Generalisable Methods for Improving CRISPR Efficiency and Outcome Specificity using Machine Learning Algorithms

Aidan Ronald O'Brien

November 2020

A thesis submitted for the degree of Doctor of Philosophy of
The Australian National University

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Aidan Ronald O'Brien

November 2020

Acknowledgements

I am enormously grateful to my chair supervisor Gaetan Burgio. His expertise in the CRISPR field and eye for detail really helped me apply my computational modelling to the real-world field of gene editing. I am also grateful to his group for the elusive data they have provided me over the years. I appreciate the John Curtin School of Medical Research for accepting me as a PhD candidate despite conducting most of my work away from the ANU.

I am also enormously grateful to my supervisor Denis Bauer. Her instinct for innovation and drive for outcomes has been a fantastic influence for producing tangible outcomes. Working with her and the team at the CSIRO has been a pleasure.

I would also like to thank Tamás Fischer and Dan Andrews. Their insightful feedback from my panel meetings was valuable in helping me see to see the bigger picture and to make my work more robust.

I am thankful to EMBL for sponsoring my visit to Heidelberg to attend their 2017 PhD symposium. It was a wonderful opportunity to meet other PhD students and to network with groups across Germany.

I would of course like to thank my friends both near and far. I think it is safe to say that I owe you a round of beers (or a hundred) now that my days of being a student are finally at an end!

Finally, I would like to thank my family. Especially my parents for their support over the years. It has meant a lot to me.

This research is supported by a JCSMR Scholarship and a CSIRO top-up scholarship.

Abstract

CRISPR (clustered regularly interspaced short palindromic repeats) based genome editing has become a popular tool for a range of disciplines, including microbiology, agricultural science, and health. Driving these applications is the ability of the “programmable” system to target a predefined location in the genome. A single guide RNA (sgRNA) defines the target through Watson-Crick base pairing, and a class 2 type II CRISPR associated protein 9 (Cas9) nuclease cleaves the target, resulting in a double-strand break (DSB). This activates DNA repair, and depending on the repair pathway initiated, can result in arbitrary insertions/deletions or a predefined variant.

Despite the versatility and ease of design enabled by this RNA-guided nuclease, it lacks specificity, regarding off-target effects, and efficiency, regarding the rate of successful editing outcomes. The overarching hypothesis of my thesis is to solve the disadvantages of CRISPR systems by using machine learning to train generalisable models on existing and novel datasets.

One pathway that demonstrates the need for prediction models is homology directed repair (HDR). HDR enables researchers to induce nearly any editing outcome, however, it is inefficient. And with an incomplete knowledge of its kinetics, no models existed for predicting its efficiency. I generated a novel dataset representing the efficiency of HDR. Using the Random Forests algorithm, I identified the sgRNA and the 3’ region of the template to modulate HDR efficiency. This novel finding relates to the kinetics of template interaction during HDR repair.

Even with efficient gene editing, a potential problem is unwanted side effects, such as embryonic lethality. This can be solved by using CRISPR to create conditional knockout alleles, to control when and where knockouts occur. To investigate the efficiency of this process, I used statistical analyses and the Random Forest algorithm to analyse a dataset generated by a consortium of 19 laboratories. I identified the inherent inefficiency of this method as defined by the efficiency of two simultaneous HDR events. Other experimental variables, like reagent concentrations or technician skill level, had no significant influence on efficiency. Because of the unrivalled versatility of this method, I created a statistical model for forecasting the efficiency of this technique from a low number of attempts, aiming to overcome its inherent inefficiency.

While Cas9 is the most cited CRISPR system, alternative CRISPR systems can further expand the gene editing repertoire. To support the uptake of the more-recent Cas12a, I performed a

comprehensive comparison between the two nucleases. I found support for Cas12a having a superior specificity. Despite this, editing outcome and efficiency prediction tools for Cas12a were scarce. Aiming to address this, I trained a Cas12a cleavage efficiency prediction model on representative data. This outperformed the current top model despite the dataset being 300x smaller, demonstrating the importance of clean data.

Altogether, this thesis improves the knowledge of different CRISPR gene editing techniques. These findings can enable researchers to design efficient experiments as well as provide researchers guidance where certain techniques may be inherently inefficient. As well as resulting in CUNE (Computational Universal Nucleotide Editor) and Cas12aRF, it also identifies the generalisability of prediction models due to the high degree of influence on efficiency by the sgRNA and repair template design.

Abbreviations

AUC	area under the curve
Cas	CRISPR-associated protein
CNN	convolutional neural network
CRISPR	clustered regularly interspaced short palindromic repeats
crRNA	CRISPR RNA
CUNE	Computational Universal Nucleotide Editor
dCas9	dead Cas9
DSB	double-strand break
ES cells	embryonic stem cells
gRNA	guide RNA
GWAS	genome-wide association study
HDR	homology directed Repair
ML	machine learning
MMEJ	microhomology-mediated end joining
MSE	mean squared error
NHEJ	non-homologous end joining
OOB	out-of-bag
PAM	protospacer adjacent motif
pre-crRNA	precursor CRISPR RNA
ROC	receiver operating characteristic
sgRNA	single guide RNA
SNV	single nucleotide variant
SRA	Sequence Read Archive
ssODN	single-stranded oligodeoxynucleotide
TALE	transcription activator-like effector
TALEN	transcription activator-like effector nuclease
tracrRNA	trans-activating CRISPR RNA
ZFN	zinc finger nuclease

Contents

Chapter 1 – Introduction

The ability to target precise DNA sequences is essential for a range of disciplines, from molecular biology to gene therapy. In molecular biology, the ability to knock out specific genes is essential for evaluating their function (Capecchi, 2005). In gene therapy, the ability to target specific alleles enables the ability to modulate gene expression (Danda et al., 2013) or the ability to insert novel DNA (Strecker et al., 2019).

At its core, genome engineering has two requirements:

- the ability to target a specific location and
- the ability to introduce the required modification.

These two steps are tightly coupled, with DNA targeting often resulting in inevitable modifications. An early example is a technique that was used to correct mutations through homologous recombination (Thomas et al., 1986). This was via a synthetic DNA template, homologous to the target. Because of this homology, the template's presence was sufficient for recombination to occur between it and the target, although this occurred infrequently. Later experiments have demonstrated methods to stimulate homologous recombination or other mechanisms to markedly improve efficiency, such as by inducing DNA damage.

1.1 Cleaving DNA to stimulate genetic modifications

The first targeted genome experiment in mouse chromosomes to demonstrate this concept used a homing endonuclease, I-SceI, to target a specific genomic region (Rouet et al., 1994). I-SceI, like other homing endonucleases, has the ability to recognise and cleave DNA, based on nucleotide sequence, resulting in a double-strand break (DSB) (Plessis et al., 1992). Uncorrected, DSBs are deleterious, potentially resulting in large deletions or chromosomal translocations (Frankenberg-Schwager et al., 1985). Because of this, cells have mechanisms to detect and aim to repair DSBs, through endogenous repair pathways (Valerie & Povirk, 2003).It is this repair event that was exploited to edit the target. Because, in mammalian cells, targeted DNA cleavage stimulated recombination with the synthetic DNA template by 2-3 orders of magnitude (Elliott et al., 1998).

Homology directed repair (HDR) is the pathway exploited to induce homologous recombination with a synthetic template. However, without the synthetic template, HDR is an error-free pathway, using endogenous homology like the sister chromatid to repair DSBs (Sargent et al., 1997). But while HDR enables versatile editing outcomes through a template, it is not the only pathway available to repair DSBs. One of the other repair pathways is

microhomology-mediated end joining (MMEJ). Like HDR, MMEJ relies on homology. However, MMEJ relies on 3 to 5 nucleotides of microhomology around the DSB, which leads to it being error prone and commonly associated with deletions (Sfeir & Symington, 2015). Another pathway is non-homologous end joining (NHEJ), which functions independently of homology, both local and otherwise. But like MMEJ, it is error-prone, and can result in small insertions or deletions (Sharma & Raghavan, 2016).

With editing outcomes being dependent on repair pathway, the ability to influence which pathway cells utilise is essential for defining the outcome. However, equally important is the ability to define the genomic locus at which the desired outcome occurs. Because of this, the ability to target any location in the genome is essential for defining where the change occurs.

1.2 Customisable targeted systems for cleaving DNA

Homing nucleases like I-SceI, and restriction enzymes, target specific sequences in the genome (Belfort & Roberts, 1997). However, despite there being thousands of different systems (Roberts et al., 2007), meaning thousands of potential targets, the number of potential targets is limited in the context of genomes with billions of nucleotides.

Over time, customisable targeting systems have emerged, such as zinc finger nucleases (ZFNs) (Durai et al., 2005; Porteus & Carroll, 2005; Urnov et al., 2005). These enable researchers to specify nearly any DNA target, rather than relying on a limited set of predefined targets. ZFNs, an engineered system, function by combining the non-specific cleavage domain of the restriction enzyme, FokI, with zinc finger domains, resulting in a custom targeted endonuclease. In theory, the design is straightforward, with each zinc finger domain interacting with three to four base pairs of DNA (Isalan et al., 1997), allowing ZFNs to be modularly designed (D. A. Wright et al., 2006). However, the modular design process of ZFNs has since been demonstrated to have a high failure rate, or a low efficiency (Ramirez et al., 2008). This is because adjacent zinc fingers have been demonstrated to interact with each other (Isalan et al., 1997), meaning that the actual DNA target can differ from the expected DNA target. While previously-established selection-based methods can allow researchers to select for optimal ZFNs (Hurt et al., 2003), such methods are expensive and time consuming.

Aiming to address this problem, other groups investigated transcription activator-like effectors (TALEs) (Bogdanove et al., 2010; Christian et al., 2010). They hypothesised that TALEs, which are proteins secreted by *Xanthomonas* bacteria to alter host plant transcription (B. Yang et al., 2006) could be engineered to function as a nuclease. This was achieved by binding TALEs to the FokI cleavage domain, resulting in TALE nucleases (TALENs) (Joung & Sander, 2013).

TALENs, like ZFNs, use a protein-guided system to direct the FokI endonuclease to a DNA target. But unlike zinc fingers which each target multiple base pairs, TALENs are guided by a set of tandem repeats which each contain two adjacent nucleotides that recognise one base pair of DNA (Boch et al., 2009; Moscou & Bogdanove, 2009). These findings resulted in ciphers that enabled the design of custom TALENs to target arbitrary targets (Christian et al., 2010).

Despite their versatility, ZFNs and TALENs do present certain challenges. One is in FokI, the cleavage domain commonly used with these systems. The challenge is that FokI must dimerise for cleavage to occur (Bitinaite et al., 1998). Because of this, ZFNs and TALENS usually function as heterodimers, requiring the targeting of two recognition sites for DNA cleavage to occur. This adds complexity to the system as for every cleavage event, two proteins must be engineered, and their precursors synthesised. Then if one of these has a low targeting efficiency, especially with the more challenging to engineer ZFN, DNA cleavage may not occur. Another challenge with TALENs is that costs can be expensive due to the custom repeats. Although alterations can be made to reduce costs by using naturally occurring TALE binding sites, or alternate cloning techniques (Cermak et al., 2011), this can introduce other limitations such as reducing the number of potentially targetable regions.

A more recent customisable targeted nuclease that may solve the shortcomings of ZFNs and TALENs are the clustered regularly interspaced short palindromic repeats (CRISPR) systems (Jinek et al., 2012).

1.3 Introducing CRISPR gene editing technologies

CRISPR and CRISPR associated (Cas) proteins have evolved as prokaryotic adaptive immune systems (Terns & Terns, 2011). They function by integrating small pieces of invading phage DNA, known as protospacers, into repeat arrays, where they become spacers. These short spacer sequences serve as a memory of the organism from which they originated, defining it as a target. However, for targeting to occur, the spacer and repeat array must first be processed. Processing depends on the CRISPR system, with the following describing the class 2 type II Cas9 system. Firstly, repeat-spacer arrays are transcribed into long transcripts, with each containing multiple spacers. These transcripts are known as precursor CRISPR RNAs (pre-crRNAs). Subsequent processing by trans-activating CRISPR RNA (tracrRNA) results in cleavage of the pre-crRNAs by RNase III into CRISPR RNAs (crRNAs), with each containing a single spacer, flanked by the repetitive region. The final effector complex consists of the ribonucleoprotein complex Cas9-crRNA-tracrRNA, with the crRNA spacer sequence defining the target through Watson-Crick base pairing to be cleaved by Cas9 (Cong et al.,

2013; Garneau et al., 2010; Mali et al., 2013). However, a noteworthy development was to synthesise a chimeric single guide RNA (sgRNA) , containing the essential components of the crRNA and tracrRNA (Jinek et al., 2012).

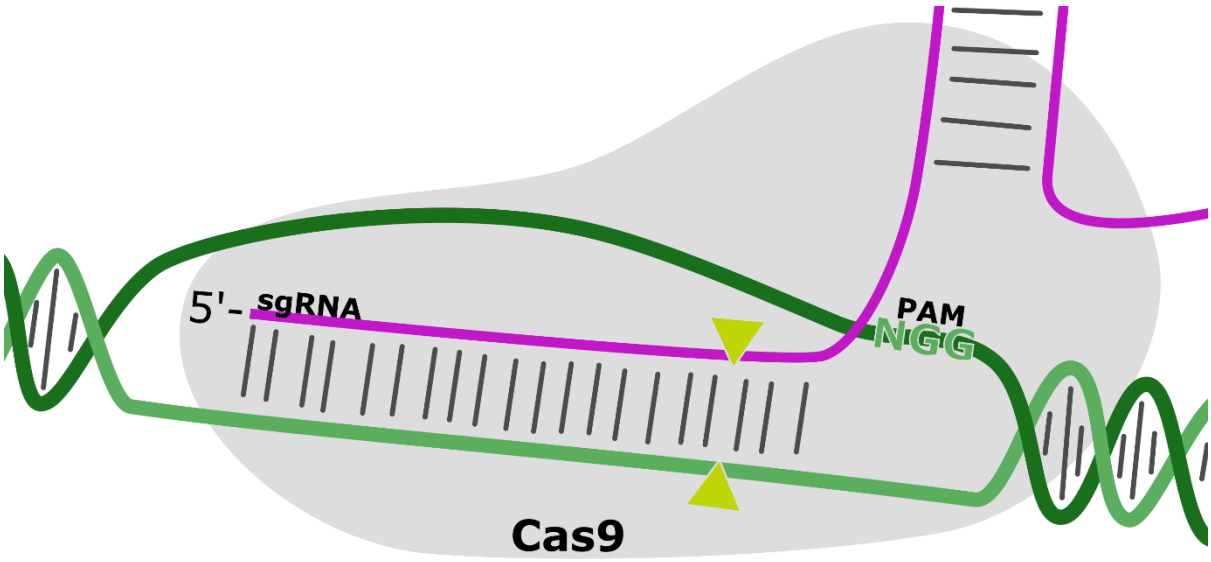


Figure 1 – a representation of the interaction between the CRISPR-Cas9 complex and its DNA target. The sgRNA forms an R-Loop with the DNA strand containing the complementary sequence, displacing the non-target strand. The two yellow triangles indicate the location of the DSB relative to the PAM.

Common to different CRISPR systems is the protospacer adjacent motif (PAM) (Mojica et al., 2009). The PAM is a short DNA motif present at the DNA target site, typically 3-5 nucleotides long (Shah et al., 2013), As the name implies, the PAM is adjacent to the protospacer, although whether this is upstream or downstream depends on the CRISPR system. For CRISPR-Cas9, the sequence of the PAM is NGG (where N can be any nucleotide), with the PAM being adjacent to the 3' end of the protospacer (Figure 1). The PAM plays a critical role in target recognition. Recognised by the PAM-interacting domain of Cas9 (Nishimasu et al., 2014), PAM-recognition promotes local duplex melting of the double-stranded DNA around the PAM (Anders et al., 2014). This process enables the CRISPR-Cas9 complex to further perform guide recognition through R-loop formation between the spacer and target strand. The requirement of the PAM is important to prevent off-targeting and self-targeting. For example, despite the target sequence being present in the CRISPR array, as a spacer, by not being PAM-proximal it is not a valid CRISPR-Cas9 target.

While Cas9 is the most cited nuclease, another frequently cited CRISPR system is the class 2 type V Cas12a, formerly known as Cpf1 (Zetsche et al., 2015). Although Cas12a is an RNA-guided nuclease like Cas9, the two CRISPR systems vary in several ways. Firstly, Cas12a does not require tracrRNA for processing of its pre-crRNA, making Cas12a easier to use.

Another difference is the resulting DSB, with Cas12a creating staggered cuts unlike the blunt ends induced by Cas9. This has been demonstrated to influence the repair outcome (Bothmer et al., 2017). Finally, the Cas12a PAM sequence is the T-rich 5'-TTTN-3', adjacent to the 5' end of the protospacer (D. Kim et al., 2016). This complements the Cas9 G-rich PAM sequence, enabling the targeting of regions where no suitable Cas9 sgRNAs are available. Although there are other differences in kinetics and activity, the availability of two differing CRISPR systems broadens the scope of gene editing with CRISPR.

CRISPR systems have several benefits over other customisable targeting system. Unlike ZFNs and TALENs which recognise DNA nucleotides through zinc fingers or protein domains, CRISPR-Cas is RNA-guided through the spacer sequence. This eliminates the need for ciphers or selection-based methods for defining DNA targets. Also, unlike ZFNs and TALENs that utilise FokI, CRISPR-Cas9 functions with a single target site and does not require heterodimeric targeting. Lastly, CRISPR can be used for results other than cleavage. For example, impairing the cleavage domains of CRISPR-Cas9 results in the catalytically inactive, dead Cas9 (dCas9), which can be used to control gene expression via transcription and chromatin remodelling (Qi et al., 2013). Other CRISPR systems can be used for other applications including nucleic acid detection (Gootenberg et al., 2018) or CRISPR-based diagnostics (Gootenberg et al., 2017), with additional uses reviewed in (Knott & Doudna, 2018).

1.4 Limitations of CRISPR gene editing technologies

Despite these benefits, CRISPR systems do have limitations. Three limitations are:

- target availability
- efficiency
- specificity

The first, target availability, results from CRISPR systems requiring the presence of a PAM for target cleavage. Because the PAM is a fixed sequence, not every sequence can be targeted. For example, with Cas9's NGG PAM, only sequences with two adjacent G nucleotides are generally considered to be potential targets. However, Cas9 PAMs will occur every eight nucleotides of double-stranded DNA on average, meaning that targets are not rare. Also, the NGG motif is not the only PAM, with other CRISPR systems requiring different PAM sequences (Westra et al., 2013). The possibility to target different PAMs further expands the target repertoire.

The second limitation, efficiency, is the likelihood of observing the desired outcome at the target. The efficiency for a given crRNA or sgRNA can vary for different sequences (T. Wang et al., 2014), cell-lines (Mali et al., 2013) and organisms. Furthermore, efficiency can also be broken down into cleavage efficiency and the efficiency of the repair mechanism. Because different repair pathways can result in different outcomes, even if cleavage occurs, the resulting outcome may not be the desired outcome. For example, NHEJ events have been demonstrated to occur more frequently than HDR events in CRISPR-Cas9 experiments (Mali et al., 2013), limiting the efficiency of editing outcomes that require HDR.

The third limitation, specificity, refers to cleavage events at targets other than the intended target. Unintended targets are referred to as off-targets (Cho et al., 2014). Off-targets are a problem because they can result in genes being unintentionally disrupted, or even chromosomal rearrangements such as translocations due to multiple cleavage events (Iarovaia et al., 2014). Off-targets can occur when the target sequence (protospacer and PAM) occurs elsewhere in the genome. Because the CRISPR-Cas system is RNA-guided, it has no way to differentiate between the intended target and off-targets. Off-targets can also occur at sequences that share homology with the target, albeit usually at a lower efficiency. This is due to alternative PAMs (Y. Zhang et al., 2014) and mismatch-tolerance (Anderson et al., 2015). But as well as off-targets, unwanted changes can also occur at the intended target (AYABE et al., 2019; Kosicki et al., 2018; H. Lee & Kim, 2018). Known as on-target effects, these changes can include large deletions, insertions, and chromosomal translocations leading to a loss of heterozygosity. On-target effects have recently been demonstrated to be a result of Cas9 interfering with the MMEJ repair pathway (Kosicki et al., 2020).

1.5 Improving CRISPR targeting

Although these problems can be addressed by trialling different targets *in vitro* to identify optimal loci for targeting, they can also be addressed *in silico* by creating models from existing data. Through such models, it is possible to take a series of inputs to calculate an output (Cox, 2006). Take, for example, identifying CRISPR targets. Targets will generally conform to two rules. The first being the presence of a PAM, and the second being homology between the crRNA spacer and the region adjacent to the PAM. If these two rules are true, then the output is true. Despite its simplicity, this rule based system is a general representation of the conclusions of the original publications (Gasiunas et al., 2012; Jinek et al., 2012). But because the crRNA is customisable, this model can be simplified further to identify CRISPR targets solely through the presence of an PAM. That is, if a 23-nucleotide sequence has a PAM at the 3’ end, it is labelled as a target. This is an example of a binary classifier, as given a set of inputs, it will produce a binary output; true or false, target or non-target. However, because

the only input is the presence of a PAM, this classifier will not differentiate between efficient and inefficient sgRNAs.

To differentiate between efficient and inefficient sgRNAs, the model can be extended to include additional rules. For example, with nucleosomes negatively influencing efficiency (Hinz et al., 2015), targets that lie within nucleosomes can be labelled as non-targets. Alternatively, because targets that lie on nucleosome boundaries can still be cleaved relatively efficiently, the rule can be altered to instead penalise targets proportionately to the number of overlapping nucleotides. This could be, for example, with a negative coefficient if the relationship between nucleosome overlap and efficiency is linear. More complex statistical models could take conditional probabilities into account, or model non-linear relationships. But now, instead of the model labelling sequences as a target or non-target, it will label sequences in a numerical range that represents efficiency. This is an example of a regressor.

Statistical and rule-based models can also be implemented to consider off-targets. GT-Scan (A. O'Brien & Bailey, 2014) and Cas-OFFinder (Bae et al., 2014) are two computational tools which identify targets and rank them by their uniqueness in the genome. Such tools are built on empirical and theoretical evidence surrounding features that abrogate CRISPR cleavage. However, they only model a limited number of features. For example, they do not model the kinetics of target recognition (Rutkauskas et al., 2015), which could be used to quantify the cleavage efficiency of off-targets. In effect, these tools only quantify the uniqueness of targets based on DNA sequence, rather than quantifying the ratio of target cleavage to off-target cleavage.

Models like these can enable researchers to identify optimal targets *in silico*. This is beneficial as it can result in less wasted time and resources trialling inefficient sgRNAs, which can save researchers time and money (Listgarten, 2017). Efficient CRISPR targeting can also minimise undesired results like somatic mosaicism (Yen et al., 2014). However, these rule-based and statistical models only capture a limited number of manually curated features, based on inferences made from experimental data. Because these models exclude variables that may modulate efficiency, their predictions will be non-optimal, which can lead to low prediction accuracy. However, manually constructing models to represent every single modulator, especially individual nucleotides, motifs, and combinations of these, can be an exponential task. Also, as more-complex systems are modelled, such models may fail to accurately represent the system (Breiman, 2001a), resulting again in a low prediction accuracy. Therefore, a solution is required to enable the accurate modelling of large and complex systems like sgRNA efficiency.

1.6 Machine learning

Machine learning (ML) has enabled the ability to model complex datasets with minimal human intervention (Domingos, 2012). ML algorithms automatically train models on large amounts of data, without the need for researchers to analyse variables one at a time (Figure 2). Models can then be used to predict labels, like efficiency, for unknown data. They can also be used to gain insights into variables that modulate the label. ML is generalisable to many data modelling problems, and its applicability to biological problems has long been recognised (Tarca et al., 2007). Early uses ranged from identifying translation initiation sites in E. coli (Stormo et al., 1982), to microarray analysis (Dudoit et al., 2002). One of the reasons ML is used in these fields is the high dimensionality (large number of variables) of datasets. Because, where statistical modelling requires manual analysis of features to identify correlations or probabilities, ML can model these complex relationships automatically. This enables models to be trained quickly and to scale, enabling the ability to efficiently model large datasets with hundreds, thousands or even millions, of samples and variables.

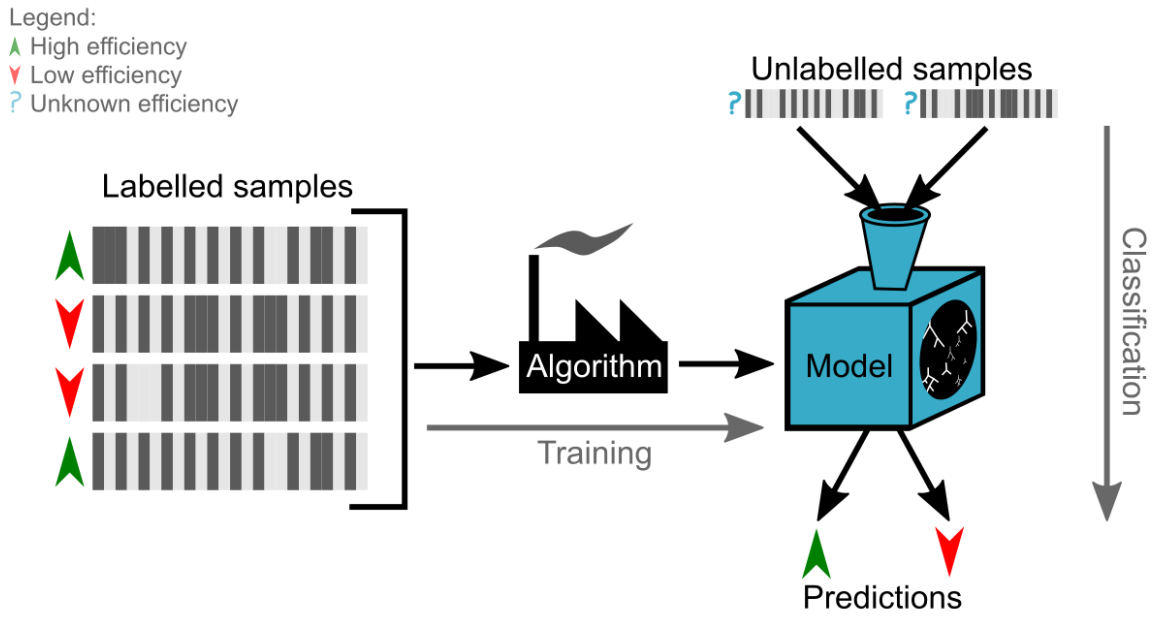


Figure 2 – an example binary classifier. Depicted horizontally from left to right is the training process. The machine learning algorithm produces a model from labelled data. Depicted vertically from top to bottom is the classification (or prediction) process. The trained model predicts labels for unlabelled data. Labelled data can also be classified to validate the model by comparing predictions to labels.

The ability to scale is especially desirable in the health field as focus shifts from single genes to genomics and the environment (Khoury, 2003). Genomic data alone can result in datasets that contain millions or even billions of features. Modelling many features can also play into one of the strengths that some ML algorithms have; the ability to model interacting features. This is where one or more features, together, modulate the outcome. This can include

polygenic effects. For example, where traditional genome-wide association studies (GWAS) typically consider the influence of individual variants on the response variable (label), ML approaches can consider and model multiple variants on a genome-wide scale (Bayat et al., 2020).

As well as the health field, ML is also becoming more prevalent in the genome engineering field, with the first sgRNA efficiency prediction model published in 2014 (Doench et al., 2014). This model, trained using the logistic regression algorithm, predicts the cleavage efficiency of CRISPR-Cas9 using features like sgRNA sequence, targeted exon, and the position of the target in the gene. Since then, a slew of models trained using different algorithms, variables, and datasets have been released and published, all with the aim of computationally improving CRISPR genome engineering by predicting the most efficient sgRNAs.

As well as predicting the efficiency of sgRNA designs, ML can also be used to predict off-target efficiency (Listgarten et al., 2018). This is an improvement on previous rule-base off-target approaches which simply identify sequence similarity. Because sgRNA efficiency for off-targets, like targets, depends on factors other than the number of mismatches. So, by firstly identifying similar sequences and secondly predicting their cleavage efficiency, hundreds of false positives can be eliminated when compared to pattern matching approaches.

Despite the automatic nature of ML, producing a useful model still involves manual steps. This can include experimental design, identifying public data sources, preprocessing, analysis, or any combination thereof. Preprocessing in particular can have a significant impact on the generalisability of the resulting model (Kotsiantis et al., 2006). It includes tasks like data labelling, feature processing and normalisation. For preprocessing, as well as the other manual steps, cross-domain knowledge is useful to ensure data is represented appropriately. For example, ML algorithms generally assume a balanced dataset (Japkowicz, 2000). This can lead to difficulty when modelling experimental data which is unbalanced due to an inherently inefficient experimental design, or from failed tests being discarded. Unbalanced datasets can be addressed in preprocessing, but ideally experiments would be designed, and data collected, with the aim of producing a quality dataset. This aim can be more-readily achieved with the knowledge of what makes a good dataset.

At the time of writing my thesis, there were 21 prediction tools utilising ML for CRISPR prediction. There are various reviews covering these tools, including a comprehensive comparison of performances and predictions (Bradford & Perrin, 2019) and a high level overview (Wilson, O'Brien, et al., 2018). However, missing was a point of reference relating

machine learning to genome engineering; a review to explain why prediction tools differ or share common ground. An example of this from another field is a review paper by Moen el al., which provides a technical summary of cellular image analysis, deep learning and why and how these two fields are relevant to each other (Moen et al., 2019).

For the rest of my introduction, I aimed to fill this gap, using minimal jargon to provide context between ML and gene editing. Although this section also serves as a literature review of current genome editing prediction tools, my primary focus is on the details that make them work and their potential limitations. This includes, for example, algorithm choice, how models represent biological data, and problems with training datasets.

1.7 The uses and benefits of ML for CRISPR prediction

Despite computational CRISPR tools using a range of algorithms, datasets, and preprocessing techniques, one common theme is that nearly all recent tools use a model trained using ML. Using these models, each tool aims to enable more-effective CRISPR experiments by predicting certain aspects of experimental outcome, i.e. cleavage efficiency. With each tool making predictions from a set of input variables, no prior knowledge of cleavage efficiency is required for researchers to take advantage of trained models.

The reason for using ML to model CRISPR cleavage is that this process is a highly complex interplay of influencing factors. As well as nucleotide sequence, this can include cellular environment and experimental conditions. ML enables capturing this complex interplay of inputs, automatically. Specifically, ML enables researchers to model systems like CRISPR, without specifying the relationship between target properties or experimental parameters (features) and the outcome (label). Instead, ML algorithms automatically learn relationships between features and labels, storing a representation of these relationships as a model. Subsequently, a model can be used to predict the outcome for experiments where the outcome is unknown, i.e. untested sgRNA designs. The primary benefit here is that ML enables researchers to predict the effectiveness of an sgRNA design *in silico*, rather than having to test every design empirically, saving effort and time.

1.8 Preprocessing

Even though ML implies automatic training, the data that ML algorithms learn from must generally be good quality and conform to certain assumptions. Therefore, before training, data should be processed. Preprocessing requires the researcher to make informed decisions and is one area where prediction tools differ. Preprocessing can also be an iterative process, with observations from modelling outcome leading to the researcher altering their preprocessing

process. Preprocessing includes tasks like labelling and feature selection, with steps taken depending on both aim and data.

1.8.1 Considerations in data labelling

Supervised ML algorithms train models from labelled data. In the context of CRISPR experiments, the label may be cleavage efficiency, the likelihood of seeing a desired mutation or the ability of CRISPR to control gene expression. So, one common step between all tools is to define an appropriate label. Labels can be represented discretely (e.g. high *or* low) or continuously (e.g. 0 *to* 1). The representation depends on various factors, such as the algorithm used, the data being modelled, and what the desired outcome is. For discrete variables, classification algorithms are used. This can be binary classification for two classes, or multiclass classification for more than two classes. For continuous variables, regression algorithms are used.

The sgRNA cleavage efficiency, for example, is continuous as efficiency is on a range from 0% to 100%. So, given a model trained using a regression algorithm, predicting the efficiencies for four unlabelled sgRNAs would result in each one being assigned a value in this range. For example, [0%, 80%, 90%, 100%]. With higher efficiencies being desired, the clear choice would be the sgRNA with a prediction of 100%. Continuous values like efficiency can also be represented discretely. In preprocessing, “< 50%” efficiency sgRNAs could be labelled “low” and “>= 50%” efficiency sgRNAs labelled “high”. In this case, the resulting model would classify the previous four targets as [low, high, high, high]. This removes the ability to discriminate between the top three targets as, now they are all simply “high”, rather than 80%, 90% and 100%, however, despite this loss in information, classification can provide benefits over regression in certain cases.

Firstly, classification is generally faster in training and predicting (Salman & Kecman, 2012). But perhaps more importantly, current regression models for CRISPR efficiency prediction do not achieve a high accuracy, with empirical observations of efficiency not necessarily correlating well with predictions (Wilson, Reti, et al., 2018). For example, an sgRNA predicted to be 100% efficient may be no more efficient, or even less efficient, than a target predicted to be 80% efficient. This is because the complexity involved in modelling biological systems can result in models with a limited sensitivity for prediction. This can be a result of inadequate datasets, missing features or false assumptions made in preprocessing. In this case, it can be beneficial to model sgRNAs with a “high” or “low” label. Although predictions will be less informative, prediction accuracy will be higher. Furthermore, this can prompt researchers to

trial different high-predicted sgRNAs, rather than basing their choice on the top result from a less-than-perfect-accuracy model.

A common pitfall with CRISPR data is imbalance (Gao et al., 2019). Imbalance is when the positive editing results outnumber the negative results, or vice versa. This can result from researchers only publishing positive results for CRISPR experiments, or from results being overwhelmingly negative due to, for example, the low efficiency of HDR (Hruscha et al., 2013; Mao et al., 2008). One way to overcome data imbalance when training a classification model is by choosing an appropriate threshold when converting efficiency from a continuous value to a binary (high/low) value (Figure 3). For example, a threshold of 50% may seem like the obvious choice, but if only 2 out of 10 targets have an efficiency > 50%, then a classification model could classify all 10 targets as low-efficiency and still have an accuracy of 80%. One potential solution is to adjust the decision threshold (He & Garcia, 2009), for example, from 50% to 20%. This results in an even number of high and low efficiency samples. However, now samples with an efficiency of > 20% are considered high efficiency, which may or may not be ideal. Another potential solution is to modify how targets are sampled. For example, rather than choosing targets randomly for training and testing, a bootstrap (sampling with replacement) method can be used to oversample the minority class, as demonstrated by CRISTA and DeepCRISPR (Abadi et al., 2017; Guohui Chuai et al., 2018).

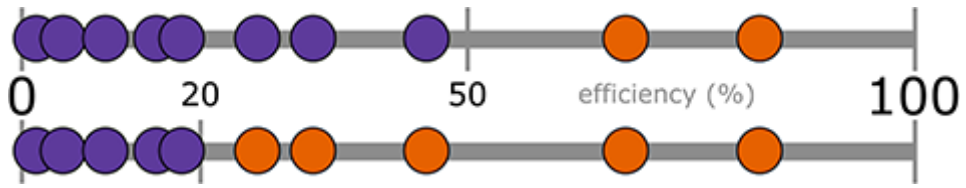


Figure 3 – this figure represents two potential decision thresholds (20% and 50%) for ten hypothetical sgRNA samples (coloured dots). Each sample has a DNA cleavage efficiency in the range of 0% and 100%. For a binary classifier, samples above the decision threshold are considered "high-efficiency" (orange) and samples below the decision threshold are considered "low-efficiency" (purple). The decision threshold can be arbitrarily set to any value between 0 and 100%, and an appropriate decision threshold can help keep data balanced. A threshold of 50% in the upper example results in two "highs" and eight "lows". This can result in a poor-performing model as a resulting model could indiscriminately targets as low-efficiency yet have a relatively good accuracy of 80% (eight out of ten are correct). However, a threshold of 20% results in five highs and five lows. Now if a model indiscriminately classifies all ten targets as low-efficiency it will have a more-appropriate accuracy of 50%.

The problem of imbalance is exacerbated for labels with more than two classes. One example is predicting editing outcome (i.e. insertion or deletion length), as attempted by FORECast (Allen et al., 2019) and SPROUT (Leenay et al., 2019). While this greatly increases the control over experimental outcomes, it also increases the number of distinct classes, which in turn requires an increase in training data size to adequately fit the model. For example, for binary labels (high/low) and a perfectly balanced dataset of 1,000 samples, each class has 500 (1000/2) samples. If the same dataset is labelled according to the deletion length, from zero

to four nucleotides, for example, then the number of samples in each class would drop to 200 (1000/5). Predicting other outcomes, like insertion length or other variants would drop the sample size of each class even further, potentially until classes contain only single samples. To combat this problem and still have enough samples for each of the combinatorial scenarios, FORECast is trained on more than 40,000 sgRNAs. However, where large sample sizes are not possible, an alternate solution is to limit the number of classes or to train multiple models. For example, rather than having a single model trained on data labelled for every type of editing outcome, SPROUT relies on multiple models, where one may model deletion length, and another may model insertion length. This allowed it to be successfully trained on 1,656 sgRNAs.

1.8.2 Selecting features for a generalisable model

As well as being labelled, each sample must include features. Features are a set of data (i.e. genetic, epigenetic, or experimental) that are abstracted to a format suitable for training a model. The challenge is to include enough data for algorithms to produce accurate models, but without including data that is difficult/expensive to obtain, overly specific or irrelevant. The aim is to produce a model that can not only make correct predictions on the validation data but is also generalisable to data from other groups or laboratories.

Used in every model mentioned throughout this review, is genetic data. This includes the sgRNA sequence, PAM, and/or adjacent nucleotides to the target. These features are used by every tool because efficient sgRNAs have been demonstrated to prefer certain nucleotides over others (Hsu et al., 2013). However, a secondary benefit is that sequence information is universal. That is, with the sgRNA sequence being essential for guiding CRISPR/Cas9 to a target, it is a property that will be known for previously conducted CRISPR experiments (resulting in more training data), as well as for future experiments. However, a difference between tools is the window size at the sgRNA target (23nt for ge-CRISPR (Kaur et al., 2016), 26nt for WU-CRISPR (Wong et al., 2015) and 30nt for sgRNA design (Doench et al., 2014), CRISPRpred (Rahman & Rahman, 2017) and TUSCAN (Wilson, Reti, et al., 2018)). But regardless of this difference, because each of these tools only requires sequence information, they can predict sgRNA efficiency agnostic to cell type or species.

With epigenetic modifications having been demonstrated to modulate CRISPR cleavage, sequence-only models can lack accuracy compared to models which include such features. For example, Chari *et al.*, identified DNase-seq and H3K4 trimethylation data to modulate efficiency (Chari et al., 2015). However, while including epigenetic information improved their model accuracy, it had the consequence of making it not only species-specific but also cell

type-specific (Chari et al., 2017). They hence opted for using only sequence information in their sgRNA scorer and sgRNA Scorer 2.0 models (Chari et al., 2015, 2017). Azimuth (Doench et al., 2016) and CRISPRpred (Rahman & Rahman, 2017) also aimed to improve accuracy by including non-sequence features. This included positional features like “exon targeted” and “position of target in gene”. Doench *et al.* demonstrated these features to improve model performance over a sequence-only model (Doench et al., 2016). However, as with epigenetic information, this also had the consequence of decreasing generalisability. This is because genetic annotations were required to predict sgRNA efficiency. To account for this, Azimuth includes two models and falls back to the sequence-only model if positional information is not available.

In the pursuit of finding features that add more information and increase accuracy, care should be taken to avoid including as much data as possible, regardless of relevance. Feature-sets should ideally include only properties that have a causal relationship to the label. This is because including irrelevant features (i.e. experiment ID in a tracking system) can be detrimental by increasing the noise and search space, thus potentially reducing model performance (Hall & Smith, 1999; Hughes, 1968; Trunk, 1979).

1.8.3 Translating data to machine-readable features

Once data has been identified for inclusion in training, it needs to be processed to meet certain criteria. This is especially true for sequence data because most ML algorithms cannot handle strings natively. For example, an algorithm may be able to identify that “CATA” is different to “CATT”, but not *how* it is different. To overcome this problem and to capture quantitative differences, sequence features therefore need to be “tokenised” (Figure 4).

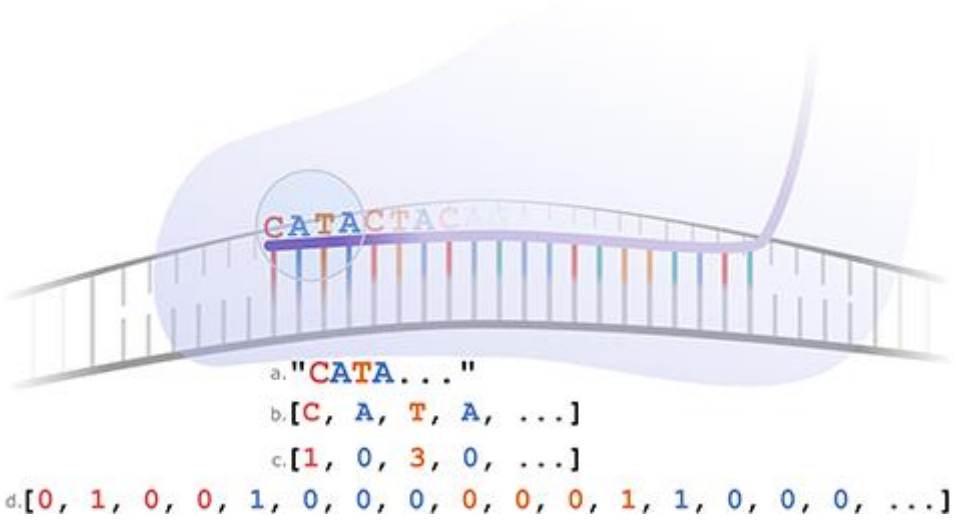


Figure 4 – four different ways to encode the sgRNA nucleotide sequence. Demonstrated, are the four encircled nucleotides (CATA). a) as a string. This will not be compatible with many ML algorithms. b) as a list of characters. Here each nucleotide has its own "feature". However, many ML algorithms require features to be represented as numbers. c) as a list of numbers. Here each nucleotide has been arbitrarily assigned a value from 0 to 3. However, algorithms that accept "continuous" features will consider T (3) to be more different from A (0), than T is from G (2) because of the larger difference in the arbitrarily assigned values. d) one-hot encoded. Here each nucleotide is represented as four list elements. One (and only one) of these elements is "hot" (i.e. 1) depending on the nucleotide. In this example, the first element being hot, i.e. [1, 0, 0, 0], represents an A. In this representation, all nucleotides are represented as being equally different.

Tokenisation generally involves breaking down an item, such as a string, into a more-generic format like an array of numbers. For example, each nucleotide in the DNA (or RNA) alphabet could be represented as a number from 0 to 3 (A=0, C=1, G=2, T=3). In this case "CATA" would become [1, 0, 3, 0] and "CATT" would become [1, 0, 3, 3]. This is effective because now any ML algorithm can see that only the fourth position has changed (0 to 3). However, this representation is not adequate for algorithms that expect continuous variables, because T (3) is more different to A (0) than T (3) is to G (2). Instead, strings can be one-hot encoded. One-hot encoding represents nucleotides as 0s and 1s by using a separate token for each position in the sequence as well as for each possible nucleotide (Figure 4).

The above processes can be extended to create additional features that represent, for example, nucleotide pairs. This simply entails creating an additional token for each permutation and combination of two nucleotides at each position along the sequence. Feature generation can also be driven by domain- or expert-knowledge. For example, a feature could be created to represent the nucleotides either side of the "GG" in the PAM ("NGGN" (Doench et al., 2016)), if this were empirically observed to influence efficiency.

1.9 Machine learning algorithms

Although a well-curated feature-set and carefully chosen labels is one requirement for training a well-performing model, so is an appropriate algorithm. There are too many different ML

algorithms to cover them all, so this section will provide an overview of algorithms frequently used in current CRISPR prediction tools. For comparisons of tools themselves, see Guo-hui 2017 (Guo hui Chuai et al., 2017) and Cui 2018 (Cui et al., 2018), and for benchmarks see Yan 2018 (Yan et al., 2018).

1.9.1 Linear regression and logistic regression

Linear regression and logistic regression are two statistical models that model a linear relationship between features and a label. Linear regression is used by CRISPRscan (Moreno-Mateos et al., 2015), and logistic regression is used by sgRNA design (Doench et al., 2014). Both are regression models, however linear regression predicts outputs onto a continuous range, whereas logistic regression predicts the probability of a binary output being true, on a scale from zero to one. This means that logistic regression is generally used for binary classification, i.e. true/false. Despite both algorithms modelling linear relationships. They can be extended to model non-linear relationships through data transformations. For example, Doench *et al.* observed a non-linear relationship between sgRNA GC content and efficiency, where a high or low GC content were correlated with a lower activity than a ~50% GC content. For this non-linear relationship, they created two disparate features (one for above 50% GC and one for below), which enabled the logistic regression algorithm to capture this non-linear relationship (Doench et al., 2014). However, this manual statistical analysis is exactly what machine learning aims to avoid, because it becomes less feasible when dealing with large datasets. It can also reduce model explainability. To avoid these manual transformations, non-linear algorithms and models are available.

1.9.2 Support vector machines

One model that supports non-linear separation are support vector machines (SVMs). There are different SVM algorithms that can train SVM classifiers or regressors, allowing them to be used for continuous or binary values. In the CRISPR space, SVM models are used in sgRNA Scorer, ge-CRISPR, sgRNA Scorer 2.0, CRISPRpred, WU-CRISPR, TSAM and CRISPR-DT (Chari et al., 2015, 2017; Kaur et al., 2016; Peng et al., 2018; Rahman & Rahman, 2017; Wong et al., 2015; Zhu et al., 2019). The SVM algorithm is also used by sgRNA design for feature selection (Doench et al., 2014). Although SVMs are a linear classifier, they can efficiently model non-linear data by implicitly transforming features into a high-dimensional representation where a linear separation of samples is possible (Hearst, 1998). However, this can obscure which features contributed to the decision process, which can limit explainability. Black box models like this have been demonstrated as generally being more accurate than explainable, white box, models (Guohui Chuai et al., 2018; Dumais et al., 1998; J. Lin & Wong, 2018; Pranckevičius & Marcinkevičius, 2017). However, this property is not absolute and

performance ultimately depends on the algorithm in question and the data being modelled (Amancio et al., 2014; Jia et al., 2013; Wilson, Reti, et al., 2018).

1.9.3 Decision trees

To address the issue of explainability, are tree-based methods, which stem from the decision tree algorithm. Decision trees also address another important property for the CRISPR space; the ability to capture higher-order interactions between features. In the context of sgRNA efficiency, this includes interacting features. That is, two or more features—nucleic, epigenetic, or otherwise—that if present together have a correlation with or influence the efficiency. Decision trees can model this process through the recursive partitioning process. This means that decision trees model data by iteratively splitting the dataset based on features that separate the data. The aim is to generate groups that are pure, i.e. groups that contain *only* high efficiency targets or *only* low efficiency targets. Consider the hypothetical example where sgRNAs with a G at position 20 (G20) *and* a <20% GC content have higher efficiencies than sgRNAs with either or none of these features. Because G20 cannot separate the data into pure groups, the recursive nature of training results in a new level being added to further separate the data, in this case based on <20% GC (Figure 5). Another benefit of tree-based methods is that they are applicable to both regression and classification (Loh, 2011). Furthermore, it is possible to interrogate tree-based models to identify which features have the most influence on efficiency prediction, hence making the prediction “explainable”.

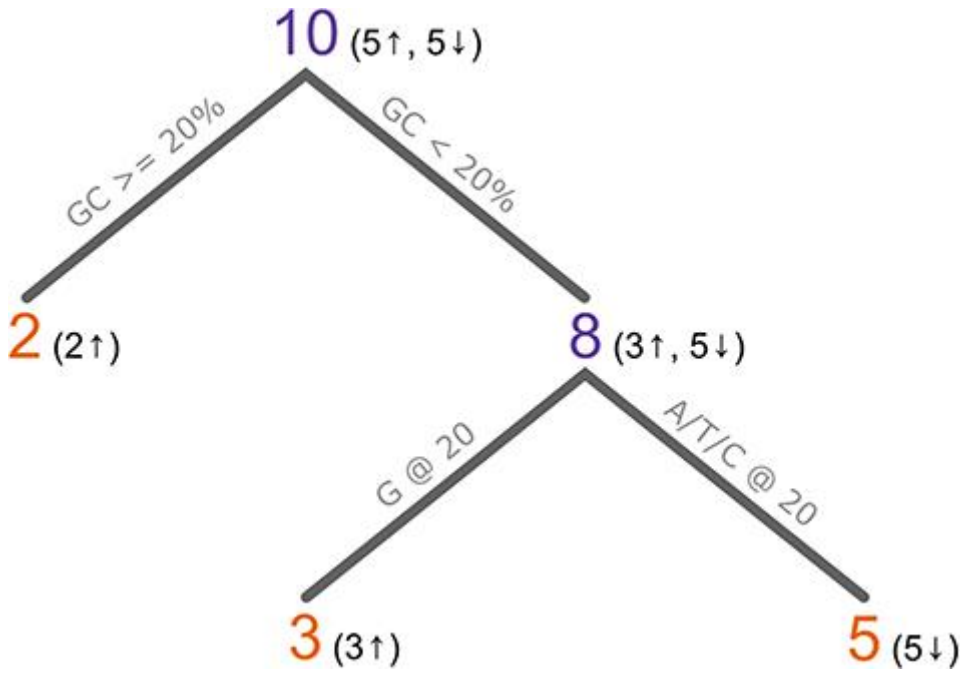


Figure 5 – a hypothetical example of a decision tree trained on 10 samples. The first split is on an sgRNA GC content above or below 20%, which separates out two samples with a GC content above 20%. As both samples are "high efficiency", it results in a pure node (orange). Of the eight sgRNAs with a GC content above 20%, three have a high efficiency and five have a low efficiency, so this node is impure (purple). The next split is on the presence (or absence) of a G at position 20 in the sgRNA. All three sgRNAs with a G have a high efficiency and all five sgRNAs without a G have a low efficiency. The resulting nodes are pure, so training concludes. This model would classify new sgRNAs as high if "the GC content is >= than 20%" or "the GC content is < 20% and there is a G at position 20". In reality, such a model would be much more complex with purity not being reached so early, or at all.

1.9.4 Random Forests and gradient boosted regression trees

Two tree-based algorithms used for predicting sgRNA efficiency prediction are Random Forests (used by CRISPRpred, CRISTA, TUSCAN, CRISPR-DT and CUNE) (Abadi et al., 2017; A. R. O'Brien et al., 2019; Rahman & Rahman, 2017; Wilson, Reti, et al., 2018; Zhu et al., 2019) and gradient boosting (used by Azimuth and SPROUT) (Doench et al., 2016; Leenay et al., 2019). These algorithms are ensemble methods, meaning they create models consisting of multiple decision trees. This collection of trees can survey a larger search space and hence are superior to single trees by improving the generalisation or by reducing the error (Breiman, 2001b).

1.9.5 Deep learning

Recent increases in compute power have enabled another group of ML, deep learning. This group includes algorithms that consist of multiple non-linear levels, such as convolutional neural networks (CNNs) (Bengio, 2009). CNNs have been demonstrated to be successful for image analysis, where the many interconnected levels allow for highly general models that can not only classify images, but also objects within images. The field of sgRNA efficiency prediction has recently started using deep learning with the development of tools like

DeepCpf1, DeepCRISPR, off_target_prediction and DeepCas9 (Guohui Chuai et al., 2018; H. K. Kim et al., 2018; J. Lin & Wong, 2018; Xue et al., 2019). However, deep learning is but one tool in the toolbox and finding the right algorithm remains critical as demonstrated by CRISPR-GNL, a Bayesian ridge regression solution that outperforms its deep learning counterpart, DeepCas9 (J. Wang et al., 2020).

However, a unique feature of deep learning is its ability to make preprocessing redundant in some circumstances. Algorithms like CNNs can decompose images containing objects at arbitrary positions/sizes/angles without the need for techniques like cropping, scaling and rotating; not only on the training set, but also for novel samples (Yann LeCun, 1995). Being able to uncover underlying patterns in arbitrary data, rather than requiring perfectly curated feature-sets is a useful capability in the CRISPR space.

1.10 Optimisations and insights

Despite algorithm choice being limited by the data being modelled, there will usually be multiple algorithms relevant to a given task. It is therefore important to compare the performance of models trained using different algorithms. However, even models trained by the same algorithm on the same data can differ, depending on options passed to the algorithm. Therefore, not only should models trained with different algorithms be compared against each other, but models trained from the same algorithm using different configurations, or hyperparameters.

1.10.1 Model hyperparameters

Hyperparameters, unlike model parameters (which are derived through training), are set by the researcher *a priori.* They modify how an algorithm learns from the data, and different configurations can result in improved, or worsened, model performance. Each algorithm has its own set of hyperparameters and each hyperparameter modifies a certain aspect of the training process. For example, the Random Forests algorithm includes “number of trees” and “maximum tree-depth”. Typically, altering a hyperparameter would alter the resulting model complexity. For example, where a high value for maximum tree-depth would result in a deep and complex tree, a low value for maximum tree-depth would result in a shallow and simple tree.

By trialling different sets of hyperparameters, one can identify the configuration to result in the optimal model. Grid search algorithms are provided by ML libraries to trial different sets of hyperparameters. However, this process can be expensive regarding compute time, scaling with the number of hyperparameters being tested, as well as the number of options for each

hyperparameter. Therefore, understanding the influence of hyperparameters can result in a reduced search space.

1.10.2 Types of error

Two types of errors influenced by hyperparameters are bias and variance. Although the aim is to minimise these errors, decreasing one (i.e. bias) will usually result in an increase of the other (i.e. variance) (Hastie et al., 2009). This is because bias and variance are related to model complexity. A complex model will have a high variance, whereas a simple model will have a high bias. As mentioned in the previous section, adjusting the “maximum tree depth” alters the model complexity, and where a shallow tree will result in a high bias and low variance, a deep tree will result in a low bias and a high variance. The aim, therefore, is to find the sweet spot where both forms of error are kept to a minimum.

These two errors have different consequences on the resulting model. A high bias (simple) model will perform poorly on all datasets. This includes training data, validation data and data from other sources. This is known as underfitting, and it arises because the model does not capture enough information. Conversely, a high variance (complex) model will perform well on the training data, and perhaps even the validation data if it is homogenous to the training data. However, a high variance leads to overfitting. This means that despite representing the data it was trained on well (or perfectly) it will not be generalisable to, for example, experiments from other laboratories. In other words, it captures information specific to the training dataset.

As well as inappropriate hyperparameters, complexity in a model can be modulated by noise or outliers in the training data. Noise can include features that happen to have a correlation with sgRNA efficiency in the training data, but not in general. Outliers, on the other hand, are samples that are dissimilar from the group they belong to. For example, a negative target (low-efficiency) that happens to have a sequence that is very different from other negative targets, in fact so much so, that it more closely resembles the sequence properties of positive targets (high-efficiency). This may be due to experimental error, missing features (like epigenetic information), or errors in preprocessing (such as representing different targets in different directions). In these cases, it may not be possible to train a model with a low bias or a low variance.

Another way to improve the error is to use a different model. Ensemble models, for example, can inherently decrease one error without increasing the other (Breiman, 2001b; Friedman, 2001). This is demonstrated by Random Forests, which typically consist of deep trees. Deep trees are complex with a high variance and low bias, making them prone to overfitting.

However, by bootstrapping (sampling with replacement) the data seen when training each tree, and by repeating this to train many trees, the resulting ensemble is a model with a low bias and low variance compared to any individual tree in the ensemble (Breiman, 2001b). The converse is achieved for gradient boosted regression trees. This model relies on an ensemble of simple (high bias, low variance) decision trees to result in a model with low bias and low variance (Friedman, 2001).

1.10.3 Quality datasets

Regardless of hyperparameters and other optimisations, algorithms must have access to a large and representative data set to train accurate models. For CRISPR experiments using the template-free repair pathways, large datasets are now available, with recently published datasets presenting 40,000 Cas9 samples (Allen et al., 2019) and 15,000 Cas12a samples (H. K. Kim et al., 2018). However, for other repair pathways (i.e. HDR), there is little data available, impeding the ability to accurately model these biological systems.

1.11 Gaining insights from CRISPR ML models

Training a model on irrelevant features can reduce a model's performance, but prior to training, it is not always obvious as to which features are relevant. For example, which, if any, epigenetic regulation properties influence sgRNA efficiency and should be included in training? One way to identify influential features is to selectively train on different subsets of features and subsequently observe variations in the model's performance as each feature is added. However, training models on different subsets of potentially thousands of features can be inefficient and time-consuming. More appropriate for identifying influential features are explainable models, such as logistic regression and tree-based methods (Breiman, 2001b; Ng, 2004). Such algorithms allow researchers to train a single model on all available features and subsequently rank features by their contribution to the model, or "feature importance".

Feature importance, as well as enabling researchers to only include relevant features, can be extended to "hypothesis generation". Whilst a feature ranking highly is not necessarily indicative of its biological influence over sgRNA efficiency, it can promote the design of further CRISPR experiments to gather support for the generated hypotheses. For instance, features such as position-independent (di)nucleotide count, location of target within the gene, and melting temperatures of the target have been demonstrated to contribute to models and therefore may be involved in DNA cleavage efficiency (Doench et al., 2016).

1.12 Room for improvement

The use of machine learning in CRISPR applications is evolving at a rapid pace, with multiple prediction tools being released every year. The broad availability of ML-based CRISPR tools has resulted in the need to empirically test CRISPR-Cas9 designs being replaced by *in silico* optimisation. In other words, researchers can design efficient sgRNAs algorithmically, optimising for maximum editing efficiency and minimum off-target effects.

But although each model aims to improve CRISPR experiments, prediction models are not a one-size-fits-all solution; hence why there are currently 20+ models in production. For example, while some models are simple and generalisable across organisms and cell types, others are more complex, capturing data like epigenetic information. Where most models predict Cas9 cleavage efficiency, others predict Cas9 editing outcome or Cas12a cleavage efficiency. Furthermore, the performance of each model varies regarding accuracy and precision. While some models perform well on paper, performing well on in-house generated data, they will perform poorly on data generated from other groups. So, despite the proliferation of models, there is still much room for improvement when applying ML to CRISPR.

1.13 Research objectives

Despite the abundance of CRISPR publications, there are still gaps in the literature regarding mechanisms of action. This can lead to non-optimal experimental design, resulting in inefficient editing or off-target effects. Machine learning has enabled the ability to efficiently model large datasets to result in both insights and predictions, which has resulted in numerous prediction tools. However, there are still gaps with current tools not modelling aspects of genome editing like repair pathways.

I aimed to identify features that modulate parameters like efficiency and specificity. This would involve using ML algorithms to model datasets and to gain insights. With data being one of the primary requirements for training ML models, this process was dependent on the availability of suitable datasets in the literature or the ability to create novel datasets where none exist. I aimed to release ML models into the public domain as design tools for use by researchers to enable more effective CRISPR experiments.

The results of my work span chapters three to five, with each covering a unique aspect of CRISPR editing, from HDR to induce single nucleotide variants in Chapter 3, to the efficiency and generalisability of simultaneous HDR events in Chapter 4, to the specificity and efficiency of Cas12a in Chapter 5. The objectives of each chapters are as follows:

- In Chapter 3 I investigated the efficiency of the HDR pathway in CRISPR experiments from a dataset I curated. HDR enables knock-in experiments but is inefficient compared to other less-versatile repair pathways. With no prediction tools available for designing efficient HDR experiments, I aimed to train one using ML.

- In Chapter 4 I explored the efficiency of using simultaneous HDR events to enable conditional knockouts. I aimed to identify whether the poor efficiency of this technique is a result of low HDR efficiency or other factors. Because the dataset included experimental data from 19 laboratories, I was able to test for the influence of laboratory-specific properties on efficiency.

- In chapter 5 I focused on CRISPR-Cas12a as previous evidence suggested it to provide benefits over CRISPR-Cas9. I aimed to quantify the benefits of Cas12a through the analysis of *in silico* and *in vivo* results. I also aimed to enable more efficient CRISPR-Cas12a experiments by training prediction models for editing outcome and efficiency.

Chapter 2 – Methods

2.1 Overview

Here I describe methods for each of the chapters in my thesis. Despite each chapter containing different methods, there is some commonality between them. Firstly, regardless of the data source, the overarching aim of each section is to create a curated and comprehensive dataset. A dataset is defined as a collection of samples, where each sample are the editing results from a unique locus. Each sample is labelled and includes a set of features. The label is generally cleavage efficiency or homology directed repair (HDR) efficiency, and features include potentially predictive information like single guide RNA (sgRNA) sequence or chromatin accessibility. Each dataset enabled analysis and modelling of the respective CRISPR system using machine learning (ML). With the resulting models it possible to identify properties that may influence the efficiency as well as predicting the efficiency for novel targets.

For each section in this chapter, implementation details can be found in the code repository. This is available at https://github.com/aydun1/aidan-anu-thesis with material from each chapter organised by directory. Code was written with reproducibility in mind and is distributed freely for non-commercial use. Code for data analyses was written in the Python programming language. Multiple Python libraries were used, including:

- pandas, for data analysis and manipulation (McKinney, 2010; The pandas development team, 2020),
- scikit-learn, for ML algorithms (Pedregosa et al., 2011),
- SciPy, for statistical tests (McKinney, 2010),
- statsmodels, for statistical modelling (Seabold & Perktold, 2010) and
- seaborn, for data visualisation (Waskom et al., 2020).

2.2 Chapter 3 methods

This section outlines the steps taken for processing and modelling CRISPR-Cas9 HDR efficiency. The data that I aimed to model were experimental results from micro-injection sessions. Each session included attempts to induce a specific single nucleotide variant (SNV) in mouse embryos. These attempts were made through HDR, using a CRISPR-Cas9 sgRNA to induce a DSB with a single-stranded oligodeoxynucleotide (ssODN) donors to define the desired mutation. The source code is available in the “chapter3” directory.

2.2.1 Curating data

Each microinjection session was stored in a separate document, so I manually collated the data into a spreadsheet. This resulted in a table, with one row per document. For each row, columns stored properties regarding the micro-injection session. I captured the following information for each session:

- sgRNA sequence
- ssODN sequence
- distance of mutation from the PAM sequence
- number of attempts
- observed mutations

Generally, each micro-injection session aimed to induce a mutation at a unique locus, although some sessions targeted the same locus. Therefore, to reduce the potential for "sampling bias", I merged these sessions. This involved summing up the number of attempts and joining the list of observed mutations for every attempt at a specific locus, regardless of which session it was a part of.

2.2.2 Processing data for training and validation

To enable training a model for efficiency prediction, an efficiency value was required for each sample. With the aim being to investigate HDR efficiency, I defined HDR efficiency as the ratio of HDR to NHEJ. For example, a session with evidence of HDR, but no evidence of NHEJ is 100% efficient. For this I used the number of times the desired point mutation was observed at a given locus (which indicates the cell repaired the DSB using HDR and the ssODN) divided by the total number of mutations (desired or otherwise). Dividing by the total number of mutations, rather than the total number of attempts removes "CRISPR cleavage efficiency" as a confounding variable. For this reason, I discarded samples with no mutations because if CRISPR fails to cleave the target, it is not possible to calculate the HDR to NHEJ ratio. Each sample now has a value from 0 to 1, where 1 indicates "100% HDR, 0% NHEJ", and 0 indicates "0% HDR, 100% NHEJ".

To enable binary classification (high vs. low HDR), I divided samples into two groups based on HDR efficiency. I defined a threshold with the aim of producing balanced classes, i.e. an equal number of high- and low-efficiency samples). This threshold was the median HDR efficiency value, 0.199.

Because I aimed to create a generalisable HDR efficiency prediction tool, the focus was on nucleotide features. These features included the sgRNA sequence and the ssODN template sequence. I processed the sequence as per the “Sequence processing” section. However, to enable the analysis of different components (i.e. the 5’ region of the ssODN versus the 3’ region), each of these regions were processed separately.

2.2.3 Machine learning and statistical analysis

To model the data, I used the Random Forests algorithm, as per the “Machine learning” section. The primary metric I used for this chapter was the out-of-bag (OOB) error. This metric takes advantage of one of the properties of the Random Forest algorithm, bootstrap aggregating (bagging). With bagging, each tree is trained on only a subset of samples. Therefore, each tree can be tested on samples unseen to that tree. This is repeated for every tree throughout the training process. Finally, the average of the errors for each tree results in the OOB error. I partitioned the dataset using cross-validation as per the “Cross-validation” section for further validation of the generalisability of the models. I trialled different hyperparameters using a grid search to iteratively compare the performance of different combinations of hyperparameters.

2.2.4 Model validation

With a lack of HDR data in the literature, I validated the model with the highest performance on more-recent data. This data was generated after training the published model and included fifteen samples, generated in the same way as the published samples. The most recent sample in this validation set was from October 2018. This validation set is referred to as V1. To validate my model against this set, I compared predicted efficiencies to real efficiencies to calculate the accuracy, precision, and recall.

2.2.5 Additional data

Since publishing the original manuscript, more data had been generated. I used this data, dated 2019 and 2020, for an additional validation set (V2). It was processed in the same way as V1 and used to validate the published model using the same metrics. With two validation sets, I then trained new models on the published data and V1, validating the highest performing model on V2.

2.3 Chapter 4 methods

This section covers the analysis of datasets generated from multiple groups regarding the efficiency of two simultaneous HDR events. The curated dataset and source code are available in the repository with the prefix “chapter4”.

2.3.1 Processing data for training and validation

For this section, the data was available in two spreadsheets, now available at (Gurumurthy et al., 2019). The first, published as “Supplementary Table 1” contained the sgRNA and ssODN sequences for each target. The second, published as “Supplementary Table 4” contained experimental results.

To calculate the efficiency of two simultaneous HDR events, I used the “correctly targeted” column divided by the “live born pups” column in the latter spreadsheet. I also created a binary representation of this efficiency label by grouping targets into two classes, “positive” and “negative”. To create classes that were as balanced as possible, I assigned loci with one or more successes to the “positive” class, and loci with zero successes to the negative class.

To calculate the efficiency of single HDR events, I used the number of *cis loxP* insertions divided by the “live born pups” column. I did this for each of the 5’ and 3’ target sites at each allele. To calculate the number of *cis loxP* insertions, I manually curated editing outcomes from the second spreadsheet.

Most features were already suitable for modelling, being numbers. However, I processed the ssODN and gRNA sequences as per the “Sequence processing” section. For the ssODN, I did this for the 5’ and 3’ region separately.

The features used in this section are:

- sgRNA sequence x2
- ssODN sequence x2
- ssODN length x2
- distance between targets
- ssODN concentration
- live-born mice

2.3.2 Statistical analysis and ML

To identify differences between two populations, I used the Mann-Whitney U test from SciPy. For more than two populations, I used the Kruskal-Wallis rank-sum test from SciPy. This tests whether one or more populations is different, but not which population(s) are different. So, when this test identified a different population, I iteratively removed one population at a time and tested the remaining populations until the test identified no significant differences. To compare the correlation between linear variables, I used the coefficient of determination (R^2) from statsmodels.

To model the data, I used the Random Forests algorithm, as per the “Machine learning” section. Because of the unbalanced classes (a low number of 1s vs. 0s), I used a custom class_weights function. This was to compensate for classifiers being biased for the majority training class (Japkowicz, 2000). A biased classifier means that when predicting samples from a 50/50 distribution, a disproportionate number of samples will be assigned to the majority training class. By adding a weight to the minority class, such misclassifications carry a higher penalty which can result in an improved model (Chen et al., 2004). The custom class_weights function was specified as an input when defining a Random Forest model.

2.3.3 Model validation

As in the previous chapter, I used the OOB error to quantify the performance of each trained models. To identify important features, I used the “feature_importances_” property of the Random Forest model. I trained models using cross-validation as per the “Cross-validation” section.

2.3.4 Success forecaster

I created a statistical model to forecast the number of successful attempts. This considers the probability of two simultaneous successful events based on the observed probability of each single event. To consider forecasting from potentially low sample sizes, the confidence interval, with a confidence level of 95%, is included in the calculation. The full code is available in the repository.

2.4 Chapter 5 methods – off-target analysis

This section details the off-target comparison between Cas9 and Cas12a. Off-targets are genomic regions that are cleaved outside of the intended target. Due to mismatch-tolerance in the sgRNA, off-targets can exist even if the target sequence is unique in the genome (Anderson et al., 2015). Off-target effects can be minimised by considering the uniqueness of targets by using computational tools to identify potential off-targets. Post targeting, off-target cleavage can be identified using methods like GUIDE-seq (Tsai et al., 2015) and DISCOVER-seq (Wienert et al., 2019).

2.4.1 *In vivo* potential off-target analysis

I analysed ten human genes, selected from the top ten most studied human genes by citation (Dolgin, 2017). I postulated this to be a more representative sample of real-world gene editing targets than randomly selected genes. I used a custom script to identify Cas9 and Cas12a targets in exons in each gene. Targets were identified based on the presence of PAM sequences. To identify potential off-targets, I used Cas-OFFinder (Bae et al., 2014).

GT-Scan (A. O'Brien & Bailey, 2014) is another tool which performs this function, however where Cas-OFFinder can identify potential off-targets with any number of mismatches, GT-Scan is limited to three. However, I compared the output with up to three mismatches to ensure the results from Cas-OFFinder were equivalent to GT-Scan, which I developed for a previous research project. For GT-Scan I used the default settings.

For my analysis of potential off-targets, I set "Mismatch Number" to five, because off-targets had previously been identified with up to five mismatches (Y. Fu et al., 2013). As input, I used the list of target sequences identified using my custom script. The output was a file containing every potential off-target for every target. To enable analysis of this file, I processed it using a custom script to create a summary for each target. The summary is in a tab-separated format, listing the number of potential off-targets with each number of mismatches (0 to 5) for each target. Due to the small size of this summary file, I used Excel to perform further analyses and visualisation on these results.

2.4.2 In vitro off-target analysis

Here I compared Cas9 and Cas12a off-targets using results from two previous GUIDE-Seq experiments. I acquired Cas9 data from (Kleinstiver, Pattanayak, et al., 2016) and Cas12a data from (Kleinstiver, Tsai, et al., 2016). I manually curated read counts for cleaved genomic sites from these papers into a spreadsheet. I grouped them by "target" and "number of mismatches" for plotting and analysis. I used Excel to analyse and visualise the data.

To compare cleaved off-targets to potential off-targets, I used Cas-OFFinder to identify potential off-targets with up to five mismatches for each target. I also used this data to ensure that these experimental targets were not outliers when compared to my *in vivo* results.

2.5 Chapter 5 methods – editing outcome prediction

This section covers the process for analysing the mutational outcome from NHEJ with Cas12a. That is, the specific mutation arising from CRISPR-Cas at a target, be that a deletion, insertion, or single nucleotide change. The aim was to identify the feasibility for an outcome prediction model for Cas12a. The curated dataset and source code are available in the repository with the prefix "chapter5".

2.5.1 Data acquisition

Here I used raw read data from (H. K. Kim et al., 2018). This data includes both lentiviral integrated target sequences (synthetic targets) and endogenous target sequences (genomic targets). The SRA accession number is SRP107920 with accession numbers for individual

runs available in Supplementary Table 1. The data includes both treated and control reads for each dataset. I used fastq-dump from the SRA toolkit to acquire read data from the SRA. The command to download the reads depended on the alignment type, with the following used for single alignments and paired-end alignments, respectively:

```
fastq-dump SRRxxxxxxx
```

```
fastq-dump -I --split-files SRRxxxxxxx
```

2.5.2 Aligning reads

To quantify the mutational landscape, reads needed to be aligned to a reference genome. For the synthetic reads, reads first needed to be trimmed of barcode and adaptor sequences. For this task I used a custom script. The endogenous reads were already trimmed. I aligned both sets of reads to the human GRCh38 genome using Bowtie 2 (Langmead & Salzberg, 2012).

```
bowtie2 -x /genomes/ensembl.release-90/Homo_sapiens.GRCh38 -1
SRRxxxxxxx_1.fastq -2 SRRxxxxxxx_2.fastq -S out_file.sam --very-sensitive -p 10
```

I used SAMtools (H. Li et al., 2009) to convert the output from Bowtie 2 into a BAM file, sort the BAM file and index it. The result here was a BAM file sorted by genomic region, and a corresponding index file.

I repeated this process for treated and control samples.

2.5.3 Quantifying reads

I used the computational tool GOANA (in review) to quantify the mutational landscape of the newly aligned reads. GOANA is a program which accepts aligned reads and outputs sgRNA efficiency and allele frequencies for a set of predefined locations. It retrieves the list of predefined locations from a BED file. However, rather than identifying mutant alleles based on variations to a reference genome, GOANA identifies mutant alleles based on variations to control reads. The control reads can be day zero reads or reads from untreated samples. The benefit of using control reads instead of a reference genome is that the chance of pre-existing differences to public reference genomes being incorrectly labelled as CRISPR-induced mutations is minimised.

The output from GOANA lists alleles and allele frequencies for each genomic region. However, alleles can have more than one mutation (i.e. a single nucleotide change and a deletion or an insertion and a deletion) and it is the mutation frequency that I am interested in. But because

GOANA excludes alleles with a low read coverage by default, common mutations will be filtered out if they occur on the same allele as rare mutations. To mitigate this, the *-mr 0* argument instructs GOANA to include all mutant alleles, regardless of read coverage.

python3 GOANA.py regions.bed control.bam treated.bam -o output.file -mr 0

I used a custom script to parse the output and read it into a pandas DataFrame. For each target, I recorded:

- sgRNA sequence and target-adjacent nucleotides
- Target coordinates
- Cleavage efficiency
- Mutations

2.5.4 Statistical analysis and validation

To ensure the datasets were representative of CRISPR Cas12a editing in general, I compared the distributions of insertions, deletions, and single nucleotide variances. To identify significant differences between distributions, I used Cohen's *d* (Cohen, 1977). This is a standardised measure of the difference between two means. As well as different editing outcomes, I also compared differences between insertions and deletions of different lengths. This was for the HT 1-1 dataset and the HEK-plasmid dataset.

To train a model to predict editing outcome, I trained Random Forest models as per the "Machine learning" section. One difference from the previous sections is that I used multiclass classification. This enabled more than two discrete outcomes to be predicted, i.e. the single nucleotide variant (A, C, T or G) or deletion/insertion length (1, 2, 3, etc.). I trained models using cross-validation as per the "Cross-validation" section. However, as a multiclass model, to quantify the outcome I used one-vs-all metrics. This produces a score based on binary values, where the correct outcome is true, and incorrect outcomes (regardless of which incorrect outcome) are false. This produced a value for each class. For the one vs. all Receiver operator characteristic (ROC) curves, these values where averaged.

2.6 Chapter 5 methods – sgRNA efficiency

This section covers the process for modelling and predicting Cas12a efficiency from raw read data. This includes acquiring the data from public data sources, processing the data, and finally modelling the data. Because the data comes from different sources, this section also includes details on merging datasets.

2.6.1 Downloading reads for model training

To train my models I used the monocistronic CRISPR/Cas9 library from (J. Liu et al., 2019). This is a pooled-library knockout screen with an SRA accession number of SRP181683. The data included reads at different timepoints (weeks 1 to 4), as well as reference reads. The data from SRA contained the time points in arbitrary concatenations so I downloaded the original files from the Google Cloud Platform (GCP). Here, each timepoint was stored across two files (part1 and part2). These files are listed as *Mini-human* in Supplementary Table 1. Because these files are hosted as "requestor pays", downloading them requires a valid GCP account with billing enabled.

```
gsutil -u username cp gs://sra-pub-src-3/file_path.fastq.gz /out_path
```

2.6.2 Aligning reads for model training

The first step for processing these reads was to align them to a human reference genome. Because of the short read-lengths (20nt), I created a custom version of the GRCh38 reference genome. This custom genome contained just the 20nt sgRNA sequences. The aim was to minimise the likelihood of the short reads aligning to incorrect regions. To create the custom genome I used a script to generate a FASTA file with an entry for each of the 2,061 targets from (J. Liu et al., 2019). I then indexed the FASTA file with bowtie2-build.

```
bowtie2-build cpf1_mono.fa cpf1-mono
```

Next, I aligned the reads from each timepoint to the custom GRCh38 genome using Bowtie 2. Each timepoint had reads in two files (part1 and part2). So, for each alignment I specified both files as a comma-separated list.

```
bowtie2 -x /genomes/cpf1-mono -U
timepointX.part1.fastq,timepointX.part2.fastq -S out_file.sam --very-sensitive -p
10
```

2.6.3 Inferring sgRNA efficiency for model training

As a pooled-library screen, efficiency can be inferred from the log-fold change in read-count, for each CRISPR target, over time. The theory is that editing "essential genes", i.e. genes that are required for cell-viability, will result in non-viable cells. This in turn will result in lower cell counts at later time points. Because of this, reads from targets with a low efficiency will be relatively high (with few edits to disrupt cell viability) compared to reads from targets with a high efficiency (with many edits to disrupt cell viability). In other words, there should be an

inverse correlation between sgRNA efficiency and change in read count over time. A confounding factor here is the “essentiality” of a gene. Because if edits do not reduce a cell’s viability, despite disrupting a gene, then reads from that target will be remain high, regardless of sgRNA efficiency. To minimise the effects of this confounding variable, I only included genes with a high Bayes Factor (BF). BFs are a statistical measure used to indicate the likelihood of a gene belonging to an essential or non-essential distribution, which can be calculated using computational tools like BAGEL (Bayesian Analysis of Gene EssentiaLity) (Hart & Moffat, 2016). I downloaded BFs from “The Toronto KnockOut Library” (Hart et al., 2015) and merged them with CRISPR targets based on gene name.

2.6.4 Model validation

To validate the Cas12a efficiency model, I used the previous HEK-plasmid dataset. In addition, I downloaded the HEK-lenti and HCT-plasmid datasets, following the same methodology. I also aligned these reads, generated a BAM file and index, and quantified reads using GOANA, following the same methodology used for the HEK-plasmid dataset.

2.6.5 Integrating samples with chromatin accessibility data

An additional feature observed to modulate sgRNA efficiency is chromatin accessibility (Xu et al., 2015). Quantified by DNase hypersensitivity, chromatin accessibility indicates the accessibility of a target due to chromatin state. To incorporate this information into the datasets, I downloaded the following narrow-peak datasets from the ENCODE portal (Davis et al., 2018; The ENCODE Project Consortium, 2012):

- ENCFF127KSH (HEK293T)
- ENCFF912FSU (HCT116)

These datasets specify regions of DNA that are DNase hypersensitive for the respective cell types (HEK293T and HCT116). To integrate this data with CRISPR targets, I use a custom script to identify whether CRISPR targets lie within DNase hypersensitive regions. The script assigns a 1 to targets that are in hypersensitive regions and a 0 to targets that are not.

2.7 Common methods

2.7.1 Sequence processing

Most machine learning algorithms are unable to handle DNA sequences (strings) directly so therefore I tokenised the sequence for each sample into a format that is suitable for modelling. This is achieved using a custom function that is applied to each row in the DataFrame. The input is a DNA sequence, and the output is an array of numerical values. This array represents

the DNA sequence string using, for example, “global nucleotide” counts and “positional nucleotide” counts. Global nucleotides represent the count of each nucleotide in the string whereas positional nucleotides represent the nucleotide at each position (Table 1). As well as single nucleotides, I also include global/positional dinucleotides (i.e. AA, AT, CT, etc.) and GC content.

	Sequence	Type	Features
Positional nucleotides	AAAG	Binary	{**1A**: 1, **2A**: 1, **3A**: 1, **4G**: 1}
Positional dinucleotides	AAAG	Binary	{**1AA**: 1, **2AA**: 1, **3AG**: 1}
Global nucleotides	AAAG	Discrete	{**A**: 3, **G**: 1}
Global dinucleotides	AAAG	Discrete	{**AA**: 2, **AG**: 1}
GC content	AAAG	Continuous	{**CG**: 0.25}

Table 1 – different tokenisation methods applied to the sequence “AAAG”. These methods enable machine learning algorithms to model nucleotide sequences.

I processed individual sequences separately. So, for example, to model the sgRNA, ssODN and nucleotides adjacent to the target I would apply the above function to each of these components separately. This minimises the sequence of relevant components being diluted by irrelevant components.

2.7.2 Machine learning

To train models I used scikit-learn in Python. This consisted of three steps:

1. defining a model,
2. fitting the model on training data and
3. predicting labels for test data.

1) Defining a model involves specifying the algorithm and the hyperparameters. Hyperparameters are parameters that are user-specified when defining the model, as opposed to parameters that are learned by the ML algorithm from data. For example, “max-depth” is a hyperparameter used by the “DecisionTreeClassifier” algorithm in sci-kit learn. As the name implies, it defines the maximum depth that the resulting decision tree may be trained to. Each algorithm has default values for hyperparameters, however by tuning the hyperparameters it may be possible to train an improved model.

2) The “fit” step involves training the model on data. This process simply requires two lists, one of the labels from each sample and one of the features. This can be performed on the entire dataset or a slice of the dataset, with the latter being useful for model validation.

3) The “predict” step takes the trained model and predicts the label for unlabelled samples. Therefore, this process just requires a list of features as the input. For the final model, the output from this step is returned to the user. However, during model design, I used the output from this step for model validation.

2.7.3 Validation

To validate a model, or quantify its performance, prediction values are compared to truth values. For example, the predicted sgRNA efficiency to experimentally measured sgRNA efficiency. Every trained model was validated to quantify its performance and compared against other models. This is achieved through an accuracy or error score for which there are various measures that I use. For regression, an option is the mean squared error (MSE). This is the average of the squared differences between the prediction and truth values for each sample, where values closer to zero indicate predictions closer to the truth. MSE is a function included in the scikit-learn Python library which I call on the two lists of values, i.e. mean_squared_error(truth, predicted). However, to be able to calculate a prediction measure for a model, samples which were not included when training said model were required. One option was to divide the data into two discrete sets, a training set and a test set. However, another option was cross-validation.

2.7.4 Cross-validation

Rather than dividing samples into two discrete sets for validation, I used k-fold cross-validation to quantify the performance of models (Geisser, 1975). Using 5 folds cross-validation, data is partitioned across samples, into five groups, where each group contains one fifth of the samples. Subsequently I trained models on each combination of four groups, and test on the fifth. This allows us to evaluate the prediction error with better generalisation to novel data than a train/test set. For classification models I used “StratifiedKFold”(Pedregosa et al., 2011) to create the folds, as this preserves the distribution of positive and negative samples. Each time with the same algorithm with the same hyperparameters, however each trained on a different subset of data. Next, I validated each model on the slice of data that it was not trained on. This resulted in a score (such as MSE) for each model, on which I calculated the average to quantify the overall performance.

2.7.5 Feature importance

An additional use of some ML algorithms is the ability to identify which features are correlated with the label. For Random Forest models, after training a model, feature importances can be retrieved using the feature_importances_ parameter. This returns a list of features (be that nucleotides or reagent quantities) with an assigned weight value for each.

2.8 Visualisation

2.8.1 Confusion matrix

To visualise classification predictions, I used a confusion matrix. For two-class predictions (i.e. high/low) this is simply a 2x2 matrix where rows indicate prediction values and columns indicate truth values. In effect, this presents the number of true positives, false positives, true negatives, and false negatives. Here I used confusion_matrix from scikit-learn which takes a list of truth values and a list of prediction values.

2.8.2 Percentile rank

The percentile rank presents how ranked prediction values compare to ranked truth values. The percentile rank illustrates whether the ordering of predictions is correct. Here I used the percentileofscore function from the SciPy statistics module to calculate the percentile ranks prediction and truth values and subsequently plot the results with Matplotlib.

2.8.3 ROC curves

Receiver operator characteristic (ROC) curves plot the true positive rate against the false positive rate. It represents the discrimination ability of a model, i.e. a model's ability to distinguish between high and low efficiency samples (Hanley & McNeil, 1982). The area under the ROC curve (AUC) provides a quantitative measure of this metric where 1 indicates a perfect discrimination and 0.5 indicates that predictions are random. I used roc_curve and roc_auc_score from the sklearn.metrics to compute these values.

Chapter 3 – Unlocking HDR-mediated nucleotide editing by identifying high-efficiency target sites using machine learning

The CRISPR-Cas9 system enables researchers to introduce precise genomic changes through different means. In this chapter I investigated different methods, their efficiencies, and I trained a prediction model to improve the efficiency of using CRISPR to introduce precise single nucleotide variants.

3.1 Introduction

As well as inducing arbitrary mutations, CRISPR systems can be used to induce precise single nucleotide variants (SNVs). This is possible through different means, such as base editing, prime editing, or CRISPR-mediated homology directed repair (HDR). Base editing is an efficient method that relies on a cytidine (C) deaminase or adenosine (A) deaminase. These enzymes enable a limited range of changes by converting C•G to T•A, or A•T to G•C, respectively (Gaudelli et al., 2017; Zheng et al., 2018). By fusing one of these enzymes to a catalytically inactive Cas9 or a Cas9 nickase, the resulting fusion can induce deamination at the CRISPR target. Different groups have demonstrated efficiencies of 44% to 100% (M = 82%) in mice, rabbits, rats and human embryos (Z. Liu et al., 2018; Y. Ma et al., 2018; Ryu et al., 2018; Zeng et al., 2018). Efficiency modulators include the sequence composition and the position of the SNV relative to the target protospacer adjacent motif (PAM). The position relative to the PAM is relevant because base editing is only efficient at a limited number of positions at the CRISPR binding site, known as an editing window (Komor et al., 2016). Different editing windows are possible through different systems (Jiang et al., 2018) or alternate cytosine deaminases (Cheng et al., 2019). However, where larger editing windows may enable more flexibility in potential targets, larger windows can have the side effect of more bystander mutations. Bystander mutations are mutations that result from deamination of other nucleotides within the editing window. As well as SNVs, resulting from deamination, bystander mutations can also include deletions (H. K. Lee et al., 2018). As well as the size of the editing window, the prevalence of bystander mutations also depends on the base editor in use (H. K. Lee et al., 2020).

Although base editing is efficient, bystander mutations, the limited range of changes and limited number of targets can restrict its application. These three considerations mean that only a limited number of single nucleotide variants will be possible with base editing. For changes that are not possible with base editing, prime editing or CRISPR-mediated HDR may be more appropriate. Prime editing, like base editing, relies on a catalytically impaired Cas9

(Anzalone et al., 2019). It is less efficient than base editing but has the benefit of enabling specific base substitutions or small indels with minimal undesired effects. However, prime editing faces limitations when the aim is to make large changes (Anzalone et al., 2019).

The third option is via CRISPR-mediated HDR. HDR is one of the endogenous DNA repair pathways that can result from CRISPR-induced cleavage. Generally, HDR repairs cleavage, or double-strand breaks (DSBs), using a homologous DNA template (Mao et al., 2008; Pardo et al., 2009). This means that HDR is usually an error-free repair pathway. However, by including a synthetic DNA template that is mostly homologous to the target, HDR can introduce any differences into the target through homologous recombination. Through the template, HDR is currently the most versatile editing solution as it allows researchers to make nearly any change, from SNVs, to insertions of thousands of nucleotides (B. Wang et al., 2015).

One downside of HDR compared to base-editing is its relative inefficiency. This is because HDR is in direct competition with other repair pathways, including non-homologous end joining (NHEJ) (Mao et al., 2008; Sargent et al., 1997) and microhomology-mediated end joining (MMEJ) (J.-L. Ma et al., 2003). However, despite factors surrounding HDR kinetics remaining unknown, evidence suggests that the initial 5' to 3' resection of the blunt ends present at a DSB guarantees an outcome of HDR over the other repair pathways (Pâques & Haber, 1999; Valerie & Povirk, 2003). Another limiting factor is that HDR is restricted to the late G2 and S phase of the cell cycle (Symington & Gautier, 2011), limiting the opportunities in which it can occur. Finally, HDR can be negatively influenced by somatic or sporadic mutation in any of the genes involved in the HDR pathway. This includes genes involved in the MRE11/RAD50/NBS1 (MRN) complex, which are essential for resection (Taylor et al., 2009), as well as RAD51, BRCA1 or BRCA2 (Ransburgh et al., 2010; Stark et al., 2004). Therefore, HDR may not be possible when working on organisms or cell-lines with pre-existing mutations in these genes or when targeting these genes.

Because of the versatility of HDR, the ability to computationally identify optimal targets would enable researchers to perform a wide range of changes more easily. But although computational tools existed for predicting CRISPR-Cas9 cleavage efficiency (Cong et al., 2013; Haeussler et al., 2016; Stemmer et al., 2015), none existed for predicting the efficiency of CRISPR-mediated HDR. This absence of tools was possibly a result of unknown influencers of HDR efficiency, but also due to a lack of data regarding CIRSPR-induced HDR results.

During my PhD, I identified factors that influence Cas9-mediated HDR efficiency using machine learning on a novel fit-for-purpose dataset (A. R. O'Brien et al., 2019). From these

insights, I trained a model using machine learning (ML) to enable researchers to identify the optimal sgRNA for inducing a specified SNV. This was released as CUNE (Computational Universal Nucleotide Editor) and was the first built for purpose tool to identify optimal HDR targets. Since then, I gained access to an additional collection of experimental data.

Here, I extended on my work as enabled by the additional data. Using this data, I aimed to:

- create a larger dataset of HDR experiments
- validate my CUNE model on unseen samples
- train a new model on the larger dataset
- perform a feature comparison between models
- release the optimal model in an update to CUNE

Because my published model performed well on the published validation data (accuracy, 0.773), I hypothesised the same would prove true using more-recent unseen data. Also, when training my published model, the small training sample size (30 loci) resulted in features like local nucleotide composition degrading model performance. I therefore hypothesised that the larger sample size enabled by combining the published data and additional data would enable further insights into features that influence HDR efficiency.

3.2 Results

3.2.1 An improved dataset of genome-wide HDR efficiencies

I curated a dataset from 186 mouse editing experiments, conducted from 2015 to 2020. The aim of each experiment was to induce a single nucleotide variant (SNV) into mouse embryos using Cas9-mediated HDR. The target SNV, as defined by a single-stranded oligodeoxynucleotide (ssODN) sequence template, varied between experiments. For each experiment, from 1 to 34 (M = 6.15, SD = 4.50) mice were sequenced, and the resulting mutations recorded. Although there were 186 experiments, there were only 108 different SNVs. This is because although 68 target SNVs were the aim of just one experiment, the remaining 40 were the aim of two or more repeated experiments. Repeated experiments were combined in the preprocessing stage, leading to a unique set of ssODN/sgRNA combinations. These ssODN/sgRNA combinations are referred to as samples. After preprocessing, there were 63 samples.

The published data contained 45 samples (30 train, 15 validation). The extra 18 samples made up the additional data. This equated to a 40% increase in sample size. In total, 536 mice were sequenced with each sample containing on average 8.51 mice (Table 2). Although samples

were not random, instead being chosen to induce desired SNVs, most target sites were in different genes with nearly all chromosomes being targeted at least once. The 63 samples covered 53 genes and 16 chromosomes with each sample including the following features:

- sgRNA sequence
- ssODN sequence
- distance of SNV from PAM
- methylation status

		Published	Additional	Combined
General	Mice	429	107	536
	Samples (unique ssODN/gRNA combinations)	45	18	63
	Mice per sample (average)	9.53	5.94	8.51
	Genes	37	18	53
	Chromosomes	14	12	16
ssODN	Length (average)	158.77	137.89	152.76
	3′ arm length (average)	80.17	68.61	76.87
	5′ arm length (average)	77.60	68.11	74.89
	GC content (average)	52.98%	51.17%	52.40%
Efficiency	HDR (average)	0.286	0.37	0.327

Table 2 – the additional dataset includes results from HDR experiments in 1,143 embryos. This includes 64 ssODN/gRNA combinations (with approximately 18.85 embryos each). The median HDR efficiencies are presented for each ssODN/gRNA combination for the published and additional data, being 0.286 and 0.5, respectively. Overlap between genes and chromosomes in each dataset is the reason for combined values not being a sum of the published and additional values.

Although the published (45) and additional (18) datasets shared similarities, such as ssODN GC content, there were also differences. One difference was the distance of the SNV from the PAM. In the published dataset, the distance ranged from 9 nucleotides downstream to 30 nucleotides upstream of the first sgRNA nucleotide (Figure 6a). In the additional samples, the range decreased to 3 nucleotides downstream and 18 nucleotides upstream. This means that every SNV in the additional dataset was within the target PAM or protospacer. The interquartile range of distances was also smaller and closer to the PAM.

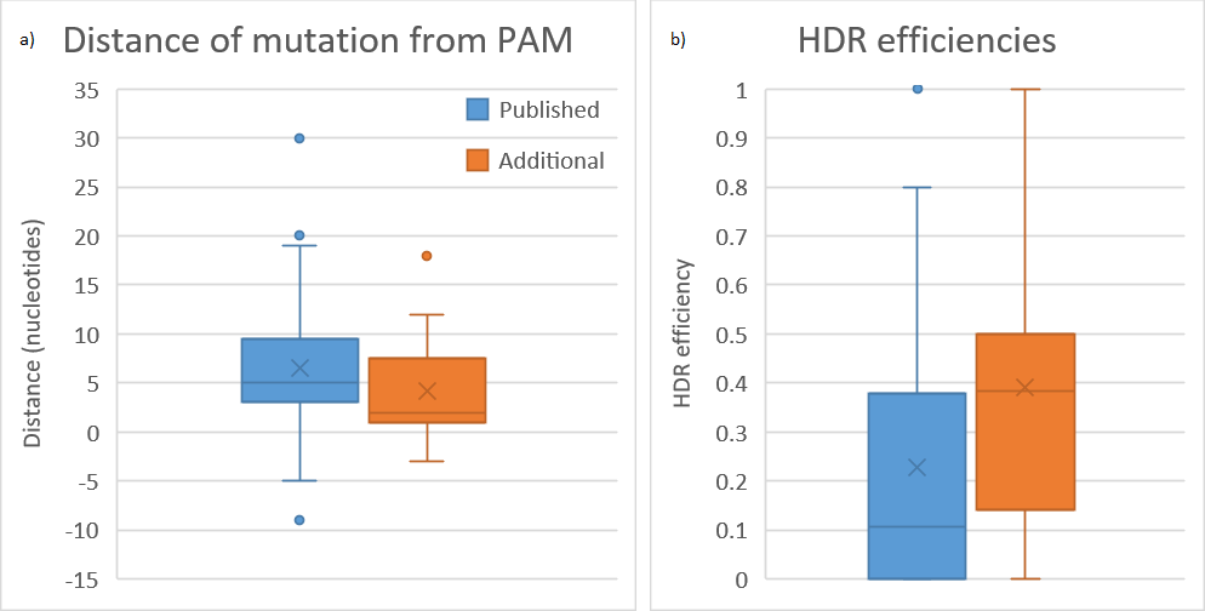


Figure 6 - differences between the published and additional datasets. a) Distances are between the desired SNV and the PAM. A value of 0 represents the first nucleotide in the protospacer. Negative values extend downstream (3') where positive values extend upstream (5'). Values from 0 to 19 represent the protospacer and -1 to -3 represent the PAM. b) the HDR efficiency is higher in the additional samples, with an average of 0.39 compared to 0.22.

To label the samples, I calculated the HDR efficiency for each one. This value was calculated for each sample from the 8.51 mice included in each sample. To enable this, I labelled mice carrying the desired SNV as having been repaired via HDR, and mice carrying insertions or deletions (indels) as having been repaired via NHEJ or MMEJ. This was on the basis that the latter two repair pathways generate indels at DSBs, whereas HDR repairs DSBs according to a template (in this case the ssODN) (S. Lin et al., 2014). The efficiency value for each sample was the number of mice with HDR repairs divided by the number of mice with any mutation. So, for a given sample, if 5 mice presented the desired SNV, and 25 mice presented any mutation, the HDR efficiency would be 0.2.

On the published training set of 30 samples, the median HDR efficiency was 0.20 (A. R. O'Brien et al., 2019). This value increased to 0.29 across the published 45 samples (Figure 6b). On the additional 18 samples, this value increased further, to 0.39. The increase suggested later experiments were more successful. Although the latter efficiency value is greater than what is typically observed, previous HDR experiments have demonstrated the variable nature of HDR efficiency. Observations have ranged from 1 to 40% (Aird et al., 2018; Guo et al., 2018), or higher than 50% under certain experimental conditions (G. Li et al., 2017). The likely reason for the increased efficiency in the additional data over the published data is that previously successful experiments influenced later experimental design, leading to the greater rate of success.

Other differences between the published and additional datasets included the number of mice sequenced for each sample. This number was nearly halved on the additional data, dropping from 9.53 to 5.94 (Table 2). This supported my previous observation of later experiments being more successful. Because the aim of each experiment was to generate a mouse with the desired SNV, rather than to generate a dataset, a lack of the desired SNV would result in more editing attempts, and more mice, until a success was achieved.

3.2.2 Validation of previous HDR model

The model I trained for CUNE was trained on the sgRNA sequence and the 3' ssODN arm (A. R. O'Brien et al., 2019). It was validated on data generated from the same group, but at a more-recent time point. This validation set, V1, included 15 samples. The prediction accuracy on this set was 0.733 (Table 3). I validated the model on additional data generated in 2019 and 2020. This set, V2, included 18 samples. The prediction accuracy on this set was 0.667, which is equivalent to two thirds of the samples being classified correctly. The precision and recall were both 0.75, which is a lower precision than the published validation set, but a higher recall.

Validation set	Samples	Precision	Recall	Accuracy
V1	15	0.889	0.727	0.733
V2	18	0.75	0.75	0.667

Table 3 – validation of the model trained on the original HDR data, on the original test set (V1) and the additional test set (V2).

3.2.3 A larger sample size enables a larger feature size

To train CUNE, I inspected features on a case-by-case basis, i.e. first sgRNA and then ssODN. I trained models on these features to identify optimal models using the average out-of-bag (OOB) error using cross-validation. I included the sgRNA based on previous observations of it to modulate Cas9 activity (Wilson, Reti, et al., 2018). This was based on the hypothesis that while efficiently inducing SNVs requires control over the repair-pathway, it may still be driven by Cas9 activity. This proved to be the case with an sgRNA model trained on sgRNA global nucleotide composition resulting in the lowest cross validated OOB error of 0.250.

The global nucleotide composition is a general representation for the nucleotide composition of a DNA or RNA sequence. It is a count of nucleotides (i.e. Ts, Gs, etc.), and adjacent nucleotides (i.e. TTs, GTs, etc.) across an entire DNA or RNA sequence. Because of this, the global nucleotide composition is to some extent, position agnostic. An alternative representation of a nucleotide sequence is the positional nucleotide composition. This results in an array representing the presence or absence of every nucleotide, and adjacent

nucleotides, at every position. Although the positional nucleotide composition can represent sequences in more detail, training a model on the positional nucleotide composition, resulted in the accuracy decreasing from 0.733 to 0.4 (Table 4). I hypothesised that the decrease in performance was due to the curse of dimensionality, a statistical phenomenon that makes it increasingly difficult for ML algorithms to find signal in the data due to a high sparsity (number of zeroes) (E. M. Wright & Bellman, 1962).

Model	Zero columns	OOB error	Precision	Recall	Accuracy	Correct
Local (published)	79	0.575	0.333	0.4	0.4	11/30
Local (additional)	32	0.402	0.9	0.636	0.611	28/46

Table 4 – cross validated scores of models trained on the sgRNA local nucleotide composition. This is a representation of the sgRNA sequence, where each feature indicates the presence (1) or absence (0) or a particular nucleotide at a particular position. For example, if a sgRNA has an A at position 5, the feature "5_A" would be "1", "5_C" would be "0", "5_G" would be "0", and "5_T" would be "0". The same is repeated for adjacent nucleotides ("AA", "AC", "CA", etc.) There are 404 possible positional (di)nucleotides features in a 20nt sequence.

In support of my hypothesis, in the original training set, 325 possible nucleotide/dinucleotide combinations occurred one or more times. This meant that 79 (di)nucleotides out of a possible 404 were not represented in the training set. However, in the new training set, 372 positional nucleotide/dinucleotides occurred one or more times. This meant that only 32 (di)nucleotide combinations did not exist in the new training set. The only way to reduce this number is through a greater distribution of sgRNA targets. In further support of the small sample size resulting in a poor positional nucleotide performance, a model trained on the new training set improved over the positional nucleotide model trained on the published data (Table 4). The accuracy improved from 0.4 to 0.611. However, despite this improvement, the model was still outperformed by the published global nucleotide model. With there still being 32 unrepresented (di)nucleotides, a larger sample size would be required to effectively model local nucleotide composition.

3.2.4 SNV-to-PAM distance is an important feature

The distance between the cleavage-site and the desired SNV has been demonstrated to have inverse relationship to HDR efficiency (Inui et al., 2014; K. Wang et al., 2016). That is, the smaller the distance, the more likely the SNV would be integrated. Because of this, I hypothesised that by including this distance in training, it would result in an improved model. However, upon training a model on the original training set with distance as a feature, I observed no improvement in model metrics. I hypothesised that this was because the dataset was designed to capture a wide range of features that influence HDR efficiency (Miyaoka et al., 2016). And relative to other features, distance was a weak modulator of efficiency as editing may inherently fail at certain loci, regardless of distance. This was supported by the

distance being an unimportant feature in the feature importance list, relative to sgRNA nucleotides. The feature importance list is a property of Random Forests that ranks features by their "Gini impurity". This is a metric on how well a feature can divide data on a feature into its correctly labelled groups. On the additional dataset, I once again observed the distance to have no influence on model metrics. However, of difference was the feature importance, where the distance ranked third in the model trained on the additional dataset. The high ranking, but lack of model improvements, provides support that distance does modulate efficiency, but with its power in this dataset being outweighed by the sgRNA sequence.

3.2.5 Global nucleotide composition is sensitive to noise

Models trained on the ssODN global nucleotide composition on the original dataset resulted in poor performance, with the lowest OOB error of 0.6. I had hypothesised that a model trained on the ssODN would be able to accurately differentiate between high- and low-efficiency targets, due to the key role of the ssODN in HDR. But this appeared to not be true. However, in this dataset, ssODNs are on average 159 nucleotides in length, compared to the 20-nucleotide long sgRNAs. And the global nucleotide composition, which is a summary of an entire sequence, is only relevant if most of the sequence being modelled is relevant. For the ssODN, given its length, this may not be the case. For example, groups have investigated the influence of ssODN symmetry and arm length on HDR efficiency, drawing the conclusion that asymmetric ssODNs can improve HDR efficiency (Liang et al., 2017; Richardson et al., 2016). The consensus was that shorter 3′ arms and longer 5′ arms were optimal for efficient HDR. Furthermore, based on the kinetics of HDR, there may be other differences in arm importance. For example, after cleavage, 5' to 3' resecting occurs at the cleavage site, resulting in a 3' overhang. This means that the 3' arm (homologous to the PAM/non-target strand) of the ssODN is the first region to interact with the target DNA. Based on this information, I hypothesised the ssODN influence on HDR efficiency to be asymmetrical, rather than constant across the entire ssODN. To test my hypothesis, I trained models on ssODN arms separately, rather than the entire ssODN. This resulted in models where the 3′ arm does inform HDR efficiency, with an OOB error of 0.275, and the 5′ arm does not, with an OOB error of 0.792 (Table 5).

Model	Region	OOB error	ROC	Precision	Recall	Correct
O1	Full	0.6	0.54	0.413	0.533	13/30
O2	3′	0.275	0.91	0.803	0.733	22/30
O3	5′	0.792	0.09	0.25	0.267	8/30

Table 5 – metrics from three Random Forest models trained on the nucleotide composition of the ssODN. O1 is trained the full ssODN. O2 is trained on the 3′ arm, and O3 is trained on the 5′ arm (all homologous to the PAM strand).

3.2.6 Using machine learning to learn from the data

Previously, I trained a model on the two feature sets that presented high performance. This included the global nucleotide composition of the sgRNA, and the global nucleotide composition of the 3' ssODN arm. This was named the M1 (mixed) model and served as the production model for CUNE. On the V1 dataset, it presented a prediction accuracy of 0.733. With the newer and larger dataset, I aimed to train a better performing model. But, instead of manually training models on features that I hypothesised to modulate HDR efficiency, I took advantage of Random Forest's ability to cope with high-dimensional data to train a model on the entire feature set. From this, the model with the optimal cross-validated score validated on the validation set, V2, with an accuracy of 0.833. From 11 positives, it correctly classified 9, and from 7 negatives, it correctly classified 6. This is an improvement on the original model, which presented an accuracy of 0.667 on V2.

From this one model, it was also possible to identify the most-influential features. For example, in the top 100 features, 33 were 3' ssODN features, 29 were sgRNA features, 19 were 5' ssODN features and 18 were overall ssODN features. Furthermore, while 16 3' ssODN positional nucleotides appeared in the top 100, zero 5' ssODN positional nucleotides appeared. This supported my hypothesis that the 3' ssODN arm is more influential in modulating HDR efficiency than the 5' arm.

3.2.7 Web service for predicting HDR efficiency

Based on the original model, I created an online prediction tool: Computational Universal Nucleotide Editor (CUNE). CUNE enables researchers to identify the optimal way to insert a specific SNV at a genomic locus. Because base editing is generally more efficient than HDR, the service identifies which, if any, base editing system is applicable, using pre-established rules (Gaudelli et al., 2017; Y. B. Kim et al., 2017; Komor et al., 2016; Nishida et al., 2016; Renaud et al., 2016). However, because of the limited scope of base editing, CUNE will identify the optimal sgRNA to induce a given SNV. Based on this work, I will update CUNE to the new model trained on the larger dataset.

3.3 Discussion

I set out to understand the factors that govern HDR-mediated SNVs. I aimed to create a computational tool to make efficiency-improving recommendations for variables that are easy for the researcher to vary, such as ssODN and sgRNA design. This was especially relevant, as the currently known factors that govern efficiency, such as cell type and locus (Miyaoka et al., 2016), are usually fixed parameters for an experiment.

I trained models on different features to investigate how they each influence HDR efficiency. I chose to use the Random Forest algorithm as it enabled the quantification of the contribution of each input feature (feature importance), as well as the modelling of feature interactions, which provided insights into mechanisms. Random Forests are also resilient to overfitting (Breiman, 2001b), which was crucial for this training set as it contained more features than samples.

I hypothesised that the ssODN nucleotide composition would be an influencing factor on HDR efficiency, due to Watson-Crick base pairing between the ssODN and the DNA target being essential for inducing HDR-mediated SNVs. While the nucleotide content of the ssODN 5′ arm was unimportant (O3), the content of the 3′ arm proved to be a major contributor to prediction accuracy (O2). The importance of the 3′ region was in agreement with the mechanism of HDR. For a cell to proceed with HDR, the 5′ strands at the DSB are degraded (Pâques & Haber, 1999). This process, known as 5′ → 3′ resection, results in 3′ overhangs at the DSB (Figure 7). Therefore, the 3′ region of the ssODN, being complementary with one of the newly formed 3′ overhangs, is the first region of the ssODN to interact with and bind to the target. I propose that if this occurs, HDR will continue regardless of the 5′ sequence, which resulted in the poor predictive performance of the 5′ ssODN models.

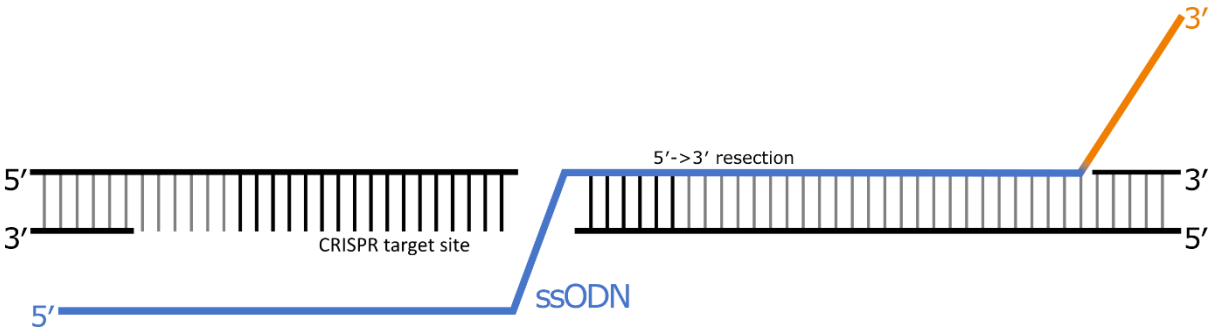


Figure 7 – an ssODN (blue/orange) annealed to 5′-3′ resected DNA (PAM strand). ssODNs with regions extending beyond the resected DNA may require further processing or strand-invasion of the DNA target. The sequence composition of this region (orange) has a strong impact on HDR-efficiency.

Liang *et al.* observed the optimal length for a 3′ arm to be 30-35 nucleotides, which they based on the 5′ → 3′ resection at the DNA target typically creating overhangs of 30 to 40 nucleotides (Liang et al., 2017). They suggested that arms extending beyond this region are accommodated by further target resecting, 3′ ssODN trimming or strand invasion of the target, while shorter arms can anneal directly to the target. I hypothesised that the efficiency of this process is influenced by nucleotide composition, which I could investigate as the ssODN arms extended beyond the resected region (Figure 7). In support of the optimal length, I observed my prediction accuracy to temporarily plateau at 20 nucleotides, before continuing to improve

at 45 nucleotides, all the way to 60 nucleotides. This indicated HDR-efficiency is especially sensitive to the nucleotide composition of the region beyond the resected DNA (Figure 7).

I hypothesised that the distance from the SNV to the PAM would contribute to my model's accuracy, but this proved not to be the case. While I observed the expected inverse correlation between distance and HDR efficiency, as reported in previous literature (Bialk et al., 2015; Inui et al., 2014; Liang et al., 2017), it did not improve model prediction accuracy. This was likely a result of the unbalanced nature of this feature in the dataset. For example, Liang *et al.* observed HDR rates of below 5% at distances over eight nucleotides away from the PAM sequence and rates of 10% to 30% at, or fewer than, eight nucleotides away. Setting eight nucleotides as the high/low threshold, a balanced dataset would require 50% of the samples to be up to (and including) eight nucleotides away, with 50% of the samples being over eight nucleotides away. However, only 13 out of 63 (21%) of the samples were over eight nucleotides away, limiting the impact of this feature in modelling. This is a result from bias in experimental design, as sgRNAs were designed based on their proximity to the target SNV. The aim of this was to improve experimental outcomes, rather than to produce a balanced dataset. Another potential reason is the high variance in HDR efficiencies in each of these two windows. From inspecting samples with an SNV over eight nucleotides away from the PAM, the average HDR efficiency was 0.24, but the standard deviation was 0.20. And for samples with an SNV at eight or fewer nucleotides away from the PAM, the average HDR efficiency was 0.40 with a standard deviation of 0.29. Although the averages are as expected, demonstrating higher HDR efficiencies for samples with an SNV near the PAM, the high standard deviations contributed to the poor predictive power of this variable.

This work resulted in the first computational method for designing efficient experiments for inducing SNVs using base editing and HDR. I have provided this as a web service, which can design sgRNAs and ssODNs to induce user-specified SNVs. In addition, the web service will also identify base editing targets using pre-existing rules. Also, with the availability of additional data, I was able to validate the published model on a second validation set. Although the published model classified most additional targets correctly, the accuracy value was lower than when validated on the original validation set (0.677 vs 0.733). This was likely due to an increase in the efficiency of experiments over time.

Although it was possible to train an accurate model from the small dataset, further investigation into features that modulate HDR efficiency would require a bespoke dataset. I started the experimentation required to create such a dataset, designing ssODNs and sgRNAs for 159 target sites. These components were synthesised for targeting in ES cells by Agilent (15K

oligonucleotide array). However, the experimental work is still ongoing. Experimental parameters for the ssODN and sgRNA design and their rationale included:

- Eight different ssODNs per sgRNA to better quantify the independent influence of these two components on HDR efficiency.
- Systematically designed SNV to PAM distances to enable this feature to be modelled.
- Targets in different epigenetic states, to identify and model the influence of features like methylation on HDR efficiency.
- ssODNs with changes other than SNVs, such as insertions or deletions, to enable modelling different HDR outcomes.

As well as enabling insights into efficiency modulators, the larger dataset of over 150 sgRNAs and 1,000 ssODNs would increase model performance by providing a more diverse set of samples to model.

Supporting the importance of more data, the model trained on the published dataset and the published validation data, and subsequently validated on the additional data, outperformed the original model. Although this model was trained on just published data, including the published validation set originally would have removed the ability to validate the published model on a validation set. The accuracy of the new model was 0.833 compared to the published 0.733.

Chapter 4 – The influence of CRISPR-Cas9 induced HDR on generating conditional knockout alleles using a 2-guide 2-oligonucleotide donor approach

CRISPR-Cas9 enables the creation of animal models, which in turn supports functional genomic approaches to better understand human disease. However, inducing knockouts using CRISPR-Cas9 can result in embryonic lethality. In this chapter I explored the application of using CRISPR-Cas9 to flank a region with two *loxP* alleles using two sgRNAs and two single-stranded oligodeoxynucleotides. This technique, herein referred to as two-donor floxing, enables conditional knockouts based on development stage or location.

4.1 Introduction

The field of functional genomics is enabled by the availability of whole-genome sequencing data (Hieter & Boguski, 1997; Lander, 1996). The broad aim of functional genomics is to elucidate gene function on a genome-wide scale. Techniques range from comparative homology searches (Dehal et al., 2009) to reverse genetics (Bhadauria et al., 2009). Where the former relies on the availability of already-annotated homologous genes, the latter relies on the control of a gene, with the resulting phenotype being observed and annotated. Gene control can be transient, for example by epigenetically controlling gene expression with RNA interference, or permanent and even heritable, by through gene targeting and mutagenesis (Alonso & Ecker, 2006; Gilchrist & Haughn, 2010). Despite the method of gene control used, this enables the study of novel, unannotated genes.

In this chapter I used mouse data as mice are often used in comparative genomic studies. Comparative genomics involves the comparison of common features between two genomes (Hardison, 2003). Rats and mice are most frequently used to study genetic disease (Rosenthal & Brown, 2007; Simmons, 2008). Of these rodents, the mouse was the first to have its genome sequenced, with the Human Genome Project including the mouse as one of its five key model organisms (Waterston et al., 2002). And with a high level of similarity between human and mouse genes (99% of mouse genes have a homologue in the human genome), paired with its small size and cost efficiency, the mouse is one of the more suitable species for performing comparative genomic studies (Vandamme, 2014; Waterston et al., 2002).

When the aim is to understand genetic disease, rather than healthy individuals, appropriate mouse models must be used. Ideally, such mouse models would contain similar genetic perturbations as their human counterparts, with database tools assisting in identifying target

genes based on homology or ontology (Hyung et al., 2019; C. L. Smith et al., 2018). After the identification of a candidate target, the target must be altered as desired. Changes can be generated in mouse embryos through spontaneous, radiation, or chemically induced mutagenesis (Hardouin & Nagy, 2000; Justice et al., 2011). However, this non-targeted approach is dependent on large scale screens, selective breeding, and chance. Furthermore, certain changes may not be possible. Instead of relying on random mutagenesis, it is also possible to create transgenic mice with inserted DNA. The first example of this, from 1976, utilised viral integration (Jaenisch, 1976). However, despite the predefined genetic payload, the target was non-specific. That is, instead of the retrovirus having just one potential integration site, it had two. A method that enabled creating more precise transgenic mice involves modified embryonic stem (ES) cells (Capecchi, 1989; Gossler et al., 1986). Instead of editing the genetic material of the embryo directly, modified ES cells are injected into the blastocyst, resulting in a mouse with the desired mutation. However, this leads to the resulting mouse being mosaic for the original genotype and that from the ES cells. Of course, now precise changes can be induced using guided nucleases such as ZFNs, TALENs, and CRISPR.

Although such technologies enable the efficient and precise editing of genes, a potential consequence of generating mice with germline mutations is genetic lethality (Bedell et al., 1997). However, a concept that allows researchers to potentially overcome genetic lethality are conditional knockouts (Sauer, 1998). Conditional knockouts enable a researcher to designate a change to be either temporal (at a specified time or development stage), spatial (in a specified cell type), or a combination of the two (Schwenk et al., 1998). Enabling conditional knockouts is a technique based on Cre-*loxP* recombination (H. Gu et al., 1994; Tsien et al., 1996). As a site-specific recombination system, the presence of Cre recombinase catalyses recombination between two *loxP* sites (N. Sternberg & Hamilton, 1981). Therefore, in mice with a genetic region flanked by *loxP* alleles, the expression of Cre will result in the deletion of this flanked region. And until Cre is expressed, the flanked region will remain unperturbed (Figure 8).

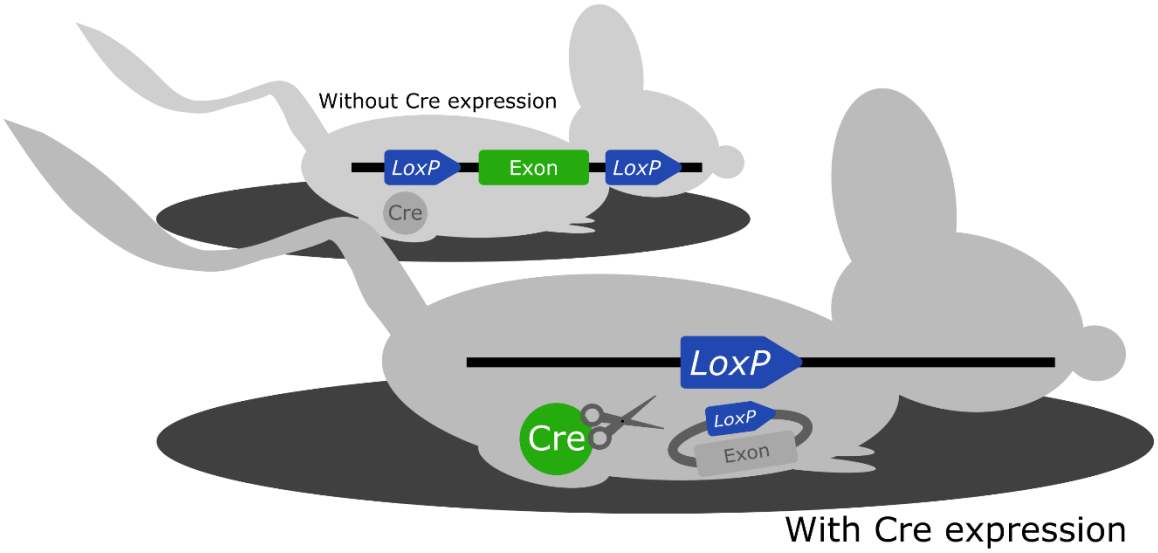


Figure 8 – the rear mouse represents a floxed (loxP flanked) exon in the genome. Without Cre recombinase expression, the gene containing the floxed exon can be expressed per usual. The foreground mouse represents one with Cre recombinase expression. This leads to excision of the flanked exon through recombination. Expression may be tissue specific or specific to a particular stage of development, thus avoiding embryonic lethality.

To use Cre-*loxP* for conditional gene knockouts, a gene must be floxed, i.e. the gene or a critical exon flanked by *loxP* sites. The process of introducing *loxP* sites is known as floxing (*loxP* flanking). Floxing has been demonstrated with the previously methods including ES cells (H. Gu et al., 1993; Hadjantonakis et al., 2008; O'Gorman et al., 1997), and more recently using targeted nucleases like ZFNs (Brown et al., 2013) and CRISPR (H. Yang et al., 2013). The latter targeted nuclease methods, now the gold standard, are known as two-donor floxing (H. Yang et al., 2013). This technique requires an ssODN for each of the cleavage sites induced by engineered nucleases, where each donor contains a copy of the *loxP* sequence, flanked by the target sequence. This enables the *loxP* sequences to be integrated into each of the target sites via homology directed repair (HDR). Because there are two targets, this process can be performed either sequentially or simultaneously. However, results from sequential injection have been mixed, ranging from a higher efficiency than simultaneous introduction (Horii et al., 2017) to almost inevitable failure at the secondary introduction (Gurumurthy et al., 2019).

But regardless of the engineered nuclease and the technique used, two-donor floxing ultimately relies on the inefficient HDR mechanism to insert *loxP* alleles into targets. And because a floxed allele requires the insertion of two *loxP* sites, two separate HDR events are required for two-donor floxing, further reducing the likelihood of success. For an example using ZFNs, out of 80 live-born rats, while 48 (60%) had a single *loxP* insertion, only seven (8.75%) had both insertions (Brown et al., 2013). Higher efficiencies have been observed using CRISPR-Cas9, with a two-donor floxing efficiency of 16% (H. Yang et al., 2013), however, this

may be optimistic for real-world two-donor floxing experiments. Because, in Chapter 3, I observed an average HDR efficiency of 39%. Based on the postulate that each occurrence of HDR is an independent event (the success of one HDR event won't influence the success of the other), the expected probability of two HDR events is P(39% and 39%), or 15.21%. However, as the two HDR events must be on the same allele, this probability is halved in diploid organism like mice and humans. This results in an expected probability of just 7.60%. Furthermore, this efficiency is for single-base payloads, i.e. a single nucleotide variant. Inserting longer payloads like *loxP*, which is 34 nucleotides (Hoess et al., 1982), has been demonstrated to be less efficient (K. Li et al., 2014; Liang et al., 2017).

Based on the requirement of two simultaneous HDR events and large payloads, I hypothesised that two-donor floxing is less efficient than expected based on currently achievable levels of HDR efficiency. To test this, I have curated and analysed results from two-donor floxing experiments. Each two-donor floxing experiment was designed to test the outcome from using CRISPR-Cas9 to perform two-donor floxing of an allele in mice embryos.

In total, there were experimental results for 54 unique genomic regions. The results from each experiment were generated by 19 laboratories across six countries, with each laboratory targeting at least one region. The six countries included Australia, Belgium, Canada, Japan, UK, and USA. Essentially, each experiment was a multiplexed CRISPR experiment with two different sgRNA/ssODN pairs, designed to target a region on either side of an exon. Each ssODN contained the *loxP* sequence, flanked by the target sequence, with the aim being to integrate the *loxP* sequence into the CRISPR-Cas9 target through homology-directed repair. Each microinjection experiment included a count of live-born mice and the resulting mutations at each of the two CRISPR-Cas9 target sites. Efficiency was defined by the number of successful floxing attempts divided by the number of live-born mice. A successful floxing attempt was defined by a *loxP* insertion at both CRISPR-Cas9 targets in *cis* (same allele). Mutations that were not counted as successes include single *loxP* insertions, simultaneous *loxP* insertions where either or both insertions are in *trans*, as well as indels and larger deletions.

With the data generated by different laboratories, each with their own methodology and protocols, other variables were present with the potential to influence experimental outcome. These included reagent concentrations, ssODN lengths, the distance between target sites and technician skill level. Each of these had previously been demonstrated to effect experimental outcomes. Firstly, ssODN concentrations at the cleavage site have been demonstrated to correlate with efficiency by increasing availability of the template (Ling et al., 2020; M. Ma et

al., 2017). Secondly, ssODN lengths have been demonstrated to modulate efficiency (B. Gu et al., 2018), and decrease undesired mutations (Yoshimi et al., 2016). Thirdly, larger distances between target sites in multiplexed CRISPR experiments have been demonstrated to increase cleavage efficiency (Xie et al., 2015). And finally, technician skill level has been demonstrated to effect experimental outcome due to the high skill level required by microinjection (Hogan et al., 1994; K. R. Smith, 2012). If not considered, these variables each had the potential to confound my analyses. Instead, I aimed to investigate them to identify their influence over efficiency to test for the generalisability of prediction models.

Although the results suggested that two-donor floxing is an inefficient technique, it is versatile. I therefore developed a method to minimise the number of failed attempts by forecasting how many attempts will be required for a successful outcome. I also found evidence supporting the generalisability of CRISPR prediction models.

4.2 Results

Two-donor floxing was unsuccessful at most targets, with only 12 out of 54 targets featuring at least one success. The average two-donor floxing efficiency was 2%. However, before testing the influence of *loxP* insertion efficiency on two-donor floxing efficiency, I analysed the data to identify any confounding variables.

4.2.1 Laboratory-specific confounding variable analysis

By virtue of being a multi-centre analysis, the data contained inter-laboratory variables that may confound results. Variables could be introduced by differences in experimental design or methodology between laboratories. One was the ssODN concentration. Despite having an interquartile range of 40 ng/μL across the dataset, the average interquartile range within each laboratory was just 1.29 ng/μL. The same phenomenon is observed with ssODN length, with an interquartile range of 25 nucleotides across the dataset, but an average interquartile range within each group of just 2.36 nucleotides. I used regression analyses to test for an influence of these variables on two-donor floxing efficiency. However, I did not identify any significant correlations (Table 6). The ssODN concentration had effectively zero correlation with efficiency. ssODN lengths presented some correlation with correlation coefficients of 0.15 for the 5’ and 3’ ssODN lengths, but these correlations were not significant (correlation coefficient, R = 0.15, *P* = 0.28). Of note are the identical values for 5’ and 3’ ssODN lengths. However, these are the same due to groups choosing similar lengths for the two CRISPR-Cas9 targets present at each allele.

	Correlation Coefficient (R)	*p*-value
5’ ssODN length	0.15	0.28
3’ ssODN length	0.15	0.28
ssODN concentration	-0.02	0.92

Table 6 – correlations between two-donor floxing efficiency and experimental parameters. The ssODN lengths present the highest correlation, although not at a significant level.

Based on these observations, ssODN length and ssODN concentration were not significantly modulating two-donor floxing efficiency. However, although these variables were provided by each laboratory, which enabled them to be tested for their influence over efficiency, it was also possible that unlabelled variables were modulating efficiency.

4.2.2 Unlabelled confounding variable analysis

Unlabelled variables are variations between samples that exist but are not included, i.e. labelled, in the process of experimental design or data collection. They may be excluded due to oversight, a lack of quantification, or simply because they were deemed to be irrelevant. One such example is technician skill level. Although not present in this dataset, technician skill level is a variable that could modulate experimental outcome. In particular with this dataset, a hypothesis is that the skill level of the technician modulates efficiency due to the high skill level required by the microinjection technique used in these experiments (Hogan et al., 1994; K. R. Smith, 2012).

Without skill level included as a variable, this hypothesis could not be tested directly. Instead, I tested it using a rank-based nonparametric test. This would identify any significant differences between distributions of efficiency results from each laboratory. Differences would support the hypothesis of skill level modulating efficiency, whereas no differences would refute it. This hypothesis was tested in the original publication, identifying no significant difference in two-donor floxing efficiency between any laboratories (Kruskal-Wallis rank-sum test, chi-squared = 22, $P = 0.16$) (Gurumurthy et al., 2019). However, with the low two-donor floxing efficiency resulting in a low effect size between groups, with most (42/54) efficiencies being zero, it was possible that any significance was undetected due to a lack of statistical power. I hypothesised that although skill level was not demonstrated to modulate two-donor floxing efficiency, that the influence of skill would be present in the more variable *loxP* insertion efficiency.

To test this, I performed the Kruskal-Wallis rank-sum test on *loxP* insertion efficiency. I calculated the *loxP* insertion efficiency for each target site from the number target sites with a *loxP* insertion divided by the total number of mice. Although there were differences present

between groups, with the experiments from the University of Adelaide having an average *loxP* insertion efficiency of over 40% (Figure 9), there were no significant differences between any one group and the rest (Kruskal-Wallis rank-sum test, chi-squared = 23, $P = 0.155$).

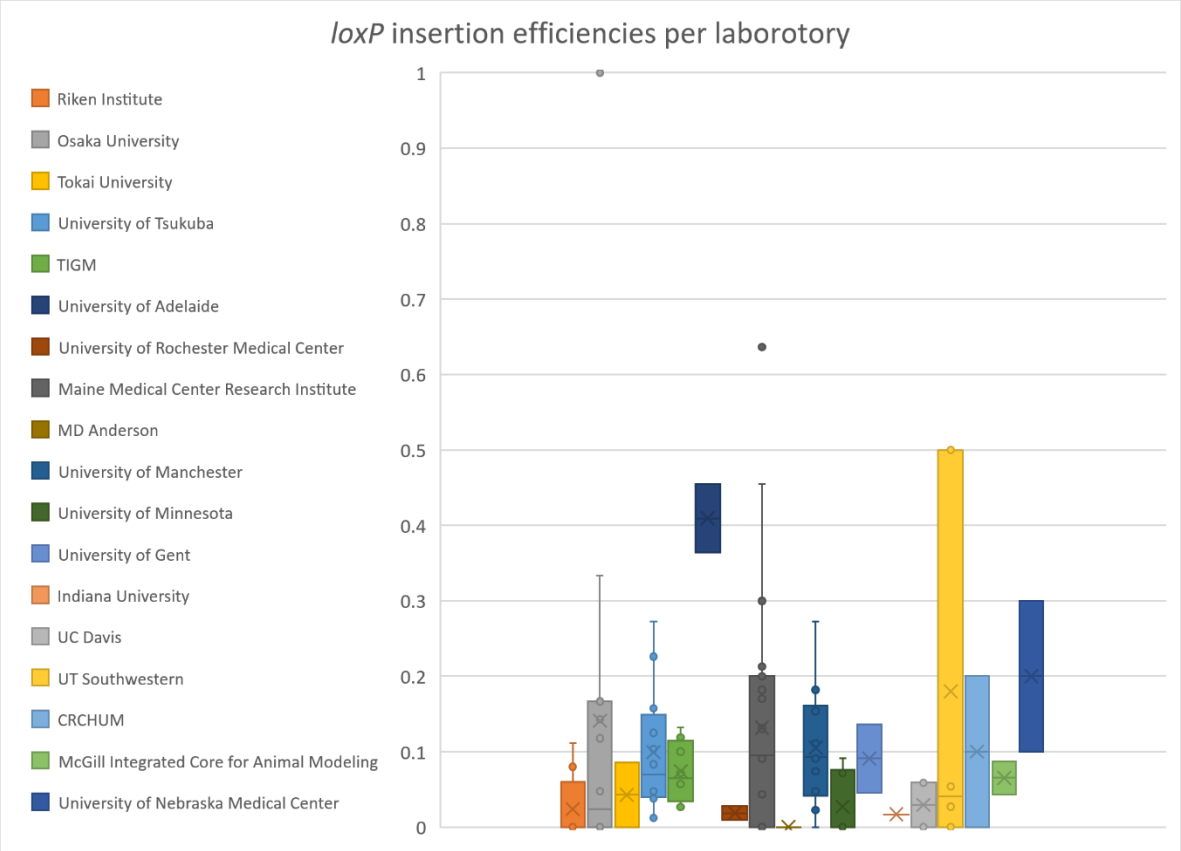


Figure 9 – distributions of loxP insertion efficiencies, across laboratories. Targets are unique, with each university targeting one or more exons in different genes. According to the Kruskal-Wallis rank-sum test, there was no significant difference between any one group and the rest. This rejects the hypothesis that any one group belongs to a different distribution, suggesting that laboratory-specific variables are not modulating efficiency.

The lack of significant differences between *loxP* insertion efficiencies from different groups provided support that differences in technician skill across these labs is not a modulator of efficiency. It also provided further support that ssODN length, concentration, or other laboratory-specific variables are not significantly modulating efficiency. Another observation is this data provides evidence for previous results to be overestimated. With a previous study observing a 16% efficiency for two successful events (H. Yang et al., 2013), here all but one group have an average efficiency for one successful event of below 20%.

4.2.3 Sample size effect on *loxP* insertion efficiency

For each of the 54 unique two-donor floxing targets, there were from 1 to 159 live-born mice (M = 31.28, SD = 32.87). The number of live-born mice is the sample size for each allele. It is an arbitrary value, decided upon by each laboratory. It could be based on a predefined goal;

but it could also be decided upon and depend on preliminary results. For example, an experiment could be stopped based on evidence for, or against, efficient two-donor floxing at an allele, or also based on a lack of any single *loxP* insertions. Despite its variable nature, the number of live-born mice was essential for calculating efficiency. For example, the number of mice with *cis loxP* sites divided by the total number of mice was used to calculate two-donor floxing efficiency. Because of its importance, I aimed to identify whether the number of live-born mice was biasing the calculated efficiency values and if so, to set a threshold to minimise the margin of error.

I hypothesised that at lower numbers of live-born mice that the calculated efficiency would not be representative of true efficiency. One reason was that fewer samples would result in a higher margin of error. When considering the entire distribution of *loxP* insertion efficiencies as the population (n = 108, M = 0.103, SD = 0.151), with 95% confidence the margin of error for *loxP* efficiency calculated from one live-born mouse was 0.296. The margin of error would decrease to 0.132 for five live born mice and to 0.094 for ten. However, while higher margins of error would lead to higher variances, efficiencies calculated from fewer samples were not significantly different from efficiencies calculated from more samples (Kruskal-Wallis rank-sum test, chi-squared = 8.9, $P = 0.628$) (Figure 10a).

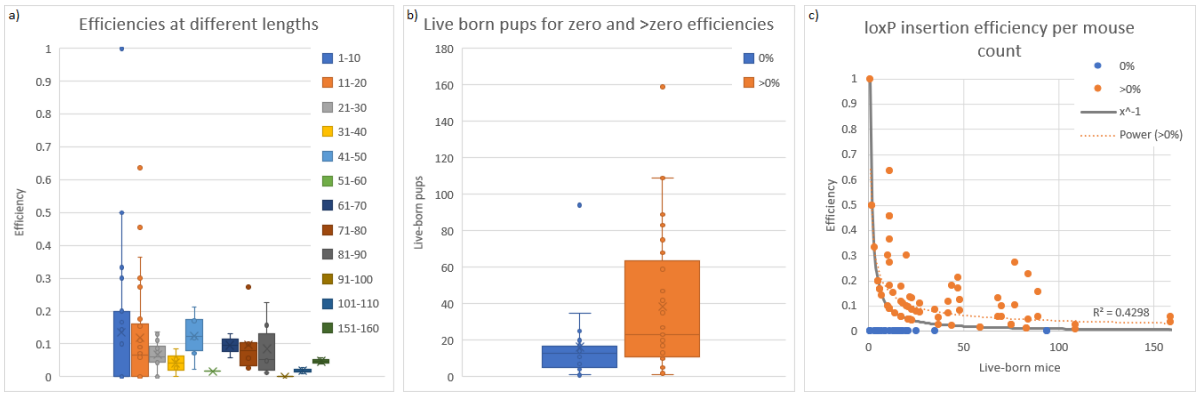


Figure 10 – a) demonstrates efficiency distributions for different numbers of live-born mice, in bins of ten mice. b) demonstrates the two significantly different distributions ($P < 0.05$). The range of 0% efficiency targets covers from 1 to 35 mice, with one outlier. The range of above zero targets includes the same minimum but extends up to 109 mice, with one outlier. c) presents these two distributions with a power function fitted against the above zero efficiency targets. The grey curve (x^{-1}) indicates the minimum non-zero efficiency for a target of a given sample size.

This result rejects the hypothesis that the efficiencies from different numbers of mice are different, despite the higher margin of error. However, there were several other observations that could be made about this data at low numbers of live-born mice. One observation was that *loxP* targets with 0% efficiencies were calculated from fewer live-born mice than *loxP* targets with efficiencies greater than 0% (Mann-Whitney U test, W = 350, $P = 7.744e-9$) (Figure 10b). This is likely due to the low effect size of *loxP* insertion. With a mean efficiency of 0.103,

ten mice on average would be required to observe one success, raising the efficiency above 0%. Another observation was that the shape of the efficiency distribution for low numbers of mice is bimodal, with the first mode at 0% and the second mode at a value above the mean efficiency value, depending on the number of live-born mice. I postulated that this bimodality was introduced by stopping experimentation early due to the observed result. As an analogy, the practise of stopping experimentation early is performed in randomised clinical trials (RCTs), with the aim of saving time and minimising cost. For RCTs stopped early due to efficacy, a review of 143 studies drew the conclusion that such trials should be viewed with scepticism due to implausibly large treatment effects (Montori et al., 2005). A later review suggested that while overestimation is present, it only becomes appreciable when true effect is close to zero (H. Wang et al., 2016). With the low effect size of inserting *loxP* alleles, this is likely what is observed in this dataset.

Another possible reason for the two modes was the minimum non-0% efficiency value for each number of live-born mice. This value is the minimum efficiency a *loxP* insertion can be, without being 0%. This is a function of sample size. For example, with one mouse, the efficiency could be 0% $(0/1)$ or 100% $(1/1)$, giving a minimum non-0% efficiency of 100%. No values between 0% and 100% were possible. With two mice, the efficiency could be 0% $(0/2)$, 50% $(1/2)$ or 100% $(2/2)$, giving a minimum non-0% efficiency of 50%. This followed the curve n^{-1}, where n is the sample size (Figure 10c). So, considering the 10% average efficiency of *loxP* insertions, for efficiencies calculated from one live-born mouse, nine out of ten alleles would be recorded as being 0% efficient. But one out of ten times the efficiency would be recorded as being 100%. Although the mean efficiency was still the expected 10%, the distribution would be bimodal with individual efficiency values being either underestimated or overestimated. To mitigate this propensity to underestimate or overestimate efficiency values, I excluded alleles with low numbers of live-born mice. A sample size of 20 resulted in a minimum non-0% efficiency value of 5%. This eliminated the bimodality that was present in the low live-born mice distribution. This sample size resulted in a confidence interval of 11.92%, with a 95% confidence. Although a lower confidence interval would have been preferable, an unrealistic number of live-born mice would have been required for each allele. For example, a confidence interval of 5% would have required 84 live-born mice per allele. This would have eliminated 49 out of 54 alleles resulting in a sample size of just five. Selecting the 20 mouse threshold resulted in a sample size of 26.

4.2.4 *LoxP* insertion efficiency modulates two-donor floxing efficiency

With no evidence for laboratory-specific confounding variables, and low live-born mice samples removed, I investigated the influence of *loxP* insertion efficiency on two-donor floxing

efficiency. With the 26 two-donor floxing targets, there were 52 different CRISPR-Cas9 target sites. Across these 52 target sites, the average *loxP* insertion efficiency was 7.5%. Although low, this was not unexpected with the average efficiency of inserting a single nucleotide variant being 39%, from Chapter 3, and the efficiency of inserting longer payloads like *loxP* being lower than shorter payloads. With each two-donor floxing target having a 5' and a 3' target site, there was no significant difference in efficiencies between 5' and 3' target sites, with average efficiencies of 8.38% and 6.80% respectively (Figure 11).

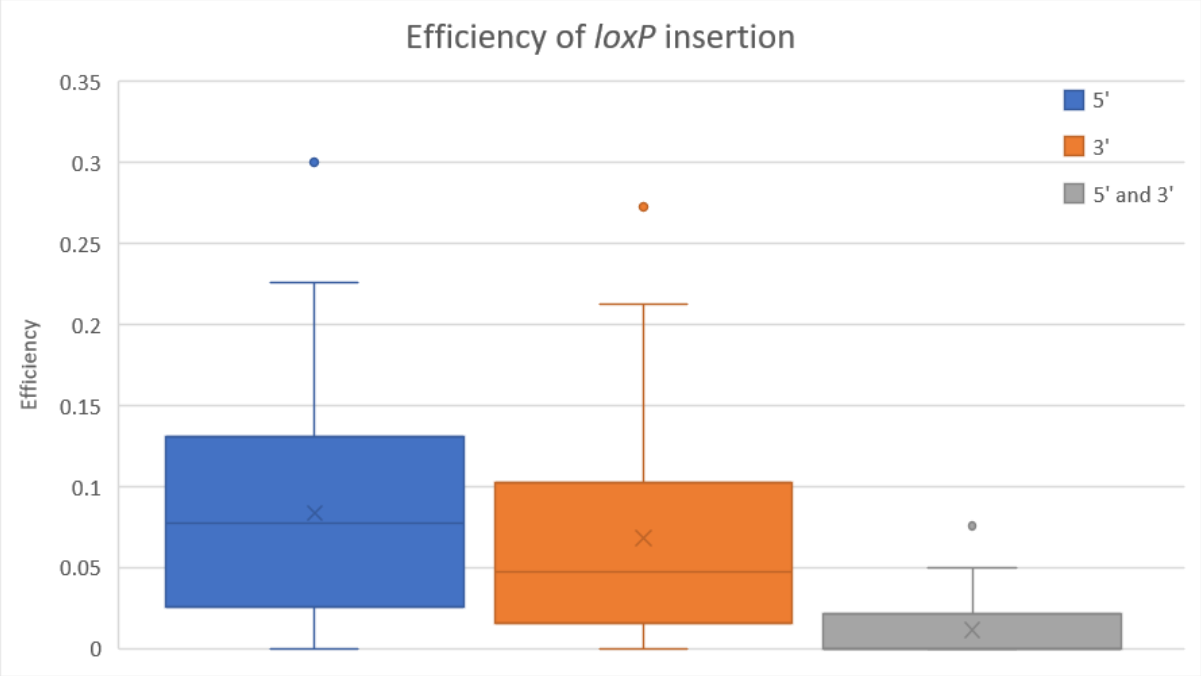


Figure 11 – box and whisker plots showing successful loxP insertion rates for each of the 26 loci. Each loci includes a 5' and a 3' target. The "5' and 3'" plot represents two-donor floxing efficiency. This plot demonstrates the low efficiency of two-donor floxing relative to single loxP insertions.

Reported in the original publication, *loxP* insertion efficiency mostly depends on *loxP* insertion efficiency at both the 5' and 3' target, with this variable explaining 80% of the variance from a regression analysis (Gurumurthy et al., 2019). Their observation supported the efficiency of two-donor floxing being a probabilistic event of two *loxP* insertion successes. To further support this hypothesis, *loxP* insertion efficiency would simply be a product of 5' and 3' *loxP* insertion efficiency. And with respective 5' and 3' efficiencies of 8.38% and 6.80%, the theoretical two-donor floxing efficiency was 0.56%. This is comparable to the observed average two-donor floxing efficiency of 0.90% (Figure 11). The increase in observed efficiency over theoretical efficiency is likely a result of the positive skew introduced by two-donor floxing efficiency having both its mode and minimum at zero.

4.2.5 The influence of simultaneous CRISPR targeting

Although these observations supported my hypothesis that two-donor floxing efficiency is primarily a product of the success and efficiency of two independent HDR events, it was possible that multiple simultaneous CRISPR events were modulating the efficiency of one or both *loxP* insertions (Xie et al., 2015).

Although there was a positive trend between 5' and 3' *loxP* insertion efficiencies for each two-donor floxing target, it was not significant and the effect size was small (coefficient of determination, $R^2 = 0.08$, $P = 0.161$). However, for cleavage efficiency, a regression analysis indicated that the cleavage efficiency of one target explained 20% of the variance of cleavage efficiency at the other target (coefficient of determination, $R^2 = 0.2048$, $P = 0.020$).

One possible reason for the correlation between simultaneous cleavage efficiencies is the proximal nature of targets. With one target being located close to another, they are more likely to share epigenetic properties than distal targets. This includes chromatin accessibility, which has been demonstrated to modulate CRISPR efficiency (Uusi-Mäkelä et al., 2018). To further test the hypothesis that proximal *loxP* insertions are not independent of each other, I investigated the influence of distance between targets on two-donor floxing efficiency.

4.2.6 Distance between targets

The distance between targets specifies how far apart the 5' and 3' CRISPR-Cas9 targets are for a given two-donor floxing target. The distance varies between two-donor floxing targets, as it is dependent on the length of the targeted region/gene/exon that is being floxed. Across the 54 two-donor floxing targets, the average distance between CRISPR-Cas9 targets was 8,122 bases (SD = 25,407). Targets are one or more exons, including up to nine exons. Across these targets, those that presented at least one successful two-donor floxing attempt had a higher average distance between CRISPR-Cas9 targets than those that presented zero successes (Figure 12a); however, this was not significant (Mann-Whitney U test, W = 211.5, $P = 0.271$). From inspecting just the positive distribution (targets with at least one two-donor floxing success), there was a significant correlation between two-donor floxing efficiency and distance (coefficient of determination, $R^2 = 0.828$, $P = 7.70e\text{-}4$) (Figure 12b).

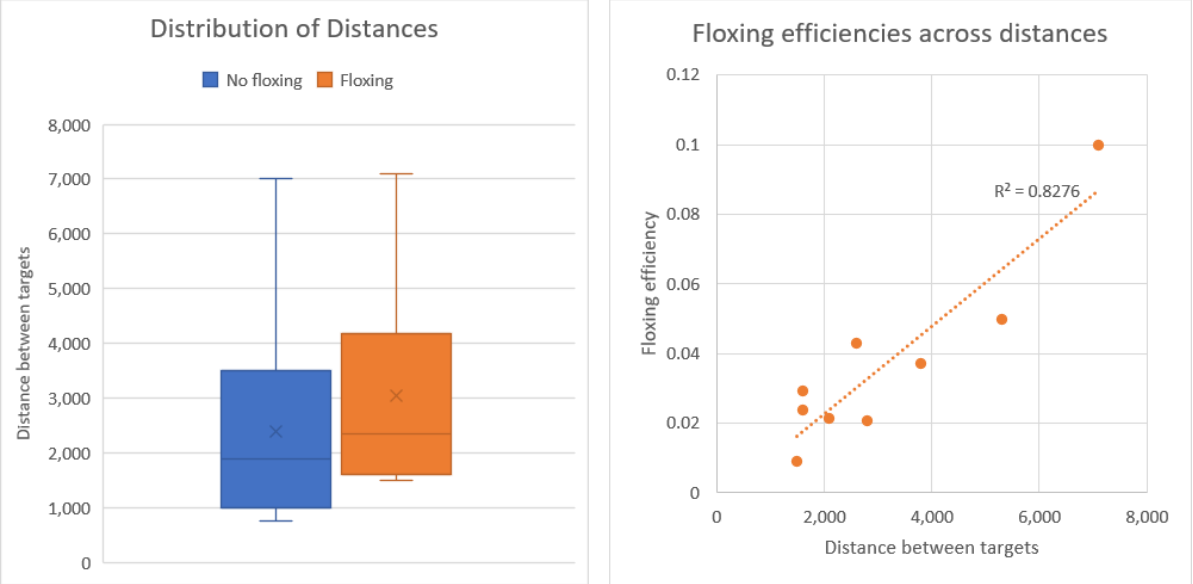


Figure 12 – a) the distribution of distances across two-donor floxing targets with zero successes and two-donor floxing targets with one or more successes. A success is a mouse with both loxP sites introduced. The average in the latter group was higher, although not significantly (P = 0.271). b) In the latter group (one or more successes), there was a significant positive correlation between distance and two-donor floxing efficiency with a coefficient of determination of 0.828 (P = 7.70e-4).

These results provided evidence that longer distances between targets results in a higher two-donor floxing efficiency, i.e. the efficiency of two simultaneous CRISPR-Cas9 cleavage events and HDR repair at both sites. A possible explanation for this is that successful cleavage of smaller regions is leading to deletion of the fragment between the CRISPR-Cas9 targets. For example, in previous multiplexed CRISPR-Cas9 experiments, a negative correlation had been demonstrated between distance and deletion efficiency (Xie et al., 2015). However, a more diverse set of samples would be required to investigate further. For example, it was not possible with this dataset to elucidate the influence of distance at distances outside those present in the dataset. Also, because distance is only significant at targets with at least one successful floxing attempt, and not significant across all targets, this indicates that targeting some alleles will fail regardless of distance. In other words, although distance may influence two-donor floxing efficiency, it is just one variable in the equation.

4.2.7 Machine learning

So far, my analyses had just considered the effect of single variables on *loxP* insertion efficiency and two-donor floxing efficiency. In this section I used machine learning to perform a multivariate analysis on the above properties. I also aimed to model the influence of nucleotide composition. For this task I used the Random Forests (Breiman, 2001b) algorithm to model the data. As another benefit, Random Forests can also model non-linear relationships.

With the low number of positives (two-donor floxing successes) and the low variance between positives, I aimed to train a binary classification model. To enable this task, I labelled samples with one or more two-donor floxing successes as true, and samples with zero two-donor floxing successes as false. Although the sample size of 26 was low, and the data was unbalanced with just eight positives (alleles with one or more two-donor floxing success), my aim was to model the data to identify features that modulate efficiency. I hypothesised that this would identify *loxP* insertion efficiency, based on previous evidence for its importance.

The set of base features is listed in Table 7. Other features that were trialled includes the nucleotide content of the sgRNAs and nucleotide content of the ssODNs. These were global counts, identical to Chapter 3. So, for example, a feature was created for the count of each individual nucleotide, as well as adjacent nucleotides. This was performed for each arm of the ssODN, and for the sgRNA. Inclusive of the comprehensive set of nucleotide features, this resulted in 131 features. Exclusive, it resulted in fifteen.

feature name	importance
3’ efficiency	0.1668
5’ ssODN 3’ flank length	0.1092
3’ ssODN 3’ flank length	0.0920
5’ efficiency	0.0827
5’ ssODN 5’ flank length	0.0719
3’ ssODN 5’ flank length	0.0718
Distance between targets	0.0690
ssODN concentration	0.0612
Live born mice	0.0564
5’ ssODN 5’ flank GC	0.0459
3’ ssODN 5’ flank GC	0.0424
3’ sgRNA GC	0.0396
5’ sgRNA GC	0.0369
5’ ssODN 3’ flank GC	0.0274
3’ ssODN 3’ flank GC	0.0268

Table 7 – feature importance values from a Random Forest model for predicting two-donor floxing efficiency. Importance values are a weighted measure, summing to one.

When including nucleotide features in modelling of two-donor floxing efficiency, performance was poor, with the bottleneck being the recall. With an average recall of 0.5 from five-fold

cross validation, positives were only being correctly identified 50% of the time. The rest of the time they were being misclassified as negative. Excluding nucleotide features from training resulted in an improved recall of 0.767. The improvement in recall from removing features highlighted the algorithm being unable to filter out noise due to the small number of samples. This was also observed in Chapter 3, and is likely a result of the curse of dimensionality, which is a result of a low sample size relative to the high number of features (E. M. Wright & Bellman, 1962).

Inspecting the importance of features used to train this model revealed similar insights to previous regression analyses. Features are presented in Table 7 alongside their importance values, which are weighted values, adding up to one. The feature with the highest importance is the efficiency of a *cis loxP* insertion at the 3' CRISPR-Cas9 target. 5' efficiency is also ranked highly but comes in at fourth. Second and third are ssODN length for 5' and 3' targets, respectively. However, of note is that for both ssODNs it was the 3' arm providing the most influence on the outcome. The 5' arms were fifth and sixth in the list, and contributed less information, in line with distance, ssODN concentration and live-born mice count. The GC content of the sgRNAs and ssODNs ranked the poorest, coming in at tenth to fifteenth.

These findings supported my hypothesis that *loxP* insertion efficiency is the primary modulator of two-donor floxing efficiency, with the 3' *loxP* insertion efficiency contributing 60% more information to the model than any other feature. Although the second and third features were the length of the 3' arm of each donor, donor lengths have been demonstrated to contribute to insertion efficiency (Shy et al., 2016). Finally, it is possible that nucleotide features would have provided more information, as in Chapter 3 and the models referenced in Chapter 1, however it is likely that that the small number of samples, and the large number of variables introduced too much noise for the algorithm to model these features.

Finally, I propose a method for forecasting the number of attempts required to achieve a success. Because despite its inefficiency, it is versatile. However, with an average *loxP* insertion efficiency of 10%, more than 100 attempts could be required to achieve just one simultaneous success. But with the average *loxP* insertion efficiency of 10%, and two target sites, a single *loxP* insertion should be observed for one in every five attempts. Therefore, the *loxP* insertion efficiency can be used to extrapolate the chance of a success. For example, if after ten attempts each target site has had zero successes, then with a 95% confidence the average number of attempts required will be 550. However, for one success at each site, this drops to 91, for two, 27 and for three, 11. Therefore if an allele has one or fewer successes at

both target sites after ten attempts, it should be reconsidered unless no alternative techniques or different alleles are available.

4.3 Limitations and improvements

These results again highlight the importance of large datasets. While statistical analyses uncovered robust evidence that *loxP* insertion efficiency is the primary modulator of two-donor floxing efficiency, Random Forest modelling failed to successfully model previously identified features like nucleotide composition. The small sample size led to a low coverage of possible nucleotide sequences, so this feature, while known to modulate insertion efficiency, was detrimental to model performance. Further exacerbating the issue of sample size, was the low number of live-born mice in many experiments. Due to their poor confidence interval, and tendency to under or overestimate efficiency, half of the 54 original samples had to be discarded.

Despite the limitations, the ability to produce results suggests the plausibility of generalisable CRISPR efficiency modelling. As even though this data was generated by 19 different laboratories, each with their own variations in methodology and experimental design, it was possible to gain insights and to a lesser extent, model. Despite the inefficiency of two-donor floxing, the inefficiency was consistent across laboratories, and was modulated by the same features. Combined with the inability to model known features, this supports my conclusion from Chapter 1; the importance of large scale experiments for training prediction models.

Chapter 5 – Generalisable Cas12a efficiency prediction

This chapter investigates the Cas9 and Cas12a CRISPR systems regarding efficiency and specificity. With a lack of computational tools for Cas12a, but with evidence of improved specificity over Cas9, I aimed to produce a quantitative comparison between the two systems to identify benefits of the latter. After which, I trained a Cas12a efficiency predictor to enable more effective use of this CRISPR system.

5.1 Introduction

Arguably the most widely used CRISPR system is CRISPR-Cas9, a class 2 type II system. As a class 2 CRISPR system, it is comprised of a single effector module that is capable of PAM recognition, R loop formation and target cleavage. This module, Cas9, is comprised of several functional domains, which each have specific functional roles in CRISPR interference. For example, to mediate double-strand break cleavage are the RuvC and HNH nuclease domains, which respectively nick the target and non-target strand of the DNA (Gasiunas et al., 2012). However, cleavage can only occur once activated, which requires the successful binding of CRISPR-Cas9 to its cognate target (S. H. Sternberg et al., 2014).

Recognition and binding in type II CRISPR systems are enabled by two RNA molecules: CRISPR RNA (crRNA) and trans-activating crRNA (tracrRNA). Together, these form a chimeric guide RNA (gRNA). The gRNA recognises its target through base complementarity, and through Watson-Crick base pairing forms an R-loop with the target strand, displacing the non-target strand. Successful binding results in a conformational change in Cas9, activating the nuclease domains and enabling cleavage (Jinek et al., 2014). Here, the effector can cleave both DNA strands simultaneously (Gong et al., 2018). But as well as being essential for target cleavage, R-loop formation also helps to minimise unintended cleavage at sites other than the target. It achieves this by being intolerant to mismatches (Rutkauskas et al., 2015). The level of intolerance is not binary (match/no-match), instead depending on factors like the number of mismatches and their location in the target (Jinek et al., 2012; X. H. Zhang et al., 2015). This means that while some mismatches will abrogate target cleavage, others will not. For example, mismatches closer to the PAM are more likely to abolish targeting than PAM-distal mismatches; possibly due to the kinetics of CIRPSR-Cas9 binding. This is because R-loops form in a “zipper effect”, propagating linearly from the PAM. So, while mismatches encountered early are likely to prevent complete R-loop formation, mismatches encountered later are more likely to allow for a stable product (Rutkauskas et al., 2015).

Unintended targets that are cleaved, also known as off-targets, can lead to cell lethality due to gene knockouts or loss of heterozygosity due to chromosomal translocations or rearrangements (Cho et al., 2014). Because of this, in the field of precision genome engineering, mismatch tolerance is usually undesirable. This has prompted groups to investigate the specificity of other Class 2 CRISPR systems, like the type V CRISPR system and its effector molecule Cas12a, to identify whether they offer a greater mismatch intolerance to Cas9 (D. Kim et al., 2016). Cas12a, formerly known as Cpf1, is a class 2 type V CRISPR system. Like Cas9, Cas12a is a class 2 CRISPR system (single effector molecule) that is RNA-guided and induces a DSB at the target site (Zetsche et al., 2015). But despite these similarities, both the mechanism and structure of Cas12a differ from Cas9.

One major difference is that while Cas9 has a cleavage domain for each strand of DNA (RuvC and HNH), Cas12a contains only one cleavage domain (RuvC). Cas12a instead uses its lone RuvC domain to cleave both strands of DNA, after successful R-loop formation (Swarts et al., 2017). Strands are therefore cleaved sequentially, with Cas12a first cleaving the non-target strand and then the target strand (Swarts & Jinek, 2019). Possibly due to the sequential cleavage events, mismatches may lead to variations in cleavage kinetics, rather than outright abrogation of cleavage (Swarts, 2019). This can include the nicking of just the non-target DNA strand (B. X. H. Fu et al., 2019; Strohkendl et al., 2018). Upon successful cleavage, Cas12a results in staggered cuts, unlike the blunt ends that result from Cas9 cleavage. This is likely due to the conformational change required to allow the single cleavage domain to access both strands. The type of cut is relevant because it may influence repair outcome (Bothmer et al., 2017). Another difference is in how each system processes the precursor crRNA (pre-crRNA). This step is required to form the mature crRNA that guides the Cas effector to a DNA target. However, while Cas9 requires the endogenous RNase III to be present in the host cell (Chylinski et al., 2013), Cas12a processes the pre-crRNA itself (Fonfara et al., 2016). And with Cas12a not requiring a trans-activating crRNA to perform this process, it results in Cas12a being an easier to use system than Cas9. Finally, Cas12a targets a different PAM to Cas9. Where Cas9 requires the GC-rich 5'-NGG-3' PAM sequence at the 3' end of the guide RNA, Cas12a requires the longer, AT-rich 5'-TTTN-3' PAM sequence at the 5' end of the guide RNA. This leads to a different landscape of genomic targets to Cas9.

Regarding specificity, comparisons suggest that Cas12a is either comparable to, or more-specific than Cas9 (Y. Kim et al., 2016; Kleinstiver, Tsai, et al., 2016). Conversely, it has been observed that Cas12a is less efficient than Cas9 (Alok et al., 2020; Bin Moon et al., 2018). Despite each system having advantages and disadvantages, it is possible to design guide RNAs using computational tools to optimise for specificity and efficiency. Such tools exist for

both Cas9 and Cas12a, however, the landscape of published tools varies greatly between these two effectors. Regarding efficiency prediction, over twenty published tools exist for Cas9, whereas only two exist for Cas12a (H. K. Kim et al., 2018; Zhu et al., 2019). Some Cas9 efficiency prediction tools, like sgRNA designer (Doench et al., 2014) and sgRNA Scorer 2.0 (Chari et al., 2017), have been retrofitted with Cas12a support, however, this work is unpublished and therefore the data not shared. As well as efficiency prediction, it is also possible to predict the mutation that will result from CRISPR mutagenesis at a given target. But again, the publication landscape favours Cas9. Where for Cas9 three tools exist (Allen et al., 2019; Leenay et al., 2019; Shen et al., 2018), for Cas12a, none exist. The landscape for off-target tools is more similar between the two systems. Yet many of these tools simply identify and rank targets by how unique they are in the genome and don't consider CRISPR kinetics (Bae et al., 2014; A. O'Brien & Bailey, 2014).

This information makes apparent the large gap between Cas9 and Cas12a prediction tools. This is likely not because of lack of desirability, but because of a lack of data. For example, Cas9 datasets exist with up to 40,000 gRNAs (Allen et al., 2019). No public Cas12a datasets exist to rival this magnitude. However, new datasets are emerging, such as the 15,000 Cas12a guides used to train DeepCpf1 (H. K. Kim et al., 2018). On the basis that Cas12a offers benefits over Cas9, more work in this area would prove beneficial. So, to support further work, I performed a comparison between the two systems.

I hypothesised that Cas12a provides benefits over Cas9, such as being inherently less susceptible to off-target effects. This is based on the observation that Cas12a kinetics can result in off-targets with mismatches having just one strand being nicked. Nicks can be repaired by the base excision repair (BER) pathway that leads to accurate repair (Dianov & Hübscher, 2013). Therefore, mismatched Cas12a targets may be less likely to exhibit off-target effects than mismatched Cas9 targets, even if successful R-loop formation does occur. Also, Cas12a has a longer PAM sequence. The PAM sequence is an essential part of CRISPR targeting, and a longer sequence should lead to lower genome-wide PAM density. It has been observed that a lower PAM-density leads to a lower off-target binding affinity (S. H. Sternberg et al., 2014).

I tested my hypothesis using computational tools to quantify the uniqueness of targets for each CRISPR system, across the genome. However, because CRISPR activity at off-targets ultimately depends on mismatch tolerance, and with not all mismatches abrogating cleavage, I further tested my hypothesis using experimental data from public datasets.

With Cas12a providing specificity benefits over Cas9, I aimed to fill the gap in prediction tools. I hypothesised that given the growing amount of public data; it will be possible to train a model using machine learning to predict Cas12a mutagenesis outcome.

5.2 Results

5.2.1 *In silico* Cas9 and Cas12a comparison

To quantify benefits in Cas12a over Cas9, I performed an *in silico* comparison between these two CRISPR systems. Because my aim was to identify real world benefits, I selected ten human genes from the top ten most studied genes by citation (Dolgin, 2017). I postulated that selecting highly studied genes would provide more relevance than selecting random genes. This sample covered six chromosomes and features diversity across exon count, exon length and GC content, albeit with a GC rich bias (Table 8).

	Chromosome	Exons	AVE. Exon length	Examined bases	GC content
TP53	17	10	116.2	1362	54.4%
TNF	6	4	173.5	774	59.7%
EGFR	7	28	127.8	4137	53.8%
VEGFA	6	6	80.0	600	50.2%
APOE	19	3	316.3	1009	64.6%
IL6	7	5	125.8	729	48.9%
TGFB1	19	7	165.6	1299	60.5%
MTHFR	1	11	177.2	2169	56.9%
ESR1	6	8	221.5	1932	54.%
AKT1	14	13	109.0	1677	59.9%

Table 8 – from the top ten list of referenced human genes, these are the genes used for the in silico comparison of Cas9 and Cas12a. The examined bases column indicates the number of bases searched for guides. This includes coding regions from the exons, plus ten bases either side to allow for guides that are only partly in exons.

The aim of my analysis was to quantify the distribution of targets for the two CRISPR systems, and the distribution of potential off-targets for these targets. I identified targets by the presence of a PAM, and quantified potential off-targets using GT-Scan (A. O'Brien & Bailey, 2014) and Cas-OFFinder (Bae et al., 2014). These tools identify potential off-targets according to sequence similarity. This was on the basis that the Cas9/Cas12a-gRNA ribonucleoprotein complex is intolerant to mismatches, so potential off-targets are sites in the genome that have an identical sequence to a target. My analysis also included sites with one to five mismatches, as up to five mismatch off-targets have been demonstrated for both CRISPR systems (Y. Fu et al., 2013).

5.2.1.1 Target distribution

One observation from identifying targets in these ten genes was the order of magnitude fewer Cas12a targets than Cas9 targets. For Cas9, 16.1% of positions are targetable. For Cas12a, just 1.9% of positions are targetable (Figure 13a). This equates to a total of 2,506 Cas9 targets and 318 Cas12a targets. Although this means there are fewer targetable regions for Cas12a than Cas9 within these ten human genes, every gene had at least three Cas12a targets. However, it also implies that there are fewer Cas12a targets across the human genome. This would potentially result in the benefit of fewer Cas12a off-targets.

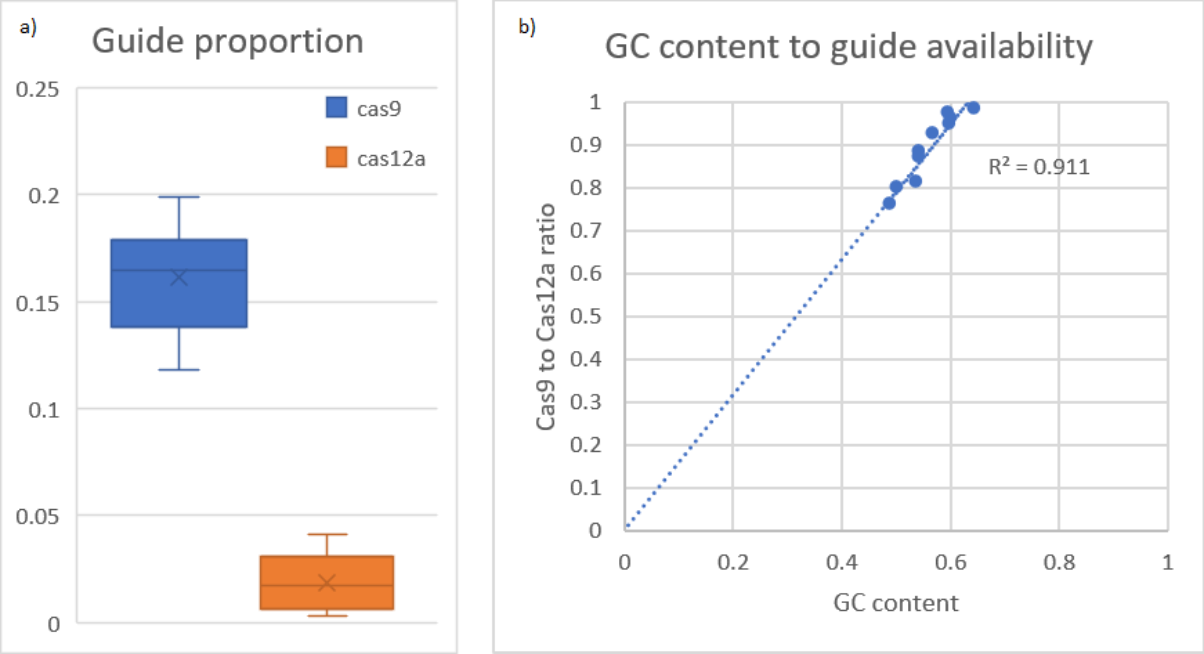


Figure 13 – the availability and ratio of Cas9 and Cas12a targets differs between genes. a) Cas9 targets are inherently more abundant than Cas12a targets across the inspected regions. For Cas9, an average of 16.1% of 23nt windows are targetable (have a PAM) but for Cas12a, an average of just 1.9% of 24nt windows are targetable. b) The ratio of Cas9 to Cas12a correlates with GC content. Regions with a higher GC content have a higher ratio of Cas9 targets (R^2 = 0.911, P = 1.777e-5). A linear model forecasts zero Cas9 targets to exist in regions with a zero GC content, as expected. However, this model forecasts Cas12a targets to reach zero at a 65% GC content. This indicates that although modulated by GC content, the lower abundance of Cas12a targets is likely due to a longer PAM sequence.

The disproportionate nature of Cas9 and Cas12a targets was not unexpected, as two-nucleotide motifs (GG) will occur more frequently by chance than three-nucleotide motifs (TTT). The extra nucleotide in the Cas12a PAM means that there will be four times fewer Cas12a PAMs than Cas9 PAMs when genomic nucleotide compositions are equal. However, because the two PAMs have different nucleotide compositions, genomic nucleotide composition, or nucleotide bias, can act as a confounding factor. Nucleotide bias, which is a preference towards A/T or C/G nucleotides, varies between organisms or even within individual genomes (Romiguier et al., 2010). This is evident by the varying GC levels in the ten human genes analysed (M = 56.29, SD = 4.69). With a positive GC bias in these ten genes, it was possible that Cas12a targets would become more abundant than Cas9 targets in less

GC-rich regions. To test this, I analysed the correlation between nucleotide bias and target ratio for each of the ten genes (Figure 13b). The results suggested that the ratio of Cas9 to Cas12 targets is correlated with GC content (coefficient of determination, R^2 = 0.911, *P* = 1.777e-5). However, based on a linear model, even with a GC content of 50%, Cas9 targets are more abundant than Cas12a targets with four Cas9 targets to one Cas12a target. The model further indicates that Cas12a targets will only outnumber Cas9 targets when the GC content drops below 30%. With the human GC content ranging from 30% to 65%, across 20kb windows (Waterston et al., 2002), these observations support my hypothesis that Cas12a targets will have fewer off-targets than Cas9 targets in the human genome. However, a disadvantage for Cas12a is that for GC-rich regions, above ~70%, Cas12a targets may become rare or non-existent.

5.2.1.2 Computationally identified potential off-targets

To further test my hypothesis that Cas12a off-targets are less abundant than Cas9 off-targets in the human genome, I performed a computational genome-wide analysis. This analysis was enabled by two computational tools, Cas-OFFinder and GT-Scan. Both tools test the uniqueness of CRISPR guides by identifying genome-wide matches (potential off-targets), based on sequence similarity. Potential off-targets can be identical, or with up to a predefined number of mismatches. However, while GT-Scan allows up to three mismatches, Cas-OFFinder does not impose a mismatch limit.

With off-target cleavage having previously been demonstrated to be possible (albeit inefficient) with up to five mismatches in human cells (Y. Fu et al., 2013), I aimed to use Cas-OFFinder to perform my analysis. However, to ensure Cas-OFFinder identified potential off-targets as expected, I compared it to GT-Scan with up to the latter's maximum of three mismatches. GT-Scan, which I developed for a previous research project, uses Bowtie (Langmead et al., 2009) to perform the potential off-target search. Across the ten genes, both tools identified identical sets of potential off-targets for 97.49% of Cas12a targets and 94.05% of Cas9 targets. For Cas12a targets, the only differences were at three mismatches. For Cas9 targets, most differences were at three mismatches, although 0.08% of targets differed at one mismatch and 0.4% of targets differed at two mismatches. Despite the few differing targets, there was no difference in the distribution of results between GT-Scan and Cas-OFFinder (Kolmogorov-Smirnov test, D = 6.38e-4, *P* = 1).

Having verified Cas-OFFinder against GT-Scan, I used Cas-OFFinder to compare Cas9 potential off-targets to Cas12a potential off-targets with up to five mismatches. For each number of mismatches, Cas12a targets had fewer potential off-targets for Cas12a. Regarding

zero-mismatch potential off-targets, every Cas12a target (100%) and most Cas9 targets (99.1%) were unique in the genome (Figure 14). Most targets remained unique in the genome with up to two mismatches, however, like with zero mismatches, more Cas12a (71.4%) targets were unique than Cas9 (50.6%) targets. For three mismatches, only 1.9% (Cas9) and 6.6% (Cas12a) of targets were unique, and for four and five mismatches, no targets were unique in the genome. As well as more Cas12a targets being unique in the genome than Cas9 targets, the number of potential off-targets varied significantly. For example, at four mismatches, Cas9 targets had twice as many potential off targets as Cas12a targets (Kolmogorov-Smirnov test, D+ = 0.33, *P* = 4.70e-27).

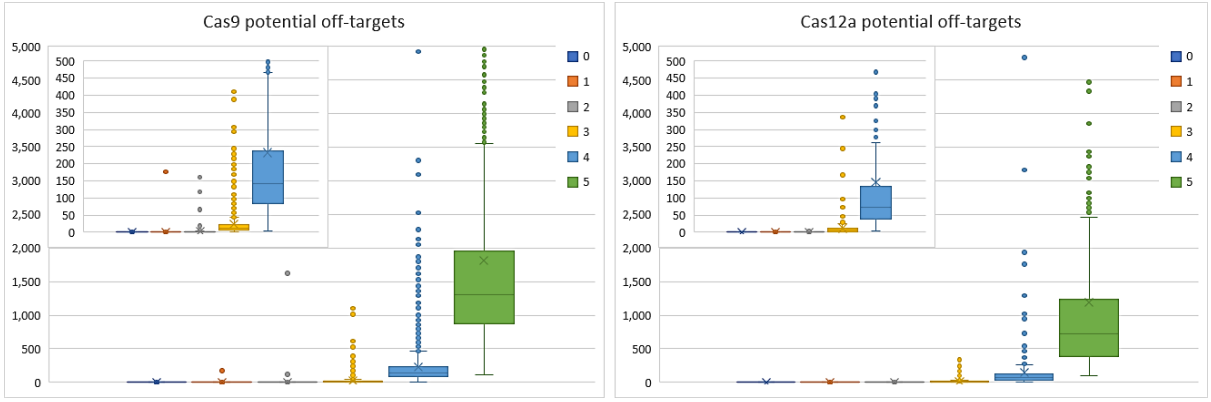


Figure 14 – the average number of sites elsewhere in the genome with zero to five mismatches to the original target in the 20nt guide sequence. Fewer sites exist for Cas12a targets than Cas9 targets for each of zero to five mismatches. For both Cas9 and Cas12a, the number of sites increases exponentially as the number of mismatches increase. The inner plots present a closer view of the distribution of zero to four mismatch sites.

These results demonstrated that Cas12a targets are more unique in the human genome than Cas9 targets, with zero or more mismatches. Because R-loop formation is intolerant to mismatches, this suggests that a randomly chosen Cas12a target will have fewer mismatches than a randomly chosen Cas9 target. Although computational tools like GT-Scan and Cas-OFFinder enable researchers to design guides systematically rather than randomly, no unique targets existed with four or more mismatches. However, with most Cas9 targets having at least 1,452 potential off-targets with four or more mismatches and most Cas12a targets having at least 792 potential off-targets with four or more mismatches, more-unique Cas12a targets can be chosen than Cas9 targets. Furthermore, with the low cleavage frequency of off-targets with this many mismatches (Y. Fu et al., 2013), most potential off-targets will be false positives. Although these findings support my hypothesis that Cas12a is more specific than Cas9, they do not take CRISPR kinetics and real-world cleavage efficiency into account. Therefore, I aimed to further test my hypothesis on *in vitro* experimental results.

5.2.2 *In vitro* Cas9 and Cas12a off-target comparison

To gather further support for my hypothesis, I analysed public *in vitro* experimental results. This meta-analysis involved two datasets; one with Cas9 data and the other with Cas12a data (Kleinstiver, Pattanayak, et al., 2016; Kleinstiver, Tsai, et al., 2016). Each dataset captured genome-wide off-targets for each target. This was achieved using a method called GUIDE-seq (Tsai et al., 2015). GUIDE-seq allows for the detection of DSBs, including those introduced by CRISPR cleavage. These are presented proportionally in terms of read counts. It is therefore possible to identify how off-target cleavage corresponds to target cleavage. One concern was the size of the datasets, of which the Cas9 dataset contained eight and the Cas12a dataset contained eighteen sgRNAs. However, these were the largest publicly available off-target datasets at the time of analysis. But although size may be a disadvantage, one advantage of these datasets was that they were generated by the same group, which can help to reduce confounding variables. Another advantage I identified was that targets shared the same distribution of potential off-targets as targets from the previously analysed genes.

5.2.2.1 Cas9

Across the eight Cas9 targets, off-targets existed with from one to five mismatches. Off-targets were cleaved less efficiently than targets, with cleavage efficiency decreasing as the number of mismatches increased (Figure 15a). Although cleavage efficiencies were lower, there were more off-targets across the genome with higher numbers of mismatches (Figure 15b). This was likely due to there being more potential off-targets (identified with Cas-OFFinder) with higher numbers of mismatches (Figure 15c). For example, although only 30% of potential off-targets with three mismatches were cleaved (Figure 15d), there were more potential off-targets with three mismatches than with two mismatches or one mismatch. The consequence of larger numbers of inefficient off-targets was higher combined off-target efficiency. For example, while the average cleavage efficiency of a three-mismatch off-target was less than 10%, the average cleavage efficiency of all three-mismatch off-targets, for a given target, was just under 40%.

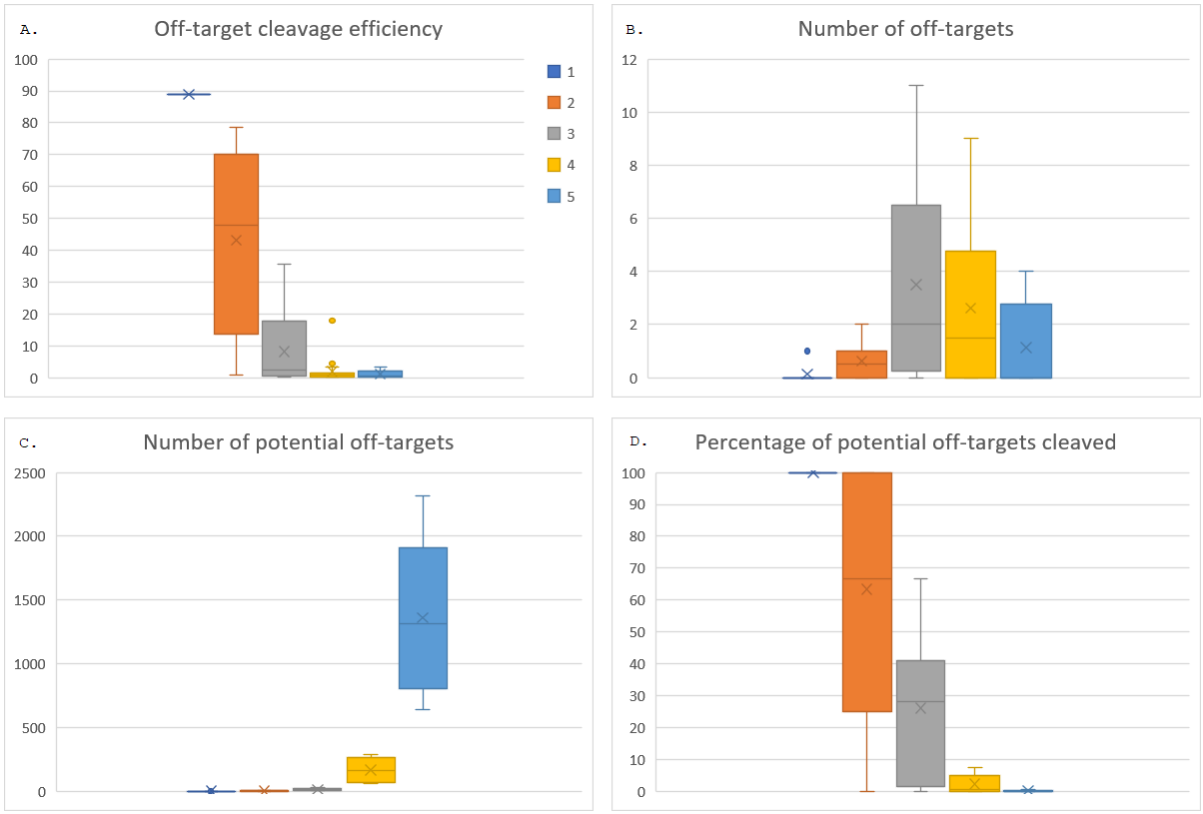


Figure 15 – each plot represents a different off-target distribution for each of one to five mismatches. a) Cas9 off-target cleavage efficiencies, relative to target efficiency. b) the number of off-target loci c) the number of potential off-targets, identified by CAS-OFFinder. d) The percentage of potential off-targets that were cleaved.

These observations demonstrated the importance of being aware of off-target effects that arise from CRISPR-Cas9 cleavage. They also demonstrate the variable nature of off-target cleavage. For example, off-targets with two mismatches presented cleavage efficiencies with an interquartile range of 12% to 70%. However, this is likely a result of the proximity of mismatches to the PAM, as based on previous observations of a "seed' region (Jinek et al., 2012; X. H. Zhang et al., 2015). This is supported by most two-mismatch off-targets with an efficiency of greater than 25% having both mutations at least nine bases from the PAM. No two-mismatch potential off-targets with both mismatches within eight bases of the PAM demonstrated cleavage. However, further highlighting the unpredictable nature of CRISPR-Cas9 cleavage is that the two-mismatch off-target with the second highest cleavage efficiency had a mismatch just two-bases from the PAM.

5.2.2.2 Cas12a

With Cas12a targets having fewer potential off-targets than Cas9 targets, I postulated that targets from the Cas12a dataset would present lower off-target cleavage than targets from the Cas9 dataset. This was indeed the case with nearly all targets. However, the proportion of Cas12a potential off-targets that were cleaved was lower than the proportion of Cas9 potential off-targets cleaved, with zero detectable off-target cleavage for most (16/18) Cas12a targets.

However, there were two exceptions. The first, DNMT1-3, had two off-targets and the second, DNMT1-7, had one. The former had a five-mismatch site with 0.3% cleavage efficiency, and a six-mismatch site with 149.9% cleavage efficiency. The latter just had a five-mismatch site with 1.6% cleavage efficiency.

The most unexpected off-target was the six-mismatch site. Not only was it more efficient than the intended target (1174 reads, compared to 783), but it was also the only off-target to have more than five mismatches. Perhaps expectedly, all six mismatches were outside the seed region, being PAM-distal and at the end of the target. However, other guides presented potential off-targets with mismatches isolated to this region, yet presented no cleavage. This indicates this off-target to be an outlier, albeit an efficient one. Despite it being an outlier, evidence suggested it was real, with validation in another cell line identifying the same six-mismatch off-target. However, in this validation set, cleavage efficiency was lower, at just 32%. No cleavage was detected at the five-mismatch site.

5.2.2.3 Cas9 vs. Cas12a

Based on these findings, Cas12a targets are more specific than Cas9 targets. While 7 out of 8 (87.5%) Cas9 targets presented off-target cleavage, only 2 out of 19 (10.5%) Cas12a targets presented off-target cleavage. Furthermore, while Cas9 targets with off-targets had an average of 16 different off-targets, Cas12a targets with off-targets only had an average of 1.5 different off-targets. Fewer different locations cleaved across the genome can help in reducing large deletions or chromosomal rearrangements. However, although off-targets were fewer with Cas12a, they were still present, and this indicates the importance of techniques like GUIDE-seq or DISCOVER-seq (Wienert et al., 2019).

These findings support my hypothesis that Cas12a is more specific than Cas9 due to a lower abundance of PAMs. However, the disproportional decrease in Cas12a off-targets, relative to the number of potential off-targets, suggests that Cas12a kinetics also lend favour to its increased specificity.

5.2.2.4 Limitations

A potential limitation to this analysis was that it only analysed results from GUIDE-seq. This is because while GUIDE-seq does detect double-strand breaks (DSBs), it does not detect other potential off-target events, like single-strand nicks or structural variants. This may lead to GUIDE-seq missing off-target effects induced by Cas12a at mismatch sites where single-strand nicks are likely. However, this may not be an issue due to how nicks are repaired. Because unlike cleavage, which can be repaired via the error prone NHEJ pathway, nicks are

usually repaired by the accurate single-strand break repair pathway (Caldecott, 2008). Although, if the nick is not repaired in a timely manner, it can cause the replication fork in replicating chromosomes to collapse, resulting in a DSB at a later point in time (Kuzminov, 2001). And although GUIDE-seq cannot capture structural variants or large-scale deletions, such variants are usually a consequence of DSBs (Kosicki et al., 2018). This means that using GUIDE-seq to understand the genomic DSB landscape can be an appropriate mitigation strategy to off-target effects.

Although useful for quantifying off-target effects, GUIDE-seq cannot be used to identify on-target effects like large deletions and translocations at the intended target (Newman et al., 2020). Although these are also usually a consequence of DSBs, with the aim of CRISPR experiments generally being to induce DSBs at the target, using GUIDE-seq to identify targets with few DSBs is counteractive. Therefore, for a comprehensive understanding of target effects, and perhaps to ensure no unexpected off-target effects slip through the cracks, sequencing is required where avoiding undesired off-target and on-target effects is imperative.

5.2.3 Predicting CRISPR-Cas12a editing outcomes

With Cas12a presenting benefits regarding specificity, I aimed to identify whether it was possible to predict template-free editing outcomes of Cas12a. This was based on the observation that the editing outcomes of NHEJ and microhomology-mediated end joining (MMEJ) are sequence dependent (Ata et al., 2018). This means that insertions or deletions that result from cleavage are not random but are instead based on the sequence of the cleaved allele. Because of this, I hypothesised the editing outcome from CRISPR-Cas12a to be predictable. Although less versatile than the template based HDR, it would provide an alternative for simple editing outcomes. One benefit is an increased efficiency due to the more-efficient nature of the repair mechanisms involved. And another benefit is that the complexity involved in designing and synthesising synthetic templates could be avoided.

For Cas9, there are already numerous computational tools that enable the prediction of editing outcome. This includes inDelphi (Shen et al., 2018), FORECasT (Allen et al., 2019) and SPROUT (Leenay et al., 2019). However, although deletions resulting from the MMEJ pathway may be independent of the CRISPR system used, Allen *et al.* proposed that some changes, like a single nucleotide thymine insertion, may be due to Cas9 kinetics. If so, such tools are unlikely to be generalisable to Cas12a, despite both nucleases triggering the same repair pathways.

The ability to test the hypothesis of predictable template-free editing with Cas12a had not been possible until recently due to the lack of a sufficiently sized dataset. However, a recent dataset provided more than 15,000 Cas12a (AsCpf1) sequenced targets (H. K. Kim et al., 2018). I postulated that a dataset of this size would be sufficient to model editing outcomes if outcome-modulating features do exist.

5.2.3.1 Insertion/deletion analysis in Cas12a targets

Before training a model, I aimed to quantify the mutational landscape of Cas12a. For this, I analysed the 15,000 samples (HT 1-1). These are synthetic targets, in that they are not endogenous to the host cell but synthesised in a plasmid vector, delivered by lentiviral particles, and integrated in the host genome. To verify the generalizability of editing outcomes, I also inspected a second dataset. This dataset included 55 targets (HEK-plasmid). However, one difference is that these are endogenous targets. Each target is present in the genome of HEK293T cell line, where the "-plasmid" suffix indicates that plasmid transfection was used as the delivery method for Cas12a and the crRNA. For each dataset, treated and control samples were available, where control samples were without Cas12a delivery and treated samples were at day six after Cas12a delivery. I used GOANA (in review) to identify variants and their respective frequencies.

In the HT 1-1 dataset, I found that deletions occurred more frequently than insertions, with 1.91 deletions for every insertion (independent *t*-test, t = 70.07, *P* = 0) (Figure 16). The difference between the two groups had a Cohen's *d* of 0.776. Deletions with a length of one (L1 deletions) were the most frequent distribution of deletion, accounting for 25% of all insertions and deletions (indels). Of note is that the most deletion distributions were significantly different ($P < 0.05$) due to the large sample size of the dataset. The only two distributions that were not different are L4 deletions and L5 deletions (independent *t*-test, t = 0.19, *P* = 0.85). These two distributions each accounted for approximately 15% of all indels. However, although the rest of the differences were significantly different, the difference between L1 deletions and L2 deletions was the greatest, with a Cohen's *d* of 0.544. For comparison, the second most different distributions, L2 deletions and L3 deletions, had a Cohen's *d* of 0.146, and the difference between L3 and L4 deletions was just 0.082.

For insertions, the most frequently occurring distribution, as with deletions, had a length of one (L1 insertions). However, unlike deletions, which had relatively similar distributions of L2 deletions and higher, insertion frequency decreased as length increased, with the longest detectable insertion being eight bases in length. The difference between every distribution was significant ($P < 0.05$) except for between L7 insertions and L8 insertions (independent *t*-test, t

= 0.94, *P* = 0.35). The difference between every group was large, with an average Cohen's *d* of 0.50.

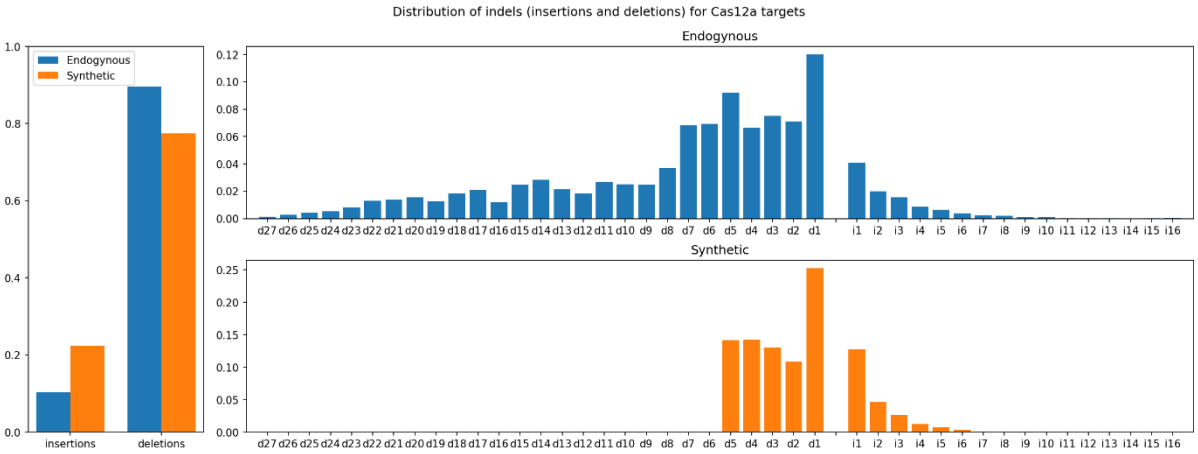


Figure 16 – distribution of mutations in Cas12a treated samples for endogenous (blue) and synthetic (orange) targets. Despite distributions being similar, synthetic targets feature a hard cut-off for deletions with a length greater than five, which is not present in endogenous targets.

Of interest is that unlike insertions, which gradually become less frequent as length increased, deletion frequencies remained relatively constant and then abruptly dropped to zero for L6 deletions. With deletions of length ten bases and longer having been observed before (Bernabé-Orts et al., 2019), I postulated that this aberrant drop in reads for deletions of six bases and longer was artefactual. Supporting the postulate that this was a technical error specific to the HT 1-1 dataset was the observation that this trend did not exist in the HEK-plasmid dataset (Figure 16). Instead, the frequency of deletions existed up to a length of 27 (L27 deletions). As with the HT 1-1 dataset, the difference between L1 deletions and L2 deletions was large (Cohen's *d* = 0.544). However, there was another drop between L7 deletions and L8 deletions (Cohen's *d* = 0.689) and between L8 deletions and L9 deletions (Cohen's *d* = 0.67).

Other properties were similar across the two datasets. Deletions again had a higher frequency than insertions (independent *t*-test, t = 13.16, *P* = 2.796e-37), however, the difference between insertion frequency and deletion frequency was greater in this dataset (Cohen's *d* = 0.958). Perhaps contributing to the increased rate of deletions in this dataset was that the full range of deletions were captured, unlike in the HT 1-1 dataset. This was supported by approximately half (53%) of all deletions being greater than five bases in length, the maximum deletion length in the HT 1-1 dataset. Adjusted for missing reads, the synthetic indel ratio would be 7.32:1, which is more comparable (1.2x) to that of the endogenous targets 8.63:1. These observations suggest that the HT 1-1 dataset, whilst large, is incomplete. This may limit its use in modelling editing outcome.

5.2.3.2 *Single nucleotide variant analysis in Cas12a targets*

Another difference between the HT 1-1 dataset (15,000) and the HEK-plasmid (55) dataset was the indel to single nucleotide variant (SNV) ratio. In the HEK-plasmid dataset, SNVs contributed to just 6.89% of all short variants. But in the HT 1-1 dataset, I identified this value to be an order of magnitude higher, with SNVs making up the majority (63.40%) of short variants. This large number of SNVs was unexpected as SNVs are not the typical outcome of targeted nucleases.

Because of the importance that the dataset was an accurate representation of CRISPR-Cas12a editing outcomes, I aimed to identify the reason behind the high SNV ratio. I plotted the SNV outcome counts at each position in the gRNA targets (Figure 17). Each column represented the number of reads where that SNV was observed, divided by the total number of reads where a SNV was observed. For control (no Cas12a) samples, SNVs existed in a uniform distribution across the length of the reads. However, for synthetic treated samples, SNVs existed in a bimodal distribution with the global maximum at nucleotide position 17, downstream from the PAM. This was three nucleotides upstream from the end of the 20nt target. Where the expected proportion of SNVs at each position in the sequenced region was 2.56%, the global maximum was more than 3x this value, at 7.82%. In fact, nearly half (47.30%) of all SNV-containing reads had an SNV at position 17 and its three adjacent positions (14 to 20). This data indicates that treating targets in the HT 1-1 dataset with Cas12a has resulted in PAM-distal SNVs at the target. It is these SNVs that result in the high SNV ratio in the HT 1-1 dataset.

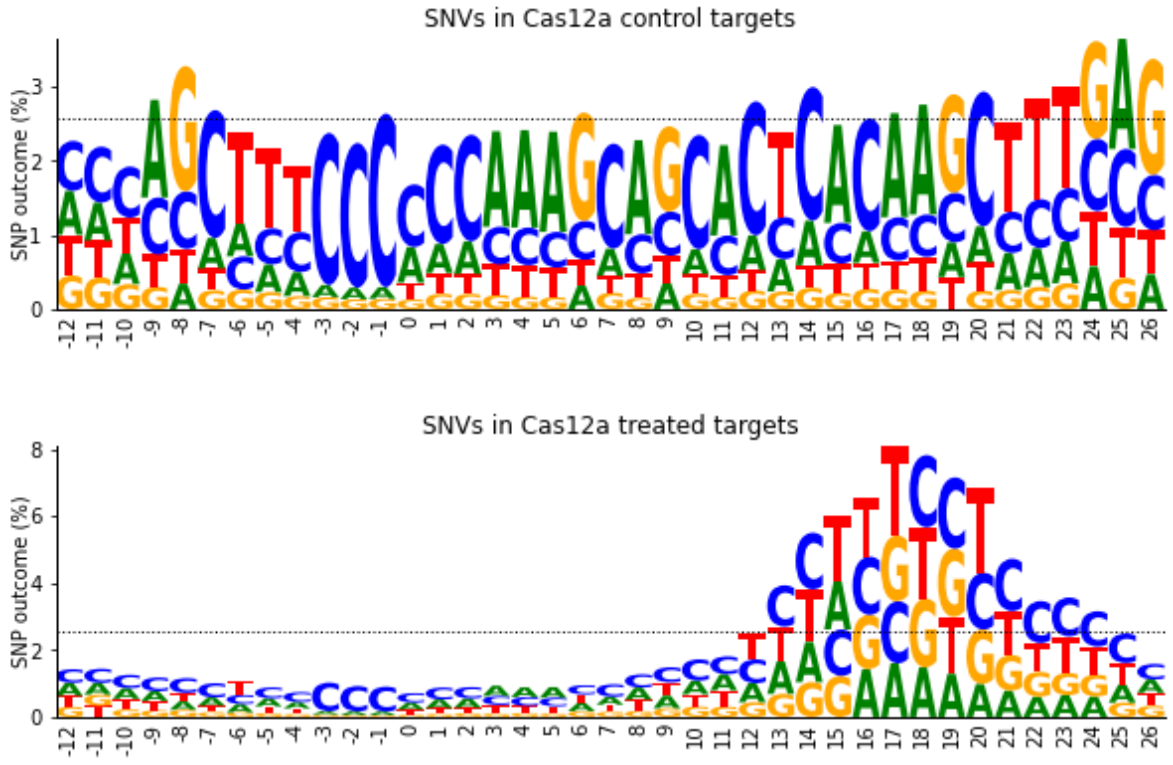
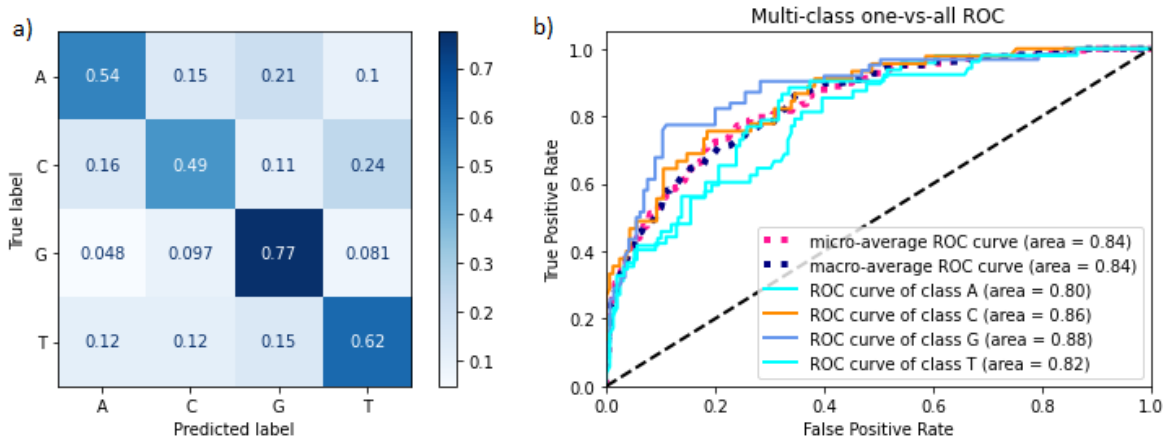


Figure 17 – SNV distribution across treated and control samples. The horizontal dotted line indicates the expected uniform distribution (2.56) for evenly distributed SNVs across the 29 positions. The letters indicate the editing outcome.

Although this observation suggested that certain positions were more prone to change, it was not possible to draw any other conclusions. To investigate whether features were present in the dataset that were modulating SNV outcome, I trained a Random Forest multiclass classifier on targets with an SNV at position 17. I labelled this set of targets with the SNV outcome (A, C, G or T). To filter out noise, I only included SNVs that were present in greater than 1% of target reads. I created features from the nucleotide sequence of the target and surrounding region, 39 bases in total. Although the feature that was resulting in this phenomenon remained unclear, I created the feature set based on the premise that prior nucleotides modulate NHEJ and MMEJ outcome. From five-fold cross-validation, I observed an average OOB error of 0.37. This indicated the five cross-validated models were predicting the outcome SNV for most samples correctly. To visualise this, I trained a model on 80% of the samples and validated the model on the remaining 20%

This model was able to predict most editing outcomes correctly. Where the outcome SNV is a G, the model was correct 77% of the time (Figure 18a). With three possible outcomes for each nucleotide, this is more than 3x greater than chance. Predicting an A outcome was the least accurate, with an accuracy of 49%, but this was still a 2x improvement over chance. To further visualise the prediction probability, I plotted the model as one vs. all receiver operating curves (ROCs) (Figure 18b). Each solid line indicates the ROC for one predicted nucleotide versus

the remaining three. The dotted lines indicate the averages. Further supporting the performance of this prediction model was the average area under the ROC being 0.84. (where 1 indicates perfect classification and 0 indicates random chance).



c)

guide_loc1_38_G	0.014740775256566946
guide_loc1_29_T	0.013783802789182113
guide_loc1_29_C	0.012834291550284396
guide_loc1_38_A	0.009995634548446021
guide_loc1_29_G	0.00913687328579708
guide_loc1_38_C	0.008113503386620894
guide_loc1_38_T	0.007590755827492714
guide_C	0.007434150472784694
guide_G:	0.0070629771389160104
guide_T:	0.0068325753393256045

Figure 18 – visualisations of a Random Forest model trained on 80% and validated on 20%. The label is the SNV outcome, which is one of ['A', 'C', 'G', 'T']. The features are a tokenised list of nucleotides at and surrounding the Cas12a 20nt target. (a) is a confusion matrix. (b) is a one vs. all ROC curve, where each label is alternately represented as true in each model. (c) is a list of the top ten important features from the RF model.

With the strong performance of the model, I inspected whether any notable features were contributing to the predicted editing outcome. From the list of 1,200 features, ten features with the highest importance are presented in Figure 18c. Of interest was that all four values representing the nucleotide at the 26th downstream position from the PAM were included in this subset. Also included were three out of the four values representing the original nucleotide in the 17th position downstream from the PAM (the position being predicted). The remaining three features represent the overall counts of C, G and T. To visualize the relationship of the nucleotide at position 26 and the outcome SNV at position 17, I generated a plot representing the original nucleotides present at each location in the sequence (Figure 19). However, I adjusted bar heights based on equality between the nucleotide being plotted, and outcome SNVs elsewhere in the sequence. As an example, if a position has no influence on SNV outcomes at other positions, the height of that bar will be one. As presented in Figure 19, this

is true for most positions. However, at position 26 the bar height is just under two. This indicates that this nucleotide is modulating the SNV outcome at other positions in the sequence, resulting in SNV outcomes being the same nucleotide as present at position 26. This is at a rate of twice as would be expected by chance. This also applies to position 25, albeit at a reduced rate of 1.25x chance.

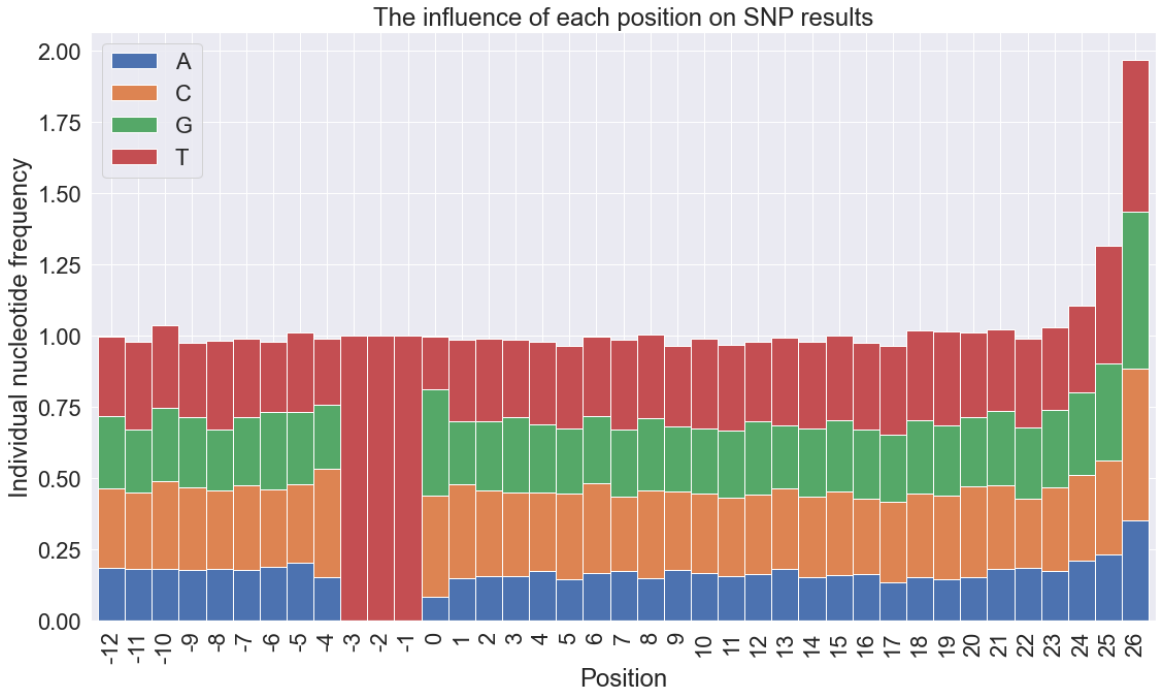


Figure 19 – this figure displays correlations between the nucleotide present at each position in the target, and outcome SNVs. For example, a bar of height one for T at position -1 indicates that for every change to a T elsewhere in the target, that there is a T present at position -1. Because position -1 is in the PAM and is guaranteed to be a T, its height will always be one. A bar height of 0.25 for T at position p indicates that for every change to a T in the target, 25% of the time the nucleotide present at position p is a T. Assuming an equal nucleotide distribution and random outcome, each colour will generally make up ~0.25, with a total bar height of ~1. A bar height > 1 indicates a correlation between the nucleotide at that position and SNV outcomes elsewhere in the target. For example, the bar at position 26 indicates that SNV outcomes across the target are on average twice as likely to be the same as the nucleotide present at position 26 as would be expected by chance.

Whilst this phenomenon is present in treated HT 1-1 samples, it is not present in control HT 1-1 samples. So, although unexpected, this suggests that the preference for SNVs over indels, and the influence of position 26, is a consequence of CRISPR-Cas12a. However, in treated samples from the HEK-plasmid dataset, results are as expected. SNV outcomes are the same as the control HT 1-1 samples, and there is a preference for indels over SNVs. From this I conclude that, while interesting, the observations from the HT 1-1 dataset are artefactual. In addition to the dataset lacking reads with deletions longer than five bases, the dataset is not a suitable candidate for modelling editing outcome.

5.2.4 Cas12a efficiency modelling

With the observation that the HT 1-1 dataset contains artefacts, I postulated that models trained on it would not be optimal for representing Cas12a. The current state of the art model for predicting Cas12a cleavage efficiency, DeepCpf1, was trained on the HT 1-1 dataset. This model is reported to perform well, with a Spearman's coefficient of 0.87 on the HEK-plasmid dataset (H. K. Kim et al., 2018). However, with this dataset having been trained on artefactual synthetic targets, I hypothesised that a model trained on endogenous targets would improve upon the performance. To support my hypothesis, I investigated differences in Cas12a cleavage efficiency between synthetic and endogenous CRISPR targets. Also, because DeepCpf1 models chromatin accessibility, I aimed to quantify the influence of chromatin accessibility on cleavage efficiency at accessible and inaccessible targets. After analysing the data, I analysed the performance of DeepCpf1 to identify what aspects of prediction performance could be improved upon. Finally, I trained Random Forest models on a public pooled-library screen dataset.

5.2.4.1 Statistical analysis of efficiency modulators

Although my analysis demonstrated a difference in editing outcome between HT 1-1 and HEK-plasmid, I had not yet analysed Cas12a cleavage efficiency in these datasets. Therefore, to quantify any differences in cleavage efficiency between synthetic and endogenous targets, I analysed cleavage efficiencies in these two datasets. I also included three more datasets in my analysis. This included HEK-lenti and HCT-plasmid. These two datasets both consisted of endogenous targets like HEK-plasmid, however, the former differed regarding CRISPR delivery (lentiviral transduced), and the latter differed regarding cell type (HCT116). I also included a synthetic lentiviral transduced dataset. Finally, I separated the three endogenous datasets into chromatin accessible and inaccessible targets. In total, this resulted in eight different datasets (Figure 20). To summarise, this included synthetic and endogenous targets, HEK293T and HCT116 cell types, lentiviral transduction and plasmid transfection CRISPR delivery methods, and chromatin accessible and inaccessible targets.

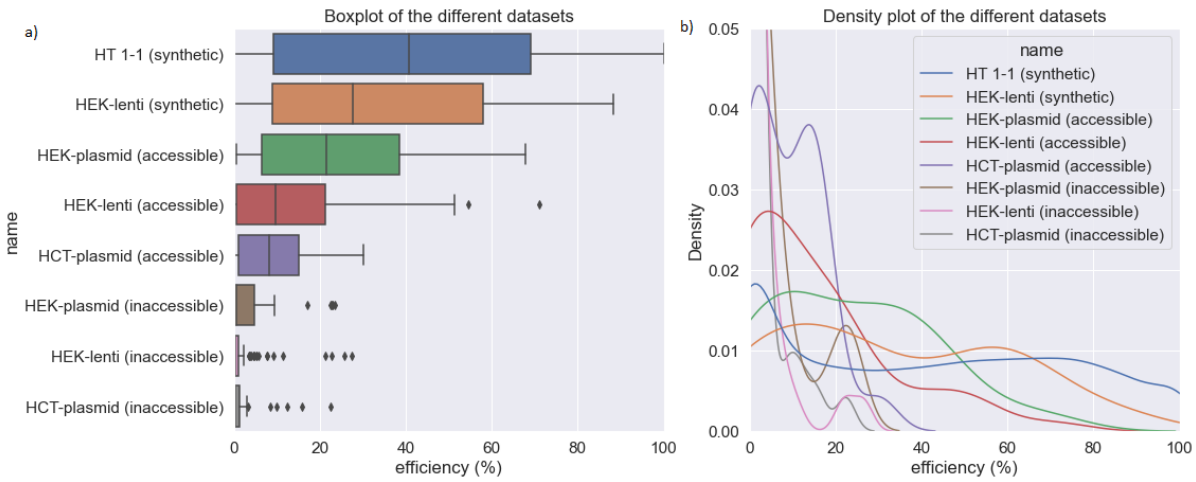


Figure 20 – distributions of efficiencies for the different datasets. (a) is a box and whisker plot demonstrating the different ranges. The inaccessible plot indicates most inaccessible targets are inefficient and that the more-efficient targets are outliers. (b) is a density plot visualising the different variances between groups.

The average cleavage efficiency had a high degree of variability between groups, with averages ranging from 2% to 45% (Figure 20a). However, cleavage efficiency also had a high degree of variability within groups, with variances ranging from 21 to 1045 (Figure 20b). To test for differences between groups, I used Welch's *t*-test. Welch's *t*-test is a version of the Students *t*-test that accounts for uneven variances (Welch, 1947). Cas12a cleavage was significantly more efficient in synthetic targets (M = 41.39, SD = 32.31) than in endogenous targets (M = 6.98, SD = 12.37); t = 43.06, *P* = 4.28e-139. However, a potential confounding variable was chromatin accessibility, as this has been observed to modulate cleavage efficiency (H. K. Kim et al., 2018). This is because while synthetic targets are inherently accessible, endogenous targets comprise of both accessible and inaccessible targets, depending on chromatin status. Out of the endogenous targets, only 34% (n = 92) were accessible. I therefore compared synthetic targets to accessible targets. I found that to a lesser degree, synthetic targets (M = 41.39, SD = 32.31) were significantly more efficient than accessible endogenous targets (M = 15.50, SD = 16.68); t = 14.72, *P* = 2.85e-26. The different distributions in cleavage efficiency support my hypothesis that synthetic data is not a good representation of CRISPR-Cas12a cleavage in general. Even when considering chromatin accessibility, the cleavage of synthetic CRISPR-Cas12a targets is significantly more efficient than that of accessible endogenous targets.

Two other factors that appeared to be correlated with cleavage efficiency were cell type and CRISPR delivery method (Figure 20a). Comparing different cell types for accessible targets with a plasmid delivery method, cleavage efficiency of targets in HEK293T cells (M = 23.85, SD = 18.91) was significantly higher than the cleavage efficiency of targets in HCT116 cells (M = 8.84, SD=8.13); t = 3.28, *P* = 0.003. Comparing different delivery methods for accessible

targets in HEK293 cells, a plasmid delivery method resulted in a higher cleavage efficiency (M = 23.85, SD = 18.91) than a lentiviral delivery method (M = 15.09, STD = 17.35), however this was not significant; t = 1.79, *P* = 0.08.

5.2.4.2 Chromatin aware efficiency prediction with DeepCpf1

Despite the significant difference in efficiency between accessible and inaccessible targets, only one currently available model uses chromatin accessibility as a feature. This is the DeepCpf1 model. All other models just use nucleotide features. This includes Seq-DeepCpf1, the model that DeepCpf1 is based on.

DeepCpf1 and Seq-DeepCpf1 are convolutional neural networks, with Seq-DeepCpf1 being trained on sequence information from the 15,000 synthetic HT 1-1 samples. However, DeepCpf1 extends Seq-DeepCpf1 by training an additional layer on chromatin accessibility information from the 148 endogenous HEK-lenti samples (H. K. Kim et al., 2018). But because DeepCpf1 extends Seq-DeepCpf1, both models include layers trained on data from the synthetic HT 1-1 dataset. Based on the differences in efficiency between synthetic and endogenous targets, as well as the artefacts present in the synthetic dataset, I hypothesised that the HT 1-1 dataset was not ideal for modelling endogenous targets.

In agreement with the published results, I found DeepCpf1 to result in higher Spearman correlation coefficients when predicting a mixture of accessible and inaccessible targets. With the HEK-lenti dataset, the coefficient improved from 0.573 (*P* = 2.684e-14) to 0.671 (*P* = 9.928e-21). The DeepCpf1 model achieves this improvement in through its additional neural network layer trained on chromatin accessibility data. From further inspection, this layer is equivalent to dividing the predicted efficiency of inaccessible targets by 7.1. Although this improves the score, it has multiple consequences. One is that it results in the maximum predicted efficiency of an inaccessible target being 14.08 $(100/7.1)$. Another is that it separates the relationship between predicted efficiencies into two distributions (Figure 21a). The samples in each distribution follow a monotonically increasing curve (Figure 21b). However, with both curves starting from zero, this results in a drop in average predicted efficiency between inaccessible and inaccessible targets. For example, a target predicted to be 10-15% efficient would be on average more efficient than a target predicted to be 20-25% efficient.

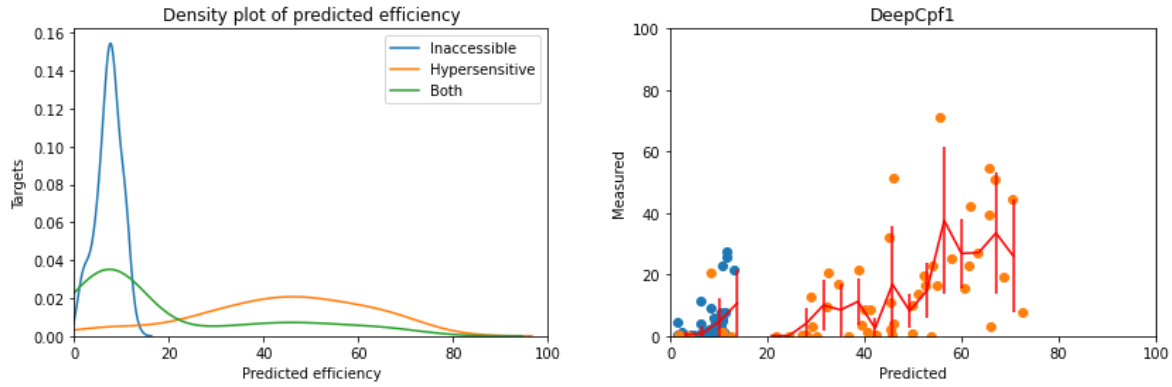


Figure 21 – a) a density plot of the predictions of the same targets to demonstrate the two distributions. The green curve represents all targets, and the blue and orange represent inaccessible and accessible targets, respectively. b) DeepCpf1 predictions for accessible (orange) and inaccessible (blue) targets. The red bar is a histogram for all targets in 20 bins which displays the average observed efficiency for predicted efficiencies. The non-linear relationship between true efficiency and predicted efficiency leads to poor predictability.

Because of these observations, I hypothesised that the Spearman correlation coefficient is overestimating the performance of DeepCpf1. Firstly, predicted efficiencies and actual efficiencies should ideally form a linear relationship, yet Spearman's coefficient measures the relationship between ranked values, rather than the linearity. Secondly, a previous study concluded that Spearman's coefficient should not be overinterpreted as a measure of strength of associations between two variables (Hauke & Kossowski, 2011). Therefore, I postulate that the Pearson correlation coefficient, which measures linear relationships, is a more appropriate measure of model performance.

Validating the DeepCp1 model on the HEK-lenti dataset, I inspected accessible and inaccessible targets separately. For accessible targets, Pearson's coefficient was 0.273 (P = 9.95e-05). For inaccessible targets, Pearson's coefficient was 0.180 (P = 1.33e-05). This indicates that predictions produced from DeepCpf1 only accounted for a small amount of variance of actual efficiencies. Pearson's coefficient was higher for the HEK-plasmid dataset but was still below 0.5 (0.466 (P = 9.14e-04) and 0.426 (P = 2.13e-05) for accessible and inaccessible targets, respectively).

These results suggested that the DeepCpf1 model has room for improvement, despite its high reported Spearman's coefficient. Firstly, all inaccessible targets were reported poorly, regardless of whether they are more efficient than accessible targets. Secondly, the linear relationship between predicted efficiencies and actual efficiencies had a high variance, as demonstrated by the low Spearman's coefficient. I proposed that the first point can be solved by not including predictions for accessible and inaccessible targets in the same set of results. However, the second point is more difficult problem. I hypothesised that a model trained on endogenous targets will outperform the DeepCpf1 model. I postulate that this is due to the

artefactual nature of the synthetic HT 1-1 dataset. To test this hypothesis, I aimed to train a machine learning model on an endogenous dataset and validate on the same datasets used to validate DeepCpf1.

5.2.4.3 Random Forest model

Although there were no endogenous datasets with the same sample size comparable to that of HT 1-1 (15,000 sgRNAs), I identified a pooled-library screen (Mini-human) with 2,061 samples across 687 human genes (J. Liu et al., 2019). In this knockout screen, Cas12a guides were designed for a set of genes, and these regions were sequenced at a series of time points. This enabled the calculation of guide efficiency through the log fold change of gene depletion. This calculation is based on the relative read count of each region. The log fold change is modulated by the confounding factor of whether a gene is essential, or not. That is, how important a gene is for the cell to remain viable. To mitigate this, I excluded genes with a low Bayes Factor (BF) in my preprocessing stage. BFs are a statistical measure of a gene belonging to an essential, or non-essential distribution (Hart & Moffat, 2016).

After preprocessing, the sample size was 306. Because of the relatively small sample size, I avoided deep learning and instead trained models using Random Forests. Because Random Forests lack the convolutional layers found in deep learning, I instead used the feature processing methodology from my previous chapters. This included global nucleotide counts and local nucleotide counts. However, I expanded on my methodology to sample discrete regions of the guide in a sliding window. This was more in line with the features that convolutional layers from neural networks can generate. For the label, I used Cas12a cleavage efficiency. I used five-fold cross validation on the Mini-human dataset for training and testing, and validated on each of the HEK-plasmid, HCT-plasmid and HEK-lenti datasets from (H. K. Kim et al., 2018).

I identified the model with the lowest OOB error and scored it using the Pearson correlation coefficient on accessible targets from each of the validation sets. These scores are reported in Table 9 along with Pearson's coefficients from the DeepCpf1 model. For each validation, the Pearson's coefficient of the Random Forest model was higher than the DeepCpf1 model. The Pearson's coefficient increased by from 14% for the HCT-plasmid dataset to 50% for the HEK-lenti dataset. This variation is likely a result of differences between the training data each model used, and differences in efficiency distributions between cell types and CRISPR delivery methods. Because where DeepCpf1 performs poorer on HEK-lenti (0.273) than HCT-plasmid (0.354), the Random Forest model has an equal performance on HEK-lenti (0.409)

and HCT-plasmid (0.404). Regardless of the differences, for all validations and with both models, the Pearson's coefficients were significant ($P < 0.05$).

	DeepCpf1	Random Forest model	Difference
HEK-plasmid	0.466 (P = 9.14e-04)	0.578 (P = 6.84e-05)	24%
HCT-plasmid	0.354 (P = 3.49e-03)	0.404 (P = 1.48e-03)	14%
HEK-lenti	0.273 (P = 9.95e-05)	0.409 (P = 5.64e-07)	50%

Table 9 – Pearson correlation coefficients for "DeepCpf1" and "RF" models on accessible targets from three validation sets. The difference column indicates the increase in these metrics for the RF model over DeepCpf1.

The improvement on the independent datasets achieved by Random Forests over DeepCpf1 was despite the training set for Random Forests being 50x smaller than the training set for DeepCpf1. This supported my hypothesis that training a model on endogenous data will result in an improved model over one trained on synthetic data. Despite the improvements, the models do share similarities regarding prediction outcomes on different datasets, which suggests that both models are lacking in features modelled. This is because the performance of both models varies depending on cell type and CRISPR delivery method. Both models tend to over-estimate or under-estimate the efficiencies of different datasets to different degrees (Figure 22).

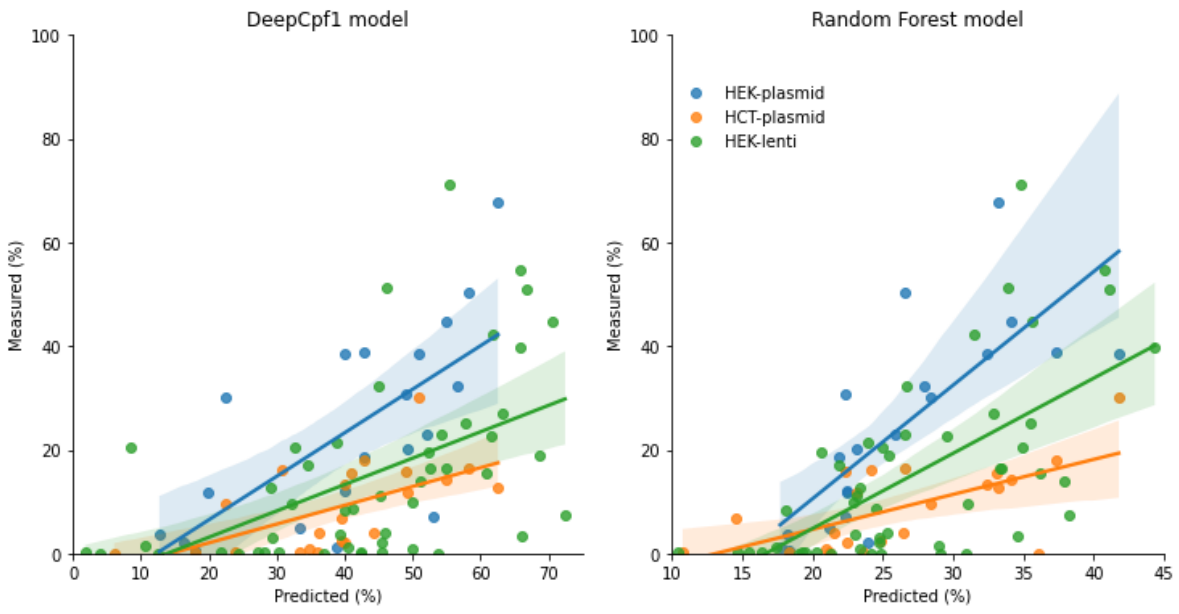


Figure 22 – prediction results for CeepCpf1 and the Random Forest model. The differing fitted linear models are due to different cell lines and CRISPR delivery methods resulting in different efficiency distributions. Although the Random Forest model predictions are closer to truth, it tends to underestimate efficiencies. The DeepCpf1 model tends to overestimate efficiencies.

The variations are correlated with the efficiency distributions of each datasets (Figure 20). For example, because the HCT-plasmid dataset features the lowest average efficiency, both

models will overestimate its efficiencies to a greater degree than targets from the more-efficient HEK datasets. This result suggests that a more-complete model would include cell type and delivery method. However, this would be at the cost of generalisability.

The results supported my hypothesis that a model trained on endogenous Cas12a targets will be more generalizable than a model trained on non-representative data. On accessible targets, the correlation between predicted cleavage efficiency and observed cleavage efficiency improved by between 14% to 50% with my Random Forest model. This improvement was despite having access to 50x fewer sgRNA samples than DeepCpf1. Based on this improvement, I released the model as Cas12aRF. Available in my repository, it can easily be loaded into Python to be used for predicting Cas12a sgRNA efficiency.

One similarity between both models was the tendency to overestimate or underestimate sgRNA efficiencies from different cell types and CRISPR delivery methods. Regarding cell types, this raises the question of what biological factors are modulating efficiency. However, to investigate this observation further, more data would be required. I propose that such a dataset would be the same size as the HT 1-1 dataset (15,000 sgRNAs), but with endogenous targets from a diverse range of annotated cell types. This would enable testing a wide range of epigenetic modifications.

Chapter 6 – General conclusions

6.1 General overview

Throughout my thesis, I aimed to improve the usability of CRISPR for genome editing. One part of this was through expanding on the knowledge of CRISPR systems and repair pathways. For example, elucidating variables that influence the likelihood of replacing a single nucleotide at a target site. The other part was to train predictive models that could be used to predict the efficiency of genome editing in mammalian cells using Cas9 or Cas12a effector nucleases. Such models could be used by researchers to design their own efficient editing experiments, improving outcomes, and reducing the number of attempts to result in the desired outcome. In each chapter I investigated and modelled a different repair pathway, CRISPR nuclease or editing technique.

For my first results chapter (Chapter 3) I aimed to improve the efficiency of using CRISPR-Cas9 to make precise edits to an allele. As well as the influence of the sgRNA, I also identified novel ssODN features that influence the efficiency of HDR-induced SNVs. This included the nucleotide content of the ssODN. However, rather than being the entire nucleotide content, I discovered that it was only the nucleotide content of the ssODN arm that initially interacts with the target site. Using this information, I trained the HDR efficiency prediction model, CUNE. This was published in *Scientific Reports* and is available freely to researchers as a web service.

My focus on precision editing continued into my second results chapter (Chapter 4) where I aimed to test the independence of two simultaneous HDR events, and to identify features that influence the efficiency. I identified a correlation between the efficiency of two successful insertion events and the distance between the two target sites. However, the feature that provided the highest correlation, was the efficiency of a single successful HDR event. Other more complex features, like experimental variables, had no significant influence on efficiency.

In my third results chapter (Chapter 5), I addressed one of the other main concerns of genome editing, off-target effects. I found evidence supporting the improved specificity that Cas12a offers over Cas9 with most analysed Cas12a targets having no detectable off-target effects. With the benefit of specificity, I aimed to train a model to predict Cas12a editing outcome. However, I identified that the only Cas12a dataset large enough for this task contained artefacts. With the current state of the art Cas12a efficiency prediction model having been trained on this dataset, I aimed to train a more accurate model on a smaller, but cleaner,

dataset. The resulting Random Forest model, Cas12aRF, improved prediction accuracy by up to 50%.

6.2 Contributions

My work resulted in a better understanding of different repair mechanisms, improving on the knowledge of this field. It also resulted in two publicly available CRISPR prediction models (CUNE and Cas12aRF). Through these models, researchers can design efficient CRISPR experiments.

My work on SNV insertion efficiency has led to a better understanding of the HDR pathway and features which modulate its efficiency. Although my results demonstrated the increased complexity involved in precision gene editing, they also expand upon the current knowledge of the pathways involved. For example, it illustrates the importance of nucleotide composition in biological processes like homologous recombination. It also indicates that gene editing with HDR is not just modulated by the sgRNA sequence, but also the ssODN sequence. These findings lead to more variables for researchers to trial when aiming to efficiently induce homologous recombination.

The model that resulted from this research was CUNE (computational universal nucleotide editor). Unlike other available CRISPR efficiency models, CUNE predicts the efficiency of precision editing using an ssODN. This enables the ability for researchers to design editing experiments *in silico* that have a higher chance of success. This can save time and money by reducing the number of editing attempts using inefficient sgRNA/ssODN. As well as predicting editing efficiency, CUNE will automatically design the sgRNA and ssODN. Therefore, CUNE can save time in the design of editing components. CUNE is generalisable to different laboratories as supported by evidence from Chapter 4 demonstrating the negligible influence of experimental parameters on HDR efficiency. CUNE is freely available to researchers as a web application, provided as part of the GT-Scan suite. It is easy to use, requiring just the loci and desired SNV. Also, as well as designing components for genome editing with HDR, CUNE will also identify whether base editing is possible for a target.

One of the other advantages of CUNE is its extendibility, which is essential in the evolving field of gene editing. Firstly, not only can rules for novel base editors be added in the future, but researchers can also define rules for their own proprietary base editors. Secondly, CUNE's serverless design means that it can be expanded with models for techniques like prime editing, allowing researchers to identify the most effective editing technique for their desired outcome.

Although computational tools like CUNE can enable efficient HDR gene editing experiments, techniques that require two simultaneous HDR events are inherently inefficient. This is because of the lower probability of two successful events. Current efficiencies can result in hundreds of failed attempts before achieving a success. Therefore, when two simultaneous edits are required, researchers may benefit from considering whether alternatives to HDR are available.

However, where long insertions are required and alternatives to HDR are not available, researchers can instead focus on two considerations to maximise efficiency. The first consideration is to ensure that the sgRNA/ssODN designs for each of the two target sites are efficient. The second consideration is that the two target sites are not too close together. My findings suggest 4,000 to 8,000 nucleotides apart is optimal, however more data is required to test longer distances.

I also proposed a method for predicting the number of attempts required, based on the observed efficiency of single HDR events. This means that researchers can identify targets with a high, or low, efficiency after just ten attempts. This can not only save researchers effort and money but can also improve animal welfare where animal models are used.

Just as important as efficiency, is specificity. The high specificity of Cas12a makes it an ideal candidate enzyme for tasks like gene therapy. This can save the effort and money involved in trialling multiple different guides to reduce off-target effects. But more importantly, as a viable alternative to Cas9, Cas12a expands the gene editing toolbox, enabling targeting of more alleles in a more diverse set of genomes.

My Cas12a prediction model, Cas12aRF aims to close this gap in this field. By providing accurate efficiency predictions, researchers can use it to enable efficient Cas12a cleavage. Because of its generalisability, researchers can be confident that its predictions will provide improvements over traditionally designed guides. In combination with computational tools to identify unique Cas12a targets, this can help researchers to design efficient and specific Cas12a targets unreachable with Cas9 enzymes. By enabling more efficient experiments, the uptake of Cas12a will likely increase, resulting in more data. This can enable further modelling of Cas12a and the ongoing improvement of Cas12aRF.

Throughout the process of locating, analysing and modelling data, one overarching observation was that large datasets were scarce or not representative. Although modelling smaller datasets is possible, as demonstrated with CUNE, larger datasets are always going to provide more accurate results than smaller ones, provided they are representative. Going forward, the ideal scenario is that more experimental data is released to the public. This may not always be possible due to data ownership concerns. However, it is a problem that needs to be considered. And considering the generalisability of models, even numerous small datasets would prove useful for future researchers.

Overall, this work demonstrates the versatility in applying machine learning algorithms to genome editing techniques. Even with a scarcity of data, CUNE and Cas12aRF accurately captured the systems they were trained to represent, in a generalisable way. Currently, and for the foreseeable future, prediction models are and will be essential for reducing wasted time and effort. This is especially true for inefficient editing techniques like HDR.

More generally, as the field of genome editing evolves and new editing techniques become available, there will be an increased need for sophisticated computational guidance. Tools that can recommend one technique over another and flag their individual limitations on the range of editing outcomes they can achieve. Prediction models will hence become essential to direct researchers to the editing technique that offers them the solution they need.

References

Abadi, S., Yan, W. X., Amar, D., & Mayrose, I. (2017). A machine learning approach for predicting CRISPR-Cas9 cleavage efficiencies and patterns underlying its mechanism of action. *PLoS Computational Biology*, *13*(10), e1005807. https://doi.org/10.1371/journal.pcbi.1005807

Aird, E. J., Lovendahl, K. N., St. Martin, A., Harris, R. S., & Gordon, W. R. (2018). Increasing Cas9-mediated homology-directed repair efficiency through covalent tethering of DNA repair template. *Communications Biology*, *1*(1), 1–6. https://doi.org/10.1038/s42003-018-0054-2

Allen, F., Crepaldi, L., Alsinet, C., Strong, A. J., Kleshchevnikov, V., De Angeli, P., Páleníková, P., Khodak, A., Kiselev, V., Kosicki, M., Bassett, A. R., Harding, H., Galanty, Y., Muñoz-Martínez, F., Metzakopian, E., Jackson, S. P., & Parts, L. (2019). Predicting the mutations generated by repair of Cas9-induced double-strand breaks. *Nature Biotechnology*, *37*(1), 64–82. https://doi.org/10.1038/nbt.4317

Alok, A., Sandhya, D., Jogam, P., Rodrigues, V., Bhati, K. K., Sharma, H., & Kumar, J. (2020). The Rise of the CRISPR/Cpf1 System for Efficient Genome Editing in Plants. *Frontiers in Plant Science*, *11*, 264. https://doi.org/10.3389/fpls.2020.00264

Alonso, J. M., & Ecker, J. R. (2006). Moving forward in reverse: Genetic technologies to enable genome-wide phenomic screens in Arabidopsis. In *Nature Reviews Genetics* (Vol. 7, Issue 7, pp. 524–536). Nature Publishing Group. https://doi.org/10.1038/nrg1893

Amancio, D. R., Comin, C. H., Casanova, D., Travieso, G., Bruno, O. M., Rodrigues, F. A., & Da Fontoura Costa, L. (2014). A systematic comparison of supervised classifiers. *PLOS ONE*, *9*(4), e94137. https://doi.org/10.1371/journal.pone.0094137

Anders, C., Niewoehner, O., Duerst, A., & Jinek, M. (2014). Structural basis of PAM-dependent target DNA recognition by the Cas9 endonuclease. *Nature*, *513*(7519), 569–573. https://doi.org/10.1038/nature13579

Anderson, E. M., Haupt, A., Schiel, J. A., Chou, E., Machado, H. B., Strezoska, Ž., Lenger, S., McClelland, S., Birmingham, A., Vermeulen, A., & Smith, A. V. B. (2015). Systematic analysis of CRISPR-Cas9 mismatch tolerance reveals low levels of off-target activity. *Journal of Biotechnology*, *211*, 56–65. https://doi.org/10.1016/j.jbiotec.2015.06.427

Anzalone, A. V., Randolph, P. B., Davis, J. R., Sousa, A. A., Koblan, L. W., Levy, J. M., Chen, P. J., Wilson, C., Newby, G. A., Raguram, A., & Liu, D. R. (2019). Search-and-replace genome editing without double-strand breaks or donor DNA. *Nature*, *576*(7785), 149–157. https://doi.org/10.1038/s41586-019-1711-4

Ata, H., Ekstrom, T. L., Martínez-Gálvez, G., Mann, C. M., Dvornikov, A. V., Schaefbauer, K. J., Ma, A. C., Dobbs, D., Clark, K. J., & Ekker, S. C. (2018). Robust activation of microhomology-mediated end joining for precision gene editing applications. *PLOS Genetics*, *14*(9). https://doi.org/10.1371/journal.pgen.1007652

AYABE, S., NAKASHIMA, K., & YOSHIKI, A. (2019). Off- and on-target effects of genome editing in mouse embryos. *Journal of Reproduction and Development*, *65*(1). https://doi.org/10.1262/jrd.2018-128

Bae, S., Park, J., & Kim, J. S. (2014). Cas-OFFinder: A fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases. *Bioinformatics*, *30*(10), 1473–1475. https://doi.org/10.1093/bioinformatics/btu048

Bayat, A., Szul, P., O'Brien, A. R., Dunne, R., Hosking, B., Jain, Y., Hosking, C., Luo, O. J.,

Twine, N., & Bauer, D. C. (2020). VariantSpark: Cloud-based machine learning for association study of complex phenotype and large-scale genomic data. *GigaScience*, *9*(8), 1–12. https://doi.org/10.1093/gigascience/giaa077

Bedell, M. A., Jenkins, N. A., & Copeland, N. G. (1997). Mouse models of human disease. Part I: Techniques and resources for genetic analysis in mice. In *Genes and Development* (Vol. 11, Issue 1, pp. 1–10). Cold Spring Harbor Laboratory Press. https://doi.org/10.1101/gad.11.1.1

Belfort, M., & Roberts, R. J. (1997). Homing endonucleases: Keeping the house in order. In *Nucleic Acids Research* (Vol. 25, Issue 17, pp. 3379–3388). Oxford University Press. https://doi.org/10.1093/nar/25.17.3379

Bengio, Y. (2009). Learning deep architectures for AI. In *Foundations and Trends in Machine Learning* (Vol. 2, Issue 1). Now Publishers Inc. https://doi.org/10.1561/2200000006

Bernabé-Orts, J. M., Casas-Rodrigo, I., Minguet, E. G., Landolfi, V., Garcia-Carpintero, V., Gianoglio, S., Vázquez-Vilar, M., Granell, A., & Orzaez, D. (2019). Assessment of Cas12a-mediated gene editing efficiency in plants. *Plant Biotechnology Journal*, *17*(10), 1971–1984. https://doi.org/10.1111/pbi.13113

Bhadauria, V., Banniza, S., Wei, Y., & Peng, Y. L. (2009). Reverse genetics for functional genomics of phytopathogenic fungi and oomycetes. In *Comparative and Functional Genomics* (Vol. 2009). https://doi.org/10.1155/2009/380719

Bialk, P., Rivera-Torres, N., Strouse, B., & Kmiec, E. B. (2015). Regulation of gene editing activity directed by single-stranded oligonucleotides and CRISPR/Cas9 systems. *PLOS ONE*, *10*(6), e0129308. https://doi.org/10.1371/journal.pone.0129308

Bin Moon, S., Lee, J. M., Kang, J. G., Lee, N. E., Ha, D. I., Kim, D. Y., Kim, S. H., Yoo, K., Kim, D., Ko, J. H., & Kim, Y. S. (2018). Highly efficient genome editing by CRISPR-Cpf1 using CRISPR RNA with a uridinylate-rich 3′-overhang. *Nature Communications*, *9*(1), 1–11. https://doi.org/10.1038/s41467-018-06129-w

Bitinaite, J., Wah, D. A., Aggarwal, A. K., & Schildkraut, I. (1998). Foki dimerization is required for dna cleavage. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(18), 10570–10575. https://doi.org/10.1073/pnas.95.18.10570

Boch, J., Scholze, H., Schornack, S., Landgraf, A., Hahn, S., Kay, S., Lahaye, T., Nickstadt, A., & Bonas, U. (2009). Breaking the code of DNA binding specificity of TAL-type III effectors. *Science*, *326*(5959), 1509–1512. https://doi.org/10.1126/science.1178811

Bogdanove, A. J., Schornack, S., & Lahaye, T. (2010). TAL effectors: Finding plant genes for disease and defense. In *Current Opinion in Plant Biology* (Vol. 13, Issue 4, pp. 394–401). Elsevier Current Trends. https://doi.org/10.1016/j.pbi.2010.04.010

Bothmer, A., Phadke, T., Barrera, L. A., Margulies, C. M., Lee, C. S., Buquicchio, F., Moss, S., Abdulkerim, H. S., Selleck, W., Jayaram, H., Myer, V. E., & Cotta-Ramusino, C. (2017). Characterization of the interplay between DNA repair and CRISPR/Cas9-induced DNA lesions at an endogenous locus. *Nature Communications*, *8*(1), 1–12. https://doi.org/10.1038/ncomms13905

Bradford, J., & Perrin, D. (2019). A benchmark of computational CRISPR-Cas9 guide design methods. *PLOS Computational Biology*, *15*(8), e1007274. https://doi.org/10.1371/journal.pcbi.1007274

Breiman, L. (2001a). Statistical modeling: The two cultures. *Statistical Science*, *16*(3), 199–215. https://doi.org/10.1214/ss/1009213726

Breiman, L. (2001b). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brown, A. J., Fisher, D. A., Kouranova, E., McCoy, A., Forbes, K., Wu, Y., Henry, R., Ji, D., Chambers, A., Warren, J., Shu, W., Weinstein, E. J., & Cui, X. (2013). Whole-rat conditional gene knockout via genome editing. *Nature Methods*, *10*(7), 638–640. https://doi.org/10.1038/nmeth.2516

Caldecott, K. W. (2008). Single-strand break repair and genetic disease. In *Nature Reviews Genetics* (Vol. 9, Issue 8, pp. 619–631). Nature Publishing Group. https://doi.org/10.1038/nrg2380

Capecchi, M. R. (1989). The new mouse genetics: Altering the genome by gene targeting. *Trends in Genetics*, *5*(C), 70–76. https://doi.org/10.1016/0168-9525(89)90029-2

Capecchi, M. R. (2005). Gene targeting in mice: Functional analysis of the mammalian genome for the twenty-first century. In *Nature Reviews Genetics* (Vol. 6, Issue 6, pp. 507–512). Nature Publishing Group. https://doi.org/10.1038/nrg1619

Cermak, T., Doyle, E. L., Christian, M., Wang, L., Zhang, Y., Schmidt, C., Baller, J. A., Somia, N. V., Bogdanove, A. J., & Voytas, D. F. (2011). Efficient design and assembly of custom TALEN and other TAL effector-based constructs for DNA targeting. *Nucleic Acids Research*, *39*(12), e82–e82. https://doi.org/10.1093/nar/gkr218

Chari, R., Mali, P., Moosburner, M., & Church, G. M. (2015). Unraveling CRISPR-Cas9 genome engineering parameters via a library-on-library approach. *Nature Methods*, *12*(9), 823–826. https://doi.org/10.1038/nmeth.3473

Chari, R., Yeo, N. C., Chavez, A., & Church, G. M. (2017). SgRNA Scorer 2.0: A Species-Independent Model to Predict CRISPR/Cas9 Activity. *ACS Synthetic Biology*, *6*(5), 902–904. https://doi.org/10.1021/acssynbio.6b00343

Chen, C., Liaw, A., & Breiman, L. (2004). Using Random Forest to Learn Imbalanced Data | Department of Statistics. In *University of California, Berkeley* (Vol. 110, Issue 24). https://statistics.berkeley.edu/tech-reports/666

Cheng, T. L., Li, S., Yuan, B., Wang, X., Zhou, W., & Qiu, Z. (2019). Expanding C–T base editing toolkit with diversified cytidine deaminases. *Nature Communications*, *10*(1), 1–10. https://doi.org/10.1038/s41467-019-11562-6

Cho, S. W., Kim, S., Kim, Y., Kweon, J., Kim, H. S., Bae, S., & Kim, J. S. (2014). Analysis of off-target effects of CRISPR/Cas-derived RNA-guided endonucleases and nickases. *Genome Research*, *24*(1), 132–141. https://doi.org/10.1101/gr.162339.113

Christian, M., Cermak, T., Doyle, E. L., Schmidt, C., Zhang, F., Hummel, A., Bogdanove, A. J., & Voytas, D. F. (2010). Targeting DNA double-strand breaks with TAL effector nucleases. *Genetics*, *186*(2), 756–761. https://doi.org/10.1534/genetics.110.120717

Chuai, Guo hui, Wang, Q. L., & Liu, Q. (2017). In Silico Meets In Vivo: Towards Computational CRISPR-Based sgRNA Design. In *Trends in Biotechnology* (Vol. 35, Issue 1, pp. 12–21). Elsevier Ltd. https://doi.org/10.1016/j.tibtech.2016.06.008

Chuai, Guohui, Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., Zhou, C., Zhu, C., Chen, K., Duan, B., Gu, F., Qu, S., Huang, D., Wei, J., & Liu, Q. (2018). DeepCRISPR: Optimized CRISPR guide RNA design by deep learning. *Genome Biology*, *19*(1), 80. https://doi.org/10.1186/s13059-018-1459-4

Chylinski, K., Le Rhun, A., & Charpentier, E. (2013). The tracrRNA and Cas9 families of type II CRISPR-Cas immunity systems. *RNA Biology*, *10*(5), 726–737. https://doi.org/10.4161/rna.24321

Cohen, J. (1977). *Statistical Power Analysis for the Behavioral Sciences*. Elsevier. https://doi.org/10.1016/c2013-0-10517-x

Cong, L., Ran, F. A., Cox, D., Lin, S., Barretto, R., Habib, N., Hsu, P. D., Wu, X., Jiang, W., Marraffini, L. A., & Zhang, F. (2013). Multiplex genome engineering using CRISPR/Cas

systems. *Science*, *339*(6121), 819–823. https://doi.org/10.1126/science.1231143

Cox, D. R. (2006). Principles of statistical inference. In *Principles of Statistical Inference*. Cambridge University Press. https://doi.org/10.1017/CBO9780511813559

Cui, Y., Xu, J., Cheng, M., Liao, X., & Peng, S. (2018). Review of CRISPR/Cas9 sgRNA Design Tools. In *Interdisciplinary Sciences: Computational Life Sciences* (Vol. 10, Issue 2, pp. 455–465). Springer Berlin Heidelberg. https://doi.org/10.1007/s12539-018-0298-z

Danda, R., Krishnan, G., Ganapathy, K., Krishnan, U. M., Vikas, K., Elchuri, S., Chatterjee, N., & Krishnakumar, S. (2013). Targeted expression of suicide gene by tissue-specific promoter and microRNA regulation for cancer gene therapy. *PLoS ONE*, *8*(12), 83398. https://doi.org/10.1371/journal.pone.0083398

Davis, C. A., Hitz, B. C., Sloan, C. A., Chan, E. T., Davidson, J. M., Gabdank, I., Hilton, J. A., Jain, K., Baymuradov, U. K., Narayanan, A. K., Onate, K. C., Graham, K., Miyasato, S. R., Dreszer, T. R., Strattan, J. S., Jolanki, O., Tanaka, F. Y., & Cherry, J. M. (2018). The Encyclopedia of DNA elements (ENCODE): data portal update. *Nucleic Acids Research*, *46*, D794–D801. https://doi.org/10.1093/nar/gkx1081

Dehal, P. S., Joachimiak, M. P., Price, M. N., Bates, J. T., Baumohl, J. K., Chivian, D., Friedland, G. D., Huang, K. H., Keller, K., Novichkov, P. S., Dubchak, I. L., Alm, E. J., & Arkin, A. P. (2009). MicrobesOnline: An integrated portal for comparative and functional genomics. *Nucleic Acids Research*, *38*(SUPPL.1), D396–D400. https://doi.org/10.1093/nar/gkp919

Dianov, G. L., & Hübscher, U. (2013). Mammalian base excision repair: The forgotten archangel. In *Nucleic Acids Research* (Vol. 41, Issue 6, pp. 3483–3490). https://doi.org/10.1093/nar/gkt076

Doench, J. G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E. W., Donovan, K. F., Smith, I., Tothova, Z., Wilen, C., Orchard, R., Virgin, H. W., Listgarten, J., & Root, D. E. (2016). Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature Biotechnology*, *34*(2), 184–191. https://doi.org/10.1038/nbt.3437

Doench, J. G., Hartenian, E., Graham, D. B., Tothova, Z., Hegde, M., Smith, I., Sullender, M., Ebert, B. L., Xavier, R. J., & Root, D. E. (2014). Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. *Nature Biotechnology*, *32*(12), 1262–1267. https://doi.org/10.1038/nbt.3026

Dolgin, E. (2017). The greatest hits of the human genome: A tour through the most studied genes in biology reveals some surprises. *Nature*, *551*(7681), 427–431. https://doi.org/10.1038/d41586-017-07291-9

Domingos, P. (2012). A few useful things to know about machine learning. In *Communications of the ACM* (Vol. 55, Issue 10, pp. 78–87). https://doi.org/10.1145/2347736.2347755

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, *97*(457), 77–86. https://doi.org/10.1198/016214502753479248

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *Proceedings of the Seventh International Conference on Information and Knowledge Management - CIKM '98*, 148–155. https://doi.org/10.1145/288627.288651

Durai, S., Mani, M., Kandavelou, K., Wu, J., Porteus, M. H., & Chandrasegaran, S. (2005). Zinc finger nucleases: Custom-designed molecular scissors for genome engineering of plant and mammalian cells. *Nucleic Acids Research*, *33*(18), 5978–5990. https://doi.org/10.1093/nar/gki912

Elliott, B., Richardson, C., Winderbaum, J., Nickoloff, J. A., & Jasin, M. (1998). Gene Conversion Tracts from Double-Strand Break Repair in Mammalian Cells. *Molecular and Cellular Biology*, *18*(1), 93–101. https://doi.org/10.1128/mcb.18.1.93

Fonfara, I., Richter, H., BratoviÄ, M., Le Rhun, A., & Charpentier, E. (2016). The CRISPR-associated DNA-cleaving enzyme Cpf1 also processes precursor CRISPR RNA. *Nature*, *532*(7600), 517–521. https://doi.org/10.1038/nature17945

Frankenberg-Schwager, M., Frankenberg, D., & Harbich, R. (1985). Potentially Lethal Damage, Sublethal Damage and DNA Double Strand Breaks. *Radiation Protection Dosimetry*, *13*(1–4), 171–174. https://doi.org/10.1093/rpd/13.1-4.171

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Fu, B. X. H., Smith, J. D., Fuchs, R. T., Mabuchi, M., Curcuru, J., Robb, G. B., & Fire, A. Z. (2019). Target-dependent nickase activities of the CRISPR–Cas nucleases Cpf1 and Cas9. *Nature Microbiology*, *4*(5), 888–897. https://doi.org/10.1038/s41564-019-0382-0

Fu, Y., Foden, J. A., Khayter, C., Maeder, M. L., Reyon, D., Joung, J. K., & Sander, J. D. (2013). High-frequency off-target mutagenesis induced by CRISPR-Cas nucleases in human cells. *Nature Biotechnology*, *31*(9), 822–826. https://doi.org/10.1038/nbt.2623

Gao, Y., Chuai, G., Yu, W., Qu, S., & Liu, Q. (2019). Data imbalance in CRISPR off-target prediction. *Briefings in Bioinformatics*, *00*(00), 1. https://doi.org/10.1093/bib/bbz069

Garneau, J. E., Dupuis, M. È., Villion, M., Romero, D. A., Barrangou, R., Boyaval, P., Fremaux, C., Horvath, P., Magadán, A. H., & Moineau, S. (2010). The CRISPR/cas bacterial immune system cleaves bacteriophage and plasmid DNA. *Nature*, *468*(7320), 67–71. https://doi.org/10.1038/nature09523

Gasiunas, G., Barrangou, R., Horvath, P., & Siksnys, V. (2012). Cas9-crRNA ribonucleoprotein complex mediates specific DNA cleavage for adaptive immunity in bacteria. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(39). https://doi.org/10.1073/pnas.1208507109

Gaudelli, N. M., Komor, A. C., Rees, H. A., Packer, M. S., Badran, A. H., Bryson, D. I., & Liu, D. R. (2017). Programmable base editing of T to G C in genomic DNA without DNA cleavage. *Nature*, *551*(7681), 464–471. https://doi.org/10.1038/nature24644

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, *70*(350), 320–328. https://doi.org/10.1080/01621459.1975.10479865

Gilchrist, E., & Haughn, G. (2010). Reverse genetics techniques: Engineering loss and gain of gene function in plants. *Briefings in Functional Genomics and Proteomics*, *9*(2), 103–110. https://doi.org/10.1093/bfgp/elp059

Gong, S., Yu, H. H., Johnson, K. A., & Taylor, D. W. (2018). DNA Unwinding Is the Primary Determinant of CRISPR-Cas9 Activity. *Cell Reports*, *22*(2), 359–371. https://doi.org/10.1016/j.celrep.2017.12.041

Gootenberg, J. S., Abudayyeh, O. O., Kellner, M. J., Joung, J., Collins, J. J., & Zhang, F. (2018). Multiplexed and portable nucleic acid detection platform with Cas13, Cas12a and Csm6. *Science*, *360*(6387), 439–444. https://doi.org/10.1126/science.aaq0179

Gootenberg, J. S., Abudayyeh, O. O., Lee, J. W., Essletzbichler, P., Dy, A. J., Joung, J., Verdine, V., Donghia, N., Daringer, N. M., Freije, C. A., Myhrvold, C., Bhattacharyya, R. P., Livny, J., Regev, A., Koonin, E. V., Hung, D. T., Sabeti, P. C., Collins, J. J., & Zhang, F. (2017). Nucleic acid detection with CRISPR-Cas13a/C2c2. *Science*, *356*(6336), 438–442. https://doi.org/10.1126/science.aam9321

Gossler, A., Doetschman, T., Korn, R., Serfling, E., & Kemler, R. (1986). Transgenesis by

means of blastocyst-derived embryonic stem cell lines. *Proceedings of the National Academy of Sciences of the United States of America*, *83*(23), 9065–9069. https://doi.org/10.1073/pnas.83.23.9065

Gu, B., Posfai, E., & Rossant, J. (2018). Efficient generation of targeted large insertions by microinjection into two-cell-stage mouse embryos. *Nature Biotechnology*, *36*(7), 632–637. https://doi.org/10.1038/nbt.4166

Gu, H., Marth, J. D., Orban, P. C., Mossmann, H., & Rajewsky, K. (1994). Deletion of a DNA polymerase β gene segment in T cells using cell type-specific gene targeting. *Science*, *265*(5168), 103–106. https://doi.org/10.1126/science.8016642

Gu, H., Zou, Y. R., & Rajewsky, K. (1993). Independent control of immunoglobulin switch recombination at individual switch regions evidenced through Cre-loxP-mediated gene targeting. *Cell*, *73*(6), 1155–1164. https://doi.org/10.1016/0092-8674(93)90644-6

Guo, Q., Mintier, G., Ma-Edmonds, M., Storton, D., Wang, X., Xiao, X., Kienzle, B., Zhao, D., & Feder, J. N. (2018). "Cold shock" increases the frequency of homology directed repair gene editing in induced pluripotent stem cells. *Scientific Reports*, *8*(1), 1–11. https://doi.org/10.1038/s41598-018-20358-5

Gurumurthy, C. B., O'Brien, A. R., Quadros, R. M., Adams, J., Alcaide, P., Ayabe, S., Ballard, J., Batra, S. K., Beauchamp, M. C., Becker, K. A., Bernas, G., Brough, D., Carrillo-Salinas, F., Chan, W., Chen, H., Dawson, R., Demambro, V., D'Hont, J., Dibb, K. M., … Burgio, G. (2019). Reproducibility of CRISPR-Cas9 methods for generation of conditional mouse alleles: A multi-center evaluation. *Genome Biology*, *20*(1), 171. https://doi.org/10.1186/s13059-019-1776-2

Hadjantonakis, A. K., Pirity, M., & Nagy, A. (2008). Cre recombinase mediated alterations of the mouse genome using embryonic stem cells. *Methods in Molecular Biology*, *461*, 111–132. https://doi.org/10.1007/978-1-60327-483-8_8

Haeussler, M., Schönig, K., Eckert, H., Eschstruth, A., Mianné, J., Renaud, J. B., Schneider-Maunoury, S., Shkumatava, A., Teboul, L., Kent, J., Joly, J. S., & Concordet, J. P. (2016). Evaluation of off-target and on-target scoring algorithms and integration into the guide RNA selection tool CRISPOR. *Genome Biology*, *17*(1), 148. https://doi.org/10.1186/s13059-016-1012-2

Hall, M. A., & Smith, L. A. (1999). Feature Selection for Machine Learning : Comparing a Correlation-based Filter Approach to the Wrapper. *Proceedings of the Twelfth International FLAIRS Conference*. https://www.aaai.org/Library/FLAIRS/1999/flairs99-042.php

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29–36. https://doi.org/10.1148/radiology.143.1.7063747

Hardison, R. C. (2003). Comparative genomics. In *PLoS Biology* (Vol. 1, Issue 2). Public Library of Science. https://doi.org/10.1371/journal.pbio.0000058

Hardouin, S. N., & Nagy, A. (2000). Mouse models for human disease. *Clinical Genetics*, *57*(4), 237–244. https://doi.org/10.1034/j.1399-0004.2000.570401.x

Hart, T., Chandrashekhar, M., Aregger, M., Steinhart, Z., Brown, K. R., MacLeod, G., Mis, M., Zimmermann, M., Fradet-Turcotte, A., Sun, S., Mero, P., Dirks, P., Sidhu, S., Roth, F. P., Rissland, O. S., Durocher, D., Angers, S., & Moffat, J. (2015). High-Resolution CRISPR Screens Reveal Fitness Genes and Genotype-Specific Cancer Liabilities. *Cell*, *163*(6), 1515–1526. https://doi.org/10.1016/j.cell.2015.11.015

Hart, T., & Moffat, J. (2016). BAGEL: A computational framework for identifying essential genes from pooled library screens. *BMC Bioinformatics*, *17*(1), 164. https://doi.org/10.1186/s12859-016-1015-8

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.

Hauke, J., & Kossowski, T. (2011). Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data. *Quaestiones Geographicae*, *30*(2), 87–93. https://doi.org/10.2478/v10117-011-0021-1

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hearst, M. A. (1998). Support Vector Machines. *IEEE Intelligent Systems*, *13*(4), 18–28. https://doi.org/10.1109/5254.708428

Hieter, P., & Boguski, M. (1997). Functional genomics: It's all how you read it. In *Science* (Vol. 278, Issue 5338, pp. 601–602). American Association for the Advancement of Science. https://doi.org/10.1126/science.278.5338.601

Hinz, J. M., Laughery, M. F., & Wyrick, J. J. (2015). Nucleosomes Inhibit Cas9 Endonuclease Activity in Vitro. *Biochemistry*, *54*(48), 7063–7066. https://doi.org/10.1021/acs.biochem.5b01108

Hoess, R. H., Ziese, M., & Sternberg, N. (1982). P1 site-specific recombination: Nucleotide sequence of the recombining sites. *Proceedings of the National Academy of Sciences of the United States of America*, *79*(11 I), 3398–3402. https://doi.org/10.1073/pnas.79.11.3398

Hogan, B., Beddington, R., Costantini, F., & Lacy, E. (1994). *Manipulating the Mouse Embryo*. Cold Spring Harbor Laboratory Press. https://www.google.com.au/books/edition/Manipulating_the_Mouse_Embryo/TNm2QgAACAAJ?hl=en

Horii, T., Morita, S., Kimura, M., Terawaki, N., Shibutani, M., & Hatada, I. (2017). Efficient generation of conditional knockout mice via sequential introduction of lox sites. *Scientific Reports*, *7*(1), 1–8. https://doi.org/10.1038/s41598-017-08496-8

Hruscha, A., Krawitz, P., Rechenberg, A., Heinrich, V., Hecht, J., Haass, C., & Schmid, B. (2013). Efficient CRISPR/Cas9 genome editing with low off-target effects in zebrafish. *Development*, *140*(24), 4982–4987. https://doi.org/10.1242/dev.099085

Hsu, P. D., Scott, D. A., Weinstein, J. A., Ran, F. A., Konermann, S., Agarwala, V., Li, Y., Fine, E. J., Wu, X., Shalem, O., Cradick, T. J., Marraffini, L. A., Bao, G., & Zhang, F. (2013). DNA targeting specificity of RNA-guided Cas9 nucleases. *Nature Biotechnology*, *31*(9), 827–832. https://doi.org/10.1038/nbt.2647

Hughes, G. F. (1968). On the Mean Accuracy of Statistical Pattern Recognizers. *IEEE Transactions on Information Theory*, *14*(1), 55–63. https://doi.org/10.1109/TIT.1968.1054102

Hurt, J. A., Thibodeau, S. A., Hirsh, A. S., Pabo, C. O., & Joung, J. K. (2003). Highly specific zinc finger proteins obtained by directed domain shuffling and cell-based selection. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(21), 12271–12276. https://doi.org/10.1073/pnas.2135381100

Hyung, D., Mallon, A.-M., Kyung, D. S., Cho, S. Y., & Seong, J. K. (2019). TarGo: network based target gene selection system for human disease related mouse models. *Laboratory Animal Research*, *35*(1), 1–7. https://doi.org/10.1186/s42826-019-0023-z

Iarovaia, O. V., Rubtsov, M., Ioudinkova, E., Tsfasman, T., Razin, S. V., & Vassetzky, Y. S. (2014). Dynamics of double strand breaks and chromosomal translocations. In *Molecular Cancer* (Vol. 13, Issue 1). BioMed Central Ltd. https://doi.org/10.1186/1476-4598-13-249

Inui, M., Miyado, M., Igarashi, M., Tamano, M., Kubo, A., Yamashita, S., Asahara, H., Fukami, M., & Takada, S. (2014). Rapid generation of mouse models with defined point mutations by the CRISPR/Cas9 system. *Scientific Reports*, *4*(5396). https://doi.org/10.1038/srep05396

Isalan, M., Choo, Y., & Klug, A. (1997). Synergy between adjacent zinc fingers in sequence-specific DNA recognition. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(11), 5617–5621. https://doi.org/10.1073/pnas.94.11.5617

Jaenisch, R. (1976). Germ line integration and Mendelian transmission of the exogenous Moloney leukemia virus. *Proceedings of the National Academy of Sciences of the United States of America*, *73*(4), 1260–1264. https://doi.org/10.1073/pnas.73.4.1260

Japkowicz, N. (2000). Learning from Imbalanced Data Sets: A Comparison of Various Strategies. *AAAI Technical Report WS-00-05*, 10–15.

Jia, S., Hu, X., & Sun, L. (2013). The Comparison between Random Forest and Support Vector Machine Algorithm for Predicting β-Hairpin Motifs in Proteins. *Engineering*, *05*(10), 391–395. https://doi.org/10.4236/eng.2013.510b079

Jiang, W., Feng, S., Huang, S., Yu, W., Li, G., Yang, G., Liu, Y., Zhang, Y., Zhang, L., Hou, Y., Chen, J., Chen, J., & Huang, X. (2018). BE-PLUS: A new base editing tool with broadened editing window and enhanced fidelity. *Cell Research*, *28*(8), 855–861. https://doi.org/10.1038/s41422-018-0052-4

Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. A., & Charpentier, E. (2012). A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science*, *337*(6096), 816–821. https://doi.org/10.1126/science.1225829

Jinek, M., Jiang, F., Taylor, D. W., Sternberg, S. H., Kaya, E., Ma, E., Anders, C., Hauer, M., Zhou, K., Lin, S., Kaplan, M., Iavarone, A. T., Charpentier, E., Nogales, E., & Doudna, J. A. (2014). Structures of Cas9 endonucleases reveal RNA-mediated conformational activation. *Science*, *343*(6176). https://doi.org/10.1126/science.1247997

Joung, J. K., & Sander, J. D. (2013). TALENs: A widely applicable technology for targeted genome editing. In *Nature Reviews Molecular Cell Biology* (Vol. 14, Issue 1, pp. 49–55). NIH Public Access. https://doi.org/10.1038/nrm3486

Justice, M. J., Siracusa, L. D., & Stewart, A. F. (2011). Technical approaches for mouse models of human disease. In *DMM Disease Models and Mechanisms* (Vol. 4, Issue 3, pp. 305–310). Company of Biologists. https://doi.org/10.1242/dmm.000901

Kaur, K., Gupta, A. K., Rajput, A., & Kumar, M. (2016). Ge-CRISPR - An integrated pipeline for the prediction and analysis of sgRNAs genome editing efficiency for CRISPR/Cas system. *Scientific Reports*, *6*(1), 1–12. https://doi.org/10.1038/srep30870

Khoury, M. J. (2003). Genetics and genomics in practice: The continuum from genetic disease to genetic information in health and disease. In *Genetics in Medicine* (Vol. 5, Issue 4, pp. 261–268). Genet Med. https://doi.org/10.1097/01.GIM.0000076977.90682.A5

Kim, D., Kim, J., Hur, J. K., Been, K. W., Yoon, S. H., & Kim, J. S. (2016). Genome-wide analysis reveals specificities of Cpf1 endonucleases in human cells. *Nature Biotechnology*, *34*(8), 863–868. https://doi.org/10.1038/nbt.3609

Kim, H. K., Min, S., Song, M., Jung, S., Choi, J. W., Kim, Y., Lee, S., Yoon, S., & Kim, H. (2018). Deep learning improves prediction of CRISPR-Cpf1 guide RNA activity. *Nature Biotechnology*, *36*(3), 239–241. https://doi.org/10.1038/nbt.4061

Kim, Y. B., Komor, A. C., Levy, J. M., Packer, M. S., Zhao, K. T., & Liu, D. R. (2017). Increasing the genome-targeting scope and precision of base editing with engineered Cas9-cytidine deaminase fusions. *Nature Biotechnology*, *35*(4), 371–376.

https://doi.org/10.1038/nbt.3803

Kim, Y., Cheong, S. A., Lee, J. G., Lee, S. W., Lee, M. S., Baek, I. J., & Sung, Y. H. (2016). Generation of knockout mice by Cpf1-mediated gene targeting. In *Nature Biotechnology* (Vol. 34, Issue 8, pp. 808–810). Nature Publishing Group. https://doi.org/10.1038/nbt.3614

Kleinstiver, B. P., Pattanayak, V., Prew, M. S., Tsai, S. Q., Nguyen, N. T., Zheng, Z., & Joung, J. K. (2016). High-fidelity CRISPR-Cas9 variants with undetectable genome-wide off-targets. *Nature*, *529*(7587), 490–495. https://doi.org/10.1038/nature16526

Kleinstiver, B. P., Tsai, S. Q., Prew, M. S., Nguyen, N. T., Welch, M. M., Lopez, J. M., McCaw, Z. R., Aryee, M. J., & Joung, J. K. (2016). Genome-wide specificities of CRISPR-Cas Cpf1 nucleases in human cells. *Nature Biotechnology*, *34*(8), 869–874. https://doi.org/10.1038/nbt.3620

Knott, G. J., & Doudna, J. A. (2018). CRISPR-Cas guides the future of genetic engineering. In *Science* (Vol. 361, Issue 6405, pp. 866–869). American Association for the Advancement of Science. https://doi.org/10.1126/science.aat5011

Komor, A. C., Kim, Y. B., Packer, M. S., Zuris, J. A., & Liu, D. R. (2016). Programmable editing of a target base in genomic DNA without double-stranded DNA cleavage. *Nature*, *533*(7603), 420–424. https://doi.org/10.1038/nature17946

Kosicki, M., Allen, F., & Bradley, A. (2020). Cas9-induced large deletions and small indels are controlled in a convergent fashion. *BioRxiv*, 2020.08.05.216739. https://doi.org/10.1101/2020.08.05.216739

Kosicki, M., Tomberg, K., & Bradley, A. (2018). Repair of double-strand breaks induced by CRISPR–Cas9 leads to large deletions and complex rearrangements. *Nature Biotechnology*, *36*(8), 765–771. https://doi.org/10.1038/nbt.4192

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International Journal of Computer Science*, *1*(1), 111–117. https://doi.org/10.1080/02331931003692557

Kuzminov, A. (2001). Single-strand interruptions in replicating chromosomes cause double-strand breaks. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(15), 8241–8246. https://doi.org/10.1073/pnas.131009198

Lander, E. S. (1996). The new genomics: Global views of biology. In *Science* (Vol. 274, Issue 5287, pp. 536–539). American Association for the Advancement of Science. https://doi.org/10.1126/science.274.5287.536

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, *9*(4), 357–359. https://doi.org/10.1038/nmeth.1923

Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, *10*(3), R25. https://doi.org/10.1186/gb-2009-10-3-r25

Lee, H. K., Smith, H. E., Liu, C., Willi, M., & Hennighausen, L. (2020). Cytosine base editor 4 but not adenine base editor generates off-target mutations in mouse embryos. *Communications Biology*, *3*(1), 1–6. https://doi.org/10.1038/s42003-019-0745-3

Lee, H. K., Willi, M., Miller, S. M., Kim, S., Liu, C., Liu, D. R., & Hennighausen, L. (2018). Targeting fidelity of adenine and cytosine base editors in mouse embryos. *Nature Communications*, *9*(1), 1–6. https://doi.org/10.1038/s41467-018-07322-7

Lee, H., & Kim, J.-S. (2018). Unexpected CRISPR on-target effects. *Nature Biotechnology*, *36*(8). https://doi.org/10.1038/nbt.4207

Leenay, R. T., Aghazadeh, A., Hiatt, J., Tse, D., Roth, T. L., Apathy, R., Shifrut, E., Hultquist,

J. F., Krogan, N., Wu, Z., Cirolia, G., Canaj, H., Leonetti, M. D., Marson, A., May, A. P., & Zou, J. (2019). Large dataset enables prediction of repair after CRISPR–Cas9 editing in primary T cells. *Nature Biotechnology*, *37*(9), 1034–1037. https://doi.org/10.1038/s41587-019-0203-2

Li, G., Zhang, X., Zhong, C., Mo, J., Quan, R., Yang, J., Liu, D., Li, Z., Yang, H., & Wu, Z. (2017). Small molecules enhance CRISPR/Cas9-mediated homology-directed genome editing in primary cells. *Scientific Reports*, *7*(1), 1–11. https://doi.org/10.1038/s41598-017-09306-x

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., & Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Li, K., Wang, G., Andersen, T., Zhou, P., & Pu, W. T. (2014). Optimization of genome engineering approaches with the CRISPR/Cas9 system. *PLoS ONE*, *9*(8). https://doi.org/10.1371/journal.pone.0105779

Liang, X., Potter, J., Kumar, S., Ravinder, N., & Chesnut, J. D. (2017). Enhanced CRISPR/Cas9-mediated precise genome editing by improved design and delivery of gRNA, Cas9 nuclease, and donor DNA. *Journal of Biotechnology*, *241*, 136–146. https://doi.org/10.1016/j.jbiotec.2016.11.011

Lin, J., & Wong, K. C. (2018). Off-target predictions in CRISPR-Cas9 gene editing using deep learning. *Bioinformatics*, *34*(17), i656–i663. https://doi.org/10.1093/bioinformatics/bty554

Lin, S., Staahl, B. T., Alla, R. K., & Doudna, J. A. (2014). Enhanced homology-directed human genome engineering by controlled timing of CRISPR/Cas9 delivery. *ELife*, *3*, e04766. https://doi.org/10.7554/eLife.04766

Ling, X., Xie, B., Gao, X., Chang, L., Zheng, W., Chen, H., Huang, Y., Tan, L., Li, M., & Liu, T. (2020). Improving the efficiency of precise genome editing with site-specific Cas9-oligonucleotide conjugates. *Science Advances*, *6*(15), eaaz0051. https://doi.org/10.1126/sciadv.aaz0051

Listgarten, J. (2017, February 18). CRISPR Gene Editing with Machine Learning. *AAAS 2017 Annual Meeting*. https://aaas.confex.com/aaas/2017/meetingapp.cgi/Paper/20738

Listgarten, J., Weinstein, M., Kleinstiver, B. P., Sousa, A. A., Joung, J. K., Crawford, J., Gao, K., Hoang, L., Elibol, M., Doench, J. G., & Fusi, N. (2018). Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. *Nature Biomedical Engineering*, *2*(1), 38–47. https://doi.org/10.1038/s41551-017-0178-6

Liu, J., Srinivasan, S., Li, C. Y., Ho, I. L., Rose, J., Shaheen, M. A., Wang, G., Yao, W., Deem, A., Bristow, C., Hart, T., & Draetta, G. (2019). Pooled library screening with multiplexed Cpf1 library. *Nature Communications*, *10*(1), 1–10. https://doi.org/10.1038/s41467-019-10963-x

Liu, Z., Chen, M., Chen, S., Deng, J., Song, Y., Lai, L., & Li, Z. (2018). Highly efficient RNA-guided base editing in rabbit. *Nature Communications*, *9*(1), 1–10. https://doi.org/10.1038/s41467-018-05232-2

Loh, W.-Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, *1*(1), 14–23. https://doi.org/10.1002/widm.8

Ma, J.-L., Kim, E. M., Haber, J. E., & Lee, S. E. (2003). Yeast Mre11 and Rad1 Proteins Define a Ku-Independent Mechanism To Repair Double-Strand Breaks Lacking Overlapping End Sequences. *Molecular and Cellular Biology*, *23*(23), 8820–8828. https://doi.org/10.1128/mcb.23.23.8820-8828.2003

Ma, M., Zhuang, F., Hu, X., Wang, B., Wen, X. Z., Ji, J. F., & Xi, J. J. (2017). Efficient

generation of mice carrying homozygous double-floxp alleles using the Cas9-Avidin/Biotin-donor DNA system. In *Cell Research* (Vol. 27, Issue 4, pp. 578–581). Nature Publishing Group. https://doi.org/10.1038/cr.2017.29

Ma, Y., Yu, L., Zhang, X., Xin, C., Huang, S., Bai, L., Chen, W., Gao, R., Li, J., Pan, S., Qi, X., Huang, X., & Zhang, L. (2018). Highly efficient and precise base editing by engineered dCas9-guide tRNA adenosine deaminase in rats. *Cell Discovery*, *4*(1), 1–4. https://doi.org/10.1038/s41421-018-0047-9

Mali, P., Yang, L., Esvelt, K. M., Aach, J., Guell, M., DiCarlo, J. E., Norville, J. E., & Church, G. M. (2013). RNA-guided human genome engineering via Cas9. *Science*, *339*(6121), 823–826. https://doi.org/10.1126/science.1232033

Mao, Z., Bozzella, M., Seluanov, A., & Gorbunova, V. (2008). Comparison of nonhomologous end joining and homologous recombination in human cells. *DNA Repair*, *7*(10), 1765–1771. https://doi.org/10.1016/j.dnarep.2008.06.018

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference (SciPy 2010)*.

Miyaoka, Y., Berman, J. R., Cooper, S. B., Mayerl, S. J., Chan, A. H., Zhang, B., Karlin-Neumann, G. A., & Conklin, B. R. (2016). Systematic quantification of HDR and NHEJ reveals effects of locus, nuclease, and cell type on genome-editing. *Scientific Reports*, *6*(1), 1–12. https://doi.org/10.1038/srep23549

Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., & Van Valen, D. (2019). Deep learning for cellular image analysis. In *Nature Methods* (Vol. 16, Issue 12, pp. 1233–1246). Nature Research. https://doi.org/10.1038/s41592-019-0403-1

Mojica, F. J. M., Díez-Villaseñor, C., García-Martínez, J., & Almendros, C. (2009). Short motif sequences determine the targets of the prokaryotic CRISPR defence system. *Microbiology*, *155*(3), 733–740. https://doi.org/10.1099/mic.0.023960-0

Montori, V. M., Devereaux, P. J., Adhikari, N. K. J., Burns, K. E. A., Eggert, C. H., Briel, M., Lacchetti, C., Leung, T. W., Darling, E., Bryant, D. M., Bucher, H. C., Schünemann, H. J., Meade, M. O., Cook, D. J., Erwin, P. J., Sood, A., Sood, R., Lo, B., Thompson, C. A., … Guyatt, G. H. (2005). Randomized trials stopped early for benefit: A systematic review. In *Journal of the American Medical Association* (Vol. 294, Issue 17, pp. 2203–2209). JAMA. https://doi.org/10.1001/jama.294.17.2203

Moreno-Mateos, M. A., Vejnar, C. E., Beaudoin, J. D., Fernandez, J. P., Mis, E. K., Khokha, M. K., & Giraldez, A. J. (2015). CRISPRscan: Designing highly efficient sgRNAs for CRISPR-Cas9 targeting in vivo. *Nature Methods*, *12*(10), 982–988. https://doi.org/10.1038/nmeth.3543

Moscou, M. J., & Bogdanove, A. J. (2009). A simple cipher governs DNA recognition by TAL effectors. *Science*, *326*(5959), 1501. https://doi.org/10.1126/science.1178817

Newman, A., Starrs, L., & Burgio, G. (2020). Cas9 Cuts and Consequences; Detecting, Predicting, and Mitigating CRISPR/Cas9 On- and Off-Target Damage. *BioEssays*, 2000047. https://doi.org/10.1002/bies.202000047

Ng, A. Y. (2004). Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the 21 St International Conference on Machine Learning*.

Nishida, K., Arazoe, T., Yachie, N., Banno, S., Kakimoto, M., Tabata, M., Mochizuki, M., Miyabe, A., Araki, M., Hara, K. Y., Shimatani, Z., & Kondo, A. (2016). Targeted nucleotide editing using hybrid prokaryotic and vertebrate adaptive immune systems. *Science*, *353*(6305). https://doi.org/10.1126/science.aaf8729

Nishimasu, H., Ran, F. A., Hsu, P. D., Konermann, S., Shehata, S. I., Dohmae, N., Ishitani, R., Zhang, F., & Nureki, O. (2014). Crystal structure of Cas9 in complex with guide RNA

and target DNA. *Cell*, *156*(5), 935–949. https://doi.org/10.1016/j.cell.2014.02.001

O'Brien, A., & Bailey, T. L. (2014). GT-Scan: identifying unique genomic targets. In *Bioinformatics* (pp. 1–3). https://doi.org/10.1093/bioinformatics/btu354

O'Brien, A. R., Wilson, L. O. W., Burgio, G., & Bauer, D. C. (2019). Unlocking HDR-mediated nucleotide editing by identifying high-efficiency target sites using machine learning. *Scientific Reports*, *9*(1), 1–10. https://doi.org/10.1038/s41598-019-39142-0

O'Gorman, S., Dagenais, N. A., Qian, M., & Marchuk, Y. (1997). Protamine-Cre recombinase transgenes efficiently recombine target sequences in the male germ line of mice, but not in embryonic stem cells. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(26), 14602–14607. https://doi.org/10.1073/pnas.94.26.14602

Pâques, F., & Haber, J. E. (1999). Multiple Pathways of Recombination Induced by Double-Strand Breaks in Saccharomyces cerevisiae. *Microbiology and Molecular Biology Reviews*, *63*(2), 349–404. https://doi.org/10.1128/mmbr.63.2.349-404.1999

Pardo, B., Gómez-González, B., & Aguilera, A. (2009). DNA Repair in Mammalian Cells: DNA Double-Strand Break Repair: How to Fix a Broken Relationship. *Cellular and Molecular Life Sciences*, *66*(6), 1039–1056. https://doi.org/10.1007/s00018-009-8740-3

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net

Peng, H., Zheng, Y., Blumenstein, M., Tao, D., & Li, J. (2018). CRISPR/Cas9 cleavage efficiency regression through boosting algorithms and Markov sequence profiling. *Bioinformatics*, *34*(18), 3069–3077. https://doi.org/10.1093/bioinformatics/bty298

Plessis, A., Perrin, A., Haber, J. E., & Dujon, B. (1992). Site-Specific Recombination Determined by I-SceI, a Mitochondrial Group I Intron-Encoded Endonuclease Expressed in the Yeast Nucleus. *Genetics*, *130*(3), 451–460. /pmc/articles/PMC1204864/?report=abstract

Porteus, M. H., & Carroll, D. (2005). Gene targeting using zinc finger nucleases. In *Nature Biotechnology* (Vol. 23, Issue 8, pp. 967–973). Nature Publishing Group. https://doi.org/10.1038/nbt1125

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Baltic Journal of Modern Computing*, *5*(2). https://doi.org/10.22364/bjmc.2017.5.2.05

Qi, L. S., Larson, M. H., Gilbert, L. A., Doudna, J. A., Weissman, J. S., Arkin, A. P., & Lim, W. A. (2013). Repurposing CRISPR as an RNA-guided platform for sequence-specific control of gene expression. *Cell*, *152*(5), 1173–1183. https://doi.org/10.1016/j.cell.2013.02.022

Rahman, M. K., & Rahman, M. S. (2017). CRISPRpred: A flexible and efficient tool for sgRNAs on-target activity prediction in CRISPR/Cas9 systems. *PLOS ONE*, *12*(8), e0181943. https://doi.org/10.1371/journal.pone.0181943

Ramirez, C. L., Foley, J. E., Wright, D. A., Müller-Lerch, F., Rahman, S. H., Cornu, T. I., Winfrey, R. J., Sander, J. D., Fu, F., Townsend, J. A., Cathomen, T., Voytas, D. F., & Joung, J. K. (2008). Unexpected failure rates for modular assembly of engineered zinc fingers. In *Nature Methods* (Vol. 5, Issue 5, pp. 374–375). Nature Publishing Group. https://doi.org/10.1038/nmeth0508-374

Ransburgh, D. J. R., Chiba, N., Ishioka, C., Toland, A. E., & Parvin, J. D. (2010). Identification of breast tumor mutations in BRCA1 that abolish its function in homologous DNA recombination. *Cancer Research*, *70*(3), 988–995. https://doi.org/10.1158/0008-5472.CAN-09-2850

Renaud, J. B., Boix, C., Charpentier, M., De Cian, A., Cochennec, J., Duvernois-Berthet, E., Perrouault, L., Tesson, L., Edouard, J., Thinard, R., Cherifi, Y., Menoret, S., Fontanière, S., de Crozé, N., Fraichard, A., Sohm, F., Anegon, I., Concordet, J. P., & Giovannangeli, C. (2016). Improved Genome Editing Efficiency and Flexibility Using Modified Oligonucleotides with TALEN and CRISPR-Cas9 Nucleases. *Cell Reports*, *14*(9), 2263–2272. https://doi.org/10.1016/j.celrep.2016.02.018

Richardson, C. D., Ray, G. J., DeWitt, M. A., Curie, G. L., & Corn, J. E. (2016). Enhancing homology-directed genome editing by catalytically active and inactive CRISPR-Cas9 using asymmetric donor DNA. *Nature Biotechnology*, *34*(3), 339–344. https://doi.org/10.1038/nbt.3481

Roberts, R. J., Vincze, T., Posfai, J., & Macelis, D. (2007). REBASE—enzymes and genes for DNA restriction and modification. *Nucleic Acids Research*, *35*, D269–D270. https://doi.org/10.1093/nar/gkl891

Romiguier, J., Ranwez, V., Douzery, E. J. P., & Galtier, N. (2010). Contrasting GC-content dynamics across 33 mammalian genomes: Relationship with life-history traits and chromosome sizes. *Genome Research*, *20*(8), 1001–1009. https://doi.org/10.1101/gr.104372.109

Rosenthal, N., & Brown, S. (2007). The mouse ascending: Perspectives for human-disease models. In *Nature Cell Biology* (Vol. 9, Issue 9, pp. 993–999). Nature Publishing Group. https://doi.org/10.1038/ncb437

Rouet, P., Smih, F., & Jasin, M. (1994). Introduction of double-strand breaks into the genome of mouse cells by expression of a rare-cutting endonuclease. *Molecular and Cellular Biology*, *14*(12), 8096–8106. https://doi.org/10.1128/mcb.14.12.8096

Rutkauskas, M., Sinkunas, T., Songailiene, I., Tikhomirova, M. S., Siksnys, V., & Seidel, R. (2015). Directional R-loop formation by the CRISPR-cas surveillance complex cascade provides efficient off-target site rejection. *Cell Reports*, *10*(9), 1534–1543. https://doi.org/10.1016/j.celrep.2015.01.067

Ryu, S. M., Koo, T., Kim, K., Lim, K., Baek, G., Kim, S. T., Kim, H. S., Kim, D. E., Lee, H., Chung, E., & Kim, J. S. (2018). Adenine base editing in mouse embryos and an adult mouse model of Duchenne muscular dystrophy. *Nature Biotechnology*, *36*(6), 536–539. https://doi.org/10.1038/nbt.4148

Salman, R., & Kecman, V. (2012). Regression as classification. *2012 Proceedings of IEEE Southeastcon*. https://doi.org/10.1109/SECon.2012.6196887

Sargent, R. G., Brenneman, M. A., & Wilson, J. H. (1997). Repair of site-specific double-strand breaks in a mammalian chromosome by homologous and illegitimate recombination. *Molecular and Cellular Biology*, *17*(1), 267–277. https://doi.org/10.1128/mcb.17.1.267

Sauer, B. (1998). Inducible gene targeting in mice using the Cre/lox system. *Methods: A Companion to Methods in Enzymology*, *14*(4), 381–392. https://doi.org/10.1006/meth.1998.0593

Schwenk, F., Kühn, R., Angrand, P. O., Rajewsky, K., & Stewart, A. F. (1998). Temporally and spatially regulated somatic mutagenesis in mice. *Nucleic Acids Research*, *26*(6), 1427–1432. https://doi.org/10.1093/nar/26.6.1427

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python. *Proceedings of the 9th Python in Science Conference*, 92–96.

https://doi.org/10.25080/majora-92bf1922-011

Sfeir, A., & Symington, L. S. (2015). Microhomology-Mediated End Joining: A Back-up Survival Mechanism or Dedicated Pathway? In *Trends in Biochemical Sciences* (Vol. 40, Issue 11, pp. 701–714). Elsevier Ltd. https://doi.org/10.1016/j.tibs.2015.08.006

Shah, S. A., Erdmann, S., Mojica, F. J. M., & Garrett, R. A. (2013). Protospacer recognition motifs: Mixed identities and functional diversity. In *RNA Biology* (Vol. 10, Issue 5, pp. 891–899). Taylor and Francis Inc. https://doi.org/10.4161/rna.23764

Sharma, S., & Raghavan, S. C. (2016). Nonhomologous DNA End Joining. In *Encyclopedia of Cell Biology*. Elsevier. https://doi.org/10.1016/B978-0-12-394447-4.10047-1

Shen, M. W., Arbab, M., Hsu, J. Y., Worstell, D., Culbertson, S. J., Krabbe, O., Cassa, C. A., Liu, D. R., Gifford, D. K., & Sherwood, R. I. (2018). Predictable and precise template-free CRISPR editing of pathogenic variants. *Nature*, *563*(7733), 646–651. https://doi.org/10.1038/s41586-018-0686-x

Shy, B. R., Macdougall, M. S., Clarke, R., & Merrill, B. J. (2016). Co-incident insertion enables high efficiency genome engineering in mouse embryonic stem cells. *Nucleic Acids Research*, *44*(16), 7997–8010. https://doi.org/10.1093/nar/gkw685

Simmons, D. (2008). The Use of Animal Models in Studying Genetic Disease. *Nature Education*, *2008*, 1–10.

Smith, C. L., Blake, J. A., Kadin, J. A., Richardson, J. E., & Bult, C. J. (2018). Mouse Genome Database (MGD)-2018: Knowledgebase for the laboratory mouse. *Nucleic Acids Research*, *46*(D1), D836–D842. https://doi.org/10.1093/nar/gkx1006

Smith, K. R. (2012). Gene Therapy: The Potential Applicability of Gene Transfer Technology to the Human Germline. *International Journal of Medical Sciences*, *1*(2), 76–91. https://doi.org/10.7150/ijms.1.76

Stark, J. M., Pierce, A. J., Oh, J., Pastink, A., & Jasin, M. (2004). Genetic Steps of Mammalian Homologous Repair with Distinct Mutagenic Consequences. *Molecular and Cellular Biology*, *24*(21), 9305–9316. https://doi.org/10.1128/mcb.24.21.9305-9316.2004

Stemmer, M., Thumberger, T., Del Sol Keyer, M., Wittbrodt, J., & Mateo, J. L. (2015). CCTop: An Intuitive, Flexible and Reliable CRISPR/Cas9 Target Prediction Tool. *PLoS ONE*, *10*(4), e0124633. https://doi.org/10.1371/journal.pone.0124633

Sternberg, N., & Hamilton, D. (1981). Bacteriophage P1 site-specific recombination. I. Recombination between loxP sites. *Journal of Molecular Biology*, *150*(4), 467–486. https://doi.org/10.1016/0022-2836(81)90375-2

Sternberg, S. H., Redding, S., Jinek, M., Greene, E. C., & Doudna, J. A. (2014). DNA interrogation by the CRISPR RNA-guided endonuclease Cas9. *Nature*, *507*(7490), 62–67. https://doi.org/10.1038/nature13011

Stormo, G. D., Schneider, T. D., Gold, L., & Ehrenfeucht, A. (1982). Use of the “perceptron” algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Research*, *10*(9), 2997–3011. https://doi.org/10.1093/nar/10.9.2997

Strecker, J., Ladha, A., Gardner, Z., Schmid-Burgk, J. L., Makarova, K. S., Koonin, E. V., & Zhang, F. (2019). RNA-guided DNA insertion with CRISPR-associated transposases. *Science*, *364*(6448), 48–53. https://doi.org/10.1126/science.aax9181

Strohkendl, I., Saifuddin, F. A., Rybarski, J. R., Finkelstein, I. J., & Russell, R. (2018). Kinetic Basis for DNA Target Specificity of CRISPR-Cas12a. *Molecular Cell*, *71*(5), 816-824.e3. https://doi.org/10.1016/j.molcel.2018.06.043

Swarts, D. C. (2019). Making the cut(s): How Cas12a cleaves target and non-target DNA. In

Biochemical Society Transactions (Vol. 47, Issue 5, pp. 1499–1510). Portland Press Ltd. https://doi.org/10.1042/BST20190564

Swarts, D. C., & Jinek, M. (2019). Mechanistic Insights into the cis- and trans-Acting DNase Activities of Cas12a. *Molecular Cell*, *73*(3), 589-600.e4. https://doi.org/10.1016/j.molcel.2018.11.021

Swarts, D. C., van der Oost, J., & Jinek, M. (2017). Structural Basis for Guide RNA Processing and Seed-Dependent DNA Targeting by CRISPR-Cas12a. *Molecular Cell*, *66*(2), 221-233.e4. https://doi.org/10.1016/j.molcel.2017.03.016

Symington, L. S., & Gautier, J. (2011). Double-Strand Break End Resection and Repair Pathway Choice. *Annual Review of Genetics*, *45*(1), 247–271. https://doi.org/10.1146/annurev-genet-110410-132435

Tarca, A. L., Carey, V. J., Chen, X., Romero, R., & Drăghici, S. (2007). Machine Learning and Its Applications to Biology. *PLoS Computational Biology*, *3*(6), e116. https://doi.org/10.1371/journal.pcbi.0030116

Taylor, E. M., Cecillon, S. M., Bonis, A., Ross Chapman, J., Povirk, L. F., & Lindsay, H. D. (2009). The Mre11/Rad50/Nbs1 complex functions in resection-based DNA end joining in Xenopus laevis. *Nucleic Acids Research*, *38*(2), 441–454. https://doi.org/10.1093/nar/gkp905

Terns, M. P., & Terns, R. M. (2011). CRISPR-based adaptive immune systems. In *Current Opinion in Microbiology* (Vol. 14, Issue 3, pp. 321–327). Curr Opin Microbiol. https://doi.org/10.1016/j.mib.2011.03.005

The ENCODE Project Consortium. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, *489*(7414), 57–74. https://doi.org/10.1038/nature11247

The pandas development team. (2020). pandas-dev/pandas: Pandas. *Zenodo*. https://doi.org/10.5281/ZENODO.3715232

Thomas, K. R., Folger, K. R., & Capecchi, M. R. (1986). High frequency targeting of genes to specific sites in the mammalian genome. *Cell*, *44*(3), 419–428. https://doi.org/10.1016/0092-8674(86)90463-0

Trunk, G. V. (1979). A Problem of Dimensionality: A Simple Example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(3), 306–307. https://doi.org/10.1109/TPAMI.1979.4766926

Tsai, S. Q., Zheng, Z., Nguyen, N. T., Liebers, M., Topkar, V. V., Thapar, V., Wyvekens, N., Khayter, C., Iafrate, A. J., Le, L. P., Aryee, M. J., & Joung, J. K. (2015). GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. *Nature Biotechnology*, *33*(2), 187–198. https://doi.org/10.1038/nbt.3117

Tsien, J. Z., Chen, D. F., Gerber, D., Tom, C., Mercer, E. H., Anderson, D. J., Mayford, M., Kandel, E. R., & Tonegawa, S. (1996). Subregion- and cell type-restricted gene knockout in mouse brain. *Cell*, *87*(7), 1317–1326. https://doi.org/10.1016/S0092-8674(00)81826-7

Urnov, F. D., Miller, J. C., Lee, Y.-L., Beausejour, C. M., Rock, J. M., Augustus, S., Jamieson, A. C., Porteus, M. H., Gregory, P. D., & Holmes, M. C. (2005). Highly efficient endogenous human gene correction using designed zinc-finger nucleases. *Nature*, *435*(7042), 646–651. https://doi.org/10.1038/nature03556

Uusi-Mäkelä, M. I. E., Barker, H. R., Bäuerlein, C. A., Häkkinen, T., Nykter, M., & Rämet, M. (2018). Chromatin accessibility is associated with CRISPR-Cas9 efficiency in the zebrafish (Danio rerio). *PLoS ONE*, *13*(4), e0196238. https://doi.org/10.1371/journal.pone.0196238

Valerie, K., & Povirk, L. F. (2003). Regulation and mechanisms of mammalian double-strand

break repair. In *Oncogene* (Vol. 22, Issue 37 REV. ISS. 3, pp. 5792–5812). Nature Publishing Group. https://doi.org/10.1038/sj.onc.1206679

Vandamme, T. (2014). Use of rodents as models of human diseases. In *Journal of Pharmacy and Bioallied Sciences* (Vol. 6, Issue 1, pp. 2–9). Medknow Publications. https://doi.org/10.4103/0975-7406.124301

Wang, B., Li, K., Wang, A., Reiser, M., Saunders, T., Lockey, R. F., & Wang, J. W. (2015). Highly efficient CRISPR/HDR-mediated knock-in for mouse embryonic stem cells and zygotes. *BioTechniques*, *59*(4), 201–208. https://doi.org/10.2144/000114339

Wang, H., Rosner, G. L., & Goodman, S. N. (2016). Quantifying over-estimation in early stopped clinical trials and the "freezing effect" on subsequent research. *Clinical Trials*, *13*(6), 621–631. https://doi.org/10.1177/1740774516649595

Wang, J., Xiang, X., Bolund, L., Zhang, X., Cheng, L., & Luo, Y. (2020). GNL-Scorer: A generalized model for predicting CRISPR on-target activity by machine learning and featurization. *Journal of Molecular Cell Biology*. https://doi.org/10.1093/jmcb/mjz116

Wang, K., Tang, X., Liu, Y., Xie, Z., Zou, X., Li, M., Yuan, H., Ouyang, H., Jiao, H., & Pang, D. (2016). Efficient Generation of Orthologous Point Mutations in Pigs via CRISPR-assisted ssODN-mediated Homology-directed Repair. *Molecular Therapy - Nucleic Acids*, *5*(11), e396. https://doi.org/10.1038/mtna.2016.101

Wang, T., Wei, J. J., Sabatini, D. M., & Lander, E. S. (2014). Genetic screens in human cells using the CRISPR-Cas9 system. *Science*, *343*(6166), 80–84. https://doi.org/10.1126/science.1246981

Waskom, M., Botvinnik, O., Gelbart, M., Ostblom, J., Hobson, P., Lukauskas, S., Gemperline, D. C., Augspurger, T., Halchenko, Y., Warmenhoven, J., Cole, J. B., Ruiter, J. de, Vanderplas, J., Hoyer, S., Pye, C., Miles, A., Swain, C., Meyer, K., Martin, M., … Brunner, T. (2020). *mwaskom/seaborn: v0.11.0 (Sepetmber 2020)*. https://doi.org/10.5281/ZENODO.4019146

Waterston, R. H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J. F., Agarwal, P., Agarwala, R., Ainscough, R., Alexandersson, M., An, P., Antonarakis, S. E., Attwood, J., Baertsch, R., Bailey, J., Barlow, K., Beck, S., Berry, E., Birren, B., Bloom, T., … Lander, E. S. (2002). Initial sequencing and comparative analysis of the mouse genome. *Nature*, *420*(6915), 520–562. https://doi.org/10.1038/nature01262

Welch, B. L. (1947). The generalisation of student's problems when several different population variances are involved. *Biometrika*, *34*(1–2), 28–35. https://doi.org/10.1093/biomet/34.1-2.28

Westra, E. R., Semenova, E., Datsenko, K. A., Jackson, R. N., Wiedenheft, B., Severinov, K., & Brouns, S. J. J. (2013). Type I-E CRISPR-Cas Systems Discriminate Target from Non-Target DNA through Base Pairing-Independent PAM Recognition. *PLoS Genetics*, *9*(9). https://doi.org/10.1371/journal.pgen.1003742

Wienert, B., Wyman, S. K., Richardson, C. D., Yeh, C. D., Akcakaya, P., Porritt, M. J., Morlock, M., Vu, J. T., Kazane, K. R., Watry, H. L., Judge, L. M., Conklin, B. R., Maresca, M., & Corn, J. E. (2019). Unbiased detection of CRISPR off-targets in vivo using DISCOVER-Seq. *Science*, *364*(6437), 286–289. https://doi.org/10.1126/science.aav9023

Wilson, L. O. W., O'Brien, A. R., & Bauer, D. C. (2018). The current state and future of CRISPR-Cas9 gRNA design tools. *Frontiers in Pharmacology*, *9*(JUN). https://doi.org/10.3389/fphar.2018.00749

Wilson, L. O. W., Reti, D., O'Brien, A. R., Dunne, R. A., & Bauer, D. C. (2018). High Activity Target-Site Identification Using Phenotypic Independent CRISPR-Cas9 Core Functionality. *The CRISPR Journal*, *1*(2), 182–190.

https://doi.org/10.1089/crispr.2017.0021

Wong, N., Liu, W., & Wang, X. (2015). WU-CRISPR: Characteristics of functional guide RNAs for the CRISPR/Cas9 system. *Genome Biology*, *16*(1), 218. https://doi.org/10.1186/s13059-015-0784-0

Wright, D. A., Thibodeau-Beganny, S., Sander, J. D., Winfrey, R. J., Hirsh, A. S., Eichtinger, M., Fu, F., Porteus, M. H., Dobbs, D., Voytas, D. F., & Joung, J. K. (2006). Standardized reagents and protocols for engineering zinc finger nucleases by modular assembly. *Nature Protocols*, *1*(3), 1637–1652. https://doi.org/10.1038/nprot.2006.259

Wright, E. M., & Bellman, R. (1962). Adaptive Control Processes: A Guided Tour. *The Mathematical Gazette*, *46*(356), 160. https://doi.org/10.2307/3611672

Xie, K., Minkenberg, B., & Yang, Y. (2015). Boosting CRISPR/Cas9 multiplex editing capability with the endogenous tRNA-processing system. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(11), 3570–3575. https://doi.org/10.1073/pnas.1420294112

Xu, H., Xiao, T., Chen, C. H., Li, W., Meyer, C. A., Wu, Q., Wu, D., Cong, L., Zhang, F., Liu, J. S., Brown, M., & Liu, X. S. (2015). Sequence determinants of improved CRISPR sgRNA design. *Genome Research*, *25*(8), 1147–1157. https://doi.org/10.1101/gr.191452.115

Xue, L., Tang, B., Chen, W., & Luo, J. (2019). Prediction of CRISPR sgRNA Activity Using a Deep Convolutional Neural Network. *Journal of Chemical Information and Modeling*, *59*(1), 615–624. https://doi.org/10.1021/acs.jcim.8b00368

Yan, J., Chuai, G., Zhou, C., Zhu, C., Yang, J., Zhang, C., Gu, F., Xu, H., Wei, J., & Liu, Q. (2018). Benchmarking CRISPR on-target sgRNA design. *Briefings in Bioinformatics*, *19*(4), 721–724. https://doi.org/10.1093/bib/bbx001

Yang, B., Sugio, A., & White, F. F. (2006). Os8N3 is a host disease-susceptibility gene for bacterial blight of rice. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(27), 10503–10508. https://doi.org/10.1073/pnas.0604088103

Yang, H., Wang, H., Shivalila, C. S., Cheng, A. W., Shi, L., & Jaenisch, R. (2013). One-step generation of mice carrying reporter and conditional alleles by CRISPR/cas-mediated genome engineering. *Cell*, *154*(6), 1370. https://doi.org/10.1016/j.cell.2013.08.022

Yann LeCun, Y. B. (1995). Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks* (Vol. 3361, Issue 10, p. 1995). MIT Press. https://nyuscholars.nyu.edu/en/publications/convolutional-networks-for-images-speech-and-time-series

Yen, S. T., Zhang, M., Deng, J. M., Usman, S. J., Smith, C. N., Parker-Thornburg, J., Swinton, P. G., Martin, J. F., & Behringer, R. R. (2014). Somatic mosaicism and allele complexity induced by CRISPR/Cas9 RNA injections in mouse zygotes. *Developmental Biology*, *393*(1), 3–9. https://doi.org/10.1016/j.ydbio.2014.06.017

Yoshimi, K., Kunihiro, Y., Kaneko, T., Nagahora, H., Voigt, B., & Mashimo, T. (2016). SsODN-mediated knock-in with CRISPR-Cas for large genomic regions in zygotes. *Nature Communications*, *7*(1), 1–10. https://doi.org/10.1038/ncomms10431

Zeng, Y., Li, J., Li, G., Huang, S., Yu, W., Zhang, Y., Chen, D., Chen, J., Liu, J., & Huang, X. (2018). Correction of the Marfan Syndrome Pathogenic FBN1 Mutation by Base Editing in Human Cells and Heterozygous Embryos. *Molecular Therapy*, *26*(11), 2631–2637. https://doi.org/10.1016/j.ymthe.2018.08.007

Zetsche, B., Gootenberg, J. S., Abudayyeh, O. O., Slaymaker, I. M., Makarova, K. S., Essletzbichler, P., Volz, S. E., Joung, J., Van Der Oost, J., Regev, A., Koonin, E. V., & Zhang, F. (2015). Cpf1 Is a Single RNA-Guided Endonuclease of a Class 2 CRISPR-

Cas System. *Cell*, *163*(3), 759–771. https://doi.org/10.1016/j.cell.2015.09.038

Zhang, X. H., Tee, L. Y., Wang, X. G., Huang, Q. S., & Yang, S. H. (2015). Off-target effects in CRISPR/Cas9-mediated genome engineering. In *Molecular Therapy - Nucleic Acids* (Vol. 4, Issue 11, p. e264). Nature Publishing Group. https://doi.org/10.1038/mtna.2015.37

Zhang, Y., Ge, X., Yang, F., Zhang, L., Zheng, J., Tan, X., Jin, Z. B., Qu, J., & Gu, F. (2014). Comparison of non-canonical PAMs for CRISPR/Cas9-mediated DNA cleavage in human cells. *Scientific Reports*, *4*(1), 1–5. https://doi.org/10.1038/srep05405

Zheng, K., Wang, Y., Li, N., Jiang, F. F., Wu, C. X., Liu, F., Chen, H. C., & Liu, Z. F. (2018). Highly efficient base editing in bacteria using a Cas9-cytidine deaminase fusion. *Communications Biology*, *1*(1), 1–6. https://doi.org/10.1038/s42003-018-0035-5

Zhu, H., Liang, C., & Hancock, J. (2019). CRISPR-DT: designing gRNAs for the CRISPR-Cpf1 system with improved target efficiency and specificity. *Bioinformatics*, *35*(16), 2783–2789. https://doi.org/10.1093/bioinformatics/bty1061

Appendix

Name	SRP	Experiment	Samples		Cell line	Type	Source	Citation
mouse	N/A	HDR	30	Embryo				
Endo-Cas9	SRP150719	NHEJ	124	Endogenous	HEK293T	treated	SRR7352858	Kim et al., 2019
						control	SRR7352859	
HT 1-1	SRP107920		15,000	Synthetic	N/A	treated	SRR6058546	Kim et al., 2018
						control	SRR6058545	
HCT-plasmid			66	Endogenous	HCT116	treated	SRR6058554	
						control	SRR6058553	
HEK-plasmid			55	Endogenous	HEK293T	treated	SRR6058552	
						control	SRR6058551	
HEK-lenti			148	Synthetic and endogenous		treated	SRR6058550	
						control	SRR6058549	
Mini-human	SRP181683		2061	Endogenous	K-562	reference	SRR8479041/MonoRef.part1	Liu et al., 2019
							SRR8479029/MonoRef.part2	
						week1	SRR8479041/MonoRep1Week1.part1	
							SRR8479029/MonoRep1Week1.part2	
						week2	SRR8479041/MonoRep1Week2.part1	
							SRR8479029/MonoRep1Week2.part2	
						week3	SRR8479044/MonoRep1Week3.part1	
							SRR8479028/MonoRep1Week3.part2	
						week4	SRR8479044/MonoRep1Week4.part1	
							SRR8479028/MonoRep1Week4.part2	
Cas9-OT	SRP181683	GUIDE-Seq	8	Endogenous	U2OS		Supplementary from original paper	Kleinstiver et al., 2016
Cas12a-OT	SRP075607		18	Endogenous	U2OS		Supplementary from original paper	Kleinstiver et al., 2016

Supplementary Table 1 – this table summarises the datasets used in this thesis.