Attention-based deep learning for DNA repair outcome prediction

Learning how the cell repairs DNA breaks using local sequence context

Jurrian de Boer

Delft University of Technology

TU Delft

Attention-based deep learning for DNA repair outcome prediction

Learning how the cell repairs DNA breaks using local sequence context

by

Jurrian de Boer

Student number:	5405289	
Masters programme:	Computing Science	
Faculty:	Electrical Engineering, Mathematics and Computer Science	
Research Group:	Pattern Recognition & Bioinformatics	
Project duration:	November 2021 - October 2022	
Thesis committee:	Prof. dr. ir. Marcel Reinders	TU Delft
	Dr. Joana Gonçalves	TU Delft, supervisor
	Dr. Odette Scharenborg	TU Delft
	MSc Colm Seale	TU Delft, daily supervisor

Cover:	Shutterstock
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

To me, biology is in its fundamentals wildly interesting. It is a strange biochemical dance; the unexpected route the universe takes while pushing towards entropy. I find it fascinating that somewhere on this path our consciousness developed which allowed us to observe where we are and who we are. We started looking outwards, exploring the earth, and we started looking inwards, exploring the microscopic world that is everywhere. In 17th century Delft, Antonie van Leeuwenhoek was the first to document microscopic observations of Biology. Ever since we have been creating tools to make such observations on an increasingly smaller scale. DNA was discovered and we started measuring and perturbing it. We found that our own DNA consists of roughly 3.2 billion elements, which is apparently enough to carry the instructions for creating and maintaining a human being. Meanwhile, the computer on which I am currently writing has roughly 3.2 billion remaining bytes available, which is insufficient to install updates. On this computer I code instructions that set off a surge of blind and uninformed calculations based on a model with randomly initialized parameters. By comparing the outputs of this model with billions of DNA repair measurements and updating its parameters accordingly, the calculations become incrementally less blind and more informed, until we are left with a model that has some understanding of the dynamics of DNA repair. I hope the reader can share my sense of wonder and enthusiasm for this subject when reading my work.

I would like to thank Marcel Reinders and Odette Scharenborg for investing their time to learn about my work. I am grateful of my friends and family for their interest in my work and for hearing my tumultuous explanations of the subject before I really understood it myself. I thank Sander, Attila, Yasin, Roy, Aaron, Matthijs, Ruben, Pia, Francesca, Caroline, Eric, Kirti, Frank and Sjoerd for their valuable feedback, interesting discussions, support and gezelligheid during my project. I am greatly thankful of my daily supervisors, Joana de Pinho Gonçalves and Colm Seale, for their availability, their flexibility, their dedication to the project and for their great enthusiasm for the subject.

Jurrian de Boer
Delft, October 2022

Attention-based deep learning for DNA repair outcome prediction

Jurrian H.D. de Boer
Delft University of Technology
MSc in Computing Science

***Abstract*—Recent advancements in quantification of repair outcomes of CRISPR-Cas9 mediated double-stranded DNA breaks (DSBs) have allowed for the use of machine learning for predicting the frequencies of these repair outcomes. Local DNA sequence context influences the frequencies of mutations that arise when DNA gets repaired after it is targeted by CRISPR (CRISPR outcomes). Contemporary models exploit this and can predict what the frequencies are of CRISPR outcomes at predetermined genomic loci. Predictions of such models are reasonably precise, but there may be opportunities for improvement in how the DNA sequence context is leveraged for making predictions. Some models only utilize a set of hand-crafted features, limiting the available information for the model. Other models do utilize broader sequence context but disregard sequence order or only predict a limited set of outcome classes. In this work we present an attention-based deep learning model that uses DNA sequence context to make fine-grained CRISPR outcome predictions. We present a custom input embedding for representing DSB repair outcomes and we expand on existing methods for analyzing attention-based models.**

I. Introduction

WITH the introduction of CRISPR-Cas9, genome editing has advanced greatly in the over the past decade. CRISPR-Cas9 (CRISPR) is a transformative technology that can be used to introduce a DNA double strand breaks (DSBs) at a predetermined location in the genome of a cell [1]. The technology operates by recruiting a Cas9 nuclease to a chosen location in the DNA (target sequence) using a synthetic guide RNA (gRNA). The nuclease cuts the DNA introducing a DSB, after which the cell will attempt to mend the DNA. CRISPR is widely used for gene editing [2] and gene-based therapeutics (e.a. [3], [4]) by exploiting this process. For these applications, it is important to gain more insights into the dynamics of CRISPR induced DSBs and subsequent DNA repair.

DSB repair is a stochastic process, where the resulting DNA sequence may differ from the original DNA sequence (mutation). While these repair outcomes of CRISPR-Cas9 induced DSBs (CRISPR outcomes, repair outcomes) were initially thought to be random [1], evidence now suggests that they are in fact partially determined by a number of factors, such as target site sequence context [5]–[7], cell-state [5] or even the orientation of binding of the Cas9-gRNA complex [8]. This suggests that CRISPR outcomes follow a probability distribution that is perhaps predictable. Indeed, various machine learning models have recently been developed attempting to predict CRISPR outcomes based on local sequence features such as cut site adjacent base pairs, microhomologies, and the protospacer-adjacent motif (PAM) [9]–[14]. Such CRISPR outcome prediction technologies could assist researchers to know what outcomes to expect when using CRISPR therapeutically. They can help us learn what influences those CRISPR outcomes and use this to design CRISPR assays in a way to maximize accuracy and minimize side effects.

Although contemporary models are able to predict CRISPR outcomes with reasonable precision, there may still be opportunities for improvement by changing how target sequence features are utilized. CRISPR outcomes and their frequencies are influenced by and can be predicted based on the DNA sequence context around the target site [5]–[7]. Some present models exploit such sequence context by making use of a set of hand-crafted features based on established knowledge [10], [11]. However, this limits the available information for the model to only what is included in the hand-crafted feature set, and it prevents us from finding novel relationships between sequence context and DNA repair outcomes. Other models do utilize the broader sequence context but have different limitations. In [9], the target sequence order is disregarded by treating sequence features as independentent, and in [12], [13], predictions are made only for broad CRISPR outcome classes like insertion/deletion ratio and frameshift frequency, instead of the more fine-grained prediction of individual repair outcome frequencies that we see in other models.

In this work we propose the use of sequence-based deep learning to leverage the complete target sequence context for predicting CRISPR outcomes on the resolution of individual outcome frequencies. A recently developed promising architecture for sequential data is the Transformer [15]. Transformers are attention-based deep learning frameworks that were able to obtain state-of-the-art results in the field of natural language processing by capturing contextual information in the input sequence (e.a. [16]). Contextual sequence information is relevant for working with features in DNA sequences such as microhomologies (MHs), which are pairs sequence features that strongly influence DNA repair outcomes [5], [7], [17], but require contextual information to be detected. Attention-based models have already been successfully employed on other DNA sequence based problems [18], [19]. Moreover, a deep learning framework for predicting CRISPR outcomes was published during our research that employed attention in conjunction with BiLSTM layers [14]. In this work, we present an attention-based deep learning model that employs a custom embedding of repair outcomes for predicting CRISP outcomes.

In addition to modeling contextual sequence information, the self-attention operation that is the fundamental building block of attention-based models provides opportunities for model interpretation by analyzing its intermediate representations (attention values) [20], [21]. In this work, we use these methods to reveal behaviour of our model in the context of CRISPR outcome prediction, and we expand on these methods by introducing alternative attention visualizations. Although the reliability of utilizing attention values for model explanation is disputed [22], [23], we validated that attention values provided a useful demonstration of model behaviour in controlled learning contexts that were relevant for CRISPR outcome prediction.

Our models did not achieve state-of-the-art performance, and we did not report novel sequence features for CRISPR outcome predictions. However, our work forms a basis for attention-based CRISPR outcome prediction. We contribute a custom input embedding that combines target sequence context with outcome specific characteristics. Our work may provide opportunities for improving our understanding of the dynamics of CRIPSR induced DSBs and resulting repair outcomes.

II. Methods

In the present study, we had two main research questions: (I) can we use an attention-based deep learning architecture to predict CRISPR outcomes, and (II) can we interpret these models to gain novel insights about DSB repair. To answer these questions, we established two intermediate prediction problems: microhomology detection (MH detection) and microhomology length detection (MH length detection). These problems simplified aspects of CRISPR outcome prediction in order to explore model training and interpretation in the present learning context and to validate our methods.

A. Prediction problem definitions

The introduction of a DSB at a given target sequence using CRISPR activates DNA repair mechanisms in the cell. DNA repair is not always faithful to the original DNA sequence, resulting in mutations of the DNA. In this way, the targeting of a given DNA sequence using CRISPR can generate a range of repair outcomes, each with a given probability. The goal of CRISPR outcome prediction is to predict the probabilities of CRISPR outcomes given a DNA target sequence. So the input into our model is the set of possible CRISPR outcomes for a given DNA target sequence, encoded using featurized sequence and position representations (see Section II-C), and the output is the probability distribution of these outcomes.

These repair outcomes include insertions, where one or more nucleotides (A, C, G, T) are inserted into the DNA at the cut site, and deletions, where one or more nucleotides are removed around the cut site. In the literature (e.a. [10]), 'CRISPR outcomes' refers to this set of repair outcomes. However, in the present study, we use 'CRISPR outcomes' to refer only to the set of possible deletions and we focus on predicting only their frequencies. We chose to focus on this aspect of the prediction task because a the set of features with predictive power over deletion outcomes is larger and more complex than the set of features for modeling insertions [10]. We left insertion modeling and the modeling of the interaction between insertion and deletion probabilities as a subject for future work.

There are two main DSB repair pathways; homology directed repair (HDR) and non-homologous end joining (NHEJ). The HDR pathway is generally accurate, while NHEJ is more error-prone [24]. Therefore, when we observe CRISPR outcomes, they are usually the result of NHEJ. The NHEJ pathway can be further subdivided into the two subcategories c-NHEJ and MMEJ (alt-NHEJ), each of which has their own characteristics and outcome profiles. MMEJ, can make use of microhomologies (MHs) to repair CRISPR-induced DSBs. MHs are small stretches of DNA located on both sides of the CRISPR cut site (i.e. the location where CRISPR indtroduces a DSB) that are identical in both nucleotide composition and order. In MMEJ, the cell uses these MHs to align loose DNA ends. The homology sequence upstream of the cut site on one DNA strand algins with the homology downstream of the cut site on the other DNA strand. The DNA is ligated and excess single-stranded DNA is removed, resulting in deletions in the sequence (Figure 1) [17], [24].

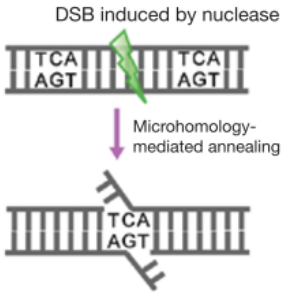


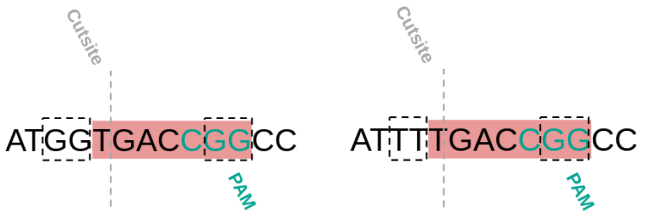
Fig. 1: Model for microhomology mediated deletions [17], [24]. After a DSB induced by a nuclease, for example CRISPR, MHs from either side of the DSB align with each other, after which excess single-stranded DNA is removed. From this process results a deletion genotype that is specific for a given MH. Figure taken from Bae *et al.* [17]

Indeed, MHs are important features for repair outcome prediction [5], [7], [17]. Therefore it is important that a repair outcome prediction model is able to detect MHs. This is by itself not a difficult problem: MHs can be detected by a simple algorithm. However, by training a model to detect MHs, we can validate that the model can distill such features from an input embedding and we can analyze if this behaviour is interpretable. Since MHs are relatively complex target sequence features, we reason that they can serve as a good proxy for other still unknown target sequence features that the model may need to identify in order to predict repair outcome frequencies.

For this reason, we first introduces two simpler reductions of the repair outcome prediction problem, namely MH detection and MH length detection. In these prediction problems, the input of the model is only a single repair outcome. The outputs

in these problems related to MH properties of these repair outcomes that are explained in the following sections.

1) Microhomology detection: Deletion repair outcomes can be broadly split into two categories based on the use of MHs. Our proposed prediction problem of MH detection entails predicting whether a deletion is microhomology-based (MH-based) or not. A deletion is MH-based if it results from MH alignment. That is, when k nucleotides adjacent to the left side of the deletion are equal to the last k deleted nucleotides on the right side, with $k \geq 1$ (Figure 2a). The problem of predicting whether a deletion is microhomology-based or not can be defined as a binary classification task, based on an encoding of a repair outcome in its target sequence context (Section II-C). For this prediction problem, in theory, the model only has to consider two positions in the target sequence once it has identified the expected MH positions, because the length of the MH is irrelevant for a correct prediction.



(a) MH-based deletion. (b) Deletion not based on an MH.

Fig. 2: Two deletion repair outcomes where one is MH-based (2a) and the other is not (2b). Both subfigures show a target sequence of 13 nts before CRISPR lesion and DSB repair. The red highlight shows which nucleotides were deleted in each example repair outcome. The cut site indicates where a DSB is introduced by the CRISPR nuclease. Potential MH regions are indicated by the dashed square boxes.

2) Microhomology length detection: MH length is a feature with predictive power towards repair outcome prediction [10], [17]. Therefore, the second prediction problem we introduced was MH length detection. For a given repair outcome, the prediction output is defined as the length of the MH (i.e. the number of homologous nucleotides) on which that repair outcome is based. If the deletion is not MH-based, the expected output value is 0. Figure 2a shows an example of an MH of 2 nts, so where the expected output value is 2. This prediction problem was slightly more complex because it requires the model to compare more than one pair of nucleotides in the expected MH positions.

3) Repair outcome prediction: The output of this prediction problem was the probability distribution of all possible CRISPR outcomes given a target sequence and the set of possible outcomes. Here, the probability of occurrence of one repair outcome depends on the probabilities of all the other repair outcomes for a given target sequence. Therefore, repair outcome scores were calculated for every possible deletion and these scores were normalized to obtain a probability distribution. This probability distribution was then compared compared to the true observed probability distribution.

B. Data and preprocessing

Several experimental CRISPR outcome datasets are available to train models on [9]–[12]. These datasets contain records of observed repair outcomes and their frequencies obtained using CRISPR-Cas9 on large collections of target sequences. A library of pseudorandomly generated target sequences, including surrounding DNA context and paired with complementary gRNAs, are transduced in Cas9-expressing cells. Cas9-gRNA complexes form and introduce a DSB in the target sequence at a predetermined cut site. The cell subsequently repairs the DSB and the resulting DNA sequence in its surrounding context is amplified and sequenced to measure the frequency of insertions and deletions (indels) that have been introduced (Supplementary Figure 12). So in these databases each sample consists of the nucleotide composition of a target sequence and a set of indel frequencies.

In the present study, we chose to work with the FORECasT database [11] since this is the largest CRISPR outcome dataset. The dataset contained $41,630$ target sequences, but for the present research only the subsets "Explorative gRNA-Targets" and "Counterpart to gRNA in Conventional Scaffold gRNA-Targets,Explorative gRNA-Targets" and only the "FORWARD" strands were selected to ensure our dataset contained no duplicate target sequences. This brought the dataset size down to $24,849$ target sequences. We truncated the target sequences and their DNA context to a length of 8 nucleotides (nts) for the MH detection problem and 52 nts for MH length detection and repair outcome prediction. That is, we truncated the DNA context respectively 4 and 26 nts upstream and downstream of the targeted cut site. Since we did not want to include outcomes where the feature vector does not contain the features relevant for detecting MH (lengths), all outcomes where the deletion extends outside of the 52 nucleotide window were filtered out. Deletions where the MH lies beyond the 52 nucleotide window were filtered out for the same reason. For the repair outcome prediction task, we also excluded target sequences with fewer than 100 mutagenic reads within the 52 nucleotide window.

The dataset was split into a 80% train set and a 20% test set. To prevent information leakage from the train set to the test set, the split was made on the level of target sequences rather than on the outcome resolution, to ensure that no two outcomes from the same target sequence could appear in both the train and test set. Cross-validation splits (see Section II-G) were made in similar fashion for the same reason.

For the MHL prediction problem we defined the following five class labels: 0 for non-MH outcomes; and 1, 2, 3, and 4+ for MH-based outcomes with MHs of the corresponding lengths, respectively. As expected, there was a strong imbalance between classes because long MHs are typically scarce. The smallest class was 4+ with $47,847$ outcomes in our train set. Since this was enough data for this prediction task, we balanced the labels by randomly sampling from the other classes until we have $47,847$ outcomes per class as well, resulting in a dataset containing $239,235$ outcomes.

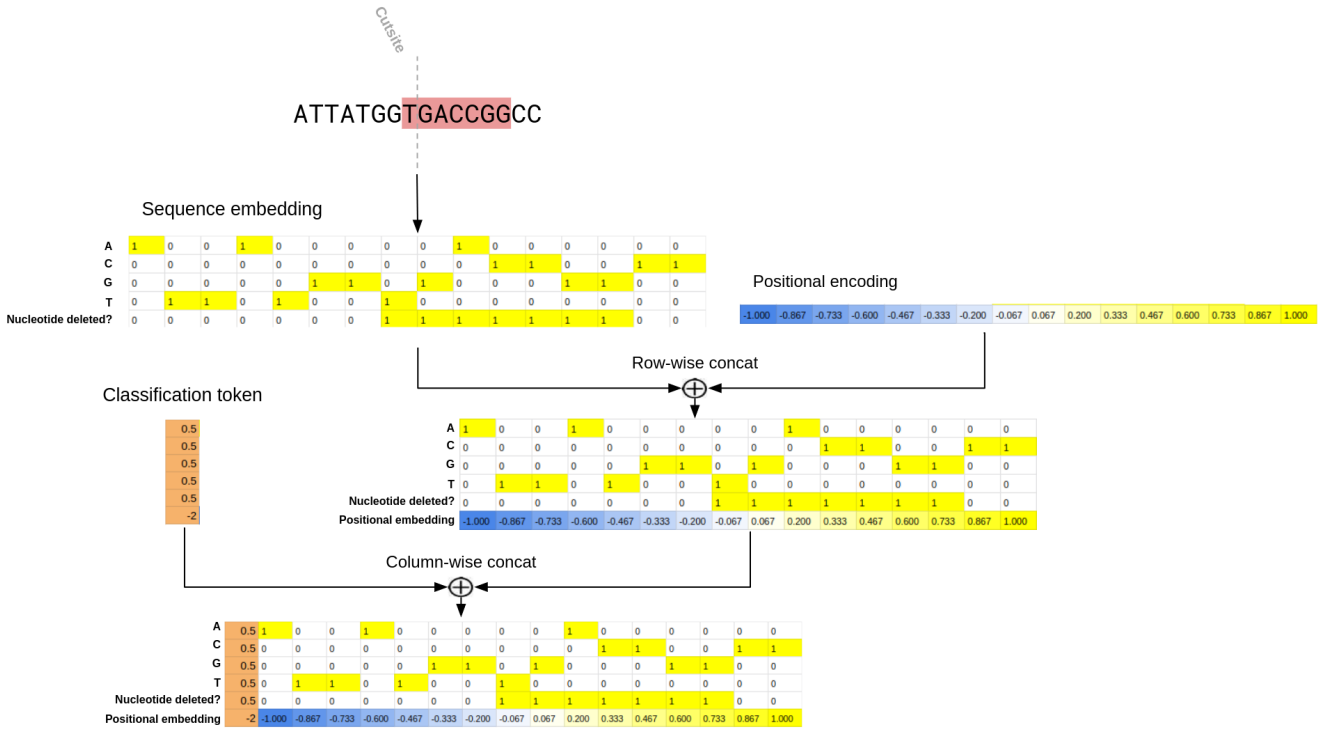


Fig. 3: Input embedding for a target sequence specific mutational outcome. On top an example target sequence is displayed with 16 nucleotides, and the red highlights show which nucleotides have been deleted in a specific repair outcome. This information was encoded position-wise into a sequence embedding, where each position contains a one-hot encoding of the nt, and a boolean encoding of the deletion status (0 = not deleted, 1 = deleted). The resulting matrix that encodes all positions was concatenated with a positional encoding, which was a linear interpolation between $[-1; 1]$ in n steps where n = target sequence length. The resulting matrix was prefixed with a classification token column containing values that were unique in their respective rows in the feature matrix.

C. Input representation: sequence embedding and positional encoding

The target sequence and a repair outcome were encoded into a single feature vector, which was one of our main contributions (Figure 3). Our prediction models make use of the following information about the samples repair outcomes: the nucleotide composition of the target sequence and outcome-specific information about the mutation (i.e. what nucleotides were deleted in a deletion), represented as a sequence embedding, and positional information of the nucleotides (i.e. the order), represented as a positional encoding. The attention operation in our architecture (Section II-D) is invariant to the order of the input [15], so the positional information is necessary to provide information about the order of the input.

The sequence embedding was a position-wise one-hot encoding of the nucleotide composition of the target sequence combined with a boolean encoding of the deletion status of each nucleotide in the given repair outcome (0 = not deleted, 1 = deleted). In the literature (e.a. [9], [13], [18]), it is common to one-hot encode k-mers of nucleotides, where multiple positions are encoded together as one of the 4^k possible k-mers. Instead, we encoded sequences on the single nucleotide resolution, using one of the possible 4-tuples for each position corresponding to each of the four possible nucleotides A, C, G, and T (similar to [25]). The use of k-mers can speed up processing time if they are used to decrease the sequence length. We chose to encode individual nucleotides because it increases the resolution of model interpretation, since features will then be on the single-nucleotide level.

The sequence embedding was extended with a positional encoding (PE) of the nucleotide. Specifically, we defined a vector of floats where each value indicated the position of a nucleotide in the sequence. This PE was concatenated row-wise to the sequence embedding (see Figure 3 caption for details). The values of the PE across the outcome embedding are a linear interpolation between $[-1; 1]$. This makes our PE symmetric around the cut site, since the DSB cut site is always centered in our samples. Because of this, the PE intrinsically informs about distance to the cut site. An alternative method of encoding positional information was considered where a PE is summed over all values in the feature vector, similar to [15], [21]. However, this approach inhibited the learning abilities of the model in our problem context. This was possibly because the PE obfuscated the signal in our comparatively small feature vector too much, or because our dataset size is much smaller than datasets generally considered in the NLP field. We favoured our linearly interpolated PE concatenated to the feature vector because we were able to obtain excellent

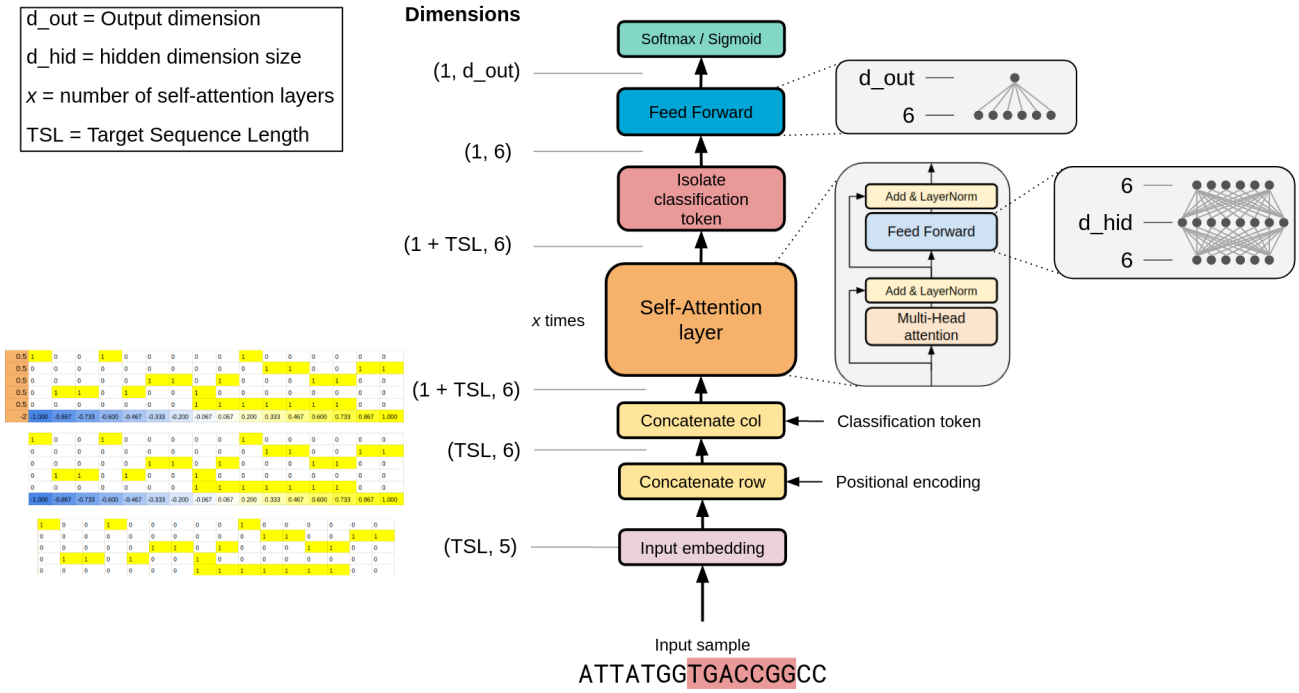


Fig. 4: Deep learning architecture of the models used in this paper (Section II-D). **Input embedding, concatenate row (PE), concatenate col (classification token)**: An input sample is first encoded into an embedding that contains the nucleotide composition and the nucleotide order of the target sequence and an encoding of the repair outcome (see Section II-C). **Self-Attention Layer**: a variable number of self-attention layers is applied to the embedding. **Isolate classification token**: after the final layer, the model output at the classification token position is isolated and the other positions are discarded. **Feed Forward**: a small linear layer maps the classification output to the desired output dimension.

performance on the MH detection and MH length detection prediction problems using it.

Finally, a classification token was prefixed to the feature vector, similar to [19]. This is a column containing values that are fixed across all samples and where all values are unique in their respective rows in the feature vector. This token was used to collect an output from the final layer of the model. This method reduced the dimension of the output of the self-attention layers, and improved model interpretability (see Section II-D).

D. Model architecture

A visual summary of the attention-based deep learning model we developed in the present research is shown in Figure 4. First, an embedding X of a repair outcome is created that contains the nucleotide composition of the target sequence, an encoding of the repair outcome, and the nucleotide order (see Section 1.3 for details). This embedding was passed forward through a series of self-attention layers. The number of self-attention layers was varied to adapt to the different prediction problems and to regulate model complexity. A self attention layer consisted of a multi-head attention operation and a feed forward layer with one hidden layer. Additionally, there were two residual connections over these respective layers.

The input embedding contained $l = TSL + 1$ positions x_i, where TSL was the target sequence length. The multi-head attention operation was applied to every position i in the input embedding separately and in parallel. The values of column x_i were divided over the 'heads' of the model. A 'head' refers to a unit of the model that performs the scaled dot-product attention operation to a subset of the values in x_i (see Section II-D1). The scaled dot-product attention outputs z_j for $j \in [0; \text{n\_heads}]$ were concatenated into one matrix Z. The multi-head attention output was calculated by multiplying Z with a learned weight matrix into a vector of the same dimension as input vector x_i.

The output of every self-attention layer was a hidden state that had the same dimensions as its input embedding, so a matrix with l tokens of 6 values. However, the goals of our predictions tasks were classification and regression, i.e. outputs of much smaller dimensionality. This was one of the reasons why a classification token was included in the input embedding. In the last hidden state of the model we isolated only the position of the classification token (so only the first column x_0 of the hidden state), similar to [19]. This token was forwarded to the last feed forward layer of the architecture, whose output dimension was adapted per prediction task. An alternative to using a classification token was considered, namely using an extra linear feed forward layer between the

output of the last self-attention layer and the input of the current feed forward layer. With this option, a larger portion of model calculation would have taken place in this feed forward layer. However, our model interpretation methods only applied to self-attention layers (see Section II-E). For this reason, model calculations outside of the self-attention layers were undesirable since they were out of sight for our interpretation methods. Using the classification token therefore improved the interpretability of our model.

The MH detection problem was a binary classification task, so the output dimension of the final layer was 1 and we regressed to a 1-dimensional value between 0 and 1 using a sigmoid. In the MH length detection problem we classified outcomes into 5 categories, so the output dimension was 5 and we softmaxed these values. In repair outcome prediction, the output dimension was 1 representing an outcome likelihood score. However, an outcome probability needs to be calculated in the context of a sample target sequence with ~ 550 possible outcomes. Therefore, all 550 outcomes of a sample were batched together and the model outputs were softmaxed to create a probability distribution.

1) What is attention: The scaled dot-product attention operation (Figure 5) is a fundamental component of Transformer networks. This operation was applied to all positions x_i in the embedding of a sample X for $i \in [0; l]$. From x_i three new vectors were calculated: q_i, k_i and v_i (query, key, value), that were linear mappings of x_i based on three learned weight matrices. Next, a score was calculated for all pairs x_i and x_j for $j \in [0; l]$ as: $\text{score}_{i,j} = x_i \cdot x_j$. These scores were divided by $\sqrt{d_k}$, where d_k is the dimension of k_i, and subsequently softmaxed into attention values $a_{i,j}$. The output o_i of the scaled dot-product operation was calculated as the weighted sum of all value vectors v_j:

$$o_i = \sum_{j=0}^{l} a_{i,j} \cdot v_j \tag{1}$$

where the weights are the attention values $a_{i,j}$. The following formula summarizes the complete scaled dot-product attention operation [15]:

$$\text{Attention(Q,K,V)} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

E. Attention visualization

The attention values a can be seen as importance scores for all pairs of positions x_i, x_j in the input. High similarity between query vector q_i and key vector k_j result in a high attention score $a_{i,j}$. This in turn means that the output value o_i will be influenced strongly by v_j. Therefore, if we display for a repair outcome the attention values a, we can start visualizing the relations between various positions in the target sequence that resulted in the predictions made by the model for a specific repair outcome. Figure 7a shows an example of such attention visualization in the context of the MH detection problem.

These attention values can be visualized for a single self-attention layer in a model, but our typical model has three or more self-attention layers. Attention values were averaged across the layers as follows:

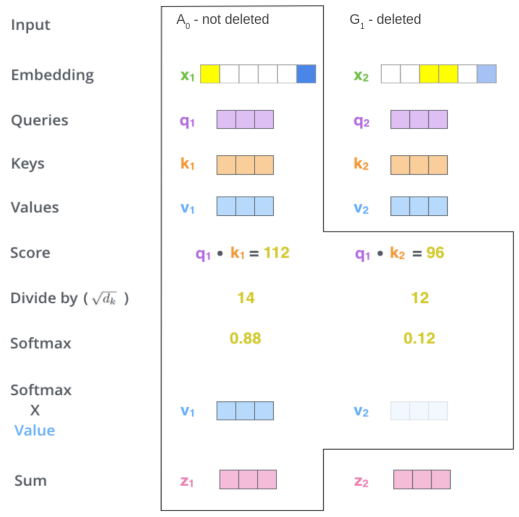


Fig. 5: Visualization of scaled dot-product attention. Figure adapted from Jay Allamar (The Illustrated Transformer)

$$a_{i,j} = \sum_{l=0}^{n\_layers} \frac{a_{i,j}}{n\_layers} \tag{3}$$

In addition to showing the attention value $a_{i,j}$ for every pair of positions i and j, we can also sum all outgoing attention values $a_j = \sum_{i=0}^{l} a_{i,j}$ per position j. These values were summed across the layers, showing the total outgoing attention of every position j throughout the model. Figure 7d shows an example of such attention visualization in the context of the MH detection problem.

Since outcomes were each encoded in their own embedding and our model could produce outputs for every outcome separately, we were able to analyze our model on the resolution of single outcomes.

F. Attention boxplotting

Attention visualization is useful for displaying model behaviour on the resolution of a single repair outcome prediction. However, it does not quantify the behaviour of the model across the dataset. For this reason we also generated attention boxplots that show attention values categorized by nucleotide type in categories as summarized in Figure 6. Specifically, for every outcome o in our dataset and for every position j in its embedding we calculated the average outgoing attention:

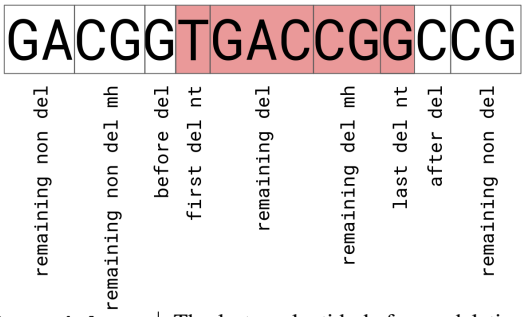
$$A_j = \sum_{i=0}^{l} \frac{a_{i,j}}{l} \tag{4}$$

where A_j is one data point displayed in the boxplots. Conceptually, we displayed the average influence strength that the value vector v_j has on any output o_i of a self-attention layer. Again, these values were available for every self-attention layer in the model, so we either displayed the attention values for a single layer or averaged across all layers of the

Prediction task	n_layers	n_heads	d_hid	d_out	window_size	n_parameters	n_epochs	batch_size	LR	γ
MH detection	3	1	6	1	8	835	500 *	64	0.0002	N/A‡
MH length detection	6	3	6	5	52	1691	200	64	0.0015	0.99
Repair outcome prediction	5	6	10	1	52	1689	400	16 $\approx$ 8800 †	0.0015	0.99

TABLE I: Hyperparameters for models used for each prediction task. **n_layers**: number of self-attention layers in the model. **n_heads**: number of attention heads in the model (see Section II-D). **d_hid**: Dimension of the hidden layer in each self-attention layer (Figure 4. **d_out**: output dimension of the final feed forward layer of the model. **window_size**: number of of nucleotides around the cut site that were included in the input embedding (sequence context). **n_parameters** number of learnable parameters in the model. **n_epochs**: the number of epochs the model was trained for. **batch_size**: number of samples in a minibatch. In MH detection and MH length detection a sample is one outcome, in repair outcome prediction a sample is the complete set of possible outcomes givne a target sequence. **LR**: learning rate the model was trained with. **gamma** (γ): accuracy of the trained model. * Convergence at 100% accuracy was already reached after 20 epochs. † We used a batch size of 16 for the repair outcome prediction problem, but in this problem context one sample consists of all the outcomes for a single target sequence. For each target sequence there are about 550 possible outcomes, so one batch contains about 8800 outcomes. ‡ No learning rate scheduling was used for the MH detection problem.

model. Separate boxplots were made for attention values that originated from outcomes that were MH-based and outcomes that were not.



`before del`	The last nucleotide before a deletion
`after del`	The first nucleotide after a deletion
`first del nt`	The first deleted nt
`last del nt`	The last deleted nt
`remaining del mh`	Remaining nucleotides that were deleted and also part of an MH
`remaining del`	Remaining nucleotides that were deleted and not part of an MH
`remaining non del mh`	Remaining nucleotides that were not deleted and were part of an MH
`remaining non del`	Remaining nucleotides that were not deleted nor part of an MH

Fig. 6: Example of an outcome for a target sequence where all nucleotides are labeled with their classes. The red hue indicates which nucleotides were deleted in this specific outcome. The various classes explained below in the table.

G. Training and evaluation

Our models were trained using minibatch stochastic gradient descent optimized with Adam [26]. For the MH length prediction problem and for the repair outcome prediction problem we used exponential learning rate (LR) scheduling to decay the LR to fine-tune model parameters as the model reaches convergence: $\mathrm{LR}(e, \gamma) = \mathrm{LR}_0 \cdot \gamma^e$, where e is the current epoch and γ is a hyperparameter regulating LR decay speed. The complete set of hyperparameters that were used for the models are listed and summarized in Table I. Hyperparameters were validated using 5-fold cross-validation. Some hyperparameters were optimized by searching a range of possible values and others were found by trial-and-error, see Results and discussion (Section III) for details.

For MH detection, which is a binary classification problem, we used binary cross-entropy as loss function. MH length detection is a multi-class classification problem so we used the cross entropy loss function. For the repair outcome prediction task, the model outputs a probability score per single outcome. These scores were batched together for all possible outcomes for a target sequence and softmaxed to create a probability distribution. This distribution was compared with the observed probability distribution using Kullback-Leibler divergence as a loss function.

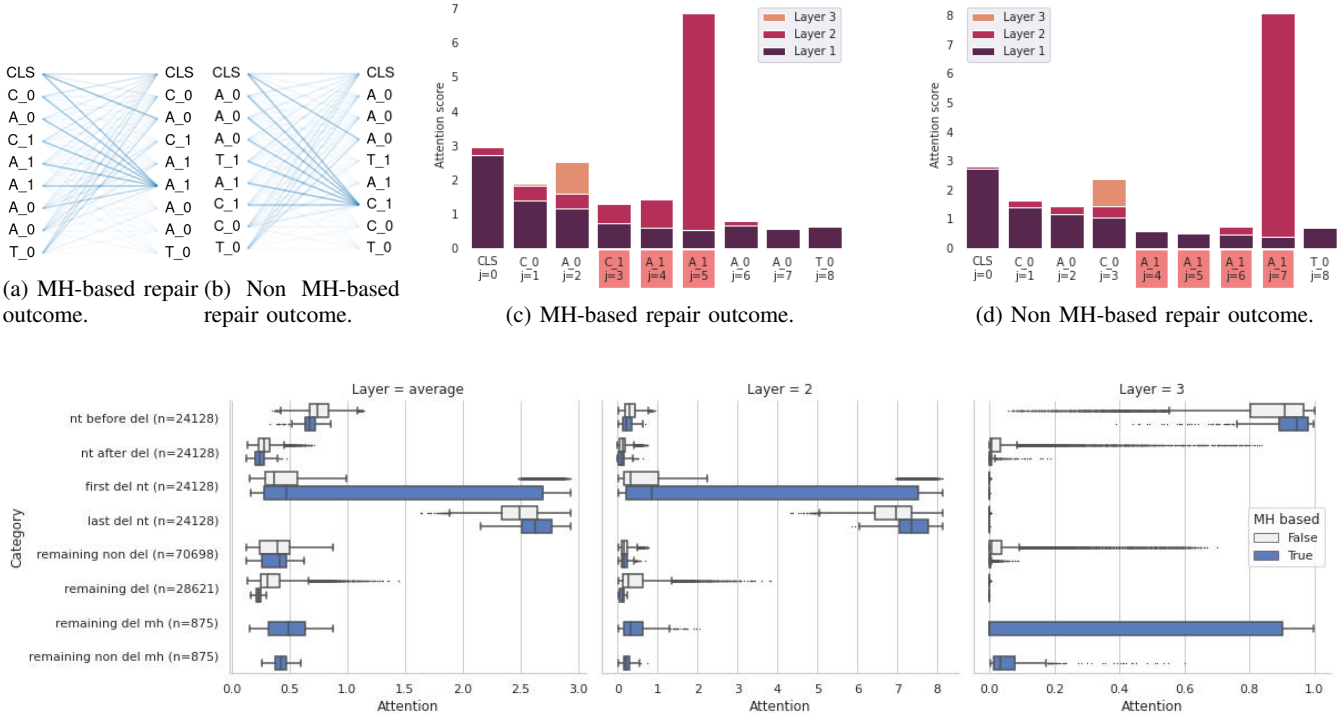
MH detection models were evaluated by calculating their accuracy on the test set. MH length detection models were also evaluated based on their test set classification accuracy. Repair outcome prediction models were evaluated by calculating the Pearson correlation per target sequence between predicted outcome frequencies and observed outcome frequencies. These correlation values were visualized using a violin plot. Correlation values from target sequences where the most frequent repair outcome was MH-based were split from correlation values where this was not the case.

III. Results & Discussion

For all three prediction tasks we trained models, evaluated their performance and analyzed their behaviour, which we discuss here.

A. MH detection

Our model was able to solve the MH detection problem, obtaining 100% accuracy on the validation set and on the test set. This result shows that a model with a limited number of self-attention layers is able to accurately distinguish between outcomes that are MH based and outcomes that are not. We were able to train models with only two self-attention layers that solved MH detection with 100% accuracy, but the results were hard to reproduce. Possibly, the capacity of a two layer model was not enough to obtain well-performing models reliably for this prediction problem. Training three layer models was more stable so we decided to use a three layer model.



(a) MH-based repair outcome. (b) Non MH-based repair outcome. (c) MH-based repair outcome. (d) Non MH-based repair outcome.

(e) Attention boxplots.

Fig. 7: Attention visualizations for the MH detection problem, as described in Sections II-E & II-F. **(a), (b):** Attention visualization of two samples for the MH detection problem. On the left side of each of these subfigures we see a vertical representation of a repair outcome. The letters indicate the nucleotide composition of the target sequence, and CLS is the classification token. The '_1' suffixes indicate nucleotides that were deleted in this repair outcome, and '_0' suffixes show non-deleted nucleotides. On the right side of (a) and (b) we see an identical representation of the same repair outcome. The line between two positions in the sequence i and j represent the attention value $a_{i,j}$ (see Section II-D1 & II-E), where a high opacity indicates a high attention value. The left side of a connection shows position i, so the position of the query value q_i and the position of the output o_i. The right side shows position j, the position whose key k_j is compared with q_i and whose v_j vector is used to build the value of output o_i. The model outputs strong attention scores for expected MH positions: the 'A_0' before 'C_1' on top and the 'A_1' before 'A_0' below. **(c), (d):** These subfigures show attention values for each position j, summed over all positions i: $A_j = \sum_{i=0}^{l} a_{i,j}/l$, and split per layer of the model. The red hue highlights nucleotides that were deleted in this each repair outcome. Again, we see high attention values for the expected microhomology positions ((c): $j = 2, 5$, (d): $j = 3, 7$). In the second layer, attention was high for the (expected) right MH position, and on the third layer, attention was high for the (expected) left MH position. **(e):** Boxplots showing attention values per nucleotide categorized as described in Figure 6, averaged across the layers of the model (left) or shown per layer (middle, right). See Section II-F for methods. On average, attention was high for expected MH positions (`nt before del` and `last del nt`) and for the first deleted nucleotide (`last del nt`).

1) Attention visualization for MH detection: As described in Section II-D1 we visualized the behaviour of a trained model on the resolution of single outcomes by visualizing the attention values it generated between positions in the target sequence. We visualized attention for two repair outcomes, one outcome that is MH-based (7a) and one outcome that is not (7d). For both repair outcomes we can see that the model is paying attention to the last nucleotide before the deletion and to the last deleted nucleotide. These positions are the expected positions for the MHs. In this controlled prediction problem, we know that those are important features for detecting MHs [5], [7], [17]. These results show that attention visualization can help to reveal important features in this problem context. However, we also note that in order to find the expected MH positions, model should somehow detect where the deletion starts and where it ends. The attention visualization does not seem to provide insight on that behaviour.

2) Attention boxplots for MH detection: We quantify the attention values of the model across the whole dataset as described in Section II-F (Figure 7e). In the averaged attention values across layers we see that indeed the attention is strong for the last deleted nucleotide (`last del nt`) and in a lesser degree also for the last nucleotide before the deletion (`nt`

before del). Interestingly, in the case of longer MHs the attention values are not strong for other nucleotides that are part of the MH (remaining del mh, remaining non del mh). This follows expectation, as for this prediction problem, the first nucleotide of the MH provides enough information to determine MH presence. We saw specialization of the second layer towards the right MH (last del nt) and a specialization of the third layer towards the left MH (nt before del). This result shows us that layers can specialize towards subtasks of a prediction problem and therefore it is important to analyze attention of different layers individually.

We saw a set of outliers in the first del nt category with high attention values. These were all deletions of length 2 where both deleted nucleotides are the same. That is, the embeddings of first del nt and last del nt were as similar as they could be in our input embedding. Apparently, this resulted in a high attention value not only for last del nt, but also for first del nt. Possibly, this was caused by the query vectors q of these nucleotides being similar as well. This would mean that the high attention values are merely an artifact caused by properties of the attention-based model, rather than having any biological meaning. This result highlights a limitation of linking model interpretation to biological context using attention values.

B. MH length detection

The MH length detection problem on a window size of 52 nts was more complex and therefore we started by optimizing our model for the prediction problem. We compared the performance of three possible options for the number of heads. The dimension of a position in the embedding dimension size must be divisible by the number of heads, since the values in the embedding are divided over the heads of the model (see Section 4). Since our embedding dimension is 6, we can choose $n\_heads \in \{1, 2, 3, 6\}$. In Figure 8a we compare the learning behaviour of models with 1, 3 and 6 heads. We see that the highest possible number of heads yields the best results in our prediction problem.

The number of outcomes in a batch also influenced the learning capabilities of the model (Figure 8b). A batch size of 64 is too small for the model to find a suitable gradient in the optimization landscape. Batch sizes of 256 and upwards, however, seem to slow down learning. These high batch sizes might even limit the final accuracy at convergence, although the limited number of epochs on this experiment prohibit us for making final conclusions on this matter. A batch size of 128 yields the best results.

As discussed in Section II-G, we used learning rate scheduling with a exponential decay. This requires hyperparameter optimization for the learning rate (LR) and the decay rate (γ). We found good performing models for $LR = 0.0015$ and $\gamma = 0.99$ (Figure 8c).

After optimization, the model attained near perfect performance, with 99.987% accuracy ($60280/60288$ correct predictions) on the test set. Our result shows that the complexity of our model is enough to determine MH length from the input embedding. MH length is an important predictor for the repair outcome prediction problem [10], [17].

1) Attention analysis for MH length detection: We analyzed how the model learned to handle this prediction problem. In Figure 8d we can see an example where the model learned to pay attention to nucleotides that are part of an MH or to nucleotides nearby. For this outcome, attention values in layer 3 were high for the edge of the MH ($j = 19$), and in layer 4 for MH nucleotides ($j = 19, 20, 49, 50$). These results suggest that layers could specialize in subproblems of the prediction problem, and that attention visualization can reveal this. Similar to Section III-A2 we quantified this behaviour by plotting attention scores for categorized nucleotides across the whole dataset (Figure 8e). For outcomes containing MHs we saw that attention scores averaged across layers (left plot) were higher for all the nucleotide categories that are part of the MH (nt before del, last del nt, remaining del mh and remaining non del mh). This is in contrast with Figure 7e, where the remaining del mh and remaining non del mh categories did not show strongly increased attention scores. This difference is explained by the prediction problem setup. For the MH detection problem, the model only needs to determine if the nucleotide before the deletion (nt before del) and the last deleted nucleotide (last del nt) are equal. If they are equal, then the outcome is MH based, and no other information is necessary. However, for detecting the length of an MH, the model needs to check all nucleotides that are part of the MH. Even the nucleotides adjacent to the MH are important, for example to determine the difference between an MH of length 2 or 3.

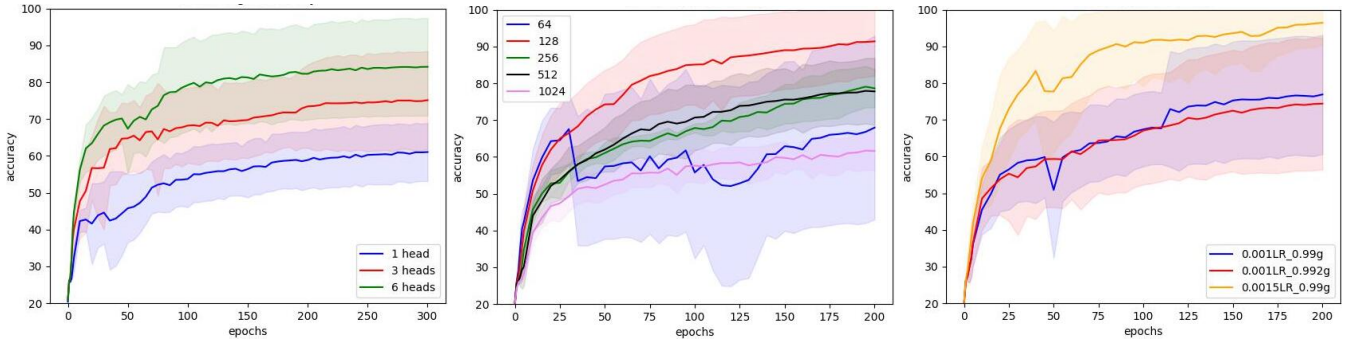
These results on this prediction problem, where the features are known, showcase how quantifying attention scores across a dataset provides insight about feature importance and in how an attention-based model works.

Another example where the attention analysis displays model behaviour is shown in the attention values of layer 1 in Figure 8e. The attention scores displayed are from the first layer of the MH length detection model. The attention scores of deleted nucleotides are much higher than other attention values. Therefore, it seems that this layer is specialized in detecting which nucleotides were deleted, which is also visible in the attention values for the repair outcome shown in Figure 8d.

Note that the MH length detection prediction problem was defined with 5 class labels, where the last class label encompasses MH-based outcomes with MH lengths of 4 or longer. In this setup the model can make correct predictions for MHs longer than 4 nts based only on their first 4 nts. These target sequences thus contain nucleotides that are part of the MH but that are not important for the prediction task. However, in our attention analyses, these nucleotides are categorized in mh classes, which may be obfuscating our results.

C. Repair outcome prediction

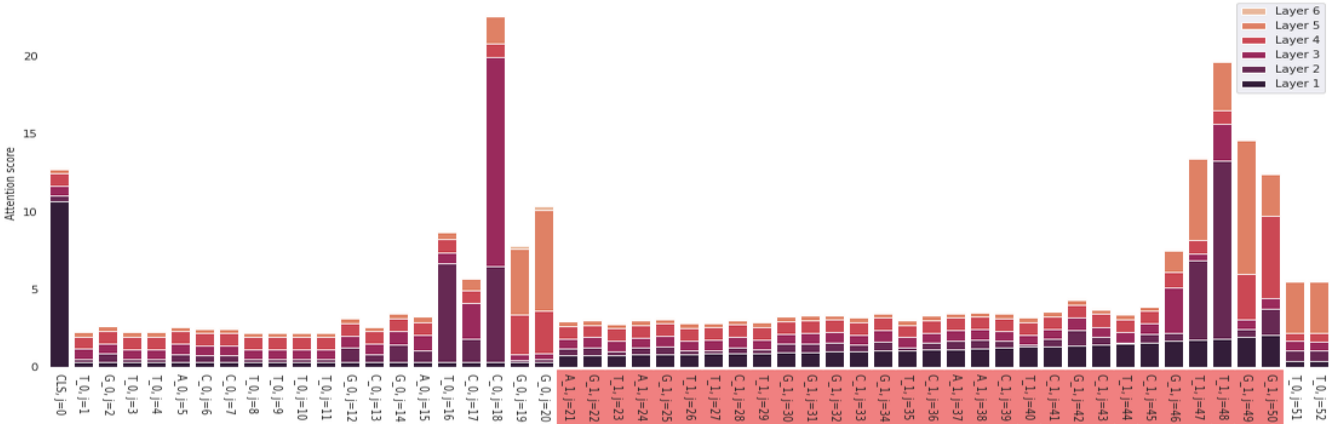
The optimized models for the MHL prediction problems provided us a starting point for what model complexity was needed to capture MH features in a the target sequence and what other hyperparameter settings could be fitting for the frequency prediction problem.



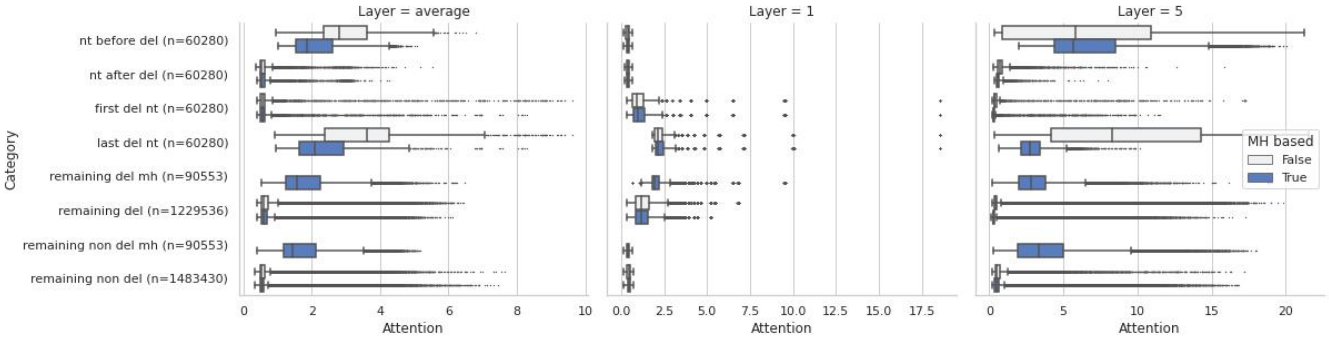
(a) Learning curves of number of heads optimization for the MH length detection problem.

(b) Learning curves of batch size optimization for the MH length detection problem.

(c) Learning curves of learning rate (LR) and gamma (g) optimization for the MH length detection problem. The curves show validation set accuracy for three combinations of settings for learning rate LR and g.



(d) Attention values for each position j, summed over all positions i: $A_j = \sum_{i=0}^{l} a_{i,j}/l$, and split per layer of the model. The red hue highlights nucleotides that were deleted in this each repair outcome. See Section II-E for details about the method. Attention values were high for all expected MH positions ($j = 17-20, 47-50$) and for the nucleotides before ($j = 16, 46$). Positions $j = 19, 20, 49, 50$ are MHs, and we saw especially high attention values for MH-adjacent positions ($j = 18, 48$).



(e) Microhomology length prediction attention boxplots. The attention values are categorized per nucleotide as described in Figure 6. See Section II-F for methods. On average, attention values were high for all MH related classes (`nt before del`, `last del nt`, `remaining del mh` and `remaining non del mh`). Attention values in layer 1 were relatively high only for deleted nucleotides. Layer 5 attention values were high for expected MH positions (`nt before del`, `last del nt`).

Fig. 8: Attention visualizations for the MH length detection problem, as described in Sections II-E & II-F

1) Model complexity: We trained models with varying numbers of self-attention layers (n_layers $\in \{4, 5, 6, 7\}$) on the repair outcome prediction problem (Figure 9). We chose this set of options for number of layers in the model because this was similar to model complexity that was needed for the MH length detection problem, and it was similar to the complexity of an attention-based model that was used for a similar prediction task [18], [21]. The model with the highest complexity had the lowest loss after 200 epochs (mean=0.00457, stdev=0.000388, KL divergence). The other three models had comparable losses after 200 epochs (mean=0.00479, stdev=0.000318, mean=0.00472, stev=0.000223, mean=0.00489, stdev=0.000283, KL divergence, 4, 5, and 6 layer model, respectively). The losses of the 5 layer model and the 7 layer model did not differ significantly ($p = 0.754$, Kruskal-Wallis).

Note that 200 epochs was not enough to reach convergence, as the loss values were still decreasing at that point. Because of time constraints and computational limitations, we chose to train a 5 self-attention model for more epochs (400). Using the 5 layer model also limited the model complexity compared with a 7 layer model, albeit possibly at the price of lower performance.

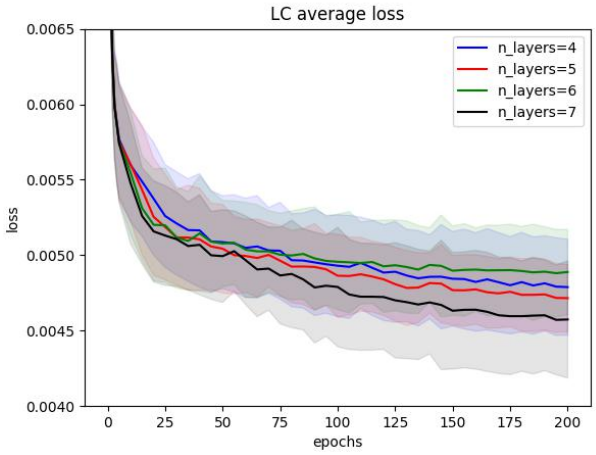
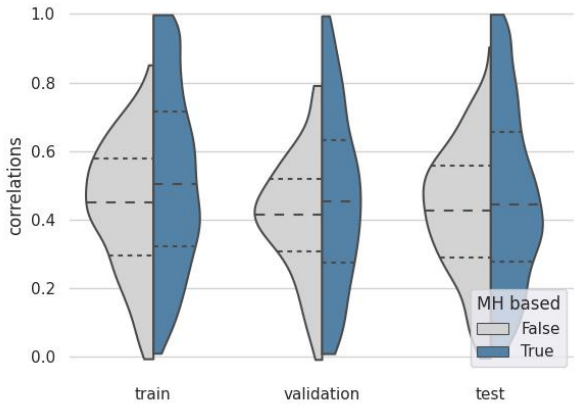


Fig. 9: Learning curves for models trained on the repair outcome prediction problem using varying number of self-attention layers. Lines show average KL Divergence loss on the validation set and hues show standard deviation over 5-fold cross-validation.

2) Correlations violin plots: In Figure 10 we show violin plots of Pearson correlation values between predicted outcome frequencies and observed outcome frequencies. Each correlation value was calculated over one sample, which was the set of all possible outcomes for one target sequence. In its present form, the model was not on par with state-of-the-art performance. The average correlation on test samples was 0.460 (Pearson's R), whereas for example Lindel [9] achieved 0.70 on their dataset. Note that because these models were trained on different datasets which can have different properties, the correlation values might not be directly comparable. Across all datasets, the correlation values varied from $0 - 1$ (Figure 10), which implied that the models can make inaccurate predictions and accurate predictions. The model performed better on the train set ($R = 0.505$) than on the validation and test set ($R = 0.453$ and $R = 0.460$, respectively), suggesting that the model did not generalized completely.

The samples in the violin plots were categorized by whether their most frequent outcome was MH-based or not. Our model performed slightly better for MH based outcomes. Samples where the model predictions correlated most strongly with the labels ($R > 0.9$) were almost exclusively samples where one outcome based on a long (>8 nt) MH was observed with a relatively high frequency and all other outcomes were relatively infrequent. This result highlights that the model was able to distinguish outcomes with long MHs from other outcomes and to link this with high outcome likelihood. Conversely, the model did not make accurate predictions for samples with high-frequent outcomes that were not MH-based, suggesting that the model was not able to find and use other features to make accurate predictions for these outcomes.

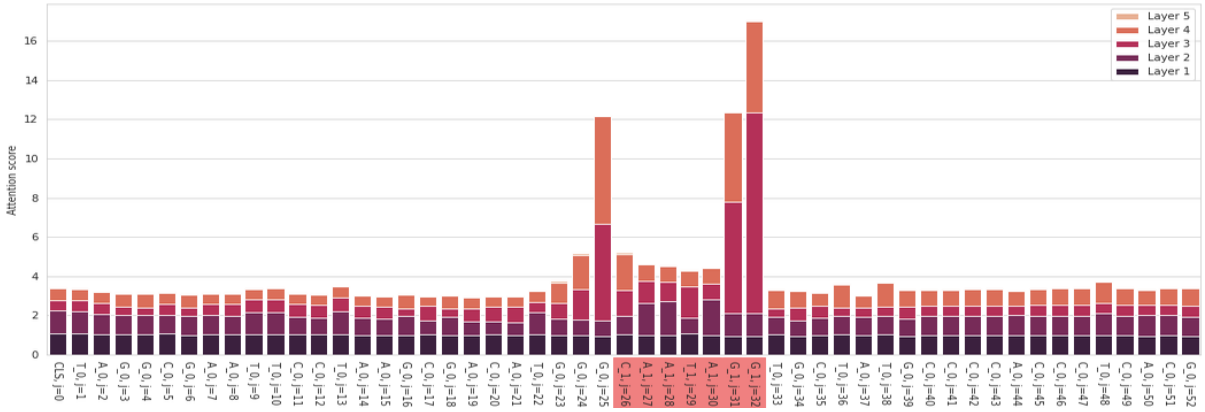
Because of the difference in model performance between MH-based and non-MH samples, model performance could possibly be improved with an approach like [10], where MH-based and non-MH samples are modeled separately. This would also provide an opportunity to interpret these models separately, perhaps providing insights in individual DNA repair mechanisms.



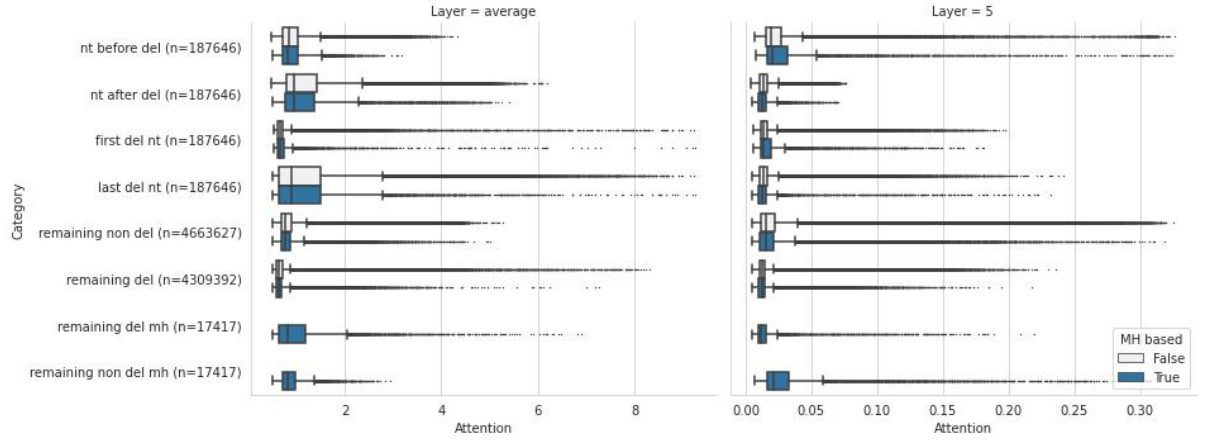
Dataset	MH-based	Non-MH	Both
Train (avg. R)	0.520 (N=1305)	0.435 (N=295)	0.505 (N=1600)
Validation (avg. R)	0.463 (N=325)	0.409 (N=75)	0.453 (N=400)
Test (avg. R)	0.470 (N=1622)	0.417 (N=364)	0.460 (N=1986)

Fig. 10: Violin plot of Pearson correlation values per sample between predicted outcome frequencies and observed outcome frequencies. The average correlation values are listed in the table. Samples in datasets were categorized by whether their most frequent outcome was MH-based or not.

3) Attention visualizations: In Figure 11a we show a visualization of only the outgoing attention of each nucleotide for one typical MH-based repair outcome. We saw high attention



(a) Attention values for each position j, summed over all positions i: $A_j = \sum_{i=0}^{l} a_{i,j}/l$, and split per layer of the model. See Section II-E for details about the method. The red hue highlights nucleotides that were deleted in this each repair outcome. The outcome shown was MH-based with a MH length of 2. We saw increased attention scores for positions that are part of the MH ($j = 24, 25, 31, 32$).



(b) Attention boxplots for the repair outcome prediction problem. The attention values are categorized per nucleotide as described in Figure 6. See Section II-F for methods. Note that the boxplots was made on a random sampled 20% subset of the test set (∼400 target sequences) because the number of data points was otherwise too large for our system.

Fig. 11: Attention visualizations for the repair outcome prediction problem.

scores for positions that are part of the MH. This observation suggests that the model has learned to use MHs to make outcome frequency predictions. The same can be observed in Supplementary Figures 14a, 14b & 14c

4) Attention boxplots: In Figure 11b we quantified the attention scores for various nucleotide types across the dataset in a similar fashion as in Figure 8e. We saw relatively high attention scores for the categories `nt after del` and `last del nt`. Other MH categories also displayed heightened attention values (`nt before del`, `remaining del mh`, `remaining non del mh`), albeit to a lesser extend. For the MH categories, we expected these results, since MHs are important predictors for CRISPR outcomes [5], [7], [10], [17]. For the `nt after del` category, we did not initially expect high attention values. Possibly, the first nucleotide to the right of the right edge of a deletion had some predictive power towards predicting CRISPR outcome.

We found that the fifth layer specialized towards the left MH part (`remaining non del mh`, `nt before del`), although the effect size is small. The remaining attention scores were distributed more uniformly across the categories compared to what we have seen in the other prediction problems. Again, this behaviour can be explained by the relation between MHs and the prediction task. For detecting MH or MH length, this connection was seemingly stronger than the relation between MHs and outcome frequency.

We created a similar boxplot exclusively for samples in the test set with strong correlations ($R > 0.85$) between predicted and observed outcomes and exclusively for outcomes where the predicted and true frequencies were above 25% (Supplementary Figure 13). These samples were almost exclusively MH-based (see Section III-C2). We saw that in this sample subset, high attention scores were less common in non-MH nucleotide categories. Indeed, the model focuses more strongly on MH nucleotides for these MH-based samples.

IV. Conclusion & Future Outlook

In the present work we presented an attention-based deep learning architecture that predicts CRISPR repair outcomes

using an embedding that combined local DNA sequence context with repair outcome specific properties. The performance of our model was not on par with existing models. Yet, our work can serve as starting point for developing attention-based models with a more competitive performance. Attention visualization methods revealed how our model behaved in CRISPR outcome prediction. We introduced some additional attention visualisation methods and validated them using simplified reductions of the prediction problem. We showed that indeed features that were previously known to be important for repair outcome prediction received high attention scores. However, our analyses also revealed some limitations of using attention values for model interpretation. We did not identify new target sequence features with predictive power for determining repair outcome frequencies. Possibly, attention values were not indicative of CRISPR outcome dynamics because our model displayed limited performance. Namely, if the model did not learn to properly predict CRISPR outcomes, than analyzing the behaviour of the model only has limited meaning towards understanding the prediction problem. The problem of using attention values for exploring model behaviour may also be a more fundamental one, as some recent research suggests that attention values do not reliably correlate with feature importance [22].

In our work we suggested that the attention mechanism could be suitable for this prediction task. However, we were not able to confirm or disprove that attention-based models would be able to achieve state-of-the-art performance. A recent study that was published during our research obtained excellent performance using an deep learning framework based on attention and BiLSTM layers [14].

For future work, we propose fine-tuning the model training process and hyperparameters settings for the repair outcome prediction task. We did not attempt a very broad range of hyperparameter settings for the repair outcome prediction problem. Our model might be underfitting the data, since the average correlations between predicted and observed outcome frequencies were much lower than models in the literature that used similar data. Possibly, for example increasing the number of layers or increasing the hidden dimension size would further improve model performance, albeit at the cost of higher complexity and thus reduced interpretability. However, we do reason that model based mainly on self-attention layers could be more interpretable in comparison to a mixed model of attention and BiLSTM layers. We suggest attempting to identify new features in local target sequence context that influence repair outcome probabilities by exploring attention values. This could provide leads for discovering novel characteristics of CRISPR induced DSBs dynamics and subsequent DNA repair. Our work forms a basis for such attention-based modeling of CRISPR outcomes.

REFERENCES

[1] Te-Wen Lo, Catherine S Pickle, Steven Lin, Edward J Ralston, Mark Gurling, Caitlin M Schartner, Qian Bian, Jennifer A Doudna, and Barbara J Meyer. Precise and heritable genome editing in evolutionarily diverse nematodes using TALENs and CRISPR/cas9 to engineer insertions and deletions. *Genetics*, 195(2):331–348, October 2013.

[2] F Ann Ran, Patrick D Hsu, Jason Wright, Vineeta Agarwala, David A Scott, and Feng Zhang. Genome engineering using the CRISPR-cas9 system. *Nature Protocols*, 8(11):2281–2308, October 2013.

[3] Leonela Amoasii, John C. W. Hildyard, Hui Li, Efrain Sanchez-Ortiz, Alex Mireault, Daniel Caballero, Rachel Harron, Thaleia-Rengina Stathopoulou, Claire Massey, John M. Shelton, Rhonda Bassel-Duby, Richard J. Piercy, and Eric N. Olson. Gene editing restores dystrophin expression in a canine model of duchenne muscular dystrophy. *Science*, 362(6410):86–91, October 2018.

[4] Hyeon-Ki Jang, Beomjong Song, Gue-Ho Hwang, and Sangsu Bae. Current trends in gene recovery mediated by the CRISPR-cas system. *Experimental Molecular Medicine*, 52(7):1016–1027, July 2020.

[5] Megan van Overbeek, Daniel Capurso, Matthew M. Carter, Matthew S. Thompson, Elizabeth Frias, Carsten Russ, John S. Reece-Hoyes, Christopher Nye, Scott Gradia, Bastien Vidal, Jiashun Zheng, Gregory R. Hoffman, Christopher K. Fuller, and Andrew P. May. DNA repair profiling reveals nonrandom outcomes at cas9-mediated breaks. *Molecular Cell*, 63(4):633–646, August 2016.

[6] Jia Shou, Jinhuan Li, Yingbin Liu, and Qiang Wu. Precise and predictable CRISPR chromosomal rearrangements reveal principles of cas9-mediated nucleotide insertion. *Molecular Cell*, 71(4):498–509.e4, August 2018.

[7] Anob M. Chakrabarti, Tristan Henser-Brownhill, Josep Monserrat, Anna R. Poetsch, Nicholas M. Luscombe, and Paola Scaffidi. Target-specific precision of CRISPR-mediated genome editing. *Molecular Cell*, 73(4):699–713.e6, February 2019.

[8] Brenda R. Lemos, Adam C. Kaplan, Ji Eun Bae, Alexander E. Ferrazzoli, James Kuo, Ranjith P. Anand, David P. Waterman, and James E. Haber. CRISPR/cas9 cleavages in budding yeast reveal templated insertions and strand-specific insertion/deletion profiles. *Proceedings of the National Academy of Sciences*, 115(9):E2040–E2047, February 2018.

[9] Wei Chen, Aaron McKenna, Jacob Schreiber, Maximilian Haeussler, Yi Yin, Vikram Agarwal, William Stafford Noble, and Jay Shendure. Massively parallel profiling and predictive modeling of the outcomes of CRISPR/cas9-mediated double-strand break repair. *Nucleic Acids Research*, 47(15):7989–8003, June 2019.

[10] Max W. Shen, Mandana Arbab, Jonathan Y. Hsu, Daniel Worstell, Sannie J. Culbertson, Olga Krabbe, Christopher A. Cassa, David R. Liu, David K. Gifford, and Richard I. Sherwood. Predictable and precise template-free CRISPR editing of pathogenic variants. *Nature*, 563(7733):646–651, November 2018.

[11] Felicity Allen, Luca Crepaldi, Clara Alsinet, Alexander J. Strong, Vitalii Kleshchevnikov, Pietro De Angeli, Petra Páleníková, Anton Khodak, Vladimir Kiselev, Michael Kosicki, Andrew R. Bassett, Heather Harding, Yaron Galanty, Francisco Muñoz-Martínez, Emmanouil Metzakopian, Stephen P. Jackson, and Leopold Parts. Predicting the mutations generated by repair of cas9-induced double-strand breaks. *Nature Biotechnology*, 37(1):64–72, November 2018.

[12] Ryan T. Leenay, Amirali Aghazadeh, Joseph Hiatt, David Tse, Judd F. Hultquist, Nevan Krogan, Zhenqin Wu, Alexander Marson, Andrew P. May, and James Zou. Systematic characterization of genome editing in primary t cells reveals proximal genomic insertions and enables machine learning prediction of CRISPR-cas9 DNA repair outcomes. *BioRxiv*, August 2018.

[13] Victoria R Li, Zijun Zhang, and Olga G Troyanskaya. CROTON: an automated and variant-aware deep learning framework for predicting CRISPR/cas9 editing outcomes. *Bioinformatics*, 37(Supplement_1):i342–i348, July 2021.

[14] Xiuqin Liu, Shuya Wang, and Dongmei Ai. Predicting CRISPR/cas9 repair outcomes by attention-based deep learning framework. *Cells*, 11(11):1847, June 2022.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics, 2019.

[17] Sangsu Bae, Jiyeon Kweon, Heon Seok Kim, and Jin-Soo Kim. Microhomology-based choice of cas9 nuclease target sites. *Nature Methods*, 11(7):705–706, June 2014.

[18] Jim Clauwaert and Willem Waegeman. Novel transformer networks for improved sequence labeling in genomics. *IEEE/ACM Transactions*

on Computational Biology and Bioinformatics, 19(1):97–106, January 2022.

[19] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, February 2021.

[20] Jesse Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 2019.

[21] Jim Clauwaert, Gerben Menschaert, and Willem Waegeman. Explainability in transformer models for functional genomics. *Briefings in Bioinformatics*, 22(5), April 2021.

[22] Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics.

[23] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation, 2019.

[24] K. K. Chiruvella, Z. Liang, and T. E. Wilson. Repair of double-strand breaks by end joining. *Cold Spring Harbor Perspectives in Biology*, 5(5):a012757–a012757, May 2013.

[25] Ramzan Umarov, Hiroyuki Kuwahara, Yu Li, Xin Gao, and Victor Solovyev. Promoter analysis and prediction in the human genome using sequence-based deep learning models. *Bioinformatics*, 35(16):2730–2737, January 2019.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

V. Supplementary Material

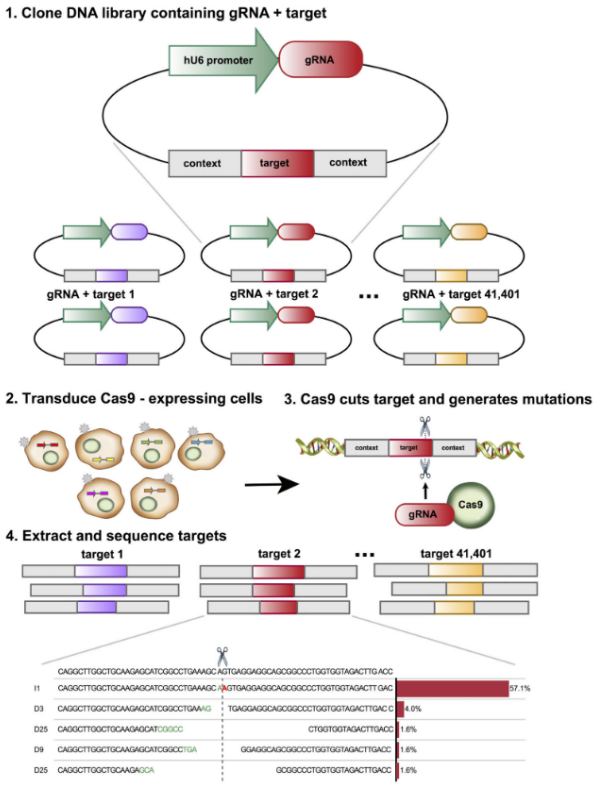


Fig. 12: Schematic overview of DNA library transduction and high-throughput measurements of indels (taken from [11])

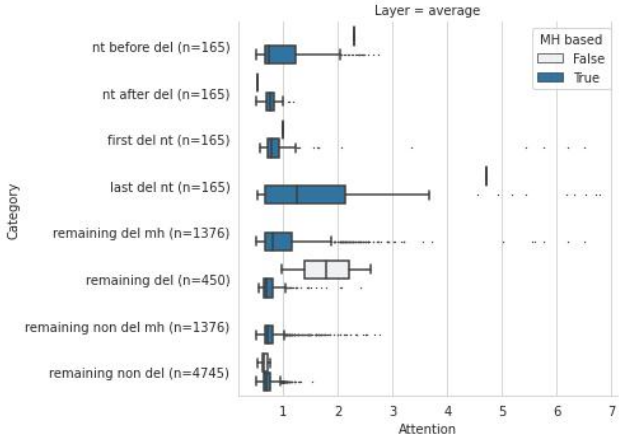
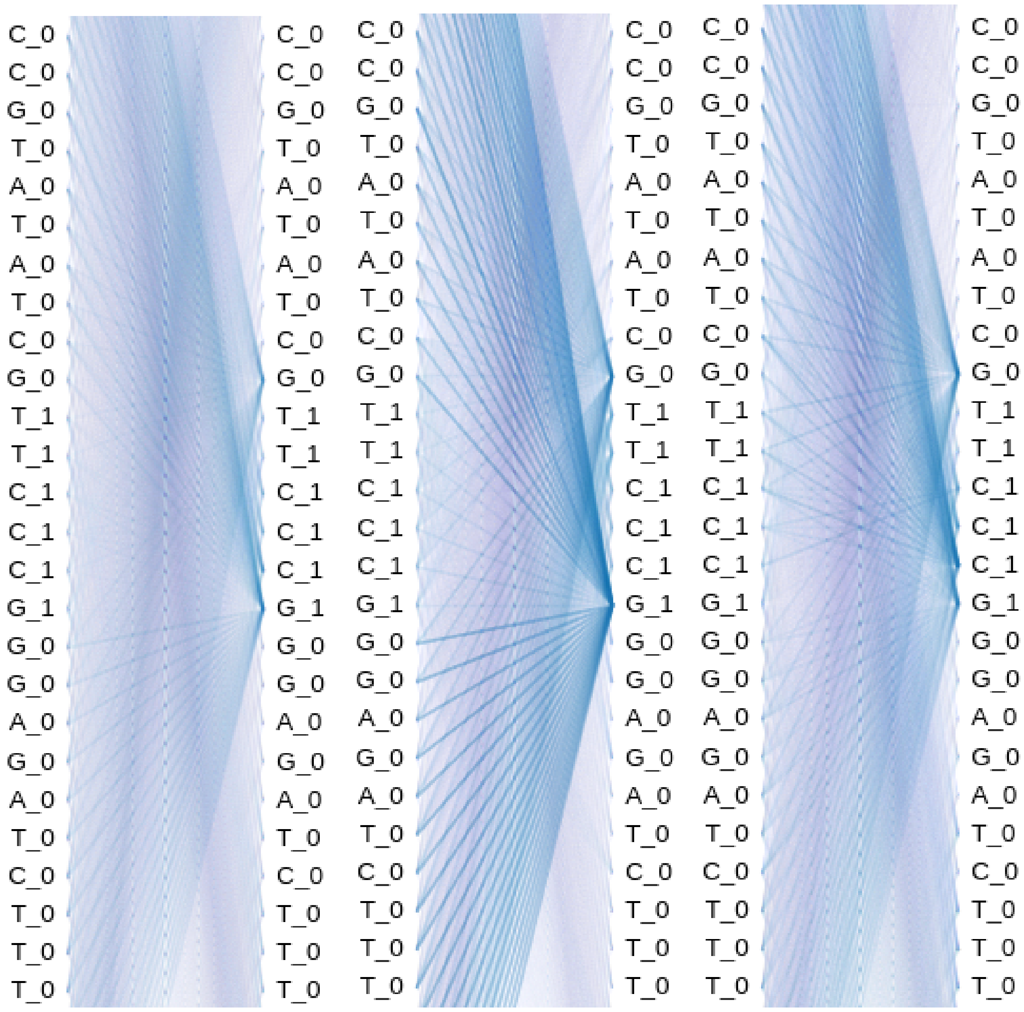


Fig. 13: Frequency attention boxplot, averaged across layers, only MH-based samples where the correlation between observed and predicted outcomes is higher than 0.85 and only outcomes where the observed and predicted frequency were above 25% (162 out of 1986 target sequences in the test set, 165 outcomes)



(a) Attention visualization for frequency prediction problem. Averaged attention across layers.

(b) Attention visualization for frequency prediction problem. Layer 3 attention.

(c) Attention visualization for frequency prediction problem. Layer 4 attention.

Fig. 14: Attention visualization for frequency prediction problem. Note that these figures are truncated on the top and bottom part. The actual window size is 52 nts. See Section II-E and Figure 7 caption for details on the attention visualization method.