Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Artificial Intelligence

sgRNA Design for CRISPR/Cas9 Gene Editing Technology with Deep Learning

Bachelor's Thesis

Author
Dávid Nagy

Advisors
Bálint Gyires-Tóth, PhD
Dániel Unyi

December 6, 2024

Contents

Kivonat

A mélytanulás a gépi tanulási technikák egyik ága a mesterséges intelligencia (MI) kutatási területén belül. Ez egy robosztus és hatékony megközelítést kínál hatalmas mennyiségű adat elemzésére és modellezésére, neurális hálózatok és korszerű számítási képességek felhasználásával. Ez a módszertan eredményesen alkalmazható összetett mintafelismerést igénylő feladatokban, különösen, a nagyméretű adathalmazokkal való munka során.

A CRISPR/Cas9 egy genomszerkesztési csúcstechnológia, amely sokoldalúsága és más DNS-szerkesztési technikákkal szembeni könnyű használhatósága miatt jelentős figyelmet kapott az orvostudományi kutatásokban. Számos területen ígéretes lehetőségeket rejt magában, többek között a mezőgazdaságban és az egészségügyben is. A rendszer kulcsfontosságú eleme az single guide RNS (sgRNS), amely a Cas fehérjéket meghatározott DNS-szekvenciákhoz irányítja. Ezek a fehérjék ezután pontosan elvágják a kettősszálú DNS-t, lehetővé téve a célzott génmódosításokat. A CRISPR-rendszerek sikeres alkalmazásához kulcsfontosságú feladat olyan sgRNS-ek tervezése, amelyek maximalizálják a hasítás hatékonyságát a célhelyen és minimalizálják a célhelyen kívüli hatást. Ennek az egyensúlynak az elérése elengedhetetlen a megbízható génszerkesztéshez.

A dolgozat célja, hogy feltárja a mély neurális hálózatok alkalmazásának lehetőségeit a hatékony sgRNS generálásban. A hangsúly három neurális hálózaton van: egy a hasítás célhelyen vett hatékonyságának előrejelzésére, egy a célhelyen kívüli hatás előrejelzésére és egy az sgRNS-ek generálására. A generált sgRNS-eket a két előrejelző hálózat segítségével értékelem. A generátorhálózathoz egyéni veszteségfüggvényt tervezek azért, hogy a generált sgRNS-eloszlást a hatékonyabbak felé toljam el. E megközelítéssel az a célom, hogy hozzájáruljak a CRISPR/Cas9 technológia különböző területeken való gyakorlati alkalmazásához.

Abstract

Deep learning is a subset of machine learning within the field of artificial intelligence. It offers a powerful and efficient approach for analyzing and modeling vast amounts of data, using neural networks and advanced computational capabilities. This methodology excels in tasks requiring complex pattern recognition, particularly when working with large-scale datasets.

CRISPR/Cas9 is a cutting-edge genome editing technology that has gained significant attention in medical research due to its versatility and ease of use compared to other DNA editing techniques. It holds promise across various fields, including agriculture and healthcare. A key element of this system is the single guide RNA (sgRNA), which directs the Cas proteins to specific DNA sequences. These proteins then precisely cut the double-stranded DNA, enabling targeted gene modifications. A vital task for the successful application of CRISPR systems is to design sgRNAs that maximize on-target efficiency while minimizing off-target effects. Achieving this balance is essential for reliable gene editing.

The goal of this work is to explore the possibilities of introducing deep neural networks to enhance sgRNA generation. The focus is on three neural networks: one for on-target efficiency prediction, one for off-target profile prediction and one for generating sgRNAs. I evaluate the generated sgRNAs using the two predictor networks. I design a custom loss function for the generator network to shift the generated sgRNA distribution to more efficient ones. With this approach, I aim to contribute to the practical application of CRISPR/Cas9 technology in various fields.

Chapter 1

Introduction

Gene editing has rapidly evolved into one of the most promising fields in biomedical research, opening new avenues for precise genetic modifications with vast implications for treating genetic disorders, cancers, and infectious diseases [23]. The advent of CRISPR/Cas9 technology has revolutionized this field by providing a highly accurate, programmable tool for targeting specific DNA sequences [59]. However, designing these tools remained challenging, highlighting the need for robust tools [12]. Hence, this potentially raises the possibility of taking advantage of the recent advancements in artificial intelligence combined with machines' computational power for addressing these design issues [19].

CRISPR, short for Clustered Regularly Interspaced Short Palindromic Repeats, refers to specific DNA sequences in bacterial genomes that store genetic information from viral attackers. The CRISPR-associated protein 9 (Cas9), guided by RNA sequences, uses this stored information to identify and cut DNA at specific sites [36]. The CRISPR/Cas9 system functions through a carefully designed single-guide RNA (sgRNA) that directs Cas9 to a specific DNA target, where it creates a double-strand break [36]. Regarding this break, on-target efficacy refers to the precision with which the system edits the intended DNA sequence, while off-target effects occur when CRISPR/Cas9 unintentionally cuts similar but non-target DNA sequences, potentially leading to undesired mutations and side effects [26]. However, designing sgRNA that maximizes on-target efficacy, while minimizing off-target effects has remained a vital challenge [37].

Artificial intelligence (AI) is transforming the status quo day by day. Deep learning (DL) is an incredibly powerful tool within this field, which deals with algorithms inspired by the structure and function of the human brain, known as artificial neural networks (ANNs) [42]. DL, by nature, enables machines to learn complex patterns from large datasets. Due to this fact, it has revolutionized fields like computer vision, natural language processing, and speech recognition, achieving remarkable performance in tasks that were once considered very challenging [72]. ANNs used in deep learning have multiple layers - hence the name "deep" - to analyze data and make predictions, making it suitable for complex biological tasks that involve large amounts of data, such as genomics and gene

editing [46]. Generative AI (GAI) models, a subset within deep learning, are designed to create or "generate" new data samples [5]. Among GAI models, variational autoencoders (VAEs) generate new data that resemble the input data. VAEs consist of two primary components: an encoder that learns to compress data into a latent space, and a decoder that reconstructs the original data from this compressed form. Unlike traditional autoencoders, VAEs introduce a probabilistic component by encoding data as a distribution rather than a fixed point, allowing for more flexible and robust data generation [22].

For sgRNA design, VAEs might be trained on vast libraries of known sgRNA sequences, trying to learn to generate new sequences with a potential for higher specificity and binding efficacy. By taking advantage of a latent space that captures key features of functional sgRNAs, VAEs can help in designing optimized sgRNA sequences that would be laborious or impractical to identify through experimental screening alone [48]. The need for designing efficient sgRNAs together with the potential of VAEs in these type of challenges is the underlying motivation of this work. I aim to give a deep learning based solution using the architecture of VAEs to generate sgRNAs while maximizing their on-target efficiency and minimizing their off-target effects. By leveraging deep learning models for this task, the accuracy, efficiency, and safety of gene-editing applications can be enhanced. This approach holds promise for personalized therapeutics, where gene-editing strategies can be tailored to an individual's genetic makeup, offering a new frontier in treating genetic disorders, infectious diseases, and cancers [7].

The second chapter presents the theoretical background of CRISPR/Cas9, the architecture of VAEs, the corresponding deep learning techniques and previous works in predicting a given sgRNA's on-target and off-target efficacy. In the third chapter, the research objectives are specified. In the fourth chapter, the methods and the implementation of the solution are demonstrated. The fifth chapter is dedicated to displaying the results. The sixth chapter records the ethical position of the author regarding the future usage of this work. Finally, in the last chapter, a summary and future research directions are pointed out.

The source code of the discussed solution is publicly available at GitHub [1]

[1] `https://github.com/flash4242/VAE_sgRNA_design.git`.

Chapter 2

Theoretical background

2.1 The CRISPR/Cas9 technology

2.1.1 Discovery and overview

CRISPR genome editing is based on a natural immune process used by bacteria and archaea to defend themselves against viruses and plasmids [36]. CRISPR sequences, which were first identified in 1987 as short genomic DNA sequences in [34], are now known to be part of an adaptive defense mechanism that also involves Cas (CRISPR-associated) enzymes. There are four bases in DNA, namely adenine (A), cytosine (C), guanine (G), and thymine (T). In DNA, adenine pairs with thymine meaning adenine is complementary with thymine (and visa versa). Similarly, cytosine pairs with guanine meaning cytosine is complementary with guanine (and visa versa) [33]. So, if a DNA sequence is given: ATTATTGCGC, its complementary sequence is: TAATAACGCG. CRISPR systems can recognize and cleave complementary DNA sequences, allowing bacteria to remember and destroy viral invaders [36].

Cas enzymes are part of a bacterial immune system that incorporates short, viral DNA sequences into the bacterial genome. This is a complicated process that is not entirely understood [29]. What is known is that these viral sequences are found at regular intervals, short distances from one another in the bacterial genome. The bacterial DNA in between these sequences has palindromic repeating patterns, hence the name, clustered regularly interspaced short palindromic repeats. The incorporated viral DNA sequences can be transcribed into guide RNA (gRNA) when needed - that is, if the same kind of virus tries to infect the bacterium again, the CRISPR system can cut the invading viral DNA through use of the gRNA and Cas enzyme [36]. This last step of the bacterial immune process, when the gRNA is combined with Cas and cleaves the target DNA, is what has been adopted for genome editing in laboratories. This fundamental discovery was pivotal, and subsequent work led by Doudna and Charpentier adapted CRISPR/Cas9 for targeted gene editing, a breakthrough published in 2012 [36].

2.1.2 Working mechanism

As a summary in [36]:

"A two-RNA structure directs an endonuclease to cleave target DNA."

CRISPR/Cas systems rely on small RNAs for sequence-specific detection and silencing of foreign nucleic acids. These systems consist of cas genes organized in operon(s) and a CRISPR array comprising unique genome-targeting sequences (called spacers) interspersed with identical repeats [69, 8, 65]. CRISPR/Cas mediated immunity occurs in three steps. In the adaptive phase, bacteria and archaea carrying one or more CRISPR loci respond to viral and plasmid attack by integrating short fragments of foreign sequence (protospacers) into the host chromosome at the proximal end of the CRISPR array [69, 8, 65]. In the next two steps, the expression and interference phases, transcription of the repeat-spacer element into precursor CRISPR RNA (pre-crRNA) molecules followed by enzymatic cleavage yields the short CRISPR RNAs (crRNAs) that can base pair with complementary protospacer sequences of invading viral or plasmid targets [20, 11, 27]. Target recognition by crRNAs directs the silencing of the foreign sequences through Cas proteins that operates in complex with the crRNAs [9, 47].

Three types of CRISPR/Cas are known [51]. The Type I and III systems have some overarching features in common: specialized Cas endonucleases process the pre-crRNAs, and once mature, each crRNA assembles into a large multi-Cas protein complex being able to recognize and cleave nucleic acids complementary to the crRNA. In contrast, Type II systems process pre-crRNAs by a different mechanism in which a trans-activating crRNA (tracrRNA) complementary to the repeat sequences in pre-crRNA triggers processing by the double-stranded RNA-specific ribonuclease RNase III in the presence of the Cas9 protein [20, 25]. Cas9 is thought to be the sole protein responsible for crRNAguided silencing of foreign DNA [6].

It's demonstrated in [36], that in Type II systems, Cas9 proteins form a family of enzymes that need a base-paired structure formed between the activating tracrRNA and the targeting crRNA to cleave target double-stranded (ds) DNA. Site-specific cleavage occurs at locations determined by both base-pairing complementarity between the crRNA and the target protospacer DNA and a short motif (referred to as the protospacer adjacent motif, or PAM) juxtaposed to the complementary region in the target DNA. Doudna et al. [36] showed that the Cas9 endonuclease family can be programmed with single RNA molecules to cleave specific DNA sites. They indicated the chance that the features needed for site-specific Cas9-catalyzed DNA cleavage could be captured in a single chimeric RNA . Chimeric RNA refers to hybrid transcripts that combine exons from two different genes [57]. In their study, they linked crRNA and tracrRNA together, creating the crRNA-tracrRNA single chimeric RNA, also called the single guide RNA. They point out that, despite the fact that tracrRNA:crRNA target selection process works efficiently in nature, the possibility of a single RNA-guided Cas9 is intruiging due to its potential utility for programmed DNA cleavage and genome editing. The natural construction is shown in the top of Figure 2.1, and the bottom presents the combined chimera RNA.

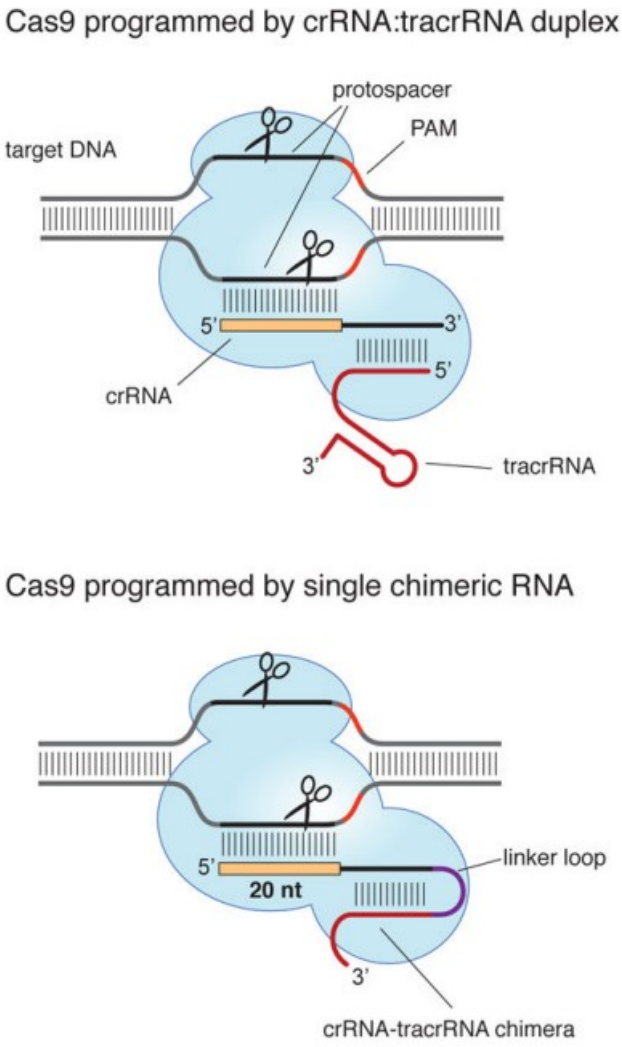

Figure 2.1: Top: In Type II CRISPR/Cas systems, Cas9 is guided by a two-RNA structure formed by activating tracrRNA and targeting crRNA to cleave site-specifically target dsDNA. Bottom: A chimeric RNA generated by combining crRNA with tracrRNA. Source: [36]

2.1.3 Applications

Shortly after the CRISPR/Cas9 mechanism was discovered, already several applications came to light, having a huge impact on the world in many areas including medicine, agriculture, and biotechnology [4]. As this technology advances, it holds promise for treating and curing diseases, develop more nutritious crops, and eradicating infectious diseases [31]. Some of the recent CRISPR/Cas9 applications are highlighted below.

First, it shows significant potential in gene therapy. More than 6000 genetic diseases have been known so far. However, most of the disorders lack effective treatment methods [35]. Gene therapy is the process of replacing the defective gene with foreign DNA and editing the mutated gene at its original location. From 1998 to August 2019, 22 gene therapies including the innovative CRISPR/Cas9 have been approved for the treatment of human diseases [50]. Since it's discovery, it has been explored for curing genetic diseases like sickle cell disease (SCD), β-thalassemia, cystic fibrosis, and muscular dystrophy [54]. Regarding Duchenne muscular dystrophy (DMD), which is caused by a mutation in the dystrophin gene and signalized by muscle weakness, it has been successfully corrected by CRISPR/Cas9 in patient-induced pluripotent stem cells [44].

Second, CRISPR/Cas9 has an important therapeutic role of treating infectious diseases caused by microorganism [64]. One area under focus is treating HIV, the virus that leads to AIDS. In May 2017, a study showed that HIV-1 replication can be shut down entirely and the virus eliminated from infected cells through excision of HIV-1 genome using CRISPR/Cas9 in animal models [49]. Furthermore, the first CRISPR-based therapy in the human trial was conducted to treat patients with refractory lung cancer. Researchers first extracted T-cells from three patients' blood and engineered them in the lab through CRISPR/Cas9 to delete genes that would hinder fighting cancer cells. Then, they infused the modified T-cells back into the patients. The modified T-cells can target specific antigens and kill cancer cells. Ultimately, no side effects were observed and engineered T-cells can be detected up to 9 months postinfusion [63].

Third, this technology influences agriculture too. Because the world population continues to grow, may arise the risk of shortage in agricultural resources. Hence, new technologies for increasing and improving natural food production are needed [4]. CRISPR/Cas9 is a valuable tool in this field, helping address food security challenges by genetically modifying crops to improve their nutritional content, extend shelf life, increase drought tolerance, and boost disease resistance [31]. CRISPR contributes to solving global food issues in three main ways: restoring food supplies, enabling crops to thrive in challenging environments, and enhancing plant health overall [2].

2.1.4 Challenges

Despite its potential, CRISPR/Cas9 technology faces significant challenges that limit its clinical application. Key obstacles include immunogenicity, the lack of a safe and effective delivery system, off-target effects, and ethical concerns [41].

The designed single guide RNA (sgRNA, defined in 2.1.2) might mismatch to the non-target DNA and can result in nonspecific, unintended genetic modification, which is called the off-target effect [16]. The CRISPR/Cas9 target efficiency is formed by the 20-nucleotide sequences of sgRNA and the PAM sequences adjacent to the target genome. It has been shown that more than three mismatches between the target sequence and the 20-nucleotide sgRNA can result in off-target effects [74]. The off-target effect may cause harmful events such as sequence mutation, deletion, rearrangement, immune response, and oncogene activation, which limits the application of the CRISPR/Cas9 editing system for therapeutic purposes [13]. To mitigate the possibility of the mentioned off-target effect, several strategies have been developed, such as optimization of sgRNA, modification of Cas9 nuclease, usage of other Cas-variants, and the use of anti-CRISPR proteins [4]. Selecting and designing an suitable sgRNA for the targeted DNA sequence is an important first step to reduce the off-target effect [52].

2.2 Variational autoencoders

Variational Autoencoders (VAEs) were introduced in 2013 by Diederik P. Kingma and Max Welling in their seminal paper, Auto-Encoding Variational Bayes [39]. VAEs are a class of deep generative models designed to learn probabilistic representations of data by combining principles from deep learning and Bayesian inference. The core idea of a VAE is to encode high-dimensional input data into a lower-dimensional latent space while preserving the underlying probabilistic structure of the data. By doing so, VAEs facilitate tasks like data generation, interpolation, and representation learning in a principled probabilistic framework .

At the heart of a VAE is the concept of variational inference, used to approximate the intractable posterior distribution of the latent variables. Kingma and Welling introduced the reparameterization trick to make backpropagation through stochastic variables feasible, enabling end-to-end training of the model using gradient-based optimization. This innovation not only made VAEs computationally efficient but also laid the foundation for subsequent developments in generative modeling. Since its introduction, the VAE framework has been widely adopted and extended, finding applications in fields like image generation, anomaly detection, and also generating novel drug molecules [53].

2.2.1 The latent space

To understand VAEs, it's crucial to grasp the concept of latent space, which refers to the collective latent variables associated with a given set of input data. Latent variables represent underlying, unobservable factors that influence the way data is structured or distributed [32]. A helpful analogy for latent variables is a bridge equipped with a weight sensor that records the weight of each vehicle crossing it. While various vehicles—such as convertibles, sedans, vans, and trucks—cross the bridge, and each has a distinct weight,

no camera is present to identify the vehicle type. However, we know that the vehicle type greatly affects its weight [32]. This scenario can be represented with two variables: x, the observable variable for vehicle weight, and z, the latent variable representing vehicle type. The primary goal in training any variational autoencoder is to learn how to effectively model the latent space of a given input set [39].

VAEs represent latent space by performing dimensionality reduction. This means the compression of data into a lower-dimensional space that retains the most meaningful information from the original input [56]. In machine learning, dimensions correspond to data features rather than physical space. For instance, a 28x28-pixel black-and-white image from the MNIST dataset (introduced in [43]) can be represented by a 784-dimensional vector, with each dimension indicating a pixel value between 0 (black) and 1 (white). In a color image, this would expand to a 2352-dimensional vector, where each pixel has three dimensions for red, green, and blue (RGB) values. Yet, not all these dimensions carry essential information. Since the actual digit occupies only a portion of the image, much of the input space represents background noise. Reducing data to only the dimensions with relevant information—the latent space—can enhance the accuracy, efficiency, and performance of various ML tasks and algorithms [56].

2.2.2 Architecture

Although different types of models alter certain parts of their architecture to better suit specific goals and data types, all variational autoencoders share three key structural elements [22]:

- The encoder captures latent variables from input data x and outputs them as a vector representing the latent space z. In a standard autoencoder, each layer of the encoder has fewer nodes than the one before it, progressively compressing the data as it moves through the network.
- The bottleneck, or "code," serves as the output layer of the encoder and the input layer of the decoder. This layer holds the latent space: a compact, lower-dimensional representation of the input data. A sufficient bottleneck is essential to prevent the decoder from merely copying or memorizing the input, which would meet its training objective but hinder the VAE's ability to learn meaningful patterns.
- The decoder then uses this latent representation to reconstruct the input by effectively reversing the encoder, with each successive layer containing more nodes.

This architecture is represented in Figure 2.2.

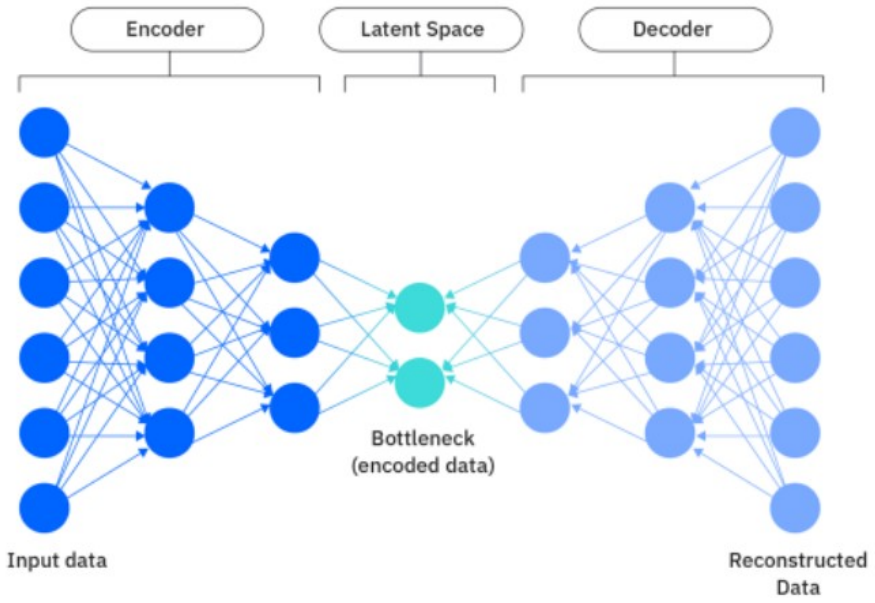

Figure 2.2: Visual representation of the architecture of a variational autoencoder neural network. Source: [32]

While many autoencoders use standard multilayer perceptrons (MLPs) as their encoder and decoder networks, they are not limited to any specific neural network type [58]. For instance, VAEs applied to computer vision tasks often use convolutional neural networks (CNNs) and are called convolutional autoencoders [55]. Autoencoders based on transformer architecture have also been successfully applied across fields, including computer vision and music [14]. A key advantage of autoencoders over traditional dimensionality reduction methods like principal component analysis (PCA) is their ability to model nonlinear relationships between variables [39]. Thus, autoencoder nodes commonly employ nonlinear activation functions. In numerous autoencoder applications, the decoder's role is primarily to optimize the encoder and is often discarded after training. However, in variational autoencoders, the decoder is retained for generating new data samples [32].

2.2.3 Probabilistic modeling in variational autoencoders

What sets VAEs apart from other autoencoders (the discovery of autoencoders is explained in papers listed in [70]) is their distinct approach to encoding latent space and the unique applications of their probabilistic encoding. While most autoencoders are deterministic, encoding a single fixed vector of discrete latent variables, VAEs are probabilistic models. Instead of encoding latent variables as a fixed value, z, VAEs represent these variables as a continuous probability distribution, $p(z)$ [39]. In Bayesian statistics ([66]), this learned range for each latent variable is known as the prior distribution. During variational inference, which is the generative process for creating new data samples,

this prior distribution helps calculate the posterior distribution, $p(z|x)$, the probability of observable variables x given a particular value for latent variable z. VAEs encode two separate latent vectors for each attribute in the data: one representing means, μ, and another for standard deviations, σ. Together, these vectors capture the range of values for each latent variable and the expected variation within each range. By sampling randomly from within these ranges, VAEs can generate new data points that, while novel, closely resemble the training data [39].

To explore how VAEs achieve this, the following elements are examined below:

- Reconstruction loss
- Kullback-Leibler (KL) divergence
- Evidence lower bound (ELBO)
- The reparameterization trick

VAEs use reconstruction loss, also called reconstruction error, as a main loss function during training [39]. This error quantifies the difference between the original input and its reconstructed output from the decoder. Various functions, such as cross-entropy loss or mean-squared error (MSE), can be applied as the reconstruction loss [10]. Minimizing reconstruction error through gradient descent (a method through which machines learn [3]) across the parameters of both the encoder and decoder progressively refines the model's weights, resulting in a more meaningful latent representation and better reconstructions [39]. Mathematically, this is achieved by optimizing $p_\theta(z|x)$, where θ are the model parameters that enable accurate reconstruction of input x from latent variable z. While reconstruction loss suffices for standard autoencoders, which focus solely on learning a compact, accurate representation, a variational autoencoder's objective extends beyond merely reconstructing the input. Instead, it aims to generate *new* samples *similar* to the original data, which requires an additional optimization term, named Kullback-Leibler divergence [22].

Regarding variational inference, meaning that a trained model generates new samples, relying on reconstruction loss alone may result in a latent space that is prone to overfitting the training data, limiting its generalizability to novel samples [22]. To counter this, VAEs add a regularization term: Kullback-Leibler (KL) divergence. For generating images, the decoder must sample from latent space. Sampling from the specific latent representations of the training data would recreate those inputs, but generating new images requires sampling freely across the entire latent space. To enable this, the latent space must have two key properties [22]:

- Continuity: Nearby points in latent space should decode to similar outputs.
- Completeness: Any point within the latent space should yield a meaningful result when decoded.

One straightforward way to encourage both continuity and completeness is by shaping the latent space to follow a Gaussian, or standard normal, distribution. However, minimizing reconstruction loss alone does not encourage the model to structure the latent space this way, as the "in-between" space is irrelevant to accurate reconstructions of the original inputs. This is where KL divergence serves its role [22]. KL divergence measures the difference between two probability distributions. Minimizing the KL divergence between the learned latent distribution and a Gaussian distribution ensures that latent variables follow a normal distribution. This supports smooth interpolation across the points of the latent space, enabling the model to generate new samples [32]. The effect of reconstruction loss and the KL divergence on the model's latent space is shown in Figure 2.3. When both applied, each digit's distribution resembles a Gaussian distribution.

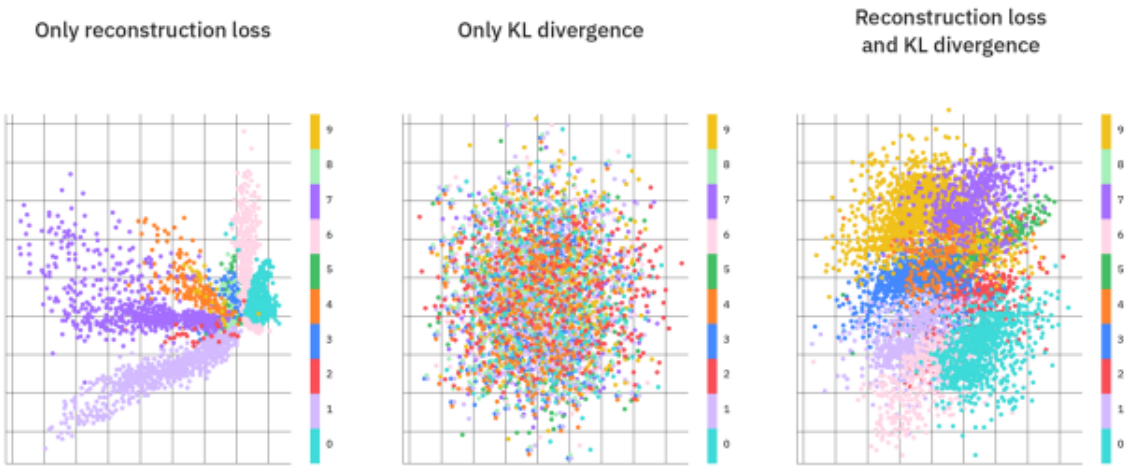

Figure 2.3: Examples of how reconstruction loss and KL divergence affect the modeling of latent space for handwritten digits of 0-9 from the MNIST dataset. Source: [32]

An obstacle in applying KL divergence to variational inference is that the equation's denominator is intractable, making direct computation theoretically infinite. To bypass this issue and combine both essential loss functions, VAEs approximate the minimization of KL divergence by maximizing the evidence lower bound (ELBO) instead (called simply "lower bound" in [39]). In statistical terms, the "evidence" in ELBO, denoted as $p(x)$, represents the observable input data ([24]) that the VAE aims to reconstruct. This input data serves as "evidence" for the latent variables learned by the VAE. The "lower bound" signifies a worst-case estimate for the log-likelihood of a distribution, implying the actual log-likelihood could be higher than the ELBO [39]. Regarding VAEs, the evidence lower bound means the worst-case estimate of the likelihood that a specific posterior dis-

tribution—in other words, a specific output of the model, conditioned by both the KL divergence loss term and the reconstruction loss term—fits the "evidence" of the training data. Thus, training a model for variational inference can be referred to in terms of maximizing the ELBO [39].

As mentioned before, the goal of variational inference is to output new data in the form of random variations of the training data. It may seem relatively straightforward: use a function f that selects a random value for the latent variable z, which can then be used by the decoder to generate an approximate reconstruction of x [32]. However, by definition, a vector of random values has no derivative, thus gradient methods can't be used to train a model which outputs such a random vector. Consequently, it is not possible to optimize a model using backpropagation, as this is a gradient-based algorithm. This would mean that a neural network that uses the preceding random sampling process cannot learn the optimal parameters to do its task [32]. To resolve this issue, VAEs use the reparameterization trick [39]. The reparameterization trick introduces a new parameter, ϵ, which is a random value selected from the normal distribution between 0 and 1. Then, it reparameterizes the latent variable z as

$$z = \mu + \epsilon\sigma \tag{2.1}$$

In other words, it chooses a value for the latent variable z by starting with the mean of that variable (represented by μ) and shifting it by a random multiple (represented by ϵ) of a standard deviation (σ). Modified with that specific value of z, the decoder outputs a new sample [32]. Because the random value ϵ is not derived from and has no relation to the VAE model's parameters, it can be ignored during backpropagation. The model is updated through some form of gradient descent—most often through Adam ([40]), a gradient-based optimization algorithm - also developed by Kingma — to maximize the ELBO.

2.3 Previous works and additional deep learning concepts

2.3.1 Previous works

In [73], the researchers developed two attention-based convolutional neural network (CNN, for further reading see [45]) frameworks, namely CRISPR-ONT and CRISPR-OFFT, for CRISPR/Cas9 sgRNA efficiency (meaning on-target efficiency) and specificity (meaning off-target effect) prediction, respectively. Both algorithms presented in their paper use CNNs frameworks to extract the contextual sequence features and have built-in attention modules to focus on the specific part of the input to help extract interpretable Cas9 binding sgRNA patterns [73]. Comprehensive tests on public datasets showed that CRISPR-ONT and CRISPR-OFFT are superior to state-of-the-art models demonstrated

in [73]. Moreover, these models are interpretable and could be used to examine mammalian genomes as stated in [60], which could be of great relevance in curing mammalian diseases.

A drawback of these models is that they are trained only on two human cell lines (namely hek293t and k562), so they do not provide a broad perspective into the human cells, as there are more than a thousand cell lines reported by [18]. The usage of CRISPR-ONT and CRISPR-OFFT models is presented in subsection 4.2.1.

Another sgRNA on-target and off-target effect prediction models are presented in [71], namely CRISPRon and CRISPRoff. This solution uses a dataset combined from multiple sources of biological data (such as genomic, transcriptomic, and epigenomic data) which enhances the model's predictive power and generalizability achieving 0.91 test Spearman score with their model CRISPRon. This result is state-of-the-art as showcased in the overview of [60].

A limitation to this study is that it was trained only on hek293t cell line, which is even less than the cell lines investigated in [73]. Also, as the researchers point out in [71] CRISPRon does not capture larger genetic changes, like deletions or rearrangements, that can arise during CRISPR editing.

However, none of these models include sgRNA generation. No other works were found that address the issue of generating sgRNAs, predicting their on-target and off-target effects and then optimizing these sgRNA sequences. Although, designing such a generative model would be beneficial as it could potentially produce efficient sgRNAs automatically, which could be then used in all of the CRISPR application fields.

2.3.2 Long short-term memory and dropout

The solution presented in Chapter 4 uses long short-term memory (LSTM) units as a building block of the VAE and a technique called dropout to prevent overfitting [62].

LSTMs were used because of the early stages of incremental model development, although in future work it's worth experimenting with transformer based architectures (introduced in [67]), as they hold promise in genome data analysis [15]. LSTM networks, a specific type of recurrent neural network (RNN), were designed to address limitations of traditional RNNs, particularly the vanishing and exploding gradient problems that make learning long-term dependencies challenging. LSTMs have become fundamental for processing sequential data, finding applications in areas such as speech recognition, natural language processing, and time series forecasting ([30]). The architecture of an LSTM unit consists of three key gates—input, forget, and output gates—each of which plays a critical role in modulating the flow of information.

- Forget gate: This gate controls what information from the previous cell state should be retained or discarded, allowing the network to clear irrelevant data over time

- Input gate: Responsible for determining which new information from the current input should be added to the cell state.
- Output gate: Regulates which information from the current cell state is passed on to the next hidden state.

These gates work together to maintain a "memory cell," enabling the network to preserve information over extended sequences, in contrast to traditional RNNs that can suffer from information loss as sequences grow longer ([**?**]). LSTMs have been applied in CRISPR-Cas9 target site prediction, where they analyze DNA sequences to predict on-target efficacy and off-target effects, significantly improving prediction accuracy compared to conventional methods ([38]) [**?**]

Deep neural networks with a large number of parameters are highly effective machine learning models, yet they are prone to overfitting. Additionally, large networks are computationally intensive, which limits the feasibility of reducing overfitting by averaging predictions from multiple large models at test time. Dropout is a technique designed to tackle this issue. The core idea of dropout is to randomly remove units (and their associated connections) from the neural network during training, which discourages excessive co-adaptation of units. This approach effectively trains an exponential number of "thinned" networks by sampling different subsets of the full network. At test time, averaging the predictions of all these thinned networks can be closely approximated by using the full network with reduced weights, significantly cutting down overfitting and providing substantial improvements compared to other regularization techniques. Dropout has been shown to enhance neural network performance across supervised learning tasks, including vision, speech recognition, document classification, and computational biology, achieving state-of-the-art results on numerous benchmark datasets [62].

Chapter 3

Objectives

While there are manual ways that aim to design sgRNA that maximizes on-target efficacy and minimizes off-target effects [21], the advancements of deep learning have not been fully utilized yet in this field. Hence, designing sgRNA with machine learning is a prospective research direction to improve manual results and provide an automated process of generating new, efficient sgRNAs.

Based on the lack of literature in the area mentioned in subsection 2.3.1, the goal of my thesis work is to build two neural networks: one to predict a given sgRNA's on-target efficacy, and one to predict its off-target effects. In addition, I also aim to construct a baseline VAE, which uses only the standard VAE loss (KL divergence and reconstruction error). Moreover, this baseline model will be enhanced by incorporating the predictions of the prediction networks in its loss function in order to produce novel, optimized sgRNAs.

A dataset from the DeepCRISPR study [17] will be used and preprocessed. The prediction networks will be trained on labelled data, implementing a supervised learning method. The accuracy of these networks should be evaluated with the common metrics used in this field, like Spearman's rank correlation coefficient [17], comparing my results to the best-performing models available. I also calculate Mean Squared Error (MSE) to provide further insights into the models' capabilities.

VAEs will be trained in an unsupervised manner, using only sgRNAs from the mentioned dataset. To measure the models' performance, I provide histograms demonstrating the efficacy distribution of the generated sgRNAs.

The following chapters will explain the exact methods used to accomplish the abovementioned tasks. First, they introduce the dataset, then present the models' architecture and training details, and finally, showcase the results.

Chapter 4

Methods and implementation

4.1 Dataset

Incorporating a dataset as large as possible usually benefits deep learning models, as it can enhance their robustness and generalization ability [1]. In [17] the labeled sgRNA dataset contains $\sim$ 0.2 million sgRNAs with known knockout efficacies. This dataset was generated from $\sim$ 15,000 sgRNAs across 1071 genes with known knockout efficacies.

For the current work, a subset of the dataset given in [17] was used, particularly, from the section "Additional file 5". This file contains four sheets, each containing data from one of the four cell lines (hct116, hl60, hela and hek293t). An extract of this data set is shown in Table 4.1. The column "sgRNA" contains sequences that are 23 bases long, where each base is one of the bases found in DNA (adenine (A), cytosine (C), guanine (G), and thymine (T)). The column "Normalized efficacy" includes values ranging from 0 to 1, a higher value meaning a more efficient sgRNA.

Chromosome	Start	End	Strand	sgRNA	Normalized efficacy
chr3	53916065	53916087	-	GAAGGGCGGCGAGAAGGAGAAGG	0.286080189
chr3	53916079	53916101	-	GAGAACGGAAAGGAGAAGGGCGG	0.400080189
chr3	53916094	53916116	-	GAGAAGGGTGATACGGAGAACGG	0.282832665

Table 4.1: Extract of the DeepCRISPR dataset. The value pairs from the columns named "sgRNA" and "Normalized efficacy" (marked with red rectangle in the figure) were used for on-target model training, validation and testing.

All models were trained and evaluated using five variations of this dataset: four were based on individual cell line data, each treated separately, and the fifth combined data from all cell lines into a single dataset.

4.1.1 Application specific preprocessing

For training, validation and testing of the on-target prediction model, only the columns named "sgRNA" and "Normalized efficacy" were kept. 70% of these datasets were used for training and 15-15% for validation and testing. The preprocessing involves encoding each sgRNA sequence as a one-hot matrix. Each base (A, T, G, C) is mapped to a unique binary vector:

$$\text{A} \rightarrow \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \quad \text{C} \rightarrow \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \quad \text{T} \rightarrow \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \quad \text{G} \rightarrow \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

Each sequence is transformed into a matrix of shape (4, 23), where:

- Rows represent nucleotide categories (A, C, T, G).
- Columns correspond to the nucleotide's position in the sequence.

This representation preserves the sequential order of nucleotides while ensuring compatibility with the input expectations of convolutional layers, which is the first layer type in the on-target prediction model.

In case of the VAEs, only the "sgRNA" column was used because the objective of the model is to generate new sgRNAs. This list of sgRNAs was further divided into training and validation sets, preserving 20% of the data for validation, opening up the possibility of tracking the model's generalization ability by means of the validation loss. Further preprocessing of 23-base sgRNA sequences, involves converting nucleotide strings into numerical representations suitable for neural networks. The sgRNA sequences are encoded into integers using a character-to-integer mapping ($'A' : 0, 'C' : 1, 'G' : 2, 'T' : 3$). Each nucleotide is represented by a unique integer. This step transforms the data into a format that is usable by deep learning models. This technique preserves the sequential order of nucleotides, which is crucial for capturing dependencies within the sequence [75]. The encoded sequences are passed through an embedding layer in the VAE. This layer projects the input sequences into a dense, continuous vector space, capturing semantic relationships between nucleotides.

4.2 Model architecture

4.2.1 Testing external models

As introduced in Chapter 2, CRISPR-ONT and CRISPR-OFFT were investigated and I found that the CRISPR-ONT model predicts negative efficacy values for some sgRNAs. No sufficient explanation was found in [73] for this behaviour, so I decided to implement

my own on-target efficacy prediction network. This was used later on for the training and testing of the VAEs.

The possibility of computing the off-target effects with the CRISPR-OFFT model was also investigated. During the implementation of the VAEs, I discovered the fact that two components are needed for off-target effect calculation: an sgRNA sequence and an off-target site. I was able to produce sgRNAs with VAEs, but the off-target site, which is at equal length with the sgRNA - 23 bases long -, has many possible variations (exactly 4^{23}, because each base can be on any of the 23 positions). The biologically relevant variations can be sorted out via the rules of biology [21]. To integrate such rules into the model was beyond the scope of my thesis, so I could not include the off-target effect prediction into the training of VAEs.

4.2.2 On-target prediction model

The model's architecture is based on convolutional neural networks (CNNs), augmented with residual connections, attention mechanisms, and dropout for regularization. All of these concepts is detailed below.

- Convolutional layers. The model employs multiple 1D convolutional layers to capture local sequence features. CNNs are effective for biological sequences as they detect patterns like motifs and nucleotide dependencies along the sequence. Batch normalization after convolution stabilizes training by normalizing activations and reducing internal covariate shift.
- Residual connections. Residual blocks, inspired by ResNet [28], are incorporated to combat the vanishing gradient problem in deep networks and allow efficient training of deeper models. Shortcut connections (identity mapping or dimensionality adjustment via 1x1 convolutions) enable information flow across layers.
- Attention mechanism. The model includes an attention mechanism, which assigns weights to different positions in the sequence, enabling the model to focus on critical regions that influence sgRNA efficacy. Attention mechanisms have been shown to improve interpretability and performance in sequence-based tasks.
- Fully connected layers. After extracting hierarchical features, a global average pooling layer reduces spatial dimensions, followed by fully connected layers to map the features to efficacy predictions. Such pooling layers condense spatial information while maintaining key features.

4.2.3 Variational autoencoders

Figure 4.1 and 4.2 show high level block diagrams of the designed VAEs. In Figure 4.1 the input is a set of sgRNAs. The model is trained via calculating the standard VAE

loss (the sum of KL divergence and reconstruction loss). After training it can generate a given number of sgRNAs as output.

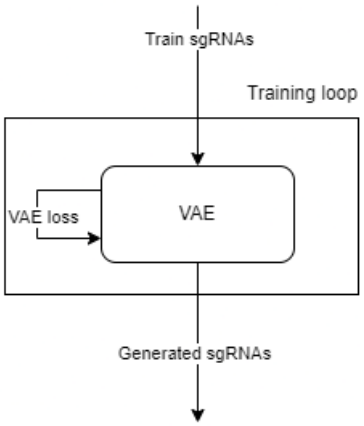

Figure 4.1: High level block diagram of the vanilla VAE architecture.

The on-target prediction model is included in the training loop of the extended VAE, as shown in Figure 4.2. This prediction model calculates the reconstructed sgRNAs' on-target efficacy loss as presented in subsection 5.1.2 and this term is included in the extended VAE's loss function. By including it in the loss function, it is also included in the optimization process of the VAE, opening up the possibility of optimized sgRNAs. The input and output of this model are the same as those of the vanilla VAE.

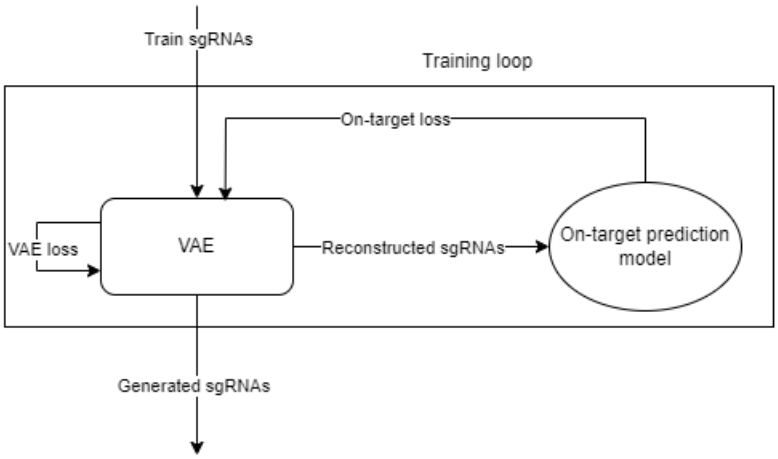

Figure 4.2: High level block diagram of the extended VAE architecture.

As mentioned before, VAEs excel at generating new samples that resemble the input data. Hence, this model architecture was chosen to generate new sgRNAs. Traditionally, a VAE consists of an encoder and a decoder, which I chose also.

- Encoder design. The encoder embeds sgRNA sequences using a learned embedding layer, mapping nucleotide sequences into a dense vector space. Two stacked LSTMs (Long Short-Term Memory networks) are used in the encoder to capture sequential dependencies in sgRNA sequences, reflecting the biological significance of nucleotide order in determining sgRNA functionality. The second LSTM's hidden state is transformed into the latent representation through two fully connected layers that estimate the mean (z_{mean}) and log variance (z_{logvar}) of the latent distribution. The latent space incorporates the reparameterization trick to ensure differentiability and facilitate stochastic sampling, as proposed by Kingma Welling in [39].

- Latent space. The latent space dimensionality (d_{latent}) is set to 32, enabling compact, expressive representations of sgRNA sequences while maintaining computational efficiency. A Gaussian prior distribution is imposed on the latent variables to regularize the model.

- Decoder design. The decoder transforms the sampled latent variables into sgRNA sequences by reversing the encoding process. A dense layer expands the latent representation to match the LSTM input dimensions. Two LSTMs decode the latent representation into sequence embeddings, followed by a linear output layer that projects the embeddings back into the nucleotide space. The output layer generates probabilities over the nucleotide vocabulary for each sequence position, allowing sgRNA reconstruction.

Loss function. The loss function of the vanilla VAE contains the first two elements of the list below, while the extended VAE comprises all three components in its loss function.

- Reconstruction loss: This is implemented using the cross-entropy loss between the reconstructed sequence and the input sgRNA sequence, ensuring fidelity in sgRNA sequence reconstruction.

- KL divergence: This regularizes the latent space by minimizing the divergence between the learned posterior distribution ($q(z|x)$) and the Gaussian prior distribution ($p(z)$) as in the standard VAE formulation [39].

- On-target loss integration: To prioritize sgRNAs with high predicted on-target efficacy, an additional term is incorporated into the loss function of the extended VAE shown in Figure 4.2. A pre-trained on-target model evaluates generated sgRNAs, and the squared error between the predicted efficacy and a desired efficacy (set to 1) is included in the loss function. This auxiliary loss is scaled by a weight ($w = 1000$) to ensure its influence during training.

The rationale for LSTM-based encoder and decoder is that LSTMs are particularly suited for sequence data due to their ability to capture long-term dependencies. This is critical for modeling sgRNA sequences, as the biological activity of sgRNAs is often influenced by dependencies between nucleotides that are not adjacent [21].

4.3 Training

4.3.1 Hardware and software environment

Using GPUs are essential to train complex models as efficient as possible. For this work, the university provided a Docker container, running on an Nvidia GRID V100DX-32C driver. Connection was established through SSH-connection.

The details of the training environment:

- System: Ubuntu 22.04.2 LTS
- CPUs: x86_64, 16 CPUs (16 sockets, 1 core/socket, 1 thread/core), Intel Xeon (Cascadelake), 6th Gen, 4199.96 BogoMIPS, 40-bit phys, 48-bit virt, Little Endian, 2.1GHz
- GPUs: Nvidia GRID V100DX-32C
- CUDA version: 12.0
- Memory: 64298 MB

All programs were written in Python, using the PyTorch library extensively. Visual Studio Code was chosen as the development environment, which also enabled the remote connection to the servers of the university. Logging was managed with Wandb, moreover the Matplotlib library was used for plotting the results.

4.3.2 Hyperparameters

From multiple hyperparameter settings the below setups proved to be the best. For the on-target prediction model:

- Dropout rate: 0.2
- Learning rate: 0.001
- Batch size: 256
- Epochs: 500
- Training dataset: 70%
- Validation and test dataset: 15%
- Optimizer: AdamW
- Loss function: MSE (Mean Squared Error)

For the vanilla (baseline) VAE:

- Latent space dimension: 32
- Dropout rate: 0.3
- Learning rate: 0.001
- Embedding dimension: 32
- Batch size: 64
- Epochs: 2000
- LSTM units in the first LSTM layer: 128
- LSTM units in the second LSTM layer: 64
- Training dataset: 80%
- Validation dataset: 20%
- Optimizer: AdamW

Both the vanilla and the extended VAE were trained on the whole DeepCRISPR dataset [17], with the same hyperparameters. The only difference is that its loss function included two more parameters: the on-target loss and the on-target loss weight. The latter was set to 1000 to make the on-target loss significant while training the model.

Chapter 5

Results

5.1 Evaluation metrics

5.1.1 On-target efficacy prediction model

To evaluate the performance of the on-target efficacy prediction network, I calculated standard metrics commonly used in the field of sgRNA design: Spearman correlation and Pearson correlation. Additionally, Mean Squared Error (MSE) and Mean Absolute Error (MAE) were tracked during training for more insights into the models' capabilities.

Spearman correlation coefficient, ρ, measures the monotonic relationship between predicted and true sgRNA efficacies. A high Spearman correlation indicates that the model effectively captures the ranking of sgRNAs based on their predicted efficacy. It can be computed as below, if all n ranks[1] are distinct integers (no ties):

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \tag{5.1}$$

where $d_i = \text{rank}(y_i) - \text{rank}(\hat{y}_i)$ is the difference in ranks of the true efficacy y_i and the predicted efficacy $\hat{y}_i$, and n is the number of data points.

Pearson correlation coefficient, r, quantifies the linear correlation between predicted and true efficacies. High Pearson values indicate strong linear alignment between the predictions and ground truth. Given paired data $\{(y_1, \hat{y_1}), \ldots, (y_n, \hat{y_n})\}$ consisting of n pairs, r is defined as

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}, \tag{5.2}$$

[1] https://en.wikipedia.org/wiki/Ranking_(statistics)

where y_i and $\hat{y}_i$ are the true and predicted efficacies, and $\bar{y}$ and $\bar{\hat{y}}$ are their respective means.

The Mean Squared Error quantifies the average of the squared differences between predicted and true values:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{5.3}$$

where y_i is the true efficacy, $\hat{y}_i$ is the predicted efficacy, and n is the number of data points.

The Mean Absolute Error measures the average of the absolute differences between predicted and true values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{5.4}$$

where y_i is the true efficacy, $\hat{y}_i$ is the predicted efficacy, and n is the number of data points.

Regarding the latter two metrics, lower values suggest better prediction performance.

5.1.2 Variational autoencoders

To evaluate the baseline VAE the standard VAE loss was used, which is the sum of the KL divergence and the reconstruction loss. For the extended VAE an $o * w$ term was added to the standard VAE loss, where o is the on-target efficacy loss and w is its weight. To compute the on-target efficacy loss, in each training epoch the actual batch of reconstructed sgRNAs were evaluated with the on-target efficacy prediction model trained on all cell lines. The average of these efficacy values, p, was chosen and used for computing the on-target loss the following way:

$$o = (p - 1)^2 \tag{5.5}$$

o is practically in the order of magnitude of 10^{-3} to 10^{-2}, while the standard VAE loss' magnitude is 10^1. This is why $10^3, 10^4$ and 10^5 were used as w values.

Furthermore, 1000 sgRNAs were generated by each trained VAE to visualize the results in form of histograms, showing the efficacy values on the x axis and the frequency of a given value on the y axis.

5.2 On-target efficacy prediction model

As mentioned in Chapter 4, the model was trained and evaluated with five different datasets, each corresponding to a separate training.

As a comparison, the most used metric in this field, the Spearman correlation coefficient, was investigated throughout each training setup. The result is shown in Figure 5.1. Each bin in the Figure corresponds to the dataset on which the model was trained, validated and tested. Only the test results are shown below, as they represent the real performance of the models.

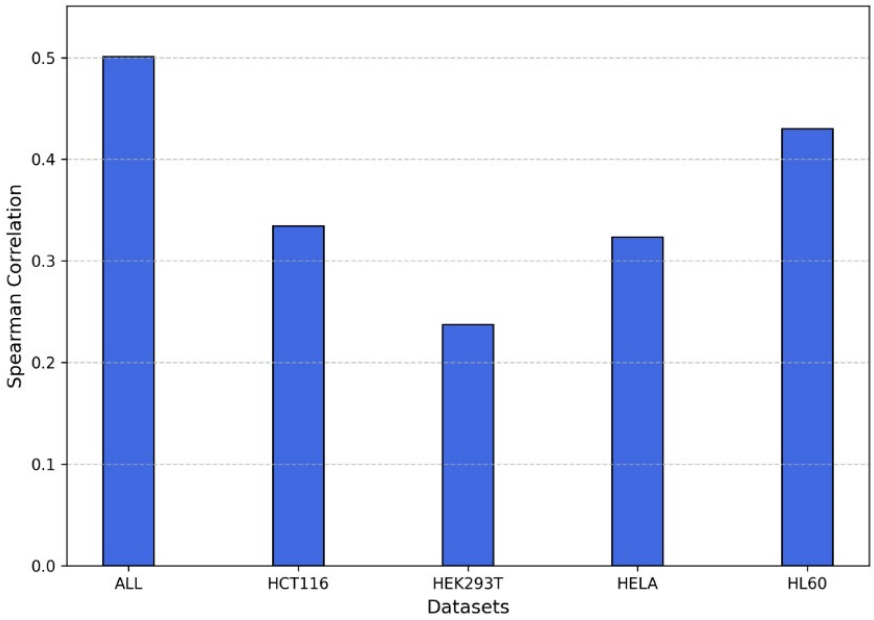

Figure 5.1: Test Spearman coefficient of the on-target efficacy prediction models trained on different cell lines.

Figure 5.2 and 5.3 provide more insight into the results of training.

Figure 5.2 highlights the performance of the on-target efficacy prediction model trained on the hct116 cell line dataset. The model was able to learn on the training data, because the training Spearman value increased, while the training MSE value decreased over the epochs (referring to the blue-coloured curves). The model, as it achieves at least 0.3 in Spearman coefficient on the test dataset (shown in Figure 5.1), outperforms the CHOPCHOP, sgRNA-Scorer and WU-CRISPR models listed in [17]. However, after the rapid initial decrease in the validation MSE score (marked with **B** in Figure 5.2) the curve showed an almost constant behaviour. This graph indicates that the model's generalization ability did not improve significantly after the first 50 epochs. Also, after a slight increase in validation MSE at around epoch 100, it starts to decrease slowly, meaning that no significant overfit is present.

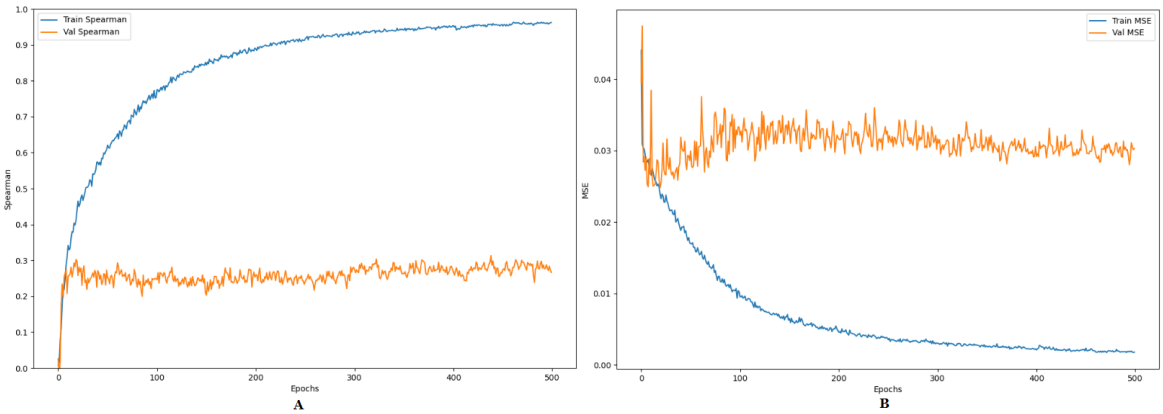

Figure 5.2: Metrics of the on-target efficacy prediction model trained on the hct116 dataset. Spearman correlation and MSE graphs are marked with **A, B** accordingly.

The performance of the on-target efficacy prediction model trained on all cell lines is shown in Figure 5.3. Similarly to the model trained on the hct116 cell line, the training loss (marked with blue in Figure 5.3, **B**) shows a strictly monotonic decrease while the training Spearman coefficient increases. Here, a more balanced validation MSE score means that even a smaller overfit occured while training than in the model trained on the hct116 cell line. This model reaches a Spearman score of 0.5 in the test dataset, resulting in a superior performance to all models except DeepCRISPR listed in [17] (Fig. 2e).

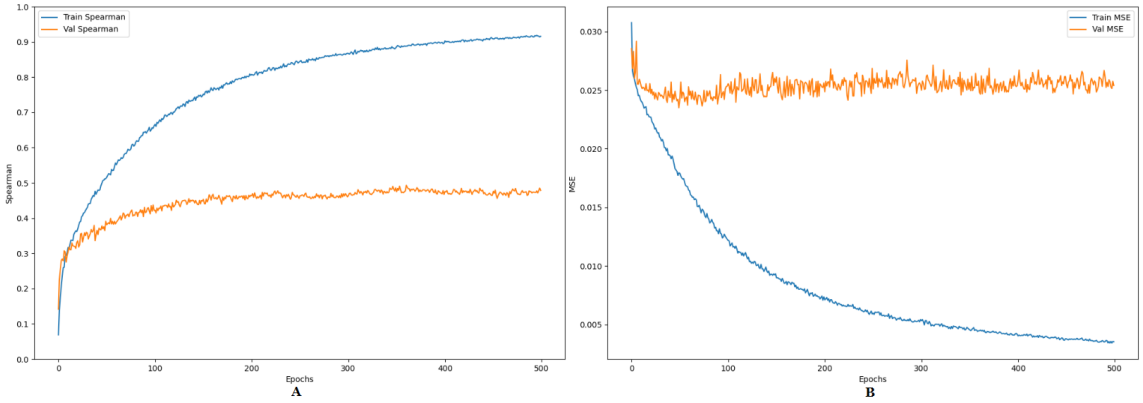

Figure 5.3: Metrics of the on-target efficacy prediction model trained on all cell lines. Spearman correlation and MSE graphs are marked with **A, B** accordingly.

In practice, the task to design sgRNAs for novel cell lines where experimental data is scarce or unavailable is crucial. A model that generalizes well across cell lines ensures accurate sgRNA predictions for these new contexts. To address this biologically relevant question how the model performs on a cell line it has never seen before, the model was trained on all cell lines except the hct116 and tested on the hct116 cell line. The training

results are shown in Figure 5.4. Overfit is clearly present here, as both validation metrics show: the Spearman score decreases and the MSE increases. Despite this fact, the model reaches 0.7 Spearman score on the test set containing data only from the hct116 cell line, meaning that it performs better than all models listed in [17]. As presented before, the model trained and tested on the hct116 cell line reaches only a little above 0.3 test Spearman value. The seemingly paradox behaviour can be explained by the fact that this model, of which training result is shown in Figure 5.2, is overfitting. This way, it might not be able to capture the underlying patterns in the hct116 dataset. However, the model trained on all cell lines except hct116 could not overfit to the hct116 data resulting in this performance.

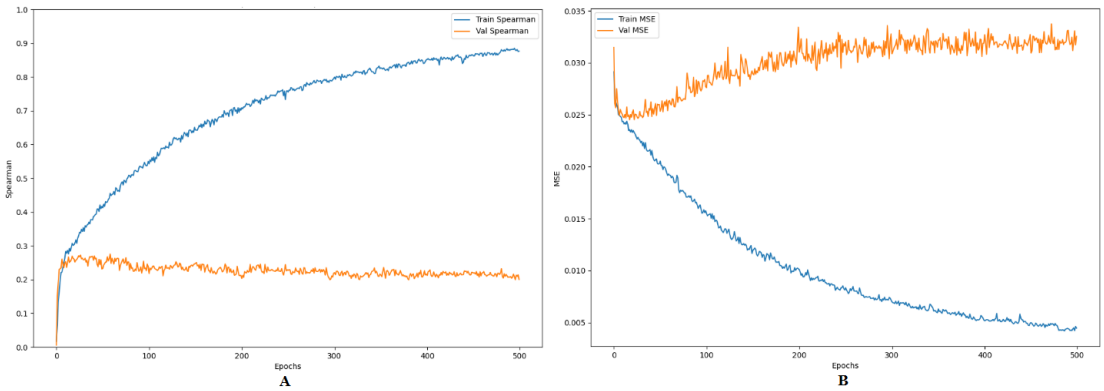

Figure 5.4: Metrics of the on-target efficacy prediction model trained on all cell lines except the hct116. Spearman correlation and MSE graphs are marked with **A, B** accordingly.

5.3 Vanilla and extended variational autoencoder

Both models were trained on the whole DeepCRISPR dataset [17]. Training and validation VAE loss was tracked during the training of the vanilla VAE, which is shown in Figure 5.5**A**. As both graphs reveal, the models learn on the training data because training losses decrease in a strictly monotonic manner. The reason for training the models for 2000 epochs is that overfit only occurs at around epoch 1500. In Figure 5.5**B** neither the training nor the validation on-target efficacy loss decreases, meaning that the extended VAE's ability of generating sgRNAs with higher on-target efficacies did not improve during training. The on-target efficacy loss remains constant even when its weight is set to 10^4 or 10^5, which is not shown in the Figures.

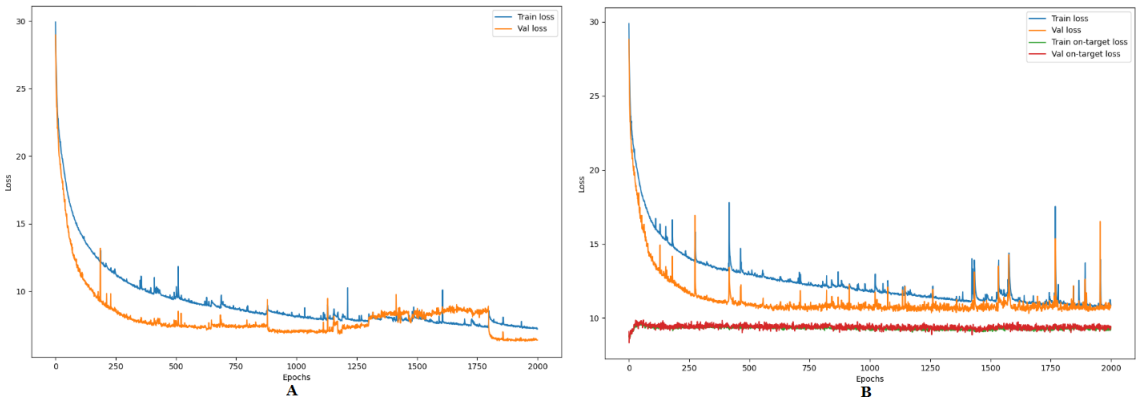

Figure 5.5: Training curve of the vanilla VAE in **A**, and of the extended VAE in **B**

To compare the models' ability of generating more efficient sgRNAs visually, the distribution of the generated sgRNAs are shown in Figure 5.6. The histograms show that the extended VAE is able to generate sgRNAs between 0.7-0.8, while the baseline produces sgRNAs below that efficacy. So in this regard the extended VAE performs better. As it is shown in the Figure, the vanilla VAE supersedes the extended VAE only in the range 0.4-0.6.

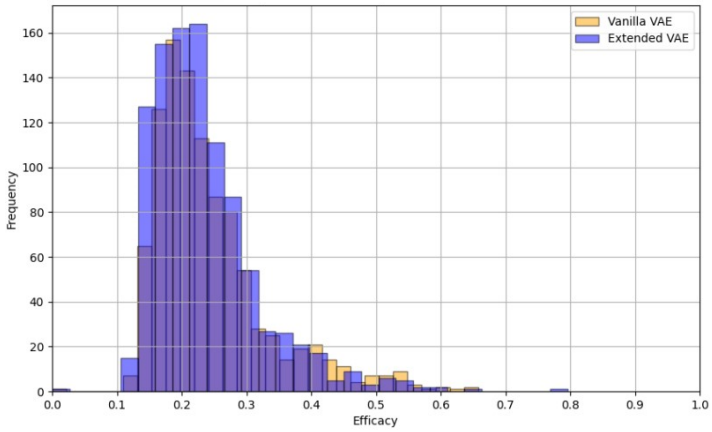

Figure 5.6: Distribution comparison of the vanilla and extended VAE.

Chapter 6

Ethical considerations

As with any new, powerful and human-related technology, CRISPR/Cas9 raises significant ethical considerations that must be addressed to ensure its responsible use [61]. In particular, the focus of my thesis on designing sgRNAs for the CRISPR technology adds another layer of responsibility, as the results of this work can directly influence how CRISPR is applied in all its application fields, since sgRNAs are central part of the CRISPR working mechanism. This chapter will explore the ethical concerns associated with CRISPR/Cas9 technology, highlighting the importance of using this technology for beneficial and moral purposes.

6.1 The potential for misuse

One of the primary ethical concerns surrounding CRISPR/Cas9 technology is the potential for misuse [61]. While the technology holds enormous promise for curing genetic diseases, its power also raises concerns about its application in ways that could have harmful consequences. The possibility of editing the human germline—modifying genes in embryos or reproductive cells—has sparked intense debate. On the one hand, germline editing could prevent the transmission of genetic diseases to future generations. On the other hand, it could also lead to the creation of "designer babies," where genetic traits are selected for non-medical reasons, such as enhancing physical appearance, intelligence, or other personal attributes. This raises questions about inequality, social pressure, and the potential for eugenics.

6.2 Equity and access

Another ethical consideration is the issue of equity and access to CRISPR-based technologies [61]. As CRISPR/Cas9 becomes more accessible, there is a risk that its benefits could be concentrated in wealthy, technologically advanced countries, while disadvantaged pop-

ulations may be excluded from these advancements. This issue is particularly pertinent in the context of gene therapies, which could revolutionize the treatment of genetic disorders, but may not be available to all who need them due to cost or lack of infrastructure.

Furthermore, the development of CRISPR technology through deep learning models, like the one in this thesis, could have an impact on the commercialization of these technologies. It is essential to ensure that the intellectual property generated through research in this field is not hoarded by a few entities but is shared equitably for the collective good. As a researcher, the responsibility lies in promoting transparency, fairness, and accessibility when applying CRISPR technologies in any field.

6.3 The environmental impact

The application of CRISPR/Cas9 in agriculture offers the potential to enhance food security by creating crops that are resistant to pests, diseases, or environmental stress. However, there are also ethical concerns related to genetically modified organisms (GMOs), which have been controversial for decades. Although CRISPR-edited crops differ from traditional GMOs in that they do not involve foreign gene insertion, their release into the environment raises questions about ecological balance and the long-term impacts on biodiversity [61].

Any use of CRISPR in agriculture must be accompanied by thorough environmental risk assessments to ensure that gene-edited organisms do not unintentionally disrupt ecosystems. As the design of sgRNAs becomes more precise and efficient, e.g. with the potential use of this work, it is crucial to consider not just the immediate benefits of gene editing but also its broader ecological and ethical consequences.

6.4 Moral responsibility in research and development

The responsibility of researchers and developers in the field of CRISPR/Cas9 technology is essential [61]. The tools and methodologies created must be used with the intent to improve human welfare and societal well-being. The deep learning model used for designing sgRNAs in my thesis is not only a technical achievement but also a tool that can be applied in both therapeutic and agricultural settings. It is vital that these results are applied ethically, for purposes that align with the greater good, such as curing genetic diseases, improving crop yields, or combating pathogens. The intent behind each application of CRISPR/Cas9 technology should be guided by a commitment to human dignity, justice, and the prevention of harm.

Using the ideas and implementation presented in my thesis responsibly can facilitate the prospective applications of CRISPR.

Chapter 7

Summary and future work

In my work, literature crucial to understanding the connection between the CRISPR/Cas9 gene editing technology and deep learning was reviewed. sgRNAs, vital components of CRISPR, were investigated from the effectiveness point of view. An ideal sgRNA maximizes on-target efficacy while minimizing off-target effects. Achieving this with the help of deep learning was the guideline of this thesis. A dataset was processed for two types of deep learning models: the on-target efficacy prediction model and variational autoencoders.

An on-target efficacy prediction model was designed, implemented and tested, which proved superior to many models listed in [17]. It has to be emphasized, that this model achieved outstanding results when testing it on a new cell line, making it applicable in real-world experiments. Moreover, the possibility of using an off-target prediction network was examined and practical obstacles were highlighted. A baseline VAE was also created, trained, and evaluated. Based on that, an extended VAE was also explored. I demonstrated that directly adding the on-target loss with weight to the standard VAE loss can slightly increase the model's ability to generate more efficient sgRNAs.

In terms of future research work, there are many ways to broaden the perspective presented in this thesis. Regarding the results of the on-target efficacy prediction model that was trained on all cell lines except one, there could be further improvements in predictive performance using the idea of leaving one cell line out. Also, the possibility of including off-target optimization during VAE training can be considered for which a potentially useful database might be found in [68]. By the design of the VAEs, further options for integrating the on-target efficacy loss should be elaborated upon.

Acknowledgements

I would like to express my sincere gratitude to Dániel Unyi and Bálint Gyires-Tóth, PhD for their professional advice and for always being open to my questions, which proved crucial in completing my work.

ChatGPT was used for language improvement and creating drafts while writing this thesis.

Bibliography

[1] European Information Technologies Certification Academy. How does adding more data to a deep learning model impact its accuracy? URL https://eitca.org/artificial-intelligence/eitc-ai-dltf-deep-learning-with-tensorflow/tensorflow/using-more-data/examination-review-using-more-data/how-does-adding-more-data-to-a-deep-learning-model-impact-its-accuracy/.

[2] Prabin Adhikari and Mousami Poudel. Crispr-cas9 in agriculture: Approaches, applications, future perspectives, and associated challenges. *Malaysian Journal of Halal Research*, 3(1):6–16, 2020. DOI: doi:10.2478/mjhr-2020-0002. URL https://doi.org/10.2478/mjhr-2020-0002.

[3] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[4] Misganaw Asmamaw and Belay Zawdie. Mechanism and applications of CRISPR/Cas-9-mediated genome editing. *Biologics*, 15:353–361, 2021.

[5] Leonardo Banh and Gero Strobel. Generative artificial intelligence. *Electronic Markets*, 33(1):63, 2023.

[6] Rodolphe Barrangou, Christophe Fremaux, Hélène Deveau, Melissa Richards, Patrick Boyaval, Sylvain Moineau, Dennis A Romero, and Philippe Horvath. CRISPR provides acquired resistance against viruses in prokaryotes. *Science*, 315(5819):1709–1712, 2007.

[7] Ajaz A Bhat, Sabah Nisar, Soumi Mukherjee, Nirmalya Saha, Nageswari Yarravarapu, Saife N Lone, Tariq Masoodi, Ravi Chauhan, Selma Maacha, Puneet Bagga, et al. Integration of crispr/cas9 with artificial intelligence for improved cancer therapeutics. *Journal of translational medicine*, 20(1):534, 2022.

[8] Devaki Bhaya, Michelle Davison, and Rodolphe Barrangou. Crispr-cas systems in bacteria and archaea: versatile small rnas for adaptive defense and regulation. *Annu. Rev. Genet.*, 45(1):273–297, 2011.

[9] Stan J J Brouns, Matthijs M Jore, Magnus Lundgren, Edze R Westra, Rik J H Slijkhuis, Ambrosius P L Snijders, Mark J Dickman, Kira S Makarova, Eugene V

Koonin, and John van der Oost. Small CRISPR RNAs guide antiviral defense in prokaryotes. *Science*, 321(5891):960–964, 2008.

[10] Shichen Cao, Jingjing Li, Kenric P Nelson, and Mark A Kon. Coupled vae: Improved accuracy and robustness of a variational autoencoder. *Entropy*, 24(3):423, 2022.

[11] Jason Carte, Ruiying Wang, Hong Li, Rebecca M Terns, and Michael P Terns. Cas6 is an endoribonuclease that generates guide RNAs for invader defense in prokaryotes. *Genes Dev.*, 22(24):3489–3496, 2008.

[12] Minjiang Chen, Aiwu Mao, Min Xu, Qiaoyou Weng, Jianting Mao, and Jiansong Ji. Crispr-cas9 for cancer therapy: Opportunities and challenges. *Cancer Letters*, 447:48–55, 2019. ISSN 0304-3835. DOI: https://doi.org/10.1016/j.canlet.2019.01.017. URL https://www.sciencedirect.com/science/article/pii/S0304383519300291.

[13] Shengmiao Chen, Yufeng Yao, Yanchun Zhang, and Gaofeng Fan. Crispr system: Discovery, development and off-target detection. *Cellular Signalling*, 70:109577, 2020. ISSN 0898-6568. DOI: https://doi.org/10.1016/j.cellsig.2020.109577. URL https://www.sciencedirect.com/science/article/pii/S0898656820300541.

[14] Kristy Choi, Curtis Hawthorne, Ian Simon, Monica Dinculescu, and Jesse Engel. Encoding musical style with transformer autoencoders, 2020. URL https://arxiv.org/abs/1912.05537.

[15] Sanghyuk Roy Choi and Minhyeok Lee. Transformer architecture and attention mechanisms in genome data analysis: a comprehensive review. *Biology*, 12(7):1033, 2023.

[16] James P. Carney Christopher A. Lino, Jason C. Harper and Jerilyn A. Timlin. Delivering CRISPR: a review of the challenges and approaches. *Drug Delivery*, 25(1):1234–1257, 2018. DOI: 10.1080/10717544.2018.1474964. URL https://doi.org/10.1080/10717544.2018.1474964.

[17] Guohui Chuai, Hanhui Ma, Jifang Yan, Ming Chen, Nanfang Hong, Dongyu Xue, Chi Zhou, Chenyu Zhu, Ke Chen, Bin Duan, et al. Deepcrispr: optimized crispr guide rna design by deep learning. *Genome biology*, 19:1–18, 2018.

[18] American Type Culture Collection. URL https://www.atcc.org/.

[19] Mahintaj Dara, Mehdi Dianatpour, Negar Azarpira, and Navid Omidifar. Convergence of crispr and artificial intelligence: A paradigm shift in biotechnology. *Human Gene*, 41:201297, 2024. ISSN 2773-0441. DOI: https://doi.org/10.1016/j.humgen.2024.201297. URL https://www.sciencedirect.com/science/article/pii/S277304412400041X.

[20] Elitza Deltcheva, Krzysztof Chylinski, Cynthia M Sharma, Karine Gonzales, Yanjie Chao, Zaid A Pirzada, Maria R Eckert, Jörg Vogel, and Emmanuelle Charpentier. CRISPR RNA maturation by trans-encoded small RNA and host factor RNase III. *Nature*, 471(7340):602–607, 2011.

[21] John G Doench, Nicolo Fusi, Meagan Sullender, Mudra Hegde, Emma W Vaimberg, Katherine F Donovan, Ian Smith, Zuzana Tothova, Craig Wilen, Robert Orchard, et al. Optimized sgrna design to maximize activity and minimize off-target effects of crispr-cas9. *Nature biotechnology*, 34(2):184–191, 2016.

[22] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[23] Jennifer A. Doudna. The promise and challenge of therapeutic genome editing. *Nature*, 578:229–236, 2020. DOI: 10.1038/s41586-020-1978-5. URL https://doi.org/10.1038/s41586-020-1978-5.

[24] Michael Evans. The concept of statistical evidence, historical roots and current developments. *Encyclopedia*, 4(3):1201–1216, 2024.

[25] Susan Gottesman. Microbiology: Dicing defence in bacteria. *Nature*, 471(7340): 588–589, 2011.

[26] Congting Guo, Xiaoteng Ma, Fei Gao, and Yuxuan Guo. Off-target effects in crispr/cas9 gene editing. *Frontiers in Bioengineering and Biotechnology*, 11, 2023. ISSN 2296-4185. DOI: 10.3389/fbioe.2023.1143157. URL https://www.frontiersin.org/journals/bioengineering-and-biotechnology/articles/10.3389/fbioe.2023.1143157.

[27] Rachel E Haurwitz, Martin Jinek, Blake Wiedenheft, Kaihong Zhou, and Jennifer A Doudna. Sequence- and structure-specific RNA processing by a CRISPR endonuclease. *Science*, 329(5997):1355–1358, 2010.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] Frank Hille and Emmanuelle Charpentier. CRISPR-Cas: biology, mechanisms and relevance. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 371(1707):20150496, 2016.

[30] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.

[31] Patrick D. Hsu, Eric S. Lander, and Feng Zhang. Development and applications of crispr-cas9 for genome engineering. *Cell*, 157(6):1262–1278, 2014. ISSN 0092-8674. DOI: https://doi.org/10.1016/j.cell.2014.05.010. URL https://www.sciencedirect.com/science/article/pii/S0092867414006047.

[32] IBM. What is a variational autoencoder? URL https://www.ibm.com/think/topics/variational-autoencoder.

[33] National Human Genome Research Institute. Base pair. URL https://www.genome.gov/genetics-glossary/Base-Pair.

[34] Yoshizumi Ishino, Hiroshi Shinagawa, Kunihiko Makino, Mariko Amemura, and Akiko Nakata. Nucleotide sequence of the iap gene, responsible for alkaline phosphatase isozyme conversion in escherichia coli, and identification of the gene product. *Journal of Bacteriology*, 169(12):5429–5433, 1987. DOI: 10.1128/jb.169.12.5429-5433.1987.

[35] Maria Jackson, Leah Marks, Gerhard H.W. May, and Joanna B. Wilson. The genetic basis of disease. *Essays in Biochemistry*, 62(5):643–723, 2018. ISSN 0071-1365. DOI: 10.1042/EBC20170053. URL https://doi.org/10.1042/EBC20170053.

[36] Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A. Doudna, and Emmanuelle Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *Science*, 337(6096):816–821, 2012. DOI: 10.1126/science.1225829. URL https://www.science.org/doi/abs/10.1126/science.1225829.

[37] Mohadeseh Khoshandam, Hossein Soltaninejad, Marziyeh Mousazadeh, Amir Ali Hamidieh, and Saman Hosseinkhani. Clinical applications of the crispr/-cas9 genome-editing system: Delivery options and challenges in precision medicine. *Genes & Diseases*, 11(1):268–282, 2024. ISSN 2352-3042. DOI: https://doi.org/10.1016/j.gendis.2023.02.027. URL https://www.sciencedirect.com/science/article/pii/S235230422300079X.

[38] Hyongbum Kim and Jin-Soo Kim. A guide to genome engineering with programmable nucleases. *Nature Reviews Genetics*, 15(5):321–334, 2014.

[39] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[40] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[41] Odatha W. Kotagama, Chanika D. Jayasinghe, and Thelma Abeysinghe. Era of genomic medicine: A narrative review on crispr technology as a potential therapeutic tool for human diseases. *BioMed Research International*, 2019(1):1369682, 2019. DOI: https://doi.org/10.1155/2019/1369682. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2019/1369682.

[42] Andrej Krenker, Janez Bešter, and Andrej Kos. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, pages 1–18, 2011.

[43] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[44] Hongmei Lisa Li, Naoko Fujimoto, Noriko Sasakawa, Saya Shirai, Tokiko Ohkame, Tetsushi Sakuma, Michihiro Tanaka, Naoki Amano, Akira Watanabe, Hidetoshi Sakurai, Takashi Yamamoto, Shinya Yamanaka, and Akitsu Hotta. Precise correction of the dystrophin gene in duchenne muscular dystrophy patient induced pluripotent stem cells by talen and crispr-cas9. *Stem Cell Reports*, 4(1):143–154, 2015. ISSN 2213-6711. DOI: https://doi.org/10.1016/j.stemcr.2014.10.013. URL https://www.sciencedirect.com/science/article/pii/S221367111400335X.

[45] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[46] Jiecong Lin and Ka-Chun Wong. Off-target predictions in crispr-cas9 gene editing using deep learning. *Bioinformatics*, 34(17):i656–i663, 2018.

[47] Nathanael G Lintner, Melina Kerou, Susan K Brumfield, Shirley Graham, Huanting Liu, James H Naismith, Matthew Sdano, Nan Peng, Qunxin She, Valérie Copié, Mark J Young, Malcolm F White, and C Martin Lawrence. Structural and functional characterization of an archaeal clustered regularly interspaced short palindromic repeat (CRISPR)-associated complex for antiviral defense (CASCADE). *J. Biol. Chem.*, 286(24):21643–21656, 2011.

[48] Qiaoyue Liu, Xiang Cheng, Gan Liu, Bohao Li, and Xiuqin Liu. Deep learning improves the ability of sgrna off-target propensity prediction. *BMC bioinformatics*, 21:1–15, 2020.

[49] Zhepeng Liu, Shuliang Chen, Xu Jin, Qiankun Wang, Kongxiang Yang, Chenlin Li, Qiaoqiao Xiao, Panpan Hou, Shuai Liu, Shaoshuai Wu, Wei Hou, Yong Xiong, Chunyan Kong, Xixian Zhao, Li Wu, Chunmei Li, Guihong Sun, and Deyin Guo. Genome editing of the HIV co-receptors CCR5 and CXCR4 by CRISPR-Cas9 protects CD4+ T cells from HIV-1 infection. *Cell Biosci.*, 7(1):47, 2017.

[50] Cui-Cui Ma, Zhen-Ling Wang, Ting Xu, Zhi-Yao He, and Yu-Quan Wei. The approved gene therapy drugs worldwide: from 1998 to 2019. *Biotechnology Advances*, 40:107502, 2020. ISSN 0734-9750. DOI: https://doi.org/10.1016/j.biotechadv.2019.107502. URL https://www.sciencedirect.com/science/article/pii/S0734975019302022.

[51] Kira S Makarova, Daniel H Haft, Rodolphe Barrangou, Stan J J Brouns, Emmanuelle Charpentier, Philippe Horvath, Sylvain Moineau, Francisco J M Mojica, Yuri I Wolf, Alexander F Yakunin, John van der Oost, and Eugene V Koonin. Evolution and classification of the CRISPR-Cas systems. *Nat. Rev. Microbiol.*, 9(6):467–477, 2011.

[52] Hakim Manghwar, Bo Li, Xiao Ding, Amjad Hussain, Keith Lindsey, Xianlong Zhang, and Shuangxia Jin. Crispr/cas systems in genome editing: Methodologies and tools for sgrna design, off-target evaluation, and strategies to mitigate off-target effects. *Advanced Science*, 7(6):1902312, 2020. DOI: https://doi.org/10.1002/advs.201902312. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201902312.

[53] Toshiki Ochiai, Tensei Inukai, Manato Akiyama, Kairi Furui, Masahito Ohue, Nobuaki Matsumori, Shinsuke Inuki, Motonari Uesugi, Toshiaki Sunazuka, Kazuya Kikuchi, et al. Variational autoencoder-based chemical latent space for large molecular structures with 3d complexity. *Communications Chemistry*, 6(1):249, 2023.

[54] Vivek Pandey, Anima Tripathi, Ravi Bhushan, Akhtar Ali, and Pawan Dubey. Application of crispr/cas9 genome editing in genetic disorders: A systematic review up to date. *Journal of Genetic Syndromes & Gene Therapy*, 08, 12 2017. DOI: 10.4172/2157-7412.1000321.

[55] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems*, 29, 2016.

[56] Gabriel San Martin, Enrique López Droguett, Viviane Meruane, and Márcio das Chagas Moura. Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis. *Structural Health Monitoring*, 18(4):1092–1128, 2019.

[57] ScienceDirect. Chimeric rna. URL https://www.sciencedirect.com/topics/neuroscience/chimeric-rna.

[58] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*, 2017.

[59] Shengfu Shen, Tiing Jen Loh, Hongling Shen, Xuexiu Zheng, and Haihong Shen. CRISPR as a strong gene editing tool. *BMB Rep.*, 50(1):20–24, 2017.

[60] Zeinab Sherkatghanad, Moloud Abdar, Jeremy Charlier, and Vladimir Makarenkov. Using traditional machine learning and deep learning methods for on-and off-target prediction in crispr/cas9: a review. *Briefings in Bioinformatics*, 24(3):bbad131, 2023.

[61] Zabta Khan Shinwari, Faouzia Tanveer, and Ali Talha Khalil. Ethical issues regarding crispr mediated genome editing. *Current issues in molecular biology*, 26(1):103–110, 2018.

[62] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[63] Edward A. Stadtmauer, Joseph A. Fraietta, and Megan M. Davis et al. CRISPR-engineered T cells in patients with refractory cancer. *Science*, 367(6481):eaba7365, 2020. DOI: 10.1126/science.aba7365. URL https://www.science.org/doi/abs/10.1126/science.aba7365.

[64] Jeffrey R. Strich and Daniel S. Chertow. CRISPR-Cas biology and its application to infectious diseases. *Journal of Clinical Microbiology*, 57(4), 2019. DOI: 10.1128/jcm.01307-18. URL https://journals.asm.org/doi/abs/10.1128/jcm.01307-18.

[65] Michael P Terns and Rebecca M Terns. CRISPR-based adaptive immune systems. *Curr. Opin. Microbiol.*, 14(3):321–327, 2011.

[66] Rens van de Schoot, Sarah Depaoli, Ruth King, Bianca Kramer, Kaspar Märtens, Mahlet G Tadesse, Marina Vannucci, Andrew Gelman, Duco Veen, Joukje Willemsen, et al. Bayesian statistics and modelling. *Nature Reviews Methods Primers*, 1(1):1, 2021.

[67] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[68] Grant Wang, Xiaona Liu, Aoqi Wang, Jianguo Wen, Pora Kim, Qianqian Song, Xiaona Liu, and Xiaobo Zhou. Crisprofft: comprehensive database of crispr/cas off-targets. *Nucleic Acids Research*, page gkae1025, 2024.

[69] Blake Wiedenheft, Samuel H Sternberg, and Jennifer A Doudna. Rna-guided genetic silencing systems in bacteria and archaea. *Nature*, 482(7385):331–338, 2012.

[70] Wikipedia. Autoencoder. URL https://en.wikipedia.org/wiki/Autoencoder#History.

[71] Xi Xiang, Giulia I Corsi, Christian Anthon, Kunli Qu, Xiaoguang Pan, Xue Liang, Peng Han, Zhanying Dong, Lijun Liu, Jiayan Zhong, et al. Enhancing crispr-cas9 grna efficiency prediction by data integration and deep learning. *Nature communications*, 12(1):3238, 2021.

[72] LeCun Y., Bengio Y., and Hinton G. Deep learning. *Nature*, 521:436–444, 2015. URL https://doi.org/10.1038/nature14539.

[73] Guishan Zhang, Tian Zeng, Zhiming Dai, and Xianhua Dai. Prediction of crispr/cas9 single guide rna cleavage efficiency and specificity by attention-based convolutional neural networks. *Computational and Structural Biotechnology Journal*, 19:1445–1457, 2021. ISSN 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2021.03.001. URL https://www.sciencedirect.com/science/article/pii/S2001037021000738.

[74] Xiao-Hui Zhang, Louis Y Tee, Xiao-Gang Wang, Qun-Shan Huang, and Shi-Hua Yang. Off-target effects in crispr/cas9-mediated genome engineering. *Molecular Therapy - Nucleic Acids*, 4:e264, 2015. ISSN 2162-2531. DOI: https://doi.org/10.1038/mtna.2015.37. URL https://www.sciencedirect.com/science/article/pii/S216225311630049X.

[75] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

All URLs are accessed in November 2024.