Supplementary material for "A tool for CRISPR-Cas9 gRNA evaluation based on computational models of gene expression"

Supplementary methods

1. Xpresso

Xpresso (1) was trained and tested on cap analysis gene expression (CAGE) data of all genes (~18k). The network consists of ~112k parameters and its architecture is made up of two sequential convolutional and max-pooling layers, after which come two fully connected layers before connecting to the output neuron (Fig. S1). Xpresso was had an r^2 of 0.59 when comparing predicted and measured mRNA levels in Human.

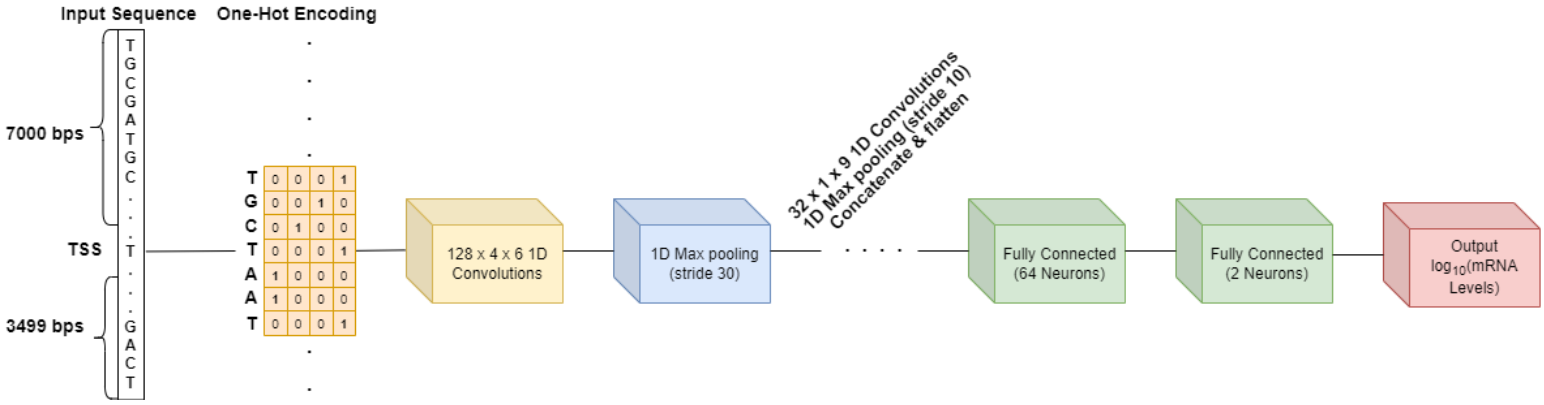

Fig S1: Schematic of Xpresso. *The neural network used to predict mRNA levels from a sequence surrounding the Transcription Start Site (TSS).*

2. SpliceAI

The input to SpliceAI (2) is a sequence of at least 10,001nt where each position of interest has at least 10,000nt of context (5000 from each side), and it will output a sequence of acceptor and donor probabilities for all positions of interest i.e. any nucleotide in the middle of the sequence that has 5000nt of context from each side. The network was trained on GENCODE-annotated pre-mRNA transcript sequences (3) on a subset of the human chromosomes to train the parameters of the neural network, and transcripts on the remaining chromosomes (paralogs excluded) to test the network's predictions. The architecture consists of 32 dilated convolutional layers (Fig. S2). For pre-mRNA transcripts in the test dataset, the network predicts spliced positions with 95% top-k accuracy, which is the fraction of correctly predicted splice sites at the threshold where the number of predicted sites is equal to the actual number of splice sites present in the test dataset (4,5) and has a PR-AUC of 0.98. In addition, the network predicted known splice junctions in long noncoding RNAs (lincRNAs) with 84% top-k accuracy.

For a mutated gene, we input a ~20k nucleotide-long sequence surrounding the start position of the mutation to SpliceAI (depends on the mutation) so that every position that can potentially be affected by the mutation is analyzed (every position up to 5000nt from a mutation is affected, and it will need 5000nt from each side to be analyzed). Therefore, passing this sequence to SpliceAI will return two vectors of ~10k probabilities – one for acceptor sites and another for donor sites.

Note that if the sequence's length is greater than the length of the pre-mRNA it's working on, we pad the input sequence with N's on the sides to make it so every position to be analyzed has 10,000nt of context.

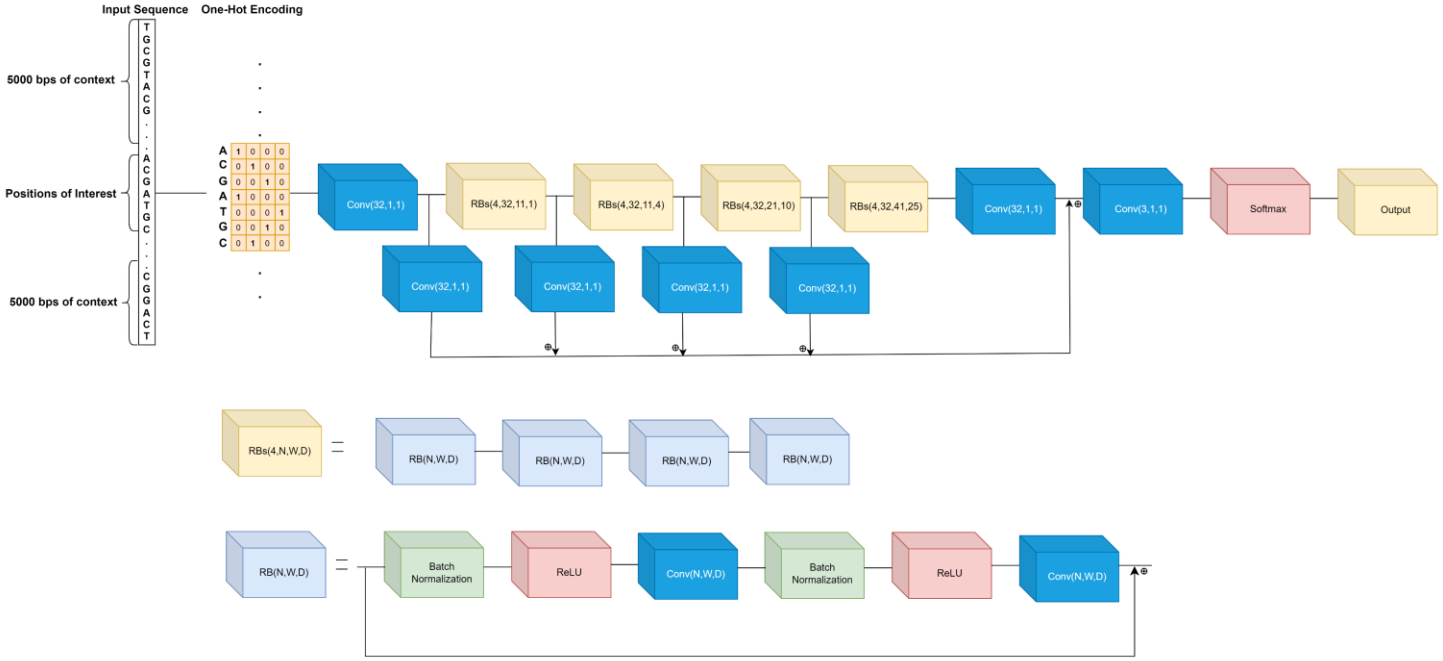

Fig S2: Architecture of SpliceAI. *Each residual block (RB) has three hyper-parameters N, W, and D, where N denotes the number of convolutional kernels, W denotes the window size, and D denotes the dilation rate of each convolutional kernel. The hyper-parameters were chosen such that the number of nucleotides around a position to be analyzed is 10000.*

3. Modeling variant mRNAs of mutated genes

We denote each analyzed position as a missed splice site if it was annotated as a donor/acceptor and its probability of being such a site, as predicted by SpliceAI, has decreased by more than 50%. Conversely, if a position's probability of being a splice site increased by more than 50%, we denote it as a discovered splice site (Fig. S3A).

Following the predicted change in splice sites, we add the discovered splice sites to the annotated sites set and remove the missed splice sites. We then consider all possible isoforms created by this set when sequentially joining splicing sites from the 5' end of the transcript to its 3' end, joining two sites only if one of them is a donor and the other is an acceptor (Fig. S3B). In addition, the isoforms must begin with a donor and end with an acceptor.

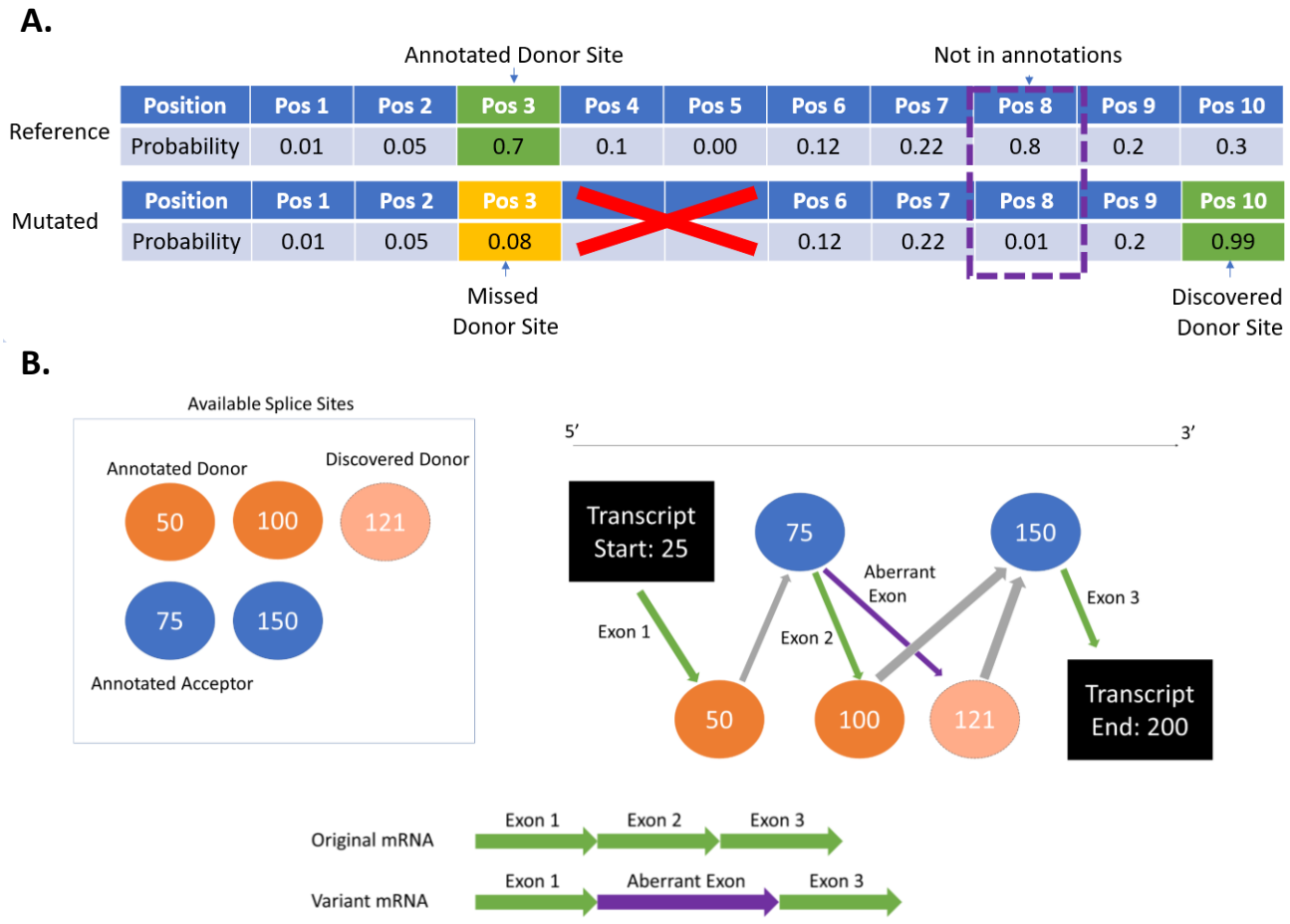

Fig S3: How isoforms are constructed following mis-splicing. A. *An example of how aberrant splicing events are found. Green/yellow cells indicate found/missed splicing sites, respectively. Positions 4-5 were deleted by CRISPR; position 3 is a missed splicing site, since it was annotated as a splicing site and its predicted probability of being a donor site decreased by more than 0.5. Position 10 is a discovered splicing site, since it did not appear in the annotations and its probability of being a donor site increased by more than 0.5. If a position received a high probability in the reference sequence but is not recognized as a splice site in the annotations, it is ignored when looking for missing splice sites.* ***B.*** *An example of constructing different isoforms of the same mRNA. This transcript originally had*

3 exons, with 2 acceptor and 2 donor sites ("Original mRNA"). Due to a mutation, a discovered donor site was added as an option. Since we cannot know which donor is more likely, we construct an additional isoform, with the discovered splice site ("Variant mRNA").

Another criterion considered when composing variant mRNAs is that the discovered exons must not exceed 2000nt since exons are usually less than 200nt in length (~80% of annotated transcripts) and are rarely more than 2000nt (less than 1%). Moreover, such long exons they are biologically unlikely and possibly decomposed by the cell (6,7). With the variant mRNA constructed, the Splicing sub-model checks if the start and stop codons have been mutated in a way that will change their function. If the Translation Initiation sub-model is selected by the user, then the alternative start codon will be the one it outputs. Otherwise, the start codon is the one in the annotations (and if it's deleted then the protein is deemed to have been erased). In case a stop codon is erased, the stop codon that will be chosen will be the first one encountered in the reading frame.

4. **Oncosplice**

Oncosplice predicts the disruption caused to a gene following a change in splicing caused by an indel (8). Its first step is using Rate4Site (9), using as input multiple sequence alignments (MSAs) downloaded from the UCSC website (10) which compare the human proteome to the homologous proteomes of 99 vertebrates. Rate4Site scores were calculated for each position in the proteins included in the MSA, covering approximately 90% of the human proteome.

The process of calculating a gene's splicing score is described here and illustrated in Fig. S4. Briefly, Oncosplice begins by calculating each isoform's score, using a sliding window of conservation scores; the average of such score over a transcript's isoforms is the transcript's score, and the maximal of such score over a gene's transcripts is the gene's score.

First, for each position i in every analyzed isoform, the normalized conservation score s_i is defined as:

$$(1)\ s_i = e^{-s_i'}$$

Where s_i' is the Rate4Site score for position i. If conservation values are not available for a given protein, each amino acid residue is given a score of $s_i = 1$. Then a sliding window is used to calculate average conservation scores along the protein; the window's length was chosen to be 76, which is the mean domain length of all human proteins (calculated using Interpro (11)). For each window, Oncosplice calculates $c = \frac{C}{C_{max}}$, where C is the average s_i score of the window, and C_{max} is the maximum C of all such windows in the proteome. Oncosplice goes over all indels that intersect with the window and defines each indel's disruption score as $s_{indel} = \max\left(1, \frac{l}{w}\right) \cdot c$, where l is the intersection length between the indel and the window, and w is the window length (as described above, $w = 76$). The window's score s_{window} would then be the sum of all such s_{indel}, and the isoform's preliminary score is $s'_{isoform} = \max s_{window}$. We normalize the isoform's score by c, the maximal such score calculated over the ClinVar dataset, such that the final isoform score is $s_{isoform} = \min\left(\frac{s'_{isoform}}{c}, 1\right)$.

Having obtained disruption scores for all isoforms, the corresponding transcript's splicing score was defined as the average disruption score over all its isoforms (assuming isoforms have equal probabilities of appearing); and finally, the gene's splicing score is the maximal score out of all its transcripts (i.e. s'_{ij} from Eq. 2 in the main paper). This score ranges from 0 to 1, with a higher score denoting higher disruption to the mutated protein's function. We note that if a transcript of interest is provided, we calculate only that transcript's score.

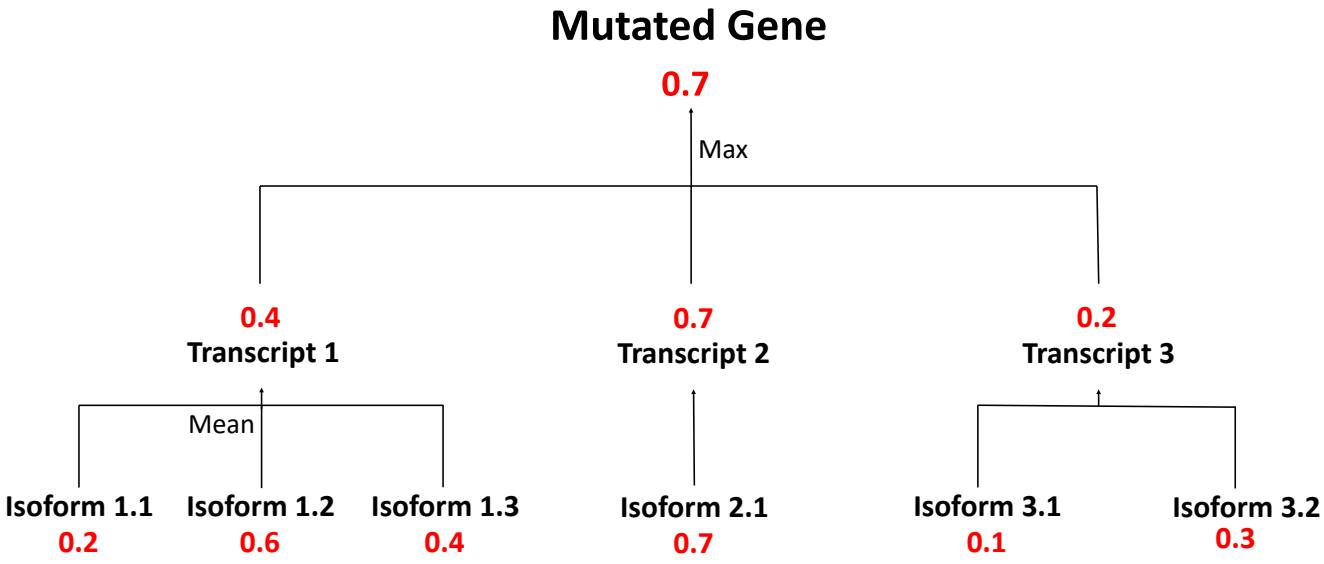

Fig S4: An example of how the splicing score is calculated for a gene. *In this example, following the mutation there are 3 isoforms in Transcript 1, one isoform in Transcript 2, and 2 isoforms in Transcript 3 of the gene. The numbers indicate the splicing score. The score for each transcript is the mean of the scores of the isoforms. The score for the gene is maximum of the scores from the transcripts. Note that this logic applies for all sub-models.*

5. TITER

The architecture of the network consists of convolutional layers followed by max pooling, then a recurrent layer of LSTM followed by a logistic layer (Fig. S5). It was trained and tested on QTI-seq data from a HEK293 cell line (12) and the annotated TISs from Ensembl v84 (13) (together denoted by Gao15 from (12)).

When tested on ACG and NUG codon (where N can be any nucleotide) TITER on a dataset constructed from QTI-seq data from (12) which had ~770 positive and ~9900 negative samples, TITER yielded areas under the receiver-operating characteristic and the precision recall curves scores of 89.1 and 61.8% respectively. It's important to note the reason the testing was done only on these codons is that in the QTI-seq data, the authors noticed that these 5 codons made up more than 75% of the cases,

The score for a codon is the product of two terms:

$$(1) \quad TISScore(s) = CodonScore(s) \times ContextScore(s)$$

Where s is the sequence profile of a codon site of interest, $ContextScore(s)$ is the output from the neural network and $CodonScore(s)$ is a term calculated independently to account for the prevalence of certain codons in the human genome (for more details, please see (14))

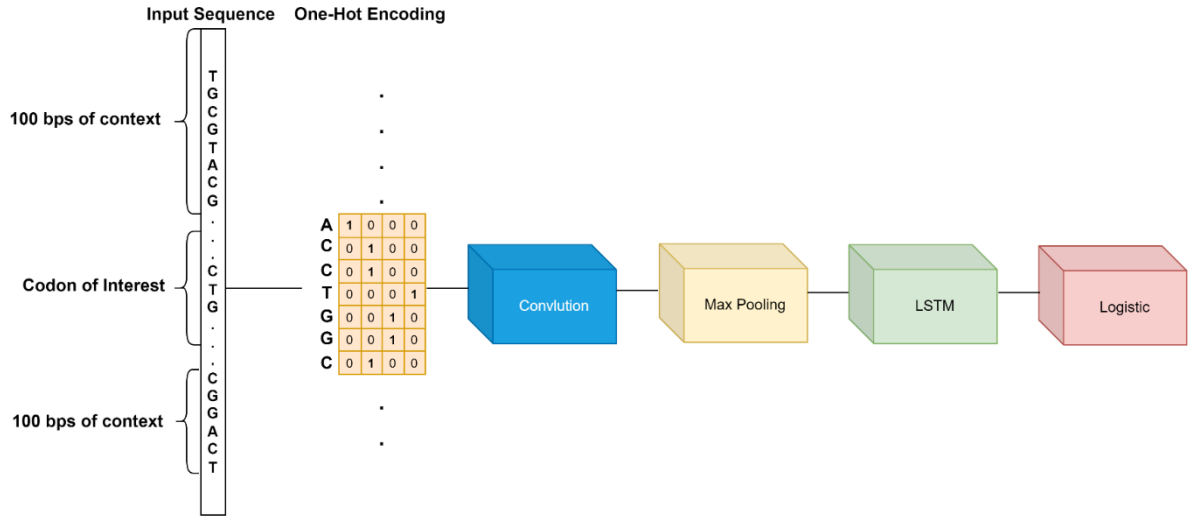

Fig S5: Schematic illustration of the architecture of TITER. *After one-hot encoding the 203-long nucleotide sequence centered around the codon in question, the sequence is passed through several convolution operators, then several pooling operators and then through recurrent layers (LSTM). Finally, the outputs from all the LSTMs are concatenated and fed into a logistic regression layer which outputs a probability of the codon in question to be a TIS.*

6. Finding a suitable start codon

For each isoform, we first conduct local alignment (using Biopython) to find the original start codon's location in the mutated mRNA sequence, using a 50nt window centered around the start codon. If the alignment failed (i.e. the score of the best alignment is 0 or less), we say that no start codon was found. Assuming the alignment succeeded, we analyze every possible start codon that is in the same frame as the original mRNA and inside a 40nt window centered around the original start codon. The translation initiation score is calculated using TITER (14), a tool which combines a deep learning algorithm and prior knowledge of preferred TIS codon composition to predict Translation Initiation Sites (TISs); See more details in the "TITER" section. The input to the tool is a 200nt window centered around a potential start codon and the output is a score (≥0) which is indicative of its likelihood to be a TIS. In accordance with the findings described in TITER's paper, the possible start codons we are looking for are NUG (where N is any nucleotide) and ACG.

If any codon has been found and either its TITER score was higher than the original start codon's score, or its rank amongst all TITER scores for the annotated TISs in the human transcriptome was within 5% of the original start codon's rank, then this codon is deemed a suitable start codon; out of all the suitable codons found this way, the new start codon is defined to be the one with the highest TITER score. If we haven't found any suitable start codons, we repeat the process with a 100nt window instead of 40nt; if we still find no suitable codons, we repeat the process with increasing window sizes: 200nt, 400nt and 800nt. If this process still hasn't yielded suitable start codons, the new start codon would be defined as the highest-scoring potential start codon found in the final, 800nt window.

7. Chosen thresholds for ClinVar analysis

We list here the 95^{th}-99^{th} percentile values of each EXPosition's gene expression sub-models' score; see "Mutations with high gene expression scores are overrepresented in ClinVar-designated pathogenic mutations" and figure 3 in the main text.

Percentile	Transcription Score	Splicing Score	Translation Initiation Score
0.99	0.119	6687	0.778
0.98	0.076	5060	0.435
0.97	0.056	4096	0.099
0.96	0.044	3388	0.026
0.95	0.035	2880	0.005

Table S1: Raw scores of sub-models in ClinVar analysis corresponding to the percentiles 95%-99%.

In table S2, we provide the maximal scores calculated by each sub-model on the ClinVar set, as well as the raw and normalized threshold values used in the final program.

	Transcription	Splicing	Translation initiation
Raw threshold	0.02	21	0.099
Maximal value	0.97	105,944	65.42
Normalized threshold	0.02	1.98×10^{-4}	0.001

Table S2: Raw threshold, maximal value achieved and normalized threshold values for each sub-model

It is also possible for the user to increase the sensitivity of the tool by taking the maximum score (instead of the average) over all mutations predicted\inserted and checking if it passes the threshold. This is done by checking the appropriate box in the tool.

8. Averaging of repeats in the Doench et al (2016) and Shalem et al. experiments

The dataset comprised two repeats of the experiments conducted using two distinct delivery systems (lentiGuide and lentiCRISPRv2) in the A375 cells, as well as three repeats for a single delivery system in the HT29 cells. In both the A375 and HT29 cells, we calculated the average values across these repetitions.

For the A375 data (both in Shalem and Doench), we first averaged the sgRNA fold change over the repeats, separately for each delivery system subset (lentiCRISPRv2 and lentiGuide); we then set the two subsets to have the same mean by multiplying all measurements in one of them with a constant. Finally, for each target site we took the average sgRNA fold change over the two delivery subsets. For the HT29 data, we simply took the average over all repeats.

9. Technical details regarding the training of the SVM classifiers and XGB regressors in figure 5

When training the XGBoost regressor, we used up to 1500 samples from the training set, which represented the distribution of the training set empirical values (using stratification if the training set was larger than 1500 samples), to select hyperparameters for the XGB regressor using GridSearchCV (using cross-validation of 5 and the scoring metric as ‘r2’). We then trained the XGB regressor using those hyperparameters and all of the training data.

When training the SVM classifier for all datasets, we assigned the label of “silencing” to a sgRNA if the depletion was at least 2-fold (i.e. the log2fold was lower than -1), and “non-silencing” otherwise (like in the analysis in the VBC paper). We stratified the data each time when dividing it into train and test. We also used under-sampling (using imblearn.under_sampling) when training the SVM’s for the data from Shalem and Doench (2016) to speed up the training process. In addition, we oversampled (using imblearn.over_sampling) when training the data from Doench (2014) and Xu et al, to account for the low number of silencing sgRNAs available in those sets.

For each dataset, sites that were not found in the genome or not analysable by EXPosition’s gene expression component / VBC / GuidePro due to technical problems were omitted.

10. Classification results on the datasets analyzed

In the main text, we described how we trained and tested a SVM classifier and a XGB regressor on data of sgRNA depletion rates from various datasets. We assessed the performance of our models using various metrics to show the results are robust. The metrics we used for classification are: AUROC (Area Under the Receiver Operating Characteristic Curve) which measures a model's ability to distinguish between classes across different classification thresholds, focusing on both true positive and false positive rates; AUPRC (Area Under the Precision-Recall Curve) emphasizes performance on the positive class by considering precision and recall; F1 score is the harmonic mean of precision and recall, balancing the trade-off between them for a specific threshold (0.5 in our case).

Tables S3-S5 describe the results when training SVM's with EXPosition's gene expression features added to VBC and GuidePro as compared to an SVM trained with just VBC or GuidePro. The overall improvement in results in the shows that additional information is embedded in the gene expression estimates, hence these estimates are valid.

A vs. B (Classification – AUROC)	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
VBC vs. VBC + EXP. gene expression	0.583 vs 0.598	* **0.563 vs 0.569**	* **0.659 vs 0.671**	* **0.755 vs 0.800**	* **0.755 vs 0.800**	* **0.550 vs 0.597**
GuidePro vs. GuidePro + EXP. gene expression	* **0.523 vs 0.542**	* **0.504 vs 0.513**	* **0.611 vs 0.617**	* **0.762 vs 0.795**	* **0.762 vs 0.795**	0.500 vs 0.507

Table S3: EXPosition's gene expression features improve AUROC performance when predicting Silencing\Non-silencing sgRNAs. *"VBC vs. VBC + EXP. gene expression" and "GuidePro vs. GuidePro + EXP. gene expression" are the comparisons between SVMs trained with VBC\GuidePro and SVMs trained with VBC\GuidePro and EXPosition's gene expression models. The positive class was defined as any sgRNA that underwent at least 2-fold depletion (as we done by the authors of VBC). Each cell contains the median AUROC's obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where the added gene expression features from EXPosition improved performance with statistical significance ($p<0.05$, Wilcoxon rank-sum test*).

A vs. B (Classification – AUPRC)	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
VBC vs. VBC + EXP. gene expression	0.147 vs 0.154	* **0.303 vs 0.313**	* **0.089 vs 0.098**	* **0.846 vs 0.889**	* **0.846 vs 0.889**	* **0.028 vs 0.033**
GuidePro vs. GuidePro + EXP. gene expression	* **0.126 vs 0.132**	* **0.267 vs 0.272**	* **0.078 vs 0.082**	* **0.867 vs 0.877**	* **0.867 vs 0.877**	0.027 vs 0.025

Table S4: EXPosition's gene expression features improve AUPRC performance when predicting Silencing\Non-silencing sgRNAs. *"VBC vs. VBC + EXP. gene expression" and "GuidePro vs. GuidePro + EXP. gene expression" are the comparisons between SVMs trained with VBC\GuidePro and SVMs trained with VBC\GuidePro and EXPosition's gene expression models. The positive class was defined as any sgRNA that underwent at least 2-fold depletion (as we done by the authors of VBC). Each cell contains the median AUPRC's obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where the added gene expression features from EXPosition improved performance with statistical significance (p<0.05, Wilcoxon rank-sum test*).

A vs. B (Classification – F1)	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
VBC vs. VBC + EXP. gene expression	* **0.233 vs 0.238**	* **0.396 vs 0.403**	0.151 vs 0.146	0.767 vs 0.764	0.767 vs 0.764	* **0.042 vs 0.046**
GuidePro vs. GuidePro + EXP. gene expression	* **0.203 vs 0.213**	0.385 vs 0.377	0.137 vs 0.137	* **0.749 vs 0.755**	* **0.749 vs 0.755**	0.037 vs 0.037

Table S5: EXPosition's gene expression features improve F1 performance when predicting Silencing\Non-silencing sgRNAs. *"VBC vs. VBC + EXP. gene expression" and "GuidePro vs. GuidePro + EXP. gene expression" are the comparisons between SVMs trained with VBC\GuidePro and SVMs trained with VBC\GuidePro and EXPosition's gene expression models. The positive class was defined as any sgRNA that underwent at least 2-fold depletion (as we done by the authors of VBC). Each cell contains the median F1 score obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where the added gene expression features from EXPosition improved performance with statistical significance (p<0.05, Wilcoxon rank-sum test*).

Tables S6-S8 describe the results when training SVM's with all the outputs of EXPosition (EXPosition's gene expression features, the predicted cutting efficiency, VBC and GuidePro) as compared to an SVM trained with just VBC or GuidePro. EXPosition outperforms VBC and GuidePro in almost all cases (and wins in the majority according to each quantification metric).

A vs. B (Classification – AUROC)	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
VBC vs. EXP.	* **0.583 vs 0.627**	* **0.563 vs 0.577**	* **0.659 vs 0.668**	* **0.755 vs 0.872**	* **0.755 vs 0.872**	0.550 vs 0.494
GuidePro vs. EXP.	* **0.523 vs 0.627**	* **0.504 vs 0.577**	* **0.611 vs 0.668**	* **0.762 vs 0.872**	* **0.762 vs 0.872**	0.500 vs 0.494

Table S6: SVM trained with EXPosition features produces better AUROC scores than SVM trained solely with features from previous tools. *"VBC vs. EXP." and "GuidePro vs. EXP." are the comparisons between SVMs trained with VBC\GuidePro and SVMs trained with all EXPostion predictions (VBC, GuidePro, CRISPRedict and EXPosition's gene expression models). The positive class was defined as any sgRNA that underwent at least 2-fold depletion (as we done by the authors of VBC). Each cell contains the median AUROC's*

obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where EXPosition outperformed the other tool with statistical significance (p<0.05, Wilcoxon rank-sum test).

A vs. B (Classification – AUPRC)	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
VBC vs. EXP.	* **0.147 vs 0.181**	* **0.303 vs 0.317**	* **0.089 vs 0.097**	* **0846 vs 0.927**	* **0.846 vs 0.927**	0.028 vs 0.027
GuidePro vs. EXP.	0.126 vs 0.181	* **0.267 vs 0.317**	* **0.078 vs 0.097**	* **0.867 vs 0.927**	* **0.867 vs 0.927**	0.027 vs 0.027

Table S7: SVM trained with EXPosition features produces better AUPRC scores than SVM trained solely with features from previous tools. *"VBC vs. EXP." and "GuidePro vs. EXP." are the comparisons between SVMs trained with VBC\GuidePro and SVMs trained with all EXPostion predictions (VBC, GuidePro, CRISPRedict and EXPosition's gene expression models). The positive class was defined as any sgRNA that underwent at least 2-fold depletion (as we done by the authors of VBC). Each cell contains the median AUPRC's obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where EXPosition outperformed the other tool with statistical significance (p<0.05, Wilcoxon rank-sum test*).

A vs. B (Classification – F1)	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
VBC vs. EXP.	* **0.233 vs 0.254**	* **0.396 vs 0.415**	0.151 vs 0.148	* **0.767 vs 0.849**	* **0.767 vs 0.849**	0.042 vs 0.032
GuidePro vs. EXP.	0.203 vs 0.254	* **0.385 vs 0.415**	* **0.137 vs 0.148**	* **0.749 vs 0.849**	* **0.749 vs 0.849**	0.037 vs 0.032

Table S8: SVM trained with EXPosition features produces better F1 scores than SVM trained solely with features from previous tools. *"VBC vs. EXP." and "GuidePro vs. EXP." are the comparisons between SVMs trained with VBC\GuidePro and SVMs trained with all EXPostion predictions (VBC, GuidePro, CRISPRedict and EXPosition's gene expression models). The positive class was defined as any sgRNA that underwent at least 2-fold depletion (as we done by the authors of VBC). Each cell contains the median F1 score obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where EXPosition outperformed the other tool with statistical significance (p<0.05, Wilcoxon rank-sum test*).

We were also interested in comparing the results from our gene expression models with VBC\GuidePro. The results can be seen in Tables S9-S10. As explained in the main text, the EXPosition's gene expression component does not compete on its own with previous tools. This is primarily attributable to the inherent bias in the datasets analyzed, which favor VBC and GuidePro over our gene expression models. These datasets are specifically designed to target the gene's coding sequence and alter its amino acid sequence, aligning with the objectives of tools like VBC and GuidePro. In contrast, they are less suited to capturing the regulatory mechanisms underlying gene expression processes—such as transcription, splicing, and translation initiation—which are the focus of EXPosition's gene expression prediction component.

GuidePro vs. EXP. gene expression	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
Spearman	* **0.044 vs 0.131**	* **0.019 vs 0.072**	0.133 vs 0.098	0.457 vs 0.290	0.414 vs 0.297	0.152 vs 0.120
AUROC	* **0.523 vs 0.550**	0.504 vs 0.533	0.611 vs 0.575	0.762 vs 0.675	0.762 vs 0.675	* **0.500 vs 0.562**
AUPRC	* **0.126 vs 0.132**	0.267 vs 0.290	0.078 vs 0.073	0.867 vs 0.799	0.867 vs 0.799	* **0.027 vs 0.032**
F1	* **0.203 vs 0.224**	0.385 vs 0.352	0.137 vs 0.124	0.749 vs 0.667	0.749 vs 0.667	0.037 vs 0.035

Table S9: Comparison between GuidePro and EXPosition's gene expression component. *"Spearman" refers to the comparisons between the regressor trained with GuidePro and the regressor trained with predictions from EXPosition's gene expression models. Each cell in that row contains the median Spearman correlations obtained from 100 cross-validations of training a XGBoost regressor on randomly chosen 80% of the data and testing on the remaining 20%. "AUROC"/"AUPRC"/"F1" are the comparisons between SVMs trained with GuidePro and SVMs trained with EXPosition's gene expression models. Each cell in those rows contains the median AUROC/AUPRC/F1 score obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where EXPosition outperformed the other tool with statistical significance (p<0.05, Wilcoxon rank-sum test*).

VBC vs. EXP. gene expression	Doench 16 (A375, N=90k, n_pos = 0.12)	Doench 16 (HT29, N=61k, n_pos = 0.26)	Shalem 14 (A375, N=53k, n_pos = 0.06)	Xu 15 (KBM7, N=2k, n_pos = 0.68)	Xu 15 (HL60, N=2k, n_pos = 0.68)	Doench 14 (A375, N=1k, n_pos = 0.02)
Spearman	0.157 vs 0.131	0.128 vs 0.072	0.210 vs 0.098	0.423 vs 0.290	0.400 vs 0.297	0.187 vs 0.120
AUROC	0.583 vs 0.550	0.563 vs 0.533	0.659 vs 0.575	0.755 vs 0.675	0.755 vs 0.675	0.550 vs 0.562
AUPRC	0.147 vs 0.132	0.303 vs 0.290	0.089 vs 0.073	0.846 vs 0.799	0.846 vs 0.799	0.028 vs 0.032
F1	0.233 vs 0.224	0.396 vs 0.352	0.151 vs 0.124	0.767 vs 0.667	0.767 vs 0.667	0.042 vs 0.035

Table S10: Comparison between VBC and EXPosition's gene expression component. *"Spearman" refers to the comparisons between the regressor trained with VBC and the regressor trained with predictions from EXPosition's gene expression models. Each cell in that row contains the median Spearman correlations obtained from 100 cross-validations of training a XGBoost regressor on randomly chosen 80% of the data and testing on the remaining 20%. "AUROC"/"AUPRC"/"F1" are the comparisons between SVMs trained with VBC and SVMs trained with EXPosition's gene expression models. Each cell in those rows contains the median AUROC/AUPRC/F1 score obtained from 100 cross-validations of training a SVM classifier on randomly chosen 80% of the data and testing on the remaining 20%, while accounting for imbalances in the labels during training. Cells with an asterisk were cases where EXPosition outperformed the other tool with statistical significance (p<0.05, Wilcoxon rank-sum test*).

References

1. Agarwal V, Shendure J. Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks. Cell Rep. 2020 May 19;31(7):107663.

2. Jaganathan K, Kyriazopoulou Panagiotopoulou S, McRae JF, Darbandi SF, Knowles D, Li YI, et al. Predicting Splicing from Primary Sequence with Deep Learning. Cell. 2019 Jan 24;176(3):535-548.e24.

3. Harrow J, Frankish A, Gonzalez JM, Tapanari E, Diekhans M, Kokocinski F, et al. GENCODE: the reference human genome annotation for The ENCODE Project. Genome Res. 2012;22(9):1760–74.

4. Yeo G, Burge CB. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. In: Proceedings of the seventh annual international conference on Research in computational molecular biology. 2003. p. 322–31.

5. Boyd S, Cortes C, Mohri M, Radovanovic A. Accuracy at the top. Adv Neural Inf Process Syst. 2012;25.

6. Singh P, Saha U, Paira S, Das B. Nuclear mRNA Surveillance Mechanisms: Function and Links to Human Disease. J Mol Biol [Internet]. 2018;430(14):1993–2013. Available from: https://www.sciencedirect.com/science/article/pii/S0022283618304042

7. Garneau NL, Wilusz J, Wilusz CJ. The highways and byways of mRNA decay. Nat Rev Mol Cell Biol [Internet]. 2007;8(2):113–26. Available from: https://doi.org/10.1038/nrm2104

8. Lynn N, Tuller T. Detecting and understanding meaningful cancerous mutations based on computational models of mRNA splicing. NPJ Syst Biol Appl. 2024;10(1):25.

9. Pupko T, Bell RE, Mayrose I, Glaser F, Ben-Tal N. Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. Bioinformatics [Internet]. 2002 Jul 1;18(suppl_1):S71–7. Available from: https://doi.org/10.1093/bioinformatics/18.suppl_1.S71

10. Pupko T, Bell RE, Mayrose I, Glaser F, Ben-Tal N. Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. Bioinformatics. 2002;18(suppl_1):S71–7.

11. Paysan-Lafosse T, Blum M, Chuguransky S, Grego T, Pinto BL, Salazar GA, et al. InterPro in 2022. Nucleic Acids Res [Internet]. 2023 Jan 6;51(D1):D418–27. Available from: https://doi.org/10.1093/nar/gkac993

12. Gao X, Wan J, Liu B, Ma M, Shen B, Qian SB. Quantitative profiling of initiating ribosomes in vivo. Nat Methods. 2015;12(2):147–53.

13. Aken BL, Ayling S, Barrell D, Clarke L, Curwen V, Fairley S, et al. The Ensembl gene annotation system. Database. 2016;2016.

14. Zhang S, Hu H, Jiang T, Zhang L, Zeng J. TITER: predicting translation initiation sites by deep learning. Bioinformatics [Internet]. 2017 Jul 15;33(14):i234–42. Available from: https://doi.org/10.1093/bioinformatics/btx247