Eng. Sci., 2025, **34**, 1452

Engineered Science
DOI: https://dx.doi.org/10.30919/es1452

CRISPR-Variational Autoencoder: An Interpretable and Efficiency-Aware Guide Ribonucleic Acid Sequence Generator

Ahmad Obeid and Hasan AlMarzouqi*

Abstract

Recent deep learning has shown significant potential in predicting guide Ribonucleic Acid (gRNA) efficiency, thereby optimizing engineered gRNAs and enhancing the application of clustered regularly interspaced short palindromic repeats (CRISPR) associated protein 9 (CRISPR-Cas) systems in genome editing. However, the black-box nature of these deep learning methods often lacks transparency, hindering our understanding of the factors that boost efficiency. Addressing this issue can significantly expand the use of CRISPR-Cas systems across various domains. We introduce CRISPR-Variational Autoencoder (CRISPR-VAE), a framework designed to interpret gRNA efficiency predictions, thereby elucidating the factors that enhance gRNA performance, specifically applied to CRISPR/Cas12a (formerly known as CRISPR/Cpf1). Our framework articulates these factors into position-specific k-mer rules. The methodology involves constructing an efficiency-aware gRNA sequence generator, trained on real-world data, to produce a large volume of synthetic sequences exhibiting desirable traits. These sequences form the basis for explaining gRNA predictions. Additionally, CRISPR-VAE functions as an independent sequence generator, providing users with fine-grained control over the sequences. This versatile framework integrates seamlessly with various CRISPR-Cas tools and datasets, demonstrating its efficacy. The complete code implementation of CRISPR-VAE can be found at github.com/AhmadObeid/CRISPR-VAE.

Keywords: Clustered regularly interspaced short palindromic repeats; Guide ribonucleic acid; Explainable deep learning.
Received: 05 January 2024; Revised: 20 September 2024; Accepted: 22 October 2024.
Article type: Research article.

1. Introduction

Genome editing is the re-engineering of Deoxyribonucleic Acid (DNA) at a specific target site through insertion, deletion, or replacement, resulting in neutralization of target genes, modification of genetic traits and possibly, correction of pathogenic mutations. The use of clustered regularly interspaced short palindromic repeats (CRISPR) associated protein 9 (CRISPR-Cas) systems for genome editing has gained significant popularity recently due to its simple design, high efficiency, reduced cost, and consistency,[1] in addition to its diverse applications in fields such as gene therapy and agricultural engineering.[2-5] This surge in interest has spurred advancements in related research, notably in predicting the efficiency of guide Ribonucleic Acid (gRNAs). In CRISPR-Cas systems, gRNAs guide endonucleases to specific DNA targets for cleavage. The subsequent DNA repair process can result in insertions/deletions or precise gene edits through homology-directed repair, enabling gene knock-ins.[6] The efficiency of this process depends on the gRNA used; hence, predicting gRNA efficacy is crucial for ensuring high on-target indel efficiency and minimizing off-target effects.

The discovery of Cas12a, also known as CRISPR from Prevotella and Francisella1 (Cpf1), as an alternative to Cas9 represents a significant development, introducing many advantageous features. For instance, Cpf1 is smaller, requires a shorter CRISPR RNA to function, and allows for precise DNA reengineering while leaving the Protospacer Adjacent Motif (PAM) intact.[7] Additionally, Cpf1 demonstrates greater specificity in human and plant cells compared to Cas9 and facilitates the editing of Corynebacterium glutamicum and Cyanobacteria, which was not achievable with Cas9.[6]

The task of predicting the quantifiable quality assessment of a gRNA sequence involves estimating its on-target efficiency and activity. The methods developed for this task can be classified as alignment-based, hypothesis-driven, or learning-based.[8] Alignment-based methods are based solely on locating the PAM to align the gRNA within the genome. Hypothesis-driven methods score aligned gRNAs on the basis of additional contextual factors. Learning-based methods, on the other hand, train predictive models to uncover hidden sequence-related

Department of Computer Science, Khalifa University, 127788, Abu Dhabi, UAE
*Email: hasan.almarzouqi@ku.ac.ae (H. AlMarzouqi)

factors that infer the on-target efficiency of gRNAs. With continuous advances in deep learning, learning- based methods have demonstrated high accuracy and promising performance in gRNA efficiency prediction.

In the study of Kim *et al*.,[9] the DeepCpf1 and seq-DeepCpf1 predictors, developed using Convolutional Neural Networks (CNNs) and dense layers, demonstrated improved performance compared to classical methods. Similarly, Zhang *et al*.[10] employed a support vector regression (SVR) in conjunction with a CNN, resulting in performance improvements. DeepCas9 also uses CNNs,[11] while DeepCRISPR incorporates an Autoencoder (AE) stage for unsupervised representation learning,[12] and C-RNNCrispr employs a Recurrent Neural Network (RNN) for enhanced sequence learning.[8] Additionally, DeepPE is a CNN-based method designed for Prime Editing.[13] DeepCas13, proposed by Cheng *et al*.,[14] uses two parallel streams of CNN-RNN-FN layers, in which features are extracted from gRNA and secondary structures of the RNA. The recent work of Wessels *et al*.[15] also uses a CNN for the prediction of the on-target and off-target activity of CRISPR-Cas13d gRNAs. A multitude of other works also employ deep learning (DL) tools to predict the off-target activity of gRNA sequences.[16-19] Despite their great potential, the vast majority of these approaches share a similar disadvantage: a lack of strong explainability that can inform a meaningful fault diagnosis stage. This explainability is essential for a deeper understanding of CRISPR systems and to identify factors that contribute to the higher on-target activity of certain gRNAs.[20] Enhanced explainability allows practitioners to design more effective gRNA sequences and enables analysts to better diagnose the decisions of their models, thereby promoting a broader application of genome editing across various fields.

Some attempts have been made to explore this research direction.[6,8,21-23] The model score is optimized with respect to the inputted gRNA sequence to identify sequences resulting in the highest scores,[8,22] while other approaches opt for classical machine learning tools that are easier to explain,[24] thus trading accuracy for explainability. In contrast to relying on deep learning interpretation, employ statistical analysis of the available data to infer position-wise base preference rules.[6,21,23]

Most current efforts to improve interpretability encounter two primary challenges: data deficiency and data incomprehensiveness. Firstly, despite the publication of extensive datasets for both Cas9 and Cpf1 endonucleases, the quantity of available data remains limited class-wise. For example, the existing Cpf1 data includes only a few sequences with efficiency scores ≥ 0.99 or ≤ 0.05. A larger number of these highly polarized sequences is essential to deduce the most significant rules with statistical robustness.[25,26] Secondly, regarding incomprehensiveness, the available data are often fragmented in terms of sequence cohesion, displaying diverse and distinct sequence-related features. These features are also ambiguously represented within the data, complicating their identification. Establishing strict quantitative comprehensiveness over all potential gRNA sequences is impractical. Instead, our objective is to meaningfully expand the data to represent various features both quantitatively and qualitatively. In this work, our aim is to develop a framework deeply embedded in the deep learning paradigm, applicable to any CRISPR-Cas system, while addressing the aforementioned data challenges.

In this work, we develop a comprehensive framework that addresses both data deficiency and incomprehensiveness simultaneously. Specifically, the publicly available data on CRISPR/Cpf1 activity exhibit gaps related to sequence and structure in a particular analysis space (to be detailed later). We bridge these gaps by developing a sequence generator named CRISPR-Variational Autoencoder (CRISPR-VAE). CRISPR-VAE is efficiency-aware and can synthesize numerous sequences with high and low efficiencies. These sequences are not random but form a structured analysis space that fills the voids left by existing datasets, displaying various sequence phenomena organized in different locations within the space. This method generates more extensive and diverse data, providing a solid foundation for identifying efficiency-promoting rules. By concentrating and amplifying the search for these factors, our framework enhances both data quality and coverage. Finally, we predict the efficiency of synthetic sequences using the deep learning-based predictor seq-DeepCpf1, which has previously shown a strong performance.[9] By establishing concordance between two methodologically distinct frameworks, a generative model and a discriminative one, we increase confidence in our findings.

In summary, the contributions of our work are as follows:

- Introducing a deep learning-based explainability framework that can be readily integrated with any CRISPR-Cas dataset.
- Semantically articulating high-quality findings within a suitable k-mer paradigm.
- Developing the standalone sequence generator CRISPR-VAE, capable of generating sequences with high (or low) efficiencies and customizable position-wise features tailored to practitioners' needs.

2. Materials and methods

In this Section, we will describe the three main components of our proposed framework, which starts with the proposed generative framework CRISPR-VAE and its advantages, then describes the subsequent feature extraction procedure.

2.1 VAE for a structured latent space

We start by describing the general paradigm of VAE, which allows the establishment of a structured latent space. In contrast to previous attempts, the proposed work aims to accentuate and distinguish existing sequence-related phenomena and explore possible ignored ones in their neighborhood. This necessitates the establishment of a structured analysis space, for which we employ the VAE paradigm. The analysis space is composed of

numerous synthetic sequences that share a resemblance with the training data and that are systematically distributed over the space.

The VAE generative process starts by generating a latent variable $\mathbf{z}$ from the prior distribution $p_\theta(\boldsymbol{z})$. Then $\boldsymbol{x}$ is generated (recon- structed) from the generative distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. In this framework, parameter estimation is difficult due to the intractability of the posterior. Alternatively, the lower-bound of the log-likelihood is used in Eq. (1).[27]

$$log(\, p_\theta(\boldsymbol{x})) \geq -DKL\,(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p_\theta(\boldsymbol{z})) \; + \\ E_{q_\phi(z|x)}[log\; p\theta\;(\boldsymbol{x}|\boldsymbol{z})] \qquad (1)$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is an approximation for the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, and $D_{KL}(.||.)$ is the KL-divergence. In our implementation, we use CNNs and dense layers for the realization of both models $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ i.e. the encoder and decoder models, respectively, as shown in Fig. 1. Furthermore, assuming a Gaussian latent variable, the empirical loss of the VAE can be written as:[27]

$$\mathbb{L} = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) + \frac{1}{L}\sum_{t=1}^{T}\log\,(p_\theta(\mathbf{x}|z^{(t)})) \quad (2)$$

where $z^{(t)}$ is a sample drawn from the generative model, *i.e.* $z^{(t)} = g_\phi(\boldsymbol{x}, \varepsilon)$, and $\varepsilon \sim N\,(0, \mathbf{1})$ is used for the so-called reparametrization trick.[28]

The encoder model learns to project sequences into a latent space, assimilating them into a normal distribution where different sequence-related features occupy distinct locations. However, the projected sequences of the training data exhibit a noncohesive latent space, leaving certain gaps for exploration. Thus, decoding from samples selected systematically and in a structured manner from the latent space results in the synthesis of novel sequences that 1) resemble the original data and 2) fill these gaps, providing a continuous and smooth transition between different sequence phenomena.

The advantages of having a structured latent space are two-fold. First, it ensures that all phenomena present and scattered in the original data are parsed and highlighted. Second, it has been empirically demonstrated that the structured space systematically distributes different sequence phenomena across its various locations (*e.g.*, quadrants in the 2D space), facilitating the analysis and search for efficiency-promoting features. Additionally, this structure grants the sequence generator low-level editing capabilities, where specific position-wise base preferences translate to sampling from different quadrants.

Without a structured latent space for analysis, comprehensive exploration of ignored potential phenomena would be resource intensive and inefficient, with an estimated upper limit complexity in the search space of $O(31^4)$ (assuming gRNA sequences of length 34, with a known PAM of TTTV). The vast majority of such sequences would lack connection to the available data, making validation impossible. Instead, the proposed framework confines the synthesis to sequences that resemble the available data. In our implementation, we sampled 10,000 latent codes arranged in a grid of 100×100, which are decoded to synthetic sequences for subsequent analysis stages.

To test the structure of the latent space, distance heat maps for the generated sequences have been constructed as follows in Eq. (3).

$$Map_{ij}^{k} = \frac{1}{L}\sum_{l=1}^{L} H(\hat{a_{ij}}, \hat{b_l})\,, \{b_l : D^{(\infty)}(a_{ij}, b_l) = \delta_k\} \quad (3)$$

where i and j denote each point in the heat map, k denotes the specific heat map that corresponds to the used δ in the model, $\hat{a} = g_\phi(\mathrm{a}, \epsilon)$, $\hat{b} = g_\phi(\mathrm{b}, \epsilon)$ H is the Hamming distance, and $D^{(\infty)}$ is the Minkowski distance of order ∞. L denotes the number of seeds in the latent space that satisfies the Minkowski distance condition. A structured space is expected to exhibit heat maps with values growing proportional to δ.

2.2 CVAE for efficiency-awareness

In our implementation, we specifically follow the conditional VAE (CVAE) paradigm inspired by Sohn *et al.*,[28] where we condition on the efficiency score of each sequence. In this Section, we will describe the needed change that grants CRISPR-VAE its efficiency-awareness. More concretely, Eq. (2) becomes Eq. (4):[28]

$$\mathbb{L}_C = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{c})\|p_\theta(\mathbf{z}|\mathbf{c})) + \frac{1}{L}\sum_{t=1}^{T}\log\,(p_\theta(\mathbf{x}|z^{(t)}, \mathbf{c})) \quad (4)$$

where we are conditioning the encoding, the decoding, and the prior distribution on the efficiency information c. This means

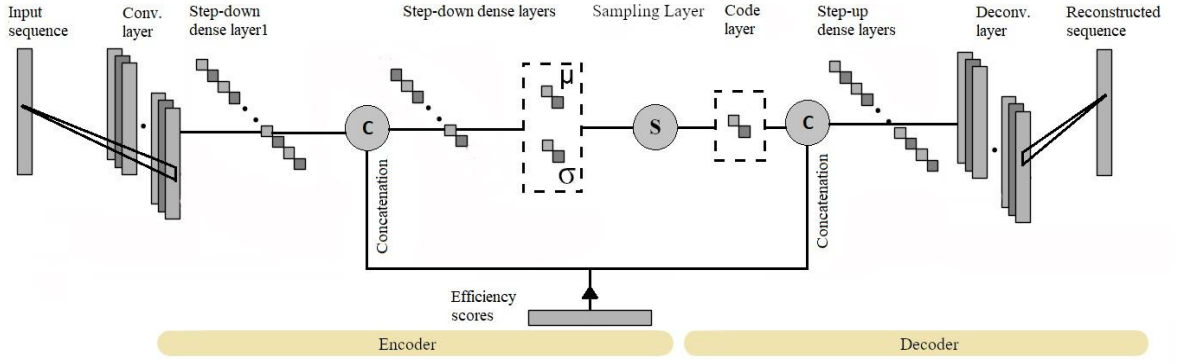

Fig. 1: CRISPR-VAE architecture, shown to integrate efficiency information at two stages: the end of the encoder and the beginning of the decoder.

that we obtain a separate latent space for each efficiency category. In our implementation, we convert the efficiency scores in the public data to integers, resulting in a 100 classes ($0-99$). Theoretically, this results in up to 100 latent spaces, each consisting of 100×100 grid of sequences. However, we confined the synthesis to classes 0-efficiency and 99-efficiency to synthesize the most polarized sequences, in order to focus on the most prominent and distinct sequence-related features that set the high-efficiency sequences apart from their low-efficiency counterparts.

Integrating the CVAE paradigm has two benefits. First, the available data provide efficiency measurements that represent useful information for improving the quality of synthesis and reconstruction of the VAE. In fact, experimentation showed that exploiting efficiency information improves VAE performance, as will be shown in Section 3. Second, exploiting the efficiency information of the data makes the VAE efficiency-aware, and makes the synthesis of the data tailored to the needs of the user (*e.g.*, focused on the high- efficiency sequences). Moreover, this enables us to investigate the agreement between CVAE and existing discriminative methods. In Section 3, we show that CRISPR-VAE and the seq-DeepCpf1 predictor are in strong agreement,[9] thus increasing confidence in the findings, without the need for laboratory tests. Here, another benefit of having a structured latent space is apparent, where we can enforce a similarity between the synthetic data and the original data, making seq-DeepCpf1 familiar with the synthetic data.

A final trick was used to improve the quality of the reconstruction of CRISPR-VAE. In particular, the first three loci in the PAM of all sequences were removed as they are constantly TTT. It was observed upon experimentation that the model finds reconstructing such motif as an easy way to score highly in the objective function; avoiding it motivated the model to rely on learning more interesting sequence-related features that improve the quality of reconstruction, which had a direct impact on said quality.

Fig. 1 illustrates the CVAE paradigm. The one-hot encoded efficiency information is fed to the network at two concatenation stages. The first stage, which comes after the first dense layer, allows the efficiency information to be blended and integrated with the sequence information via the subsequent dense layers, and into the embedding of the code layer, establishing the latent space. The second stage, which follows the code layer, allows the decoder to be a standalone efficiency-aware sequence generator. The sampling layer employs the aforementioned reparametrization trick to convert μ and σ to latent codes.

2.3 Data usage

As efficiency scores are used, it is applicable to split our data into training and testing sets to confirm the generality of our findings. We use the data provided in Kim *et al.*,[9] where high-throughput experiments were used to generate sets HT1, HT2, and HT3. We use the set HT1 for training, which consists of ~16300 sequences, while sets HT2 and HT3 were used for testing. These sets do not share any sequences, which excludes any possibility of data leakage. We also applied data augmentation by causing small perturbations in the promiscuous region of each sequence in HT1 such that the efficiency scores are likely maintained according to Kim *et al.*,[23] (2017) resulting in ~85,000 sequences to train CRISPR-VAE.

2.4 Feature extraction

Having built CRISPR-VAE, numerous sequences can be synthesized that exhibit the two main characteristics missing from the orig- inal data: comprehensiveness and plentifulness. What remains is to extract the sequnce-related features that are responsible for the disparity in the sequences that belong to 0-efficiency and 99-efficiency classes. To that end, two methods were used to extract and articulate the most prominent features explored in the synthetic data. The first consists of k-mer histogramming analysis to build Mer Significance Maps (MSM), and the second consists of visualizing class activation maps (CAM) produced by a binary classifier that distinguishes high-efficiency from low-efficiency sequences.[29]

Firstly, following Kim *et al.*,[9] we define the three regions of seed, trunk, and promiscuous, in addition to PAM, pre-PAM, and post-seq regions, as illustrated in Fig. 2. In our initial method, we utilized a moving overlapping window encapsulating k-mers to segregate feature extraction based on the position within the gRNA sequence, providing the necessary contextual specificity for the analysis. We chose a step size of 1 base, resulting in $L - k + 1$ sub-regions for each parent region, where L is the length of the region (*e.g.*, $L_{trunk} = 12$), and k is the mer size. In this paper, we restricted the experiments to $k = 3$, but other values can be easily implemented.

Additionally, we divided the latent space into equal-sized locations (*e.g.*, four quadrants in a 2D space), where histogramming occurs independently. This approach is designed to highlight specific phenomena that might otherwise be overshadowed by more dominant ones. It was observed that different sequence features become prominent at various locations within the latent space.

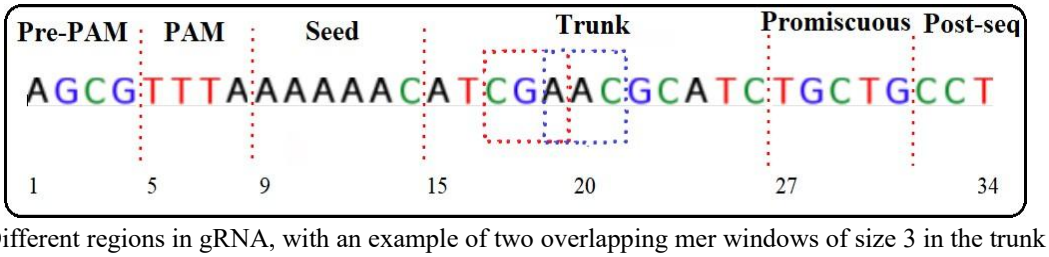

Fig. 2: Different regions in gRNA, with an example of two overlapping mer windows of size 3 in the trunk region.

After pooling all possible features, we filter them by setting an empirical significance threshold (η). Features with a frequency below this threshold are discarded. The threshold is chosen as m% (typically 7–10%) of the number of possible mers, multiplied by the number of sequences under analysis (N), as shown in equation (5). N can refer to the entire set of generated sequences or only sequences within a specific quadrant. η is defined separately for each quadrant, each region in the gRNA, and for high-efficiency and low-efficiency sequences, but with a constant m% throughout.

Next, we discarded features that are simultaneously above the low-efficiency and high-efficiency significance thresholds, *i.e.*, the common features between the two classes. Additionally, we highlight novel features that are obscure in the training data but discovered due to the advantages introduced by the synthetic data and later found in the testing data. In other words, these features would probably have been overlooked if the analysis had not involved synthesizing sequences, similar to Zhu and Liang,[6] alluding to the added generality conferred by the proposed framework.

Thus, we can identify the differences between the proposed paradigm and that of Zhu and Liang as follows:[6] Our approach introduces more specificity by exploiting the hidden cohesion in the training data, greater sensitivity by highlighting potentially overshadowed features, and greater generality by discovering the neighborhood of the real sequences.

$$\eta = \frac{m}{100}(L - K + 1) * N \tag{5}$$

In the second method, we trained a binary classifier on the synthetic data and visualized the CAMs for each quadrant separately.[29] These CAMs are obtained by maximizing the score of the binary efficiency classifier with respect to the first convolutional layer. As such, we produce attention maps that visually justify the decision of the classifier. We produced CAMs for all 99-efficiency sequences and averaged them. Moreover, CAMs features are compared with MSMs features, thus enabling us to gauge the agreement between both methods. Moreover, the two methods are complementary to each other. The first method has a finer granularity in terms of the location of prominent features, while the second is an automatic method that directly explains the decision of the binary classifier. Upon investigation of both methods simultaneously, numerous features can be extracted, and a detailed understanding of the composition of high-efficiency sequences can be made.

2.5 Considerations on the latent space dimension

The latent space used is two-dimensional (2D), but the analysis can be extended to higher dimensions, enhancing the VAE architecture's capability to reconstruct and synthesize higher-quality data and generate a larger quantity of synthetic data. This extension comes at the cost of increased complexity and higher storage requirements. In detail, consider $z \in \mathrm{R}^d$, where d is the dimension of the latent space. In the CRISPR-VAE pipeline, the final step-down dense layer will connect to $2 * d$ neurons. As such, increasing d will cause a proportional increase in the connected neurons. The same effect will also take place in the decoding direction, as can be seen in Fig. 1. Having more neurons in the pipeline can potentially increase the prowess of the network, resulting in better analysis, and higher-quality synthetic sequences. However, this is subject to training on larger training data; otherwise, the model may overfit the small-scale data and produce inferior results. If data availability is a constraint, various regularization techniques must be employed such as dropout and weight decay. Moreover, as d increases, the interpretability of the latent space decreases. Specifically, confirming the structure of that the constructed space becomes challenging, as the distance heat maps become highly convoluted with $d > 3$.

From another perspective, with all things held constant, increasing the latent space dimension d would also increase the number of analysis hyperplanes. The analysis takes place per quadrants with $d = 2$, and per octants with $d = 3$. Similarly, the number of analysis hyperplanes increases exponentially as per the relation $p = 2^d$, where p denotes the number of equisized analysis hyperplanes. The statistical analysis over an increased number of samples may increase the confidence in the findings; however, it may also be constrained by memory/storage limitations, and may possibly incur increased time requirements; effects that may hinder the synthesis and analysis of gRNA sequences.[30,31]

2.6 Ethical considerations for CRISPR-VAE usage

The ethical considerations surrounding CRISP-VAE intersect with those related to CRISPR technology and machine learning (ML). Regarding CRISPR, one of the most critical concerns is the risk of off-target effects.[32] Currently, CRISP-VAE focuses on discovering high-efficiency features but is not yet equipped to reject sequences that might cause unintended gene edits. This limitation underscores the importance of enhancing future versions of our system to incorporate safety checks, such as filtering for sequences that reduce off-target risks, aligning with ongoing advances in the field.

In the context of machine learning, ethical issues include bias in training data and algorithmic transparency.[31] Since CRISPR- VAE relies on large datasets of gRNA sequences, it is essential for users to ensure that the data used in their implementation is diverse and representative, avoiding any biases that could skew results. Additionally, the transparency and interpretability of machine learning models are critical, particularly in sensitive fields like gene editing.[30] CRISPR-VAE promotes interpretability by not only generating high-efficiency sequences but also highlighting the key features that contribute to their effectiveness.

3. Results and discussion

We report our findings in two parts. Firstly, we show results that confirm the efficacy of the proposed paradigm. Secondly, we summarize the inferred sequence-related features from the two methods mentioned in 2.4.

3.1 Confirming the validity of the proposed framework

We start by investigating the proper structuring of the constructed latent space. A structured latent space exhibits smooth transitions between different sequences, placing similar sequences in close proximity. Consequently, varying δ (Eq. (3)) in a structured latent space should produce heat maps with values proportional to δ. In Fig. 3, we use $\delta = \{3, 11, 29\}$ to compare sequences with small, medium, and large separations. The heat maps shown correspond to the class of 99-efficiency sequences. As expected from a structured latent space, the heat maps display the smallest values for $\delta = 3$ and the largest for $\delta = 29$.

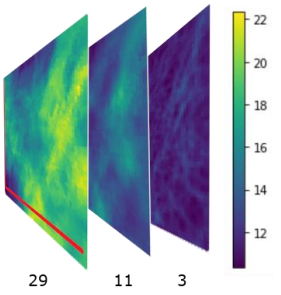

Fig. 3: Results of heat map generation by Eq. (3), showing a monotonic positive relationship between δ = {3, 11, 29} and the Hamming distance, indicating the sequential structure in the synthetic latent space.

Furthermore, Fig. 4 provides a visual assessment of the agreement between the proposed generative CRISPR-VAE and the discrim- inative seq-deepCpf1 predictor. The figure presents the predictions of seq-deepCpf1 on the synthetic data belonging to 0-efficiency and 99-efficiency classes as produced by CRISPR-VAE. Ideally, the figure should have two disjoint peaks at the extremes of 0 and 99. The tendency towards this behavior is demonstrated in Fig. 4, and is quantified by a Spearman's correlation coefficient of $0.798, CI : [0.792, 0.800]$. This indicates that according to seq-deepCpf1, most sequences produced for the 99-efficiency class have higher efficiencies than those produced for the 0-efficiency class, thus facilitating the identification of the most prominent features distinguishing the two types of sequences.

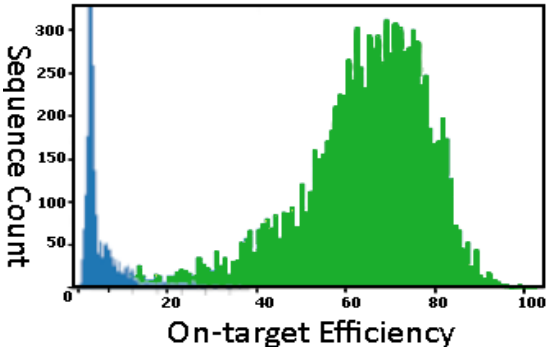

Fig. 4: Seq-deepCpf1 prediction on the synthetic data of efficiencies 0 (left) & 99 (right).[9]

3.2 High-efficiency features

In this paper, we include Fig. 5**Fig. 5** for a holistic summary of the sequence-related features in different sequence regions and quadrants in the latent space. These features pass the filtering stages and are significantly prominent in the sequences of class 99-efficiency in contrast to sequences of class 0-efficiency. Filtering with the significance threshold results in focusing on the prominent features and consequently having some empty subregions in Fig. 5c. The MSMs consist of concentric significance circles, whose radii are inversely proportional to their significance. The different regions are color-coded as per the legend in Fig. 5a. The discovered mers are dispersed in the MSMs on the basis of their position in the gRNA; the larger the angle at which the mer is located, the farther down the gRNA stream it exists. For clarity, we also segregated each region into three subregions separated by dashed lines. These subregions describe the beginning, middle, and end of each region. In Fig. 5c we highlight the significant features found in the synthetic data that are obscure in the training data HT1, but are confirmed by the testing data HT2 and HT3 by circling them.

Fig. 5 also provides a pictorial summary of the distinguishing trends that separate the two types of sequence using CAMs via a specialized classifier. The classifier is capable of classifying synthetic data with high accuracy after very few epochs (accuracy ~95%, with 5 epochs), alluding to the distinct features in both categories. We also include Fig. 5b to show the benefit of segregating the analysis to different quadrants in the latent space. Otherwise, the analysis reveals an averaged version in which only the globally prominent features are highlighted while ignoring many other valid ones. For example, the averaged summary in Fig. 5b reveals a disfavoring of Thymine right after the PAM, which agrees with the existing findings,[6] but does not reveal much more than that.

Various mers that make up high-efficiency sequences can be inferred by looking at Fig. 5**Fig. 5**. We especially focus on the features that are agreed upon by the two methods of CAMs and MSMs. Moreover, we report the confidence in the extracted features in Table 1. Note that the probability of choosing a 3-mer at random at any location in the gRNA is $1/4^3 = 0.0156$. Our results show that most observed probabilities are significantly higher than the random chance probability.

To start with, Adenine is preferred mainly in the promiscuous region, as shown in Quadrant 1 and Quadrant 3 (Q1 and Q3). Adenine therein especially prefers to combine with Cytosine or Guanine (*e.g.*, AAA and AAG in Q1, ACA and CAA in Q3). In terms of cytosine, it can be positioned everywhere in the gRNA, albeit in different combinations depending on the region, as revealed by looking at the different quadrants. More concretely, for Cytosine in to be in the seed region, it prefers to combine with the TG pair (*e.g.*, TGC and CTG in Q2), or preceded by Guanine (*e.g.*, AGC, GCA in Q3), or followed by Adenine (*e.g.*, GCA, CCA, CAT in Q3). As such, one can conclude that a motif of TGCA is observed in the seed region of efficient sequences. For Cytosine to be in the

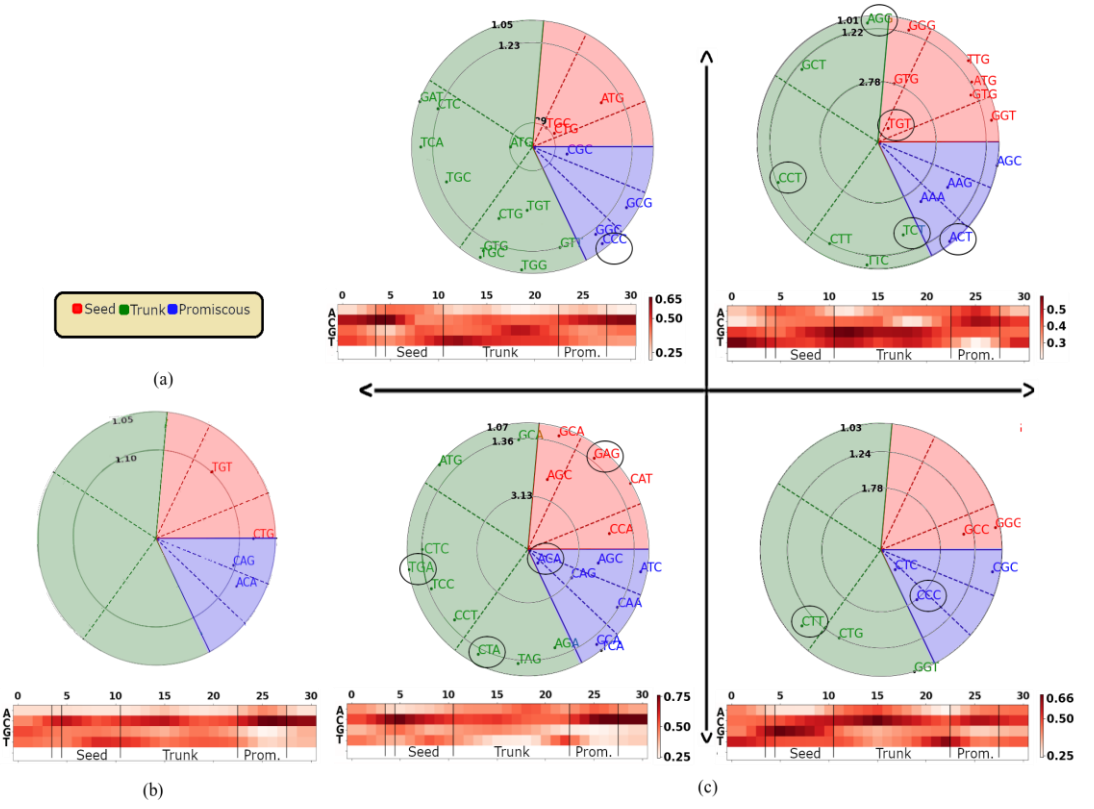

Fig. 5: Summary of high-efficiency sequence-related features as MSMs and CAMs, showing the benefit of a quadrant-based analysis (c), as compared to non-quadrant-based (b). The sequence regions are color-coded according to the legend in (a), and separated into three sub-regions (beginning→mid→end) going counter-clockwise; the circled mers in (c) can be exclusively found from the synthetic data, and not the training data, alluding to the added generality conferred by the proposed paradigm. In the MSMs, more significant features are closer to the center. Only the last nucleotide in the PAM is shown in the CAMs.

Table 1: The 95% confidence intervals (CI) of 3-mers at various gRNA locations, categorized by analysis quadrants. The highest CI in each Quadrant is in Bold.

Quadrant 1		Quadrant 2		Quadrant 3		Quadrant 4	
3-mer	CI	3-mer	CI	3-mer	CI	3-mer	CI
S-beg: GGT	[0.113 - 0.147]	S-mid: CTG	[0.198 - 0.239]	S-beg: CCA	[0.191 - 0.236]	S-beg: GCC	[0.137 - 0.172]
S-mid: ATG	[0.127 - 0.162]	S-mid: ATG	[0.145 - 0.182]	S-mid: CAT	[0.109 - 0.145]	S-beg: GGG	[0.110 - 0.142]
S-mid: GTG	[0.125 - 0.160]	S-mid: TGC	[0.201 - 0.242]	S-mid: GAG	[0.132 - 0.171]	T-mid: CTT	[0.117 - 0.150]
S-mid: TTG	[0.106 - 0.139]	T-mid: CTC	[0.125 - 0.160]	S-end: AGC	[0.220 - 0.267]	T-end: CTG	[0.129 - 0.163]
S-mid: TGT	[0.303 - 0.351]	T-mid: GAT	[0.107 - 0.139]	S-end: GCA	[0.124 - 0.162]	T-end: GGT	[0.106 - 0.137]
S-end: GTG	[0.223 - 0.266]	T-mid: ATG	[0.201 - 0.242]	T-beg: GCA	[0.141 - 0.181]	P-beg: CTC	[0.189 - 0.229]
S-end: GGG	[0.120 - 0.155]	T-mid: TCA	[0.115 - 0.149]	T-beg: ATG	[0.109 - 0.145]	P-beg: CCC	[0.158 - 0.195]
T-beg: AGG	[0.114 - 0.148]	T-mid: TGC	[0.134 - 0.169]	T-mid: CTC	[0.139 - 0.179]	P-end: CGC	[0.112 - 0.144]
T-beg: GCT	[0.136 - 0.172]	T-end: CTG	[0.149 - 0.186]	T-mid: TGA	[0.112 - 0.148]		
T-mid: CCT	[0.129 - 0.165]	T-end: GTG	[0.116 - 0.150]	T-mid: TCC	[0.145 - 0.186]		
T-end: CTT	[0.126 - 0.162]	T-end: TGC	[0.110 - 0.143]	T-mid: CCT	[0.154 - 0.196]		
T-end: TTC	[0.092 - 0.124]	T-end: TGT	[0.164 - 0.203]	T-end: CTA	[0.127 - 0.166]		
T-end: TCT	[0.054 - 0.080]	T-end: TGG	[0.110 - 0.143]	T-end: TAG	[0.133 - 0.172]		
P-beg: AAA	[0.197 - 0.238]	T-end: GTT	[0.127 - 0.162]	T-end: AGA	[0.159 - 0.200]		
P-beg: ACT	[0.107 - 0.140]	P-beg: GGC	[0.123 - 0.157]	P-beg: ACA	[0.342 - 0.395]		
P-mid: AAG	[0.177 - 0.217]	P-beg: CCC	[0.112 - 0.145]	P-beg: CCA	[0.123 - 0.161]		
P-end: AGC	[0.102 - 0.135]	P-mid: GCG	[0.117 - 0.151]	P-beg: TCA	[0.107 - 0.143]		
		P-end: CGC	[0.189 - 0.229]	P-mid: CAG	[0.262 - 0.312]		
				P-mid: CAA	[0.142 - 0.182]		
				P-end: AGC	[0.217 - 0.264]		
				P-end: ATC	[0.122 - 0.160]		

middle towards the end of the trunk region, it prefers to be combined with Thymine as observed in different mers in various quadrants. As for the promiscuous region, Cytosine prefers to combine with Guanine (*e.g.*, CGC, GCG in Q2) or to be followed by Adenine (*e.g.*, ACA, CAG, CAA, CCA, TCA in Q3).

As for Guanine, it can be placed at the beginning of the seed region if followed by Cytosine as revealed in Q4 (*e.g.*, GCC), or anywhere in the seed region if preceded by Thymine (*e.g.*, TGT, TTG, GTG, ATG), or at the beginning of the trunk region if preceded by Adenine (*e.g.*, AGG), as revealed in Q1. Thymine appears in various places. It prefers to combine with Cytosine in the middle toward the end of the trunk region, as revealed in Q1, Q3, and Q4. For Thymine to be at the beginning towards the middle of the trunk region, it prefers to be followed by Guanine (*e.g.*, TGT, TGG, TGC, GTG), as shown in Q2. Thymine is disfavored in the seed and promiscuous regions, except as auxiliary bases in a few cases.

The aforementioned features and many others can be inferred and summarized, particularly those that have been revealed exclu- sively by the synthetic data. In MSMs, multiple such features are included, such as the TGT mer in the seed region, ACT, CCC, and ACA in the promiscuous region, and many other ones in the trunk region across the different quadrants. These features exist in the testing data HT2 and HT3, but are obscurely observed in the training data HT1. This showcases the direct benefit of the suggested paradigm, where it is possible to discover obscure features that lie in the neighborhood of the prominent ones of the training data. The most confident features in our analysis were ACA and CRC in the promiscuous region ($CI[0.342 - 0.395]$, and $CI[0.189 - 0.229]$), TGT and TGC in the seed region ($CI[0.303 - 0.351]$, and $CI[0.201 - 0.242]$), and ATG in the trunk ($CI[0.201 - 0.242]$).

4. Conclusion

In this paper, we develop a comprehensive paradigm to improve the explainability of deep learning-based models for predicting the efficiency of the gRNA sequence in CRISPR systems. Our approach involves building a generative framework that produces synthetic data that resembles labeled training data and fills in sequence-related gaps. The alignment between the proposed generative framework and the discriminative seq-DeepCpf1 model enhances confidence in the findings and provides explainability for the discriminative method's decisions. We employ two analysis methods to infer and summarize the most prominent features of the synthetic data. The first method is manual histogramming, and the second method utilizes class activation maps. Through these methods, we discovered and highlighted many features, including particularly obscure ones, which were later confirmed in the testing data.

We note that one of the limitations of the proposed system is that it was trained and tested specifically on the CRISPR/Cas12a system. In future work, we aim to extend the applicability of our approach to other CRISPR systems, such as Cas13d, a compact RNA-targeting Cas system. This extension can be easily accomplished by adapting the training data to include sequences from these systems, making the model more comprehensive and generalizable. Additionally, we plan to explore state-of-the-art sequence analysis tools, such as the Transformer paradigm, which is renowned for its superior performance in sequence analysis tasks.

Acknowledgements

This work was supported by the Abu Dhabi Award for Research Excellence under ASPIRE/Advanced Technology Research Council.

Conflict of interest

There are no conflicts to declare.

Supporting Information

Not applicable.

References

[1] Y. Xu, Z. Li, CRISPR-Cas systems: Overview, innovations and applications in human disease research and gene therapy, *Computational and Structural Biotechnology Journal*, 2020, **18**, 2401-2415, doi: 10.1016/j.csbj.2020.08.031.

[2] M. Adli, The CRISPR tool kit for genome editing and beyond, *Nature Communications*, 2018, **9**, 1911, doi: 10.1038/s41467-018-04252-2.

[3] Y. Wu, D. Liang, Y. Wang, M. Bai, W. Tang, S. Bao, Z. Yan, D. Li, J. Li, Correction of a genetic disease in mouse *via* use of CRISPR-Cas9, *Cell Stem Cell*, 2013, **13**, 659-662, doi: 10.1016/j.stem.2013.10.016.

[4] R. J. Platt, S. Chen, Y. Zhou, M. J. Yim, L. Swiech, H. R. Kempton, J. E. Dahlman, O. Parnas, T. M. Eisenhaure, M. Jovanovic, D. B. Graham, S. Jhunjhunwala, M. Heidenreich, R. J. Xavier, R. Langer, D. G. Anderson, N. Hacohen, A. Regev, G. Feng, P. A. Sharp, F. Zhang, CRISPR-Cas9 knockin mice for genome editing and cancer modeling, *Cell*, 2014, **159**, 440-455, doi: 10.1016/j.cell.2014.09.014.

[5] L. Cong, F. A. Ran, D. Cox, S. Lin, R. Barretto, N. Habib, P. D. Hsu, X. Wu, W. Jiang, L. A. Marraffini, F. Zhang, Multiplex genome engineering using CRISPR/Cas systems, *Science*, 2013, **339**, 819-823, doi: 10.1126/science.1231143.

[6] H. Zhu, C. Liang, CRISPR-DT: designing gRNAs for the CRISPR-Cpf1 system with improved target efficiency and specificity, *Bioinformatics*, 2019, **35**, 2783-2789, doi: 10.1093/bioinformatics/bty1061.

[7] A. Alok, D. Sandhya, P. Jogam, V. Rodrigues, K. K. Bhati, H. Sharma, J. Kumar, The Rise of the CRISPR/Cpf1 system for efficient genome editing in plants, *Frontiers in Plant Science*, 2020, **11**, 264, doi: 10.3389/fpls.2020.00264.

[8] G. Zhang, Z. Dai, X. Dai, C-RNNCrispr: Prediction of CRISPR/Cas9 sgRNA activity using convolutional and recurrent neural networks, *Computational and Structural Biotechnology Journal*, 2020, **18**, 344-354, doi: 10.1016/j.csbj.2020.01.013.

[9] H. K. Kim, S. Min, M. Song, S. Jung, J. W. Choi, Y. Kim, S.

Lee, S. Yoon, H. H. Kim, Deep learning improves prediction of CRISPR–Cpf1 guide RNA activity, *Nature Biotechnology*, 2018, **36**, 239-241, doi: 10.1038/nbt.4061.
[10] G. Zhang, X. Dai, CNN-SVR for CRISPR-Cpf1 guide RNA activity prediction with data augmentation, *Proceedings of the 2019 9th International Conference on Bioscience, Biochemistry and Bioinformatics*, Singapore Singapore, ACM, 2019: 43-47, doi: 10.1145/3314367.3314383.
[11] L. Xue, B. Tang, W. Chen, J. Luo, Prediction of CRISPR sgRNA activity using a deep convolutional neural network, *Journal of Chemical Information and Modeling*, 2019, **59**, 615-624, doi: 10.1021/acs.jcim.8b00368.
[12] G. Chuai, H. Ma, J. Yan, M. Chen, N. Hong, D. Xue, C. Zhou, C. Zhu, K. Chen, B. Duan, F. Gu, S. Qu, D. Huang, J. Wei, Q. Liu, DeepCRISPR: optimized CRISPR guide RNA design by deep learning, *Genome Biology*, 2018, **19**, 80, doi: 10.1186/s13059-018-1459-4.
[13] H. K. Kim, G. Yu, J. Park, S. Min, S. Lee, S. Yoon, H. H. Kim, Predicting the efficiency of prime editing guide RNAs in human cells, *Nature Biotechnology*, 2020, **39**, 198-206, doi: 10.1038/s41587-020-0677-y.
[14] X. Cheng, Z. Li, R. Shan, Z. Li, S. Wang, W. Zhao, H. Zhang, L. Chao, J. Peng, T. Fei, W. Li, Modeling CRISPR-Cas13d on-target and off-target effects using machine learning approaches, *Nature Communications*, 2023, **14**, 752, doi: 10.1038/s41467-023-36316-3.
[15] H.-H. Wessels, A. Stirn, A. Méndez-Mancilla, E. J. Kim, S. K. Hart, D. A. Knowles, N. E. Sanjana, Prediction of on-target and off-target activity of CRISPR–Cas13d guide RNAs using deep learning, *Nature Biotechnology*, 2023, **42**, 628-637, doi: 10.1038/s41587-023-01830-8.
[16] Q. Liu, X. Cheng, G. Liu, B. Li, X. Liu, Deep learning improves the ability of sgRNA off-target propensity prediction, *BMC Bioinformatics*, 2020, **21**, 51, doi: 10.1186/s12859-020-3395-z.
[17] J. Lin, K.-C. Wong, Off-target predictions in CRISPR-Cas9 gene editing using deep learning, *Bioinformatics*, 2018, **34**, i656-i663, doi: 10.1093/bioinformatics/bty554.
[18] J. Charlier, R. Nadon, V. Makarenkov, Accurate deep learning off-target prediction with novel sgRNA-DNA sequence encoding in CRISPR-Cas9 gene editing, *Bioinformatics*, 2021, **37**, 2299-2307, doi: 10.1093/bioinformatics/btab112.
[19] W. Xiang, D. Chen, Y. Cui, S. Peng, H-VAE: A Hybrid Variational AutoEncoder with Data Augmentation in Predicting CRISPR/Cas9 Off-target, *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, December 09-12, Houston, TX, USA, 2021, 550-555, doi: 10.1109/BIBM52615.2021.9669570.
[20] V. Konstantakos, A. Nentidis, A. Krithara, G. Paliouras, CRISPR-Cas9 gRNA efficiency prediction: an overview of predictive tools and the role of deep learning, *Nucleic Acids Research*, 2022, **50**, 3616-3637, doi: 10.1093/nar/gkac192.
[21] H. Xu, T. Xiao, C.-H. Chen, W. Li, C. A. Meyer, Q. Wu, D. Wu, L. Cong, F. Zhang, J. S. Liu, M. Brown, X. S. Liu, Sequence determinants of improved CRISPR sgRNA design, *Genome Research*, 2015, **25**, 1147-1157, doi: 10.1101/gr.191452.115.
[22] G. Zhang, T. Zeng, Z. Dai, X. Dai, Prediction of CRISPR/Cas9 single guide RNA cleavage efficiency and specificity by attention-based convolutional neural networks, *Computational and Structural Biotechnology Journal*, 2021, **19**, 1445-1457, doi: 10.1016/j.csbj.2021.03.001.
[23] H. K. Kim, M. Song, J. Lee, In vivo high-throughput profiling of CRISPR–Cpf1 activity, *Nature Methods*, 2017, **14**, 153-159.
[24] A. R. O'Brien, G. Burgio, D. C. Bauer, Domain-specific introduction to machine learning terminology, pitfalls and opportunities in CRISPR-based gene editing, *Briefings in Bioinformatics*, 2021, **22**, 308-314, doi: 10.1093/bib/bbz145.
[25] X. Wang, X. Wang, R. K. Varma, L. Beauchamp, S. Magdaleno, T. J. Sendera, Selection of hyperfunctional siRNAs with improved potency and specificity, *Nucleic Acids Research*, 2009, **37**, e152, doi: 10.1093/nar/gkp864.
[26] N. Wong, W. Liu, X. Wang, WU-CRISPR: characteristics of functional guide RNAs for the CRISPR/Cas9 system, *Genome Biology*, 2015, **16**, 218, doi: 10.1186/s13059-015-0784-0.
[27] D. P. Kingma, M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114*, 2013, doi: 10.48550/arXiv.1312.6114.
[28] K. Sohn, X. Yan, H. Lee, Learning structured output representation using deep conditional generative models, *Advances in Neural Information Processing Systems*, 2015, **28**, 3483-3491.
[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 27-30, Las Vegas, NV, USA, IEEE, 2016, 2921-2929, doi: 10.1109/CVPR.2016.319.
[30] Z. C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery, *Queue, ACM*, 2018, **16**, 31-57, doi: 10.1145/3236386.3241340.
[31] A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian, J. Vertesi, Fairness and abstraction in sociotechnical systems, *Proceedings of the Conference on Fairness, Accountability, and Transparency*, Atlanta GA USA, ACM, 2019, 59-68, doi: 10.1145/3287560.3287598.
[32] S. Q. Tsai, Z. Zheng, N. T. Nguyen, M. Liebers, V. V. Topkar, V. Thapar, N. Wyvekens, C. Khayter, A. J. Iafrate, L. P. Le, M. J. Aryee, J. K. Joung, GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases, *Nature Biotechnology*, 2014, **33**, 187-197, doi: 10.1038/nbt.3117.
.