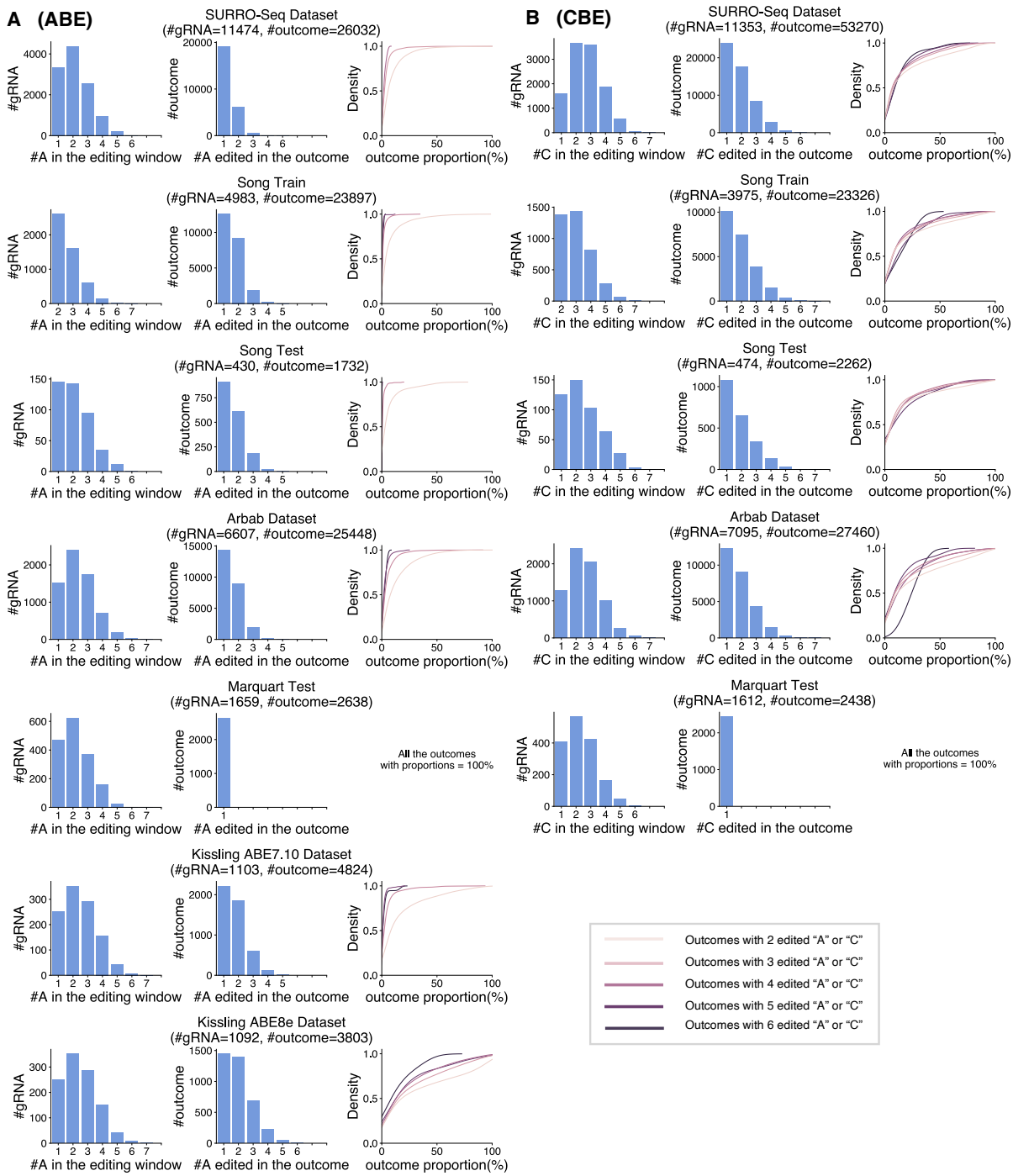
Supplementary Fig. 4: Ablation analysis for features involved in CRISPRon-ABE/CBE model.

Three features are involved in the ablation analysis, including: gRNA-DNA binding energy ΔG_B (named "Energy"), predicted Cas9 indel frequency from CRISPRon (named "Cas9") and dataset source (named "data"). The sequences including 30nt target DNA sequence and labeling positions with edited nucleotides in the edited outcomes are always kept in the model (named "sequence"). The models are further named accordingly based on the features involved in the model. For example, the model "sequence+Energy+data(220)" was trained using 30nt target sequence, edited nucleotide positions from the outcome products and binding energy ΔG_B. The numbers show the weights tuning for the test set prediction. For example, "220" means the equal weight to SURRO-seq and Song datasets. (A) ABE; (B) CBE.



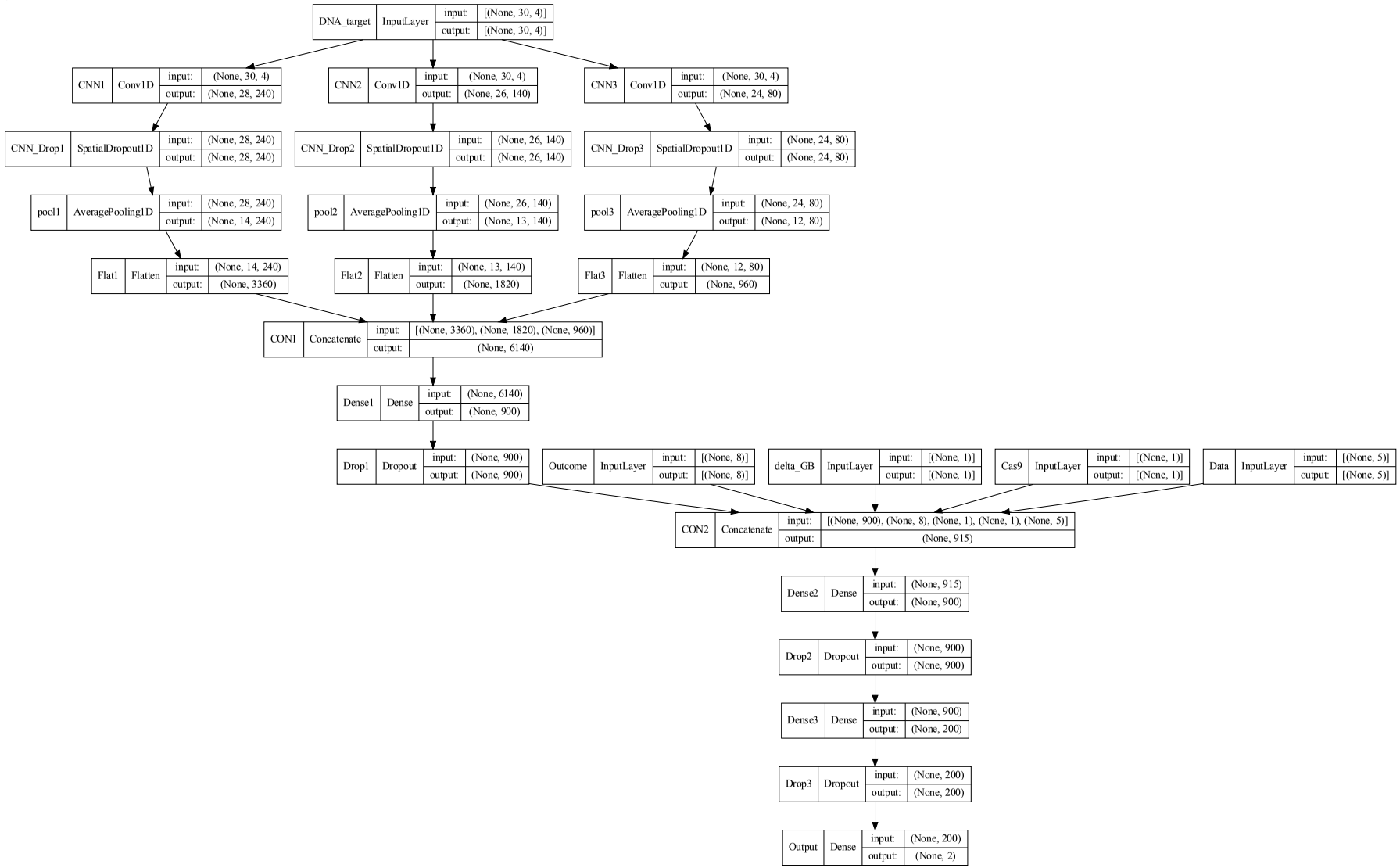
Supplementary Fig 5: Performance comparison of CRISPRon-ABE and CRISPRon-CBE with other existing models on independent test sets.
Performance comparison of several models for (A) ABE and (B) CBE efficiency and outcome predictions. The CRISPRon-ABE/CBE models were evaluated with different weights to the training data from SURRO-seq, Song, Arbab, Kissling ABE7.10 (ABE only) and Kissling ABE8e (ABE only) datasets, where "1" means a weight of 100%, "2" means a weight of 50%, "3" a weight of 33%, "4" a weight of 25% and "0" a weight of zero. Hence, "220" means 50% to SURRO-seq and Song datasets and zero weight to the Arbab dataset for CBE. The models named "our_model_trained_SURRO-seq", "our_model_trained_Song" and "our_model_trained_Arbab", "our_model_trained_KisslingABE7.10" and "our_model_trained_KisslingABE8e" are models trained only on the respective datasets using our architecture. DeepABE/CBE, BE-HIVE, BE-DICT, BE_Endo, BEDICT2.0 and DeepBE

refer to the published models. For benchmarking on the respective test sets, the gRNAs in all the test sets overlap to the data points in any training data were removed (see Methods "Test sets for the evaluation and comparison of models"). N.a. means not available due to lack of explicit train-test separation or no model available for the lack of suitable training sets. The two-sided Steiger's test *p*-values were used for comparing correlation coefficients between two models. * represents *p*-value < 0.05, ** represents *p*-value < 0.01, *** represents *p*-value < 0.001.

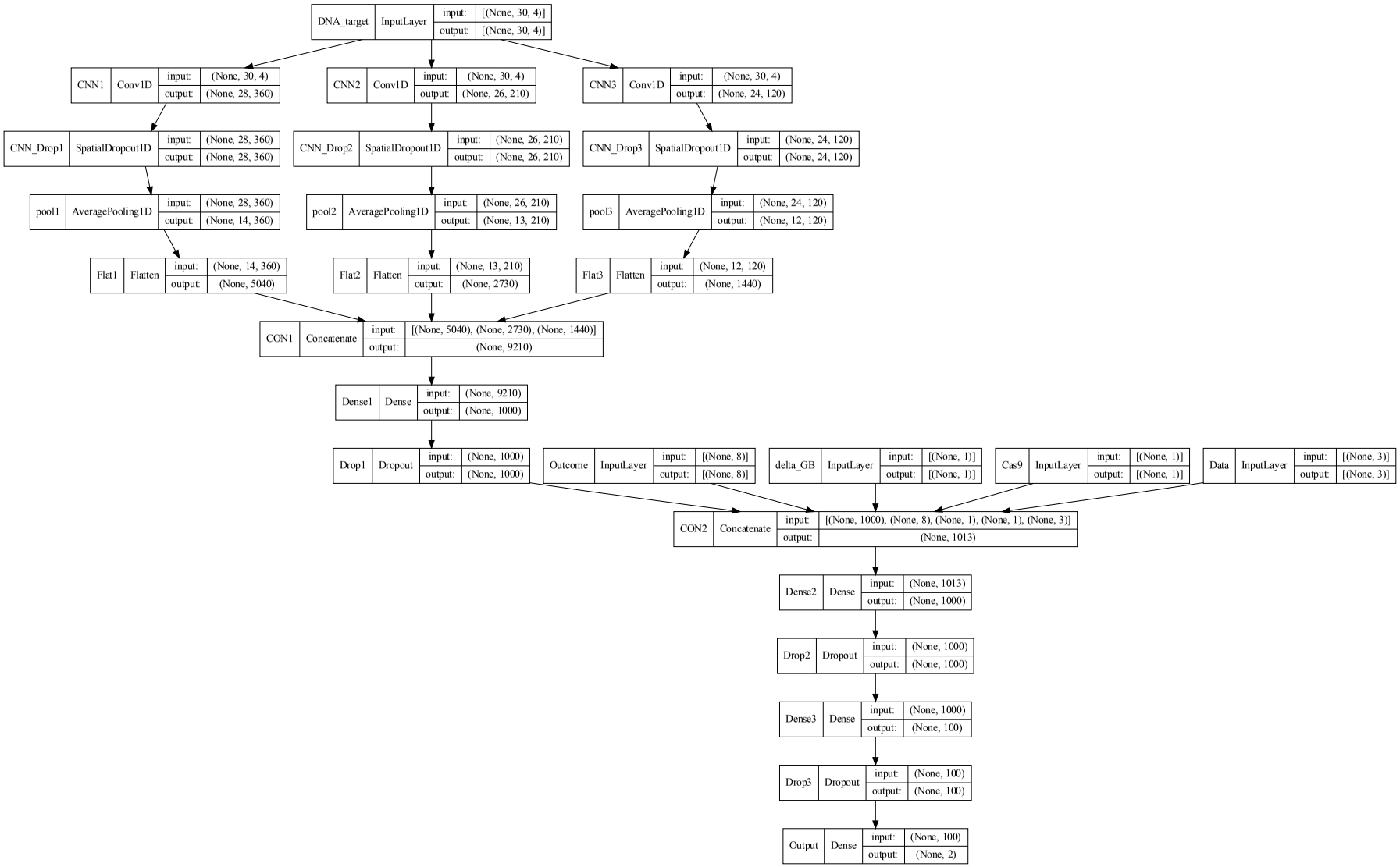


Supplementary Fig 6: Variety of dataset from diverse sources.
Distribution of the number of edited nucleotides and outcome proportions from outcomes with experimental reads supported for ABE (A) and CBE (B). Source data are provided as a Source Data file.

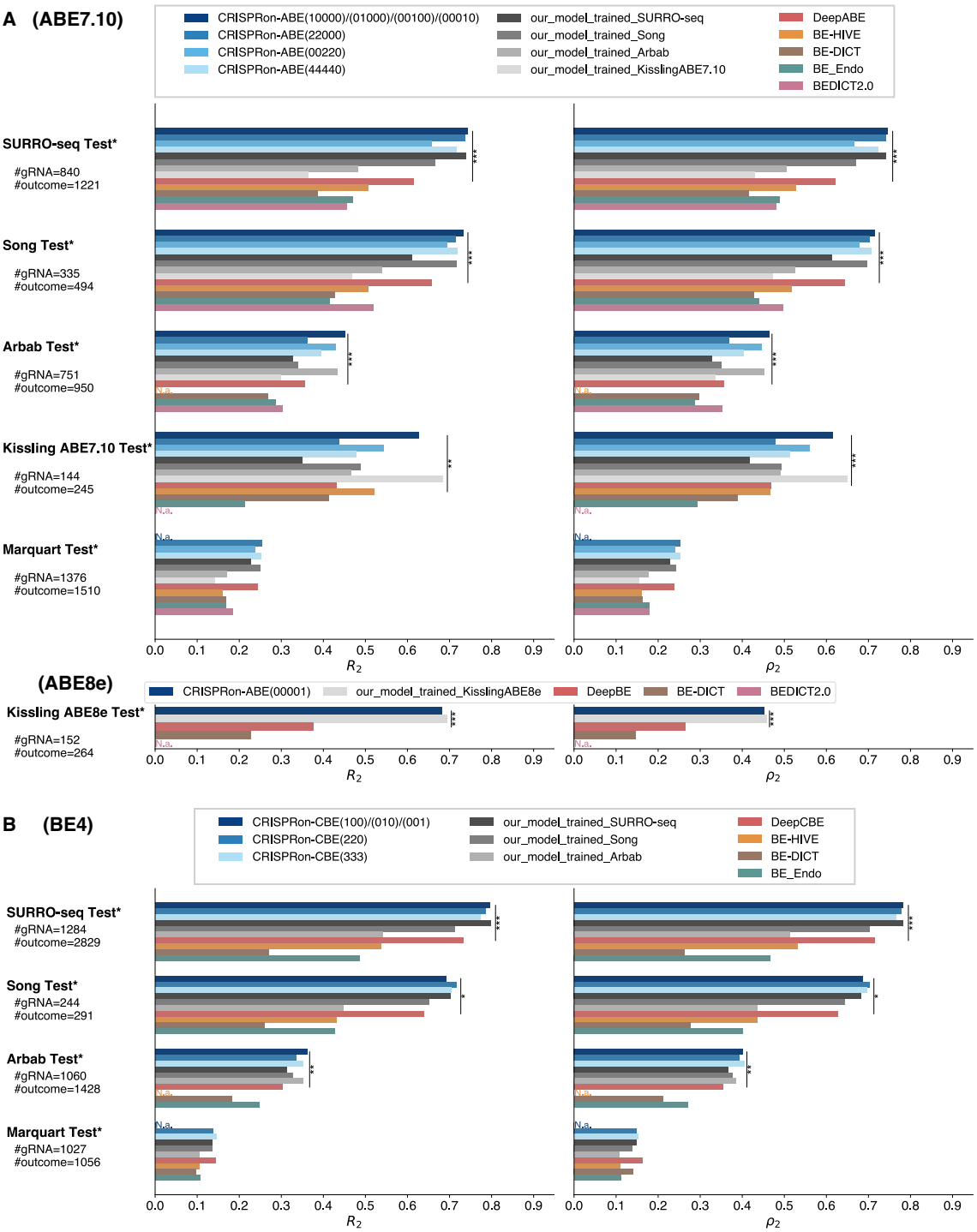
(A) ABE


DNA_target | InputLayer | input: [(None, 30, 4)] | output: [(None, 30, 4)]
CNN1 | Conv1D | input: (None, 30, 4) | output: (None, 28, 240)
CNN2 | Conv1D | input: (None, 30, 4) | output: (None, 26, 140)
CNN3 | Conv1D | input: (None, 30, 4) | output: (None, 24, 80)
CNN_Drop1 | SpatialDropout1D | input: (None, 28, 240) | output: (None, 28, 240)
CNN_Drop2 | SpatialDropout1D | input: (None, 26, 140) | output: (None, 26, 140)
CNN_Drop3 | SpatialDropout1D | input: (None, 24, 80) | output: (None, 24, 80)
pool1 | AveragePooling1D | input: (None, 28, 240) | output: (None, 14, 240)
pool2 | AveragePooling1D | input: (None, 26, 140) | output: (None, 13, 140)
pool3 | AveragePooling1D | input: (None, 24, 80) | output: (None, 12, 80)
Flat1 | Flatten | input: (None, 14, 240) | output: (None, 3360)
Flat2 | Flatten | input: (None, 13, 140) | output: (None, 1820)
Flat3 | Flatten | input: (None, 12, 80) | output: (None, 960)
CON1 | Concatenate | input: [(None, 3360), (None, 1820), (None, 960)] | output: (None, 6140)
Dense1 | Dense | input: (None, 6140) | output: (None, 900)
Drop1 | Dropout | input: (None, 900) | output: (None, 900)
Outcome | InputLayer | input: [(None, 8)] | output: [(None, 8)]
delta_GB | InputLayer | input: [(None, 1)] | output: [(None, 1)]
Cas9 | InputLayer | input: [(None, 1)] | output: [(None, 1)]
Data | InputLayer | input: [(None, 5)] | output: [(None, 5)]
CON2 | Concatenate | input: [(None, 900), (None, 8), (None, 1), (None, 1), (None, 5)] | output: (None, 915)
Dense2 | Dense | input: (None, 915) | output: (None, 900)
Drop2 | Dropout | input: (None, 900) | output: (None, 900)
Dense3 | Dense | input: (None, 900) | output: (None, 200)
Drop3 | Dropout | input: (None, 200) | output: (None, 200)
Output | Dense | input: (None, 200) | output: (None, 2)

(B) CBE

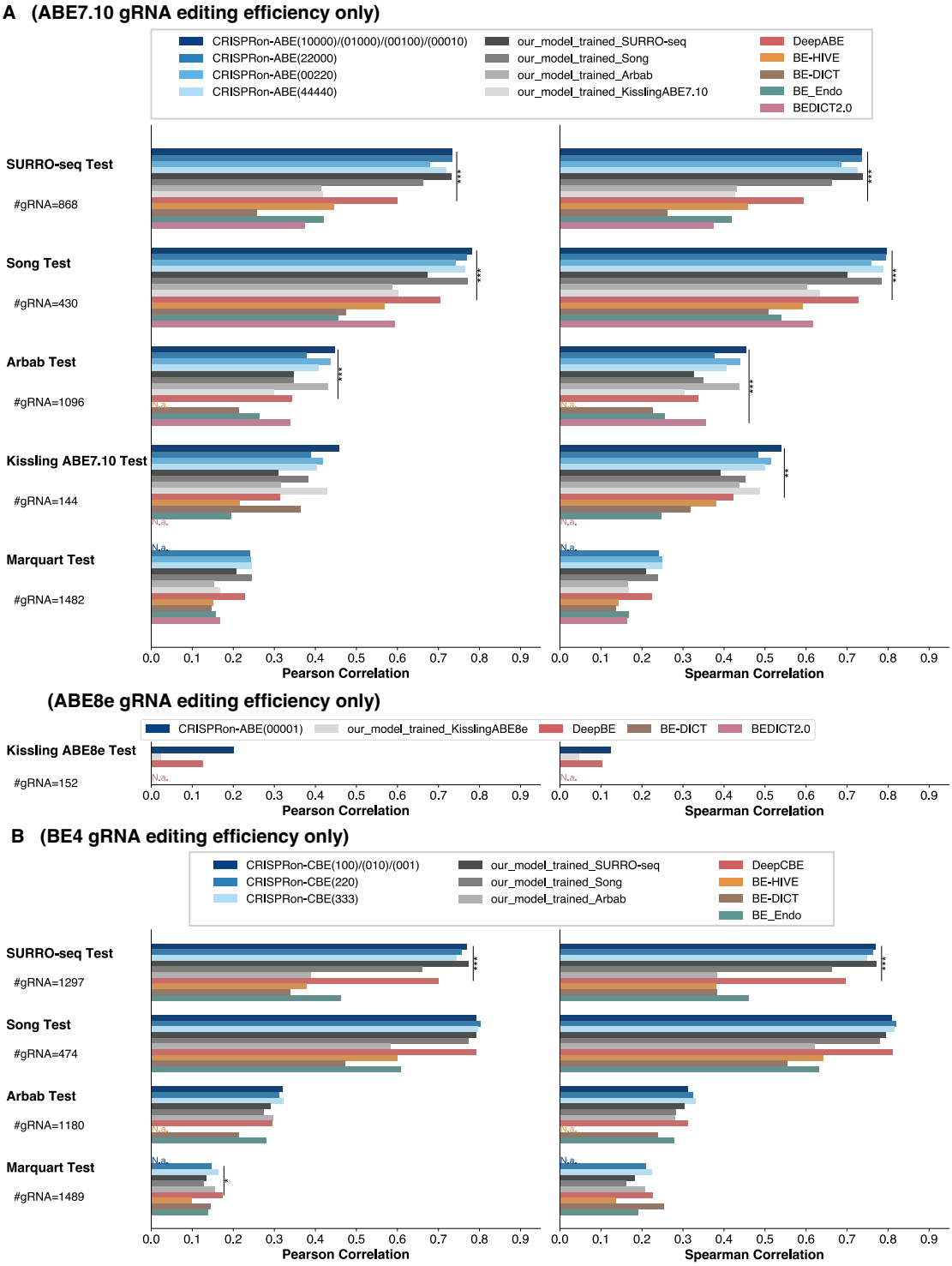

DNA_target | InputLayer | input: [(None, 30, 4)] | output: [(None, 30, 4)]
CNN1 | Conv1D | input: (None, 30, 4) | output: (None, 28, 360)
CNN2 | Conv1D | input: (None, 30, 4) | output: (None, 26, 210)
CNN3 | Conv1D | input: (None, 30, 4) | output: (None, 24, 120)
CNN_Drop1 | SpatialDropout1D | input: (None, 28, 360) | output: (None, 28, 360)
CNN_Drop2 | SpatialDropout1D | input: (None, 26, 210) | output: (None, 26, 210)
CNN_Drop3 | SpatialDropout1D | input: (None, 24, 120) | output: (None, 24, 120)
pool1 | AveragePooling1D | input: (None, 28, 360) | output: (None, 14, 360)
pool2 | AveragePooling1D | input: (None, 26, 210) | output: (None, 13, 210)
pool3 | AveragePooling1D | input: (None, 24, 120) | output: (None, 12, 120)
Flat1 | Flatten | input: (None, 14, 360) | output: (None, 5040)
Flat2 | Flatten | input: (None, 13, 210) | output: (None, 2730)
Flat3 | Flatten | input: (None, 12, 120) | output: (None, 1440)
CON1 | Concatenate | input: [(None, 5040), (None, 2730), (None, 1440)] | output: (None, 9210)
Dense1 | Dense | input: (None, 9210) | output: (None, 1000)
Drop1 | Dropout | input: (None, 1000) | output: (None, 1000)
Outcome | InputLayer | input: [(None, 8)] | output: [(None, 8)]
delta_GB | InputLayer | input: [(None, 1)] | output: [(None, 1)]
Cas9 | InputLayer | input: [(None, 1)] | output: [(None, 1)]
Data | InputLayer | input: [(None, 3)] | output: [(None, 3)]
CON2 | Concatenate | input: [(None, 1000), (None, 8), (None, 1), (None, 1), (None, 3)] | output: (None, 1013)
Dense2 | Dense | input: (None, 1013) | output: (None, 1000)
Drop2 | Dropout | input: (None, 1000) | output: (None, 1000)
Dense3 | Dense | input: (None, 1000) | output: (None, 100)
Drop3 | Dropout | input: (None, 100) | output: (None, 100)
Output | Dense | input: (None, 100) | output: (None, 2)

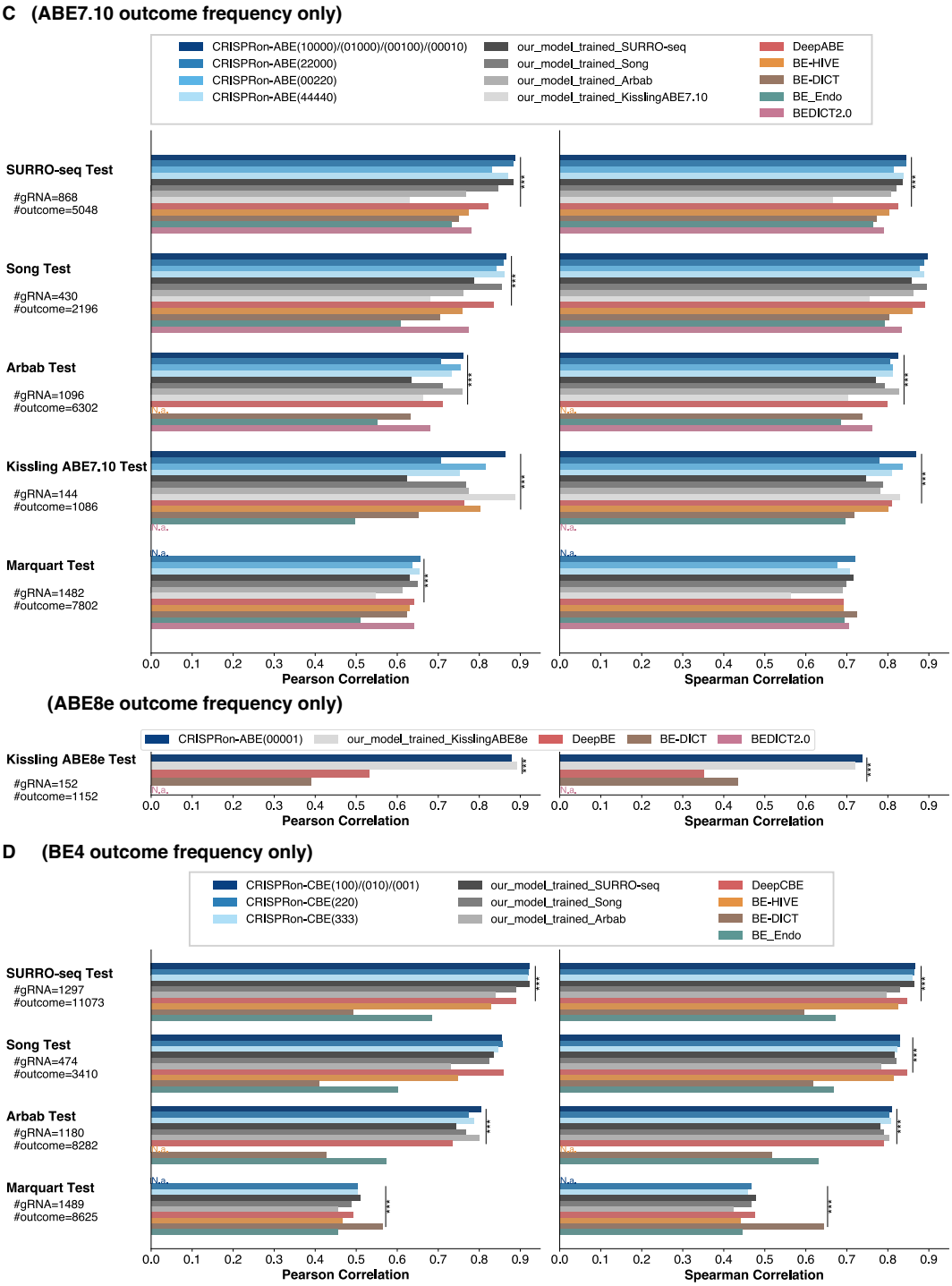
Supplementary Fig. 7: Model architecture for CRISPRon-ABE (A) and CRISPRon-CBE (B) with dataset features.
The figure was generated by TensorFlow[18]. The model requires five inputs: (1) one-hot encoded 30nt DNA sequence (named “DNA_target”, with a size of [None, 30, 4]); (2) one-hot encoded outcome sequence (named “Outcome”, with a size of [None, 8]); (3) gRNA-DNA binding energy ΔG_B (named “delta_GB”, with a size of [None, 1]); (4) predicted Cas9 indel frequency (named “Cas9”, with a size of [None, 1]), (5) dataset weight (named “Data”, with a size of [None, 5] for ABE or [None, 3] for CBE), where "None” represents the variable number of samples for training or evaluation.



Supplementary Fig. 8: Performance comparison of CRISPRon-ABE and CRISPRon-CBE with other existing models on independent test sets when outcome frequency is higher than 5%.

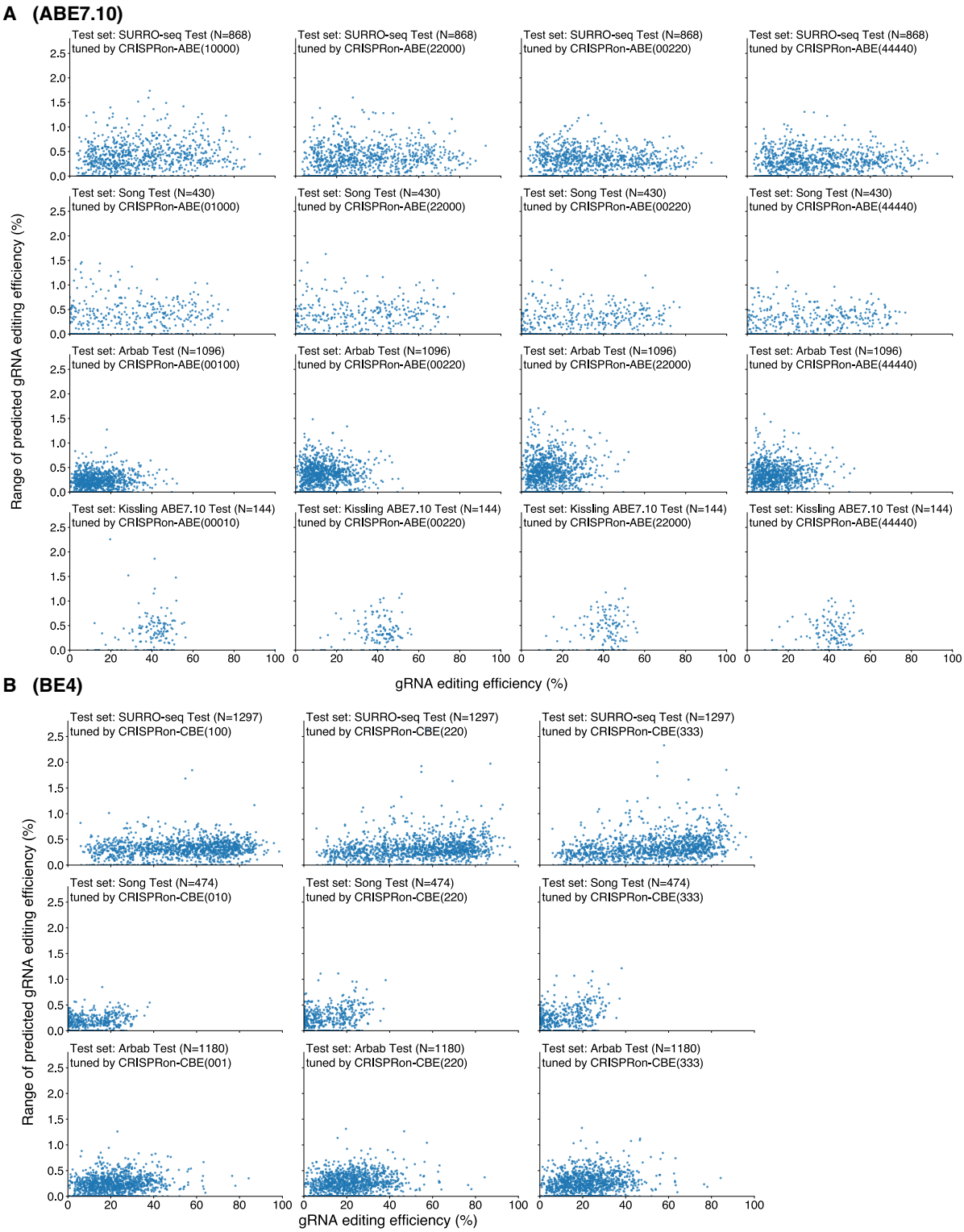
The same methods as in Supplementary Fig. 5 were evaluated on test sets without low efficient ones (See "Test sets for the evaluation and comparison of models for details"). Considering that the outcomes with high frequency are more critical in application, the subset of outcomes with frequency greater than 5% was selected for benchmark analysis, which are named as "SURRO-seq Test*", "Song Test*", "Arbab Test*", "Marquart Test*", "Kissling ABE7.10 Test*" and "Kissling ABE8e Test*", respectively. (A) evaluation on ABE data. (B) evaluation on CBE data.


A (ABE7.10 gRNA editing efficiency only)
CRISPRon-ABE(10000)/(01000)/(00100)/(00010)
CRISPRon-ABE(22000)
CRISPRon-ABE(00220)
CRISPRon-ABE(44440)
our_model_trained_SURRO-seq
our_model_trained_Song
our_model_trained_Arbab
our_model_trained_KisslingABE7.10
DeepABE
BE-HIVE
BE-DICT
BE_Endo
BEDICT2.0
SURRO-seq Test
#gRNA=868
Song Test
#gRNA=430
Arbab Test
#gRNA=1096
Kissling ABE7.10 Test
#gRNA=144
Marquart Test
#gRNA=1482
N.a.
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Pearson Correlation
Spearman Correlation
(ABE8e gRNA editing efficiency only)
CRISPRon-ABE(00001)
our_model_trained_KisslingABE8e
DeepBE
BE-DICT
BEDICT2.0
Kissling ABE8e Test
#gRNA=152
B (BE4 gRNA editing efficiency only)
CRISPRon-CBE(100)/(010)/(001)
CRISPRon-CBE(220)
CRISPRon-CBE(333)
our_model_trained_SURRO-seq
our_model_trained_Song
our_model_trained_Arbab
DeepCBE
BE-HIVE
BE-DICT
BE_Endo
SURRO-seq Test
#gRNA=1297
Song Test
#gRNA=474
Arbab Test
#gRNA=1180
Marquart Test
#gRNA=1489

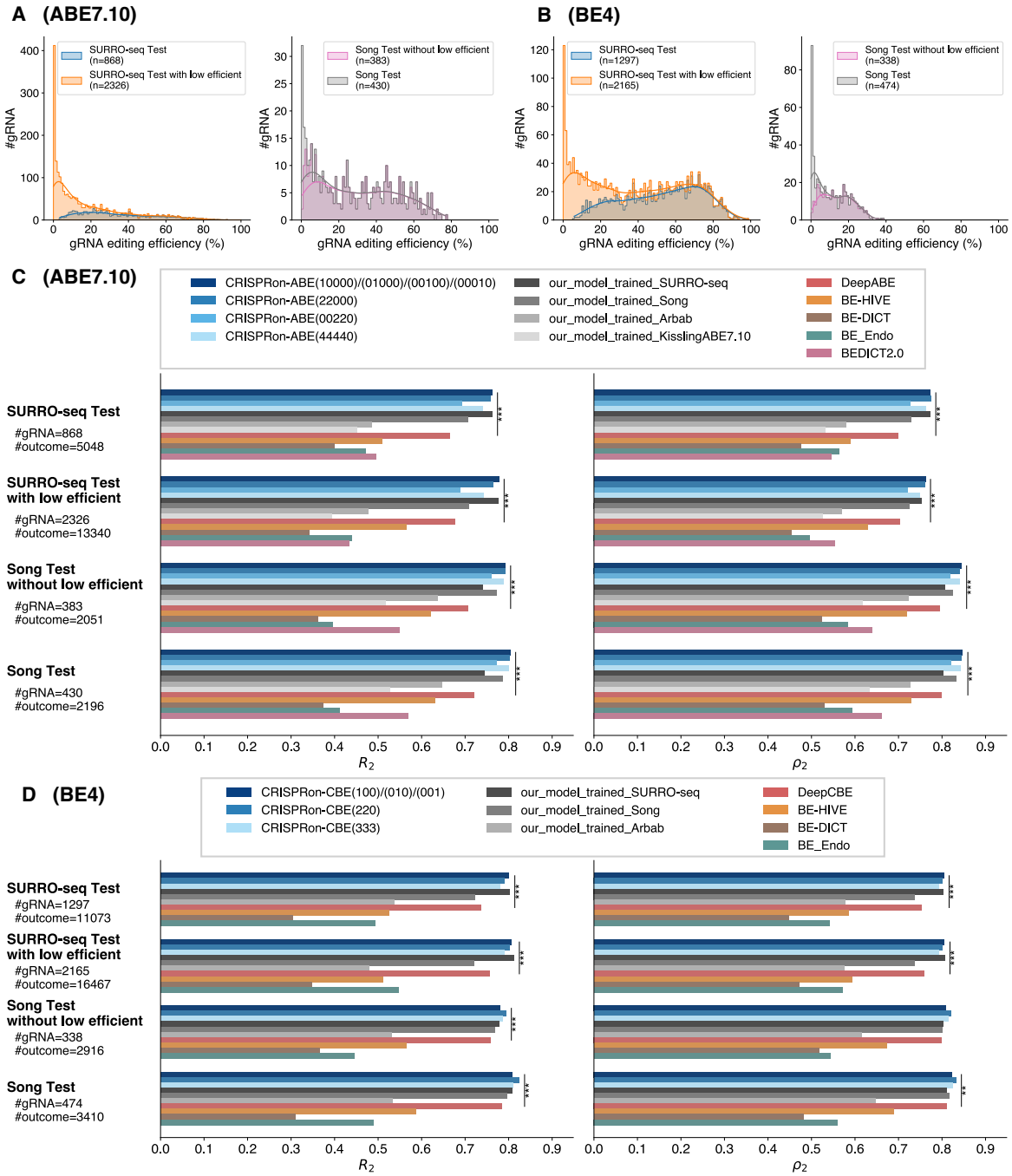


Supplementary Fig. 9: Performance comparison of CRISPRon-ABE and CRISPRon-CBE with other existing models on independent test sets.

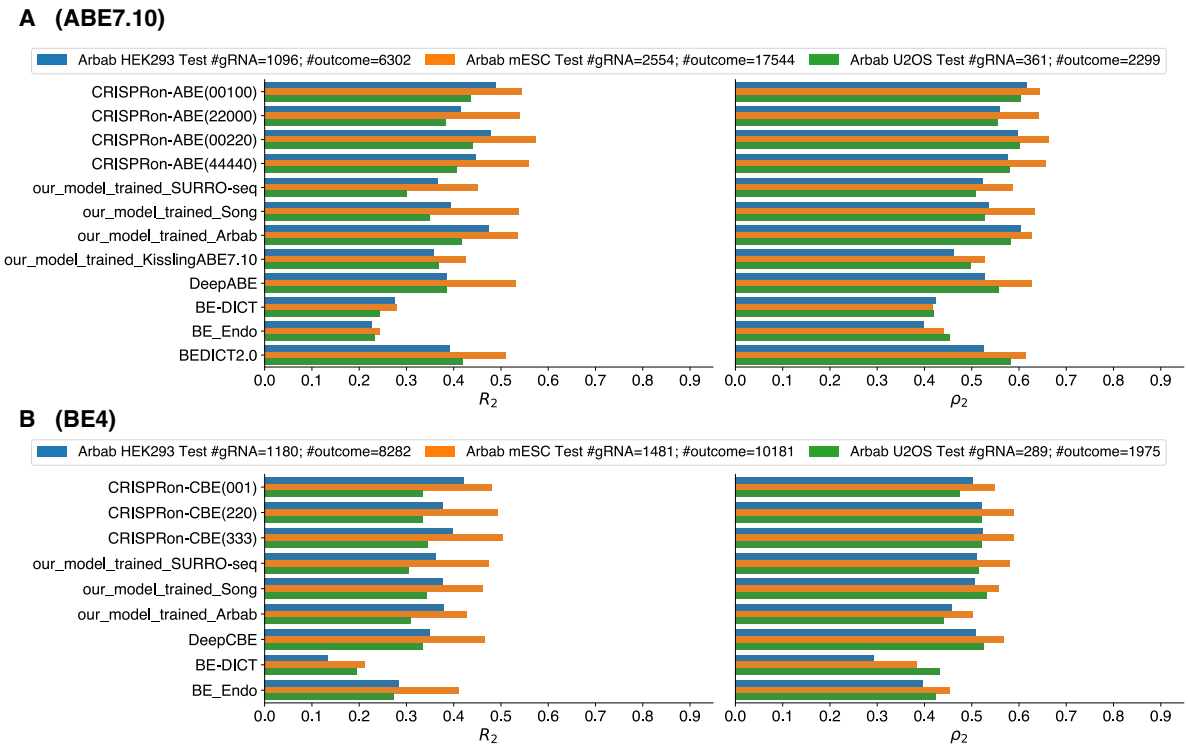
A. and B. performance comparison of CRISPRon-ABE/CBE by evaluating gRNA editing efficiency (A) ABE and CBE (B), respectively; C. and D. performance comparison of CRISPRon-ABE/CBE by evaluating outcome frequency (C) ABE (D) CBE, respectively; N.a. means not available due to lack of explicit train-test separation or no model available for the lack of suitable training sets. The two-sided Steiger's test *p*-values were used for comparing correlation coefficients between the two models. * represents *p*-value < 0.05, ** represents *p*-value < 0.01, *** represents *p*-value < 0.001.



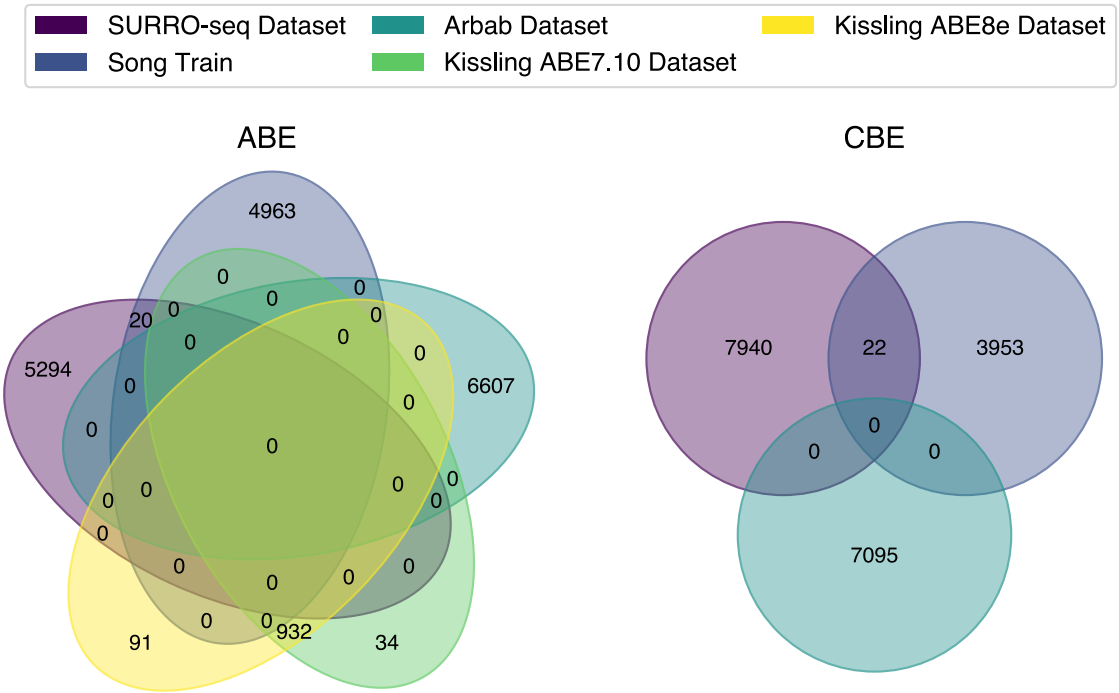
Supplementary Fig. 10: Range of predicted gRNA editing efficiency for individual gRNA. (A) CRISPRon-ABE; (B) CRISPRon-CBE. N represents the number of gRNAs. Source data are provided as a Source Data file.



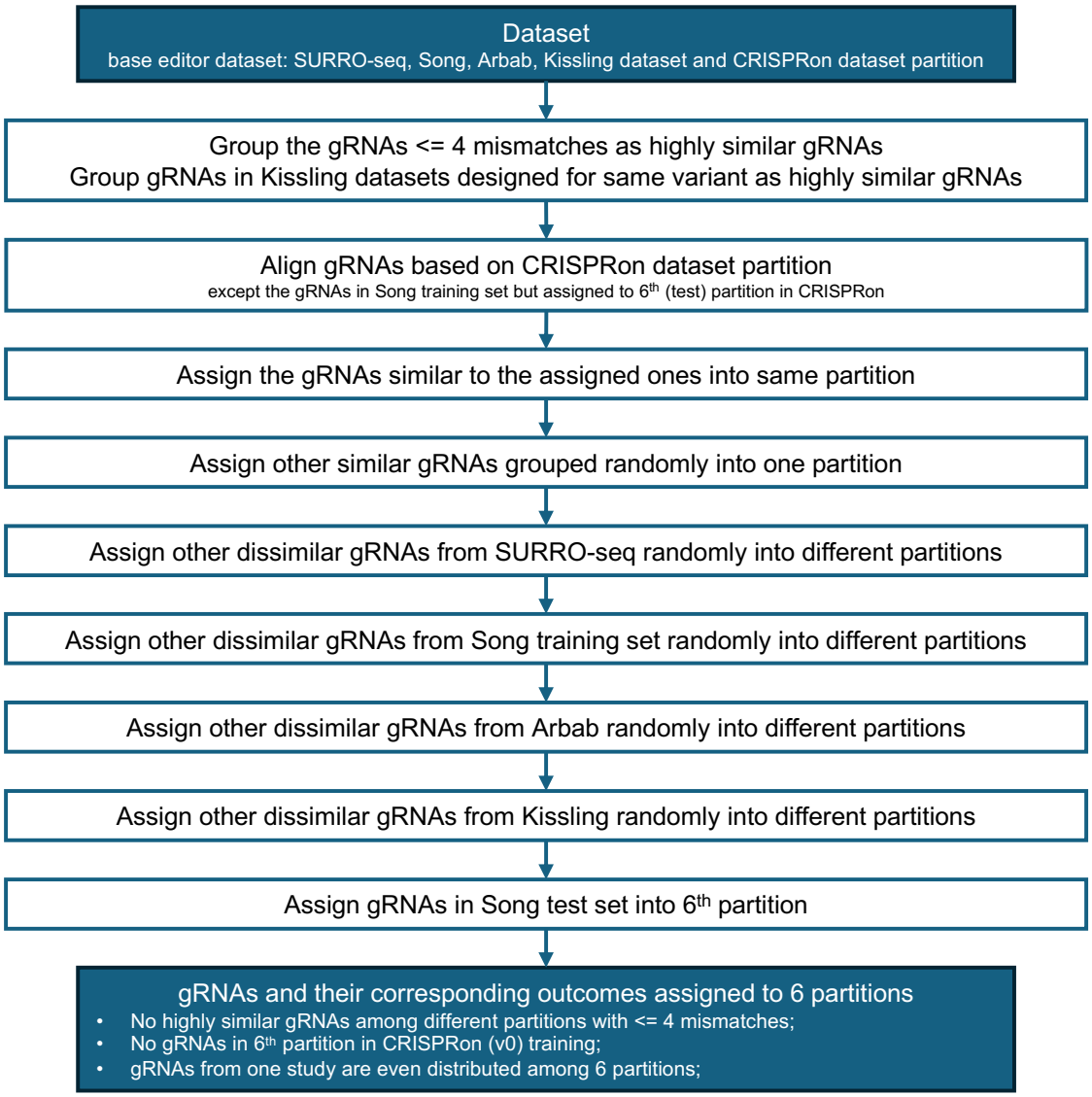
Supplementary Fig 11: Performance comparison of CRISPRon-ABE and CRISPRon-CBE with other existing models on independent test sets with/without low efficient gRNAs. The performance of CRISPRon-ABE and CRISPRon-CBE on low efficient gRNAs (< 100 edited reads, not used for training the model) are calculated for a systematical evaluation. Four independent test sets are used for comparison: SURRO-seq Test (6^{th} partition, ⩾100 edited reads), SURRO-seq Test with low efficient (6^{th} partition + gRNAs < 100 edited reads), Song Test (all test set), Song Test without low efficient (test set removing gRNAs <100 edited reads). Distribution of gRNA editing efficiency from test sets for ABE (A) and CBE (B); Performance comparison of CRISPRon-ABE (C) and CRISPRon-CBE (D) together with other existing models on independent test sets with/without low efficient gRNAs. Source data are provided as a Source Data file.



Supplementary Fig 12: Performance comparison of CRISPRon-ABE and CRISPRon-CBE with other existing models on independent Arbab test sets across cell-lines.
Datasets from different cell-lines are collected for benchmarking, including HEK293, mESC and U2OS cell-lines. The test sets from HEK293 cell-lines have been used for benchmarking analysis (named “Arbab Test”). (A) ABE; (B) CBE.



Supplementary Fig 13: Identical 30nt target DNA sequences across different datasets. The datasets containing the same 30nt target DNA sequence among different datasets are counted.



Supplementary Fig 14: Workflow for creating independent dataset for CRISPRon-ABE/CBE. These steps prevent highly similar data points between training and test sets, reducing the risk of overfitting in model evaluation.

References

1. Song M*, et al.* Sequence-specific prediction of the efficiencies of adenine and cytosine base editors. *Nat Biotechnol* **38**, 1037-1043 (2020).

2. Arbab M*, et al.* Determinants of Base Editing Outcomes from Target Library Analysis and Machine Learning. *Cell* **182**, 463-480 e430 (2020).

3. Marquart KF*, et al.* Predicting base editing outcomes with an attention-based deep learning algorithm trained on high-throughput target library screens. *Nat Commun* **12**, 5114 (2021).

4. Yuan T*, et al.* Deep learning models incorporating endogenous factors beyond DNA sequences improve the prediction accuracy of base editing outcomes. *Cell Discov* **10**, 20 (2024).

5. Kissling L*, et al.* Predicting adenine base editing efficiencies in different cellular contexts by deep learning. *Genome Biol* **26**, 115 (2025).

6. Kim N*, et al.* Deep learning models to predict the editing efficiencies and outcomes of diverse base editors. *Nat Biotechnol*, (2023).

7. Xiang X*, et al.* Enhancing CRISPR-Cas9 gRNA efficiency prediction by data integration and deep learning. *Nat Commun* **12**, 3238 (2021).

8. Lundberg SM, Lee SI. A Unified Approach to Interpreting Model Predictions. *Adv Neur In* **30**, (2017).

9. Chen TQ, Guestrin C. XGBoost: A Scalable Tree Boosting System. *Kdd'16: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785-794 (2016).

10. Lorenz R*, et al.* ViennaRNA Package 2.0. *Algorithms Mol Biol* **6**, 26 (2011).

11. Alkan F, Wenzel A, Anthon C, Havgaard JH, Gorodkin J. CRISPR-Cas9 off-targeting assessment with nucleic acid duplex energy parameters. *Genome Biol* **19**, 177 (2018).

12. Cock PJA*, et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422-1423 (2009).

13. Breiman L. Random forests. *Mach Learn* **45**, 5-32 (2001).

14. Gorodkin J. Comparing two K-category assignments by a K-category correlation coefficient. *Comput Biol Chem* **28**, 367-374 (2004).

15. Bush K*, et al.* Utilizing directed evolution to interrogate and optimize CRISPR/Cas guide RNA scaffolds. *Cell Chem Biol* **30**, 879-892 e875 (2023).

16. Komor AC*, et al.* Improved base excision repair inhibition and bacteriophage Mu Gam protein yields C:G-to-T:A base editors with higher efficiency and product purity. *Sci Adv* **3**, eaao4774 (2017).

17. Pan X*, et al.* Massively targeted evaluation of therapeutic CRISPR off-targets in cells. *Nat Commun* **13**, 4049 (2022).

18. Abadi M*, et al.* TensorFlow: A system for large-scale machine learning. *Proceedings of Osdi'16: 12th Usenix Symposium on Operating Systems Design and Implementation*, 265-283 (2016).