Structure-aware and interpretable machine learning for CRISPR-Cas9 cleavage activity prediction

Jeffrey Kelvin Mak
Keble College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2025

Abstract

The approval of the *ex vivo* CRISPR-based gene therapy Casgevy in 2023 shows the potential of CRISPR-based cures for genetic diseases. However, off-target effects induced by heteroduplex mismatches tolerated by Cas9 nucleases limit clinical adoption of the genome editing technology. Despite the advances in developing accurate CRISPR-Cas9 cleavage activity tools and determining factors influencing Cas9 cleavage activity, most tools remain confined to features relating to the spacer-target interface. To move beyond spacer-target sequence features, the first part of the thesis questions the utility of the four gold standard experimental epigenetic features — DNase I, CTCF, RRBS and H3K4me3 — in deep learning-based SpCas9 cleavage activity tools like DeepCRISPR. Considering a total of 13 computed nucleosome organization-related scores and 6 experimental epigenetic features available in the SpCas9 off-target database crisprSQL, we show that Block Decomposition Method-based scores correlate stronger with SpCas9 activity compared to the experimental epigenetic features. Looking at NuPoP (Affinity) and Nucleotide BDM within the correlation and SHAP importance analyses, we conclude that it is nucleosome positioning rather than nucleosome occupancy which inhibit SpCas9 cleavage. Finally, we suggest GC147, NuPoP (Affinity) and Nucleotide BDM as important computed nucleosome organization-related scores to be used when building a full deep learning-based SpCas9 cleavage activity model which also incorporates sequence, epigenetic and R-loop formation energy features.

Having established that primary chromatin structure influences SpCas9 activity, we next turn to the Cas9 nuclease and ask whether the protein structure can be leveraged for SpCas9 cleavage activity prediction. To achieve this, we exploit the information-rich internal protein 3D nanoenvironment surrounding the sgRNA-target heteroduplex by reframing the computational problem to map between the nanoenvironment and cleavage activity rather than between the guide-target pair and cleavage activity. By determining the most relevant residue-level features for CRISPR-Cas9 off-target cleavage activity, we developed STING_CRISPR, a machine learning model delivering accurate predictive performance of off-target cleavage activity for the type of single-base mutations considered in this study. By interpreting STING_CRISPR, we identified four important Cas9 residue spatial hotspots and associated structural/physico-chemical descriptor classes influencing CRISPR-Cas9 (off-)target cleavage activity for the sgRNA-target strand pairs covered in this study.

Nonetheless, our approach for building STING_CRISPR suffers from the need to perform all-atom molecular dynamics for every guide-target pair in the dataset, which is computationally expensive. Additionally, the majority of such tools remain limited to predictions for one or few Cas9 variants, making it difficult to quantify the effects of Cas9 residues on cleavage activity. To bridge the gap, we introduce 4 interpretable DeepEmbCas9 models for the cleavage activity prediction of 40 Cas9 variants — DeepEmbCas9, DeepEmbCas9-MVE, DeepEnsEmbCas9_naive, and DeepEnsEmbCas9 — leveraging protein and RNA

language model embeddings to encode Cas9 and sgRNA, respectively. Among the 4 neural network models, DeepEnsEmbCas9_naive performed the best in both in-distribution and out-of-distribution settings, where DeepEnsEmbCas9_naive outperformed individual Cas9 cleavage activity prediction tools on 18 out of 51 and 17 out of 48 benchmark test sets, respectively, and performed comparably otherwise. Concerning uncertainty quantification, DeepEnsEmbCas9 yields quantile-calibrated uncertainty estimates while keeping a minimal performance drop compared to DeepEnsEmbCas9_naive. SHAP importance analysis on DeepEmbCas9 reaffirms the importance of Cas9-target PAM binding as a first step for Cas9 cleavage, and reveals the L2 linker and PLL-WED-PI as important Cas9 domains modulating DeepEmbCas9's predicted activity change when introducing increased-fidelity and PAM-altering Cas9 mutations, respectively. Our findings demonstrate the usefulness of protein language model embeddings in uncertainty-aware Cas9 cleavage activity prediction. More generally, DeepEmbCas9 models serve as an initial step towards cleavage activity prediction modelling for the whole Cas9 protein family.

In summary, this collection of work paves the way forward for the next generation of structure-aware and interpretable machine and deep-learning models for CRISPR-Cas9 cleavage activity prediction.

Acknowledgements

> "A course is identified by its course number and the department where it is given."
>
> *a much-overlooked sentence from Databases practical 1*

This work would not have been possible without the support of many people and communities in the past six years — all of whom I am grateful for.

To begin, I thank the Department of Computer Science for the 5-year graduate teaching assistantship. As a former teaching assistant, I extend my gratitude to members of the practical teaching group, the 4th year/graduate/postdoc volunteer practical demonstrators, and the countless students that I have signed off over the years for a welcoming, supportive, enjoyable, and memorable teaching experience at Oxford. Shoutouts to whoever replenishes the whiteboards with new whiteboard markers, and IT support too for hosting the server required for the Databases practicals. As a former stipendiary lecturer at Keble, many thanks to Prof. Alfonso Bueno-Orovio for supporting my lectureship, and thanks to my wonderful first and second year students AB, AT, IB, MB, MC, MM, MW, OO, RB, SN, TT, WF, and WO for engaging in the tutorials and revision classes that I held at various teaching and seminar rooms.

As a graduate researcher, I am most grateful to my supervisor, Prof. Peter Minary, for his expertise in primary chromatin structure and CRISPR-Cas9 modelling, as well as his humble guidance and encouragement throughout my research journey. This DPhil thesis benefited from internal collaborations with group members Florian, Furkan, and Yongyao, as well as external collaborations with Walter and Artemi from CONCEPT lab in Istituto Italiano di Tecnologia and Goran, José, Ivan, Fabio and Luiz from Embrapa Digital Agriculture and other Brazilian institutions. I also thank Kathy for the opportunity to co-supervise their third-year project.

I also extend my thanks to the numerous friends that I met at Oxford (list not exhaustive and in no particular order): Kevin W., Yordan Y. Sami L., Kutsi A., Udai D., Benjie W., Quincy v/d B., Wilfried B., Emanuele P., Gordon K. and Samuel W. from Keble; Dorde Z., Siddhartha D., Vivek K., Stefano G., and Minghao L. from the department; Wyatt W. and Thilo S. from St. Hugh's; and Peter W. and Hugo T. from OxHKScholars association. Also shoutouts to Macdonald and Pepe for helping prepare MCR brunches on Sunday, and OUSS for their wonderful salsa and bachata teachers. Finally, heartfelt thanks to my parents and extended family members for their patience, encouragement, and love throughout the journey. Indeed, there is no great genius without a mixture of madness.

Contents

List of Figures

List of Tables

List of Abbreviations

BDM Block Decomposition Method.

bp Base pair(s).

Cα, CA Alpha-carbon.

CA Vleavage activity.

Cas CRISPR-associated.

Cas9 CRISPR-associated protein 9.

CNN Convolutional neural network.

CRISPR Clustered Regularly Interspaced Short Palindromic Repeats.

crRNA CRISPR RNA.

Cryo-EM Cryogenic electron microscopy.

dHMM duration Hidden Markov Model.

DL Deep learning.

DSB Double-stranded break.

dsDNA Double-stranded DNA.

ED Entropy Density.

EP Electrostatic Potential.

FLAG FLAG epitope tag.

GD Graph Descriptor.

gLM Genomic language model.

GN Graph Neighbors.

GRU Gated recurrent unit.

HNH HNH nuclease domain.

HPR Heteroduplex-proximal residue.

IFR Interface forming residue.

Indels insertions and deletions.

LHA Last heavy atom on the side-chain.

LSTM Long short-term memory.

MD Molecular dynamics.

MFE Minimum free energy.

ML Machine learning.

MSA Multiple sequence alignment.

NHEJ Non-homologous end joining.

NLS Nuclear localization signal.

NPT Isothermal-isobaric ensemble.

nt Nucleotides.

NTS Non-target strand.

NUC lobe nuclease lobe.

NVT Canonical ensemble.

P2A Self-cleaving 2A peptide.

PAM protospacer-adjacent motif.

PDB Protein Data Bank.

PI PAM-interacting.

PLL Phosphate lock loop.

pLM Protein language model.

RC Residue Contacts.

REC lobe alpha-helical recognition lobe.

rLM RNA language model.

RMSD Root-mean-square deviation.

RNN Recurrent neural network.

RuvC RuvC nuclease domain.

SCO Side Chain Orientation.

sgRNA Single guide RNA.

SHAP SHapley Additive exPlanations.

Solv Solvation energy.

SV40 Simian Virus 40.

SW Sliding Window.

TMD Targeted molecular dynamics.

tracrRNA Trans-activating CRISPR RNA.

tRNA transfer RNA.

TS Target strand.

UC Unused contacts.

VD Voronoi Diagram.

WCN Weighted Contact Number.

WED Wedge domain.

WNA Weighted Neighbor Average.

Chapter 1

Background

1.1 Biology background

This section provides essential concepts relating to CRISPR-Cas9 biology. We first introduce CRISPR-Cas9 systems and their use for genome editing and other applications. We then examine the protein structure of CRISPR-Cas9, followed by its cleavage mechanism. Next, we describe the off-target effect and finally conclude by protein engineering and bioprospecting efforts for overcoming various limitations of SpCas9.

1.1.1 CRISPR-Cas9 for genome editing

Clustered regularly interspaced short palindromic repeats-associated protein 9 (CRISPR-Cas9, Cas9 in short) is a family of programmable RNA-guided endonucleases [11] which originate from bacteria and archaea adaptive immune systems [12]. The repurposing of *Streptococcus pyogenes* Cas9 (SpCas9) and other Cas9 endonucleases such as for genome editing in mammalian cells [13] has revolutionized the field of gene therapy [14, 15], as evidenced by the Nobel Prize in Chemistry 2020 [16] and the U.S. Food and Drug Administration's approval of Casgevy, the first CRISPR-Cas9 gene therapy for treating sickle cell disease, in 2024 [17, 18]. Functionally, CRISPR-Cas9 systems are powerful tools for site-directed binding and mutagenesis across a wide variety of eukaryotic species [12, 11, 13, 19, 20, 21, 22]. As a result, CRISPR-Cas9 has many applications, including targeted genome editing, modulation of gene expression [23, 24, 25], chromatin visualization [26, 27], epigenetic modifications [28, 29], and chromatin reorganization [30]. Beyond Cas9's endonuclease activity, other editing modalities such as base editing [31, 32] and prime editing [33] have also been developed to further the potential of CRISPR-Cas9 for clinical gene therapy.

The CRISPR-Cas9 genome editor has two components — the Cas9 nuclease and a single guide RNA (sgRNA) [34]. The sgRNA's spacer sequence, typically 20 nucleotides (nt) in length, in CRISPR-Cas9 is highly programmable and easy to design. In essence, by programming the spacer sequence to have base pair complementarity to the genomic target site of interest, one can direct the Cas9-sgRNA binary complex to any target of interest for genome editing [11, 35], provided that the selected target site has a protospacer-adjacent motif (PAM) compatible with the nuclease. For example, SpCas9 has a short 5'-NGG-3' PAM sequence, making it amenable for mammalian genome editing as the PAM is commonly found in GC-rich mammalian genomes.

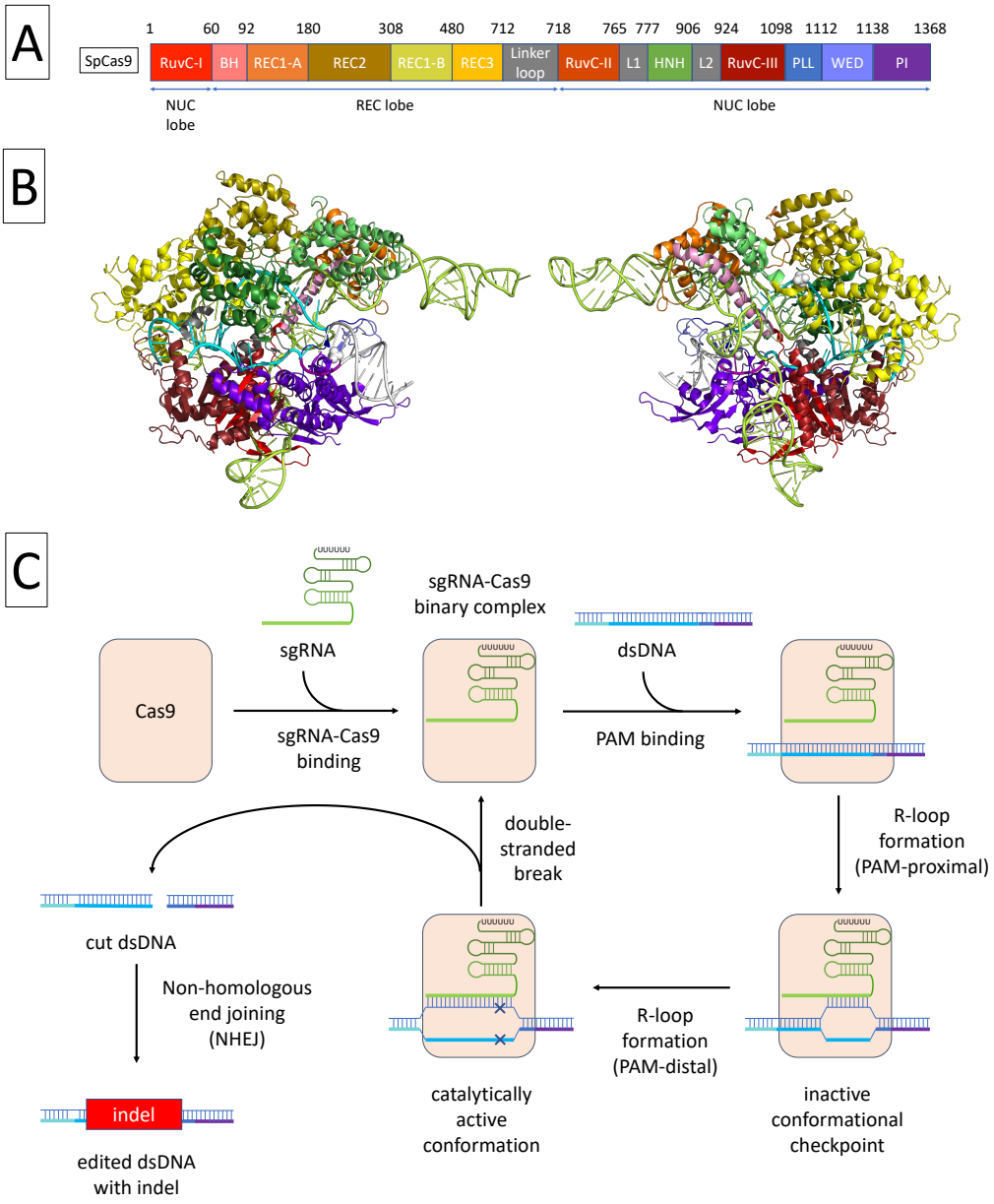


Figure 1.1: (A) SpCas9 protein domains. (B) Front (left) and back (right) view of SpCas9 complex bound to a single guide RNA and double stranded DNA (PDB 5F9R). (C) CRISPR-Cas9 cleavage mechanism.

1.1.2 CRISPR-Cas9 structure and mechanism

Structurally, Cas9 is a bi-lobed enzyme, where the sgRNA is placed between the alpha-helical recognition (REC) lobe and nuclease (NUC) lobe (Figure 1.1A-B). The REC lobe consists of REC domains (REC1-3 for SpCas9) which mediate nuclease binding, whereas the NUC lobe consists of the HNH and RuvC domains used for mediating target strand (TS) and non-target strand (NTS) DNA cleavage, respectively.

For the genomic target site to be edited, the CRISPR-Cas9 molecular machinery mechanistically operates in several steps (Figure 1.1C). First, Cas9 searches the genome for a target site with a protospacer adjacent motif (PAM) compatible with the particular Cas9 nuclease used. Once a target is found, Cas9's PAM-interacting (PI) and Wedge (WED) domains bind to the PAM sequence [36]. PAM binding then initiates unwinding of the target site's dsDNA, which is stabilized by the phosphate lock loop (PLL) [37]. Unwinding of dsDNA at the target site then starts R-loop formation, where the sgRNA-TS heteroduplex forms via complementary base pairing between the sgRNA's spacer sequence and the target DNA strand at the target site [11]. At the same time, Cas9's HNH domain repositions next to the TS. Once R-loop formation is complete, Cas9's HNH and RuvC domains cleave the TS and NTS 3-4 bp upstream of PAM, which create a blunt double-stranded break (DSB) [38]. The DSB is then repaired via non-homologous end joining (NHEJ). Since NHEJ is error-prone, insertions and deletions (indels) are introduced into the DSB, thereby editing the genomic target site. In the case of SpCas9, the CRISPR-Cas9 complex adopts an inactive checkpoint conformation with rearranged REC2, REC3 and HNH domains upon partial sgRNA-DNA hybridization [39], and adopts a catalytically active conformation with HNH positioned next to the target strand [40, 41] once sgRNA-target hybridization is complete.

1.1.3 Off-target effects

An issue with CRISPR-Cas9 systems is that they may cleave *off-targets* [42, 43, 44, 45, 46], i.e., genomic DNA sequences containing mismatches with respect to the sgRNA, which results in undesired cleavage. This is because SpCas9 tolerates base pair mismatches between the sgRNA spacer sequence and target dsDNA. The possibility of off-target cleavage depends on the number of mismatches, their position, and the type of mismatch [47, 39]. For example, a PAM-distal 4-bp mismatch can trap the catalytic HNH domain in an inactive conformation, but mismatches at PAM-proximal positions preserve the shape of the RNA:DNA hybrid [48]. As a result, CRISPR-Cas9 systems are currently not widely adopted in medical applications, since potential off-target Cas9 endonuclease activity [44, 49, 50] may result in undesirable biological effects [51]. To better understand off-target activity, various studies have sought to determine the different factors which influence off-target activity.

As we will see in a later section, computational methods can help accurately identify all potential off-targets and evaluate the activities of such targets [52, 53, 54, 55, 56, 57, 58, 59, 60].

1.1.4 Effects of epigenetics on CRISPR-Cas9 cleavage activity

Various studies have sought to determine the different factors influencing off-target activity. One such factor is the hierarchical chromatin structure which may block off certain genomic regions. Specifically, previous experimental studies reported less CRISPR-Cas9 cleavage for target sites in heterochromatin compared to those in euchromatin in Cas9

mutagenesis experiments [61, 62]. A similar phenomenon with CRISPR-Cas9 binding activity is observed in dead Cas9 (dCas9) binding experiments [63, 64, 65]. Similarly, chromatin accessibility was observed to positively correlate with CRISPR-Cas9 activity [66, 67]. Chromatin state can be inferred by experimental epigenetic features such as DNase I hypersensitivity, CpG methylation and histone marks. These three features can be experimentally measured by DNase-seq [68], reduced representation bisulfite sequencing (RRBS) [69, 70] and histone ChIP-seq screens [71]. Because of this, various biological studies have used these experimental techniques for investigating the impact of the three epigenetic features (or scores) on off-target activity [72]. In particular, DNase I hypersensitivity and CpG methylation were observed to be highly indicative of dCas9 off-target activity [73]. However, CpG methylation was shown to indirectly contribute to off-target activity. This is because it is the DNA-binding methylation-associated factors which likely block Cas9 binding, rather than CpG methylation [61].

Alternatively, local chromatin structure can be defined as the nucleosome organization at the local region. As the basic packaging unit of local chromatin structure, a nucleosome core particle is characterized by the tight wrapping of nucleosomal DNA around a histone octamer. Determined experimentally via X-ray crystallography, it was found that a nucleosomal DNA sequence of length 147 bp is required for the DNA to fully wrap around the histone octamer [74]. Nucleosome organization can be described by nucleosome occupancy or nucleosome positioning. Nucleosome occupancy is defined as the cell and time-averaged probability that a given base pair participates in the nucleosomal DNA wrapping around any histone octamer. Nucleosome positioning is defined as the cell and time-averaged probability that a given base pair sits at the center of any 147bp nucleosomal DNA [75]. Nucleosome occupancy is typically measured by Micrococcal Nuclease digestion with deep sequencing (MNase-seq) [76, 77]. Various studies demonstrate that nucleosomes directly inhibit Cas9 binding and cleavage both *in vitro* and *in vivo* [78, 64, 79, 72]. However, access to nucleosomal DNA can be partially recovered via chromatin remodelling [64] and spontaneous nucleosome breathing [79].

1.1.5 Overcoming limitations of SpCas9

Despite SpCas9's success in genome editing, SpCas9 has several major limitations. Apart from off-target effects, SpCas9 also suffers from:

1. Variable on-target efficiency for different spacer sequences [42, 80];
2. SpCas9 cannot access a large portion of the human genome as SpCas9 can only bind to target sites with the PAM sequence NGG next to it;
3. Because of SpCas9's large gene size of 4.1 kb, an all-in-one SpCas9-sgRNA expression cassette would exceed the ∼4.7 kb packaging limit [81] of an adeno-associated virus (AAV) [81] — a common approach for delivering Cas9 genome editors into mammalian cells [82].

Given a target site of interest, it also remains challenging to optimize the combination of Cas9 variant and sgRNA for efficient and specific genome editing. To overcome these issues, researchers have pursued four research directions: engineering of the SpCas9 nuclease [83, 84, 85, 86, 87, 10], metagenomic mining [88] and engineering [83, 89, 90, 91, 92, 93, 94] of smaller Cas9 orthologs with alternative PAMs, gRNA scaffold optimization [95], and the use of computational modelling of cleavage activity for optimal sgRNA design [96].

Engineered SpCas9 variants

Enhanced efficiency and specificity for SpCas9 can be achieved by mutating protein residues in the nuclease, as motivated by the extensive contacts between SpCas9 and the guide-target heteroduplex in the CRISPR-Cas9 complex. Using structure-guided rational engineering, researchers developed high-fidelity variants such as eSpCas9(1.1) [83], SpCas9-HF1 [84], and HypaCas9 [85]. Alternatively, researchers have used yeast-based directed evolution (DE) [97] to develop evoCas9 [86], and *E. coli*-based DE to develop Sniper-Cas9 [87], Sniper2P and Sniper2L [10]. Sc++ [98] was also rationally engineered from *Streptococcus canis* Cas9 (ScCas9) via multiple sequence alignment between ScCas9 and closely related *Streptococcus* orthologs. As for the broadening of SpCas9's PAM requirement, structure-guided rational engineering gave rise to SpCas9-VQR (VQR onwards) [99], SpCas9-VRER (VRER onwards) [99], SpCas9-VRQR (VRQR onwards) [84], VRQR-HF1 [84], QQR1 [100], SpCas9-NG [101], SpG [102], and SpRY [102], whereas phage-assisted (non-)continuous evolution [103, 104, 105] gave rise to xCas9(3.7) (referred to as xCas9 onwards) [106], SpCas9-NRRH, SpCas9-NRTH, and SpCas9-NRCH [107].

Small Cas9 orthologs and engineered variants

As mentioned earlier, fitting an all-in-one SpCas9-sgRNA expression cassette into an AAV vector is not feasible. Early approaches addressed this issue by identifying smaller Cas9 orthologs that show mammalian genome editing activity, which included St1Cas9 [99, 108, 109, 110, 111], Nm1Cas9 [108, 112, 113, 114], SaCas9 [115, 116, 99, 117, 118, 119], CjCas9 [120, 121], Nm2Cas9 [122], SlugCas9 [94, 91] and SauriCas9 [123]. However, these nucleases lack specificity, so increased fidelity variants such as eSaCas9 [83], efSaCas9 [89], SaCas9-HF [90], SlugCas9-HF [91], and enCjCas9 [92] were developed. The nucleases also have lengthy PAMs, so PAM-relaxed variants such as SaCas9-KKH [93] and SauriCas9-KKH [123] were developed. Further protein engineering gave SaCas9-KKH-HF [90], Sa-SlugCas9 [91], and sRGN3.1 [94], which were developed by stacking SaCas9-KKH and SaCas9-HF mutations, replacing SaCas9's PI domain with that of SlugCas9, and shuffling of protein fragments from Cas9 orthologs similar to SlugCas9, respectively.

gRNA scaffold optimization

Changes to the gRNA scaffold also boosted on-target activity. For SpCas9, this involved increasing the length of the repeat-antirepeat and a T-to-C mutation that broke the 5' continuous stretch of thymines which was misinterpreted as a transcription termination signal. Various SaCas9 [116, 118, 93], St1Cas9 [108, 110], NmCas9 [103, 112, 122], and CjCas9 [120, 121] gRNA scaffolds have also been considered for boosting the on-target efficiency of small Cas9 nucleases.

1.2 Machine and deep learning

This section provides an overview of machine learning and deep learning algorithms used in this thesis.

1.2.1 Supervised learning

Machine learning (ML) [124, 125] is a field of study which leverages data and pattern recognition algorithms for making novel predictions as part of decision making. Supervised learning is a subfield of ML which concerns the learning of a mapping from input features

to labels. Formally, given a *labelled dataset* $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of size N for a task $\mathcal{T}$ of interest, ML algorithms learn a mapping f between *input features* $\mathbf{x}$ and *predicted labels* $\hat{y}$ such that the distance (i.e., *loss function* or *loss objective*) between $\hat{y}$ and *observed labels* y is minimized. The mapping f is also called a *ML model*, whose parameters consist of *learnable parameters* and user-defined *hyperparameters*. The ML model can be seen as a *function approximator*, since the model is trying to approximate the true mapping $\mathbf{x}_i \mapsto y_i$ for each datapoint.

In this thesis, we mainly focus on *regression* tasks, where predicted and observed labels are real-valued, i.e., $y, \hat{y} \in \mathbb{R}$. As a result, the ML algorithms used in this thesis often utilize *mean squared error* $l(\hat{y}, y) = (\hat{y} - y)^2$ as a loss function. When training machine learning algorithms, we divide the dataset into *training*, *validation*, and *test* datasets. Whereas the training dataset is used for training the ML model, the validation dataset is used for tuning model hyperparameters, and the test dataset is used for assessing model performance. To avoid *data leakage*, great care is taken to prevent overlapping of datapoints between the three datasets.

1.2.2 Extreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) [126] is a gradient boosting algorithm [127] based on an ensemble of *classification and regression trees* (CART). In particular, CARTs differ from decision trees by having real values instead of decision values in its leaves. XGBoost is a *boosting* algorithm in the sense that it is a *strong learner* built out of a collection of *weak learners*, i.e., individual CARTs. Formally, predictions for an XGBoost model with CART ensemble $\{f_i\}_{i=1}^t$ are made by summing predictions from each tree i.e., $\hat{y}_i = \sum_{i=1}^K f_k(x_i)$, where K is the tree ensemble size and f_k is the kth tree in the ensemble.

XGBoost is trained using an objective function equal to the training loss function (e.g., MSE) plus a regularization term to prevent model overfitting. Specifically, the regularization term for a tree f with T leaves is defined as $\omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^T w_j^2$, where $w_j \in \mathbb{R}$ is the value attached at the jth leaf node of tree f, and γ, λ are hyperparameters. At iteration step $t \in [T]$, XGBoost grows a new tree f_t, and the loss objective function becomes

$$
\begin{aligned}
\mathcal{L}^{(t)} &= \sum_{i=1}^N \ell(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \omega(f_k) \\
&= \sum_{i=1}^N \ell(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{k=1}^t \omega(f_k) && \text{(tree predictions are summed)} \\
&= \sum_{i=1}^N \ell(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \omega(f_t) + C && \because \text{only } f_t \text{ is optimized at step } t \\
&= \sum_{i=1}^N \left[\ell(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \omega(f_t) + C && \text{(2nd order Taylor series approximation)}
\end{aligned}
$$

where $g_i = \frac{\partial}{\partial \hat{y}_i^{(t-1)}} \ell(y_i, \hat{y}_i^{(t-1)})$, $h_i = \frac{\partial^2}{\partial (\hat{y}_i^{(t-1)})^2} \ell(y_i, \hat{y}_i^{(t-1)})$, and constant $C = \sum_{k=1}^{t-1} \omega(f_k)$

By removing constants $\ell(y_i, \hat{y}_i^{(t-1)})$ and C, we see that minimizing $\mathcal{L}^{(t)}$ is equivalent to

minimizing the objective:

$$
\begin{aligned}
\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{N}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \omega(f_t) \\
&= \sum_{j=1}^{T}\sum_{i\in I_j}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \\
&= \sum_{j=1}^{T}\sum_{i\in I_j}\left[g_i w_j + \frac{1}{2}h_i w_j^2\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 && \because q(x_i) = j \iff j \in I_j \\
&= \sum_{j=1}^{T}\left[\Big(\sum_{i\in I_j} g_i\Big)w_j + \frac{1}{2}\Big(\sum_{i\in I_j} h_i + \lambda\Big)w_j^2\right] + \gamma T + && \text{(group indices by leaves)} \\
&= \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2\right] + \gamma T
\end{aligned}
$$

where q is a function mapping an input to its corresponding leaf's index, $I_j = \{i : q(x_i) = j\}$ is the set of input indices corresponding to leaf j, $G_j = \sum_{i\in I_j} g_i$, and $H_j = \sum_{i\in I_j} h_i$. By using this loss objective, XGBoost can be trained using any training loss function. To learn (a.k.a. grow) the tree structure of each CART, XGBoost greedily selects the input feature which maximizes information gain.

Various limitations of XGBoost led to the development of CatBoost [128] and LightGBM [129]. We refer readers to the respective publications for more information on CatBoost and LightGBM.

1.2.3 Deep learning

Deep learning [130, 131] is a field of study which leverages *artificial neural networks* (neural networks or NN onwards) for pattern recognition. The success of deep learning has seen countless applications in many domains, the most notable being image recognition [132, 133, 134] and text processing [135]. Formally, a neural network is a function approximator with *weight parameters*, where *inferences* or *forward passes* through a neural network are made to make predictions. A neural network learns by updating its weight parameters to minimize a loss objective. More concretely, weights are updated by a user-selected *optimizer* such as stochastic gradient descent (SGD) [136, 137] and Adam [138], where the optimzer uses *backpropagation*, i.e., recursive application of *chain rule* in the neural network's *computational graph*, to perform *gradient descent*, which updates the neural network weights. Metaphorically, the process of gradient descent can be thought of as a ball rolling down a bowl, where the loss is optimized when the ball reaches the bottom of the bowl.

Different types of neural network *architectures* introduce different *inductive biases*, i.e., assumptions used when making predictions for unseen input data. Inductive biases help neural networks become more data efficient during training, as well as improve generalization to unseen data. For example, translation equivariance induced by the convolution operation make convolutional neural networks amenable for image processing tasks, whereas sequential dependency among recurrent units make recurrent neural networks amenable for text-related tasks.

In practice, Python packages such as Tensorflow [139], Keras [140], PyTorch [141, 142, 143] and JAX [144] abstract away the implementation details of deep learning, allowing

users to focus on neural network model development, including selection of model and optimizer hyperparameters, for the task of interest. Regarding hardware, graphical processing units (GPU) help speed up neural network training and inference by parallelizing the many linear algebra operations required in model inference and optimization.

With the above in mind, this section surveys the various types of neural networks used in this thesis, namely fully connnected neural networks, convolutional neural networks and recurrent neural networks.

Fully connected neural network

Also known as *multi-layer perceptrons* (MLP), *fully connected* or *feedforward neural networks* are composed of multiple layers of fully connected layers. Mathematically, on input $\mathbf{x} \in \mathbb{R}^m$, a fully connected layer (i.e., Dense in PyTorch) parameterized by a *weight* matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ and *bias* vector $\mathbf{b} \in \mathbb{R}^b$ computes the output $\mathbf{xW} + \mathbf{b}$. Since the composition of linear functions remains linear, non-linear activations, e.g., the rectified linear unit (ReLU) $f(x) = \max(0, x)$, are appended after all fully connected layers apart from the last one. This allow MLPs to learn piecewise linear functions, and hence be effective function approximators. To prevent model overfitting, *dropout* layers are often inserted between the fully connected and activation layers, where dropout randomly zeroes out each input tensor element with a user-defined probability typically 0.1-0.5.

Convolutional neural networks

Unlike multi-layer perceptrons, *convolutional neural networks* (CNN) use *convolutional layers* for automatic feature extraction. Since DNA and RNA sequences are one-dimensional, we mainly use 1D convolutional layers in subsequent chapters. Formally, given input $\mathbf{X} \in \mathbb{R}^{N \times C_{in} \times L_{in}}$, a 1D convolutional layer (named Conv1D in PyTorch) parameterized by weight $\mathbf{W} \in \mathbb{R}^{C_{out} \times C_{in} \times K}$ and bias $\mathbf{b} \in \mathbb{R}^{C_{out}}$ yields an output $\mathbf{Y} \in \mathbb{R}^{N \times C_{in} \times L_{in}}$ with

$$\mathbf{Y}_{n,c_0,\ell} = \mathbf{b}_{c_0} + \sum_{c=0}^{C_{in}-1} \sum_{k=0}^{K-1} \mathbf{W}_{c_0,c,k} \mathbf{X}_{n,c,\ell+k}$$

for datapoint index $n \in [0, N)$, output channel $c_0 \in [0, C_{out})$, and output sequence position $\ell \in [0, L_{out})$, where:

- N denotes the batch size;
- C_{in} denotes the number of input channels;
- C_{out} denotes the number of output channels;
- L_{in} denotes the input sequence length;
- L_{out} denotes the output sequence length;
- K denotes the convolutional *filter* or *kernel* size.

Note that we assumed default parameters for stride, padding, dilation and groups in this definition. We refer readers to `https://docs.pytorch.org/docs/stable/generated/torch.nn.Conv1d.html` for further explanations of these other parameters. Similar to feedforward neural networks, dropout and non-linear activation layers are added after the convolutional layer for the same reasons explained above. In PyTorch, CNNs typically have a series of Conv1D-Dropout-ReLU layers, followed by a Flatten layer to merge the

feature and channel dimensions, and then followed by a series of Dense-Dropout-ReLU layers.

Recurrent neural networks

Recurrent neural networks (RNN) are inspired by the need to persist information across a sequence of inputs. One way of building an RNN is to define a neural network f parameterized by weights W, where f at time step t takes a feature vector $x^{(t)}$ and hidden state $h^{(t-1)}$ as input and outputs a vector $y^{(t)}$ a new hidden state $h^{(t)}$, i.e., $(y^{(t)}, h^{(t)}) = f(x^{(t)}, h^{(t-1)}; W)$. However, such an implementation suffers from the *vanishing gradient* problem, where the gradients for the earlier layer weights become exponentially smaller due to increasing number of gradient multiplications.

To fix this, *gated* RNNs were proposed, with the main architectures being *long short-term memory* (LSTM) [145] and *gated recurrent unit* (GRU) neural networks. Unlike the previous naive implementation, LSTMs and GRUs leverage gates to capture long-term sequence dependencies. Formally, an LSTM layer parameterized by weights W_{fi}, W_{fh}, W_{ii}, W_{ih}, W_{ci}, W_{ch}, W_{oi}, W_{oh} and biases b_{fi}, b_{fh}, b_{ii}, b_{ih}, b_{ci}, b_{ch}, b_{oi}, b_{oh} takes a feature vector $x^{(t)}$, a previous hidden state $h^{(t-1)}$ and a previous cell state $C^{(t-1)}$ as inputs, and outputs a new hidden state $h^{(t)}$ and a new cell state $C^{(t)}$, where

$$
\begin{aligned}
f^{(t)} &= \sigma(W_{fi}x^{(t)} + b_{fi} + W_{fh}x^{(t)} + b_{fh}) && \text{(Forget gate layer)} \\
i^{(t)} &= \sigma(W_{ii}x^{(t)} + b_{ii} + W_{ih}x^{(t)} + b_{ih}) && \text{(Input gate layer)} \\
\tilde{C}^{(t)} &= \tanh(W_{ci}x^{(t)} + b_{ci} + W_{ch}x^{(t)} + b_{ch}) && \text{(Create new values to write)} \\
C^{(t)} &= f^{(t)} \odot C^{(t-1)} + i^{(t)} \odot \tilde{C}^{(t)} && \text{(Update cell state)} \\
o^{(t)} &= \sigma(W_{oi}x^{(t)} + b_{oi} + W_{oh}x^{(t)} + b_{oh}) && \text{(Compute output)} \\
h^{(t)} &= o^{(t)} \odot \tanh(c_t) && \text{(Update and output hidden state)}
\end{aligned}
$$

with $\odot$ denoting element-wise multiplication.

GRUs were then introduced to lower the computational cost compared to LSTMs. Compared to LSTMs, the forget and input gates are merged into one update gate, and the cell and hidden states are merged into one hidden state in GRUs.

Formally, a GRU layer parameterized by weights W_{zi}, W_{zh}, W_{ri}, W_{rh}, W_{hi}, W_{hh} and biases b_{zi}, b_{zh}, b_{ri}, b_{rh}, b_{hi}, b_{hh} takes a feature vector $x^{(t)}$ and a previous hidden state $h^{(t-1)}$ as input, and outputs the feature vector $h^{(t)}$, where

$$
\begin{aligned}
z^{(t)} &= \sigma(W_{zi}x^{(t)} + b_{zi} + W_{zh}h^{(t-1)} + b_{zh}) && \text{(Update gate layer)} \\
r^{(t)} &= \sigma(W_{ri}x^{(t)} + b_{ri} + W_{rh}h^{(t-1)} + b_{rh}) && \text{(Reset gate layer)} \\
\tilde{h}^{(t)} &= \tanh(W_{hi}x^{(t)} + b_{hi} + W_{hh}(r^{(t)} \odot h^{(t-1)}) + b_{hh}) && \text{(Create new values to write)} \\
h^{(t)} &= (1 - z^{(t)}) \odot h^{(t-1)} + z^{(t)} \odot \tilde{h}_t && \text{(Update and output hidden state)}
\end{aligned}
$$

Bidirectional RNNs are built by stacking forward direction and reverse direction LSTMs/GRUs. This architecture enable bidirectional RNNs to learn sequence dependencies in either directions.

1.2.4 SHapley Additive exPlanations

Inspired by coalition game theory, SHapley Additive exPlanations (SHAP) [146] computes the contribution of individual features towards the XGBoost model's predictions. SHAP

does this for a given datapoint by assigning SHAP values to each input feature such that the SHAP values sums to the model's prediction minus a constant baseline. Mathematically, for datapoint i:

$$m(X^{(i)}) = \hat{y}^{(i)} = t(X^{(i)}) = b + \sum_{j=1}^{|F|} \phi_j^{(i)} \tag{1.1}$$

where:

- m is the ML model,
- t is the explanation model,
- F is the set of input features to m,
- $\hat{y}^{(i)} \in \mathbb{R}$ is the model's prediction for datapoint i,
- $X^{(i)} \in \mathbb{R}^{|F|}$ is the input feature vector for datapoint i,
- $b \in \mathbb{R}$ is some constant baseline, and
- $\phi_j^{(i)}$ the SHAP value assigned to feature j for datapoint i.

SHAP values can be used for both local or global interpretation. While local interpretation allows one to explain individual predictions, we are more interested in global interpretation. Specifically, global interpretation allow us to quantify and rank the importance of input features. Namely, from SHAP values ϕ, the SHAP importance I_j of a feature j in a model m can be quantified by the following equation:

$$I_j = \frac{1}{N} \sum_{i=1}^{N} |\phi_j^{(i)}| \tag{1.2}$$

where N is the number of datapoints. Similarly, the SHAP importance I_J of a feature group J can be quantified by the following equation:

$$I_J = \frac{1}{N} \sum_{i=1}^{N} \Big| \sum_{j \in J} \phi_j^{(i)} \Big| \tag{1.3}$$

SHAP values also allow us to identify how variations in the feature value for a single feature impact the model's output across the whole dataset. This allows us to study how changes in input feature values affect model predictions within the dataset.

The Python SHAP package provides an API for producing SHAP summary plots from SHAP values. Normally limited to 20 rows, each row in a SHAP summary plot is a horizontal beeswarm plot for each input feature, where the input features are ordered by decreasing SHAP feature importance. Dots in the beeswarm plot for each feature are colored by the datapoint's feature value. Red, purple and blue dots in the plot correspond to high, medium and low feature values, respectively.

1.2.5 Uncertainty quantification

Deep ensembles

Predictions made by neural networks are overconfident, making it problematic to use neural networks in safety-critical applications like genome editing. One way of overcoming

this is to use deep ensembles [147] for estimating predictive uncertainty. Formally, a deep ensemble of size N makes predictions by combining the predictions of multiple mean-variance neural networks [148], which have two output heads — one for the predicted mean $\hat{\mu}$ and one for the predicted variance $\hat{\sigma}^2$ — and are trained using a Gaussian negative log likelihood loss:

$$\mathcal{L} = \frac{1}{2}\left(\log(\max\{\hat{\sigma}^2, \epsilon\}) + \frac{(y-\hat{\mu})^2}{\max\{\hat{\sigma}^2, \epsilon\}} \right) + C \tag{1.4}$$

where y is the label, C is some constant, and ϵ (typically 1×10^{-6}) is a constant added to stabilize training.

In a deep ensemble with N mean-variance NNs, given mean-variance predictions $\{(\hat{\mu}_i, \hat{\sigma}_i^2)\}_{i=1}^{20}$ for input $\mathbf{x}_i$, the deep ensemble's predicted mean and variance is given by $\mu = M^{-1}\sum_{i=1}^{M}\hat{\mu}_i$ and $\sigma^2 = M^{-1}\sum_{i=1}^{M}(\hat{\sigma}_i^2 + \hat{\mu}_i^2) - \mu_{\text{naive}}^2$, respectively.

Quantile calibration

To assess the quality of uncertainty estimates produced by deep ensembles, we adopt Kuleshov et al.'s definition of quantile calibration [149]. In the regression setting, Kuleshov et al.'s definition of quantile calibration states that a ML model generating a predictive distribution for datapoint i with label y_i and cumulative distribution function (CDF) $F_i : \mathbb{R} \to [0, 1]$ (i.e., quantile function F_i^{-1}) is quantile calibrated if

$$\forall p \in [0, 1] : \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{y_i \leq F_i^{-1}(p)\} = p$$

with N as the dataset size. We estimate this by selecting confidence levels $p_j \in \{0, 0.01, \ldots, 1\}$ and plotting $\{(p_j, \hat{p}_j)\}_{j=1}^{101}$ where $\hat{p}_j = \frac{1}{N}|\{y_i | F_i(y_i) \leq p_j, i \in [N]\}|$ is the empirical frequency.

Kuleshov et al.'s definition can also be adjusted to use confidence intervals (CI) instead, where a ML model is calibrated if

$$\forall p \in [0, 1] : \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{F_i^{-1}(0.5 - \frac{p}{2}) \leq y_i \leq F_i^{-1}(0.5 + \frac{p}{2})\} = p$$

Similarly, we estimate this by selecting confidence intervals $p_j = 0, 0.01, \ldots, 0.99, 1$ and plotting $\{(p_j, \hat{p}_j)\}_{i=1}^{101}$ where $\hat{p}_j = \frac{1}{N}|\{y_i | 0.5 - \frac{p_j}{2} \leq F_i(y_i) \leq 0.5 + \frac{p_j}{2}, i \in [N]\}|$ is the CI-based empirical frequency. Finally, Kuleshov et al.'s approach for computing the quantile calibration error is given by

$$\text{cal}(\{(\hat{p}_j, p_j)\}_{j=1}^{101}) = \sum_{j=1}^{101}(\hat{p}_j - p_j)^2$$

for both calibration definitions defined above.

1.3 Computational modelling of CRISPR-Cas9 cleavage activity prediction

Accurate identification of all potential off-target sites and evaluation of their activities have been the goals of various computational tools [52, 53, 54, 55, 56, 57, 58, 59, 60]. The

main motivation for developing such tools stems from the potential of such tools to become cheap *in silico* alternatives to the low-throughput and costly *in vitro* and *in vivo* surrogate reporter [150, 151] and gold-standard indel frequency measurement assays [152, 153, 154] typically required for optimal sgRNA design. In this section, we first describe how the CRISPR-Cas9 cleavage activity problem is phrased as a computational problem, and then describe the various types of algorithms used to solve the computational problem.

1.3.1 Problem Formulation

Since the spacer and target sequences of the CRISPR-Cas9 complex are the primary factors influencing Cas9 cleavage activity [42], most approaches in the literature [80, 56, 52, 155, 53] aim to accurately capture the functional relationship between the spacer-target interface and CRISPR-Cas9 cleavage activity. Mathematically, the relationship can be defined as a function $f : \Sigma^{\ell} \times \Gamma^{\ell'} \rightarrow \mathbb{R}$ with the mapping $(s_g, s_t) \mapsto a$ for a length-ℓ spacer sequence s_g, length-ℓ' target sequence s_t, and experimentally measured Cas9 cleavage activity a, with $\Sigma = \{A, U, C, G\}$ and $\Gamma = \{A, T, C, G\}$ denoting the nucleic acid alphabets for RNA and DNA, respectively. In the case of SpCas9, we typically have $\ell = 20$ and $\ell' = 23$ as a result of the 20nt spacer and 3nt PAM. Computationally, the spacer and target sequences are represented as *strings*, whereas the cleavage activity value is represented as a *floating point number*. In this thesis, we use Cas9 cleavage activities which are measured using indel frequencies [152, 153, 154] and cleavage rates [156, 102, 157] rather than those measured log2 fold change [80, 56], as the latter relies on changes in gene expression from the reporter gene construct to indirectly capture DSBs, which introduces variance to the activity label.

The input tuple (s_g, s_t) can be extended to model the influence of other biological factors influencing Cas9 activity. Such factors include epigenetic features in the *in vivo* setting as we explore in chapter 2 and Cas9 residue-related features as we explore in chapters 3 and 4.

1.3.2 Rule- and alignment-based approaches

Early pioneering approaches relied on rule- and alignment-based algorithms for ranking candidate sgRNAs and listing candidate off-targets, respectively. The MIT (Hsu-Zhang) score aggregates position-specific mismatch penalties across all potential off-target sites of a given sgRNA to calculate a sgRNA specificity score. CHOPCHOP [158] is a web server which ranks candidate sgRNAs by five criteria: the number of off-targets, the number of mismatches within the off-targets, the sgRNA's GC content and the presence of a guanine at position 20 (counting 5' to 3'). CCTop [159] also produces sgRNA aggregate scores to rank sgRNAs, with off-target scores calculated by summing exponentials across mismatch positions. Similar to CHOPCHOP and CCTop, CROP-IT [160] uses a rule-based algorithm for ranking sgRNAs, but incorporates DNase-I seq chromatin accessibility information to boost CROP-IT's off-target prediction performance. Cas-OFFinder [161] is a fast OpenCL-based sequence alignment algorithm which identifies potential genomic off-target sites given the spacer sequence and nuclease of interest. Since SpCas9 can tolerate spacer-target interfaces with up to 5-6 mismatches [42, 162], researchers often configure Cas-OFFinder so that it finds off-targets with ≤ 6 mismatches, which is helpful in SpCas9 off-target datset curation workflows.

1.3.3 Traditional machine learning

Whereas alignment- and rule-based algorithms directly define the function based on domain knowledge of CRISPR-Cas9 systems, ML–based methods build on labeled datapoints containing the spacer sequence, target sequence, and experimentally measured Cas9 cleavage activity. A given dataset with N such datapoints, $\left\{\left(s_g^{(i)}, s_t^{(i)}, a_i\right)\right\}_{i=1}^{N}$ can be partitioned into training, validation and test sets so that models for (off-)target cleavage activity prediction can be constructed. In terms of feature representation, the spacer and target sequences are typically represented using *one-hot encodings*. ML models require the careful selection of relevant features related to the activity of a given sgRNA at a potential (on/off) target site. Some of the most widely used observed features originate from pioneering work on optimised sgRNA design [56, 80] and include (but are not limited to) dinucleotide and single-nucleotide identities at each position of the sgRNA, position independent nucleotide counts, the location of the sgRNA within the gene, the GC count of the sgRNA as well as thermodynamic features. These features were first used to feed "traditional" predictive ML methods, e.g., regularized linear regression, support vector machines [163], random forest [164] and gradient-boosted regression trees [165, 126, 129, 128]. Prominent ML-based SpCas9 activity models include Rule Set 1 [80], CRISPOR [162], sgRNA scorer 2.0 [166], Azimuth [56], and Elevation [57].

1.3.4 Deep learning

Deep learning [132, 130, 133, 134, 145] has been instrumental in building the most widely used and efficient models for on/off-target activity prediction [52, 155, 53, 167, 53]. Deep neural networks have the advantage of high prediction accuracy but make model interpretation more challenging and need a large amount of training data typically obtained from *in vitro/vivo* genome-wide off-target cleavage detection assays [168, 169, 170, 171, 172, 173, 174, 175, 176] or high-throughput guide-target lentiviral library screens [177, 178, 5]. Notably, DL models are favored over ML models due to their ability to perform automatic feature extraction and superior predictive performance [5, 179] over ML models. A variety of neural network architectures have been used, including convolutional neural networks (CNN) [52, 167, 180], recurrent neural networks (CNN) [4], convolutional-recurrent neural networks (C-RNN) [181, 182, 53, 183], and kinetically interpretable neural network (KINN) [184] for Cas9 cleavage activity prediction.

Since the spacer and target sequences are the primary determinants affecting CRISPR-Cas9 cleavage activity, the majority of ML- and DL-based models have relied on representations of the spacer-target interface for input features to the neural network. More concretely, such features include one-hot encoding of the spacer-target interface [167], GC counts, computed DNA melting temperatures and computed minimum free energy of the sgRNA [8].

Beyond the guide-target interface, some DL-based models [185, 181, 186, 187] complement the sequence features with a diverse set of physically-inspired scores such as approximate energy terms [59] for R-loop formation and sgRNA–target strand DNA (TS) hybridization. Given that cleavage activity can be modulated by epigenetics [185], specifically primary chromatin structure [78, 64, 79, 188, 189], some DL-based models [52, 182, 155, 181] also incorporate computed and/or epigenetic input features to represent the chromatin state at off-target sites. Such features include CCCTC-binding factor (CTCF, [190]), chromatin immunoprecipitation (ChIP, [191]), histone-3 lysine-4 trimethylation (H3K4me3, [192]), reduced representation bisulfite sequencing (RRBS, [69, 70]) and Deoxyribonuclease-I hypersensitive sites sequencing (DNase-seq, [68]) assays. Available in

crisprSQL [54], DNA:RNA ImmunoPrecipitation and high-throughput sequencing (DRIP) is an epigenetic score which measures R-loop formation in the genome [193, 194]. Notably, R-loops play a role in regulating chromatin states [195].

Similar approaches have been used for dealing with Cas9 variants apart from SpCas9. Example of such tools include DeepxCas9 [6], DeepSpCas9-NG [6], DeepSpCas9variants [7], DeepSmallCas9 [8], DeepCas9variants [9], and DeepSniper [10].

1.3.5 Biophysical models

Since R-loop formation is the rate-limiting step in Cas9 cleavage, researchers hypothesized that the R-loop formation energy would correlate with Cas9 cleavage activity, which inspired the development of thermodynamic free energy-based models such as uCRISPR [60], CRISPRoff [59], CRISPRspec [59], and CRISPRspecExt [196]. Later research revealed that R-loop formation process is under kinetic control rather than in thermodynamic equilibrium, inspiring the development of kinetic models for wild-type and engineered SpCas9 variants such as those developed by Jones Jr. et al [197] and Eslami-Mossallam et al. [198]. CRISOT uses molecular dynamics to derive RNA-DNA molecular interaction fingerprints, which are then used for SpCas9 cleavage activity prediction.

1.4 Nanoenvironment approach

In this thesis, the concept "nanoenvironment" refers to a specific internal protein region with well-defined characteristics and a unique set of corresponding STING descriptors [199, 200, 201, 202], i.e., physicochemical and structural descriptors, that are able to select only the amino acid residues that make up that part of the protein region. As we will see in Chapter 3, we only care about the protein region surrounding the sgRNA-TS heteroduplex, so the nanoenvironment of interest in Chapter 3 is defined by the set of Cas9 residue-resolved physicochemical and structural descriptor values located on heteroduplex-proximal Cas9 residues. This is in contrast to previous definitions of the nanoenvironment, where such functionally distinct regions were named as protein districts, using a common analogy of internal protein regions with city districts. Previous work [199, 200, 201, 202, 203, 204] has been successfully connected to the similar characterization of certain residues within a protein region with some functional properties (such as enzyme activity or protein interfaces) of the system in the study.

1.5 Motivation & Objective

This thesis is largely motivated by the following questions: Can CRISPR-Cas9 cleavage activity prediction be better modelled using ML and DL if we consider structural factors beyond the guide-target sequences? If such ML/DL models were built, does model interpretation enable us to better understand the CRISPR-Cas9 machinery? Does the model interpretation corroborate with our current understanding of CRISPR-Cas9 biology? These questions stemmed from the fact that most prediction tools at the time relied on the spacer-target interface of being the primary determinant of Cas9 cleavage activity. Additionally, wet-lab and structural CRISPR-Cas9 studies had determined that epigenetics in the form of chromatin accessibility and nucleosomes at the target site, as well as certain SpCas9 residues, influenced SpCas9 cleavage activity. If these questions could be adequately addressed, then the accurate predictions made from the resulting enhanced ML/DL models would allow us to facilitate safety quantification in genome editing

experiments and clinical gene therapy.

Chapter 2 was mainly motivated by DeepCRISPR, one of the first DL-based SpCas9 cleavage activity prediction tools. Published in 2018, it was also the first tool to featurize the epigenetic environment of a (off-)target site by incorporating four epigenetic markers (DNase I, RRBS, CTCF and H3K4me3) into the DL model's input. However, it was unclear whether the four epigenetic features had meaningfully contributed to DeepCRISPR's performance. The issue was compounded by a lack of computational studies which assess the impact of primary chromatin structure on Cas9 activity. Given that nucleosome position and breathing were shown to influence SpCas9 cleavage activity in *in vitro* and *in vivo* experiments, we hypothesized that experimental nucleosome occupancy and positioning features such as MNase-seq data could be useful as input features for ML/DL-based SpCas9 cleavage activity prediction models. However, such experimental data is expensive to obtain, so we also considered scores produced by existing computational nucleosome occupancy and positioning tools as input features for the ML/DL models. As a result, for Chapter 2, we had the objective of utilizing various *experimental epigenetic features* and *computed nucleosome organization-related features*, and determining which features were found to be important by the trained ML/DL models when predicting SpCas9 cleavage activity.

The study in Chapter 3 was inspired by the fact that most existing ML/DL-based SpCas9 cleavage activity tools only featurized the spacer-target interface of the CRISPR-Cas9 complex as input. This was a research gap that we paid attention to, as structural studies had showed that the SpCas9 plays an integral role in conformational proofreading of the sgRNA-TS heteroduplex, and yet SpCas9 was not being modelled in existing computational cleavage activity prediction tools. Mechanistically, changes to the heteroduplex nucleotides induce change in the heteroduplex's configuration and physicochemical properties, which in turn changes physicochemical/structural properties of heteroduplex-proximal Cas9 residues due to SpCas9-heteroduplex interactions. Therefore, monitoring the changes in physicochemical/structural descriptors characterizing heteroduplex-proximal Cas9 residues can capture (due to SpCas9-heteroduplex interactions) subtle changes (e.g. mutations, mismatches) in the heteroduplex sequence content. Because of this, we hypothesized that the variability in per-residue descriptor values for different spacer-target interfaces would enable one to build a functional mapping between residue-resolved physicochemical/structural descriptor properties and SpCas9 activity (e.g. by ML). In other words, the SpCas9 activity problem can be reformulated from the usual spacer-target sequence pair to SpCas9 activity functional mapping to the functional mapping between the aforementioned Cas9 residue features and SpCas9 activity. Interpretation of a ML model that has learned such a functional mapping would then allow us to deduce and quantify the importance of SpCas9 residues and physicochemical/structural descriptors pertinent to SpCas9 cleavage activity. With the above framework in mind, we sought to build such a ML model and interpret the model, which would hopefully reveal SpCas9 residues relevant to SpCas9's cleavage mechanism. As part of a research collaboration, this chapter was completed with Walter and Artemi from Istituto Italiano di Tecnologia, who ran the all-atom molecular dynamic simulations, and Goran's research group from Embrapa Digital Agriculture, which has expertise on the physicochemical and structural descriptors used in this chapter.

Nonetheless, one major limitation of the approach taken in Chapter 3 is that the approach cannot be scaled to large numbers of spacer-target interfaces due to the large computational resources required for protein-nucleic acid molecular dynamics simulation. At the same time, the CRISPR-Cas9 literature saw the development of many single-variant

DL-based cleavage activity prediction models for a wide range of SpCas9 and smallCas9 variants, where the models include DeepxCas9, DeepSpCas9-NG, DeepSpCas9variants, DeepSmallCas9, DeepCas9variants and DeepSniper. Given the above observations, we hypothesized that another way of incorporating information relating to the Cas9 nuclease in a ML model is to leverage protein language model (pLM) embeddings as input features, since pLM embedding have been shown to efficiently capture protein coevolution (i.e., residue-residue contact) information. However, unlike the residue-resolved physicochemcial/structural features extracted from catalytically active Cas9 structures complexed with a sgRNA and target double stranded DNA (dsDNA), pLM embeddings derived solely from Cas9 protein sequences do not capture Cas9-heteroduplex interactions. To compensate for this, a DL model using such pLMs would need to be trained on indel frequency datasets arising from a diverse set of Cas9 variants. Combined with the use of RNA language model (rLM) model embeddings to encode the entire sgRNA as input, the resulting DL model would be the first to featurize all three components of the complex — the sgRNA, target sequence and the Cas9 nuclease. Given the above ideas, Chapter 4 has the goal of building a DL model enriched with Cas9 protein coevolution and sgRNA structural information using a carefully curated dataset spanning many Cas9 variants, and also interpreting the DL model to see what biological insights can be obtained.

Chapter 2

Comprehensive computational analysis of epigenetic descriptors affecting CRISPR-Cas9 off-target activity

This chapter has been published in *BMC Genomics* with the citation *Mak, J. K., Störtz, F., & Minary, P. (2022). Comprehensive computational analysis of epigenetic descriptors affecting CRISPR-Cas9 off-target activity. BMC Genomics, 23(1).* `https://doi.org/10.1186/s12864-022-09012-7`. The nucleosomal features generated in this chapter have also been used in a related publication reproduced in Appendix D.1.

2.1 Background

In this chapter, we aim to conduct a comprehensive computational investigation on the impact of structural epigenetic features on CRISPR-Cas9 off-target activity. We use the Cas9 off-target activity database crisprSQL [54] with over 25,000 guide-off-target datapoints and a comprehensive set of computational tools in this study. By doing so, we find that several nucleosome organization-related features attain higher correlation with off-target activity compared to the existing experimental epigenetic scores. In particular, this correlation is significantly higher for two Block Decomposition Method-based features [205, 206]. We also build physically inspired off-target activity prediction models that are purely based on empirical free energy estimates of the sgRNA-DNA heteroduplex and epigenetic features. This allows us to evaluate the impact of epigenetic features in the context of CRISPR-Cas9 activity model prediction. We find that said models take advantage of the computed nucleosome organization-related features but pay less attention to the commonly used experimental epigenetic scores.

2.2 Results

2.2.1 Spearman and Pearson correlation analysis

Figure 2.1 shows two heatmaps denoting the Spearman and Pearson correlations of off-target cleavage activity with 19 epigenetic features (see exact values in Appendix Table A.1). The 19 epigenetic features consist of 6 experimental epigenetic features (names

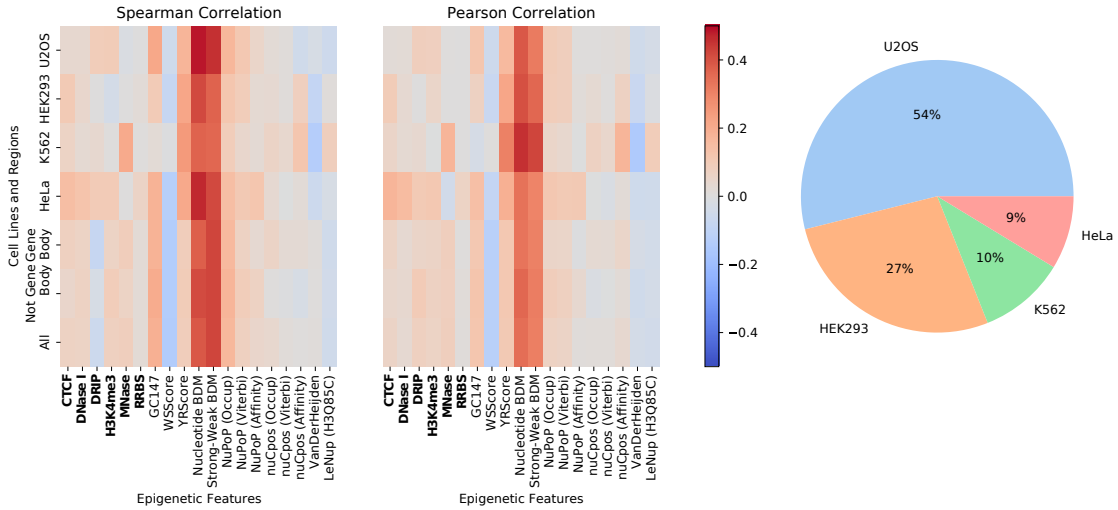


Figure 2.1: Heatmaps showing Spearman (left) and Pearson (middle) correlations between 19 epigenetic features and SpCas9 off-target cleavage activities. Red and blue colors represent positive and negative correlations, respectively. The 19 epigenetic features consists of six experimental epigenetic features (bolded) and 13 nucleosome organization-related scores. The first four rows in the heatmaps display cell line-specific correlations. The fifth and sixth row display correlations for off-target sites in gene body and non-gene body regions. The final row displays the overall correlation for the epigenetic features. (Right) Pie plot showing the dataset's cell line composition including all cell lines that contribute more than 1% to the crisprSQL dataset.

bolded in the figure) and 13 computed nucleosome organization-related features. Heatmap correlations are calculated for target sites in human cell lines HeLa, K562, HEK293 and U2OS from the CRISPR-Cas9 activity cleavage crisprSQL database [54]. To investigate whether correlation values vary between cell lines and genomic regions, heatmap correlations are displayed for all data, individual cell lines and gene/non-gene body regions. The rightmost pie chart shows the cell line composition of the dataset used for analysis. Overall, Spearman and Pearson correlations for the 19 epigenetic features considered range between -0.5 and 0.5. Only Nucleotide BDM and Strong-Weak BDM, i.e. BDM-based scores (see subsubsection "Block Decomposition Method-based Measures" in subsection 2.5.4), exhibit highly positive correlations when considering all cell lines. Specifically, Nucleotide BDM has Spearman and Pearson correlations of 0.388 and 0.345, and Strong-Weak BDM has correlations of 0.423 and 0.310. Similar values are obtained for Nucleotide BDM and Strong-Weak BDM when considering cell lines individually. When filtering off-target sites by gene body and non-gene body regions, similar Spearman and Pearson correlations are observed across all epigenetic features. This indicates that correlations are not dependent on whether off-targets are in gene bodies. A similar trend is observed when considering each cell line separately (see Appendix Figures A.2-A.4).

Table 2.1 highlights the correlation coefficients for the experimental epigenetic features shown in Figure 2.1. In the table, Spearman/Pearson correlations between the six experimental features and off-target cleavage activities in any human cell lines range between -0.1 and 0.1. MNase, which is indicative of nucleosome occupancy rather than nucleosome positioning, has a Spearman and Pearson correlation of 0.08 and 0.08, respectively. Similar values are obtained for the various MNase-seq data across HeLa, K562 and U2OS (see

Experimental Epigenetic Feature	Spearman	Pearson
CTCF	0.07	0.06
DNase I	0.07	0.03
DRIP	-0.06	0.08
H3K4me3	0.07	0.07
MNase	0.08	0.08
RRBS	0.02	0.01

Table 2.1: Spearman and Pearson correlation values between SpCas9 off-target cleavage activities and each experimental epigenetic scores for the crisprSQL dataset used in Figure 2.1. The experimental epigenetic scores are CTCF, DNase I, DRIP, H3K4me3, MNase and RRBS.

Appendix Figures A.2, A.3 and A.4, respectively).

Figure 2.2 shows the violin and distribution plots for Nucleotide BDM, GC147, YR Scheme and MNase when splitting cleavage activities (CA) into three bins. These bins are CA $= -4$, CA ≤ 2 and CA > 2 (see Appendix Figures A.5 and A.6 for all epigenetic features). In the leftmost column for Nucleotide BDM, most off-target sites with low Nucleotide BDM value fall under the lowest cleavage activity bin CA$=-4$. The lowest cleavage activity datapoints are almost exclusively composed of augmented datapoints with sequence alignment-derived putative off-target sites. Such putative off-target sites are assigned the lowest cleavage activity value CA $= -4$ on the assumption that such sites have no off-target activity. Therefore, these datapoints do not carry experimentally derived cleavage activity labels. In addition, these datapoints constitute the larger fraction (52%) of all datapoints. A similar phenomenon is observed for Strong-Weak BDM (see Appendix Figures A.5 and A.6).

2.2.2 Machine/Deep Learning-based SHAP analysis

We saw that some computed nucleosome organization-related features correlate with CRISPR-Cas9 off-target activity. As a result, we sought to determine whether the aforementioned features also show patterns in machine and deep learning off-target cleavage activity prediction models. We also sought to investigate the importance of said features without the influence of explicitly encoded base pair identities. To achieve this, we built two models. The first model is an extreme gradient boosted (XGBoost) tree model. The second model is a convolutional neural network (CNN) model (see Appendix Figure A.1 for neural network architecture). Both models take all 19 epigenetic features and three binding energy scores as input and predict off-target cleavage activities. Included in crisprSQL, the three energy scores represent free energy estimates used for estimating the DNA-RNA heteroduplex formation's free energy. These energy terms have been generated by using the CRISPRspec [59] biophysical interaction model, which provides various binding energies scores (called CRISPRspec binding energy scores). These binding energies scores are further explained in the Methods section (see subsection "CRISPRspec"). The XGBoost and CNN models expect nucleosome organization-related features (scores) at base-pair resolution (23 scores per target site). sgRNA-DNA sequences were not included as input to both models. This is to avoid the interference of sequence features with epigenetic features when computing feature importance scores after training. Instead, we included the sgRNA-DNA sequences-derived CRISPRspec binding energy scores. When testing on the held out 20%, the XGBoost model achieves a Spearman and Pearson correlation of 0.419

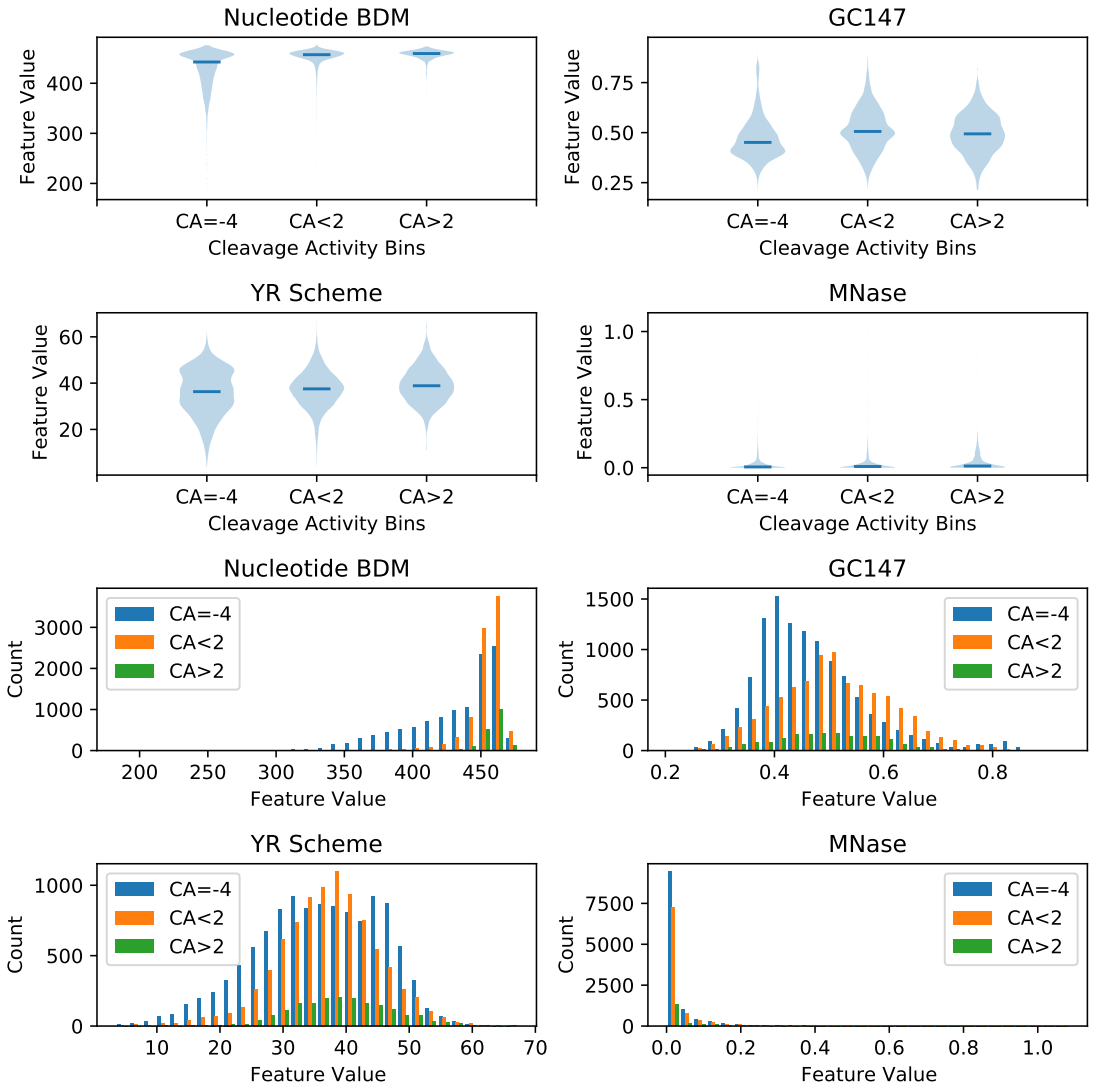


Figure 2.2: Violin (top) and distribution (bottom) plots for the epigenetic features with high Pearson correlation, namely Nucleotide BDM, GC147, YR Scheme and MNase. Cleavage activities (CA) are separated into three bins representing low (CA $= -4$, colored blue), medium (CA ≤ 2, colored orange) and high (CA > 2, colored green) cleavage activity. See violin and distribution plots for other epigenetic features in Appendix Figures A.5 and A.6, respectively.

and 0.617, respectively. The CNN model yields similar correlations, namely a Spearman and Pearson correlation of 0.424 and 0.594, respectively.

Next, we interpret the two models using SHAP (see Methods) after training and evaluating the contributions of each input feature. To evaluate a model, a randomly drawn test dataset containing 2000 points is used. Figure 2.3 and Appendix Figure A.7 show the resulting feature-based SHAP summary plot and base pair resolution heatmap, respectively, for the trained XGBoost model. An analogous summary plot and heatmap for the CNN model can be found in Figure 2.4 and Appendix Figure A.8. In the two SHAP summary plots, the distribution of SHAP value contributions is shown for every

input feature present in the model. Model input features are ordered in decreasing SHAP feature importance. In other words, features at the top carry high SHAP feature importance, and features at the bottom carry low SHAP feature importance. In both SHAP summary plots, the SHAP feature importance of the six experimental epigenetic scores are not comparable to the nucleosome organization-related scores. In addition, the top five scores with highest SHAP feature importance include Nucleotide BDM and NuPoP (Affinity). These two features display similar correlations between feature value and SHAP value across Figures 2.3 and 2.4. Notably, low Nucleotide BDM values and high NuPoP (Affinity) values correspond to negative impact on off-target activity. As for the three CRISPRspec binding energy scores, they attain comparable SHAP feature importance to top-performing nucleosome organization-related scores in both models.

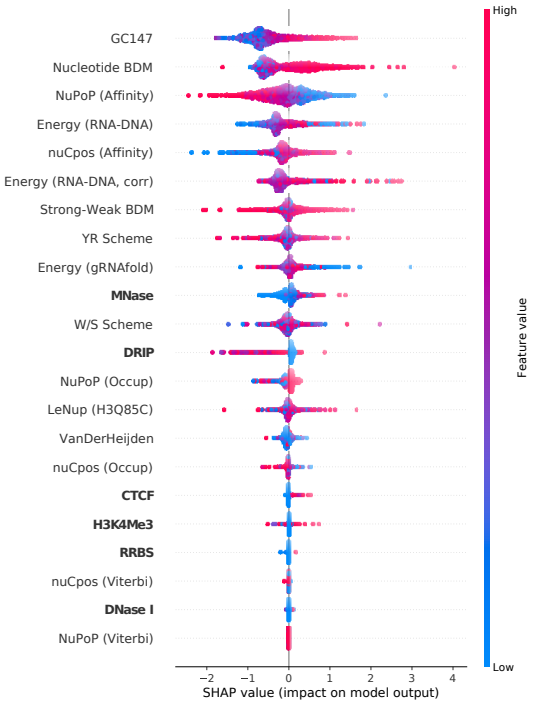


Figure 2.3: SHAP summary plot for the trained extreme gradient boosted (XGBoost) tree model. The model's input consists of three CRISPRspec-derived energy terms, six experimental epigenetic scores (bolded), and 13 computed nucleosome organization-related scores. The three CRISPRspec-derived energy terms are $E_{\text{RNA-DNA}}$, $E^{\text{corr}}_{\text{RNA-DNA}}$ and E_{gRNAfold}. The six experimental epigenetic scores are CTCF, DNase I, DRIP, H3K4me3, MNase and RRBS. The 13 computed nucleosome organization-related scores are GC147 [207], W/S scheme, YR scheme [208, 209], Strong-Weak BDM, Nucleotide BDM [205, 206], NuPoP (Occupancy), NuPoP (Affinity), NuPoP (Viterbi) [210], nuCpos (Occupancy), nuCpos (Affinity), nuCpos (Viterbi) [211], VanDerHeijden [212] and LeNup (H3Q85C) [213]. The base pair-resolved SHAP contributions for each data point are summed for each computed nucleosome organization-related score.

2.3 Discussion

MNase-seq, a common genome-wide experimental technique, appears to be an attractive option for obtaining raw nucleosome occupancy data. In addition, nucleosome occupancy

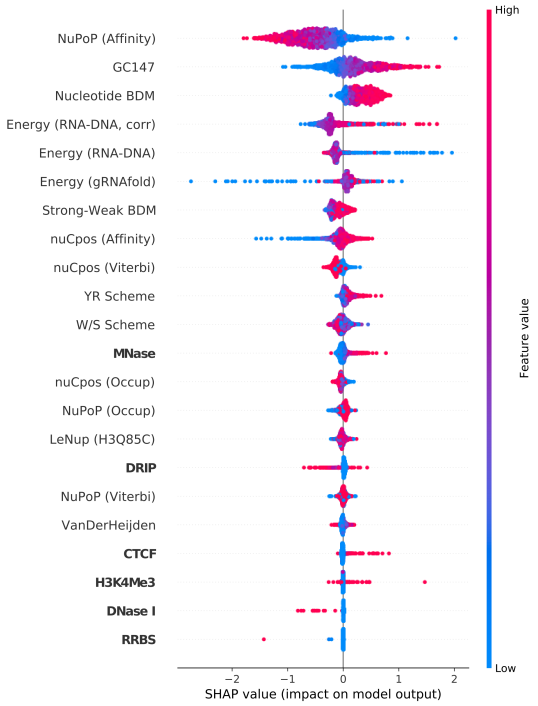


Figure 2.4: SHAP summary plot for the trained convolutional neural network (CNN) model (see Appendix Figure A.1 for architecture details). The model's input consists of three CRISPRspec-derived energy terms, six experimental epigenetic scores (bolded), and 13 computed nucleosome organization-related scores. The three CRISPRspec-derived energy terms are $E_{\text{RNA-DNA}}$, $E^{\text{corr}}_{\text{RNA-DNA}}$ and E_{gRNAfold}. The six experimental epigenetic scores are CTCF, DNase I, DRIP, H3K4me3, MNase and RRBS. The 13 computed nucleosome organization-related scores are GC147 [207], W/S scheme, YR scheme [208, 209], Strong-Weak BDM, Nucleotide BDM [205, 206], NuPoP (Occupancy), NuPoP (Affinity), NuPoP (Viterbi) [210], nuCpos (Occupancy), nuCpos (Affinity), nuCpos (Viterbi) [211], VanDerHeijden [212] and LeNup (H3Q85C) [213]. The base pair-resolved SHAP contributions for each data point are summed for each computed nucleosome organization-related score.

data may be indicative of CRISPR-Cas9 off-target activity. On account of this, we sought to obtain MNase-seq data from NucPosDB [214] where available for human cell lines. Nonetheless, we found MNase-seq data only for U2OS, K562 and HeLa in NucPosDB. In particular, MNase-seq data must be measured for each cell line of interest in order to curate sufficient data for analysis. This makes MNase-seq data cell-based and difficult to obtain. Such qualities are the opposite of computed nucleosome organization-related scores, which are not only genome-wide but also easy to obtain and cell-line independent.

The experimental features CTCF, DNase I, RRBS and H3K4me3 are commonly used as input features in multiple state-of-the-art deep learning-based CRISPR-Cas9 off-target activity prediction tools [52, 215, 155]. Despite this, we can see in Figure 2.1 and Table 2.1 that BDM-based scores attain much higher Spearman and Pearson correlations with off-target cleavage activities. This is in contrast to the six experimental epigenetic features listed in Table 2.1, which do not strongly correlate with cleavage activity.

Scrutinizing the distributions in Figure 2.2, we observe that most off-target sites with low Nucleotide BDM value fall under the lowest cleavage activity bin CA$=-4$. Moreover,

crisprSQL is augmented with sequence alignment-derived putative off-target sites. Such putative sites are assigned the lowest cleavage activity value CA= -4 on the assumption that such sites have no off-target activity. As a result, among the putative sites, Nucleotide BDM is better at separating sites without activity from sites with activity, compared to other epigenetic features. The aforementioned observations can be explained by the correspondence between low Nucleotide BDM values and proximity to nucleosome dyad positions [206]. Since these positions are blocked by nucleosomes, they are inaccessible for Cas9 binding and cleavage, thus resulting in low off-target activity. This is a possible explanation on why these off-target sites have not been experimentally identified as active. In practice, the application of Nucleotide BDM for data filtering can be useful when preparing data for CRISPR-Cas9 off-target model training. This is because such a filtering might help resolve any class imbalances between experimentally measured and putative off-targets. Moreover, Nucleotide BDM is a fundamental property of the 147 bp nucleosomal DNA sequence which is not dependent on any training dataset. Deeper understanding of why augmented datapoints (i.e., lowest cleavage activity datapoints) have no off-target activity is currently lacking. To the best of our knowledge, there has not been any existing target sequence-based measure that could separate augmented datapoints from experimentally-derived datapoints. Figure 2.2 indicates that low values of Nucleotide BDM can separate these augmented datapoints remarkably well compared to other similar measures.

In Figures 2.3 and 2.4, the six experimental scores' low SHAP feature importance demonstrates that they are inappropriate for informing off-target cleavage activity prediction models. This corroborates with the Spearman and Pearson correlation values in Table 2.1. The top five scores with highest SHAP feature importance include Nucleotide BDM and NuPoP (Affinity). The two features show similar correlations between feature value and SHAP value across the two plots. Notably, low Nucleotide BDM values and high NuPoP (Affinity) values correspond to negative impact on off-target activity. This observation corroborates the fact that such feature values often are signals of positioned nucleosomes. It follows that information in BDM-based scores and NuPoP (Affinity), alongside other nucleosome organization-related scores, are well suited for informing off-target cleavage activity prediction models. The importance of GC147 (see subsubsection "GC Content" in subsection 2.5.4) as a feature in both machine learning models is in agreement with latest findings [216] that CRISPR–Cas9 bends DNA to read its sequence. Specifically, DNA bendability is very highly correlated with GC content [217]. Such a fact could explain the findings of the SHAP summary plot, namely that high GC147 has a positive impact on (off-)target cleavage activity. The three CRISPRspec binding energy scores contribute significantly towards model predictions in both models, which confirms these scores' usefulness for CRISPR-Cas9 off-target activity prediction. Despite interesting structures in the heatmaps of Appendix Figures A.7 and A.8, a thorough analysis of such structures is beyond the scope of this study. In off-target prediction, the most suitable use case for Nucleotide BDM and other relevant measures is to incorporate them in 'complete' deep learning models. Together with measures like NuPoP (Affinity) and GC147, they can be combined with the guide-RNA–(off-)target DNA sequence pair as input to such models.

Interestingly, only BDM-based scores have noticeable correlation with (off-)target activities. However, the NuPoP (Affinity) score has comparable SHAP feature importance to Nucleotide BDM score in both machine learning models considered in this work. Appendix Figure A.9 shows that the correlation between NuPoP (Affinity) and Nucleotide BDM is relatively low. This observation agrees with the finding that only one of the two

scores (Nucleotide BDM) correlates with (off-)target activity. However, it does not alone explain why the other score, NuPoP (Affinity), is still a comparably impactful feature in both machine learning models. To investigate this further, we obtained SHAP dependence plots for both models, which include NuPoP (Affinity) and Nucleotide BDM (see Appendix Figure A.11 and A.12). These plots show that a given NuPoP (Affinity) value can have different impact (importance) based on the corresponding Nucleotide BDM value of a data point. This last observation explains why NuPoP (Affinity) does not noticeably correlate with (off-)target activity, yet is an important feature for both models, since they include NuPoP (Affinity) and Nucleotide BDM scores simultaneously.

Our results indicate that only a few out of 13 nucleosome organization-related scores show noticeable correlation with (off-)target activity or are important for model predictions. Most of these high-importance features 'measure' nucleosome affinity rather than nucleosome occupancy. Consequently, we speculate that the influence of high nucleosome affinity on Cas9 (off-)target activity exceeds that of high nucleosome occupancy. Such speculation is in concordance with the low impact of the NuPoP (Occupancy) score (see Figures 2.3 and 2.4) on model predictions.

2.4 Conclusions

For all off-target sites featured in the crisprSQL Cas9 off-target database, we obtained 19 epigenetic features, 15 of which were newly considered. The introduced computed features characterize nucleosome organization, and include features based on BDM-based or NuPoP (Affinity). We also considered six experimental epigenetic features, namely CTCF, DNase I, DRIP, H3K4me3, MNase and RRBS. We showed that the computed features exhibited considerably larger correlation with off-target cleavage activity when compared to the six experimental epigenetic features. Interestingly, only the features CTCF, DNase I, H3K4me3 and RRBS have been frequently used in deep learning-based off-target activity prediction models. As expected, nucleosome positioning negatively impacts off-target activity. This is shown by the low Nucleotide BDM scores assigned to putative off-target sites with no detectable off-target activity. We explain this phenomenon by the presence of positioned nucleosomes which inhibit Cas9 binding. Including empirical estimates of sgRNA-DNA heteroduplex binding energies as inputs, we constructed an XGBoost tree and a CNN model. The two models were used in order to gain feature importance values of all epigenetic features. Next, we created a SHAP summary plot for each model, with feature contribution quantified by the average SHAP feature importance value across data points. The plots showed GC147, Nucleotide BDM and NuPoP (Affinity) as features among the top five which contribute most to the model's output in both models. Their importance in the two models are unlike the six experimental epigenetic scores. We uploaded the off-target cleavage activity dataset used in order to make the experimental epigenetic and computed nucleosome organization-related scores available for further research. This dataset can be found as a compressed Parquet file at `https://crisprsql.com/downloads/260520_putative_nucleosomal.parquet.gz`. For future work, computed scores could be combined with target sequence and binding energy features in more robust and complete CRISPR-Cas9 off-target activity prediction models. Notably, BDM-derived and NuPoP scores could be used in such models. It would also be fruitful to scrutinize whether BDM-derived and NuPoP (Affinity) are also predictive of off-target activity in other CRISPR-Cas systems.

2.5 Methods

2.5.1 crisprSQL

The crisprSQL database consists of experimental off-target sites and cleavage activities from 15 human CRISPR-Cas9 off-target studies. In order to conduct a comprehensive investigation on the effect of epigenetics and nucleosomes on CRISPR-Cas9 off-target activity, we utilize crisprSQL [54]. crisprSQL is an up-to-date Cas9 off-target database containing sequence and epigenetic information for over 25,000 gRNA-off-target pairs from various human and rodent cell lines. Different experimental techniques were used to measure off-target activity in different studies. Consequently, we combine the experimental off-target cleavage activities from each study by applying a Box-Cox transformation. The transformation is such that the resulting combined cleavage activity data approximates a Gaussian with mean = 0 and standard deviation = 2, as suggested in [54]. Transformed values were clipped to the $[-4, 4]$ range, with cleavage activity values below the lowest reported assay accuracy of 10^{-5} set to -4. We furthermore augment the sites in crisprSQL with those in the respective genome which have less than seven mismatches compared to any experimental data point. These data points are assumed to have no off-target activity (CA = -4). Using the sequence alignment tool batmis for this [218], we generate 226,682 augmented data points. This results in a total of 251,854 data points in our dataset. In summary, the above steps yield a crisprSQL-derived dataset which was augmented with putative off-targets.

2.5.2 Experimental Nucleosome Occupancy Data

The NucPosDB database [214] consists of experimental nucleosome positioning and occupancy data aggregated from various biological publications. Micrococcal Nuclease digestion with deep sequencing (MNase-seq) data are indicative of nucleosome occupancy and chromatin accessibility. In addition, MNase-seq may be indicative of CRISPR-Cas9 off-target activity. Consequently, MNase-seq data for human cell lines present in crisprSQL are extracted from NucPosDB where available. This yields three HeLa (GSM1602359 [219], GSM2680344-2680347 [220]), five K562 (GSE78984 [221], GSM920557 [222], GSM2083137-2083140 [221]) and two U2OS (GSM1838910-1838911 [223]) MNase-seq tracks. Such tracks for HeLa, K562 and U2OS are then used for annotating crisprSQL off-target sites observed in the corresponding cell line.

2.5.3 Adding Epigenetic Scores

To construct the dataset for our study, we extract the 23bp target DNA sequence and 169bp target-centered sequence context for all gRNA-target pairs in crisprSQL. We also extract the experimental epigenetic (i.e., CTCF, DNase I, H3K4me3 and RRBS) scores and the normalized off-target cleavage activity for all aforementioned gRNA-target pairs. To create a single experimental epigenetic MNase feature from the cell-specific tracks, we first average HeLa data from replicate tracks GSM2680344 and GSM2680345. Next, we average U2OS data from replicate tracks GSM1838910 and GSM1838911, and directly adopt GSM2083140 for K562. We then linearly rescale each of the three resulting sets of MNase data to $[0, 1]$, and concatenate the sets together into a single feature. We assign zeros to off-target sites with no available MNase data. In summary, this yields a crisprSQL-derived dataset with 6 experimental epigenetic scores for each of the experimental and putative off-target sites.

2.5.4 Adding Nucleosome Organization-Related Scores

Various existing procedural and training-based data-driven computational tools are used for predicting nucleosome organization-related scores such as nucleosome occupancy and positioning. Whereas training-free procedural tools are adopted wherever available, only three recently developed training model-based tools, namely, NuPoP [210], nuCpos [211] and LeNup, were adopted. This is because these tools attain similar performances to the gold standard nucleosome occupancy model from Kaplan et al. [224, 225]. Alternatively, they use chemical cleavage-based nucleosome positioning data [211, 226] which have higher resolution compared to the MNase-seq data used in the gold standard model.

We further augment the crisprSQL dataset with nucleosome organization-related scores. This is done by computing nucleosome occupancy and/or positioning-related scores for each base pair in the 23bp target sequence for all off-target sites. To compute a variety of scores for each 169bp sequence context, we use a comprehensive set of nucleosome organization-related tools. The names of these tools are GC content (abbreviated GC147) [207], W/S scheme [208, 209], YR scheme [208, 209], Van Der Heijden [212], Block Decomposition Method (BDM) [205, 206], NuPoP [210], nuCpos [211], and LeNup [213]. Note that nucleosome organization-related tools like BDM [205] cannot handle 'N'-containing input sequences. As a result, the dataset used in this study only consider off-target sites with non-'N'-containing sequence contexts.

The following subsections details how each tool is used for computing one or more nucleosome organization-related scores. Since NuPoP and nuCpos both output histone affinity, nucleosome occupancy, and Viterbi scores, we include all three scores as separate features for both tools. We also derive Nucleotide BDM and Strong-Weak BDM scores from BDM. As a result, the 8 tools above generate 13 computed scores. In summary, the above steps yield a crisprSQL-derived dataset which was augmented with putative off-targets. In terms of features, it has a total of 6 experimental epigenetic and 13 nucleosome organization-related computed features. We further refer to these 19 features as epigenetic features.

GC Content

GC content (or GC147 as abbreviated here for clarity) is a simple training-free measure. It is defined as the fraction of guanine and cytosine residues present in the 147 bp nucleosomal sequence around a given nucleotide. Details on the use of GC content for predicting nucleosome occupancy can be found in Appendix subsection A.2.1.

We compute base pair-resolved GC147 values for each (off-)target site in the crisprSQL dataset. To do this, we slide a 147 bp (= 73 + 1 + 73) window across the (off-)target site's 169bp (= 73 + 23 + 73) context sequence, thereby obtaining 23 subsequences of length 147. A GC147 value is then computed for each of these subsequences.

W/S and YR Schemes

W/S and YR schemes are training-free scores used for the prediction of rotational and translational nucleosome positioning, respectively [208]. The two schemes are available on the web platform nuMap [209], and are based on sequence-dependent DNA anisotropy. Details regarding how W/S and YR schemes work can be found in Appendix subsection A.2.2.

We compute base pair-resolved W/S and YR Scheme values for each (off-)target site in the crisprSQL dataset. The general approach for doing this is identical to that of GC147. Namely, we slide a 147 bp window across the (off-)target site's 169bp context sequence,

thereby obtaining 23 subsequences of length 147. The only difference is that we use W/S and YR Scheme instead of GC147 when computing values for each of the 23 subsequences.

Van Der Heijden Algorithm

In reference [212], the authors propose a method for predicting the intrinsic nucleosome position of a genome based on statistical mechanics. We abbreviate this method as VanDerHeijden. Details regarding how VanDerHeijden works can be found in Appendix subsection A.2.3.

We compute base pair-resolved VanDerHeijden values for all (off-)target sites in the crisprSQL dataset. To compute a VanDerHeijden score for a given (off-)target site, we first obtain the 169bp context sequence of the given site. The context sequence is then padded with 73 A nucleotides on both ends, and then passed into the Van Der Heijden algorithm (see Appendix subsection A.2.3). Reading the middle 23 values in the array of 169 values produced by the algorithm then yields the base pair-resolved values. We use the following hyperparameters for VanDerHeijden:

- a nucleosome positioning window of $N = 147$,
- probability amplitude $B = 0.16$,
- dinucleotide periodicity $p = 10.1$, and
- chemical potential $\mu = -0.6$.

An implementation of the algorithm can be found at `https://github.com/JvN2/NucTool`.

Block Decomposition Method-based Measures

Many recent nucleosome occupancy tools such as NuPoP are statistical and entropy-based. However, such tools often require the use of experimental nucleosome occupancy data for the training of many parameters in the model, which is computationally expensive. To resolve this, we can use the Block Decomposition Method (BDM) [205], which is a training-free method for approximating the algorithmic complexity of sequences. A consequence of this definition is that repetitive sequences, e.g., "ATATATATAT", have low BDM values. A recent study [206] showed that BDM scores of 147 bp candidate DNA sequences carry valuable information related to nucleosome organization.

Based on BDM, we derive Nucleotide BDM, which computes the BDM of the 147 bp DNA string. We also derive Strong-Weak BDM, which applies the strong-weak transformation before computing the BDM of the resulting modified string. The strong-weak transformation replaces 'G' and 'C' with 'S' (Strong) and 'A' and 'T' with 'W' (Weak) in the DNA string. We compute base pair-resolved Nucleotide BDM and Strong-Weak BDM values for each (off-)target site in the crisprSQL dataset. The general approach for doing this is identical to that of GC147. Namely, we slide a 147 bp window across the (off-)target site's 169bp context sequence, thus obtaining 23 subsequences of length 147. We then use PyBDM, a Python [227] implementation of BDM, to compute Nucleotide BDM and Strong-Weak BDM values for each of the 147 bp subsequences. The Python implementation of BDM can be found in `https://github.com/sztal/pybdm`.

NuPoP

Using a duration Hidden Markov Model (dHMM), NuPoP [210] predicts nucleosome positioning and occupancy. NuPoP accounts for the different linker length distributions or

base compositions in different eukaryotes in order to make better predictions [228]. Details on NuPoP can be found in Appendix subsection A.2.5. An implementation of NuPoP can be found at `https://github.com/jipingw/NuPoP`.

We compute base pair-resolved NuPoP (Affinity), NuPoP (Occupancy) and NuPoP (Viterbi) values for each (off-)target site in the crisprSQL dataset. First, the 294,989 context sequences in the crisprSQL dataset were split into 9 sets of size 31,645 and 1 set of 10,184. This is to accommodate the fact that NuPoP requires an input sequence length of at least 1000bp. Long strings of length $147 + (147 + 169) * 31,645 = 9,999,967$ were created for the first 9 set by adding 147 A nucleotides between each context sequence. To remove end effects, the long string also contains 147 A nucleotides both before the first context sequence and after the last context sequence. In the same way, a short string of length $147 + (147 + 169) * 10,184 = 3,218,291$ is created for the final set. The 10 long strings are then fed into the NuPoP R package individually using the `predNuPoP` function. This gives rise to 10 TSV files containing the base pair-resolved histone binding affinity, occupancy and Viterbi values. When calling `predNuPoP`, we use parameters `species=1` and `model=4`.

nuCpos

Building on NuPoP, nuCpos [211] is a recent dHMM-based algorithm for predicting nucleosome positioning. nuCpos uses the same training and inference algorithms as NuPoP. However, it improves upon NuPoP by using high-resolution H3Q85C-seq budding yeast data [226] instead of the low-resolution MNase-seq data. Similar to NuPoP, nuCpos produces histone binding affinity, predicted nucleosome occupancy and Viterbi scores. More details on the algorithm can be found in [211]. An implementation of nuCpos can be found at `https://github.com/hkatomed/nuCpos`.

We compute base pair-resolved nuCpos (Affinity), nuCpos (Occupancy) and nuCpos (Viterbi) values for each (off-)target site in the crisprSQL dataset. The nuCpos R package has similar input-output interfaces to NuPoP. Consequently, we use the same approach as that described for NuPoP above in order to produce these base pair-resolved values. When calling `predNuCpos`, we use parameters `species="sc"`, `smoothHBA=FALSE` and `ActLikePredNuPoP=TRUE`.

LeNup

In light of the recent rise of state-of-the-art deep learning methods for data-based models, LeNup uses a convolutional neural network (CNN) with gated Inception-like modules [229, 230]. LeNup is used for nucleosome positioning prediction in a variety of eukaryotic genomes [213]. The original implementation of LeNup is available at `https://github.com/biomedBit/LeNup`.

LeNup was originally trained for separating nucleosomal and non-nucleosomal DNA. Consequently, we retrained the neural network used in LeNup using high resolution H3Q85C chemical cleavage-seq [226] yeast data. Because of this modification, we will refer to this measure as LeNup (H3Q85C). The retrained PyTorch [141] model can be found at `https://github.com/jeffmak/crispr-cas9-epigenetics`. We compute base pair-resolved LeNup (H3Q85C) values for all (off-)target site in the crisprSQL dataset. For each (off-)target site, we one-hot encode its context sequence and pass it into the PyTorch model, which outputs the base pair-resolved value.

2.5.5 Correlation and Distribution Analysis

We compute the Spearman and Pearson correlations with off-target cleavage activities for all epigenetic features. This enables us to examine the relationship between each epigenetic feature and off-target cleavage activity, and to identify features which significantly correlate with off-target activity. We also consider whether such correlations vary between gene and non-gene bodies or across cell lines. The calculation of gene bodies is not cell line dependent. The nucleosome organization-related scores are at base-pair resolution. Consequently, we take the mean of the values at each (off-)target if the score is not binary and the median of the values otherwise. Using the dataset which was augmented with putative off-targets, we separate the data points into lowest (CA $= -4$), low (CA ≤ 2) and high (CA > 2) cleavage activity. We also visualize the epigenetic score distributions for these data points. In order to compare cleavage frequencies across studies, we use the nonlinear Box–Cox transformation [231] to transform cleavage rates. We transform cleavage rates to approximate a Gaussian with zero mean and standard deviation $\sigma = 2$ for each study individually. To achieve a fixed value range and treat outliers efficiently, this distribution has been clipped at -2σ and 2σ. This has been used in the literature [57, 232] before. Based on these, we separate the data points into lowest cleavage activity (CA $= -2\sigma = -4$), low cleavage activity (CA $\leq \sigma = 2$) and high cleavage activity (CA $> \sigma$).

2.5.6 CRISPRspec

The crisprSQL database includes estimates for the free energy of the DNA-RNA heteroduplex generated by the CRISPRspec [59] biophysical interaction model. These interaction energies are features derived from secondary structures. These features shape the thermodynamic advantage to gRNA-DNA hybrid formation upon binding of the gRNA-Cas9 complex to the off-target site. Computationally, for a given (off-)target region, CRISPRspec uses four empirical free energy contributions terms, namely:

- a PAM-dependent correcting factor δ_{PAM},
- free energy $\Delta G_H^{\text{RNA:DNA}}$ from hybridizing the gRNA and target strand, weighted by a position-wise estimate of the Cas9 influence in the binding,
- free energy $\Delta G_U^{\text{RNA:RNA}}$ from forming the secondary structure of the 20nt gRNA spacer sequence, computed using RNAFold,
- free energy $\Delta G_O^{\text{DNA:DNA}}$ from forming the dsDNA duplex from the target and non-target DNA strands.

These four terms are used for computing the total binding free energy

$$\Delta G_B = \delta_{\text{PAM}}(\Delta G_H^{\text{RNA:DNA}} - \Delta G_U^{\text{RNA:RNA}} - \Delta G_O^{\text{DNA:DNA}}).$$

From the values given in the crisprSQL database, we calculate three key energy features to be included in our model, namely

- $E_{\text{RNA-DNA}} = \delta_{\text{PAM}} \Delta G_H^{\text{RNA:DNA}}$,
- $E_{\text{RNA-DNA}}^{\text{corr}} = \delta_{\text{PAM}}(\Delta G_H^{\text{RNA:DNA}} - \Delta G_O^{\text{DNA:DNA}})$,
- $E_{\text{gRNAfold}} = \Delta G_U^{\text{RNA:RNA}}$.

2.5.7 Model and SHAP

CRISPR recently saw an increase in computational tools for Cas9 off-target activity prediction [233], with recent tools using machine and deep learning techniques [162, 52, 215, 155, 234]. To determine how all 19 epigenetic scores relate to off-target activity within a Cas9 off-target cleavage activity prediction model, we build two machine learning models. The first one is an extreme gradient boosted (XGBoost) tree model [126], and the second one a convolutional neural network (CNN) model. These models take three CRISPRspec-derived energy features [59], experimental epigenetic features and nucleosomal organization-related features. The CNN's model architecture is similar to DeepCRISPR's [52] Siamese neural network, but lacks the sequence arm (see Appendix Figure A.1 for details on the architecture). Any nucleosome organization-related feature is calculated at base pair resolution leading to 23 values for an (off-)target DNA. In contrast, the mean value across the 23 (off-)target base pairs is presented for any experimental epigenetic feature.

Regarding training and evaluation for the XGBoost and CNN models, the dataset is randomly split into a training dataset and test dataset. A ratio of 80%-20% is used for the splitting. The train-test split is done in a way so as to ensure equal amounts of experimentally measured and augmented data in both datasets. For XGBoost, the tree model is trained for 70 epochs, where a new training batch with 50,000 data points is sampled in each epoch. We chose hyperparameters `eta`=0.5, `colsample_bytree`=0.7, `max_depth`=7. As for CNN, the model is trained for 70 epochs, where a new training batch with 35,000 data points is sampled in each epoch. We use hyperparameters `lr`=0.001, `batchnorm_momentum`=0.1, together with early stopping. For both models, bootstrap sampling ensures that each training batch contains equal amounts of active (CA > -4) and inactive/putative (CA $= -4$) (off-)targets. We then use the Shapley Additive Explanation (SHAP) library's Tree Explainer and Deep Explainer [146]. We use these explainers on a batch of 10,000 datapoints randomly sampled from the test data. This allows us to measure the contribution of each input feature towards the XGBoost and CNN model's prediction respectively. Contributions for each input features are then visualized using SHAP summary plots. When creating the SHAP summary plots, for each data point, we compute the SHAP contribution of each computed feature in the SHAP summary plots. The SHAP contribution for each computed feature is computed by summing up the corresponding base pair-resolved SHAP contributions.

Chapter 3

Critical assessment of 3D nanoenvironment-based rational descriptors pertinent to CRISPR-Cas9 cleavage activity

This chapter has been published in *NAR Genomics and Bioinformatics* with the citation *Mak, J. K., Bendandi, A., Salim, J. A., Mazoni, I., de Moraes, F. R., Borro, L., Störtz, F., Rocchia, W., Neshich, G., & Minary, P. (2025). Learning to utilize internal protein 3D nanoenvironment descriptors in predicting CRISPR–Cas9 off-target activity. NAR Genomics and Bioinformatics, 7(2).* `https://doi.org/10.1093/nargab/lqaf054`. All-atom molecular dynamics simulation and part of the heteroduplex stability analysis have been done by Artemi and Walter from Istituto Italiano di Tecnologia. STING descriptors for the CRISPR-Cas9 complexes involved have been produced by Goran, Ivan and José from Embrapa Digital Agriculture.

3.1 Introduction

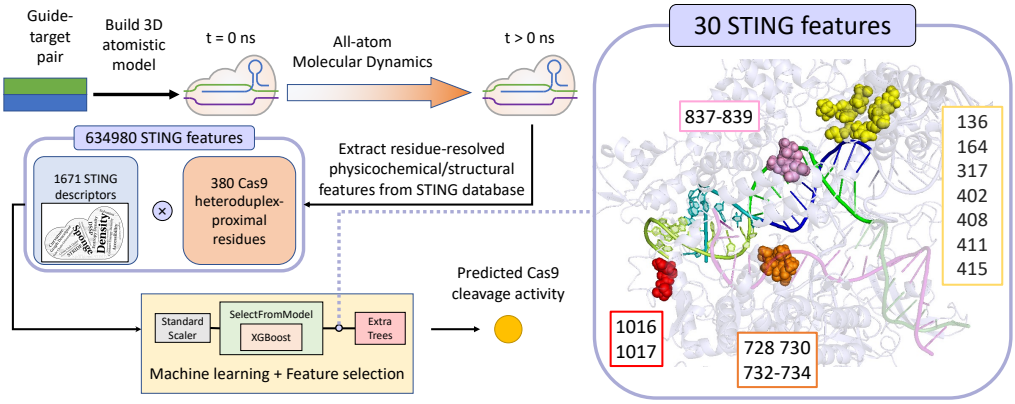


Figure 3.1: Graphical abstract for Chapter 3

Current state-of-the-art deep learning approaches [185, 181, 186, 187] for off-target activity prediction complement the sequence features with a diverse set of physically-inspired scores such as approximate energy terms [59] for sgRNA–target strand DNA (TS)

hybridization and epigenetic features essential for off-target activity [185], but have not yet directly exploited knowledge based on the information-rich internal 3D local structure (protein) environment surrounding the sgRNA–TS sequence pair, which has been investigated in various experimental studies [235, 39]. The present work aims to make the first step towards filling this gap and paves the way for a new generation of models that are rooted in the paradigms of rational design, interpretability and explainability and therefore aspires to deliver a deeper insight into the mechanistic factors that underlie (off-)target cleavage activity in CRISPR-Cas9 gene editing.

Atomistic Molecular Dynamics (MD) has been used to characterize the functioning of the CRISPR-Cas9 systems, providing trajectories and therefore a series of conformations for systems with distinct base pair mismatches at PAM-distal sites of the sgRNA-TS heteroduplex. Here, we found that the modulation of cleavage activity induced by a base pair mismatch at PAM-distal sites is captured by the internal protein 3D nanoenvironment of the sgRNA-TS pair, hereon referred to as "nanoenvironment". In particular, we studied the role of different descriptors and amino acid residues in order to build and train a ML model — named STING_CRISPR — for CRISPR-Cas9 off-target activity prediction of all possible single PAM-distal mismatches of the target of a given single guide RNA. This novel approach led to high accuracy (measured in terms of Spearman and Pearson correlations) of experimental off-target activity prediction for sgRNA-TS pairs with single PAM-distal mismatches of a given sgRNA (further referred to as studied sgRNA-TS pairs). However, our presented model unlike established models is not yet capable of predicting cleavage activity for any sgRNA-TS pair. Therefore, the current study does not aim for the development of a general CRISPR-Cas9 off-target activity prediction model but the presentation of a proof-of-concept investigation of utilizing the internal protein 3D nanoenvironment for CRISPR-Cas9 off-target activity prediction. Scikit-learn's SelectFromModel feature selection step [236] in the trained ML pipeline (Figure 3.1) revealed that density, side chain orientation, accessibility, weighted contact number entropy density, electrostatic potential, sponge, cross presence order, contact energy density, graph descriptor and solvation, measured at 23 Cas9 residues are of fundamental importance for off-target cleavage activity prediction for the studied sgRNA-TS pairs (see Appendix section B.1 for the specific definition of each descriptor). Our results lay the foundations for a new type of interpretable ML models capable of predicting CRISPR-Cas9 off-target activity.

3.2 Materials and Methods

Relying on the availability of comprehensive datasets [54], most deep learning-based CRISPR-Cas off-target activity prediction approaches aim to learn the following function:

$$f_a : S_g \times S_t \to \mathbb{R}, (s_g, s_t) \mapsto f_a(s_g, s_t), \tag{3.1}$$

where S_g and S_t are the sets of all guide and target sequences, respectively, and f_a is a function which maps guide sequence s_g and target sequence s_t to activity $f_a(s_g, s_t)$ for a particular Cas enzyme (see sequence approach in Figure 3.2A). As such, these approaches only implicitly learn the underlying physics of the CRISPR-Cas system via two-dimensional one-hot encoding of the guide-target pair. Since such tools have to use data restricted to a particular Cas enzyme (most commonly SpCas9), they are incapable of predicting changes in activity caused by amino acid residue mutations in the Cas enzyme by construction. The availability of models that predict cleavage activity based on local physical and chemical properties which can be traced back to the amino acid composition

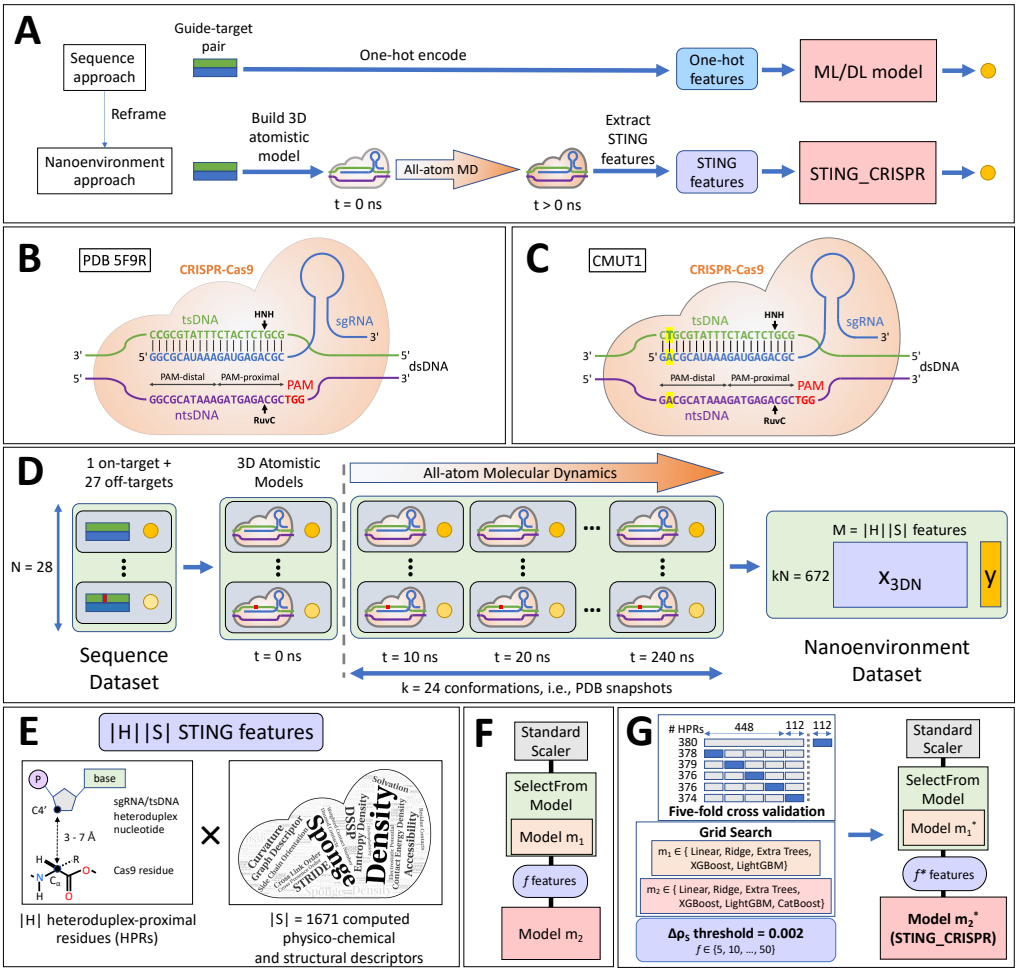


Figure 3.2: Schematic summary for obtaining STING_CRISPR, our machine learning model predicting CRISPR-Cas9 cleavage activity for the studied sgRNA-TS pairs. (A) Comparison between the purely sequence-based and the nanoenvironment-based approaches for CRISPR-Cas9 cleavage activity prediction by using a machine/deep learning model (ML/DL) and STING_CRISPR, respectively. (B, C) Catalytically-active CRISPR-Cas9 complexes with single guide RNA (blue) and double stranded DNA (target: green, non-target: purple) in PDB 5F9R crystal structure (B) and CMUT1 (C). Yellow highlights show the nucleotides mutated in CMUT1 compared to 5F9R. The leftmost PAM-distal base pair is +20, and the rightmost PAM-proximal base pair is +1. The 20 sgRNA-TS base pairs (vertical black lines) form the heteroduplex. Both DNA strands are cleaved (black arrows) by the HNH and RuvC domains of CRISPR-Cas9, respectively. (D) The three-step data pipeline for generating the residue-resolved nanoenvironment dataset from the guide-target dataset. (E) The nanoenvironment dataset contains $|H||S|$ residue-resolved STING features, namely $|S| = 1671$ STING, i.e., physico-chemical and structural, descriptors, each one evaluated at $|H|$ (i.e., between 374 and 380) sgRNA-TS heteroduplex proximal residues (HPRs). (F) Our machine learning pipeline which predicts CRISPR-Cas9 cleavage activity, with hyperparameters m_1, m_2 and f (see "Machine learning model" in subsection 3.2.3). (G) Grid search with five-fold cross validation to optimize models m_1 and m_2, followed by feature set size reduction via thresholding of Spearman correlation change ($\Delta\rho_S$) to find f^*, resulting in a pipeline with hyperparameters m_1^*, m_2^* and f^*. Shown on the top left, the number of HPRs $|H|$ vary for different train-test splits (with the training and test partitions in grey and blue, respectively) during performance evaluation and five-fold cross validation.

of the Cas enzyme would be of utmost importance as they would catalyse the development of bioengineered Cas enzymes with maximal specificity and efficiency.

To meet this objective we reframe the learning task to that of deciphering the relationship between target cleavage activity and the 3D nanoenvironment — a collection of features characterising the sgRNA-dsDNA-Cas9 complex, namely the Cas enzyme and the environment encapsulating the guide/target pair in the CRISPR-Cas9 complex (see Figure 3.2A). The 3D nanoenvironment is represented by a vector in $\mathbb{R}^M$ where M is the number of features used for characterizing the system. A vector can be derived based on a conformation of the sgRNA-dsDNA-Cas9 complex with zero or more nucleotide mutations in the sgRNA, target strand DNA (TS) and/or non-target strand DNA (NTS). We can obtain a vector for a given sgRNA-dsDNA-Cas9 complex via the following two steps: (1) construct a 3D atomistic model of the said complex; (2) obtain M residue-resolved features characterising the structural and physico-chemical properties of the complex by calculating the Sequence To and withIN Graphics (STING) features for its atomistic model (see nanoenvironment approach in Figure 3.2A). We realise that the same sgRNA-dsDNA-Cas9 complex may assume various distinct conformations each giving rise to a potentially distinct 3D nanoenvironment. Therefore as the conformation of the sgRNA-dsDNA-Cas9 complex may dynamically change so does the 3D nanoenvironment calculated from it. To account for having multiple conformations representing sgRNA-dsDNA-Cas9 complexes, we performed MD calculations to generate dynamical trajectories based on the atomistic model for each sgRNA-dsDNA-Cas9 complex and obtain the M features (representing the 3D nanoenvironment) for each of the k model conformations (snapshots) we sample from each MD trajectory (see Figure 3.2D). The implementation details of these steps are discussed in subsections "Molecular dynamics of the CRISPR-Cas9 complex with guide-target pair" and "STING descriptors for CRISPR-Cas9 complex with a guide-target pair". Given that in this study we consider $N = 28$ distinct sgRNA-dsDNA-Cas9 complexes (based on the distinct sgRNA-dsDNA pairs) and k conformations for each complex (obtained from the corresponding MD trajectories), we altogether consider kN conformations. By obtaining the 3D nanoenvironment for each of these conformations results in kN distinct 3D nanoenvironments. Furthermore, we may label each 3D nanoenvironment with the experimental cleavage activity of the corresponding sgRNA-dsDNA pair. Thus, we can obtain a labeled dataset $D = \left\{\left(x_i, a_i\right)\right\}_{i=1}^{kN}$, where $x_i \in \mathbb{R}^M$ and $a_i \in \mathbb{R}$ (see nanoenvironment dataset in Figure 3.2D). Having this labeled dataset enables us learn the relationship between 3D nanoenvironment and cleavage activity. Therefore, formally we aim to learn the following function:

$$\bar{f}_a : \Omega_{3DN} \to \mathbb{R}, x \mapsto \bar{f}_a(x) \tag{3.2}$$

where $\Omega_{3DN} \subset \mathbb{R}^M$ and $\bar{f}_a(x)$ is a functional map that takes a vector in $\mathbb{R}^M$ as input and then return a cleavage activity. The dimension M of the vector x_i can depend on the degree of detail we choose for describing the 3D nanoenvironment.

Having the dataset $\left\{\left(x_i, a_i\right)\right\}_{i=1}^{kN}$ enables us to train a regression model to decipher the relationship between experimental cleavage activity and 3D nanoenvironment. Details on the regression model with feature selection are discussed in subsection "Machine learning models for CRISPR-Cas9 cleavage activity prediction from STING descriptors".

3.2.1 Molecular dynamics of the CRISPR-Cas9 complex with guide-target pair

MD simulations were performed using GROMACS version 2020.2 [237], using bsc1 and AMBER force fields for nucleic acids and protein atoms, respectively. For water molecules, the TIP3P model was used. Protonation states of titratable residues were estimated using the pypKa server [238]. Before the production runs, structures were subjected to NVT equilibration for 400ps using the modified Berendsen thermostat, and to 1ns of NPT equilibration using the Parinello-Rahman barostat.

Targeted MD (TMD)

We chose as a reference structure for the enzyme and RNA sequence the crystal structure of the catalytically-active *Streptococcus pyogenes* Cas9, primed for target DNA cleavage, in complex with single-stranded guide RNA and double-stranded DNA (both target and non-target strands). The PDB code of this structure is 5F9R, released in 2016. 5F9R has become the most commonly used reference in the literature in recent years. Interestingly, in 2019 the 6O0Y structure was released [239]. Obtained via Cryo-EM, 6O0Y shows the conformation of the two key domains RuvC and HNH in the catalytically competent state. 5F9R and 6O0Y have the same single guide RNA sequence. However, 6O0Y is lacking some key residues and atoms. Therefore, we decided to use the structural information contained in 6O0Y to adopt the conformation of the more complete 5F9R structure. To do this, we performed all-atom explicit solvent TMD using PLUMED [240, 241, 242] as a plugin of GROMACS, in order to bring the RuvC and HNH domains of 5F9R to their catalytically active conformation, mutuated from the 6O0Y structure. More specifically, the bias was applied to the heavy atoms of the two protein domains. The collective variable used was the RMSD, using a moving restraint with κ going from 0 to 10^5 in 1.5×10^8 steps.

Reference choice and mutants generation

In order to identify our reference sequence for the analysis, we applied the following requirements by filtering the crisprSQL database [54]: having a single guide RNA sequence identical or as close as possible to that of the structural reference; having a sufficient number of singly mutated entries in the PAM-distal region of the target DNA strand; the candidate sequence and the mutated entries must have experimental off-target cleavage activity data. We therefore selected an entry which differs only in one position (RNA base number 2) with respect to the 5F9R and 6O0Y structures and fulfils the other mentioned requirements. For this entry, 28 singly mutated and experimentally annotated other entries were found in the database. We then first mutated base 2 of RNA to adenine and base 29 of the target strand DNA to thymine in our reference structure in order to make it identical to the reference sequence, and called it CMUT1 (see Figures 3.2B and 3.2C). Then we generated the same 28 mutations that were also present in the database on the DNA target strand of CMUT1. Base mutations were done using the software UCSF CHIMERA [243]. Each of them presents only a single mutation, located in the target DNA strand with respect to our reference. A table of the mutations, with associated nomenclature, can be found in Table S1.

Unbiased MD

We performed 1µs of all-atom explicit solvent unbiased MD on the TMD's output. This allowed us to evaluate the structure's dynamics and to obtain a reference which can be used in comparison with structures generated in further simulations. Using the TMD's output, we then created structures corresponding to each of the 28 guide-target pairs by mutating TS and NTS nucleotides in the TMD's output. This yielded 28 structures — one for each guide-target pair in the labelled sequence dataset. Finally, for each guide-target pair, we performed 250ns of all-atom, explicit solvent, unbiased MD, starting the MD from the structure created for the guide-target pair.

Electrostatic calculations

We performed electrostatic calculations using the Poisson-Boltzmann Equation (PBE) Finite Differences solver DelPhi [244]. We calculated the electrostatic energies (partitioned in Coulombic and reaction-field contributions) and the electrostatic potential at the atom centers in order to characterise the local potential on snapshots extracted every 10ns from the MD trajectory of each mutation. Atomic radii and charges were taken from the AMBER force field [245].

RMSD calculations

To evaluate the dynamics of the system, we calculated the RMSD of the following residues for each mutation along the MD trajectory:

- Protein residues (136, 164, 268, 317, 402, 408, 411, 415, 728, 730, 732, 733, 734, 837, 838, 839, 908, 919, 1010, 1016, 1017, 1025, 1122). These residues were selected based on the following two criteria: they either emerged as significant residues from our ML analysis (see subsubsection "Characterization of the heteroduplex-proximal CRISPR-Cas9 internal protein nanoenvironment" in subsection "Machine learning models for CRISPR-Cas9 cleavage activity prediction from STING descriptors").
- RNA and DNA bases belonging to the heteroduplex: chains B and C

We calculated the RMSD of the nucleic backbone and of the following atoms: C4 and N9 (purines); C6 and N1 (pyrimidines). We also calculated the RMSD of the phosphorus atoms and the N9 and N1 atoms (respectively). This analysis was performed using the MDAnalysis python package [246].

Structure naming scheme

Structures were given a 4-character identifier, similar to a PDB code. The first character is a letter, identifying the starting structure for the mutation. We had two kinds of starting structures, the result of our TMD (C for Cryo) and 5F9R (X for X-ray). The second character is either a number from 0 to 9 or a letter from A to Z, and identifies the specific mutation in numerical order from 0 to 9 for the first 10 mutants and then letters in alphabetical order for the remaining ones. The third and fourth character are digits which indicate the snapshot number.

Parent Descriptor Classes	Associated Neighbor Descriptor Classes
Accessibility	-
Cross Link Order (CLO)	**CLO-GN, CLO-SW, CLO-WNA, CLO-VD**
Cross Presence Order (CPO)	**CPO-GN, CPO-SW, CPO-WNA, CPO-VD**
Curvature (Curv)	**Curv-GN, Curv-SW, Curv-WNA, Curv-VD**
Density	**Density-GN, Density-SW, Density-WNA, Density-VD**
DSSP	-
Contact Energy Density (CED)	**CED-GN, CED-SW, CED-WNA, CED-VD**
Electrostatic Potential (EP)	**EP-GN, EP-SW, EP-WNA, EP-VD**
Entropy Density (ED)	**ED-GN, ED-SW, ED-WNA, ED-VD**
Graph Descriptor (GD)	**GD-GN, GD-SW, GD-WNA, GD-VD**
Hydrophobicity	-
Residue Contacts (RC)	**RC-GN, RC-SW, RC-WNA, RC-VD**
Side Chain Orientation (SCO)	**SCO-GN, SCO-SW, SCO-WNA, SCO-VD**
Solvation (Solv)	**Solv-GN, Solv-SW, Solv-WNA, Solv-VD**
Sponge	**Sponge-GN, Sponge-SW, Sponge-WNA, Sponge-VD**
STRIDE	-
Unused Contacts (UC)	**UC-GN, UC-SW, UC-WNA, UC-VD**
Weighted Contact Number	**WCN-GN, WCN-SW, WCN-WNA, WCN-VD**

Table 3.1: List of 60 STING descriptor classes (bolded) considered in this study for characterizing the internal protein 3D nanoenvironment of CRISPR-Cas9's sgRNA-TS heteroduplex. Originating from 18 parent descriptor classes (left column), the 60 descriptor classes consist of 4 parent descriptor classes (bolded, left column) and 56 neighbor descriptor classes (bolded, right column) arising from the application of Graph Neighbors (GN), Sliding Window (SW), Weighted Neighbor Average (WNA) and Voronoi Diagram (VD) aggregations to 14 other parent descriptor classes (unbolded, left column).

3.2.2 STING descriptors for CRISPR-Cas9 complex with a guide-target pair

In this study, we consider 60 physico-chemical/structural descriptor classes available from the STING platform database (see Table 3.1). The 60 descriptor classes were directly adopted from previous work which studied the internal protein nanoenvironment [199, 200, 201, 202, 203, 204]. The 60 descriptor classes translate to 1671 descriptors being organized into the relational database STING_RDB_2_CRISPR, namely one that allows the simultaneous analysis of multiple structures. A concise outline of the 1671 descriptors is included in section 1 of the SI, and full descriptions for all STING parameters/descriptors published previously on STING's web-server site can be found at `http://www.cbi.cnptia.embrapa.br/SMS/STINGm/help/MegaHelp_JPD.html` and in several papers [199, 200, 201, 202, 203]. In this work, we first adopted and then used STING SDL (Sting Descriptor Library), an in-house program able to calculate the descriptors in all possible variants (meaning, using all values for variables employed into formulas that calculate each one of STING descriptors) and applying batch calculations on the sgRNA-dsDNA-Cas9 complexes analysed in MD simulations.

These descriptors were calculated in correspondence of all atoms and in the presence of DNA or RNA bases at distances of 3, 5 and 12 Å from the phosphates for each snapshot. Atom presence lists were generated using custom Python scripts, in which atomic coordinates were parsed using Biopython [247].

3.2.3 Machine learning for CRISPR-Cas9 cleavage activity prediction from STING descriptors

Dataset

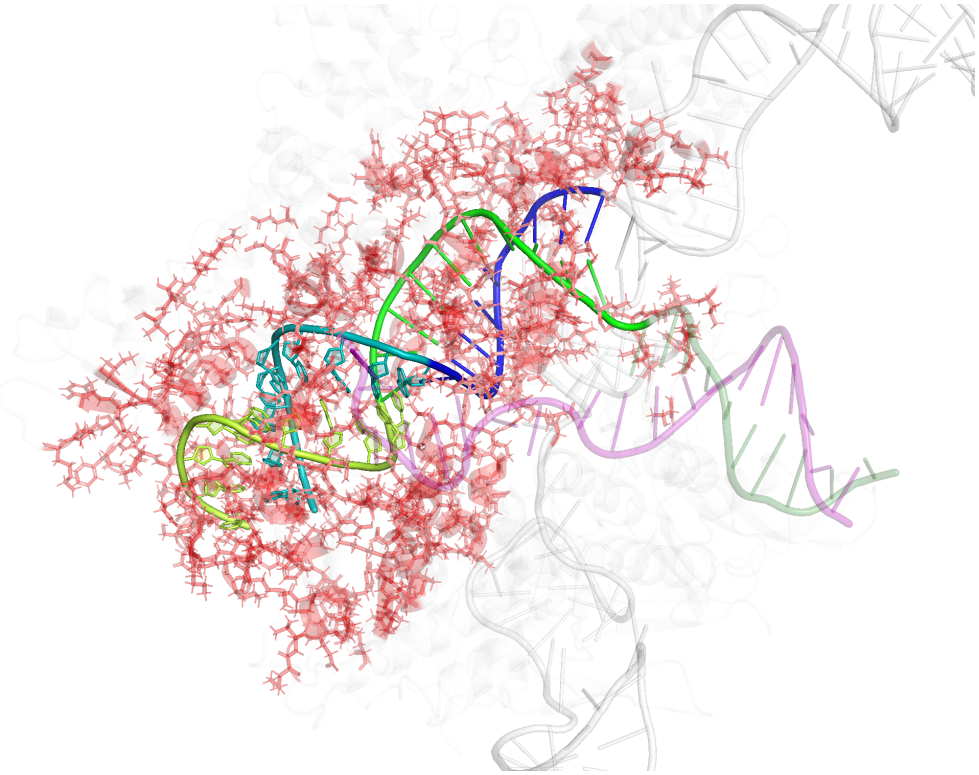


Figure 3.3: PyMOL cartoon visualization of the internal protein 3D nanoenvironment proximal to Cas9's sgRNA-target strand DNA heteroduplex in the catalytically active conformation. Cas9 residues part of the nanoenvironment are shown as red sticks, and the rest of the Cas9 residues are visualized as grey ribbons. Shown as ribbons, the color scheme is as follows for non-Cas9 components: PAM-distal sgRNA = teal, PAM-proximal sgRNA = blue, PAM-distal target DNA strand = limon, PAM-proximal target DNA strand = green, non-target DNA strand = transparent purple.

Fig. 3.2 outlines our approach for building STING_CRISPR. Namely, by generating atomistic MD trajectories and computing residue-resolved STING feature values for the atomistic model conformations, we are able to convert our labelled sequence dataset containing 1 on-target and 27 single-mismatch off-target sites into a labelled nanoenvironment dataset of size 672 (Figure 3.2D, see raw cleavage rate data in Figure B.1 and Table B.1). Labels used in both datasets are cleavage rates rather than indel frequency percentages.

We hypothesize that the internal protein 3D nanoenvironment proximal to Cas9's sgRNA-target strand DNA heteroduplex in the catalytically active conformation (drawn in Figure 3.3) is a primary factor affecting CRISPR-Cas9 cleavage activity. Moreover, a STING descriptor's value varies across Cas9 residues, as the value of a physico-chemical or structural property is always tied to a local region/district, i.e., a Cas9 residue in our case. Taking these two ideas into account, we formulate x_i as a vector of length $M = |H||S|$ (see Figure 3.2E), where:

- H denotes the set of heteroduplex-proximal residues (HPRs) whose α-carbon atoms

are $3 - 7$Å away from the C4' atoms of any sgRNA-TS heteroduplex nucleotide in at least one of the training PDB snapshots, and

- S denotes the set of 1671 STING neighbor descriptors available in STING_RDB_2_CRISPR (see Tables 3.1 and B.2) [248, 249, 250].

In other words, x_i is a vector containing features (or independent variables) defined by a given STING descriptor at a particular heteroduplex-proximal Cas9 residue, i.e., a STING descriptor-Cas9 residue pair. Since the exact set of residues in H changes depending on the set of PDB trajectory snapshots used in the residue-nucleotide distance calculations, we limit such residue-nucleotide distance calculations to training PDB snapshots when computing H to avoid data leakage when training ML models.

For STING_CRISPR, we compute 1671 physico-chemical and structural descriptors on 380 heteroduplex-proximal residues, which resulted in a nanoenvironment dataset with 634,980 STING features (Figure 3.2E, see a breakdown of the feature counts in Table B.2), where the feature values are aggregated over residues within a local neighborhood as defined by four different aggregation methods available in the STING_RDB_2_CRISPR database — Graph Neighbors (GN), Sliding Window (SW), Weighted Neighbor Average (WNA) and Voronoi Diagram (VD). See subsubsection "Training" for an explanation on how 380 heteroduplex-proximal residues were obtained for STING_CRISPR.

Exploratory analysis with heteroduplex base pair distances

For each PDB trajectory snapshot in the dataset, we compute the Euclidean distance between the two C4' atoms in each of the 19 PAM-proximal base pairs. We then use a heatmap for each off-target trajectory in order to visualize the heteroduplex base pair distances across all snapshots within each off-target trajectory. As a measure of heteroduplex plasticity, we sum all Euclidean distances across the 19 base pairs over all snapshots for all on- and off-target trajectories. To examine the relationship between this measure and CRISPR-Cas9 cleavage activity, we create violin plots for four groups of sums, namely the sums corresponding to the on-target trajectory, trajectories with low (< 0.01) activity, trajectories with medium ($0.01 - 0.1$) activity and trajectories with high (> 0.1) activity. We also create a scatter plot between the sums and cleavage activities.

Machine learning model

To decipher the relationship between experimental cleavage activity and the 3D nanoenvironment, we build an interpretable scikit-learn [236] ML pipeline (see Figure 3.2F) consisting of the following three steps:

1. StandardScaler. This scales features to zero mean and unit variance.
2. SelectFromModel utilises base model m_1 and all $|H||S|$ features to train m_1 and SelectFromModel selects the $f \ll |H||S|$ most important features from the $|H||S|$ available features.
3. Machine learning model m_2 with f input features.

Notably, we embed a feature selection step, i.e., SelectFromModel, into our pipeline, in order to combat the curse of dimensionality [251], and to ensure that f is significantly smaller than the training dataset size in our final interpretable ML model.

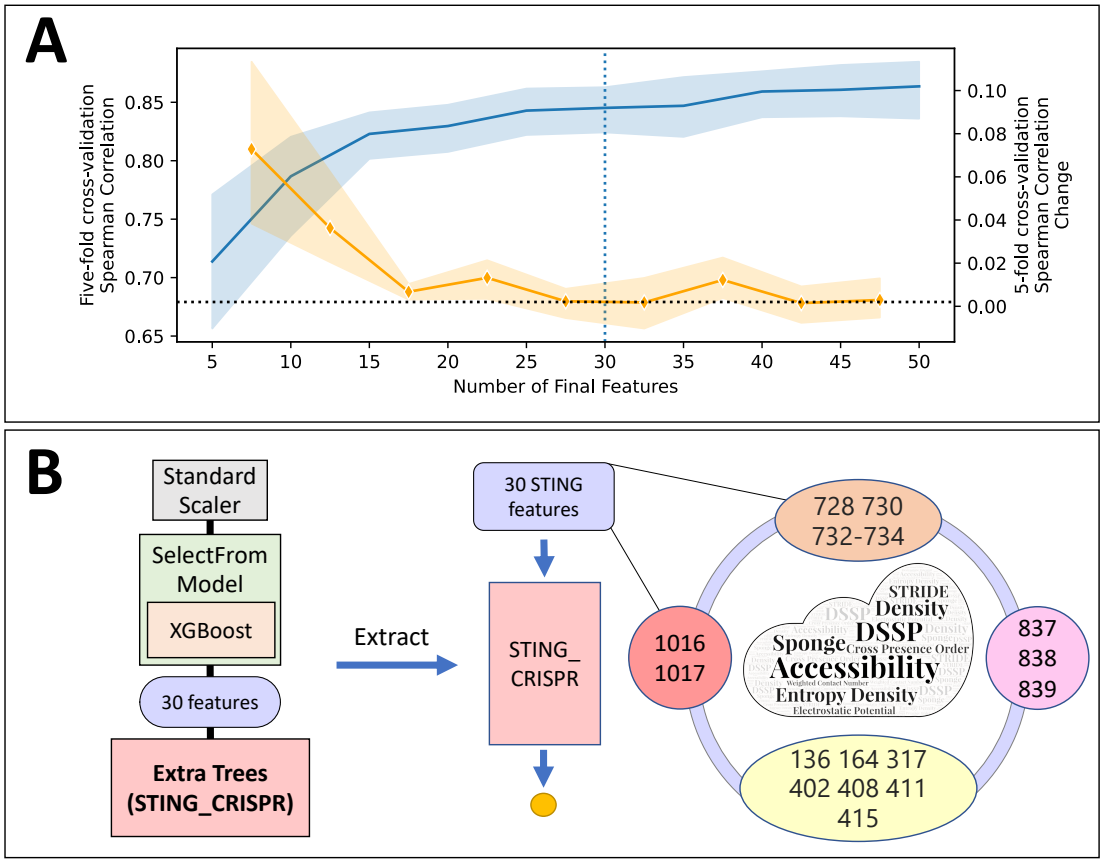


Figure 3.4: STING_CRISPR is an extra trees model with 30 STING features at 4 residue clusters. (A) Hyperparameter tuning of input feature set size in the ML pipeline after grid search with five-fold cross validation. The solid blue line (left y-axis) indicates average five-fold Spearman test correlation, and the solid orange line (right y-axis) indicates average change in the average five-fold test Spearman correlation when increasing the input feature set size in increments of 5. Black dotted horizontal line indicates the Spearman change threshold $\Delta\rho_S = 0.002$, and the blue dotted vertical line indicates the final input feature size selected. (B) Extraction of the second ML model (red, bolded) from the hyperparameter-optimized ML pipeline with m_1 = XGBoost, m_2 = extra trees and f=30 features yields STING_CRISPR, an extra trees model with 30 STING features. Among the 30 STING features, 17 of them form 4 residue clusters (defined below) found to be important in cleavage activity prediction for the studied sgRNA-TS pairs.

Training

Summarized in Figure 3.2G, the training procedure for obtaining STING_CRISPR is as follows. To prepare the data partitions, we first split the dataset into training and test partitions of size 560 and 112 by holding out the last 4 PDB snapshots in from all MD trajectories for testing. Such a split ensures that points in the training and test datasets are distributed similarly. We then randomly split the training partition into five folds for five-fold cross validation, resulting in five sets of training and validation datasets of size 448 and 112, respectively. Given that models m_1 and m_2 are tunable hyperparameters in the ML pipeline, we first perform grid search with five-fold cross validation to optimize hyperparameters m_1 and m_2 in the ML pipeline. Specifically, we use grid search to consider the following $5 \cdot 6 \cdot 10 = 300$ ML pipelines by using the following hyperparameter ranges:

- model m_1 being either a linear, ridge, XGBoost [126], extra trees [252] or LightGBM [129] model with default hyperparameters (all together 5 possibilities);
- model m_2 being either a linear, ridge, XGBoost, extra trees, LightGBM, or CatBoost [128] model with default hyperparameters (all together 10 possibilities); and
- number of possible feature size selections $|F| = 10$, where $F = \{5, 10, ..., 50\}$. We choose such an F not only because all elements $f \in F$ satisfy $f \ll |H||S|$, but also because many of the STING features are correlated, meaning that the optimal feature set size is approximately $\sqrt{448} \approx 21.2$ given a training data size of 448 during five-fold cross validation [253].

We then measure the mean five-fold Spearman correlation validation performance $\rho_S(m_1, m_2, f)$ of each combination (m_1, m_2, f), and subsequently find the model pair (m_1^*, m_2^*) with the highest validation performance when averaging the mean Spearman correlation across the 10 possible feature size selections. Once the model pair is found, we pick the smallest feature set size f^* such that increasing the selected feature set size by 5 improves the resulting mean five-fold Spearman correlation validation performance by no more than $\Delta\rho_S = 2 \times 10^{-3}$ (a hyperparameter which thresholds Spearman improvement). Using the hyperparameter configuration (m_1^*, m_2^*, f^*), we then train a single ML pipeline on all 560 points from the training partition. Once trained, we extract m_2 from the pipeline to obtain STING_CRISPR.

Since the HPR set H is dependent on the training PDB snapshots, it is worth noting that the training procedure uses 6 HPR sets, namely one for each fold in five-fold cross validation, and one extra when training the final model (see top left of Figure 3.2G for the HPR set sizes, and Appendix subsection B.2.3 "Heteroduplex-proximal residues" for the specific residues in the 6 HPR sets). In practice, HPR set sizes of 378, 379, 376, 376 and 374 are obtained for the training sets used in folds 1-5 during five-fold cross validation, respectively. When performing residue-heteroduplex nucleotide distance calculations on the entire training partition of the nanoenvironment dataset, we identify 380 sgRNA-TS heteroduplex-proximal residues (HPRs).

In practice, this grid search strategy (see the bullet points above) yields the XGBoost-extra trees combination, which has a mean five-fold cross validation Spearman correlation of 0.826 when averaged across 10 XGBoost-extra trees pipelines with 5-50 features (Figure 3.2G). Illustrated in Figure 3.4A, subsequent application of the Spearman correlation change threshold with value 0.002 on the XGBoost-extra trees combination results in a pipeline with 30 features (see Table B.3 for the list of 30 features). By setting such a threshold, we are able to minimize the feature set size without sacrificing model performance.

Together with SelectFromModel, the threshold drastically reduces the ML pipeline's feature set size from 634,980 to 30 features. By extracting the cleavage activity model from the ML pipeline (see Figure 3.4B), we obtain an extra trees model with 30 features, which we name as STING_CRISPR (Figure 3.4, red vertical box) in this study. In summary, from the nanoenvironment dataset, the training procedure produced a ML pipeline which feeds the top 30 most important STING features selected from the trained XGBoost surrogate model into an extra trees model for Cas9 activity prediction.

Evaluation

We record STING_CRISPR's performance on the test dataset for the following metrics: Spearman correlation, Pearson correlation, mean squared error and mean absolute error. Using test data, we also use bar plots to visualize the mean and standard deviation of the square errors between predicted and actual cleavage activities for the on-target interface, PAM-distal mismatch positions and mismatch interface types.

Model Interpretation

Our framework for interpreting STING_CRISPR is founded on feature counts and SHapley Additive exPlanations (SHAP) [146] (a summary of the theory behind SHAP can be found in the Background chapter). Using STING_CRISPR and the SHAP TreeExplainer model [254], we obtain SHAP values ϕ for all PDB snapshots in the ML dataset, where $\phi_j^{(i)}$ denotes the SHAP value assigned to the jth feature for the ith datapoint. We also obtain the SHAP importance of each features in STING_CRISPR, where the SHAP importance of the jth feature is given by $I_j = \frac{1}{|D|}\sum_{i=1}^{|D|}\phi_j^{(i)}$.

Each input feature in STING_CRISPR has the following 6 properties: an associated Cas9 residue, Cas9 domain, contiguous Cas9 domain, parent descriptor class, (neighbor) descriptor class and neighbor aggregation method. For example, the feature `Cas9_733_neighbors_side_chain_angle_3_VD` has properties Cas9 residue 733, Cas9 domain RuvC, contiguous Cas9 domain RuvC-II, parent descriptor class Side Chain Orientation (SCO), descriptor class Side Chain Orientation with VD (SCO-VD) and neighbor aggregation method Voronoi Diagram (VD). Since we can group features in STING_CRISPR by a certain property, count the number of features in each feature group, and compute the SHAP importance $I_J = \frac{1}{|D|}\sum_{i=1}^{|D|}|\sum_{j\in J}\phi_j^{(i)}|$ of each feature group J, we compute feature counts and SHAP importances for each of the feature groups arising from each of the aforementioned 6 properties, and subsequently use bar plots for data visualization.

Cas9 residues appearing far apart in the sequence space may actually be spatially proximal in the Cas9 complex. In light of this, to identify the residue clusters (i.e., hotspots) found by our training procedure, we measure the pairwise distances between two residues in STING_CRISPR averaged across the 672 PDB snapshots, and subsequently use Seaborn's clustermap algorithm to create the clusters, while setting a maximum distance of 12Å for any two residues within the same cluster. Based on these residue clusters, we compute the feature counts and SHAP importance of each residue cluster, with residues in STING_CRISPR not belonging to any residue cluster placed into the "Other" residue group. To gain spatial intuition, we use PyMOL [255] to visualize the residue clusters. Specifically, we use the last PDB snapshot from the on-target trajectory CMUT1 for visualization. For each residue-base combination formed between the STING_CRISPR residues and heteroduplex bases, we also count the number of PDB snapshots where the residue's α-carbon atom is 3-7Å away from the heteroduplex base's C4' atom, and use heatmaps to visualize the counts.

3.2.4 Evaluation of the structural impact of the mutations

The impact of the TS mutations on the overall dynamics of the system structure was evaluated by performing a parametric analysis of the stability of the most relevant residues/bases of the system. The considered parameters are average and standard deviation of the RMSD with respect to the initial conformation. Under normality assumption, the Kullback-Leibler divergences between the RMSD distributions of the residues which emerged as the most informative from the ML analysis as well as those of the bases involved in the heteroduplex complex were calculated considering as a reference the trajectory of the CMUT1 system, data shown in the SI. This allows to immediately pinpoint the sites where the difference in behavior is maximal. After doing this, a more detailed distinction was performed, separating the sites differing because of being more mobile from those differing because of being more stable.

3.3 Results

3.3.1 Structural determinants of cleavage activity

Consistency with the latest experimental structures

As more thoroughly described in the Materials and Methods, our starting structure, referred to as CMUT1 (see Figure 1C), was derived from the closest entry of the sequence database to the available structures including also the DNA and the SpCas9 (referred to as Cas9 onwards) counterparts. This structure is complete and conformationally consistent with the catalytically active structure published in [239], PDB code 6O0Y. In order to expand our analysis, we included in our evaluations also the structure published in the work by Bravo et al. [39] (PDB code 7S4X). In the latter work, catalytically active conformations of Cas9 in presence of mismatches were determined through kinetics-guided cryo-EM. Therefore, we also decided to check that the key structural features reported in this work are reflected in our analysis. Four structural features of the 7S4X structure are shown by the authors to be significant for its catalytic activity:

- Kinkedness of the RNA/DNA heteroduplex (residues B1-15 D1-20 in 7S4X; C14-30 B2-17 in CMUT1) – this characteristic is shared;
- Conformation of the L1 loop (residues A765-780 in 7S4X and in CMUT1) – the conformations are virtually identical;
- Conformation of the L2 loop (residues A906-918 in 7S4X and in CMUT1) – Average heavy atom RMSD against 250ns CMUT1 MD trajectory: 3.8 Å;
- Conformation of the RuvC loop (residues A1010-1030 in 7S4X and in CMUT1) – Average heavy atom RMSD against 250ns CMUT1 MD trajectory: 3.8 Å;

Mismatch-induced dynamical effects

We challenge the idea that a single PAM-distal mismatch between the sgRNA and the target strand DNA (TS) always destabilizes the system. This is done by comparing the RMSD distributions along the dynamics of individual sites, i.e., protein residues or sgRNA/TS bases, with respect to the corresponding distributions obtained from the dynamics of the reference structure CMUT1, which has no mismatch. Summarizing the results, which are detailed in Appendix B, we can say that point mutations in the TS result in a local destabilization of the sgRNA bases in the PAM-distal region, where they are located,

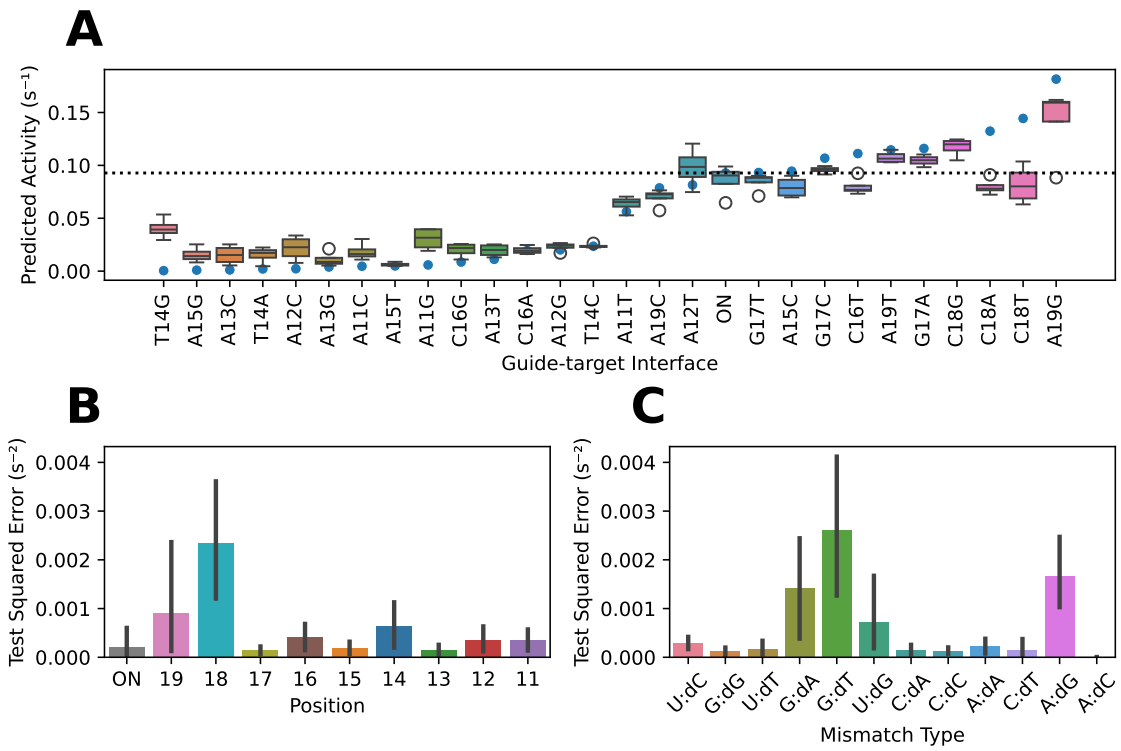


Figure 3.5: Test performance of STING_CRISPR. (A) STING_CRISPR's predicted cleavage activities for the hold-out test set containing the last 4 snapshots from each of the 28 MD trajectories. Blue dots indicate experimental cleavage activity labels for the 28 interfaces. Guide-target interfaces listed on the x-axis are sorted by increasing experimental activity. ON = on-target interface. (B) STING_CRISPR's squared error between predicted and actual CRISPR-Cas9 cleavage activity values for snapshots in the test set, categorized by being an on-target interface or a PAM-distal mismatch position. (C) STING_CRISPR's test squared error between predicted and actual CRISPR-Cas9 cleavage activity values for the different off-target mismatch interface types.

but seem also to stabilize some RNA bases in the PAM-proximal region and induce a remarkable stabilization, quantified by the RMSD standard deviation along the trajectories, of some TS bases, again in the PAM-proximal region. This finding could explain why some PAM-distal point mutations lead to increased cleavage activity. Furthermore, some degree of stabilization is observed in some Cas9 residues emerging as important from our ML approach, as shown in the stability analysis results included in Appendix B. The finding also corroborates with the positive correlation (Spearman: 0.418, Pearson: 0.503) found between heteroduplex base pair distance sums, a quantity informative on the overall stability of the guide RNA–TS heteroduplex, and CRISPR-Cas9 cleavage activities (see Figure B.8). In summary, this analysis shows that the local destabilization induced by a single mismatch between the sgRNA and the target strand DNA (TS) in the PAM-distal region can be compensated by the stabilization in other nearby positions. A possible explanation of such compensation is further elaborated in the Discussion section.

3.3.2 Test performance and model interpretation of STING_CRISPR

On the hold-out test dataset of size 112, STING_CRISPR attains a Spearman correlation of 0.819, a Pearson correlation of 0.916, a mean squared error of 5.92×10^{-4} s^{-2} and

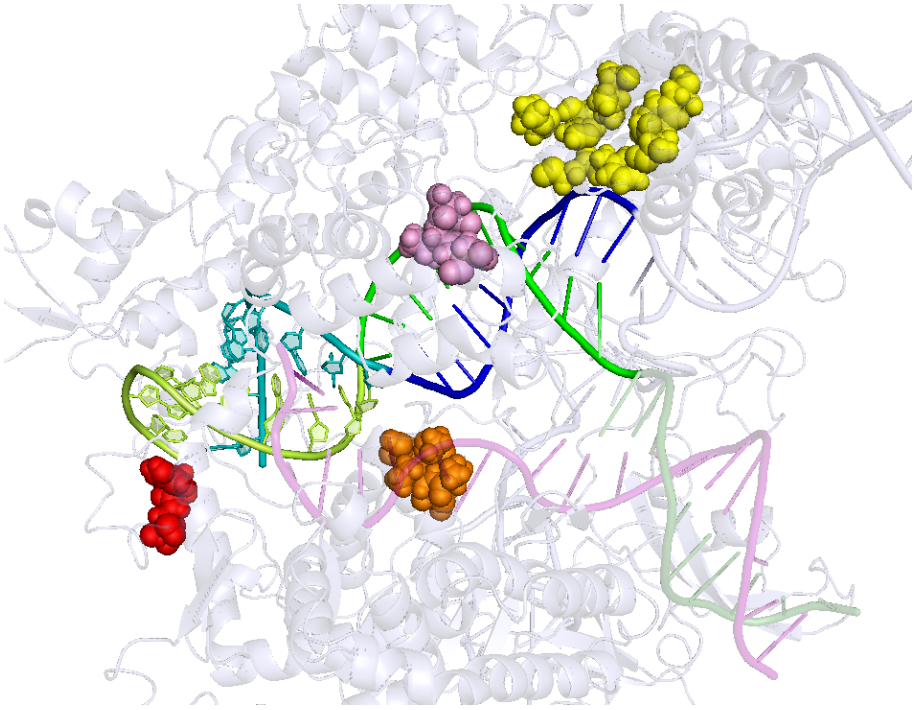


Figure 3.6: PyMOL cartoon visualization of the sgRNA-dsDNA-Cas9 complex, taken from the last (i.e., 24th) snapshot of CMUT1's MD trajectory. Shown as spheres, the 4 CRISPR-Cas9 residue clusters 136/164/317/402/408/411/415, 728/730/732-734, 837-839 and 1016/1017 are highlighted in yellow, orange, pink and red, respectively. Other parts of the Cas9 are visualized as grey ribbons. Shown as ribbons, the color scheme is as follows for non-Cas9 components: PAM-distal sgRNA = teal, PAM-proximal sgRNA = blue, PAM-distal target DNA strand = limon, PAM-proximal target DNA strand = green, non-target DNA strand = transparent purple.

a mean absolute error of 1.68×10^{-2} s^{-1}, demonstrating high model performance and affirming that residue-resolved physico-chemical/structural features can be utilized for CRISPR-Cas9 cleavage activity prediction. Ordered by increasing cleavage activity, we can see that the predicted and actual cleavage activities differ by at most 3.90×10^{-2} s^{-1} across all guide-target interfaces in this study (Figure 3.5A) apart from base mutations T14G, C18A, C18T and A19G. Such an observation is corroborated by high test square errors in positions 14, 18 and 19 (Figure 3.5B) and mismatch interface types G:dA, G:dT, U:dG and A:dG (Figure 3.5C).

Using various physico-chemical and structural descriptors, the 30 residue-resolved input features of STING_CRISPR characterize 23 Cas9 residues. The SHAP summary plot generated from STING_CRISPR using all 672 conformations shows `Cas9_733_neighbors_side_chain_angle_3_VD` as the most important feature in STING_CRISPR, where increasing its feature value increases predicted cleavage activity (see Figure B.2). Through hierarchical clustering of pairwise residue distance calculations between the C-α atoms of these 23 residues (see Figure 3.7A), we see that 17 of the 23 residues form four residue clusters, namely:

- Group 1 with residues 1016 and 1017;
- Group 2 with residues 728, 730, 732, 733 and 734;

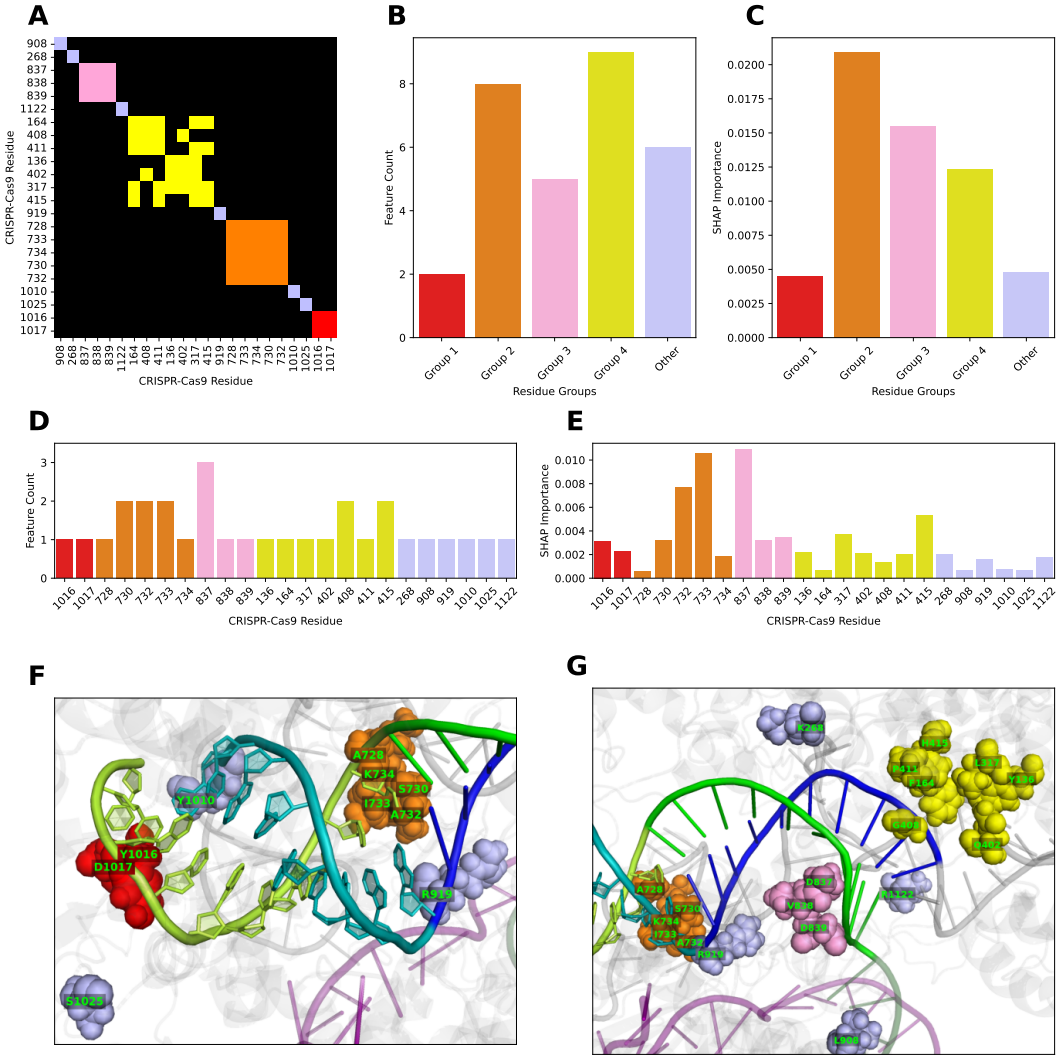


Figure 3.7: The ML pipeline identifies four residue clusters, namely Group 1 (residues 1016/1017, colored red), Group 2 (residues 728/730/732-734, colored orange), Group 3 (residues 837-839, colored pink) and Group 4 (residues 136/164/317/402/408/411/415, colored yellow). The fifth group "Other" consists of residues identified by the pipeline that do not belong to the above clusters (residues 268/908/919/1010/1015/1122, colored light blue). (A) Binarized hierarchically-clustered heatmap for the 23 Cas9 residues identified by the ML pipeline. Heatmap cells for residue pairs whose C_α atoms are less than 12 Å apart are colored according to their associated residue groups, and black otherwise. (B, C) Feature counts (B) and SHAP importances (C) of the five residue groups. (D, E) Feature counts (D) and SHAP importances (E) of the 23 important Cas9 residues, with residues grouped and colored by the five residue groups. (F, G) PyMOL cartoon visualization of the PAM-distal (F) and PAM-proximal (G) portions of the sgRNA-dsDNA heteroduplex taken from the last (i.e., 24th) snapshot of CMUT1's MD trajectory. Shown as labelled spheres, the 23 CRISPR-Cas9 important residues are colored by their residue groups. Shown as ribbons, the color scheme of other components is as follows: other parts of Cas9 = grey, PAM-distal sgRNA = teal, PAM-proximal sgRNA = blue, PAM-distal target DNA strand = limon, PAM-proximal target DNA strand = green, non-target DNA strand = transparent purple.

- Group 3 with residues 837, 838 and 839; and
- Group 4 with residues 136, 164, 317, 402, 408, 411 and 415,

which are colored red, orange, pink and yellow, respectively (see right part of Figure 3.4B and Figure 3.7F-G). Such localization of residue clusters likely indicates some biological, functional, constitutive, or structural importance within those regions. For completeness, we also group the remaining 6 residues 268, 908, 919, 1010, 1025 and 1122 to form the "Other" residue group (colored light blue).

Using these five residue groups, we see high feature counts and SHAP importances for groups 2 and 4 (Figure 3.7B-C), showing that groups 2 and 4 significantly contribute to STING_CRISPR's predicted cleavage activity. As for the feature counts of 23 residues, we see that most residues only have one feature, with residue 837 having the highest feature count of 3 (Figure 3.7D). SHAP importances vary widely between the 23 residues, with residues 733 and 837 having the highest SHAP importances. Specifically, residues 1016, 733, 837 and 415 have the highest SHAP importances in residue groups 1-4, respectively.

The residue clusters are spatially located next to different parts of the heteroduplex, and come from various Cas9 domains (Figures 3.6, 3.7F-G and B.7). Specifically, group 1 consists of RuvC residues located in the PAM-distal part of the heteroduplex, group 2 consists of RuvC residues located at the midde part of the heteroduplex, group 3 consists of HNH residues located at the catalytic site which cuts the TS, and group 4 consists of Rec I residues located on the sgRNA side of the PAM-proximal portion of the heteroduplex. As for the other residues, residues 1010 and 1025 flank group 1 on the sgRNA and TS sides, respectively. Located in the middle part of the heteroduplex, residue 919 is also spatially close to residue group 2. Using a similar approach, we also see that the four residue clusters draw features from different parent descriptor classes, which have varying SHAP importances in the different residue clusters (see Figure B.6).

To varying degrees, predictions made by STING_CRISPR are influenced by the different parent descriptor classes and Cas9 domains associated with the 30 input features. In terms of parent descriptor classes, Density, Entropy Density and Cross Presence Order have the most features, and Density, Side Chain Orientation and Accessibility have the highest SHAP importances (Figure 3.8). In terms of Cas9 domains, RuvC is shown to have the highest feature count and SHAP importance among the RuvC, HNH, REC and PAM-interacting domains. In a similar fashion, feature count and SHAP importance analysis of the four neighbor aggregation methods show that SW and VD have high feature counts and SHAP importances (Figure B.3). The same analysis but for descriptor classes show that Density with SW has highest count, but Side Chain Orientation with VD and Accessibility have the highest SHAP importance.

When considering all 672 atomistic model conformations, all residues apart from 411 and 733 are surface residues, but only residues 136, 164, 268, 402, 408, 728, 730, 919, 1016 and 1122 are interface residues according to Surfv, NACCESS and NSC (Figure B.4). In addition, in the 672 conformations, most residues are surface residues (Figure B.5A), and on average there are around 12 interfaces residues in a given conformation (Figure B.5B). In terms of SHAP importances, we see that surface residues have a much higher SHAP importances than non-surface residues (Figure B.5D), and that interface residues have less SHAP importance than non-interface residues (Figure B.5E). Averaged across the 672 PDB snapshots, 55.6%, 60.5% and 52% of the residues among the four residue clusters are residues located at the interface between Cas9 and the R-loop complex (i.e., interface residues), according to SurfV, NACCESS and NSC, respectively. When rerunning the training procedure to train on residues 3-1363 instead of just the HPRs, we find that both

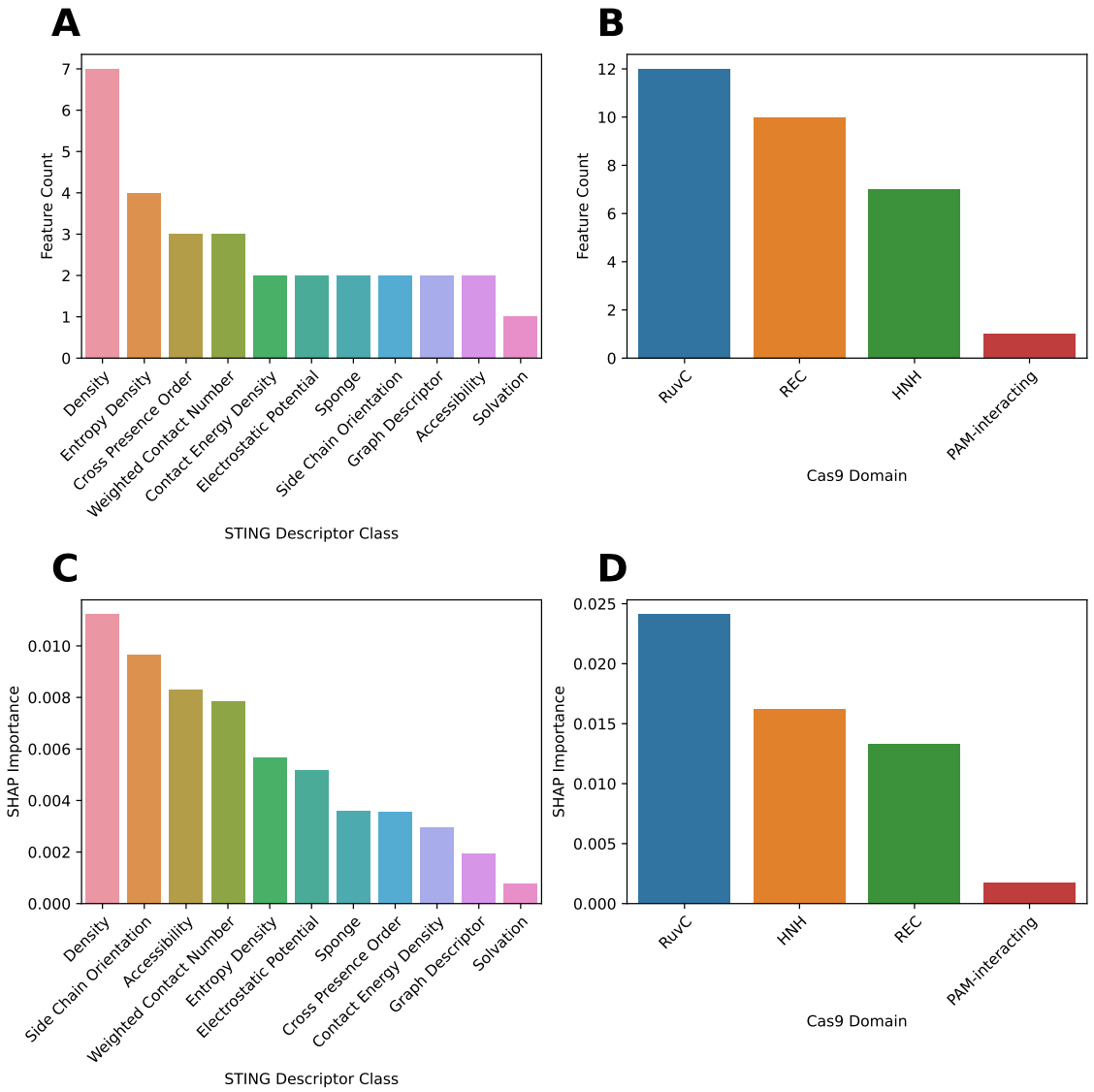


Figure 3.8: (Top) STING_CRISPR's feature counts categorized by STING descriptor classes (A) and CRISPR-Cas9 domains (B), respectively, sorted by decreasing count. (Bottom) STING_CRISPR's SHAP importance values for STING descriptor classes (C) and CRISPR-Cas9 domains (D), respectively, sorted by decreasing SHAP importance. Only STING descriptor classes or Cas9 domains with non-zero count or SHAP importance are shown.

the feature count and the SHAP importance of HPRs are higher than those of non-HPRs (Figures B.5C and B.5F).

3.3.3 Test performance when generalizing to unseen guide-target interfaces

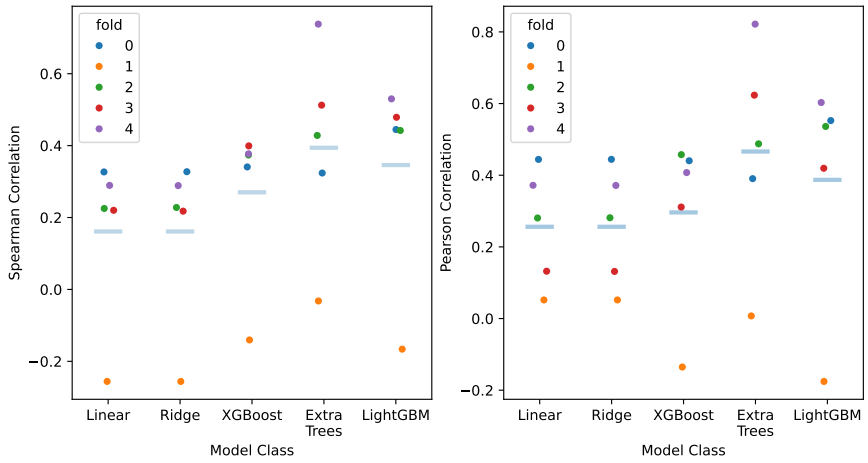


Figure 3.9: 5-fold cross validation Spearman (left) and Pearson (right) correlation performance when using linear regression, ridge regression, XGBoost, Extra Trees and LightGBM. Test sets for each cross validation fold was constructed by binning snapshots associated with the trajectory with the nth lowest cleavage activity into the test partition of fold $n \mod 5$, and into the training partition in the other folds. Light blue horizontal line represents the mean correlation across the 5 folds.

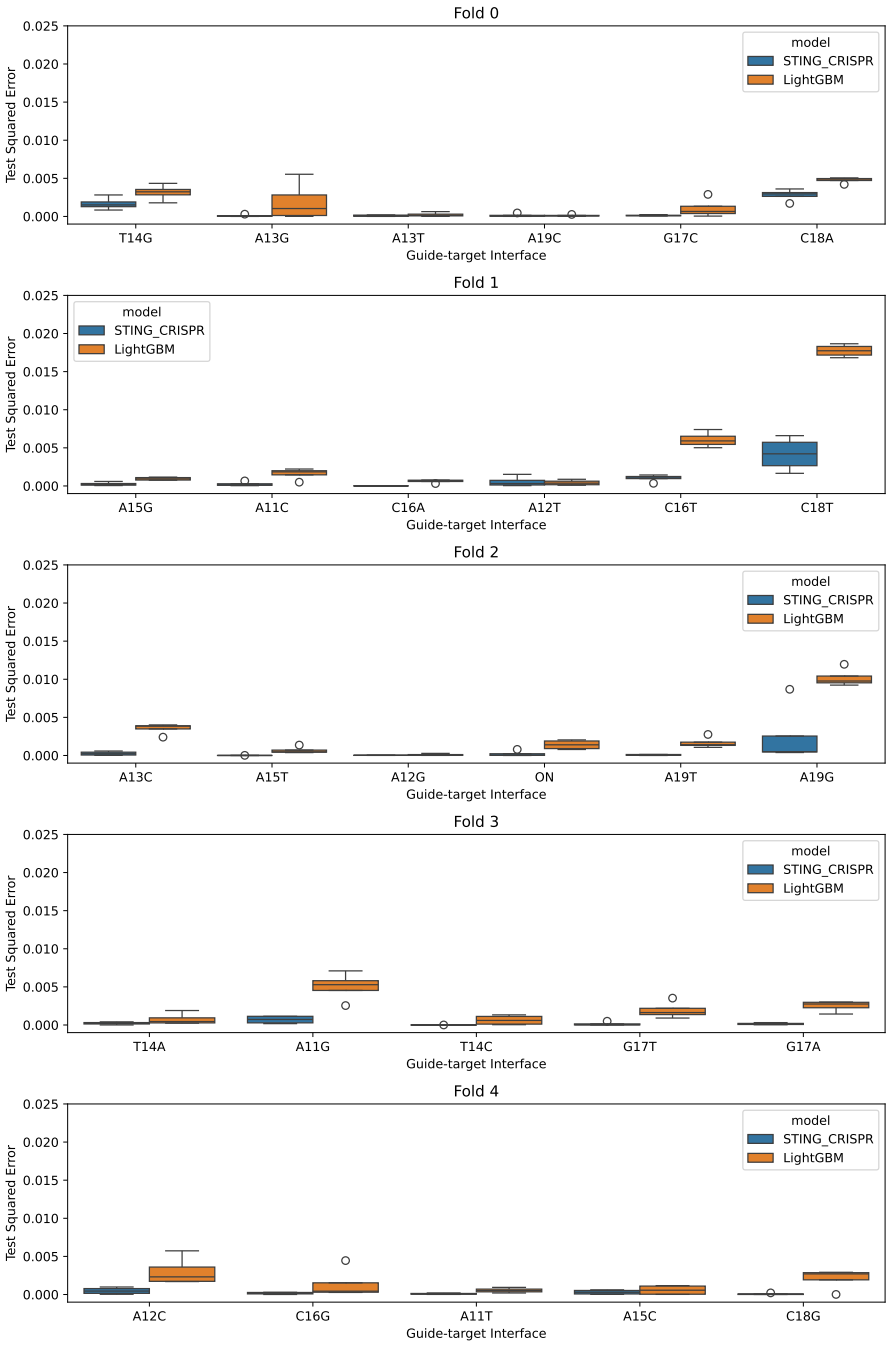


Figure 3.10: Box plots comparing test squared errors between STING_CRISPR and the new LightGBM model trained in Figure 3.9. The x-axis lists the guide-target interfaces held-out in each of the five cross validation folds. Circles represent outliers in the box plot.

We also tried withholding snapshots from entire sgRNA-target pair trajectories instead of the last four snapshots, as holding out sgRNA-target pairs would serve as a better test for evaluating the ML model's ability to generalize to unseen sgRNA-target pairs – an ability observed in many existing ML-based off-target activity prediction tools. However, the test performance varies across the five folds in 5-fold cross validation when a variety of ML models without feature selection (linear regression, ridge regression, XGBoost, extra trees and LightGBM) are used (see Figure 3.9). As seen in the figure, all ML model types fail to generalize on fold 1. Examining the distribution of test squared errors per sgRNA-target pair in the LightGBM model, we observe variance in predicted activities within a sgRNA-target pair MD trajectory, indicating variability between snapshots within the trajectory (see Figure 3.10). Owing to poor test performances, we do not proceed with SHAP interpretation of these ML models. Details on methods can be found in Appendix B.2.6 "Holding out trajectories as test sets".

3.4 Discussion

3.4.1 Structural plasticity of the heteroduplex: structural stability of mismatches

According to our MD simulations, introducing a mismatching mutation in the PAM-distal region of the TS does not necessarily produce a major structural instability in the overall structure of the heteroduplex nor in that of the Cas9 protein. By using the analysis described in Appendix subsection B.2.5, we actually found that these mutations produce minor perturbations in the dynamics of the sgRNA in the PAM-distal region, but also, unexpectedly, a stabilizing effect on some RNA bases in the PAM-proximal region and on some residues of the Cas9 protein. This is consistent both with the experimental cleavage activity data and with the observations concerning the heteroduplex base pair distance sums. We suspect such stabilizing effect arises from a release of mechanical strain in the heteroduplex, where the mechanical strain originates from differing helical parameters between RNA-DNA heteroduplexes (closer to A-form than B-form) and A-form RNA or B-form DNA duplexes [256, 257].

3.4.2 Nanoenvironment approach

In this work, we identified four specific hotspots (residues 136/164/317/402/408/411/415; 730/732-734; 837-839 and 1016-1017) which are borderline with the interface between the Cas9 protein and the heteroduplex. Namely, approximately half of the hotspot residues are part of the protein-heteroduplex interface (formed by the Interface Forming Residues – IFRs) and the other half belong to the immediate next layer leaning on the IFRs. Those hotspots are actually groups of amino acid residues to which specific STING Descriptors [199, 200, 201, 202, 203, 204] are attached. The localization of amino acid residues within hotspots is indicative of their functional importance in terms of modulating off-target cleavage activity. To get the location of hotspots, however, it was first necessary to obtain a list of residue-resolved physico-chemical/structural features.

3.4.3 Cleavage activity prediction models and their interpretability

Some of the most successful models for CRISPR-Cas9 off-target activity prediction are based on deep learning and managed to reach high predictive performance in terms of classification [53, 155, 52, 181]. The building of sufficiently accurate regression models for the

problem of off-target cleavage activity prediction is still an open challenge in spite of the increasing sophistication of deep learning approaches and encoding practices applied on the sgRNA-TS (guide-target) sequence pair [53, 155, 52]. A recent advance utilized structural information of the guide-target sequence pair extracted from MD simulations in order to construct RNA-DNA molecular interaction fingerprints, i.e., structurally-informed encodings of the guide-target heteroduplex [258]. However, none of the previous works leveraged the information from the entire CRISPR-Cas9 complex, especially from the Cas9 protein. The current state of the field suggests that it has reached its possible best performance on this type of learning problem associated with mainly describing a datapoint with a guide-target sequence pair or a structurally-inspired heteroduplex encoding from it.

As an alternative to proposing another new learning model on existing datasets based on guide-target sequence pairs, our work proposes a new learning approach/problem that takes into account the whole sgRNA-dsDNA-Cas9 complex in its entire physico-chemical/structural internal 'reality'. This is achieved by obtaining a set of physico-chemical/structural features characterising all guide-target proximal residues in a given sgRNA-dsDNA-Cas9 complex that accommodates a given guide-target pair. Unlike Chen et al. [258], our physico-chemically/structurally-informed features are obtained from MD simulation of the entire CRISPR-Cas9 complex, which includes the Cas9 protein in addition to the guide-target heteroduplex and other parts of the R-loop.

We work under the assumption that the 3D internal protein nanoenvironments, and features therein, of guide-target pairs are able to provide an information-rich representation of the guide-target pairs themselves. We therefore trained a ML pipeline with a built-in feature selection step, i.e., scikit-learn's SelectFromModel, in order to simultaneously identify the most important features informative for cleavage activity prediction and train a ML model which predicts cleavage activities. We then evaluate the ML model's ability to predict cleavage activities for unseen 3D protein nanoenvironments (associated with guide-target pairs) in the test set. Our results indicate that the trained model successfully captures the relationship between 3D protein nanoenvironments and cleavage activities for the studied sgRNA-TS pairs. In particular, the trained model is capable of predicting experimental cleavage activities with an accuracy of 0.819 Spearman and 0.916 Pearson correlation coefficients. While this delivers a high level of accuracy, the current model presented in our study was only trained on a small subset of experimentally available sgRNA-TS pairs. Another limitation of our approach is that the activity prediction of any unseen sgRNA-TS pair would require performing a new molecular dynamics trajectory. Therefore, the current model is not expected to replace existing high-throughput methods aiming at predicting off-target cleavage activity at the genomic scale for any sgRNA-TS pair.

However, the advantage of our method consists in leveraging often neglected factors such as features related to Cas9 residues influencing off-target activity. These features are descriptors characterising a particular residue. We found that the parent descriptor classes in order of decreasing SHAP importance are: density, side chain orientation, accessibility, weighted contact number, entropy density, electrostatic potential, sponge, cross presence order, contact energy density, graph descriptor, and solvation. Our analysis also identifies the most significant residue hotspots 136/164/317/402/408/411/415, 730/732-734, 837-839 and 1016-1017 responsible for modulating cleavage activity for the studied sgRNA-TS pairs. Our study highlights the importance of more general characteristics than mere residue identity. The most important residues identified in this work are in fact carriers of important characteristics rather than pure amino acid properties. Furthermore, we found that general determinants of internal protein packing is of fundamental importance

and this is obvious from the presence of descriptors such as density, sponge and weighted contact number. In addition, general geometry (accessibility), physico-chemical features (electrostatic potential) and finally the evolutionary preservation of sequences (entropy density) are pertinent and crucial for the determination of cleavage activity for the studied sgRNA-TS. Further studies are needed in order to establish whether our findings still apply for any sgRNA-TS pair such as ones containing multiple PAM-distal or PAM-proximal mismatches and for any sgRNA. While these investigations are not in the scope of our current proof-of-concept study, the agreement with experimental findings are encouraging.

The identity of residues in some of the residue hotspots is in concordance with recent experimental findings. For example, residue 837 has been hypothesised to aid in the positioning of the target DNA relative to the HNH domain [259] and to function as a catalytic residue [260, 261], although the latter hypothesis has been questioned by more recent findings [259]. Along with 837, residues 838 and 839 are of known importance as parts of the catalytically active site of the HNH domain, coordinating the metal ions [262, 259]. Indeed, the mutation D839A was shown to compromise gene editing activity in site-directed mutagenesis experiments [259]. Proximal to 402 and 408, residue 406 is part of the negative pocket of the REC-I domain which is instrumental in RNA recruitment [41]. Residues 1016 and 1017, together with residues 1010 and 1025 detected by STING_CRISPR, are part of a RuvC loop which was shown to only stabilize PAM-distal mismatches in the heteroduplex rather than activate on-target interfaces [39]. In addition to these residues, our analysis characterises Cas9 residues 268, 908, 919 and 1122 as important residues. Interestingly, residues 908 and 919 are part of the L2 loop, which interacts with the NTS in order to dock HNH to the TS, i.e., activate the HNH domain [40], and reposition the NTS in the RuvC cleavage site [39]. Residue 908 also interacts with the unwound DNA in cases of multiple PAM-distal mismatches, thereby hampering HNH cleavage activation [263], though 908 is not shown to interact with the PAM-distal region in the 672 PDB snapshots. Residue 268 detected by STING_CRISPR is next to residues 267 and 269, both of which were shown to form contact with target strand that kink the NTS [264]

The approach we took in this paper would be also capable of predicting the effect of certain residue mutations on cleavage activity for sgRNA-TS pairs including, but not limited to, the ones covered by this work. Such an approach would be similar to Similar to Venanzi et al. [265]'s approach in using MD simulation-derived features for enyzme variant activity prediction. In fact, the present model is already fully functional in this regard since it has learned the relationship between the protein 3D nanoenvironments of guide-target pairs and cleavage activities and is, therefore, capable of making a prediction of cleavage activity based on the protein 3D nanoenvironment of a guide-target pair irrespective of 'how the protein 3D nanoenvironment is realized'. Therefore our trained model already has the ability (by construction) to predict the effect of any Cas9 residue mutation on (off-)target cleavage activity provided that the protein 3D nanoenvironment of corresponding guide-target pair is computed consistently via molecular dynamics. This later task can be automated following the same steps outlined in Fig. 3.2 but using the initial systems in which Cas9 has the desired residue mutations. While the aim of the paper was not to predict the effect of residue mutations on (off-) target cleavage activity, our proposed approach also offers a possible computational solution to tackle this important and very timely problem. This type of computational approach would pave the way for *in silico* design of optimal 3D protein nanoenvironments of desired guide-target pairs (representing optimal combination of mutations of Cas9) that would maximise on-target activity and minimise off-target effects.

The current limitations of our approach include the necessity of performing a molecular

dynamics trajectory in order to generate the protein 3D nanoenvironment for a given sgRNA-TS pair. Therefore, our approach is not expected to compete with the currently available state-of-the-art methods [56, 52, 60, 167, 198, 258] for predicting off-target activity for any sgRNA-TS pair. To mitigate this limitation, future work could consider the use of alternative cheaper ways, e.g., AlphaFold 3 [266], for generating CRISPR-Cas9 complex conformational ensembles and hence protein 3D nanoenvironments for sgRNA-TS pairs.

3.4.4 Limitations

The 23 Cas9 residues found in this study are important only for the 28 "studied sgRNA-TS pairs", rather than for all possible SpCas9 guide-target interfaces. While the 28 sgRNA-TS pairs are all annotated with experimental (off-)target cleavage activities measured in Jone Jr et al. [156], we acknowledge that data from further experimental biochemical assays could help to (in)validate the 23 Cas9 residues identified in STING_CRISPR, thus allowing one to assess the extent to which STING_CRISPR is able to identify Cas9 residues which significantly modulate cleavage activity (e.g., via Precision/Recall scores). For example, one could perform alanine scanning at the 23 Cas9 residues for all 28 studied sgRNA-TS pairs and measure experimental cleavage activities for the 23*28 combinations. However, such an experiment is beyond the scope of this study.

Nonetheless, in the previous subsection, we have been able to relate 8 of the 23 Cas9 residues to the existing literature, which highlight the importance of these 8 residues. Furthermore, the assessment of Cas9 residue importance in cleavage activity via ML model interpretation is unprecedented. Based on the above two statements, we believe that this provides sufficient evidence for STING_CRISPR to lay the foundations for a new type of interpretable ML models which account for the ways in which Cas9 residues affect cleavage activity.

We also tried withholding snapshots from entire sgRNA-target pair trajectories instead of the last four snapshots, as holding out sgRNA-target pairs would serve as a better test for evaluating the ML model's ability to generalize to unseen sgRNA-target pairs. However, the test performance varies across the five folds in 5-fold cross validation when a variety of ML models without feature selection (linear regression, ridge regression, XGBoost, extra trees and LightGBM) are used (see Figure 3.9). Examining the distribution of test squared errors per sgRNA-target pair in the LightGBM model, we observe variance in predicted activities within a sgRNA-target pair MD trajectory, indicating variability between snapshots within the trajectory (see Figure 3.10).

Regarding model performance in Figure 3.9, we acknowledge that all ML models fail to generalize in fold 1. This is likely because the data used for ML model training does not contain sgRNA-target mismatch interfaces which cover all base pair positions and mismatch types. This issue could easily be resolved by including trajectories of guide-target interfaces with multiple mismatches in the ML dataset. In particular, one would ensure that all heteroduplex base pair positions are covered in the training set while making sure that there are no overlapping guide-target interfaces between the training and test sets (to avoid data leakage). Nonetheless, such a proposal is beyond the scope of this study due to computational resources.

3.5 Conclusions

Research efforts and applications using CRISPR-Cas9-based genome engineering have been increasing since the discovery of the CRISPR-Cas9-based "genetic scissors", which has transformed industrial biotechnology and modern agriculture. CRISPR-Cas9-based genome engineering shows great promise for curing diseases with an unparalleled efficiency that would have been inconceivable at the beginning of the century. However, its ability to transform medicine strongly relies on the understanding of possible side effects caused by the off-target activity of the CRISPR-Cas9 gene editing system. This research challenge catalyzed tremendous efforts in both experimental and computational sciences. As a result, the most successful computational models, which are based on deep neural networks or biological fingerprinting, managed to deliver accurate results in the activity classification of guide-target sequence pairs but interpreting these models does not deliver information on the importance of Cas9 residues in modulating cleavage activity. Therefore, building accurate and explainable models that facilitate the design of CRISPR-Cas9-based gene editing experiments is among the greatest challenges of present-day computational biology.

The present work is one step forward towards meeting this challenge and introduces a reformulation of the learning task for CRISPR-Cas9 off-target cleavage activity prediction with the ultimate goal of building explainable machine learning models capable of predicting CRISPR-Cas9 off-target cleavage activity with high accuracy. The contributions of this work are the following:

1. Successfully deriving a novel and powerful "physico-chemical and structural" information-enriched representation for guide-target sequence pairs consisting of 30 features (capturing the protein 3D nanoenvironment of the guide-target pair);

2. Training a machine learning model to learn the relationship between the said representation and the off-target cleavage activity; and

3. Shedding light on the structural and physico-chemical determinants of CRISPR-Cas9 off-target cleavage activity and identifying the most important residues, whose structural and physico-chemical descriptors modulate (off-)target activity for the studied sgRNA-TS pairs, by interpreting the successful machine learning predictions.

For the first time, our machine learning model STING_CRISPR is also capable of predicting the effect of CRISPR-Cas9 residue mutations on off-target cleavage activity, paving the way for further exploration and discoveries.

Chapter 4

DeepEmbCas9: Cas9 coevolution and sgRNA structural information for CRISPR-Cas9 cleavage activity prediction

This chapter has been preprinted on *bioRxiv* with the citation *Mak, J. K., & Minary, P. (2025). DeepEmbCas9: Cas9 coevolution and sgRNA structural information for CRISPR-Cas9 cleavage activity prediction. Cold Spring Harbor Laboratory.* `https://doi.org/10.1101/2025.10.08.681228`.

4.1 Introduction

STING_CRISPR [267] tried to addresses this by predicting SpCas9 cleavage activity from molecular dynamics (MD)-derived residue-level physicochemical/structural descriptors for 27 SpCas9 spacer-target interfaces. However, the approach taken for developing STING_CRISPR cannot be scaled to full SpCas9 cleavage activity datasets containing hundreds of thousands of guide-target pairs used for developing DL-based Cas9 cleavage activity tools, as the approach would require computationally expensive all-atom MD simulations to be run for every guide-target pair in the dataset to generate the residue-level physicochemical/structural descriptors needed for training STING_CRISPR. Unideepsg [268] is a DL model trained on SpCas9, eSpCas9(1.1) and SpCas9-HF1, but indirectly captures differences between SpCas9 nucleases in two non-trivial dataset-trained parameters. As part of machine learning-assisted directed evolution (MLDE) for Cas9 nucleases, Thean et al. [269] trained Cas9-activity ML models for multiple SpCas9 or SaCas9-KKH guide-target interface using protein embeddings from Georgiev et al. [270] and Bepler et al. [271], despite these models not being able to generalize to unseen guide-target interfaces. Indirectly related to spacer-target cleavage activity prediction, PAMmla [157] takes a 4nt PAM sequence and a one-hot encoding of 6 PAM-interacting residues and returns the cleavage rate for the specified SpCas9 variant and PAM. Similarly, CICERO [272] leverages ESM2 embeddings for Cas9 protein PAM prediction. Riding the recent success of protein language models (pLM) [273, 274, 275, 276, 277, 278], PLM-CRISPR [179] leverages ESM2 [279] embeddings and DL for SpCas9 variant cleavage activity prediction, but is only limited to NGG PAM on-target interfaces and 7 increased-fidelity SpCas9 variants. Though not related to activity prediction, pLM and genomic language models

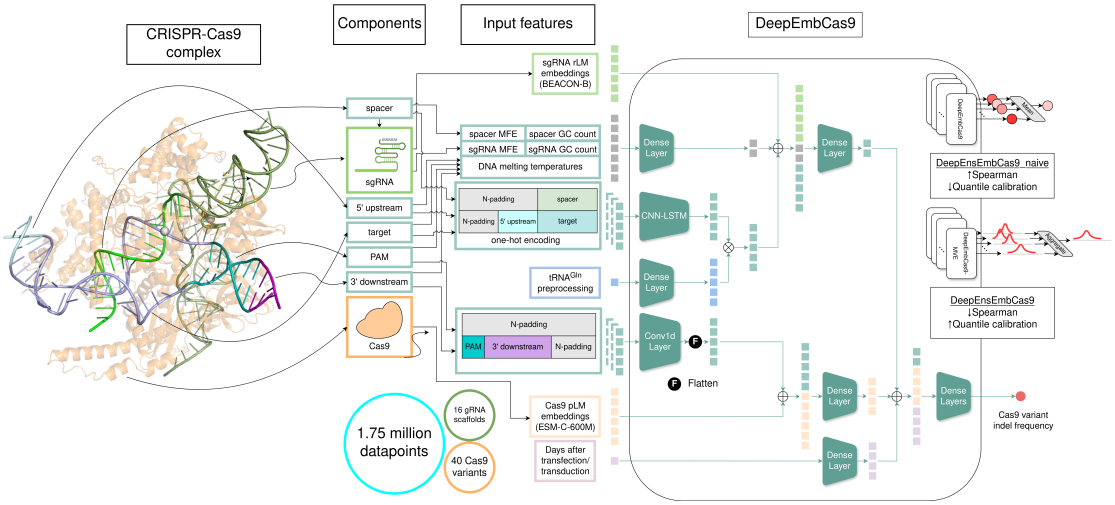


Figure 4.1: Overview of DeepEmbCas9. All components of the CRISPR-Cas9 complex are featurized and fed as input to DeepEmbCas9. DeepEmbCas9, a biophysically inspired neural network, was developed using a dataset with 1.75 million datapoints spanning 40 Cas9 variants and 16 gRNA scaffolds. Ensembling of neural networks with and without mean-variance estimation yields DeepEnsEmbCas9 and DeepEnsEmbCas9_naive.

(gLM) have also been used for CRISPR-Cas protein classification [280] and *de novo* Cas9 protein design [281, 282, 277]. The field has also seen tools, e.g., CCLMoff [283], which use RNA language model (rLM, i.e., RNA foundation models) embeddings [284, 285] for SpCas9 cleavage activity prediction.

In light of the above, we introduce a family of 4 DeepEmbCas9 models — the first set of DL models to featurize all three components (sgRNA, DNA, Cas9) in the CRISPR-Cas9 complex. DeepEmbCas9 models introduce a new guide-target encoding scheme that enables the unification of indel frequency data from various Cas9 orthologs and their variants. The latter enables us to curate the largest guide-target lentiviral library-based CRISPR-Cas9 indel frequency dataset, which has over 1.75 million datapoints spanning 40 Cas9 variants and 16 gRNA scaffolds, and to use this dataset for the training and benchmarking of DeepEmbCas9 models. It also makes DeepEmbCas9 models applicable to more Cas9 variants compared to PLM_CRISPR. In terms of neural network architecture, DeepEmbCas9 models use inductive biases informed by Cas9's biophysical mechanism, where early fusions of sgRNA/target DNA-related and Cas9/PAM-related features in the architecture are inspired by sgRNA-target DNA and Cas9-PAM interactions in the CRISPR-Cas9 complex, respectively. The use of concatenated region-wise pLM mean embeddings for encoding Cas9 variants (further referred to as Cas9 pLM embeddings onwards) enable DeepEmbCas9 models to parallel STING_CRISPR's ability to assess the impact of Cas9 domains on activity. Analogously, concatenated region-wise rLM mean embeddings for encoding sgRNAs (further referred to as sgRNA rLM embeddings) enables one to assess the impact of various parts of the sgRNA (scaffold) on activity. Compared to PLM-CRISPR, DeepEnsEmbCas9_naive – the flagship model among the four models — has comparable test performance when benchmarked against existing single-variant deep learning-based indel frequency prediction tools in both the in-distribution and leave-one-nuclease-out extrapolation settings. Equipped with deep ensembles of mean-variance estimators, DeepEnsEmbCas9 captures both aleatoric and epistemic uncertainty, yielding calibrated uncertainties and good test performance. SHAP importance analysis on

DeepEmbCas9 emphasizes on the importance of CRISPR-Cas9 PAM binding on cleavage activity prediction. Comparing between 39 nuclease pairs, DeepEmbCas9 is seen to regard Linker and PLL-WED-PI domains as being important for nuclease pairs containing increased-fidelity and PAM-altering mutations, respectively. Taken together, DeepEmbCas9 models form the first step towards a general CRISPR-Cas9 activity model capable of making accurate prediction for any Cas9 variant available in the literature.

4.2 Methods

Since different studies use different nuclear localization signals (NLS), we distinguish between nucleases and variants, where nucleases refer to the Cas9 domains, and variants refer to the Cas9 domains together with the NLS, FLAG and P2A components. For convenience, the term "variants" also include wild-type nucleases.

4.2.1 Dataset construction

Study	Published tools	Cas9 variants	gRNA scaffolds	No. of unique datapoints (% mismatched)
Kim, Kim et al. (2019) [5]	DeepSpCas9	SpCas9	SpCas9 scaffold 1	13,359 (72.89%)
Wang et al. (2019) [4]	DeepHF	SpCas9, eSpCas9(1.1), SpCas9-HF1	SpCas9 scaffold 1 (5T)	171,109 (0%)
Kim et al. (2020) [6]	DeepxCas9, DeepSpCas9-NG	SpCas9, xCas9, SpCas9-NG	SpCas9 scaffold 1	88,786 (19.04%)
Kim, Kim et al. (2020) [7]	DeepSpCas9variants	HypaCas9, Sniper-Cas9, SpCas9-HF1, SpCas9, eSpCas9(1.1), evoCas9, xCas9, SpCas9-NG, VRQR, QQR1, VQR, VRER, VRQR-HF1	SpCas9 scaffold 2	318,136 (51.76%)
Seo et al. (2023) [8]	DeepSmallCas9	SpCas9, SaCas9, SaCas9*, SaCas9-KKH, SaCas9-HF, SaCas9-KKH-HF, eSaCas9, efSaCas9, SauriCas9, SauriCas9-KKH, SlugCas9, SlugCas9-HF, Sa-SlugCas9, sRGN3.1 CjCas9, enCjCas9, Nm1Cas9, Nm2Cas9, St1Cas9,	SpCas9 scaffold 2, SaCas9 scaffolds 1-3, CjCas9 scaffolds 1-2, NmCas9 scaffolds 1-3, St1Cas9 scaffolds 1-5	713,163 (73.56%)
Kim, Kim, Okafor et al. (2023) [10]	DeepSniper	Sniper-Cas9, Sniper2P, Sniper2L	SpCas9 scaffold 1	127,034 (51.80%)
Kim, Choi et al. (2024) [9]	DeepCas9variants	SpRY, Sc++, SpCas9-NRCH, SpCas9-NRRH, SpCas9-NRTH, SpG, SpCas9, SpCas9-NG, VRQR, xCas9	SpCas9 scaffold 2	205,358 (59.04%)
Kim, Kim et al. (2020) [7] and Kim, Choi et al. (2024) [9]	DeepSpCas9variants, DeepCas9variants	SpCas9-NG, SpCas9, VRQR, xCas9	SpCas9 scaffold 2	109,741 (56.06%)
Total	8 tools	40 Cas9 variants	16 gRNA scaffolds	1,746,686 (55.22%)

Table 4.1: List of studies used for curating the Cas9 variant indel frequency dataset consisting of 1.75 million points spanning 40 Cas9 variants and 16 gRNA scaffolds, in addition to corresponding tools used as baselines for this study. Indel frequencies were aggregated across studies such that a datapoint is uniquely identified by its spacer, target, gRNA scaffold, Cas9 variant, number of days post-transduction/transfection, and use of $tRNA^{Gln}$ preprocessing. Numbers enclosed within parentheses in the last column indicate the percentage of unique datapoints which contain spacer-target mismatches, either as a mismatched 5' guanine or mismatches elsewhere in the heteroduplex.

To build a CRISPR-Cas9 activity prediction tool spanning multiple Cas9 variants and gRNA scaffolds, we built a dataset by combining high-throughput guide-target lentiviral library-based indel frequency data from 6 studies [4, 5, 6, 7, 8, 10] (see Table 4.1). A list of column descriptions for the dataset can be found in Table C.1, noting that the primary keys of the dataset are "Spacer sequence", "Target context sequence", "Variant", "gRNA scaffold", "Day" and "tRNA feature", the tuple of which forms a unique experimental configuration. Frequency counts on the number of mismatches among guide-target interfaces in the dataset can be found in Table C.2. The names of the gRNA scaffold in the dataset are identical to those used in Seo et al. [8], apart from naming of SpCas9 gRNA scaffolds,

where "SpCas9 scaffold 1" is the standard unoptimized scaffold with a 6nt poly(U) tail, "SpCas9 scaffold 1 (5T)" is the standard unoptimized scaffold with a 5nt poly(U) tail, and "SpCas9 scaffold 2" is the optimized scaffold described by Dang et al. [95]. To avoid over-representation of experimental configurations with multiple indel frequency measurements within a study and/or across studies, e.g., due to biological replicates, we average indel frequencies within each study and across the whole dataset to build per-study and cross-study ML-ready labels with values ranging up to 100%. Experimental indel frequencies from Wang et al. [4] were also scaled up 100× to match the indel frequency ranges used in the other studies.

Cas9 and guide RNA (gRNA) scaffold sequences were also curated as part of the dataset. The 16 gRNA scaffold sequences were derived from sequences in Supplementary Table 9 of Seo et al. [8], noting that Wang et al. [4] uses "SpCas9 scaffold 1 (5T)". Protein and codon sequences of the 40 Cas9 variants (including the nuclear localization sequence, FLAG tag and P2A peptide) were obtained from Addgene and supplementary files of the original studies (additional details available in Appendix 4.2.1 "Dataset construction").

4.2.2 Input feature encodings

The dataset was preprocessed in order to construct deep learning-ready data representations of components in the guide-target-Cas9 variant R-loop complex (Figure C.4A). Illustrated in Figures 4.2A and C.4B, a one-hot representation of the spacer and target context sequence was built by stacking of one-hot encodings of the padded spacer and target context sequences, where the padded sequences were constructed by flanking the raw spacer and target context sequences with padding (gray), represented by the letter 'N' in both sequences, such that:

- both padded sequences have length 42;
- heteroduplex nucleotides in both padded sequences are positionally aligned; and
- the PAM sequence starts at the 28th nucleotide in the padded target context sequence.

In the one-hot encoding, 'N' maps to the zero vector, i.e., 'N' becomes zero-padding. Values in the "tRNA feature" and "Day" dataset columns are stored as a binary value and a real number, respectively, where a visual depiction of $tRNA^{Gln}$ preprocessing is shown in Figure 4.2B.

Similar to Seo et al. [8], melting DNA temperatures, RNA minimum free energy (MFE) and GC counts were calculated for all guide-target interface in the dataset. `Tm_NN` with default parameters from Biopython's Bio.SeqUtils package [247] (`https://biopython.org/docs/dev/api/Bio.SeqUtils.html`) was used for computing DNA melting temperatures for 39 subsequences of the target context sequence, namely those with zero-indexed ranges [0, 5), [5, 14), [14, 20), [20, 27), [0, 7), [7, 18), [18, 22), [22, 27), [0, 3), [3, 10), [10, 17), [17, 27), [0, 4), [4, 12), [12, 19), [19, 27), [4, 11), [11, 17), [0, 6), [6, 15) and [15, 20) for the 5' upstream and protospacer parts of the target context sequence, and [27, 30), [30, 33), [27, 34), [34, 36), [27, 36), [36, 38), [27, 34), [34, 37), [27, 33), [33, 36), [27, 35), [35, 38), [27, 33), [33, 38), [27, 31), [31, 34), [27, 30) and [30, 38) for the PAM and 3' downstream parts of the target context sequence. DNA melting temperatures for 'N'-containing subsequences were handled by replacing 'N' with 'G' (i.e., guanine) prior to MFE calculations. ViennaRNA's Python API [286] was used for calculating the minimum free energy (MFE) of the spacer and sgRNA (spacer + scaffold) sequences, with T replaced with U before

MFE calculations. Bio.SeqUtils's `gc_fraction` was used for calculating GC counts of the spacer and protospacer sequences.

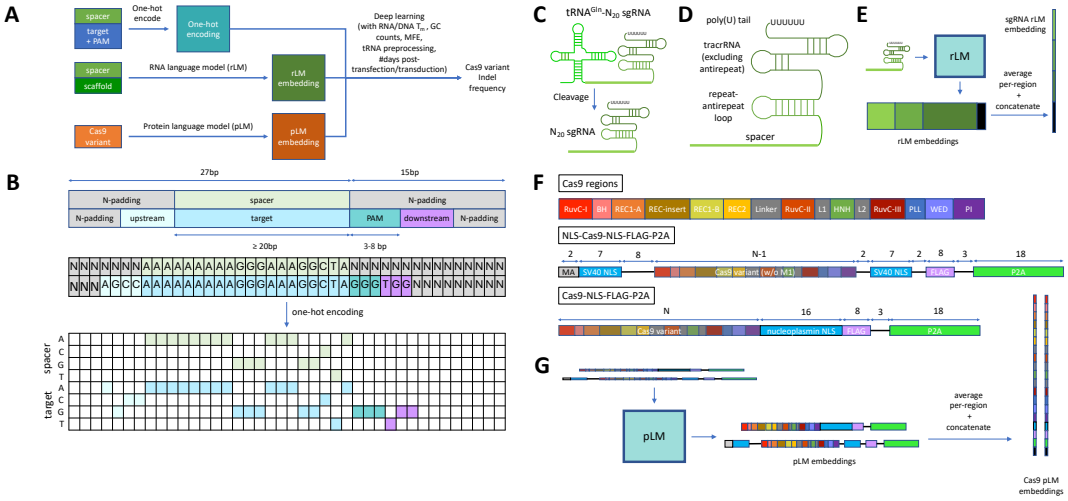


Figure 4.2: Conceptual overview of DeepEmbCas9 (A), and featurization of the full CRISPR-Cas9 complex (B-G). (B) Unified guide-target interface (top) with an example matched SpCas9 interface (middle) and its one-hot encoding (bottom). (C) Formation of perfectly matching N_{20} sgRNAs, i.e., sgRNAs with a 20nt spacer sequence, via tRNA^{Gln}-N_{20} preprocessing (for SpCas9 variants only). (D) The four sgRNA regions used in this study, namely the spacer (light green), repeat-antirepeat loop (green), tracrRNA excluding the antirepeat sequence (dark green) and poly(U) tail (black). (E) concatenated region-wise rLM mean embeddings for encoding sgRNA (sgRNA rLM embedding) are formed by passing sgRNA sequences to RNA language models (rLMs), averaging per-nucleotide embeddings in the sgRNA regions, and concatenation of the averaged embeddings in the feature dimension. (F) Protein regions (color-coded) used for constructing the concatenated region-wise pLM mean embeddings for encoding Cas9 variants (Cas9 pLM embedding). These include regions within a Cas9 variant (top, of length N), as well as nuclear localization signals (NLS, sky blue), FLAG-tag (FLAG, light purple), self-cleaving P2A peptide (neon green) and other inter-region residues (black text and lines), as seen in the NLS-Cas9-NLS-FLAG-P2A (middle) and Cas9-NLS-FLAG-P2A (bottom) sequences. (G) Cas9 pLM embeddings are formed by passing Cas9 sequences to protein language models (pLMs), averaging per-residue embeddings in the Cas9 regions, and concatenation of the averaged embeddings in the feature dimension. Inter-region residues are pooled into the "Other" region in black.

4.2.3 sgRNA regions

We define 4 non-overlapping sgRNA regions to structurally align the 16 gRNA scaffolds considered in this study (Figure 4.2D). The four regions are

- spacer sequence;
- repeat-antirepeat loop, consisting of the CRISPR RNA (crRNA) repeat sequence, GAAA tetraloop and trans-activating crRNA (tracrRNA) antirepeat sequence;
- tracrRNA with the antirepeat excluded; and

- poly(U) tail, typically the last 5-7 ribonucleotides in the sgRNA,

Boundaries delineating the regions for the gRNA scaffolds were determined via manual inspection (see Table C.3 for sgRNA region lengths for each gRNA scaffold).

4.2.4 sgRNA rLM embeddings

From sgRNA sequences in the dataset, per-nucleotide rLM embeddings were generated from RNA-FM [284], RiNALMo [287], BEACON-B [285] and BEACON-B512 [285] using a 32GB V100 NVIDIA GPU. Since genomic language models potentially encode RNA-related information, per-nucleotide genomic language model embeddings were generated from evo-1-8k [282] using a 40GB A100 NVIDIA GPU. rLM/gLM embeddings were then converted into sgRNA rLM embeddings by averaging (ribo)nucleotide embeddings within each sgRNA region, followed by concatenation of the sgRNA region embeddings in the feature dimension (Figure 4.2D).

4.2.5 Cas9 regions

We use Cas9 regions, i.e., contiguous protein segments in the Cas9 nuclease, to structurally align the 40 Cas9 variants (Figures 4.2E top and C.3B). The term "region" is used to distinguish from "domain", which is defined as one or more noncontiguous protein segments. To clarify, "REC-insert" (Figure C.3, beige-colored) denotes the domain inserted into the REC1 domain, i.e., Wing in St1Cas9, and REC2 domains in SpCas9 and ScCas9 variants. "REC1-A" denotes the first REC domain in the Cas9 nuclease, i.e., REC1-A in SpCas9, St1Cas9 and ScCas9 variants, REC1 in Nm1Cas9 and Nm2Cas9, and REC in the other small Cas9 variants. Boundaries delineating Cas9 regions for the 40 Cas9 variants were gathered from literature and predicted from multiple sequence alignment (MSA) of wild-type Cas9 nucleases. Specifically,

- SpCas9 regions were obtained from Huai et al. [261], with the additional REC3-linker loop boundary at residue 712 from Jiang et al. [38];
- SaCas9* and SaCas9 regions were obtained from Nishimasu et al. [115], with SaCas9 having inserted glycine at position 2 (the same applies for other SaCas9 variants);
- St1Cas9 regions were obtained from Zhang et al. [111];
- SlugCas9, SauriCas9, Sa-SlugCas9 regions were obtained from Hu et al. [91];
- CjCas9 regions were obtained from Yamada et al. [121]; and
- Nm1Cas9 and Nm2Cas9 regions were obtained from Sun et al. [288].

Region boundaries with unknown positions were predicted through MSA of Cas9 nucleases, followed by projection of the region boundary from a similar nuclease to the target nuclease via the MSA. Specifically, we built an MSA consisting of SpCas9, SaCas9*, SaCas9, St1Cas9, sRGN3.1, SlugCas9, SauriCas9, Sa-SlugCas9, CjCas9, Nm1Cas9, Nm2Cas9, ScCas9 and 8313 other type II CRISPR RNA-guided endonuclease Cas9 of length > 800 from UniRef100 [289] using Clustal Omega [290, 291] program. Using the MSA, we projected SpCas9's WED start position to St1Cas9, Nm1Cas9 and Nm2Cas9, which resulted in predicted WED start site positions at 831, 851, and 850, respectively. ScCas9's region boundaries were generated by using the same MSA and projecting all SpCas9 region boundaries to ScCas9. sRGN3.1's region boundaries were generated by aligning ShyCas9,

SmiCas9, SpaCas9, SlugCas9 and sRGN3.1 via Clustal Omega, followed by projection all SlugCas9 region boundaries to sRGN3.1. The resulting sets of Cas9 region boundaries are visualized in Figure C.3, noting that Cas9 variants sharing the same base nuclease (as indicated in Table C.4) share the same region boundaries.

In addition to the contiguous Cas9 regions, we devised four non-contiguous regions — NLS, FLAG, P2A and Other — to account for Cas9 nuclease-flanking residues which are part of the nuclear localization signal (sky blue), FLAG tag (light purple), P2A peptide (neon green) and other residues not part of the aforementioned regions (black horizontal lines and "MA" text at the N-terminal of NLS-Cas9-NLS-FLAG-P2A), respectively (Figure 4.2E middle and bottom). More concretely, NLS covers the nucleoplasmin NLS and two SV40 NLS sequences in Cas9-NLS-FLAG-P2A and NLS-Cas9-NLS-FLAG-P2A, repectively.

4.2.6 Cas9 pLM embeddings

From the 40 Cas9 sequences in the dataset, per-residue pLM embeddings for each sequence were generated from ProtT5 (specifically Rostlab/prot_t5_xl_half_uniref50-enc) [275], Ankh-large [276], ESM3-1.4B [274, 273], ESM-C-300M [278, 273], ESM-C-600M [278, 273], ESM-C-6B [278, 273] using a 16GB V100 NVIDIA GPU (Figure 4.2F). Since genomic language models (gLM) potentially encode protein-related information, per-nucleotide genomic language model embeddings were also generated from gLM2_650M [292] for codon sequences corresponding to the 40 Cas9 variants. pLM/gLM embeddings were then processed into Cas9 pLM embbedings by averaging residue/nucleotide embeddings within each Cas9 region, followed by concatenation of the Cas9 region embeddings in the feature dimension. We did not use ESM2 [279] to generate Cas9 pLM embeddings, as the length of SpCas9 exceeds ESM2's context window size of 1024 (i.e., maximum amino acid sequence length of 1022).

4.2.7 Neural network architecture and training

DeepEmbCas9 is a deep learning model that predicts the indel frequency for a given sgRNA, target context sequence, Cas9 variant with known structural domain annotations, and time since transfection/transduction of the sgRNA and Cas9 plasmids. DeepEmbCas9's neural network architecture and relevant hyperparameters are detailed in Figure 4.3. To build DeepEmbCas9, we reused design ideas from existing deep learning-based CRISPR-Cas9 activity prediction models. Notably,

- design of the guide-target branch (consisting of convolutional and bidirectional long short-term memory (biLSTM) layers) and Cas9-PAM branch was inspired by the sequence and epigenetic arms from crispAI [293], respectively;
- element-wise multiplication operation between the "$tRNA^{Gln}$ preprocessing" and spacer-target embeddings for feature fusion was adopted from DeepSpCas9variants [7]; and
- concatenation of the guide embedding, target embedding, DNA melting temperature features, GC counts, MFEs and mismatch encodings before the fully connected layers was adopted from DeepSmallCas9 [8].

As PAM recognition and binding by Cas9 are required for R-loop formation and double strand DNA cleavage [38], we used separate feature extractors for nucleotides upstream of the PAM (including heteroduplex nucleotides) and nucleotides within or downstream of the

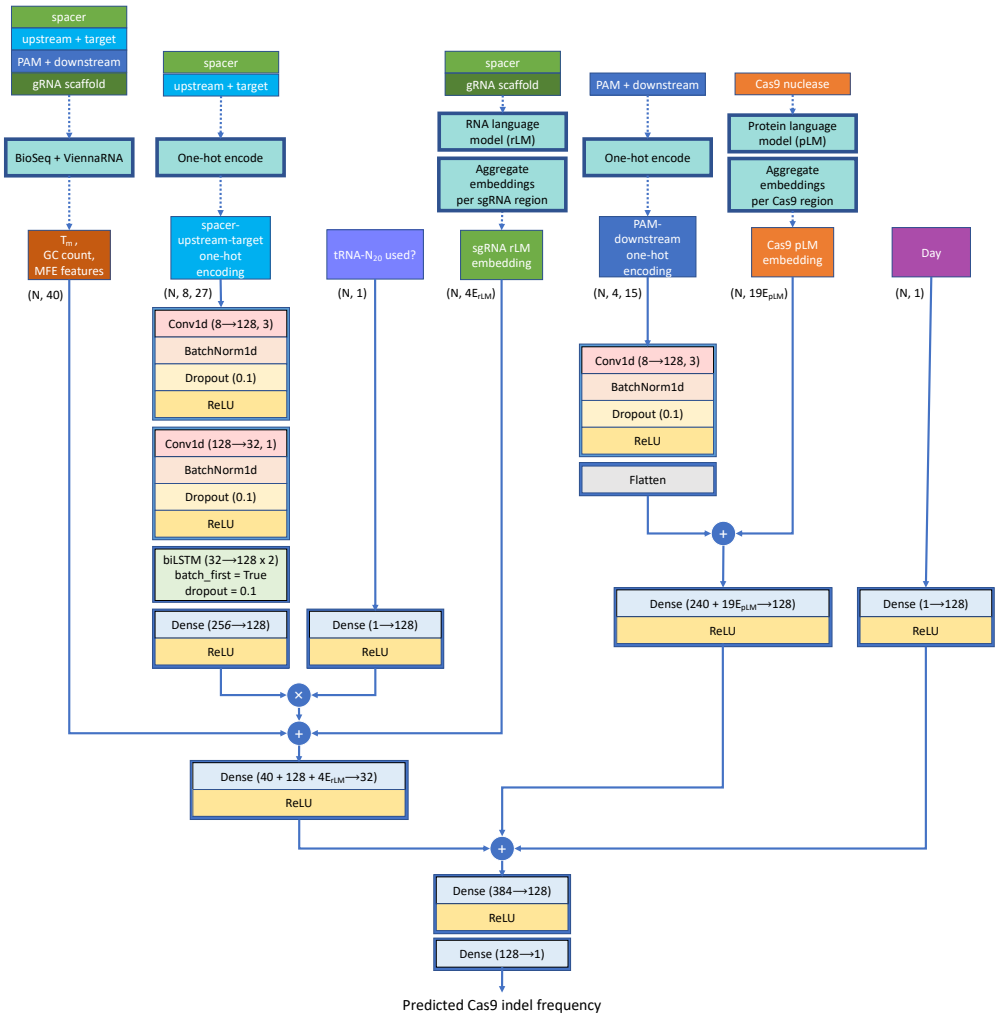


Figure 4.3: DeepEmbCas9's neural network architecture. DeepEmbCas9 mainly consists of a spacer-target branch (left side) and a Cas9-PAM branch (right side). The sgRNA's rLM embedding (green), tRNA^{Gln} preprocessing feature (cornflower blue), and BioSeq/ViennaRNA-derived RNA/DNA melting temperature (T_m)/GC count/minimum free energy (MFE) features (brown) are integrated into the spacer-target branch, whereas the Cas9 variant's pLM embedding (orange) is integrated into the PAM branch. The Day feature (purple) is integrated later in the neural network via the rightmost branch. Different branches in the neural network are fused together via concatenations and dense layers. Dotted arrows indicate the preprocessing steps used for generating DeepEmbCas9's input feature matrices/tensors. Conv1d($x \to y, z$) denotes a 1D convolutional layer with x input channels, y output channels, z kernel size and same padding, whereas Dense($x \to y$) denotes a fully connected layer with x input and y output nodes. Blue circle with + and $\times$ denotes tensor concatenation and element-wise multiplication, respectively. N denotes the number of datapoints used in a forward pass through DeepEmbCas9.

PAM into two separate neural network arms, respectively, with the Cas9 pLM embedding as input in the Cas9-PAM branch. We also chose to integrate the "Day" feature in the later layers of the fully connected layers, as the feature is unrelated to the CRISPR-Cas9 complex.

DeepEmbCas9 was implemented in PyTorch [143] and PyTorch Lightning [142]. To

train DeepEmbCas9, we used mean squared error (MSE) as the loss objective, Adam [138] with learning rate 1×10^{-3} as the optimizer and `StepLR` with `gamma=0.1` and `step_size=1` as the learning rate scheduler. To avoid overfitting, `EarlyStopping` from PyTorch Lightning with default parameters was used, which halted training after no improved validation loss for three consecutive epochs. The model weights for the training epoch with the lowest validation loss is then saved. Default neural network weight initializations were used in DeepEmbCas9. All DeepEmbCas9 models were trained on a single 16GB V100 NVIDIA GPU for no more than 4 hours.

4.2.8 Benchmark comparisons

Benchmark activity prediction tools

We opt to use deep learning tools published in the six studies listed in Table 4.1 as baselines. Specifically, DeepHF [4] is a set of 4 bidirectional LSTM models:

- DeepHF_T7-SpCas9 and DeepHF_SpCas9 predict SpCas9 activity;
- DeepHF_eSpCas9(1.1) predict eSpCas9(1.1) activity; and
- DeepHF_SpCas9-HF1 predict SpCas9-HF1 activity

for matched A/GN_{19} interfaces, i.e., spacer-target interfaces with a matched 5' adenine (AN_{19}) or a matched 5' guanine (GN_{19}). DeepSpCas9 [5] is a convolutional neural network (CNN) model which predicts SpCas9 activity for matched G/gN_{19}, i.e., spacer-target interfaces with a matched 5' guanine (GN_{19}) or mismatched 5' guanine (gN_{19}). Similar to DeepSpCas9, DeepxCas9 and DeepSpCas9-NG [6] are CNN models which predict xCas9 and SpCas9-NG activity, respectively, for matched G/gN_{19} interfaces.

DeepSpCas9variant [7] is a set of 9 CNN models accepting matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces, i.e., 20nt spacer-target interfaces with full base pair complementarity created using $tRNA^{Gln}$-N_{20} sgRNAs (see Figure 4.2C). In DeepSpCas9variant, each model predicts activity for one of 9 SpCas9 variants: SpCas9, SpCas9-VRQR, SpCas9-NG, xCas9, Sniper-Cas9, eSpCas9(1.1), SpCas9-HF1, HypaCas9 and evoCas9. DeepSmallCas9 [8] is a set of 17 CNN models accepting matched and mismatched (abbreviated (mis)matched) spacer-target interfaces, where matched/mismatched refers to the 19 nucleotides apart from the 5' guanine. In DeepSmallCas9, each model predicts activity for one of 17 small Cas9 nucleases/variants: sRGN3.1, SlugCas9, Sa-SlugCas9, SlugCas9-HF, SauriCas9, SauriCas9-KKH, SaCas9, eSaCas9, efSaCas9, SaCas9-HF, SaCas9-KKH, SaCas9-KKH-HF, St1Cas9, CjCas9, enCjCas9, Nm1Cas9 and Nm2Cas9. DeepSpCas9-v2 [8] is similar to DeepSmallCas9, but predicts (mis)matched G/gN_{19} SpCas9 activity.

DeepCas9variants [9] is a collection of 9 CNN models accepting matched G/gN_{19} interfaces, where each model predicts activity for one of 9 SpCas9/ScCas9 variants: SpCas9, SpCas9-VRQR, SpCas9-NG, SpCas9-NRRH, SpCas9-NRTH, SpCas9-NRTH, SpG, SpRY and Sc++. DeepSniper [10] is a collection of 4 tools which we name DS_Sniper1_on, DS_Sniper2L_on, DS_Sniper1_off and DS_Sniper1_off. Specifically, DS_Sniper1_on and DS_Sniper2L_on predict Sniper-Cas9 and Sniper2L activity for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces, respectively, and DS_Sniper1_off and DS_Sniper2L_off predict Sniper-Cas9 and Sniper2L activity for mismatched G/gN_{19} interfaces, respectively. For ease of notation, we use the abbreviation DS_Sniper1 for the joint use of DS_Sniper1_on and DS_Sniper1_off for predicting Sniper-Cas9 activity. Likewise, we use the abbreviation DS_Sniper2L for the joint use of DS_Sniper2L_on and DS_Sniper2L_off for predicting Sniper2L activity.

To fairly compare between DeepEmbCas9 models and DeepSniper on datasets containing both matched and mismatched interfaces, we refer to DS_Sniper1, which abbreviates for using DS_Sniper1_on when the input interface is matched, and using DS_Sniper1_off when the input interface is mismatched. Similarly, DS_Sniper2L refers to the use of DS_Sniper2L_on for matched interfaces and DS_Sniper2L_off for mismatched interfaces.

For notational convenience, we abbreviate DeepSpCas9variants, DeepCas9variants and DeepSniper as DSpCv, DCv and DS, respectively in all figures in this chapter and Appendix C.

In-distribution performance

To enable fair performance comparisons between DeepEmbCas9 and existing published cleavage activity prediction tools for CRISPR-Cas9 variants, we trained DeepEmbCas9 on train-valid-test splits compatible with such tools. To achieve this, we first obtained test partitions used for evaluating DeepHF, DeepSpCas9, DeepxCas9, DeepSpCas9-NG, DeepSpCas9variants, DeepSmallCas9, DeepCas9variants and DeepSniper from their respective source studies [4, 5, 6, 7, 8, 9, 10]. Following GitHub code provided by Wang et al. [4], we also used scikit-learn's `train_test_split` [294] with `random_state=40` and a 85%-15% train-test ratio to obtain the test partition data from Wang et al. To avoid data leakage and overrepresentation of specific experimental configurations, experimental configurations which have datapoints in both training and test partitions are relabelled to be part of the training partition. These three steps resulted in two non-overlapping partitions: a non-test partition of size 1582129 with 40 Cas9 variants and 16 gRNA scaffolds, and a test partition of size 164557 with 39 Cas9 variants and 7 gRNA scaffolds. Notably, the curated cross-study test set was a strict subset of the union of the test sets from the 6 source studies. We then randomly split the non-test partition into training and validation partitions using a 80%-20% split.

DeepEmbCas9 was trained on the resulting training partition, with the validation partition used for early stopping, where both partitions used the "Mean background subtracted indel frequency (%)" label column. Once trained, DeepEmbCas9 was then evaluated on the test partition, which used labels from the "Mean background subtracted indel frequency (source, %)", using Spearman rank correlation and root mean squared error (RMSE) as evaluation metrics. DeepEmbCas9-MVE, DeepEnsEmbCas9_naive and DeepEnsEmbCas9 (see subsection "Uncertainty Quantification via Deep Ensembles") were evaluated in a similar way.

We used DeepHF, DeepSpCas9, DeepxCas9, DeepSpCas9-NG, DeepSpCas9variants, DeepSmallCas9, DeepCas9variants and DeepSniper as baselines to compare with DeepEmbCas9 and DeepEnsEmbCas9. In short, each tool was evaluated on test sets sharing the same nuclease and guide length as the tool (e.g., DeepSpCas9 was evaluated on all test sets with matched G/gN_{19} SpCas9 interfaces) via Spearman correlation and RMSE. In addition to the GitHub model provided in `https://github.com/izhangcd/DeepHF`, DeepHF models retrained using the GitHub code provided were also used as baselines.

Leave-one-nuclease-out extrapolation

39 leave-one-nuclease-out train-test splits were formed by excluding test Cas9 nuclease training data from the training partition. We then use the same procedures described above to train the neural network, which we name DeepEmbCas9_omit to distinguish from DeepEmbCas9. Existing tools sharing the same spacer length as the test nuclease yet not trained on test nuclease data are used as baselines for DeepEmbCas9_omit. For

example, DeepSniper is one of the baselines used for the DeepEmbCas9_omit variant which excluded wild-type SpCas9 training data.

4.2.9 Model interpretation

We interpret DeepEmbCas9 using SHapley Additive exPlanations (SHAP) [146], an additive feature attribution method. Namely, SHAP values were approximated using DeepExplainer and a background dataset consisting of 100 randomly sampled datapoints. Computed SHAP values were then used for deriving SHAP importances for individual input features, where the SHAP importance of the jth input feature given by $I_j = \frac{1}{N}\sum_{i=1}^{N}|\phi_j^{(i)}|$ for dataset size N and SHAP value $\phi_j^{(i)}$ attributed to the ith datapoint-jth feature pair. Leveraging SHAP's additivity property, we also computed SHAP importances for set of input features (i.e., feature group) using the formula $I_J = \frac{1}{N}\sum_{i=1}^{N}|\sum_{j\in J}\phi_j^{(i)}|$, where J denotes the set of input features.

To systematically analyze the different parts of the CRISPR-Cas9 complex, we computed SHAP importances for CRISPR-Cas9 complex components. For fine resolution of the components we considered the following 28 feature groups, namely `spacer + spacer_MFE + spacer_GCcount`, `upstream + protospacer + protospacer_Tm`, `PAM + downstream + PAM_downstream_Tm`, `tRNA preprocessing`, `Day`, `spacer_scaffold_MFE` and 22 Cas9 feature groups — one for each Cas9 region (see Table C.6 for a list of features and feature counts for each feature group). For SHAP importance of Cas9 complex components in coarse resolution we considered the following 6 feature groups:

- spacer one-hot encoding, spacer MFE, $\frac{1}{2}\times$ sgRNA MFE, spacer GC count and spacer region part of the sgRNA rLM embedding (labelled as "spacer");
- target context sequence one-hot encoding, DNA melting temperatures features and protospacer GC count (labelled as non-target strand "NTS");
- Cas9 pLM embedding features for all Cas9 regions (labelled as "Cas9");
- rLM embedding features for all regions excluding the spacer and $\frac{1}{2}\times$ sgRNA MFE (labelled as "gRNA"); and
- tRNA^{Gln} preprocessing (labelled as "tRNA preprocessing"),

where $\frac{1}{2}\times$ sgRNA MFE denotes the fact that only half of sgRNA MFE's SHAP value is used in the SHAP importance calculation (Table C.7). As for SHAP importance analysis for each component, in addition to importances for each Cas9 region and sgRNA region, we computed the SHAP importance of each Cas9 domain, i.e.,

- RuvC, by grouping features from RuvC-I, RuvC-II and RuvC-III;
- REC1, by grouping features from REC1-A and REC1-B;
- Linker, by grouping features from the linker loop, L1 and L2;
- PLL-WED-PI, by grouping features from the PLL, WED and PI; and
- NLS-FLAG-P2A, by grouping features from NLS, FLAG, P2A and Other,

in addition to feature groups for BH, REC_insert, REC2 and HNH. Since the spacer-target interface plays a primary role in Cas9 cleavage activity prediction, we use heatmaps to visualize the SHAP importance of guide-target positions and nucleotides.

We also devised a framework for assessing the influence of input features on cleavage activity change given specific Cas9 mutation(s). Specifically, given a base Cas9 nuclease p_1 and Cas9 nuclease p_2, with p_2 typically obtained by introducing residue mutations to p_1, we can attribute the difference $\phi^{(i)}_{j,p_1 \to p_2} = \phi^{(i)}_{j,p_2} - \phi^{(i)}_{j,p_1}$ to the ith datapoint-jth feature pair due to SHAP value's additivity property. Note that summing the differences over all features yields the activity change. Based on this, we can calculate the jth feature's SHAP importance in predicting activity change, given by $I_{j,p_1 \to p_2} = \frac{1}{N} \sum_{i=1}^{N} |\phi^{(i)}_{j,p_1 \to p_2}|$. SHAP importances can be extended to groups via the formula $I_{J,p_1 \to p_2} = \frac{1}{N} \sum_{i=1}^{N} |\sum_{j \in J} \phi^{(i)}_{j,p_1 \to p_2}|$ for feature group J. We considered the following 39 pairs of Cas9 nucleases in the framework, where A > B denotes the mutation from nuclease A to nuclease B: SpCas9 > eSpCas9(1.1), SpCas9 > SpCas9-HF1, SpCas9 > evoCas9, SpCas9 > HypaCas9, SpCas9 > Sniper-Cas9, SpCas9 > xCas9, SpCas9 > VRQR-HF1, SpCas9 > QQR1, SpCas9 > SpCas9-NG, SpCas9 > VQR, SpCas9 > VRER, SpCas9 > VRQR, SpCas9 > SpCas9-NRCH, SpCas9 > SpCas9-NRRH, SpCas9 > SpCas9-NRTH, SpCas9 > SpG, SpCas9 > SpRY, Sniper-Cas9 > Sniper2P, Sniper-Cas9 > Sniper2L, VQR > VRQR, VQR > VRER, VRQR > SpG, SpG > SpRY, VRQR > SpRY, VRQR > VRQR-HF1, SpCas9-HF1 > VRQR-HF1, NLS-SaCas9 > NLS-eSaCas9, NLS-SaCas9 > NLS-efSaCas9, NLS-SaCas9 > NLS-SaCas9-HF, NLS-SlugCas9 > NLS-SlugCas9-HF, NLS-CjCas9 > NLS-enCjCas9, NLS-SaCas9 > NLS-SaCas9-KKH, NLS-SaCas9 > NLS-Sa-SlugCas9, NLS-SauriCas9 > NLS-SauriCas9-KKH, NLS-SaCas9 > NLS-SaCas9-KKH-HF, NLS-SaCas9-HF > NLS-SaCas9-KKH-HF, NLS-SaCas9-KKH > NLS-SaCas9-KKH-HF, NLS-SlugCas9 > NLS-Sa-SlugCas9, and NLS-SlugCas9 > NLS-sRGN3.1.

4.2.10 Uncertainty quantification

We augmented DeepEmbCas9 with uncertainty estimates by using mean-variance estimation [148] and/or deep ensembles [147]. Specifically, we built:

- DeepEmbCas9-MVE by training a single heteroscedastic DeepEmbCas9 model with separate mean and variance-predicting stems (Figure C.5) and initialized with different seeds using the Gaussian NLL loss objective. We apply softplus to the variance-predicting stem's output to ensure non-negativity of the predicted variance. We also add 1×10^{-6} to the variance-predicting stem's output and clip the global norm of mini-batch gradients to ≤ 5 to maintain numerical stability during training.

- DeepEnsEmbCas9_naive by training 20 (homoscedastic) DeepEmbCas9 models initialized with different seeds using the MSE loss objective; and

- DeepEnsEmbCas9 by training 20 DeepEmbCas9-MVE models.

For DeepEmbCas9-MVE, the predicted mean and variance are given directly by its output heads. For DeepEnsEmbCas9_naive, given point predictions $\{\hat{y}_i\}_{i=1}^{20}$ for input $\mathbf{x}_i$, the predicted mean and variance is given by $\mu_{\text{naive}} = M^{-1} \sum_{i=1}^{M} \hat{y}_i$ and $\sigma^2_{\text{naive}} = M^{-1} \sum_{i=1}^{M} (\hat{y}_i - \mu_{\text{naive}})^2$, respectively. For DeepEnsEmbCas9, the predicted mean and variance is as defined for deep ensembles [147].

4.2.11 Quantile calibration

We generated quantile calibration plots and calculated quantile calibration errors to assess whether the predicted uncertaintes were quantile calibrated. Since quantile calibration was assessed for each test set in the benchmark comparisons, we are technically assessing for

group-conditioned quantile calibration, where groups are defined by a specific experimental configuration. To achieve the above, we adopted Kuleshov et al.'s definition of quantile calibration in the regression setting [149], namely that a ML model generating a predictive distribution for datapoint i with label y_i and cumulative distribution function (CDF) $F_i : \mathbb{R} \to [0, 1]$ (i.e., quantile function F_i^{-1}) is quantile calibrated if

$$\forall p \in [0, 1] : \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{y_i \leq F_i^{-1}(p)\} = p$$

with N as the dataset size. We estimate this by selecting confidence levels $p_j \in \{0, 0.01, \ldots, 1\}$ and plotting $\{(p_j, \hat{p}_j)\}_{j=1}^{101}$ where $\hat{p}_j = \frac{1}{N}|\{y_i|F_i(y_i) \leq p_j, i \in [N]\}|$ is the empirical frequency. We also adopted Kuleshov et al.'s definition to confidence intervals (CI), i.e., a ML model is calibrated if

$$\forall p \in [0, 1] : \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{F_i^{-1}(0.5 - \frac{p}{2}) \leq y_i \leq F_i^{-1}(0.5 + \frac{p}{2})\} = p$$

Similarly, we estimate this by selecting confidence intervals $p_j = 0, 0.01, \ldots, 0.99, 1$ and plotting $\{(p_j, \hat{p}_j)\}_{i=1}^{101}$ where $\hat{p}_j = \frac{1}{N}|\{y_i|0.5 - \frac{p_j}{2} \leq F_i(y_i) \leq 0.5 + \frac{p_j}{2}, i \in [N]\}|$ is the CI-based empirical frequency.

We also follow Kuleshov et al.'s approach for computing the quantile calibration error, given by

$$\text{cal}(\{(\hat{p}_j, p_j)\}_{j=1}^{101}) = \sum_{j=1}^{101} (\hat{p}_j - p_j)^2$$

for the two calibration definitions defined above.

4.3 Results

4.3.1 Ranking of pLM-rLM embedding combinations

pLM (↓), rLM(→)	RNA-FM	BEACON-B512	BEACON-B	evo-1-8k	RiNALMo	Average pLM performance
ESM-C-600M	0.901 ± 0.002	0.900 ± 0.002	**0.903 ± 0.002**	0.895 ± 0.003	0.899 ± 0.001	0.900 ± 0.003
ESM-C-300M	0.898 ± 0.002	0.896 ± 0.002	0.898 ± 0.001	0.896 ± 0.002	0.898 ± 0.002	0.897 ± 0.002
ESM-C-6B	0.898 ± 0.004	0.894 ± 0.002	0.899 ± 0.002	0.889 ± 0.004	0.896 ± 0.004	0.895 ± 0.005
ProtT5	0.895 ± 0.002	0.885 ± 0.006	0.890 ± 0.004	0.889 ± 0.003	0.894 ± 0.002	0.890 ± 0.005
Ankh-large	0.892 ± 0.002	0.886 ± 0.001	0.889 ± 0.004	0.889 ± 0.002	0.890 ± 0.003	0.889 ± 0.003
gLM2_650M	0.826 ± 0.003	0.823 ± 0.005	0.799 ± 0.064	0.820 ± 0.006	0.824 ± 0.007	0.818 ± 0.028
ESM3	0.828 ± 0.001	0.795 ± 0.065	0.799 ± 0.068	0.792 ± 0.070	0.752 ± 0.100	0.793 ± 0.068
Average rLM performance	0.877 ± 0.032	0.868 ± 0.045	0.868 ± 0.055	0.867 ± 0.047	0.865 ± 0.063	N/A

Table 4.2: DeepEmbCas9's performance on the validation sets during five-fold cross validation, with results sorted in descending order of averaged pLM and rLM performances.

To determine the best pLM-rLM combination for DeepEmbCas9, we assessed five-fold cross validation Spearman correlations for the 30 pLM-rLM embedding combinations (Table 4.2). ESM-C-600M with BEACON-B yielded the highest average Spearman correlation of 0.903 ± 0.002. Averaging pLM-rLM performances for each of the 6 pLM embeddings, we see that ESM-C embeddings ranked the best with ~0.9 Spearman correlation, followed

by ProtT5 and Ankh-large embeddings with ~0.89 Spearman correlation. Combinations with gLM2_650M and ESM3 embeddings yielded ~0.8 Spearman correlation. Averaging pLM-rLM performances for each of the 5 rLM embeddings, we see that the rLM embeddings perform similarly at $0.86-0.88$ Spearman correlation, with RNA-FM ranked highest at 0.877 ± 0.032.

4.3.2 In-distribution performance comparisons

DeepEnsEmbCas9_naive attains higher Spearman correlation than all individual activity prediction tools on 18 out of 51 benchmark test sets (Figure 4.4, black bars), including 4 mismatched G/gN$_{19}$ interfaces for SpCas9, Sniper-Cas9 and Sniper2L (Figures 4.4B and 4.4G), 10 small Cas9 test sets (2 SlugCas9 variants, 2 SauriCas9 variants, 5 SaCas9 increased-fidelity variants and enCjCas9) from Seo et al. [8] (Figure 4.4I), and 4 other test sets with matched SpCas9, xCas9 and SpCas9-NG interfaces (Figures 4.4A, C and H). As for the remaining 33 test sets, DeepEnsEmbCas9_naive has an average Spearman performance drop of 3.43×10^{-2} compared to the best-performing individual activity prediction tools, with the test set containing matched A/GN$_{19}$ SpCas9-HF1 interfaces from Wang et al. [4] yielding the largest Spearman drop of 0.137 (DeepHF_SpCas9-HF1's 0.881 vs. DeepEnsEmbCas9_naive's 0.744). Among test sets which are outside of the 51 benchmark test sets and lack baselines, DeepEnsEmbCas9_naive attains 0.440-0.922 Spearman correlation in 9 out of 10 test sets (Figures C.10 rows 2-3, C.11, C.15, C.21 row 3, C.22 row 3 and C.23 row 3).

Analogous test performance comparisons for DeepEmbCas9, DeepEnsEmbCas9 and DeepEmbCas9-MVE can be found in Appendix subsubsection C.2.1 "In-distribution performance comparisons of DeepEmbCas9, DeepEnsEmbCas9 and DeepEmbCas9-MVE". Detailed Spearman correlation comparisons between DeepEmbCas9 and individual activity prediction tools for each benchmark test set is provided in Appendix subsubsection C.2.1 "Detailed analysis of DeepEmbCas9's in-distribution performance".

4.3.3 Impact of deep ensembles on in-distribution performance

When considering averaged Spearman correlations across the 51 benchmark test sets, DeepEnsEmbCas9_naive (0.834) attains slightly higher Spearman correlation compared to DeepEmbCas9 (0.814). Likewise, DeepEnsEmbCas9 (0.817) attains slightly higher Spearman correlation compared to DeepEmbCas9-MVE (0.800). In sum, comparing among the 4 DeepEmbCas9 models with and without mean variance estimation and/or ensembling, DeepEnsEmbCas9_naive, DeepEnsEmbCas9, DeepEmbCas9-MVE attain the highest Spearman correlation in 39, 11 and 1 benchmark test set(s) with baselines out of the 51 in total, respectively (Figure 4.4). In particular, we observed the ranking DeepEnsEmbCas9_naive > DeepEmbCas9 > DeepEnsEmbCas9 > DeepEmbCas9-MVE for test sets with matched SpCas9 interfaces from Wang et al. [4] (Figures 4.4A and C.7 row 1) and Kim, Kim et al. [5] (Figures 4.4A and C.8), mismatched G/gN$_{19}$ and matched GN$_{20}$ SpCas9 interfaces from Seo et al. [8] (Figures 4.4B-C and C.16 rows 2 and 4), matched eSpCas9(1.1) and SpCas9-HF1 interfaces from Wang et al. [4] (Figures 4.4D and C.7 rows 2 and 3), matched Sniper-Cas9 interfaces from Kim, Kim, Okafor et al. [10] (Figures 4.4F and C.21 row 1), matched SpCas9-NG interfaces from Kim et al. [6] (Figures 4.4H and C.9 row 3), matched xCas9, SpCas9-NG and VRQR interfaces from Kim, Kim et al. [7] (Figures 4.4H and C.14), matched VRQR, SpCas9-NRCH, SpCas9-NRRH and SpCas9-NRTH from Kim, Choi et al. [9] (Figures 4.4H , C.18 row 3, and C.19), and (mis)matched G/gN21 interfaces for SaCas9 variants, sRGN3.1, SlugCas9 variants and SauriCas9 variants from

Seo et al. [8] (Figures 4.4I and C.24 and C.25). In addition, we observed the ranking DeepEnsEmbCas9_naive > DeepEnsEmbCas9 > DeepEmbCas9 > DeepEmbCas9-MVE in test sets for matched SpCas9 interfaces from Kim, Kim et al. [7] (Figures 4.4A and C.12), matched Sniper2L interfaces from Kim, Kim, Okafor et al. [10] (Figures 4.4F and C.21 row 2), mismatched Sniper-Cas9 and Sniper2L from Kim, Kim, Okafor et al. [10] (Figures 4.4G and C.22 rows 1-2), matched xCas9 interfaces from Kim et al. [6] (Figures 4.4H and C.9 row 2), matched SpCas9-NG and SpRY interfaces from Kim, Choi et al. [9] (Figures 4.4H, C.18 row 2 and C.20 row 2) and (mis)matched Nm1Cas9 interfaces from Seo et al. [8] (Figures 4.4I and C.26 row 4). We observed the ranking DeepEnsEmbCas9_naive > DeepEnsEmbCas9 > DeepEmbCas9-MVE > DeepEmbCas9 for matched HypaCas9 and Sniper-Cas9 interfaces from Kim, Kim et al. [7] (Figures 4.4D, 4.4F and C.13 rows 1 and 3), matched xCas9 interfaces from Kim, Choi et al. [9] (Figures 4.4H and C.18 row 1) and (mis)matched St1Cas9 interfaces from Seo et al. [8] (Figures 4.4I and C.26 row 1).

We saw various rankings among the 11 test sets where DeepEnsEmbCas9 performs best within the four DeepEmbCas9 models. Among the four models, DeepEnsEmbCas9 and DeepEmbCas9 have the highest and lowest Spearman correlations in 9 out of 11 test sets. Among the 9 test sets, DeepEnsEmbCas9_naive ranked higher than DeepEmbCas9-MVE in 6 test sets (i.e., those with matched SpCas9 interfaces from Kim et al. [6] (Figures 4.4A and C.9 row 1), matched eSpCas9(1.1) and SpCas9-HF1 interfaces from Kim, Kim et al. [7] (Figures 4.4D and C.12 rows 2-3), matched Sc++ interfaces from Kim, Choi et al [9] (Figures 4.4E and C.17 row 2), and (mis)matched CjCas9 and enCjCas9 interfaces from Seo et al. [8] (Figures 4.4I and C.26 rows 2-3)), and DeepEmbCas9-MVE ranked higher than DeepEnsEmbCas9_naive in the other 3 test sets (i.e., those with mismatched SpCas9 interfaces from Seo et al. [8] (Figures 4.4B and C.16 row 2), matched evoCas9 interfaces from Kim, Kim et al. [7] (Figures 4.4D and C.13 row 2) and (mis)matched Nm2Cas9 interfaces from Seo et al. [8] (Figures 4.4I and C.26 row 5)). The four models are ranked DeepEnsEmbCas9 > DeepEmbCas9 > DeepEmbCas9-MVE > DeepEnsEmbCas9_naive on the test set with matched SpCas9 interfaces from Kim, Choi et al. [9] (Figures 4.4A and C.17 row 1), and DeepEnsEmbCas9 > DeepEnsEmbCas9_naive > DeepEmbCas9 > DeepEmbCas9-MVE for the test set with matched SpG interfaces from Kim, Choi et al. [9] (Figures 4.4H and C.20 row 1). DeepEmbCas9-MVE ranked best among the four models (DeepEmbCas9-MVE > DeepEnsEmbCas9 > DeepEnsEmbCas9_naive > DeepEmbCas9) in the test set with matched SpCas9 interfaces from Seo et al. [8] (Figures 4.4A and C.16 row 1).

4.3.4 Leave-one-nuclease-out performance comparisons

In the leave-one-nuclease-out extrapolation setting, DeepEnsEmbCas9_naive_omit (Figures C.7-C.26, red bars) attains higher Spearman correlation than all individual activity prediction tools on 17 out of 48 test sets (i.e., the 51 benchmark test sets excluding matched G/gN_{20} SpCas9, St1Cas9 and Nm2Cas9 interfaces). These include 2 mismatched G/gN_{19} SpCas9 interface test sets (Figures C.10 row 1 and C.16 row 2), 1 matched eSpCas9(1.1) interface test set from Kim, Kim et al. [7] (Figure C.12 row 2), 2 matched xCas9 interface test sets from Kim et al. [6] and Kim, Kim et al. [7] (Figures C.9 row 2 and C.14 row 1), 1 matched SpCas9-NG interface test set from Kim et al. [6] (Figure C.9 row 3), 3 matched SpCas9-NRCH, SpCas9-NRRH and SpCas9-NRTH interface test sets from Kim, Choi et al. [9] (Figure C.19), 2 mismatched Sniper-Cas9 and Sniper2L interface test sets from Kim, Kim, Okafor et al. [10] (Figure C.22 rows 1-2), and 6 small Cas9 test sets (3 SaCas9 variants, SauriCas9-KKH, SlugCas9-HF and Nm1Cas9) from Seo et al. [8] (Figures C.24-C.26). As for the remaining 34 test sets, DeepEnsEmbCas9_naive_omit has an

average Spearman performance drop of 5.09×10^{-2} compared to the best-performing individual activity prediction tools not trained on the test sets' nucleases, with the test set containing matched G/gN$_{19}$ Sc++ interfaces from Kim, Choi et al. [9] yielding the largest Spearman drop of 0.266 (DSpCv_Sniper-Cas9's 0.554 vs. DeepEnsEmbCas9_naive_omit's 0.288; Figure C.17 row 2).

Among the test sets which have extrapolation baselines and are not in the 51 benchmark test sets (excluding Kim, Kim et al. [7]'s QQR1 test set), DeepEnsEmbCas9_naive_omit outperforms all individual activity prediction tools on 7 out of 14 test sets, namely mismatched SpCas9 and xCas9 interface test sets from Kim et al. [6] (Figure C.10 rows 1 and 2), (mis)matched SpCas9, xCas9 and SpCas9-NG interface test sets from Kim et al. [6] (Figure C.11), a matched VRER test set from Kim, Kim et al. [6] (Figure C.15 row 3), and (mis)matched Sniper2L interface test sets from Kim, Kim, Okafor et al. [10] (Figure C.23 row 2). As for the remaining 7 test sets, DeepEnsEmbCas9_naive_omit has an average Spearman performance drop of 2.24×10^{-2} compared to the best-performing individual activity prediction tools, with the test set containing matched VQR G/gN$_{19}$ interfaces from Kim, Kim et al. [6] yielding the largest Spearman drop of 5.34×10^{-2} (DCv_VRQR's 0.691 vs. DeepEnsEmbCas9_naive_omit 0.637; Figure C.20 row 2).

Analogous test performance comparisons for DeepEmbCas9, DeepEnsEmbCas9 and DeepEmbCas9-MVE can be found in Appendix subsection "Further leave-one-nuclease-out extrapolation performance" C.2.2. Detailed Spearman correlation comparisons between DeepEmbCas9 and individual activity prediction tools for each benchmark test set is provided in Appendix subsection "DeepEmbCas9 extrapolates to unseen Cas9 variants" C.2.2.

4.3.5 Impact of deep ensembles on leave-one-nuclease-out performance

When considering averaged Spearman correlations across the 51 benchmark test sets in the leave-one-nuclease-out extrapolation setting, DeepEnsEmbCas9_naive (0.786) attains slightly higher Spearman correlation compared to DeepEmbCas9 (0.760). Likewise, DeepEnsEmbCas9 (0.774) attains slightly higher Spearman correlation compared to DeepEmbCas9-MVE (0.756). In sum, comparing among the 4 DeepEmbCas9 models with and without mean variance estimation and/or ensembling, DeepEnsEmbCas9_naive, DeepEnsEmbCas9, DeepEmbCas9-MVE and DeepEmbCas9 attain the highest Spearman correlation in 40, 15, 7 and 1 benchmark test set(s) with(out) baselines out of the 63 in total, respectively (Figures C.7-C.26). Specifically, we observed the ranking DeepEnsEmbCas9_naive > DeepEnsEmbCas9 > DeepEmbCas9-MVE > DeepEmbCas9 for test sets with matched SpCas9-HF1 and eSpCas9(1.1) interfaces from Wang et al. [4], matched SpCas9 interfaces from Kim, Kim et al. [5], matched SpCas9-NG interfaces from Kim et al. [6], matched VQR, VRER and VRQR-HF1 interfaces from Kim, Kim et al. [7], matched SpCas9-NRCH, SpCas9-NRRH, SpCas9-NRTH and SpRY interface from Kim, Choi et al. [9], matched Sniper-Cas9 and Sniper2P interfaces from Kim, Kim, Okafor et al. [9], mismatched Sniper-Cas9 and Sniper2L interfaces from Kim, Kim, Okafor et al. [9], (mis)matched Sniper2P interfaces from Kim, Kim, Okafor et al. [9], and (mis)matched SaCas9, eSaCas9, efSaCas9, SaCas9-HF, SaCas9-KKH, SaCas9-KKH-HF, Sa-SlugCas9 and SlugCas9-HF interfaces from Seo et al. [8]. In addition, we observed the ranking DeepEnsEmbCas9_naive > DeepEnsEmbCas9 > DeepEmbCas9 > DeepEmbCas9-MVE for test datasets with matched xCas9 interfaces from Kim et al. [6], matched VRQR and QQR1 interfaces from Kim, Kim et al. [7], matched Sniper2L interfaces from Kim, Kim, Okafor et al. [10], (mis)matched Sniper-Cas9 and Sniper2L interfaces from Kim, Kim, Okafor et al. [10], and (mis)matched SlugCas9 interfaces from Seo et al. [8]. We observed

DeepEnsEmbCas9_naive > DeepEnsEmbCas9 > DeepEmbCas9-MVE > DeepEmbCas9 for test sets with matched SpCas9 and Sniper-Cas9 interfaces from Kim, Kim et al. [7], mismatched SpCas9 interfaces from Seo et al. [8], matched VRQR interfaces from Kim, Choi et al. [9], and (mis)matched CjCas9 and enCjCas9 interfaces from Seo et al. [8].

We observed various rankings among the 15 test sets where DeepEnsEmbCas9 performs best within the four DeepEmbCas9 models. DeepEmbCas9 has the lowest Spearman correlation among the four models in 12 out of 15 test sets. Among the 12 test sets, DeepEmbCas9-MVE rank higher than DeepEnsEmbCas9_naive for the following 9 test sets (i.e., those with mismatched SpCas9 and xCas9 interfaces from Kim et al. [6], (mis)matched SpCas9 interfaces from Kim et al. [6], matched SpCas9-HF1 interfaces from Kim, Kim et al. [7], matched SpCas9, Sc++ and SpG interfaces from Kim, Choi et al. [9], and matched St1Cas9 and Nm2Cas9 interfaces from Seo et al. [8]), and DeepEnsEmbCas9_naive ranks higher than DeepEmbCas9-MVE for the other 3 test sets (i.e., those with matched SpCas9 interfaces from Kim et al. [6], (mis)matched xCas9 interfaces from Kim et al. [6], and matched eSpCas9(1.1) interfaces from Kim, Kim et al. [7]).

We observed various rankings among the 7 test sets where DeepEmbCas9-MVE performs best within the four DeepEmbCas9 models. DeepEmbCas9 has the lowest Spearman correlations among the four models in 6 out of the 7 test sets. Among the 6 test sets, DeepEnsEmbCas9 ranked higher than DeepEnsEmbCas9_naive on 3 test sets (i.e., those with matched evoCas9 interfaces from Kim, Kim et al. [7], matched SpCas9-NG interfaces from Kim, Choi et al. [9], and (mis)matched Nm1Cas9 interfaces from Seo et al. [8]), and DeepEnsEmbCas9_naive ranked higher than DeepEnsEmbCas9 on the other 3 test sets (i.e., those with matched GN_{20} interfaces from Seo et al. [8], matched xCas9 interfaces from Kim, Choi et al. [9], and (mis)matched SauriCas9 interfaces from Seo et al. [8]).

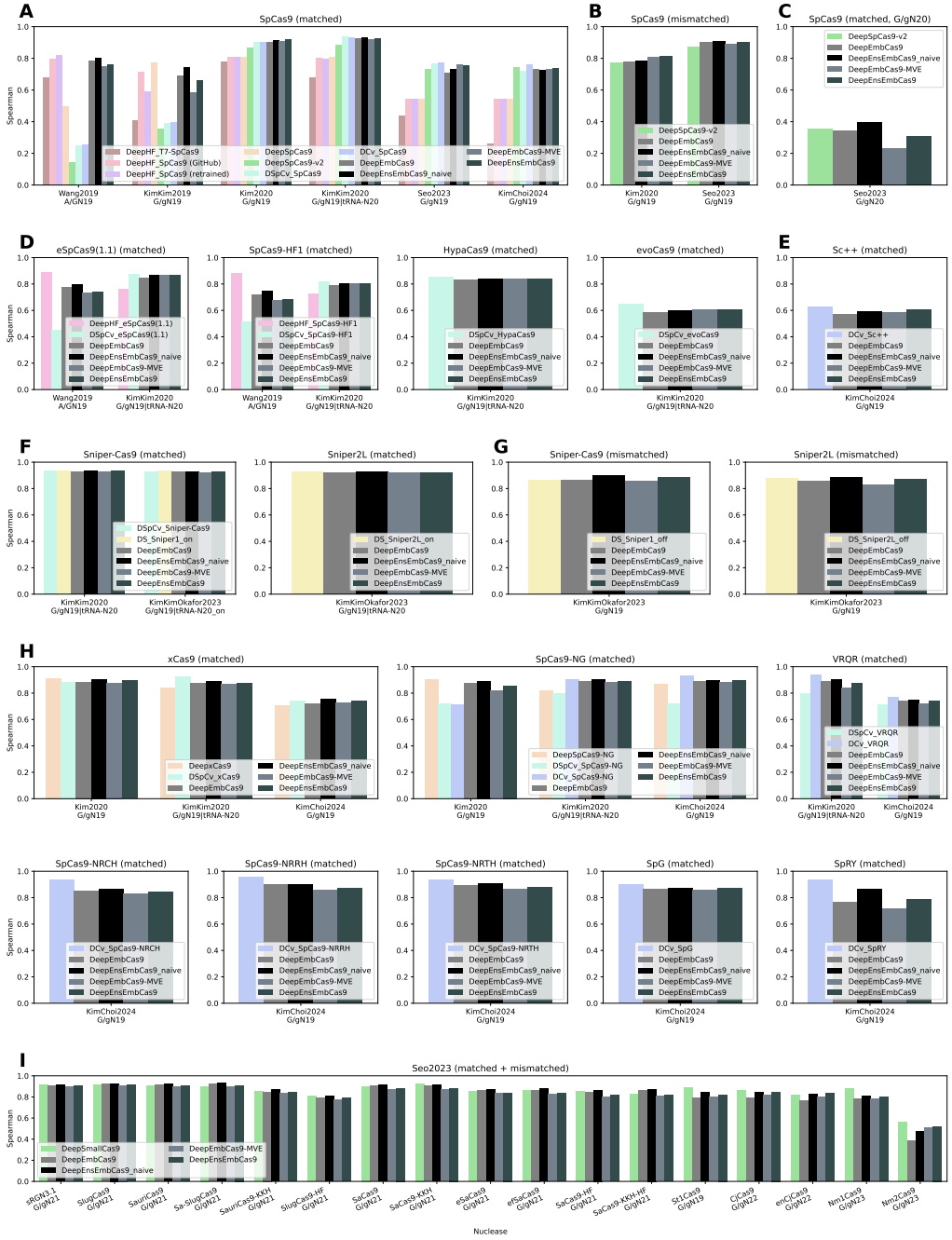


Figure 4.4: Benchmark test Spearman correlation comparison for DeepEmbCas9, DeepEnsEmbCas9_naive, DeepEmbCas9-MVE and DeepEnsEmbCas9 against DeepHF, DeepSpCas9, DeepxCas9, DeepSpCas9-NG, DeepSpCas9variants, DeepSmallCas9, DeepSpCas9-v2, DeepCas9variants and DeepSniper across 39 Cas9 nucleases. The test sets consist of (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} wild type SpCas9 interfaces; (C) matched G/gN_{20} wild type SpCas9 interfaces; (D,E,F) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for 4 increased-fidelity SpCas9 variants (D), Sc++ (E), and 2 Sniper variants (F); (G) mismatched G/gN_{19} interfaces for 2 Sniper variants; (H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for 8 PAM-altered SpCas9 variants; and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

4.3.6 Uncertainty estimation via mean-variance estimation

In the in-distribution setting, DeepEnsEmbCas9 (Figures 4.5 and C.31) and DeepEmbCas9-MVE (Figures C.32 and C.33) attain lower quantile calibration errors than DeepEnsEmbCas9_naive (Figures C.34 and C.35) among the majority of the 51 benchmark datasets. Specifically, as seen in Figure C.36, the left-tailed quantile calibration error of DeepEnsEmbCas9_naive is higher than that of DeepEmbCas9-MVE and DeepEnsEmbCas9 in all 51 benchmark test sets except for those with matched G/gN_{19} SpCas9 interfaces from Seo et al. [8] and matched xCas9 interfaces from Kim et al. [6]. As for CI-based quantile calibration, DeepEnsEmbCas9_naive has higher error than DeepEmbCas9-MVE and DeepEnsEmbCas9 in all 51 benchmark test sets except for those with matched G/gN_{19} and G/gN_{20} SpCas9 interfaces from Seo et al. [8] (Figure C.37).

In the leave-one-nuclease-out extrapolation setting on the 51 benchmark test sets, DeepEnsEmbCas9_omit (Figures C.38 and C.39) and DeepEmbCas9-MVE_omit (Figures C.40 and C.41) generally have worse quantile calibration compared to their in-distribution counterparts. Like DeepEnsEmbCas9_naive, DeepEnsEmbCas9_naive_omit is not quantile calibrated (Figures C.42 and C.43). DeepEnsEmbCas9_naive_omit has higher left-tailed quantile calibration error than DeepEnsEmbCas9_omit and DeepEmbCas9-MVE_omit for all benchmark test sets except for those with mismatched SpCas9 interfaces from Seo et al. [8], matched SpCas9-NG interfaces from Kim et al. [7], matched SpCas9-NG and SpG interfaces from Kim, Choi et al. [9], and (mis)matched Nm1Cas9 and Nm2Cas9 interfaces from Seo et al. [8] (Figure C.44). As for CI-based quantile calibration, DeepEnsEmbCas9_naive_omit has higher left-tailed quantile calibration error than DeepEnsEmbCas9_omit and DeepEmbCas9-MVE_omit for all benchmark test sets except for those with matched GN_{20} SpCas9 interfaces from Seo et al. [8], matched SpCas9-NG interfaces from Kim et al. [6], and (mis)matched St1Cas9 and Nm2Cas9 interfaces from Seo et al. [8] (Figure C.45). Comparing between source studies, we see that DeepEmbCas9-MVE and DeepEnsEmbCas9 have higher left-tailed and CI-based quantile calibration for test sets from Wang et al. [4], Kim, Kim et al. [5] and Kim et al. [6] than in the other 4 studies.

4.3.7 SHAP importance analysis reveals PAM and Cas9 driving DeepEmbCas9 predictions

SHAP importance analysis on the benchmark test sets using different sets of feature groups reveal pertinent feature groups influencing DeepEmbCas9's predicted Cas9 cleavage activity. When calculating SHAP importance of CRISPR-Cas9 complex components in fine resolution, the top three feature groups with the highest SHAP importance are "PAM + downstream + PAM_downstream_Tm", "spacer + spacer_MFE + spacer_GCcount" and "upstream + protospacer + protospacer_Tm" (Figure 4.6A). When calculating SHAP importance of CRISPR-Cas9 complex componets in coarse resolution, the top three feature groups with the highest SHAP importance are the NTS, Cas9 and the spacer (Figure 4.6B). As for Cas9 domains, PLL-WED-PI, Linker and REC_insert have the highest SHAP importance (Figure 4.6C). Using Cas9 regions as feature groups, REC_insert, L2 and REC1-B have the highest SHAP importance (Figure 4.6D). sgRNA regions are ranked spacer > repeat-anti-repeat > polyT > trcrRNA_rest by SHAP importance (Figure 4.6E). Positions -2 and -3, specifically -2G and -3G, on the target strand have the highest SHAP importance among PAM and downstream TS nucleotides (Figures 4.6G and 4.6I). +1G on the spacer is the most important nucleotide among the upstream and heteroduplex nucleotides, with target positions +22 to +17 and spacer/target positions +3 to +1 being important (Figures 4.6F and 4.6H). Importance at positions +1 to +7 is consistent with

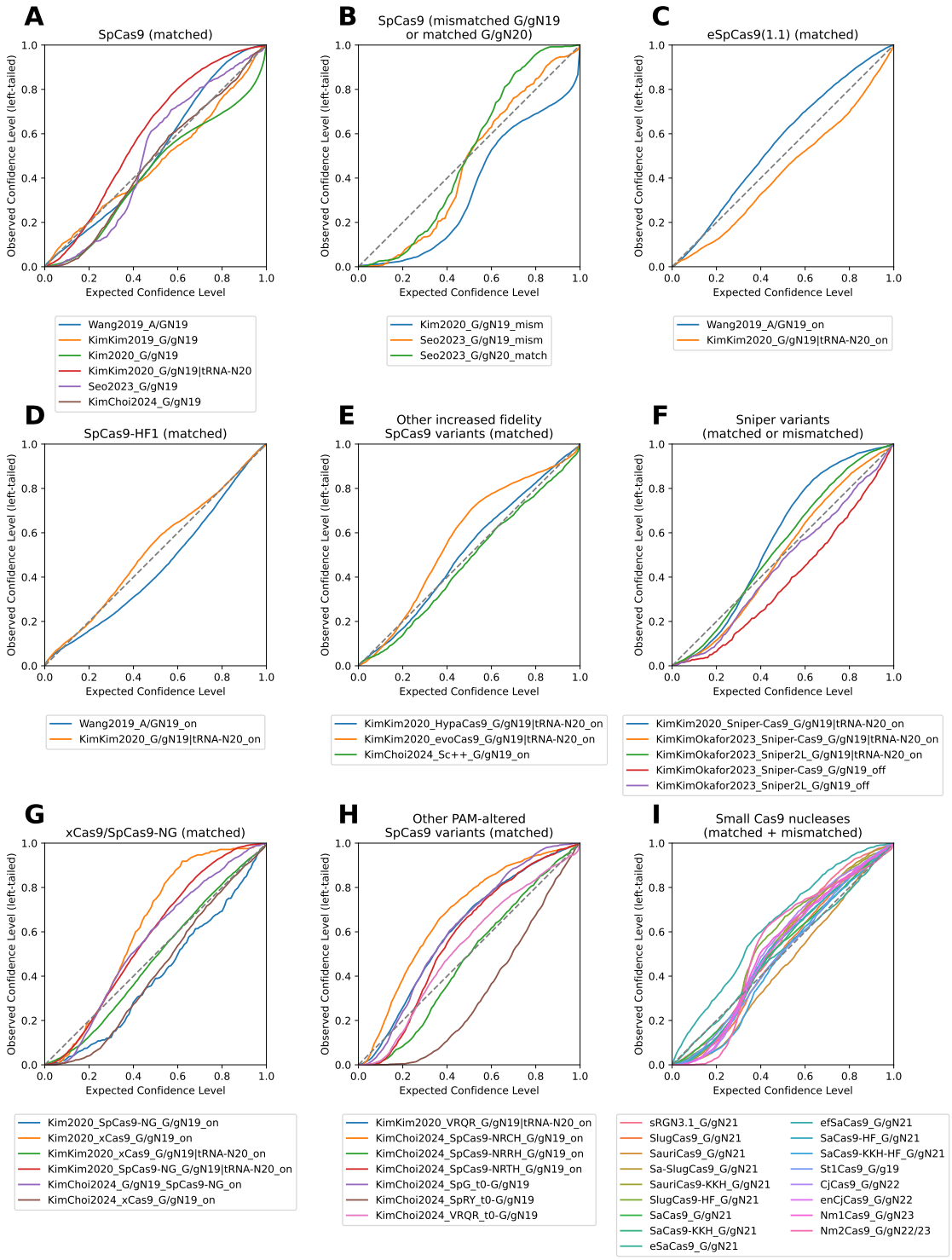


Figure 4.5: Quantile calibration plots for DeepEnsEmbCas9, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

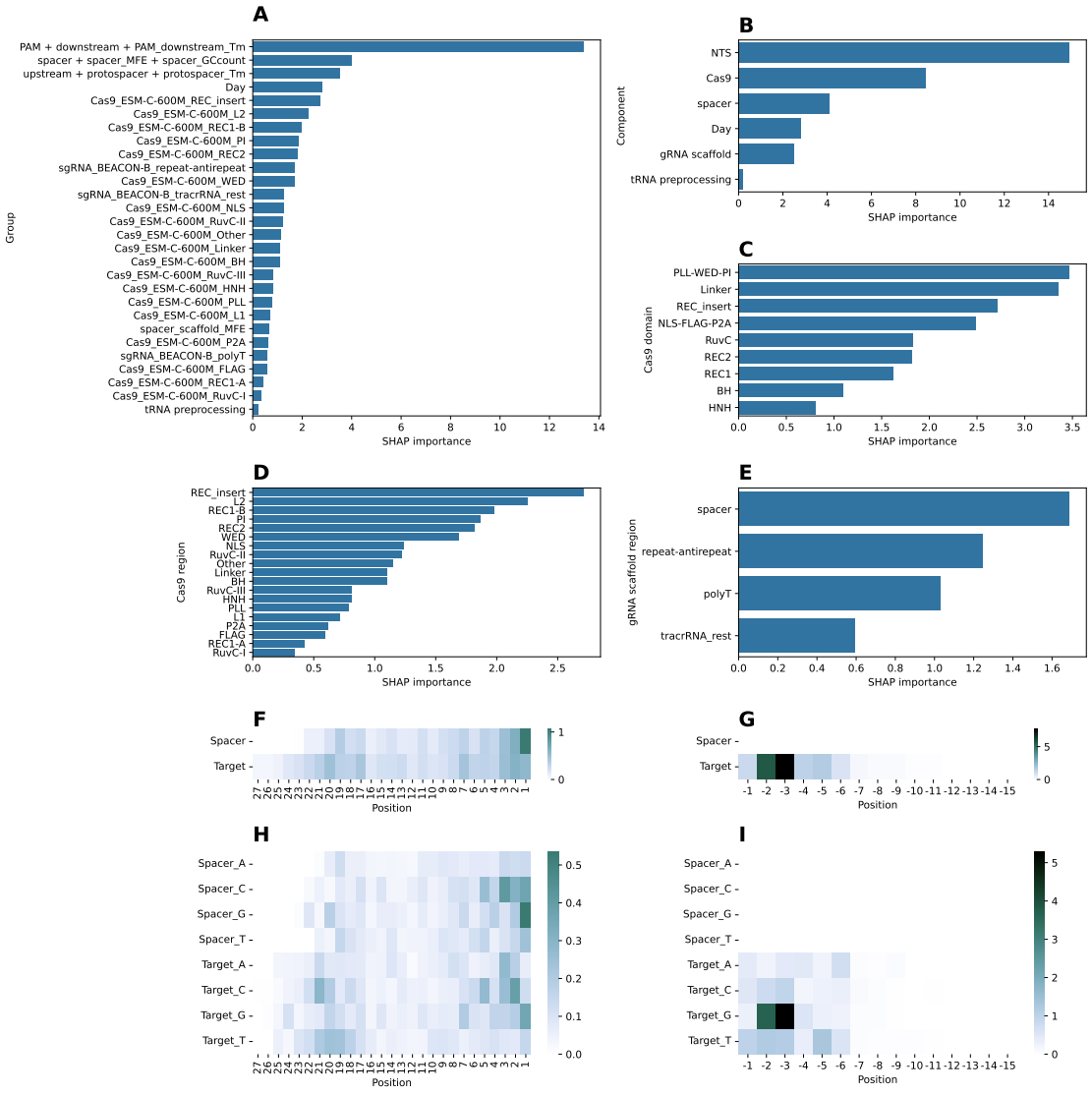


Figure 4.6: SHAP importance analysis of input features in DeepEmbCas9 (ESM-C-600M-BEACON-B combination) on benchmark test sets. The analysis consists of SHAP importance of Cas9 complex components in fine (A) and coarse (B) resolutions; (coarse-grained) Cas9 domains (C) and (fine-grained) Cas9 regions (D) as defined in Figure 4.2C; sgRNA regions (E) as defined in Figure 4.2A; spacer-target nucleotide positions in the upstream-heteroduplex (F) and PAM-downstream regions (G); and spacer-target one-hot encoding features in the upstream-heteroduplex (H) and PAM-downstream regions (I).

the seed region present in all 40 Cas9 variants studied (Figure C.49-C.54).

4.3.8 DeepEmbCas9's predicted activity change from Cas9 mutations reflected in Cas9 domain/region SHAP importances

Using the framework for assessing the influence of input features on cleavage activity change given specific Cas9 mutation(s) or domain substitutions, we see that SHAP importances vary only for Cas9 and NTS (Figure 4.7A). Interestingly, Cas9 SHAP importance positively correlates with NTS SHAP importance (0.593 Spearman correlation; Fig-

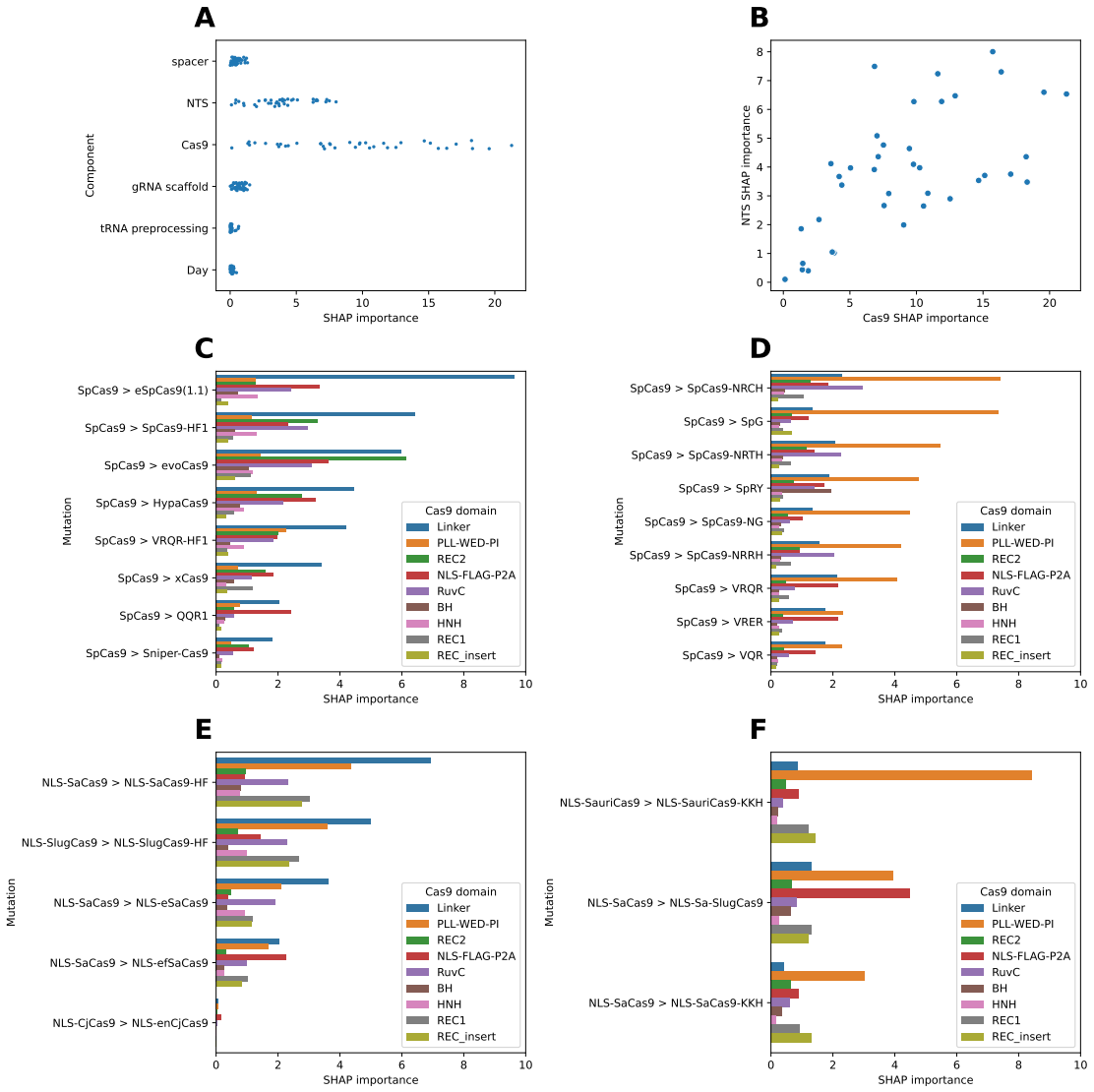


Figure 4.7: CRISPR-Cas9 complex components and Cas9 domains driving DeepEmb-Cas9's change in predicted activity when introducing residue mutations in Cas9. (A) SHAP importance of CRISPR-Cas9 complex components in the 39 nuclease pairs considered. (B) Cas9 vs. NTS SHAP importance in the 39 nuclease pairs considered. (C,D) Cas9 domain importances for SpCas9 variants without (C) and with (D) D1135L/V/N mutations. (E,F) Cas9 domain importances for small Cas9 variants with (E) increased fidelity and (F) PAM-altering variants. Names of Cas9 variants are abbreviated by removing the suffix "-NLS-FLAG-P2A".

ure 4.7B).

When assessing Cas9 domain importances, the Linker Cas9 domain (specifically L2; Figure C.47A) dominates for D1135L/V/N mutation-lacking SpCas9 variants (Figure 4.7C), all of which are increased fidelity variants apart from xCas9 and QQR1. The phenomenon is most pronounced for nuclease pairs SpCas9 > eSpCas9(1.1) and SpCas9 > SpCas9-HF1. In contrast, the PLL-WED-PI domain dominates for D1135 mutation-containing SpCas9 variants (Figure 4.7C), all of which are PAM-altered variants. Specifically, WED among the PLL, WED and PI regions (Figure C.47B). Likewise, L2 in Linker dominates in small

Cas9 mutations resulting in increased fidelity (Figures 4.7E and C.47C), and PLL in PLL-WED-PI has significant importance in PAM-altering small Cas9 mutations (Figure 4.7F and Figure C.47D).

Other nuclease pairs show similar SHAP importance patterns. Low Cas9 region/domain SHAP importances were recorded for nuclease pairs Sniper-Cas9 > Sniper2P, Sniper-Cas9 > Sniper2L, VQR > VRER and VQR > VRQR (Figures C.46A-B and C.48A-B). WED in PLL-WED-PI was found to be important in VRQR > SpG, VRQR > SpRY and SpCas9-HF1 > VRQR-HF1 (Figures C.46C-D and C.48C-D). Both the bridge helix (BH) from Linker and WED in PLL-WED-PI were important in SpG > SpRY(Figures C.46C and C.48C). In VRQR > VRQR-HF1, L2 in Linker and REC2 were found to be important (Figures C.46D and C.48D). nuclease pairs SaCas9-KKH > SaCas9-KKH-HF and SaCas9 > SaCas9-KKH-HF showed high SHAP importance for L2 in Linker and PLL-WED-PI(Figures C.46E and C.48E). PLL in PLL-WED-PI was important in SaCas9-HF > SaCas9-KKH-HF, Linker was important in SlugCas9 > sRGN3.1, and L2 in Linker and NLS-FLAG-P2A were important in SlugCas9 > Sa-SlugCas9 (Figures C.46E-F and C.48E-F).

4.4 Discussion

4.4.1 Ranking of pLM-rLM embedding combinations

When varying pLM embeddings used in DeepEmbCas9, we saw pLM-rLM combinations with ESM-C embeddings attain higher Spearman correlations compared to combinations with ProtT5, Ankh-large, gLM2_650M or ESM3 (Table 4.2). This is possibly a result of ESM-C pLMs being trained on datasets with a significant portion of metagenomic sequences, which would include Cas9 sequences [278]. With regards to the impact of ESM-C model size on performance, we observed higher Spearman correlations observed for ESM-C-600M combinations compared to ESM-C-6B combinations, which is consistent with previous claims saying that medium-sized pLMs already perform well on downstream tasks [295]. The poor performance of ESM3-containing combinations likely reflects the fact that ESM3 [274] — a multimodal generative language model — was trained for controllable protein sequence generation rather than for producing high-quality sequence embeddings. Despite gLM2_650M [292] being a mixed-modality gLM with residue-level sequence representation for protein-coding genes, gLM2_650M combinations did not outperform ESM-C, ProtT5 and Ankh-large combinations, perhaps indicating weaker protein coevolutionary signals in gLM2_650M than in ESM-C models. As for RNA embeddings, evo-1-8k combinations performs comparably to RiNALMo combinations, perhaps indicating weaker RNA coevolution signals in evo-1-8k compared to RNA-FM and BEACON-B.

4.4.2 In-distribution and leave-one-nuclease-out performance

We developed DeepEnsEmbCas9_naive, an ensemble model which predicts CRISPR-Cas9 on/off-target cleavage activity prediction for 40 wild-type/increased-fidelity/PAM-altered SpCas9/small Cas9 variants. In the in-distribution setting, DeepEnsEmbCas9_naive attained comparable Spearman performance to individual activity prediction tools on 38 nucleases across 51 benchmark test sets covering (mis)matched spacer-target interfaces with varying spacer lengths (Figure 4.4). DeepEnsEmbCas9_naive outperformed on several mismatched interface tests, suggesting that pooling of mismatched SpCas9 variant data into one dataset improves model performance. DeepEnsEmbCas9_naive also outperformed DeepSmallCas9 on wild-type and high-fideltiy SaCas9 variants, again suggesting that the

pooling of data from similar variants improves model performance. Comparing between models, DeepEnsEmbCas9_naive has slightly higher Spearman performance than DeepEmbCas9, which is explained by DeepEnsEmbCas9_naive's ensembling of 20 predictions. Evidenced by DeepEmbCas9-MVE and DeepEnsEmbCas9s' Spearman performances, use of mean-variance estimation and a Gaussian negative log-likelihood loss objective reduced Spearman performance, suggesting that the point estimates made by DeepEmbCas9 or DeepEnsEmbCas9_naive are overconfident. It is likely that hyperparameter optimization would further boost DeepEnsEmbCas9_naive's in-distribution performance.

Unlike previous Cas9 cleavage activity models, DeepEnsEmbCas9_naive in theory is able to make indel frequency predictions for any Cas9 nucleases, especially type II-A nucleases given our training dataset's bias towards SpCas9 variants. We demonstrated this in leave-one-nuclease-out extrapolation tasks, where DeepEnsEmbCas9_naive attained comparable extrapolation performance to the best-performing individual activity prediction models. As expected, extrapolation performance deteriorates when extrapolating to nucleases like St1Cas9, Nm1Cas9 and Nm2Cas9, highlighting the need for more cleavage activity data from diverse Cas9 nucleases and variants.

4.4.3 Uncertainty estimation

Our study adds DeepEnsEmbCas9 to the family of uncertainty-aware CRISPR-Cas9 cleavage activity prediction models, including crispAI [293], CRISPR-DBA [296] and CRISPR_-DeepEnsemble [297]. Theory-wise, DeepEnsEmbCas9's built-in uncertainty estimates prevents the model from being overconfident, and allows users to judge the reliability of DeepEnsEmbCas9's prediction, especially when using DeepEnsEmbCas9 to make extrapolation predictions on type II-B or II-C Cas9 nucleases, both of which are severely underrepresented by the 40 Cas9 variants used in our training dataset.

Analyzing left-tailed and CI-based quantile calibration curves in the in-distribution setting, we see that DeepEmbCas9-MVE (Figures C.32 and C.33) and DeepEnsEmbCas9 (Figures 4.5 and C.31) are quantile calibrated (i.e., have good quality uncertainty estimates), whereas DeepEnsEmbCas9_naive (Figures C.34 and C.35) is not quantile calibrated, as corroborated by significantly higher left-tailed and CI-based quantile calibration errors for DeepEnsEmbCas9_naive compared to the low calibration errors for DeepEmbCas9-MVE and DeepEnsEmbCas9 (Figures C.36 and C.37) on most benchmark test sets. Given that mean-variance estimation and ensembling of output predictions captures aleatoric and epistemic uncertainty [298], respectively, similar quantile calibration errors between DeepEmbCas9-MVE and DeepEnsEmbCas9 suggests that DeepEnsEmbCas9's aleatoric uncertainty is larger than epistemic uncertainty. Coupled with DeepEnsEmbCas9's high in-distribution test Spearman correlations, it follows that there is a tradeoff between Spearman performance and quantile correlation among the 3 DL models considered.

As for the leave-one-nuclease-out extrapolation setting, we see that DeepEmbCas9-MVE (Figure C.40 and C.41) and DeepEnsEmbCas9 (Figure C.38 and C.39) exhibit mixed level of quantile calibration dependent on the nuclease and study, whereas DeepEnsEmbCas9_naive (Figures C.42 and C.43) is not quantile calibrated. Interestingly, DeepEmbCas9-MVE and DeepEnsEmbCas9 have higher left-tailed and CI-based quantile calibration errors for Wang et al. [4], Kim, Kim et al. [5] and Kim et al. [6], which is likely a result of training data imbalance arising from discrepancies in genome editing experimental protocols among the different studies.

4.4.4 SHAP importance analysis

SHAP analysis on DeepEmbCas9 primarily reflects the importance of the PAM sequence and Cas9's PAM-interacting (PI) domain. Specifically, -2G and -3G PAM features are shown to be very important (Figure 4.6G and 4.6I), as corroborated by high SHAP importance of feature group "PAM + downstream + PAM_downstream_Tm" in Figure 4.6A and feature group "NTS" in Figure 4.6B. This is consistent with SpCas9 and its increased fidelity variants recognizing NGG PAM. SHAP importances in the NTS-downstream region are observed to taper off beyond PAM position -6 (Figure 4.6G and 4.6I), a result of 36 out of 40 Cas9 variants having $\leq$4 nt PAM. Cas9's PI domain also has high importance, as shown by feature groups "Cas9_ESM-C-600M_REC_insert", "PLL-WED-PI" and"PI" in Figure 4.6A, 4.6C and 4.6D, respectively. This is consistent with Cas9's PI domain participation in PAM binding — the first step towards Cas9 cleavage [38].

SHAP analysis also shows the importance of the PAM-proximal heteroduplex region on CRISPR-Cas9 cleavage activity. In particular, the spacer sequence is shown to be important, as demonstrated by the high SHAP importance of "spacer + spacer_MFE + spacer_GCcount" and "spacer" feature groups in Figure 4.6A, 4.6B and 4.6E, respectively. Heteroduplex positions +1 to +7 have high SHAP importance (Figure 4.6F), relative to other heteroduplex nucleotide positions, corroborating with the low spacer-target mismatch tolerance in the PAM-proximal seed region during R-loop formation [11]. The +1G spacer nucleotide is consistent with feature importance analysis of other prediction tools [158, 80, 56, 4, 183] , with the literature reporting said nucleotide being associated with improved SpCas9 cleavage activity possibly due to its importance during the SpCas9 loading of sgRNA [299].

Cas9 domains apart from Cas9's PI domain are also influential in DeepEmbCas9's cleavage activity prediction. Surprisingly, we see REC_insert, L2, REC1-B and REC2 listed among the top 5 important Cas9 regions (Figure 4.6D). Given that:

- the REC_insert embedding is a non-zero vector only for SpCas9/Sc++ (encoding the REC2 domain) and St1Cas9 (encoding the Wing domain);
- the L2 embedding is a non-zero vector for all Cas9 nucleases except for CjCas9 and enCjCas9;
- the REC1-B embedding is a non-zero vector for only SpCas9, St1Cas9 and Sc++; and
- the REC2 embedding is a non-zero vector for only SpCas9 (encoding the REC3 domain) and St1Cas9/CjCas9/Nm1Cas9/Nm2Cas9/Sc++ (encoding the REC2 domain),

it is possible that DeepEmbCas9 took advantage of zero-valued features from these Cas9 regions — where Cas9 protein architectures differ — in order to distinguish between different Cas9 nucleases.

4.4.5 SHAP importance analysis of nuclease pairs

When using the framework used for assessing the impact of input features on DeepEmbCas9's predicted activity change in 39 Cas9 nuclease pairs, we see that SHAP importance varies substantially only for Cas9 and NTS (Figure 4.7A). Together with Cas9 SHAP importance being positively correlated with NTS SHAP importance (Figure 4.7B), the two observations highlight the presence of feature interactions between the Cas9 and NTS

components. Plotting SHAP domain and region importances for the 39 Cas9 nuclease pairs, we see that L2 region in the Linker domain (abbreviated as Linker/L2 onwards) dominates in SHAP importance when mutating from SpCas9 to increased-fidelity SpCas9 variants (i.e., eSpCas9(1.1), SpCas9-HF1, evoCas9, HypaCas9, Sniper-Cas9 and xCas9; Figure 4.7C). Interestingly, xCas9 does not have high PLL-WED-PI importance, possibly hinting at the conflation between SpCas9 and xCas9's PAM preferences by DeepEmbCas9. VRQR-HF1 has higher PLL-WED-PI importance in addition to Linker/L2 importance, which is consistent with the Cas9 PI domain VRQR mutations in VRQR-HF1. evoCas9 has high REC2 (i.e., SpCas9 REC3) importance in addition to high Linker/L2 importance, which is consistent with the 4 SpCas9 REC3 mutations possessed by evoCas9. QQR1's low Linker/L2 and PLL-WED-PI/WED importance is likely due to QQR1's overall low cleavage activity. We observe similar importance patterns for SaCas9 and SlugCas9 high-fidelity nuclease pairs (Figure 4.7E).

In contrast, we see that the WED region in the PLL-WED-PI domain (abbreviated as PLL-WED-PI/WED onwards) dominates in SHAP importance for all PAM-altered SpCas9 variants except xCas9 and QQR1. Moreover, only xCas9 and QQR1 lack mutations at WED residue D1135 among the PAM-altered SpCas9 variants. Combined, this suggests the use of D1135-related features in the WED part of the Cas9 pLM embedding by DeepEmbCas9 for Cas9 cleavage activity prediction (Figure 4.7E). Indeed, D1135 is a residue which interacts with the minor groove of the PAM duplex via electrostatic repulsion between the negatively charged aspartate and sugar-phosphate backbone, whereas:

- D1135V in VQR, VRER and VRQR stabilizes the PAM duplex by replacing the electrostatic repulsion with van der Waals forces between valine and the sugar-phosphate backbone;
- D1135L in SpG and SpRY acts similarly to D1135V, but introduces a hydrophobic bulky leucine instead, which together with S1136V sterically pushes the third PAM base towards Q1335; and
- D1135N is a consensus mutation found during directed evolution when developing SpCas9-NRRH, SpCas9-NRTH and SpCas9-NRCH.

We also observe similar importance patterns for PAM-altered SaCas9 and SauriCas9 mutants (Figure 4.7F).

4.4.6 Limitations

We acknowledge several limitations in this study. Regarding data, there is an uneven amount of Cas9 nucleases and gRNA scaffolds in our training and test datasets, so DeepEmbCas9 may overfit certain experimental configuration types from specific studies. Due to limited DNA/RNA bulge data in existing indel frequency library screens, e.g., in Seo et al. [8], this study does not consider guide-target interfaces with DNA/RNA bulges. Library data used in this study also does not contain interfaces with 4-6 mismatches, limiting DeepEmbCas9's predictive accuracy on such interfaces.

As for the embeddings, the Cas9 pLM and sgRNA rLM embeddings used in DeepEmbCas9 do not account for Cas9-gRNA scaffold interactions, so the effect of such interactions would need to be learned from the activity labels. DeepEmbCas9 also only accept Cas9 proteins with known domain boundaries as input, since this information is required for generating Cas9 pLM embeddings from the per-residue pLM embedding matrix. For analogous reasons, DeepEmbCas9 can only work with sgRNA scaffolds that have similar structure to those in our dataset.

With respect to SHAP interpretation, the low number of Cas9 variants limits the accuracy of the SHAP importances generated for the protein domains. The low count is also why we did not consider per-Cas9-residue protein representations and by extension per-Cas9-residue SHAP importances. There are also limitations to using SHAP feature importance scores for ML model interpretability, as suggested by Kumar et al. [300]. In terms of mathematical issues, conditional-based SHAP implementations output differing Shapley value attributions from the additive explanation model depending on whether highly correlated features are included in the input feature set, whereas interventional-based SHAP implementations rely on model predictions on out-of-distribution datapoints. Additionally, explanations derived from SHAP values are not naturally contrastive.

4.5 Conclusion

In this study, we developed a family of 4 DeepEmbCas9 models — DeepEmbCas9, DeepEmbCas9-MVE, DeepEnsEmbCas9_naive and DeepEnsEmbCas9 — for cleavage activity prediction of Cas9 variants by:

1. representing all three components of the CRISPR-Cas9 complex, i.e., sgRNA, DNA and Cas9, as input features;
2. using a unified guide-target interface to align spacer and target sequences from different Cas9 nucleases;
3. adopting inductive biases compatible with Cas9's biophysical mechanism; and
4. training on a curated dataset with >1.75 million datapoints spanning 40 Cas9 variants and 16 gRNA scaffolds.

Obtained by ensembling predictions, DeepEnsEmbCas9_naive attains comparable performance in both in-distribution and leave-one-nuclease-out extrapolation settings when compared to suitable individual cleavage activity models. We also built DeepEnsEmbCas9, trading off a small Spearman performance drop with well-calibrated uncertainty estimates. SHAP importance analysis of DeepEmbCas9 on all benchmark test set datapoints reaffirms the structural and functional importance of Cas9's PLL-WED-PI domain and the PAM sequence for Cas9 binding — a prerequisite for Cas9 cleavage. Furthermore, SHAP importance analysis of DeepEmbCas9 on 39 nuclease pairs show that Linker and PLL-WED-PI features contribute significantly to predicted activity change for increased-fidelity and PAM-altering Cas9 mutations, respectively.

DeepEmbCas9 models confer advantages over existing Cas9 activity models leveraging Cas9 protein information. Compared to PLM_CRISPR [179], DeepEmbCas9 trains on 33 more Cas9 variants spanning beyond increased-fidelty SpCas9 variants. Unlike STING_CRISPR [267], DeepEmbCas9 scales better with increasing guide-target training data while maintaining the ability to assess Cas9 domain importances. In contrast to PAMmla [157] and CICERO [272], DeepEmbCas9 models directly address the problem of cleavage activity prediction rather than the subproblem of Cas9-target PAM binding prediction. Altogether, DeepEmbCas9 models serve as the first step towards generalistic interpretable DL-based models capable of predicting cleavage activity for diverse combinations of wild-type/engineered/pLM-generated nucleases and guide-target interfaces in the Cas9 protein family.

Chapter 5

Conclusion & future work

In Chapter 2, we have considered 19 epigenetic features (13 nucleosome organization-related features and 6 experimental epigenetic features) available from the crisprSQL Cas9 off-target database. Correlation analysis between epigenetic feature and Cas9 activity showed that experimental epigenetic features did not correlate with Cas9 cleavage activity. This is in contrast to the computed BDM-based scores, which correlate much stronger than the experimental epigenetic features. In particular, Nucleotide BDM's ability to separate between sites with and without activity, together with the finding [206] that low Nucleotide BDM values are good indicators of nucleosome positions, highlight that nucleosome positioning can inhibit Cas9 activity. SHAP analysis also revealed GC147, Nucleotide BDM and NuPoP (Affinity) as important features influencing Cas9 off-target activity in both XGBoost and CNN models. However, NuPoP (Occup) referring to NuPoP occupancy score did not show notable importance, so it is nucleosome positioning rather than nucleosome occupancy that inhibits Cas9 activity. Specifically, we saw that low Nucleotide BDM and high NuPoP (Affinity) values negatively impact off-target activity, corroberating with our theory of positioned nucleosome inhibiting Cas9 activity. GC147's high importance is also explained by its high correlation with DNA bendability and high DNA bendability being favored during R-loop formation. The six experimental epigenetic scores did not show notable importance in the SHAP summary plots, which together with the previous correlation analysis casts doubt in the utility of the 4 experimental epigenetic features widely used in deep learning models such as DeepCRISPR. In sum, the correlation and SHAP analyses informs us of 3 computational nucleosome organization-related scores — GC147, NuPoP (Affinity), and Nucleotide BDM — that have notable importance when building a full SpCas9 cleavage activity model with spacer-target interface features and R-loop formation energy scores.

Indeed, insights from Chapter 2 have already been leveraged for the construction of a SpCas9 off-target cleavage activity prediction model built using both sequence features, R-loop formation energy scores and 3 computational nucleosome organization-related scores in my colleague Florian's work (see Appendix D.1), where the 3 computational nucleosome organization-related scores do contribute to the model's prediction. As for future work, it would be interesting to see whether the same conclusions hold in the case of other SpCas9 variants and small Cas9 nucleases, other editing modalities (e.g., base editing and prime editing), and other computational nucleosomal or epigenetic tools (e.g., Enformer [301] and Borzoi [302]) that were developed after the completion of research in Chapter 2.

In Chapter 3, we used traditional machine learning and feature importances to filter for a set of 30 residue-resolved physico-chemical/structural features which capture the protein 3D nanoenvironment of the guide-target pair for the 28 on- and off-target SpCas9

interfaces we investigated using all-atom molecular dynamics. Training a ML model then enabled us to learn the functional relationship between the nanoenvironment and Cas9 activity for these 28 guide-target pairs. Interestingly, by construction, such a ML model would be able to predict the effect of SpCas9 residue mutations on cleavage activity. Finally, SHAP importance analysis of the ML model revealed physicochemical/structural descriptors and SpCas9 residues that were influential in the model's prediction, where some SpCas9 residues from the four residue hotspots overlapped with SpCas9 residues discussed in the literature. For future work, we envision to scale up the study by expanding the MD-derived dataset to include more guide-target pairs. Alternatively, we could use the framework developed in this study to examine other SpCas9-nucleic acid interactions in the CRISPR-Cas9 complex, e.g., interactions between the PAM-interacting domain and PAM sequence.

Chapter 4 then concerns the development of DeepEmbCas9 — a DL-based cleavage activity model which theoretically is capable of making predictions for any Cas9 variant-sgRNA scaffold combination. Trained on a large and carefully curated guide-target lentiviral library-based indel frequency dataset consisting of 40 Cas9 variants and 16 gRNA scaffolds, DeepEmbCas9 was the first to explicitly represent all three components of the CRISPR-Cas9 complex in its input feature set, and used inductive biases inspired by Cas9's cleavage mechanism. To accommodate Cas9 nucleases with varying spacer and PAM lengths, DeepEmbCas9 also used a unified guide-target interface which right aligns the spacer sequence immediately before the start of the target's PAM sequence. Further considerations on boosting performance and uncertainty quantification then led to the development of DeepEmbCas9-MVE, DeepEnsEmbCas9_naive and DeepEnsEmbCas9. Considering all four models, we saw that DeepEnsEmbCas9_naive had the best test performance among the four models, and performed favourably when compared to existing single-variant Cas9 cleavage activity prediction tools. As for uncertainty quantification, DeepEnsEmbCas9 was shown to have quantile-calibrated uncertainty estimates albeit with lower Spearman test performance compared to DeepEnsEmbCas9_naive. Through SHAP importance analysis of DeepEmbCas9 on all benchmark test sets, we saw that Cas9's PLL-WED-PI domain and the target PAM sequence both have high SHAP importance, which reaffirms the prerequisite step of Cas9 binding required for Cas9 activity. A case study on 39 Cas9 nuclease pairs showed us that the Linker and PLL-WED-PI are influential in changing DeepEmbCas9's predicted cleavage activity when swapping one Cas9 variant to another. For future work, we envision the use of AlphaFold 3 residue embeddings for cross-variant Cas9 cleavage activity prediction, since such embeddings would be able to capture protein-nucleic acid interactions — something that DeepEmbCas9 has to learn from the cleavage activity labels. Furthermore, hyperparameter optimization and imbalanced sampling of datapoints would likely boost DeepEmbCas9 performance on the current benchmark test sets. Finally, additional availability of cleavage from future experimental studies would enhance

All in all, by studying structural features beyond the guide-target interface, assessing the impact of structural features on Cas9 cleavage activity, and building interpretable ML/DL model which incorporate the structural features, we pave the way towards building structure-aware and generalistic Cas9 cleavage activity prediction models.

Bibliography

[1] Sridharan, S., Nicholls, A., and Honig, B. (1992) A new vertex algorithm to calculate solvent accessible surface-areas. In *Faseb Journal* FEDERATION AMER SOC EXP BIOL 9650 ROCKVILLE PIKE, BETHESDA, MD 20814-3998 Vol. 6, pp. A174–A174.

[2] Hubbard, S. J., Thornton, J. M., et al. Naccess. `http://www.bioinf.manchester.ac.uk/naccess/` (1993) Last accessed June 4, 2024.

[3] Shrake, A. and Rupley, J. A. (1973) Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *Journal of molecular biology,* **79**(2), 351–371.

[4] Wang, D., Zhang, C., Wang, B., Li, B., Wang, Q., Liu, D., Wang, H., Zhou, Y., Shi, L., Lan, F., et al. (2019) Optimized CRISPR guide RNA design for two high-fidelity Cas9 variants by deep learning. *Nature communications,* **10**(1), 4284.

[5] Kim, H. K., Kim, Y., Lee, S., Min, S., Bae, J. Y., Choi, J. W., Park, J., Jung, D., Yoon, S., and Kim, H. H. (2019) SpCas9 activity prediction by DeepSpCas9, a deep learning–based model with high generalization performance. *Science advances,* **5**(11), eaax9249.

[6] Kim, H. K., Lee, S., Kim, Y., Park, J., Min, S., Choi, J. W., Huang, T. P., Yoon, S., Liu, D. R., and Kim, H. H. (2020) High-throughput analysis of the activities of xCas9, SpCas9-NG and SpCas9 at matched and mismatched target sequences in human cells. *Nature biomedical engineering,* **4**(1), 111–124.

[7] Kim, N., Kim, H. K., Lee, S., Seo, J. H., Choi, J. W., Park, J., Min, S., Yoon, S., Cho, S.-R., and Kim, H. H. (2020) Prediction of the sequence-specific cleavage activity of Cas9 variants. *Nature Biotechnology,* **38**(11), 1328–1336.

[8] Seo, S.-Y., Min, S., Lee, S., Seo, J. H., Park, J., Kim, H. K., Song, M., Baek, D., Cho, S.-R., and Kim, H. H. (2023) Massively parallel evaluation and computational prediction of the activities and specificities of 17 small Cas9s. *Nature Methods,* **20**(7), 999–1009.

[9] Kim, N., Choi, S., Kim, S., Song, M., Seo, J. H., Min, S., Park, J., Cho, S.-R., and Kim, H. H. (2024) Deep learning models to predict the editing efficiencies and outcomes of diverse base editors. *Nature Biotechnology,* **42**(3), 484–497.

[10] Kim, Y.-h., Kim, N., Okafor, I., Choi, S., Min, S., Lee, J., Bae, S.-M., Choi, K., Choi, J., Harihar, V., et al. (2023) Sniper2L is a high-fidelity Cas9 variant with high activity. *Nature chemical biology,* **19**(8), 972–980.

[11] Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. A., and Charpentier, E. (2012) A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity. *Science,* **337**(6096), 816–821.

[12] Barrangou, R., Fremaux, C., Deveau, H., Richards, M., Boyaval, P., Moineau, S., Romero, D. A., and Horvath, P. (2007) CRISPR Provides Acquired Resistance Against Viruses in Prokaryotes. *Science,* **315**(5819), 1709–1712.

[13] Cong, L., Ran, F. A., Cox, D., Lin, S., Barretto, R., Habib, N., Hsu, P. D., Wu, X., Jiang, W., Marraffini, L. A., and Zhang, F. (2013) Multiplex Genome Engineering Using CRISPR/Cas Systems. *Science,* **339**(6121), 819–823.

[14] Barrangou, R. (2014) Cas9 targeting and the CRISPR revolution. *Science,* **344**(6185), 707–708.

[15] Barrangou, R. and Doudna, J. A. (2016) Applications of CRISPR technologies in research and beyond. *Nature biotechnology,* **34**(9), 933–941.

[16] Ledford, H. and Callaway, E. (2020) Pioneers of CRISPR gene editing win chemistry Nobel. *Nature,* **586**(7829), 346–347.

[17] Philippidis, A. (2024) CASGEVY makes history as FDA approves first CRISPR/Cas9 genome edited therapy. *Human gene therapy,* **35**(1-2), 1–4.

[18] Singh, A., Irfan, H., Fatima, E., Nazir, Z., Verma, A., and Akilimali, A. (2024) Revolutionary breakthrough: FDA approves CASGEVY, the first CRISPR/Cas9 gene therapy for sickle cell disease. *Annals of Medicine and Surgery,* **86**(8), 4555–4559.

[19] Sander, J. D. and Joung, J. K. (Apr, 2014) CRISPR-Cas systems for editing, regulating and targeting genomes. *Nat Biotechnol,* **32**(4), 347–355.

[20] Tsai, S. Q. and Joung, J. K. (May, 2016) Defining and improving the genome-wide specificities of CRISPR-Cas9 nucleases. *Nat Rev Genet,* **17**(5), 300–312.

[21] Adli, M. (05, 2018) The CRISPR tool kit for genome editing and beyond. *Nat Commun,* **9**(1), 1911.

[22] Zhang, F. (2019) Development of CRISPR-Cas systems for genome editing and beyond. *Quarterly Reviews of Biophysics,* **52**, e6.

[23] Parsi, K. M., Hennessy, E., Kearns, N., and Maehr, R. (2017) Using an Inducible CRISPR-dCas9-KRAB Effector System to Dissect Transcriptional Regulation in Human Embryonic Stem Cells. *Methods Mol Biol,* **1507**, 221–233.

[24] Maeder, M. L., Linder, S. J., Cascio, V. M., Fu, Y., Ho, Q. H., and Joung, J. K. (Oct, 2013) CRISPR RNA-guided activation of endogenous human genes. *Nat Methods,* **10**(10), 977–979.

[25] Perez-Pinera, P., Kocak, D. D., Vockley, C. M., Adler, A. F., Kabadi, A. M., Polstein, L. R., Thakore, P. I., Glass, K. A., Ousterout, D. G., Leong, K. W., Guilak, F., Crawford, G. E., Reddy, T. E., and Gersbach, C. A. (Oct, 2013) RNA-guided gene activation by CRISPR-Cas9-based transcription factors. *Nat Methods,* **10**(10), 973–976.

[26] Ma, H., Naseri, A., Reyes-Gutierrez, P., Wolfe, S. A., Zhang, S., and Pederson, T. (2015) Multicolor CRISPR labeling of chromosomal loci in human cells. *Proceedings of the National Academy of Sciences,* **112**(10), 3002–3007.

[27] Shao, S., Zhang, W., Hu, H., Xue, B., Qin, J., Sun, C., Sun, Y., Wei, W., and Sun, Y. (05, 2016) Long-term dual-color tracking of genomic loci by modified sgRNAs of the CRISPR/Cas9 system. *Nucleic Acids Res,* **44**(9), e86.

[28] Kearns, N. A., Pham, H., Tabak, B., Genga, R. M., Silverstein, N. J., Garber, M., and Maehr, R. (May, 2015) Functional annotation of native enhancers with a Cas9-histone demethylase fusion. *Nat Methods,* **12**(5), 401–403.

[29] Kwon, D. Y., Zhao, Y. T., Lamonica, J. M., and Zhou, Z. (05, 2017) Locus-specific histone deacetylation using a synthetic CRISPR-Cas9-based HDAC. *Nat Commun,* **8**, 15315.

[30] Wang, H., Xu, X., Nguyen, C. M., Liu, Y., Gao, Y., Lin, X., Daley, T., Kipniss, N. H., La Russa, M., and Qi, L. S. (2018) CRISPR-Mediated Programmable 3D Genome Positioning and Nuclear Organization. *Cell,* **175**(5), 1405 – 1417.e14.

[31] Komor, A. C., Kim, Y. B., Packer, M. S., Zuris, J. A., and Liu, D. R. (2016) Programmable editing of a target base in genomic DNA without double-stranded DNA cleavage. *Nature,* **533**(7603), 420–424.

[32] Gaudelli, N. M., Komor, A. C., Rees, H. A., Packer, M. S., Badran, A. H., Bryson, D. I., and Liu, D. R. (2017) Programmable base editing of A• T to G• C in genomic DNA without DNA cleavage. *Nature,* **551**(7681), 464–471.

[33] 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., et al. (2019) Search-and-replace genome editing without double-strand breaks or donor DNA. *Nature,* **576**(7785), 149–157.

[34] Doudna, J. A. and Charpentier, E. (nov, 2014) The new frontier of genome engineering with CRISPR-Cas9. *Science,* **346**(6213), 1258096.

[35] Gasiunas, G., Barrangou, R., Horvath, P., and Siksnys, V. (2012) Cas9–crRNA ribonucleoprotein complex mediates specific DNA cleavage for adaptive immunity in bacteria. *Proceedings of the National Academy of Sciences,* **109**(39), E2579–E2586.

[36] Fonfara, I., Le Rhun, A., Chylinski, K., Makarova, K. S., Lécrivain, A.-L., Bzdrenga, J., Koonin, E. V., and Charpentier, E. (11, 2013) Phylogeny of Cas9 determines functional exchangeability of dual-RNA and Cas9 among orthologous type II CRISPR-Cas systems. *Nucleic Acids Research,* **42**(4), 2577–2590.

[37] Anders, C., Niewoehner, O., Duerst, A., and Jinek, M. (2014) Structural basis of PAM-dependent target DNA recognition by the Cas9 endonuclease. *Nature,* **513**(7519), 569–573.

[38] Jiang, F. and Doudna, J. A. (May, 2017) CRISPR–Cas9 Structures and Mechanisms. *Annual Review of Biophysics,* **46**(1), 505–529.

[39] Bravo, J. P. K., Liu, M.-S., Hibshman, G. N., Dangerfield, T. L., Jung, K., McCool, R. S., Johnson, K. A., and Taylor, D. W. (mar, 2022) Structural basis for mismatch surveillance by CRISPR–Cas9. *Nature,* **603**(7900), 343–347.

[40] Palermo, G., Miao, Y., Walker, R. C., Jinek, M., and McCammon, J. A. (September, 2016) Striking Plasticity of CRISPR-Cas9 and Key Role of Non-target DNA, as Revealed by Molecular Simulations. *ACS Central Science,* **2**(10), 756–763.

[41] Palermo, G., Miao, Y., Walker, R. C., Jinek, M., and McCammon, J. A. (June, 2017) CRISPR-Cas9 conformational activation as elucidated from enhanced molecular simulations. *Proceedings of the National Academy of Sciences,* **114**(28), 7260–7265.

[42] 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., et al. (2013) DNA targeting specificity of RNA-guided Cas9 nucleases. *Nature biotechnology,* **31**(9), 827–832.

[43] Pattanayak, V., Lin, S., Guilinger, J. P., Ma, E., Doudna, J. A., and Liu, D. R. (2013) High-throughput profiling of off-target DNA cleavage reveals RNA-programmed Cas9 nuclease specificity. *Nature biotechnology,* **31**(9), 839–843.

[44] Fu, Y., Foden, J. A., Khayter, C., Maeder, M. L., Reyon, D., Joung, J. K., and Sander, J. D. (2013) High-frequency off-target mutagenesis induced by CRISPR-Cas nucleases in human cells. *Nature biotechnology,* **31**(9), 822–826.

[45] Cho, S. W., Kim, S., Kim, Y., Kweon, J., Kim, H. S., Bae, S., and Kim, J.-S. (2014) Analysis of off-target effects of CRISPR/Cas-derived RNA-guided endonucleases and nickases. *Genome research,* **24**(1), 132–141.

[46] Zhang, X.-H., Tee, L. Y., Wang, X.-G., Huang, Q.-S., and Yang, S.-H. (2015) Off-target effects in CRISPR/Cas9-mediated genome engineering. *Molecular therapy Nucleic acids,* **4**.

[47] Zhang, L., Rube, H. T., Vakulskas, C. A., Behlke, M. A., Bussemaker, H. J., and Pufall, M. A. (apr, 2020) Systematic in vitro profiling of off-target affinity, cleavage and efficiency for CRISPR enzymes. *Nucleic Acids Research,* **48**(9), 5037–5053.

[48] Mitchell, B. P., Hsu, R. V., Medrano, M. A., Zewde, N. T., Narkhede, Y. B., and Palermo, G. (mar, 2020) Spontaneous Embedding of DNA Mismatches Within the RNA:DNA Hybrid of CRISPR-Cas9. *Frontiers in Molecular Biosciences,* **7**, 39.

[49] Cradick, T. J., Fine, E. J., Antico, C. J., and Bao, G. (Nov, 2013) CRISPR/Cas9 systems targeting $\hat{\mathrm{I}}^{\mathbf{2}}$-globin and CCR5 genes have substantial off-target activity. *Nucleic Acids Res,* **41**(20), 9584–9592.

[50] Lin, Y., Cradick, T. J., Brown, M. T., Deshmukh, H., Ranjan, P., Sarode, N., Wile, B. M., Vertino, P. M., Stewart, F. J., and Bao, G. (Jun, 2014) CRISPR/Cas9 systems have off-target activity with insertions or deletions between target DNA and guide RNA sequences. *Nucleic Acids Res,* **42**(11), 7473–7485.

[51] Guilinger, J. P., Pattanayak, V., Reyon, D., Tsai, S. Q., Sander, J. D., Joung, J. K., and Liu, D. R. (Apr, 2014) Broad specificity profiling of TALENs results in engineered nucleases with improved DNA-cleavage specificity. *Nat Methods,* **11**(4), 429–435.

[52] Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., Zhou, C., Zhu, C., Chen, K., Duan, B., et al. (2018) DeepCRISPR: optimized CRISPR guide RNA design by deep learning. *Genome biology,* **19**(1), 80.

[53] Lin, J., Zhang, Z., Zhang, S., Chen, J., and Wong, K.-C. (2020) CRISPR-Net: A Recurrent Convolutional Network Quantifies CRISPR Off-Target Activities with Mismatches and Indels. *Advanced Science,* **7**(13), 1903562.

[54] Störtz, F. and Minary, P. (10, 2020) crisprSQL: a novel database platform for CRISPR/Cas off-target cleavage assays. *Nucleic Acids Research,* gkaa885.

[55] 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., and Zhang, F. (Sep, 2013) DNA targeting specificity of RNA-guided Cas9 nucleases. *Nat Biotechnol,* **31**(9), 827–832.

[56] Doench, J. G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E. W., Donovan, K. F., Smith, I., Tothova, Z., Wilen, C., Orchard, R., et al. (2016) Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature biotechnology,* **34**(2), 184–191.

[57] Listgarten, J., Weinstein, M., Kleinstiver, B. P., Sousa, A. A., Joung, J. K., Crawford, J., Gao, K., Hoang, L., Elibol, M., Doench, J. G., et al. (2018) Prediction of off-target activities for the end-to-end design of CRISPR guide RNAs. *Nature biomedical engineering,* **2**(1), 38–47.

[58] Yan, J., Xue, D., Chuai, G., Gao, Y., Zhang, G., and Liu, Q. (11, 2020) Benchmarking and integrating genome-wide CRISPR off-target detection and prediction. *Nucleic Acids Research,* **48**(20), 11370–11379.

[59] Alkan, F., Wenzel, A., Anthon, C., Havgaard, J. H., and Gorodkin, J. (2018) CRISPR-Cas9 off-targeting assessment with nucleic acid duplex energy parameters. *Genome biology,* **19**(1), 177.

[60] Zhang, D., Hurst, T., Duan, D., and Chen, S.-J. (2019) Unified energetics analysis unravels SpCas9 cleavage activity for optimal gRNA design. *Proceedings of the National Academy of Sciences,* **116**(18), 8693–8698.

[61] Fujita, T., Yuno, M., and Fujii, H. (07, 2016) Allele-specific locus binding and genome editing by CRISPR at the p16INK4a locus. *Sci Rep,* **6**, 30485.

[62] Kallimasioti-Pazi, E. M., Thelakkad Chathoth, K., Taylor, G. C., Meynert, A., Ballinger, T., Kelder, M. J. E., Lalevée, S., Sanli, I., Feil, R., and Wood, A. J. (12, 2018) Heterochromatin delays CRISPR-Cas9 mutagenesis but does not influence the outcome of mutagenic DNA repair. *PLOS Biology,* **16**(12), 1–22.

[63] O'Geen, H., Henry, I. M., Bhakta, M. S., Meckler, J. F., and Segal, D. J. (Mar, 2015) A genome-wide analysis of Cas9 binding specificity using ChIP-seq and targeted sequence capture. *Nucleic Acids Res,* **43**(6), 3389–3404.

[64] Horlbeck, M. A., Witkowsky, L. B., Guglielmi, B., Replogle, J. M., Gilbert, L. A., Villalta, J. E., Torigoe, S. E., Tjian, R., and Weissman, J. S. (mar, 2016) Nucleosomes impede Cas9 access to DNA in vivo and in vitro. *eLife,* **5**, e12677.

[65] Kuscu, C., Arslan, S., Singh, R., Thorpe, J., and Adli, M. (Jul, 2014) Genome-wide analysis reveals characteristics of off-target sites bound by the Cas9 endonuclease. *Nat Biotechnol,* **32**(7), 677–683.

[66] Chen, Y., Zeng, S., Hu, R., Wang, X., Huang, W., Liu, J., Wang, L., Liu, G., Cao, Y., and Zhang, Y. (08, 2017) Using local chromatin structure to improve CRISPR/Cas9 efficiency in zebrafish. *PLOS ONE,* **12**(8), 1–19.

[67] Jensen, K. T., Fløe, L., Petersen, T. S., Huang, J., Xu, F., Bolund, L., Luo, Y., and Lin, L. (07, 2017) Chromatin accessibility and guide sequence secondary structure affect CRISPR-Cas9 gene editing efficiency. *FEBS Lett,* **591**(13), 1892–1901.

[68] Song, L. and Crawford, G. E. (February, 2010) DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells. *Cold Spring Harbor protocols,* **2010**(2), pdb.prot5384.

[69] Meissner, A., Gnirke, A., Bell, G. W., Ramsahoye, B., Lander, E. S., and Jaenisch, R. (01, 2005) Reduced representation bisulfite sequencing for comparative high-resolution DNA methylation analysis. *Nucleic Acids Research,* **33**(18), 5868–5877.

[70] Gu, H., Smith, Z. D., Bock, C., Boyle, P., Gnirke, A., and Meissner, A. (Apr, 2011) Preparation of reduced representation bisulfite sequencing libraries for genome-scale DNA methylation profiling. *Nat Protoc,* **6**(4), 468–481.

[71] O'Geen, H., Echipare, L., and Farnham, P. J. (2011) Using ChIP-seq technology to generate high-resolution profiles of histone modifications. *Methods Mol Biol,* **791**, 265–286.

[72] Verkuijl, S. A. and Rots, M. G. (02, 2019) The influence of eukaryotic chromatin state on CRISPR-Cas9 editing efficiencies. *Curr Opin Biotechnol,* **55**, 68–73.

[73] Wu, X., Scott, D. A., Kriz, A. J., Chiu, A. C., Hsu, P. D., Dadon, D. B., Cheng, A. W., Trevino, A. E., Konermann, S., Chen, S., Jaenisch, R., Zhang, F., and Sharp, P. A. (Jul, 2014) Genome-wide binding of the CRISPR endonuclease Cas9 in mammalian cells. *Nat Biotechnol,* **32**(7), 670–676.

[74] Davey, C. A., Sargent, D. F., Luger, K., Maeder, A. W., and Richmond, T. J. (2002) Solvent mediated interactions in the structure of the nucleosome core particle at 1.9 Å resolution. *Journal of molecular biology,* **319**(5), 1097–1113.

[75] Struhl, K. and Segal, E. (March, 2013) Determinants of nucleosome positioning. *Nature Structural & Molecular Biology,* **20**(3), 267–273.

[76] Schones, D. E., Cui, K., Cuddapah, S., Roh, T. Y., Barski, A., Wang, Z., Wei, G., and Zhao, K. (Mar, 2008) Dynamic regulation of nucleosome positioning in the human genome. *Cell,* **132**(5), 887–898.

[77] Kuan, P. F., Huebert, D., Gasch, A., and Keles, S. (2009) A non-homogeneous hidden-state model on first order differences for automatic detection of nucleosome positions. *Stat Appl Genet Mol Biol,* **8**, Article29.

[78] Hinz, J. M., Laughery, M. F., and Wyrick, J. J. (2015) Nucleosomes inhibit Cas9 endonuclease activity in vitro. *Biochemistry,* **54**(48), 7063–7066.

[79] Isaac, R. S., Jiang, F., Doudna, J. A., Lim, W. A., Narlikar, G. J., and Almeida, R. (apr, 2016) Nucleosome breathing and remodeling constrain CRISPR-Cas9 function. *eLife,* **5**, e13450.

[80] Doench, J. G., Hartenian, E., Graham, D. B., Tothova, Z., Hegde, M., Smith, I., Sullender, M., Ebert, B. L., Xavier, R. J., and Root, D. E. (2014) Rational design of highly active sgRNAs for CRISPR-Cas9–mediated gene inactivation. *Nature biotechnology,* **32**(12), 1262–1267.

[81] Wu, Z., Yang, H., and Colosi, P. (2010) Effect of genome size on AAV vector packaging. *Molecular Therapy,* **18**(1), 80–86.

[82] Wang, D., Zhang, F., and Gao, G. (2020) CRISPR-based therapeutic genome editing: strategies and in vivo delivery by AAV vectors. *Cell,* **181**(1), 136–150.

[83] Slaymaker, I. M., Gao, L., Zetsche, B., Scott, D. A., Yan, W. X., and Zhang, F. (2016) Rationally engineered Cas9 nucleases with improved specificity. *Science,* **351**(6268), 84–88.

[84] Kleinstiver, B. P., Pattanayak, V., Prew, M. S., Tsai, S. Q., Nguyen, N. T., Zheng, Z., and Joung, J. K. (2016) High-fidelity CRISPR–Cas9 nucleases with no detectable genome-wide off-target effects. *Nature,* **529**(7587), 490–495.

[85] Chen, J. S., Dagdas, Y. S., Kleinstiver, B. P., Welch, M. M., Sousa, A. A., Harrington, L. B., Sternberg, S. H., Joung, J. K., Yildiz, A., and Doudna, J. A. (2017) Enhanced proofreading governs CRISPR–Cas9 targeting accuracy. *Nature,* **550**(7676), 407–410.

[86] Casini, A., Olivieri, M., Petris, G., Montagna, C., Reginato, G., Maule, G., Lorenzin, F., Prandi, D., Romanel, A., Demichelis, F., et al. (2018) A highly specific SpCas9 variant is identified by in vivo screening in yeast. *Nature biotechnology,* **36**(3), 265–271.

[87] Lee, J. K., Jeong, E., Lee, J., Jung, M., Shin, E., Kim, Y.-h., Lee, K., Jung, I., Kim, D., Kim, S., et al. (2018) Directed evolution of CRISPR-Cas9 to increase its specificity. *Nature communications,* **9**(1), 3048.

[88] Ciciani, M., Demozzi, M., Pedrazzoli, E., Visentin, E., Pezzè, L., Signorini, L. F., Blanco-Miguez, A., Zolfo, M., Asnicar, F., Casini, A., et al. (2022) Automated identification of sequence-tailored Cas9 proteins using massive metagenomic data. *Nature Communications,* **13**(1), 6474.

[89] Xie, H., Ge, X., Yang, F., Wang, B., Li, S., Duan, J., Lv, X., Cheng, C., Song, Z., Liu, C., et al. (2020) High-fidelity SaCas9 identified by directional screening in human cells. *PLoS biology,* **18**(7), e3000747.

[90] Tan, Y., Chu, A. H., Bao, S., Hoang, D. A., Kebede, F. T., Xiong, W., Ji, M., Shi, J., and Zheng, Z. (2019) Rationally engineered Staphylococcus aureus Cas9 nucleases with high genome-wide specificity. *Proceedings of the National Academy of Sciences,* **116**(42), 20969–20976.

[91] Hu, Z., Zhang, C., Wang, S., Gao, S., Wei, J., Li, M., Hou, L., Mao, H., Wei, Y., Qi, T., Liu, H., Liu, D., Lan, F., Lu, D., Wang, H., Li, J., and Wang, Y. (03, 2021) Discovery and engineering of small SlugCas9 with broad targeting range and high specificity and activity. *Nucleic Acids Research,* **49**(7), 4008–4019.

[92] Nakagawa, R., Ishiguro, S., Okazaki, S., Mori, H., Tanaka, M., Aburatani, H., Yachie, N., Nishimasu, H., and Nureki, O. (2022) Engineered Campylobacter jejuni Cas9 variant with enhanced activity and broader targeting range. *Communications biology,* **5**(1), 211.

[93] Kleinstiver, B. P., Prew, M. S., Tsai, S. Q., Nguyen, N. T., Topkar, V. V., Zheng, Z., and Joung, J. K. (2015) Broadening the targeting range of Staphylococcus aureus CRISPR-Cas9 by modifying PAM recognition. *Nature biotechnology,* **33**(12), 1293–1298.

[94] Schmidt, M. J., Gupta, A., Bednarski, C., Gehrig-Giannini, S., Richter, F., Pitzler, C., Gamalinda, M., Galonska, C., Takeuchi, R., Wang, K., et al. (2021) Improved CRISPR genome editing using small highly active and specific engineered RNA-guided nucleases. *Nature communications,* **12**(1), 4219.

[95] Dang, Y., Jia, G., Choi, J., Ma, H., Anaya, E., Ye, C., Shankar, P., and Wu, H. (2015) Optimizing sgRNA structure to improve CRISPR-Cas9 knockout efficiency. *Genome biology,* **16**(1), 280.

[96] Kim, H. K. and Kim, H. H. (2025) Evaluation and prediction of guide RNA activities in genome-editing tools. *Nature Reviews Bioengineering,* pp. 1–16.

[97] Kuchner, O. and Arnold, F. H. (1997) Directed evolution of enzyme catalysts. *Trends in biotechnology,* **15**(12), 523–530.

[98] Chatterjee, P., Jakimo, N., Lee, J., Amrani, N., Rodríguez, T., Koseki, S. R., Tysinger, E., Qing, R., Hao, S., Sontheimer, E. J., et al. (2020) An engineered ScCas9 with broad PAM range and high specificity and activity. *Nature Biotechnology,* **38**(10), 1154–1158.

[99] Kleinstiver, B. P., Prew, M. S., Tsai, S. Q., Topkar, V. V., Nguyen, N. T., Zheng, Z., Gonzales, A. P., Li, Z., Peterson, R. T., Yeh, J.-R. J., et al. (2015) Engineered CRISPR-Cas9 nucleases with altered PAM specificities. *Nature,* **523**(7561), 481–485.

[100] Anders, C., Bargsten, K., and Jinek, M. (2016) Structural plasticity of PAM recognition by engineered variants of the RNA-guided endonuclease Cas9. *Molecular cell,* **61**(6), 895–902.

[101] Nishimasu, H., Shi, X., Ishiguro, S., Gao, L., Hirano, S., Okazaki, S., Noda, T., Abudayyeh, O. O., Gootenberg, J. S., Mori, H., et al. (2018) Engineered CRISPR-Cas9 nuclease with expanded targeting space. *Science,* **361**(6408), 1259–1262.

[102] Walton, R. T., Christie, K. A., Whittaker, M. N., and Kleinstiver, B. P. (2020) Unconstrained genome targeting with near-PAMless engineered CRISPR-Cas9 variants. *Science,* **368**(6488), 290–296.

[103] Esvelt, K. M., Carlson, J. C., and Liu, D. R. (2011) A system for the continuous directed evolution of biomolecules. *Nature,* **472**(7344), 499–503.

[104] Suzuki, T., Miller, C., Guo, L.-T., Ho, J. M., Bryson, D. I., Wang, Y.-S., Liu, D. R., and Söll, D. (2017) Crystal structures reveal an elusive functional domain of pyrrolysyl-tRNA synthetase. *Nature chemical biology,* **13**(12), 1261–1266.

[105] Miller, S. M., Wang, T., and Liu, D. R. (2020) Phage-assisted continuous and non-continuous evolution. *Nature protocols,* **15**(12), 4101–4127.

[106] Hu, J. H., Miller, S. M., Geurts, M. H., Tang, W., Chen, L., Sun, N., Zeina, C. M., Gao, X., Rees, H. A., Lin, Z., et al. (2018) Evolved Cas9 variants with broad PAM compatibility and high DNA specificity. *Nature,* **556**(7699), 57–63.

[107] Miller, S. M., Wang, T., Randolph, P. B., Arbab, M., Shen, M. W., Huang, T. P., Matuszek, Z., Newby, G. A., Rees, H. A., and Liu, D. R. (2020) Continuous evolution of SpCas9 variants compatible with non-G PAMs. *Nature biotechnology,* **38**(4), 471–481.

[108] Esvelt, K. M., Mali, P., Braff, J. L., Moosburner, M., Yaung, S. J., and Church, G. M. (2013) Orthogonal Cas9 proteins for RNA-guided gene regulation and editing. *Nature methods,* **10**(11), 1116–1121.

[109] Müller, M., Lee, C. M., Gasiunas, G., Davis, T. H., Cradick, T. J., Siksnys, V., Bao, G., Cathomen, T., and Mussolino, C. (2016) Streptococcus thermophilus CRISPR-Cas9 systems enable specific editing of the human genome. *Molecular Therapy,* **24**(3), 636–644.

[110] Agudelo, D., Carter, S., Velimirovic, M., Duringer, A., Rivest, J.-F., Levesque, S., Loehr, J., Mouchiroud, M., Cyr, D., Waters, P. J., et al. (2020) Versatile and robust genome editing with Streptococcus thermophilus CRISPR1-Cas9. *Genome research,* **30**(1), 107–117.

[111] Zhang, Y., Zhang, H., Xu, X., Wang, Y., Chen, W., Wang, Y., Wu, Z., Tang, N., Wang, Y., Zhao, S., et al. (2020) Catalytic-state structure and engineering of Streptococcus thermophilus Cas9. *Nature Catalysis,* **3**(10), 813–823.

[112] Hou, Z., Zhang, Y., Propson, N. E., Howden, S. E., Chu, L.-F., Sontheimer, E. J., and Thomson, J. A. (2013) Efficient genome engineering in human pluripotent stem cells using Cas9 from Neisseria meningitidis. *Proceedings of the National Academy of Sciences,* **110**(39), 15644–15649.

[113] Lee, C. M., Cradick, T. J., and Bao, G. (2016) The Neisseria meningitidis CRISPR-Cas9 system enables specific genome editing in mammalian cells. *Molecular Therapy,* **24**(3), 645–654.

[114] Amrani, N., Gao, X. D., Liu, P., Edraki, A., Mir, A., Ibraheim, R., Gupta, A., Sasaki, K. E., Wu, T., Donohoue, P. D., et al. (2018) NmeCas9 is an intrinsically high-fidelity genome-editing platform. *Genome Biology,* **19**(1), 214.

[115] Nishimasu, H., Cong, L., Yan, W. X., Ran, F. A., Zetsche, B., Li, Y., Kurabayashi, A., Ishitani, R., Zhang, F., and Nureki, O. (2015) Crystal structure of Staphylococcus aureus Cas9. *Cell,* **162**(5), 1113–1126.

[116] Ran, F. A., Cong, L., Yan, W. X., Scott, D. A., Gootenberg, J. S., Kriz, A. J., Zetsche, B., Shalem, O., Wu, X., Makarova, K. S., et al. (2015) In vivo genome editing using Staphylococcus aureus Cas9. *Nature,* **520**(7546), 186–191.

[117] Friedland, A. E., Baral, R., Singhal, P., Loveluck, K., Shen, S., Sanchez, M., Marco, E., Gotta, G. M., Maeder, M. L., Kennedy, E. M., et al. (2015) Characterization of Staphylococcus aureus Cas9: a smaller Cas9 for all-in-one adeno-associated virus delivery and paired nickase applications. *Genome biology,* **16**(1), 257.

[118] Najm, F. J., Strand, C., Donovan, K. F., Hegde, M., Sanson, K. R., Vaimberg, E. W., Sullender, M. E., Hartenian, E., Kalani, Z., Fusi, N., et al. (2018) Orthologous CRISPR–Cas9 enzymes for combinatorial genetic screens. *Nature biotechnology,* **36**(2), 179–189.

[119] Tycko, J., Barrera, L. A., Huston, N. C., Friedland, A. E., Wu, X., Gootenberg, J. S., Abudayyeh, O. O., Myer, V. E., Wilson, C. J., and Hsu, P. D. (2018) Pairwise library screen systematically interrogates Staphylococcus aureus Cas9 specificity in human cells. *Nature communications,* **9**(1), 2962.

[120] Kim, E., Koo, T., Park, S. W., Kim, D., Kim, K., Cho, H.-Y., Song, D. W., Lee, K. J., Jung, M. H., Kim, S., et al. (2017) In vivo genome editing with a small Cas9 orthologue derived from Campylobacter jejuni. *Nature communications,* **8**(1), 14500.

[121] Yamada, M., Watanabe, Y., Gootenberg, J. S., Hirano, H., Ran, F. A., Nakane, T., Ishitani, R., Zhang, F., Nishimasu, H., and Nureki, O. (2017) Crystal structure of the minimal Cas9 from Campylobacter jejuni reveals the molecular diversity in the CRISPR-Cas9 systems. *Molecular cell,* **65**(6), 1109–1121.

[122] Edraki, A., Mir, A., Ibraheim, R., Gainetdinov, I., Yoon, Y., Song, C.-Q., Cao, Y., Gallant, J., Xue, W., Rivera-Pérez, J. A., et al. (2019) A compact, high-accuracy Cas9 with a dinucleotide PAM for in vivo genome editing. *Molecular cell,* **73**(4), 714–726.

[123] Hu, Z., Wang, S., Zhang, C., Gao, N., Li, M., Wang, D., Wang, D., Liu, D., Liu, H., Ong, S.-G., et al. (2020) A compact Cas9 ortholog from Staphylococcus Auricularis (SauriCas9) expands the DNA targeting scope. *PLoS biology,* **18**(3), e3000686.

[124] Bishop, C. M. and Nasrabadi, N. M. (2006) Pattern recognition and machine learning, Vol. 4, Springer, .

[125] Murphy, K. P. (2022) Probabilistic machine learning: an introduction, MIT press, .

[126] Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* New York, NY, USA: Association for Computing Machinery KDD '16 p. 785–794.

[127] Friedman, J. H. (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics,* pp. 1189–1232.

[128] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018) CatBoost: unbiased boosting with categorical features. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., (eds.), *Advances in Neural Information Processing Systems*, Curran Associates, Inc. Vol. 31, pp. 6639–6649.

[129] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., (eds.), *Advances in Neural Information Processing Systems*, Curran Associates, Inc. Vol. 30, pp. 3146–3154.

[130] LeCun, Y., Bengio, Y., and Hinton, G. (2015) Deep learning. *nature,* **521**(7553), 436–444.

[131] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016) Deep learning, Vol. 1, MIT press Cambridge, .

[132] Fukushima, K. (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics,* **36**(4), 193–202.

[133] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989) Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems,* **2**.

[134] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012) Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems,* **25**.

[135] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017) Attention is all you need. *Advances in neural information processing systems,* **30**.

[136] Bottou, L. (1998) Online algorithms and stochastic approximations. *Online learning in neural networks,*.

[137] Ruder, S. (2016) An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747,*.

[138] Kingma, D. P. and Ba, J. (2015) Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y., (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, .

[139] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015) Software available from tensorflow.org.

[140] Chollet, F. et al. Keras. `https://keras.io` (2015).

[141] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035 Curran Associates, Inc. Red Hook, NY, USA.

[142] Falcon, W. and The PyTorch Lightning team PyTorch Lightning. (March, 2019).

[143] Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., and Chintala, S. (April, 2024) PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)* ACM.

[144] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs. (2018).

[145] Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural computation,* **9**(8), 1735–1780.

[146] Lundberg, S. M. and Lee, S.-I. (2017) A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* Red Hook, NY, USA: Curran Associates Inc. NIPS'17 p. 4768–4777.

[147] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017) Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., (eds.), *Advances in Neural Information Processing Systems*, Curran Associates, Inc. Vol. 30, .

[148] Nix, D. and Weigend, A. (1994) Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)* Vol. 1, pp. 55–60 vol.1.

[149] Kuleshov, V., Fenner, N., and Ermon, S. (2018) Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning* PMLR pp. 2796–2804.

[150] Kim, H., Um, E., Cho, S.-R., Jung, C., Kim, H., and Kim, J.-S. (2011) Surrogate reporters for enrichment of cells with nuclease-induced mutations. *Nature methods,* **8**(11), 941–943.

[151] Ramakrishna, S., Cho, S. W., Kim, S., Song, M., Gopalappa, R., Kim, J.-S., and Kim, H. (2014) Surrogate reporter-based enrichment of cells containing RNA-guided Cas9 nuclease-induced mutations. *Nature communications,* **5**(1), 3378.

[152] Mashal, R. D., Koontz, J., and Sklar, J. (1995) Detection of mutations by cleavage of DNA heteroduplexes with bacteriophage resolvases. *Nature genetics,* **9**(2), 177–183.

[153] Vouillot, L., Thélie, A., and Pollet, N. (03, 2015) Comparison of T7E1 and Surveyor Mismatch Cleavage Assays to Detect Mutations Triggered by Engineered Nucleases. *G3 Genes—Genomes—Genetics,* **5**(3), 407–415.

[154] Ran, F. A., Hsu, P. D., Wright, J., Agarwala, V., Scott, D. A., and Zhang, F. (2013) Genome engineering using the CRISPR-Cas9 system. *Nature protocols,* **8**(11), 2281–2308.

[155] Liu, Q., He, D., and Xie, L. (2019) Prediction of off-target specificity and cell-specific fitness of CRISPR-Cas System using attention boosted deep learning and network-based gene feature. *PLoS computational biology,* **15**(10), e1007480.

[156] Jones, S. K., Hawkins, J. A., Johnson, N. V., Jung, C., Hu, K., Rybarski, J. R., Chen, J. S., Doudna, J. A., Press, W. H., and Finkelstein, I. J. (01, 2021) Massively parallel kinetic profiling of natural and engineered CRISPR nucleases. *Nat Biotechnol,* **39**(1), 84–93.

[157] Silverstein, R. A., Kim, N., Kroell, A.-S., Walton, R. T., Delano, J., Butcher, R. M., Pacesa, M., Smith, B. K., Christie, K. A., Ha, L. L., et al. (2025) Custom CRISPR—Cas9 PAM variants via scalable engineering and machine learning. *Nature,* pp. 1–3.

[158] Montague, T. G., Cruz, J. M., Gagnon, J. A., Church, G. M., and Valen, E. (05, 2014) CHOPCHOP: a CRISPR/Cas9 and TALEN web tool for genome editing. *Nucleic Acids Research,* **42**(W1), W401–W407.

[159] Stemmer, M., Thumberger, T., del Sol Keyer, M., Wittbrodt, J., and Mateo, J. L. (2015) CCTop: an intuitive, flexible and reliable CRISPR/Cas9 target prediction tool. *PloS one,* **10**(4), e0124633.

[160] Singh, R., Kuscu, C., Quinlan, A., Qi, Y., and Adli, M. (2015) Cas9-chromatin binding information enables more accurate CRISPR off-target prediction. *Nucleic acids research,* **43**(18), e118–e118.

[161] Bae, S., Park, J., and 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.

[162] Haeussler, M., Schönig, K., Eckert, H., Eschstruth, A., Mianné, J., Renaud, J.-B., Schneider-Maunoury, S., Shkumatava, A., Teboul, L., Kent, J., et al. (2016) Evaluation of off-target and on-target scoring algorithms and integration into the guide RNA selection tool CRISPOR. *Genome biology,* **17**(1), 148.

[163] Cortes, C. and Vapnik, V. (1995) Support-vector networks. *Machine learning,* **20**(3), 273–297.

[164] Breiman, L. (2001) Random forests. *Machine learning,* **45**(1), 5–32.

[165] Freund, Y. and Schapire, R. E. (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences,* **55**(1), 119–139.

[166] Chari, R., Yeo, N. C., Chavez, A., and Church, G. M. (2017) sgRNA Scorer 2.0: a species-independent model to predict CRISPR/Cas9 activity. *ACS synthetic biology,* **6**(5), 902–904.

[167] Lin, J. and Wong, K.-C. (09, 2018) Off-target predictions in CRISPR-Cas9 gene editing using deep learning. *Bioinformatics,* **34**(17), i656–i663.

[168] Crosetto, N., Mitra, A., Silva, M. J., Bienko, M., Dojer, N., Wang, Q., Karaca, E., Chiarle, R., Skrzypczak, M., Ginalski, K., et al. (2013) Nucleotide-resolution DNA double-strand break mapping by next-generation sequencing. *Nature methods,* **10**(4), 361–365.

[169] Tsai, S. Q., Nguyen, N. T., Malagon-Lopez, J., Topkar, V. V., Aryee, M. J., and Joung, J. K. (2017) CIRCLE-seq: a highly sensitive in vitro screen for genome-wide CRISPR–Cas9 nuclease off-targets. *Nature methods,* **14**(6), 607–614.

[170] 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., et al. (2015) GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. *Nature biotechnology,* **33**(2), 187–197.

[171] Kim, D., Bae, S., Park, J., Kim, E., Kim, S., Yu, H. R., Hwang, J., Kim, J.-I., and Kim, J.-S. (2015) Digenome-seq: genome-wide profiling of CRISPR-Cas9 off-target effects in human cells. *Nature methods,* **12**(3), 237–243.

[172] 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., et al. (2019) Unbiased detection of CRISPR off-targets in vivo using DISCOVER-Seq. *Science,* **364**(6437), 286–289.

[173] Wienert, B., Wyman, S. K., Yeh, C. D., Conklin, B. R., and Corn, J. E. (2020) CRISPR off-target detection with DISCOVER-seq. *Nature protocols,* **15**(5), 1775–1799.

[174] Lazzarotto, C. R., Malinin, N. L., Li, Y., Zhang, R., Yang, Y., Lee, G., Cowley, E., He, Y., Lan, X., Jividen, K., et al. (2020) CHANGE-seq reveals genetic and epigenetic effects on CRISPR–Cas9 genome-wide activity. *Nature biotechnology,* **38**(11), 1317–1327.

[175] Kwon, J., Kim, M., Hwang, W., Jo, A., Hwang, G.-H., Jung, M., Kim, U. G., Cui, G., Kim, H., Eom, J.-H., et al. (2023) Extru-seq: a method for predicting genome-wide Cas9 off-target sites with advantages of both cell-based and in vitro approaches. *Genome biology,* **24**(1), 4.

[176] Zhu, M., Xu, R., Yuan, J., Wang, J., Ren, X., Cong, T., You, Y., Ju, A., Xu, L., Wang, H., et al. (2025) Tracking-seq reveals the heterogeneity of off-target effects in CRISPR–Cas9-mediated genome editing. *Nature Biotechnology,* **43**(5), 799–810.

[177] Kim, H. K., Song, M., Lee, J., Menon, A. V., Jung, S., Kang, Y.-M., Choi, J. W., Woo, E., Koh, H. C., Nam, J.-W., et al. (2017) In vivo high-throughput profiling of CRISPR–Cpf1 activity. *Nature methods,* **14**(2), 153–159.

[178] Sanjana, N. E., Shalem, O., and Zhang, F. (2014) Improved vectors and genome-wide libraries for CRISPR screening. *Nature methods,* **11**(8), 783–784.

[179] Hou, Y., Li, Y., Zheng, R., Zhang, F., Guo, F., Li, M., and Zeng, M. (2025) Leveraging protein language models for cross-variant CRISPR/Cas9 sgRNA activity prediction. *Bioinformatics,* p. btaf385.

[180] Fu, R., He, W., Dou, J., Villarreal, O. D., Bedford, E., Wang, H., Hou, C., Zhang, L., Wang, Y., Ma, D., et al. (2022) Systematic decomposition of sequence determinants governing CRISPR/Cas9 specificity. *Nature communications,* **13**(1), 474.

[181] Störtz, F., Mak, J. K., and Minary, P. (2023) piCRISPR: Physically informed deep learning models for CRISPR/Cas9 off-target cleavage prediction. *Artificial Intelligence in the Life Sciences,* **3**, 100075.

[182] Liu, Q., Cheng, X., Liu, G., Li, B., and Liu, X. (2020) Deep learning improves the ability of sgRNA off-target propensity prediction. *BMC bioinformatics,* **21**(1), 51.

[183] Xiang, X., Corsi, G. I., Anthon, C., Qu, K., Pan, X., Liang, X., Han, P., Dong, Z., Liu, L., Zhong, J., et al. (2021) Enhancing CRISPR-Cas9 gRNA efficiency prediction by data integration and deep learning. *Nature communications,* **12**(1), 3238.

[184] Zhang, Z., Lamson, A. R., Shelley, M., and Troyanskaya, O. (2023) Interpretable neural architecture search and transfer learning for understanding CRISPR–Cas9 off-target enzymatic reactions. *Nature Computational Science,* **3**(12), 1056–1066.

[185] Mak, J. K., Störtz, F., and Minary, P. (2022) Comprehensive computational analysis of epigenetic descriptors affecting CRISPR-Cas9 off-target activity. *BMC genomics,* **23**(1), 805.

[186] Sherkatghanad, Z., Abdar, M., Charlier, J., and Makarenkov, V. (2023) Using traditional machine learning and deep learning methods for on-and off-target prediction in CRISPR/Cas9: a review. *Briefings in Bioinformatics,* **24**(3), bbad131.

[187] Ham, D. T., Browne, T. S., Banglorewala, P. N., Wilson, T. L., Michael, R. K., Gloor, G. B., and Edgell, D. R. (2023) A generalizable Cas9/sgRNA prediction model using machine transfer learning with small high-quality datasets. *Nature Communications,* **14**(1), 5514.

[188] Yarrington, R. M., Verma, S., Schwartz, S., Trautman, J. K., and Carroll, D. (2018) Nucleosomes inhibit target cleavage by CRISPR-Cas9 in vivo. *Proceedings of the National Academy of Sciences,* **115**(38), 9351–9358.

[189] Nagamura, R., Kujirai, T., Kato, J., Shuto, Y., Kusakizako, T., Hirano, H., Endo, M., Toki, S., Saika, H., Kurumizaka, H., et al. (2024) Structural insights into how Cas9 targets nucleosomes. *Nature Communications,* **15**(1), 10744.

[190] Kim, S., Yu, N. K., and Kaang, B. K. (Jun, 2015) CTCF as a multifunctional protein in genome regulation and gene expression. *Exp Mol Med,* **47**, e166.

[191] Landt, S. G., Marinov, G. K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., Bernstein, B. E., Bickel, P., Brown, J. B., Cayting, P., Chen, Y., DeSalvo, G., Epstein, C., Fisher-Aylor, K. I., Euskirchen, G., Gerstein, M., Gertz, J., Hartemink, A. J., Hoffman, M. M., Iyer, V. R., Jung, Y. L., Karmakar, S., Kellis, M., Kharchenko, P. V., Li, Q., Liu, T., Liu, X. S., Ma, L., Milosavljevic, A., Myers, R. M., Park, P. J., Pazin, M. J., Perry, M. D., Raha, D., Reddy, T. E., Rozowsky, J., Shoresh, N., Sidow, A., Slattery, M., Stamatoyannopoulos, J. A., Tolstorukov, M. Y., White, K. P., Xi, S., Farnham, P. J., Lieb, J. D., Wold, B. J., and Snyder, M. (Sep, 2012) ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res,* **22**(9), 1813–1831.

[192] Liu, X., Wang, C., Liu, W., Li, J., Li, C., Kou, X., Chen, J., Zhao, Y., Gao, H., Wang, H., Zhang, Y., Gao, Y., and Gao, S. (09, 2016) Distinct features of H3K4me3 and H3K27me3 chromatin domains in pre-implantation embryos. *Nature,* **537**(7621), 558–562.

[193] Ginno, P. A., Lott, P. L., Christensen, H. C., Korf, I., and Chédin, F. (Mar, 2012) R-loop formation is a distinctive characteristic of unmethylated human CpG island promoters. *Mol Cell,* **45**(6), 814–825.

[194] Ginno, P. A., Lim, Y. W., Lott, P. L., Korf, I., and Chédin, F. (Oct, 2013) GC skew at the 5' and 3' ends of human genes links R-loop formation to epigenetic regulation and transcription termination. *Genome Res,* **23**(10), 1590–1600.

[195] Al-Hadid, Q. and Yang, Y. (Jul, 2016) R-loop: an emerging regulator of chromatin dynamics. *Acta Biochim Biophys Sin (Shanghai),* **48**(7), 623–631.

[196] Corsi, G. I., Qu, K., Alkan, F., Pan, X., Luo, Y., and Gorodkin, J. (2022) CRISPR/Cas9 gRNA activity depends on free energy changes and on the target PAM context. *Nature Communications,* **13**(1), 3006.

[197] Jones, S. K., Hawkins, J. A., Johnson, N. V., Jung, C., Hu, K., Rybarski, J. R., Chen, J. S., Doudna, J. A., Press, W. H., and Finkelstein, I. J. (2021) Massively parallel kinetic profiling of natural and engineered CRISPR nucleases. *Biophysical Journal,* **120**(3), 138a.

[198] Eslami-Mossallam, B., Klein, M., Smagt, C. V., Sanden, K. V., Jones, S. K., Hawkins, J. A., Finkelstein, I. J., and Depken, M. (2022) A kinetic model predicts SpCas9 activity, improves off-target classification, and reveals the physical basis of targeting fidelity. *Nature communications,* **13**(1), 1–10.

[199] Neshich, G., Borro, L. C., Higa, R. H., Kuser, P. R., Yamagishi, M. E., Franco, E. H., Krauchenco, J. N., Fileto, R., Ribeiro, A. A., Bezerra, G. B., Velludo, T. M., Jimenez, T. S., Furukawa, N., Teshima, H., Kitajima, K., Bava, A., Sarai, A., Togawa, R. C., and Mancini, A. L. (Jul, 2005) The Diamond STING server. *Nucleic Acids Res,* **33**(Web Server issue), 29–35.

[200] Neshich, G., Mancini, A. L., Yamagishi, M. E., Kuser, P. R., Fileto, R., Pinto, I. P., Palandrani, J. F., Krauchenco, J. N., Baudet, C., Montagner, A. J., and Higa, R. H. (Jan, 2005) STING Report: convenient web-based application for graphic and tabular presentations of protein sequence, structure and function descriptors from the STING database. *Nucleic Acids Res,* **33**(Database issue), D269–274.

[201] Mancini, A. L., Higa, R. H., Oliveira, A., Dominiquini, F., Kuser, P. R., Yamagishi, M. E., Togawa, R. C., and Neshich, G. (Sep, 2004) STING Contacts: a web-based application for identification and analysis of amino acid contacts within protein structure and across protein interfaces. *Bioinformatics,* **20**(13), 2145–2147.

[202] Neshich, G., Mazoni, I., Oliveira, S. R., Yamagishi, M. E., Kuser-Falcão, P. R., Borro, L. C., Morita, D. U., Souza, K. R., Almeida, G. V., Rodrigues, D. N., Jardine, J. G., Togawa, R. C., Mancini, A. L., Higa, R. H., Cruz, S. A., Vieira, F. D., Santos, E. H., Melo, R. C., and Santoro, M. M. (Dec, 2006) The Star STING server: a multiplatform environment for protein structure analysis. *Genet Mol Res,* **5**(4), 717–722.

[203] Higa, R. H., Montagner, A. J., Togawa, R. C., Kuser, P. R., Yamagishi, M. E. B., Mancini, A. L., Pappas, G., J., Miura, R. T., Horita, L. G., and Neshich, G. (03, 2004) ConSSeq: a web-based application for analysis of amino acid conservation based on HSSP database and within context of structure. *Bioinformatics,* **20**(12), 1983–1985.

[204] Neshich, G., Pena Neshich, I. A., Moraes, F., Salim, J. A., Borro, L., Yano, I. H., Mazoni, I., Jardine, J. G., and Rocchia, W. Using Structural and Physical–Chemical Parameters to Identify, Classify, and Predict Functional Districts in Proteins—The Role of Electrostatic Potential pp. 227–254 Springer International Publishing Cham (2015).

[205] Zenil, H., Hernández-Orozco, S., Kiani, N. A., Soler-Toscano, F., and Rueda-Toicen, A. A Decomposition Method for Global Evaluation of Shannon Entropy and Local Estimations of Algorithmic Complexity. (2016).

[206] Zenil, H. and Minary, P. (09, 2019) Training-free measures based on algorithmic probability identify high nucleosome occupancy in DNA sequences. *Nucleic Acids Research,* **47**(20), e129–e129.

[207] Tillo, D. and Hughes, T. R. (December, 2009) G+C content dominates intrinsic nucleosome occupancy. *BMC Bioinformatics,* **10**(1).

[208] Cui, F. and Zhurkin, V. B. (June, 2010) Structure-based Analysis of DNA Sequence Patterns Guiding Nucleosome Positioningin vitro. *Journal of Biomolecular Structure and Dynamics,* **27**(6), 821–841.

[209] Alharbi, B. A., Alshammari, T. H., Felton, N. L., Zhurkin, V. B., and Cui, F. (October, 2014) nuMap: A Web Platform for Accurate Prediction of Nucleosome Positioning. *Genomics, Proteomics & Bioinformatics,* **12**(5), 249–253.

[210] Xi, L., Fondufe-Mittendorf, Y., Xia, L., Flatow, J., Widom, J., and Wang, J.-P. (2010) Predicting nucleosome positioning using a duration Hidden Markov Model. *BMC Bioinformatics,* **11**(1), 346.

[211] Kato, H., Shimizu, M., and Urano, T. (2019) Chemical map–based prediction of nucleosome positioning using the Bioconductor package nuCpos. *bioRxiv,*.

[212] van der Heijden, T., van Vugt, J. J., Logie, C., and van Noort, J. (2012) Sequence-based prediction of single nucleosome positioning and genome-wide nucleosome occupancy. *Proceedings of the National Academy of Sciences,* **109**(38), E2514–E2522.

[213] Zhang, J., Peng, W., and Wang, L. (January, 2018) LeNup: learning nucleosome positioning from DNA sequences with improved convolutional neural networks. *Bioinformatics,* **34**(10), 1705–1712.

[214] Shtumpf, M., Piroeva, K. V., Agrawal, S. P., Jacob, D. R., and Teif, V. B. (Jan, 2022) NucPosDB: a database of nucleosome positioning in vivo and nucleosomics of cell-free DNA. *Chromosoma,*.

[215] Zhang, G., Dai, Z., and Dai, X. (2020) C-RNNCrispr: Prediction of CRISPR/Cas9 sgRNA activity using convolutional and recurrent neural networks. *Computational and Structural Biotechnology Journal,* **18**, 344 – 354.

[216] Cofsky, J. C., Soczek, K. M., Knott, G. J., Nogales, E., and Doudna, J. A. (2022) CRISPR–Cas9 bends and twists DNA to read its sequence. *Nature Structural & Molecular Biology,* **29**(4), 395–402.

[217] Vinogradov, A. E. (Apr, 2003) DNA helix: the importance of being GC-rich. *Nucleic Acids Res,* **31**(7), 1838–1844.

[218] Tennakoon, C., Purbojati, R. W., and Sung, W. K. (Aug, 2012) BatMis: a fast algorithm for k-mismatch mapping. *Bioinformatics,* **28**, 2122–2128.

[219] Kfir, N., Lev-Maor, G., Glaich, O., Alajem, A., Datta, A., Sze, S. K., Meshorer, E., and Ast, G. (Apr, 2015) SF3B1 association with chromatin determines splicing outcomes. *Cell Rep,* **11**(4), 618–629.

[220] Schwartz, U., Németh, A., Diermeier, S., Exler, J. H., Hansch, S., Maldonado, R., Heizinger, L., Merkl, R., and Längst, G. (11, 2018) Characterizing the nuclease accessibility of DNA in human cells to map higher order structures of chromatin. *Nucleic Acids Research,* **47**(3), 1239–1254.

[221] Mieczkowski, J., Cook, A., Bowman, S. K., Mueller, B., Alver, B. H., Kundu, S., Deaton, A. M., Urban, J. A., Larschan, E., Park, P. J., Kingston, R. E., and Tolstorukov, M. Y. (05, 2016) MNase titration reveals differences between nucleosome occupancy and chromatin accessibility. *Nat Commun,* **7**, 11485.

[222] Kundaje, A., Kyriazopoulou-Panagiotopoulou, S., Libbrecht, M., Smith, C. L., Raha, D., Winters, E. E., Johnson, S. M., Snyder, M., Batzoglou, S., and Sidow, A. (Sep, 2012) Ubiquitous heterogeneity and asymmetry of the chromatin environment at regulatory elements. *Genome Res,* **22**(9), 1735–1747.

[223] Devaiah, B. N., Case-Borden, C., Gegonne, A., Hsu, C. H., Chen, Q., Meerzaman, D., Dey, A., Ozato, K., and Singer, D. S. (06, 2016) BRD4 is a histone acetyltransferase that evicts nucleosomes from chromatin. *Nat Struct Mol Biol,* **23**(6), 540–548.

[224] Liu, H., Zhang, R., Xiong, W., Guan, J., Zhuang, Z., and Zhou, S. (09, 2013) A comparative evaluation on prediction methods of nucleosome positioning. *Briefings in Bioinformatics,* **15**(6), 1014–1027.

[225] Kaplan, N., Moore, I. K., Fondufe-Mittendorf, Y., Gossett, A. J., Tillo, D., Field, Y., LeProust, E. M., Hughes, T. R., Lieb, J. D., Widom, J., and Segal, E. (December, 2008) The DNA-encoded nucleosome organization of a eukaryotic genome. *Nature,* **458**(7236), 362–366.

[226] Chereji, R. V., Ramachandran, S., Bryson, T. D., and Henikoff, S. (February, 2018) Precise genome-wide mapping of single nucleosomes and linkers in vivo. *Genome Biology,* **19**(1).

[227] Van Rossum, G. and Drake, F. L. (2009) Python 3 Reference Manual, CreateSpace, Scotts Valley, CA.

[228] Burshtein, D. (1996) Robust parametric modeling of durations in hidden Markov models. *IEEE Transactions on Speech and Audio Processing,* **4**(3), 240–242.

[229] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014) Going Deeper with Convolutions. *CoRR,* **abs/1409.4842**.

[230] Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2016) Language Modeling with Gated Convolutional Networks. *CoRR,* **abs/1612.08083**.

[231] Box, G. E. and Cox, D. R. (1964) An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological),* **26**(2), 211–243.

[232] Wang, J., Xiang, X., Bolund, L., Zhang, X., Cheng, L., and Luo, Y. (01, 2020) GNL-Scorer: a generalized model for predicting CRISPR on-target activity by machine learning and featurization. *Journal of Molecular Cell Biology,* **12**(11), 909–911.

[233] Bradford, J. and Perrin, D. (08, 2019) A benchmark of computational CRISPR-Cas9 guide design methods. *PLoS Comput Biol,* **15**(8), e1007274.

[234] Charlier, J., Nadon, R., and Makarenkov, V. (02, 2021) Accurate deep learning off-target prediction with novel sgRNA-DNA sequence encoding in CRISPR-Cas9 gene editing. *Bioinformatics,* btab112.

[235] Pacesa, M., Lin, C.-H., Cléry, A., Saha, A., Arantes, P. R., Bargsten, K., Irby, M. J., Allain, F. H.-T., Palermo, G., Cameron, P., et al. (2022) Structural basis for Cas9 off-target activity. *Cell,* **185**(22), 4067–4081.

[236] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013) API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* pp. 108–122.

[237] Lindahl, Abraham, Hess, and van der Spoel GROMACS 2020 Manual. (January, 2020).

[238] Reis, P. B. P. S., Vila-Viçosa, D., Rocchia, W., and Machuqueiro, M. (aug, 2020) PypKa: A Flexible Python Module for Poisson–Boltzmann-Based pK_a Calculations. *Journal of Chemical Information and Modeling,* **60**(10), 4442–4448.

[239] Zhu, X., Clarke, R., Puppala, A. K., Chittori, S., Merk, A., Merrill, B. J., Simonović, M., and Subramaniam, S. (jul, 2019) Cryo-EM structures reveal coordinated domain motions that govern DNA cleavage by Cas9. *Nature Structural & Molecular Biology,* **26**(8), 679–685.

[240] Bonomi, M., Branduardi, D., Bussi, G., Camilloni, C., Provasi, D., Raiteri, P., Donadio, D., Marinelli, F., Pietrucci, F., Broglia, R. A., and Parrinello, M. (oct, 2009) PLUMED: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications,* **180**(10), 1961–1972.

[241] Tribello, G. A., Bonomi, M., Branduardi, D., Camilloni, C., and Bussi, G. (feb, 2014) PLUMED 2: New feathers for an old bird. *Computer Physics Communications,* **185**(2), 604–613.

[242] The PLUMED consortium (jul, 2019) Promoting transparency and reproducibility in enhanced molecular simulations. *Nature Methods,* **16**(8), 670–673.

[243] Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., and Ferrin, T. E. (2004) UCSF Chimera?A visualization system for exploratory research and analysis. *Journal of Computational Chemistry,* **25**(13), 1605–1612.

[244] Rocchia, W., Alexov, E., and Honig, B. (jun, 2001) Extending the Applicability of the Nonlinear Poisson-Boltzmann Equation: Multiple Dielectric Constants and Multivalent Ions. *The Journal of Physical Chemistry B,* **105**(28), 6754–6754.

[245] Tian, C., Kasavajhala, K., Belfon, K. A. A., Raguette, L., Huang, H., Migues, A. N., Bickel, J., Wang, Y., Pincay, J., Wu, Q., and Simmerling, C. (nov, 2019) ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *Journal of Chemical Theory and Computation,* **16**(1), 528–552.

[246] Gowers, R. J., Linke, M., Barnoud, J., Reddy, T. J. E., Melo, M. N., Seyler, S. L., Domański, J., Dotson, D. L., Buchoux, S., Kenney, I. M., and Beckstein, O. (2016) MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. In Sebastian Benthall and Scott Rostrup, (eds.), *Proceedings of the 15th Python in Science Conference*, pp. 98 – 105.

[247] Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., et al. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics,* **25**(11), 1422–1423.

[248] de Moraes, F. R., Neshich, I. A. P., Mazoni, I., Yano, I. H., Pereira, J. G. C., Salim, J. A., Jardine, J. G., and Neshich, G. (01, 2014) Improving Predictions of Protein-Protein Interfaces by Combining Amino Acid-Specific Classifiers Based on Structural and Physicochemical Descriptors with Their Weighted Neighbor Averages. *PLOS ONE,* **9**(1), 1–15.

[249] Mazoni, I., Borro, L. C., Jardine, J. G., Yano, I. H., Salim, J. A., and Neshich, G. (2018) Study of specific nanoenvironments containing α-helices in all-α and (α+ β)+(α/β) proteins. *PloS one,* **13**(7), e0200018.

[250] Mazoni, I., Salim, J. A., de Moraes, F. R., Borro, L., and Neshich, G. (2020) A comparison between internal protein nanoenvironments of α-helices and β-sheets. *PLoS One,* **15**(12), e0244315.

[251] Bellman, R. (1966) Dynamic programming. *Science,* **153**(3731), 34–37.

[252] Geurts, P., Ernst, D., and Wehenkel, L. (2006) Extremely randomized trees. *Machine learning,* **63**, 3–42.

[253] Hua, J., Xiong, Z., Lowey, J., Suh, E., and Dougherty, E. R. (2005) Optimal number of features as a function of sample size for various classification rules. *Bioinformatics,* **21**(8), 1509–1515.

[254] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020) From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence,* **2**(1), 2522–5839.

[255] Schrödinger, LLC (November, 2015) The PyMOL Molecular Graphics System, Version 1.8. PyMOL.

[256] Horton, N. C. and Finzel, B. C. (1996) The structure of an RNA/DNA hybrid: a substrate of the ribonuclease activity of HIV-1 reverse transcriptase. *Journal of molecular biology,* **264**(3), 521–533.

[257] Liu, Q., He, D., and Xie, L. (10, 2019) Prediction of off-target specificity and cell-specific fitness of CRISPR-Cas System using attention boosted deep learning and network-based gene feature. *PLOS Computational Biology,* **15**(10), 1–22.

[258] Chen, Q., Chuai, G., Zhang, H., Tang, J., Duan, L., Guan, H., Li, W., Li, W., Wen, J., Zuo, E., et al. (2023) Genome-wide CRISPR off-target prediction and optimization using RNA-DNA interaction fingerprints. *Nature Communications,* **14**(1), 7521.

[259] Zuo, Z., Zolekar, A., Babu, K., Lin, V. J., Hayatshahi, H. S., Rajan, R., Wang, Y.-C., and Liu, J. (jul, 2019) Structural and functional insights into the *bona fide* catalytic state of *Streptococcus pyogenes* Cas9 HNH nuclease domain. *eLife,* **8**, e46500.

[260] Chen, J. S. and Doudna, J. A. (October, 2017) The chemistry of Cas9 and its CRISPR colleagues. *Nature Reviews Chemistry,* **1**(10), 0078.

[261] Huai, C., Li, G., Yao, R., Zhang, Y., Cao, M., Kong, L., Jia, C., Yuan, H., Chen, H., Lu, D., et al. (2017) Structural insights into DNA cleavage activation of CRISPR-Cas9 system. *Nature Communications,* **8**(1), 1375.

[262] Palermo, G. (February, 2019) Structure and Dynamics of the CRISPR–Cas9 Catalytic Complex. *Journal of Chemical Information and Modeling,* **59**(5), 2394–2406.

[263] Ricci, C. G., Chen, J. S., Miao, Y., Jinek, M., Doudna, J. A., McCammon, J. A., and Palermo, G. (March, 2019) Deciphering Off-Target Effects in CRISPR-Cas9 through Accelerated Molecular Dynamics. *ACS Central Science,* **5**(4), 651–662.

[264] Jiang, F., Taylor, D. W., Chen, J. S., Kornfeld, J. E., Zhou, K., Thompson, A. J., Nogales, E., and Doudna, J. A. (feb, 2016) Structures of a CRISPR-Cas9 R-loop complex primed for DNA cleavage. *Science,* **351**(6275), 867–871.

[265] Venanzi, N. A. E., Basciu, A., Vargiu, A. V., Kiparissides, A., Dalby, P. A., and Dikicioglu, D. (2024) Machine learning integrating protein structure, sequence, and dynamics to predict the enzyme activity of bovine enterokinase variants. *Journal of Chemical Information and Modeling,* **64**(7), 2681–2694.

[266] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. (2024) Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature,* **630**(8016), 493–500.

[267] Mak, J. K., Bendandi, A., Salim, J. A., Mazoni, I., de Moraes, F. R., Borro, L., Störtz, F., Rocchia, W., Neshich, G., and Minary, P. (2025) Learning to utilize internal protein 3D nanoenvironment descriptors in predicting CRISPR–Cas9 off-target activity. *NAR Genomics and Bioinformatics,* **7**(2), lqaf054.

[268] Zhong, Z., Li, Z., Yang, J., and Wang, Q. (2023) Unified Model to Predict gRNA Efficiency across Diverse Cell Lines and CRISPR-Cas9 Systems. *Journal of Chemical Information and Modeling,* **63**(23), 7320–7329.

[269] Thean, D. G., Chu, H. Y., Fong, J. H., Chan, B. K., Zhou, P., Kwok, C. C., Chan, Y. M., Mak, S. Y., Choi, G. C., Ho, J. W., et al. (2022) Machine learning-coupled combinatorial mutagenesis enables resource-efficient engineering of CRISPR-Cas9 genome editor activities. *Nature Communications,* **13**(1), 2219.

[270] Georgiev, A. G. (2009) Interpretable numerical descriptors of amino acid space. *Journal of Computational Biology,* **16**(5), 703–723.

[271] Bepler, T. and Berger, B. (2019) Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661,*.

[272] Fang, T., Bogensperger, L., Feer, L., Allam, A., Bezshapkin, V., Balázs, Z., von Mering, C., Sunagawa, S., Krauthammer, M., and Schwank, G. (2025) Uncovering Cas9 PAM diversity through metagenomic mining and machine learning. *bioRxiv,*.

[273] EvolutionaryScale Team evolutionaryscale/esm. (2024).

[274] Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., Badkundri, R., Shafkat, I., Gong, J., Derry, A., Molina, R. S., Thomas, N., Khan, Y. A., Mishra, C., Kim, C., Bartie, L. J., Nemeth, M., Hsu, P. D., Sercu, T., Candido, S., and Rives, A. (2025) Simulating 500 million years of evolution with a language model. *Science,*.

[275] Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. (2022) ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **44**(10), 7112–7127.

[276] Elnaggar, A., Essam, H., Salah-Eldin, W., Moustafa, W., Elkerdawy, M., Rochereau, C., and Rost, B. (2023) Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling. *arXiv preprint arXiv:2301.06568,*.

[277] Yang, K. K., Alamdari, S., Lee, A. J., Kaymak-Loveless, K., Char, S., Brixi, G., Domingo-Enrich, C., Wang, C., Lyu, S., Fusi, N., et al. (2025) The Dayhoff Atlas: scaling sequence diversity for improved protein generation. *bioRxiv,* pp. 2025–07.

[278] ESM Team ESM Cambrian: Revealing the mysteries of proteins with unsupervised learning. (2024).

[279] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. (2023) Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science,* **379**(6637), 1123–1130.

[280] Yan, C., Zhang, Z., Xu, J., Meng, Y., Yan, S., Wei, L., Zou, Q., Zhang, Q., and Cui, F. (2025) CasPro-ESM2: Accurate identification of Cas proteins integrating pre-trained protein language model and multi-scale convolutional neural network. *International Journal of Biological Macromolecules,* **308**, 142309.

[281] Ruffolo, J. A., Nayfach, S., Gallagher, J., Bhatnagar, A., Beazer, J., Hussain, R., Russ, J., Yip, J., Hill, E., Pacesa, M., et al. (2025) Design of highly functional genome editors by modelling CRISPR–Cas sequences. *Nature,* pp. 1–8.

[282] Nguyen, E., Poli, M., Durrant, M. G., Kang, B., Katrekar, D., Li, D. B., Bartie, L. J., Thomas, A. W., King, S. H., Brixi, G., Sullivan, J., Ng, M. Y., Lewis, A., Lou, A., Ermon, S., Baccus, S. A., Hernandez-Boussard, T., Ré, C., Hsu, P. D., and Hie, B. L. (2024) Sequence modeling and design from molecular to genome scale with Evo. *Science,* **386**(6723), eado9336.

[283] Du, W., Zhao, L., Diao, K., Zheng, Y., Yang, Q., Zhu, Z., Zhu, X., and Tang, D. (2025) A versatile CRISPR/Cas9 system off-target prediction tool using language model. *Communications Biology,* **8**(1), 882.

[284] Chen, J., Hu, Z., Sun, S., Tan, Q., Wang, Y., Yu, Q., Zong, L., Hong, L., Xiao, J., Shen, T., King, I., and Li, Y. Interpretable RNA Foundation Model from Unannotated Data for Highly Accurate RNA Structure and Function Predictions. (2022).

[285] Ren, Y., Chen, Z., Qiao, L., Jing, H., Cai, Y., Xu, S., Ye, P., Ma, X., Sun, S., Yan, H., Yuan, D., Ouyang, W., and Liu, X. (2024) BEACON: Benchmark for Comprehensive RNA Tasks and Language Models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[286] Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011) ViennaRNA Package 2.0. *Algorithms for molecular biology,* **6**(1), 26.

[287] Penić, R. J., Vlašić, T., Huber, R. G., Wan, Y., and Šikić, M. (2025) Rinalmo: General-purpose rna language models can generalize well on structure prediction tasks. *Nature Communications,* **16**(1), 5671.

[288] Sun, W., Yang, J., Cheng, Z., Amrani, N., Liu, C., Wang, K., Ibraheim, R., Edraki, A., Huang, X., Wang, M., et al. (2019) Structures of Neisseria meningitidis Cas9 complexes in catalytically poised and anti-CRISPR-inhibited states. *Molecular cell,* **76**(6), 938–952.

[289] Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (03, 2007) Uni comprehensive and non-redundant UniProt reference clusters. *Bioinformatics,* **23**(10), 1282–1288.

[290] Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. D., and Higgins, D. G. (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology,* **7**(1), 539.

[291] Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J., and Lopez, R. (05, 2010) A new bioinformatics analysis tools framework at EMBL–EBI. *Nucleic Acids Research,* **38**(suppl_2), W695–W699.

[292] Cornman, A., West-Roberts, J., Camargo, A. P., Roux, S., Beracochea, M., Mirdita, M., Ovchinnikov, S., and Hwang, Y. (2025) The OMG dataset: An Open MetaGenomic corpus for mixed-modality genomic language modeling. In *The Thirteenth International Conference on Learning Representations*.

[293] Özden, F. and Minary, P. (09, 2024) Learning to quantify uncertainty in off-target activity for CRISPR guide RNAs. *Nucleic Acids Research,* **52**(18), e87–e87.

[294] 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., and Duchesnay, E. (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* **12**, 2825–2830.

[295] Vieira, L. C., Handojo, M. L., and Wilke, C. O. (2025) Medium-sized protein language models perform well at transfer learning on realistic datasets: LC Vieira et al.. *Scientific Reports,* **15**(1), 21400.

[296] Cao, X. and Minary, P. (2024) CRISPR-DBA: a deep learning framework for uncertainty quantification of CRISPR off-target activities. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* IEEE pp. 274–281.

[297] Schmitz, C., Bradford, J., Salomone, R., and Perrin, D. (07, 2025) Leveraging uncertainty quantification to optimise CRISPR guide RNA selection. *Biology Methods and Protocols,* p. bpaf054.

[298] Hora, S. C. (1996) Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability engineering & system safety,* **54**(2-3), 217–223.

[299] Wang, T., Wei, J. J., Sabatini, D. M., and Lander, E. S. (2014) Genetic screens in human cells using the CRISPR-Cas9 system. *Science,* **343**(6166), 80–84.

[300] Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. (2020) Problems with Shapley-value-based explanations as feature importance measures. In *International conference on machine learning* PMLR pp. 5491–5500.

[301] Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. (2021) Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods,* **18**(10), 1196–1203.

[302] Linder, J., Srivastava, D., Yuan, H., Agarwal, V., and Kelley, D. R. (2025) Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. *Nature Genetics,* **57**(4), 949–961.

[303] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013) Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing.*

[304] Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thåström, A., Field, Y., Moore, I. K., Wang, J.-P. Z., and Widom, J. (July, 2006) A genomic code for nucleosome positioning. *Nature,* **442**(7104), 772–778.

[305] Minary, P. and Levitt, M. (2014) Training-free atomistic prediction of nucleosome occupancy. *Proceedings of the National Academy of Sciences,* **111**(17), 6293–6298.

[306] Satchwell, S. C., Drew, H. R., and Travers, A. A. (1986) Sequence periodicities in chicken nucleosome core DNA. *Journal of Molecular Biology,* **191**(4), 659 – 675.

[307] Vanderlick, T. K., Scriven, L. E., and Davis, H. T. (Dec, 1986) Solution of Percus's equation for the density of hard rods in an external field. *Phys Rev A Gen Phys,* **34**(6), 5130–5131.

[308] Levenberg, K. (1944) A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics,* **2**(2), 164–168.

[309] Delahaye, J.-P. and Zenil, H. (2012) Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of randomness. *Applied Mathematics and Computation,* **219**(1), 63–77 Towards a Computational Interpretation of Physical Theories.

[310] Soler-Toscano, F., Zenil, H., Delahaye, J.-P., and Gauvrit, N. (05, 2014) Calculating Kolmogorov Complexity from the Output Frequency Distributions of Small Turing Machines. *PLOS ONE,* **9**(5), 1–18.

[311] Kolmogorov, A. N. (1998) On Tables of Random Numbers (Reprinted from "Sankhya: The Indian Journal of Statistics", Series A, Vol. 25 Part 4, 1963).. *Theor. Comput. Sci.,* **207**(2), 387–395.

[312] Solomonoff, R. (1964) A formal theory of inductive inference. Parts I and II. *Information and Control,* **7**(1), 1–22 and 224–254.

[313] Tsodikov, O. V., Record Jr, M. T., and Sergeev, Y. V. (2002) Novel computer program for fast exact calculation of accessible and molecular surface areas and average surface curvature. *Journal of computational chemistry,* **23**(6), 600–609.

[314] Honig, B. and Nicholls, A. (1995) Classical Electrostatics in Biology and Chemistry. *Science,* **268**(5214), 1144–1149.

[315] Rocchia, W. and Neshich, G. (Oct, 2007) Electrostatic potential calculation for biomolecules–creating a database of pre-calculated values reported on a per residue basis for all PDB protein structures. *Genet Mol Res,* **6**(4), 923–936.

[316] Sander, C. and Schneider, R. (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics,* **9**(1), 56–68.

[317] Brinda, K. and Vishveshwara, S. (2005) A Network Representation of Protein Structures: Implications for Protein Stability. *Biophysical Journal,* **89**(6), 4159–4170.

[318] Dokholyan, N. V., Li, L., Ding, F., and Shakhnovich, E. I. (2002) Topological determinants of protein folding. *Proceedings of the National Academy of Sciences,* **99**(13), 8637–8641.

[319] Greene, L. H. and Higman, V. A. (2003) Uncovering Network Systems Within Protein Structures. *Journal of Molecular Biology,* **334**(4), 781–791.

[320] Vendruscolo, M., Dokholyan, N. V., Paci, E., and Karplus, M. (Jun, 2002) Small-world view of the amino acids that play a key role in protein folding. *Phys. Rev. E,* **65**, 061910.

[321] Agnarsson, G. and Greenlaw, R. (2006) Graph Theory: Modeling, Applications, and Algorithms, Prentice-Hall, Inc., USA.

[322] Oehlers, M. and Fabian, B. (2021) Graph Metrics for Network Robustness—A Survey. *Mathematics,* **9**(8), 895.

[323] Newman, M. J. (2005) A measure of betweenness centrality based on random walks. *Social Networks,* **27**(1), 39–54.

[324] Radzicka, A. and Wolfenden, R. (1988) Comparing the polarities of the amino acids: side-chain distribution coefficients between the vapor phase, cyclohexane, 1-octanol, and neutral aqueous solution. *Biochemistry,* **27**(5), 1664–1670.

[325] Kyte, J. and Doolittle, R. F. (1982) A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology,* **157**(1), 105–132.

[326] Kabsch, W. and Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules,* **22**(12), 2577–2637.

[327] Frishman, D. and Argos, P. (1995) Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Bioinformatics,* **23**(4), 566–579.

[328] Chien, Y.-T. and Huang, S.-W. (10, 2012) Accurate Prediction of Protein Catalytic Residues by Side Chain Orientation and Residue Contact Density. *PLOS ONE,* **7**(10), 1–11.

[329] Ooi, T., Oobatake, M., Némethy, G., and Scheraga, H. A. (1987) Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides.. *Proceedings of the National Academy of Sciences,* **84**(10), 3086–3090.

[330] Porollo, A. and Meller, J. (2007) Prediction-based fingerprints of protein–protein interactions. *Proteins: Structure, Function, and Bioinformatics,* **66**(3), 630–645.

Appendix A

Supplementary materials for "Comprehensive computational analysis of epigenetic descriptors affecting CRISPR-Cas9 off-target activity"

A.1 Appendix tables and figures

Extending Figure 2.1, Appendix Table A.1 lists the off-target cleavage activity Spearman and Pearson correlation values for all experimental epigenetic and computed nucleosome organization-related features. Figure A.1 shows the architecture of the convolutional neural network used for CRISPR-Cas9 off-target activity prediction and SHAP value analysis. Extending Figure 2.1, Appendix Figures A.2, A.3 and A.4 show the cell line-based heatmaps indicating Spearman and Pearson correlations between the epigenetic features and CRISPR-Cas9 off-target activity for cell lines HeLa, K562 and U2OS, respectively. Extending Figure 2.2, Appendix Figures A.5 and A.6, respectively, show the violin and distribution plots for CRISPR-Cas9 off-target cleavage activity for all 19 experimental epigenetic features and computed nucleosome organization-related, with the experimental epigenetic features highlighted in bold. Extending Figures 2.3 and 2.4, Appendix Figures A.7 and A.8 visualize the SHAP contribution of each input feature in a trained XGBoost and CNN model, respectively, both of which predict CRISPR-Cas9 off-target activity, where all computed nucleosome organization-related scores are base pair-resolved.

Using only the 'on-target' datapoints that correspond to guide-RNA–on-target DNA sequence pairs, Appendix Figures A.10 shows an overall correlation analysis. It can be seen that Nucleotide (and Strong-Weak) BDM still show the highest Spearman correlation with on-target cleavage activity, even though the difference in correlation values is not as pronounced as found for the off-target cleavage activity dataset.

Epigenetic Feature	**Spearman**	**Pearson**
Strong-Weak BDM	**0.423**	0.310
Nucleotide BDM	0.388	**0.345**
GC147	0.191	0.117
NuPoP (Occupancy)	0.167	0.068
YR Scheme	0.087	0.108
MNase	0.082	0.083
NuPoP (Viterbi)	0.078	0.060
H3K4me3	0.075	0.066
CTCF	0.070	0.059
DNase I	0.065	0.033
NuPoP (Affinity)	0.048	0.011
nuCpos (Occupancy)	0.040	0.015
RRBS	0.022	0.009
nuCpos (Viterbi)	0.014	0.015
VanDerHeijden	0.009	-0.036
nuCpos (Affinity)	0.005	0.037
LeNup (H3Q85C)	-0.050	-0.043
DRIP	-0.059	0.076
W/S Scheme	-0.141	-0.122

Table A.1: Spearman and Pearson correlation values between epigenetic features and SpCas9 off-target cleavage activities. Epigenetic features include all nucleosome organization-related scores and six experimental epigenetic scores (bolded). The correlations were derived using the said scores and cleavage activities for all datapoints defined in the Materials and Methods section. The features are sorted by decreasing Spearman correlations and the highest Spearman and Pearson correlation values are highlighted in bold.

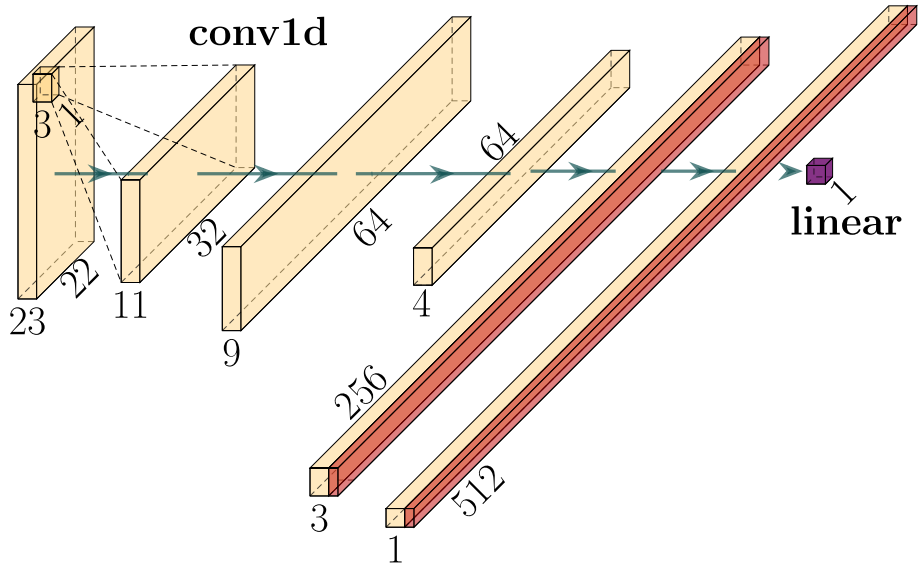


Figure A.1: Convolutional neural network architecture used for CRISPR-Cas9 off-target activity prediction as mentioned in the Methods section. The architecture is implemented in PyTorch [141]. The input to the neural network is a 23bp $\times$ 22 features input matrix, and the output is a scalar value indicative of the CRISPR-Cas9 off-target activity prediction. The architecture consists of five one-dimensional convolutional (Conv1D) layers followed by one fully connected layer. The first layer is a Conv1D layer with 32 channels, 3×3 kernel size, stride of 2 and padding of 0, followed by leaky rectified linear unit activation (LeakyReLU) [303] with a negative slope of 0.2. The second layer is a Conv1D layer with 64 channels, 3×3 kernel size, stride of 1 and padding of 0, followed by LeakyReLU [303] with a negative slope of 0.2. The third layer is a Conv1D layer with 128 channels, 3×3 kernel size, stride of 2 and padding of 0, followed by 1D batch normalization and subsequently LeakyReLU [303] with a negative slope of 0.2. The fourth layer is a Conv1D layer with 256 channels, 3×3 kernel size, stride of 1 and padding of 0, followed by 1D 3×3 max pooling with padding of 1 and stride of 1, and subsequently rectified linear unit activation (ReLU). The fifth layer is a Conv1D layer with 512 channels, 2×2 kernel size, stride of 1 and padding of 0, followed by 1D 3×3 max pooling with padding of 1 and stride of 1, and subsequently rectified linear unit activation (ReLU). The final layer is a fully connected layer which outputs a scalar value.

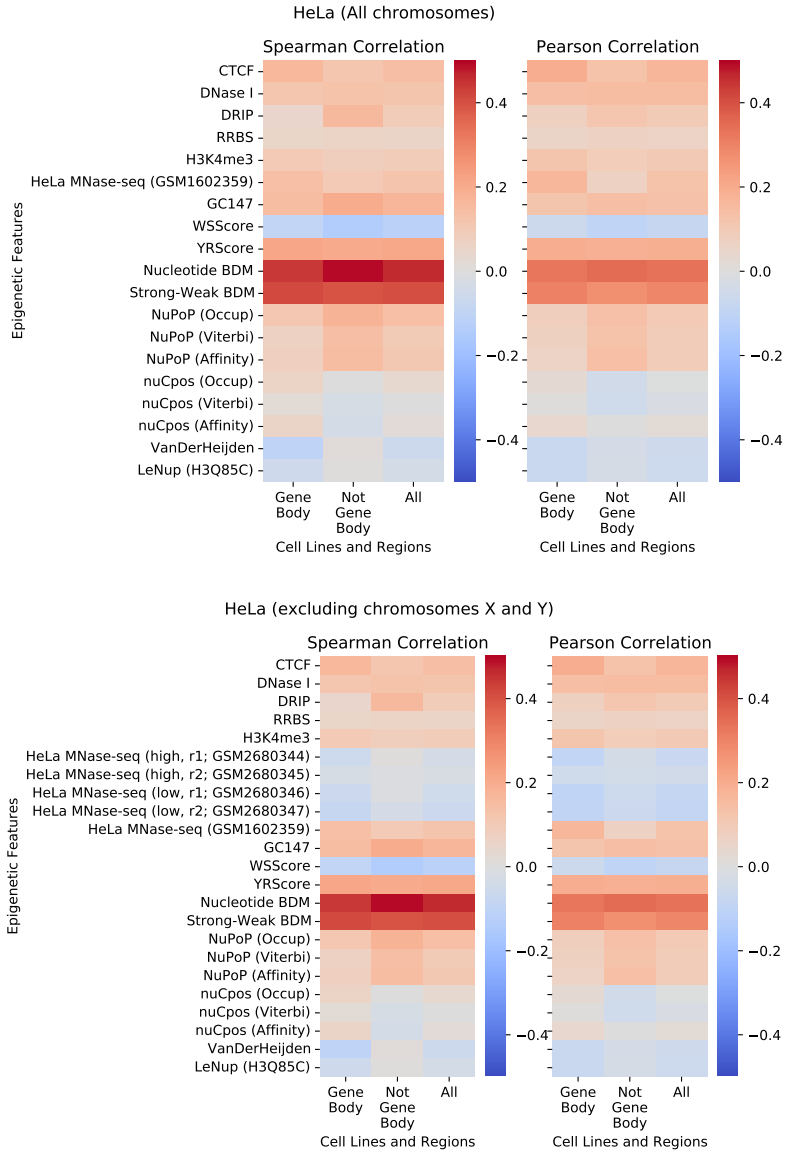


Figure A.2: (Top) Heatmaps showing Spearman (top left) and Pearson (top right) correlations between SpCas9 off-target cleavage activities and 19 epigenetic features, namely 13 computed nucleosome organization-related scores, 5 experimental epigenetic scores (bolded) and one HeLa MNase-seq score for HeLa-only nucleosome organization-related score-augmented off-target cleavage activity data. (Bottom) Heatmaps showing Spearman (bottom left) and Pearson (bottom right) correlations between SpCas9 off-target cleavage activities and 23 epigenetic features, namely 13 computed nucleosome organization-related scores, 5 experimental epigenetic scores (bolded) and 5 HeLa MNase-seq scores for HeLa-only nucleosome organization-related score-augmented off-target cleavage activity data in chromosomes 1-22.

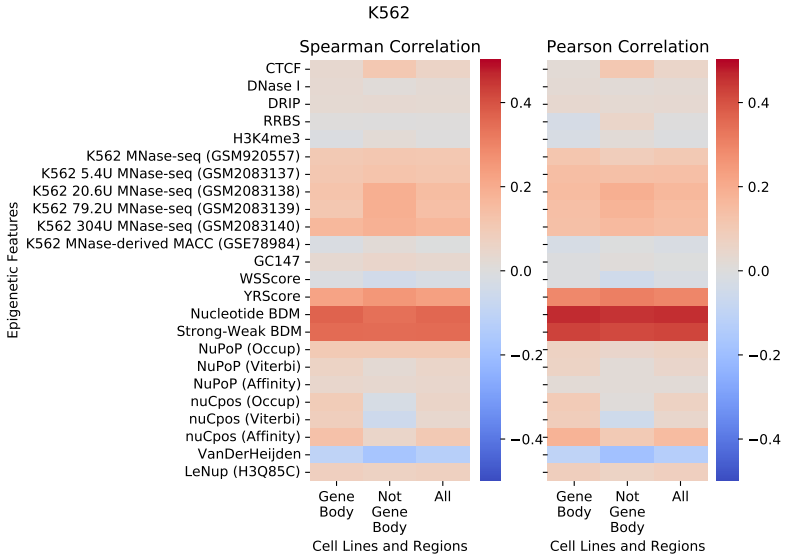


Figure A.3: Heatmaps showing Spearman (left) and Pearson (right) correlations between SpCas9 off-target cleavage activities and 24 epigenetic features, namely 13 computed nucleosome organization-related scores, 5 experimental epigenetic scores (bolded) and 6 K562 MNase-seq score for K562-only nucleosome organization-related score-augmented off-target cleavage activity data.

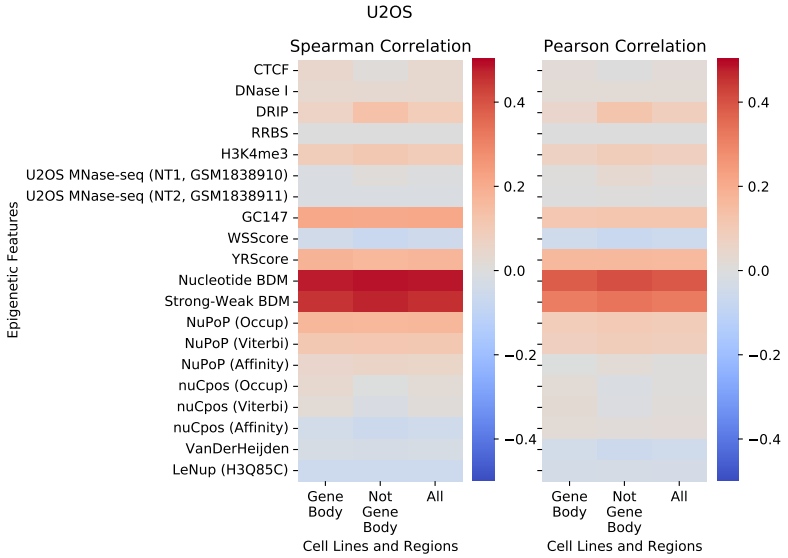


Figure A.4: Heatmaps showing Spearman (left) and Pearson (right) correlations between SpCas9 off-target cleavage activities and 20 epigenetic features, namely 13 computed nucleosome organization-related scores, 5 experimental epigenetic scores (bolded) and 2 U2OS MNase-seq score for U2OS-only nucleosome organization-related score-augmented off-target cleavage activity data.

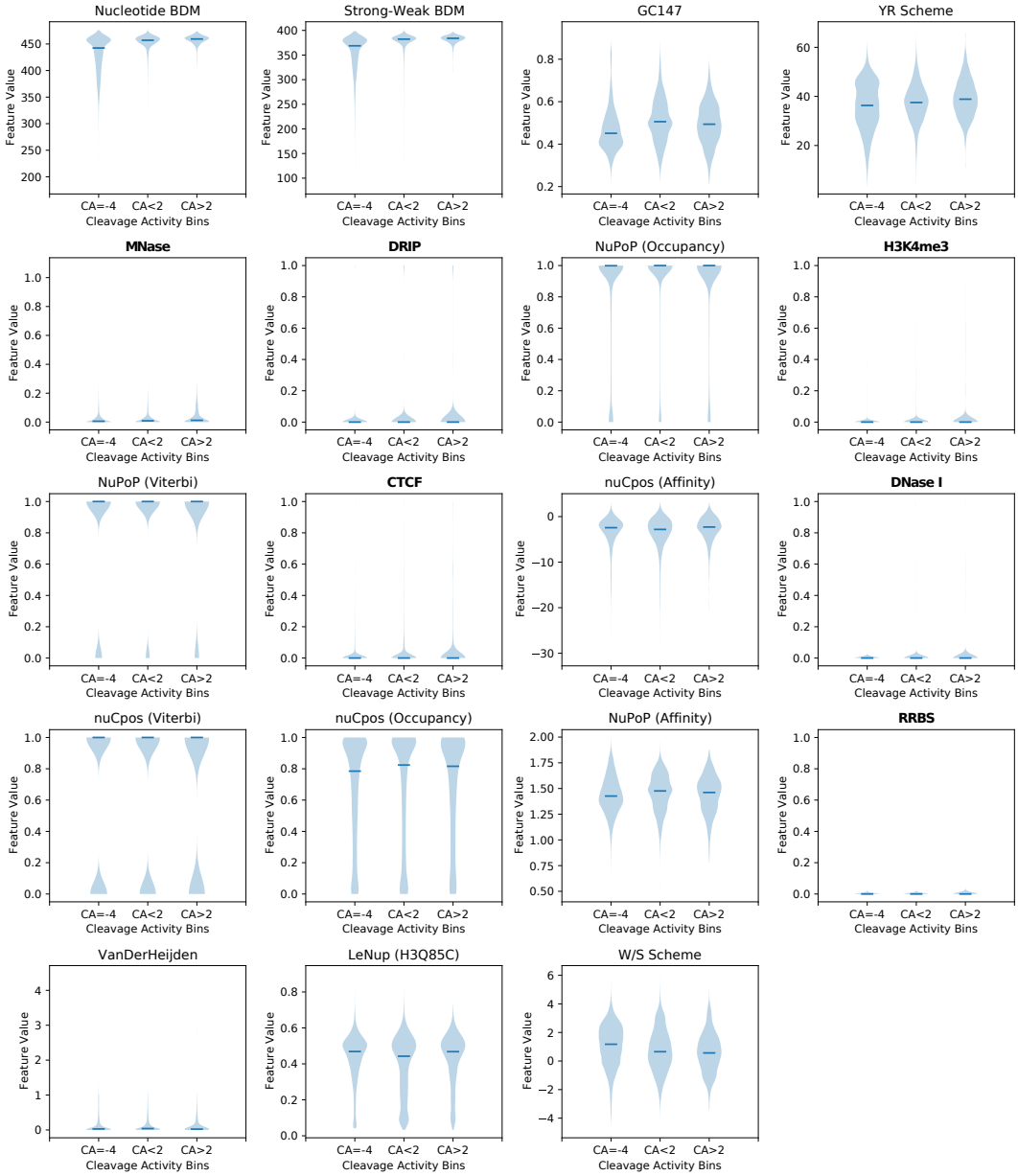


Figure A.5: Violin plots for all nucleosome organization-related features, with the features sorted by decreasing Pearson correlation with CRISPR-Cas9 activity values and the experimental epigenetic features CTCF, DNase I, DRIP, H3K4me3, MNase and RRBS highlighted in bold. Cleavage activities (CA) are separated into three bins, namely CA $= -4$, CA < 2 and CA > 2.

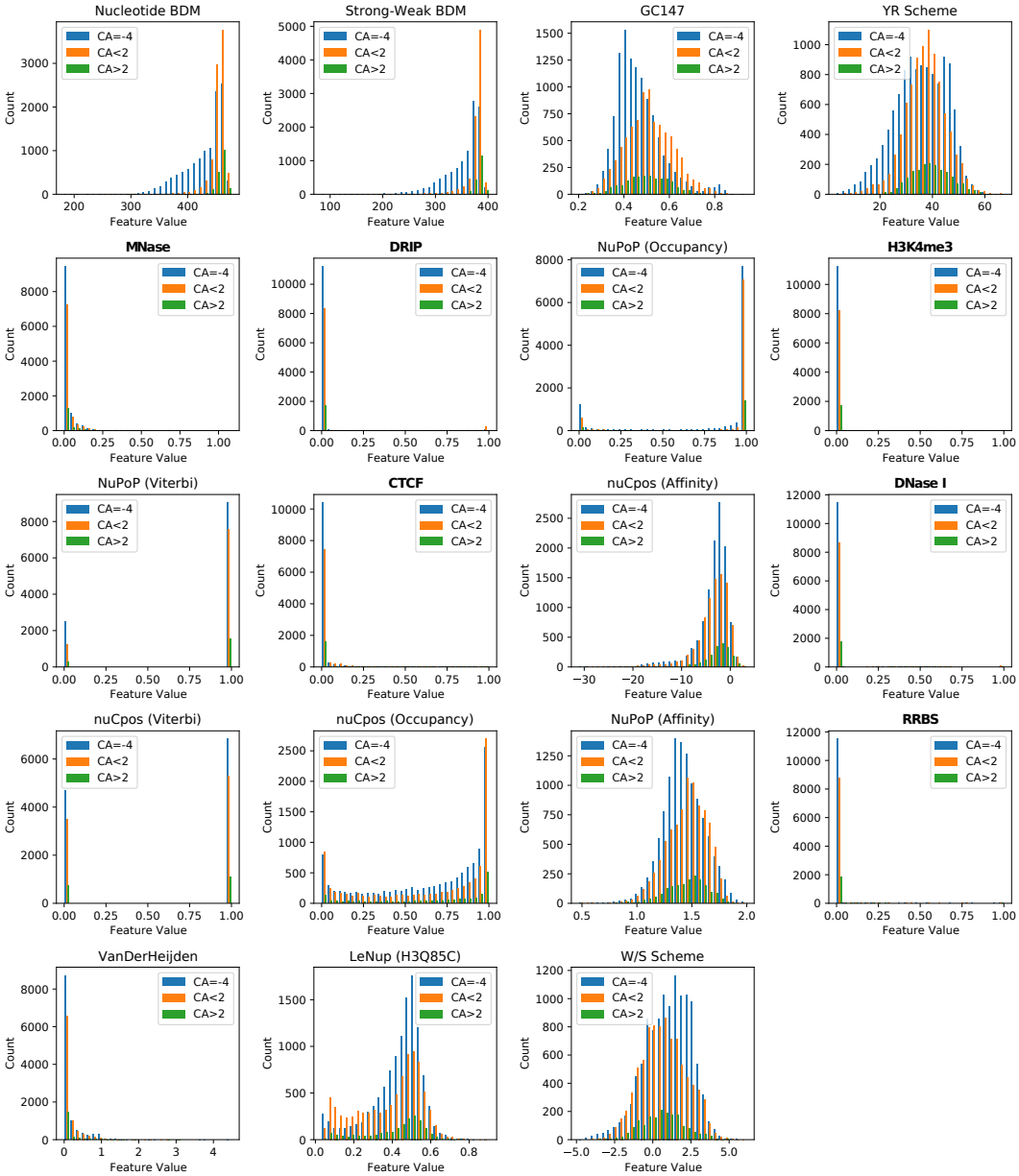


Figure A.6: Distribution plots for all nucleosome organization-related features, with the features sorted decreasing Spearman correlation with CRISPR-Cas9 activity values and the experimental epigenetic features CTCF, DNase I, DRIP, H3K4me3, MNase and RRBS highlighted in bold. Cleavage activities (CA) are separated into three bins, namely CA $= -4$, CA < 2 and CA > 2, which are colored blue, orange and green, respectively, in the plots.

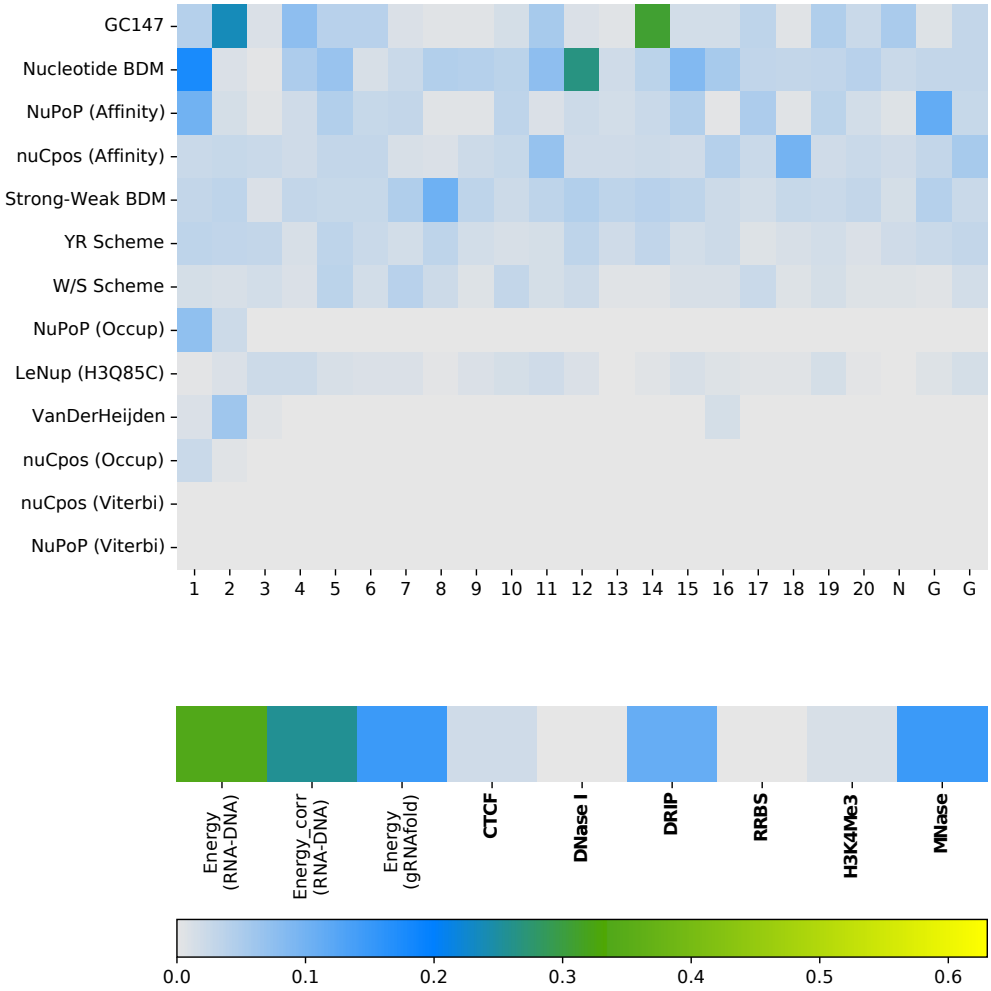


Figure A.7: Heatmap showing the mean absolute value of the SHAP values for the trained extreme gradient boosted (XGBoost) tree's base pair-resolved input features, which consist of the three CRISPRspec-derived energy terms $E_{\text{RNA-DNA}}$, $E^{\text{corr}}_{\text{RNA-DNA}}$ and E_{gRNAfold}, the six experimental epigenetic scores CTCF, DNase I, DRIP, RRBS, H3K4me3 and MNase (bolded), and the computed nucleosome organization-related scores GC147 [207], W/S scheme, YR scheme [208, 209], Strong-Weak BDM, Nucleotide BDM [205, 206], NuPoP (Occupancy), NuPoP (Affinity), NuPoP (Viterbi) [210], nuCpos (Occupancy), nuCpos (Affinity), nuCpos (Viterbi) [211], VanDerHeijden [212] and LeNup (H3Q85C) [213], with the computed scores sorted by decreasing SHAP value importance as shown in Figure 3.

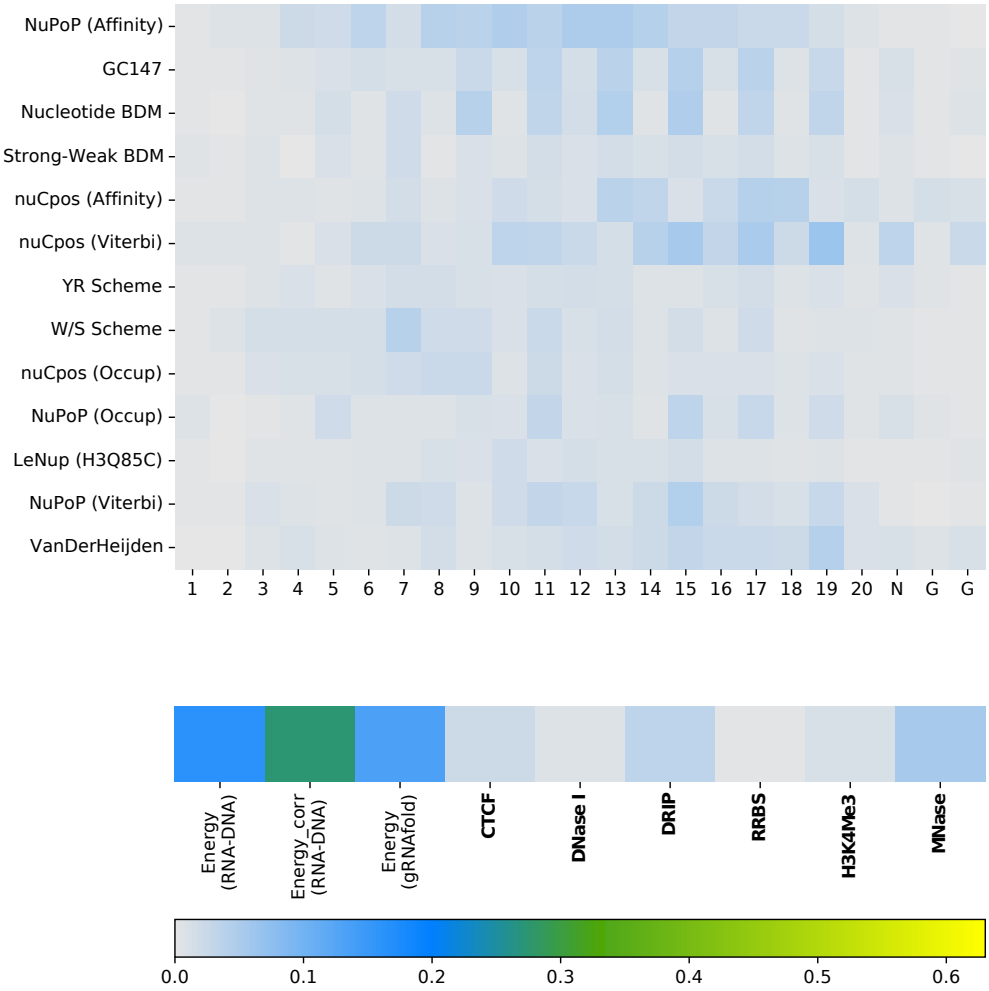


Figure A.8: Heatmap showing the mean absolute value of the SHAP values for the trained convolutional neural network's (CNN) base pair-resolved input features, which consist of the three CRISPRspec-derived energy terms $E_{\text{RNA-DNA}}$, $E^{\text{corr}}_{\text{RNA-DNA}}$ and E_{gRNAfold}, the four experimental epigenetic scores CTCF, DNase I, DRIP, RRBS, H3K4me3 and MNase (bolded), and the computed nucleosome organization-related scores GC147 [207], W/S scheme, YR scheme [208, 209], Strong-Weak BDM, Nucleotide BDM [205, 206], NuPoP (Occupancy), NuPoP (Affinity), NuPoP (Viterbi) [210], nuCpos (Occupancy), nuCpos (Affinity), nuCpos (Viterbi) [211], VanDerHeijden [212] and LeNup (H3Q85C) [213], with the computed scores sorted by decreasing SHAP value importance as shown in Figure 4.

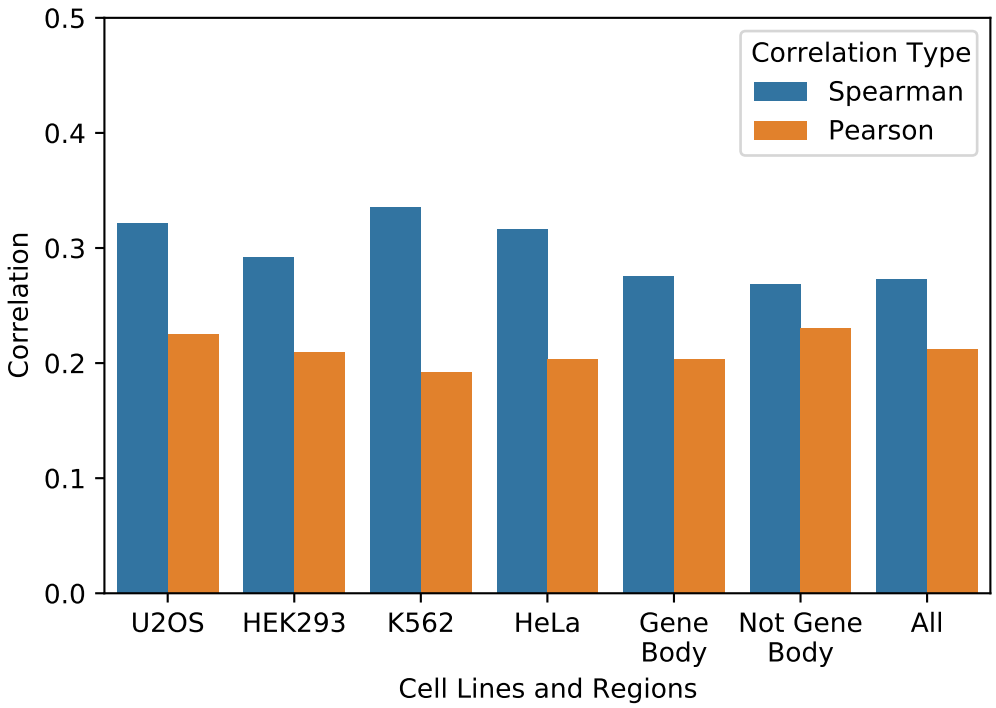


Figure A.9: Spearman and Pearson Correlations between NuPoP (Affinity) and Nucleotide BDM across different cell lines (U2OS, HEK293, K562, HeLa) and regions (Gene Body, Not Gene Body) for the dataset used in Figure 1.

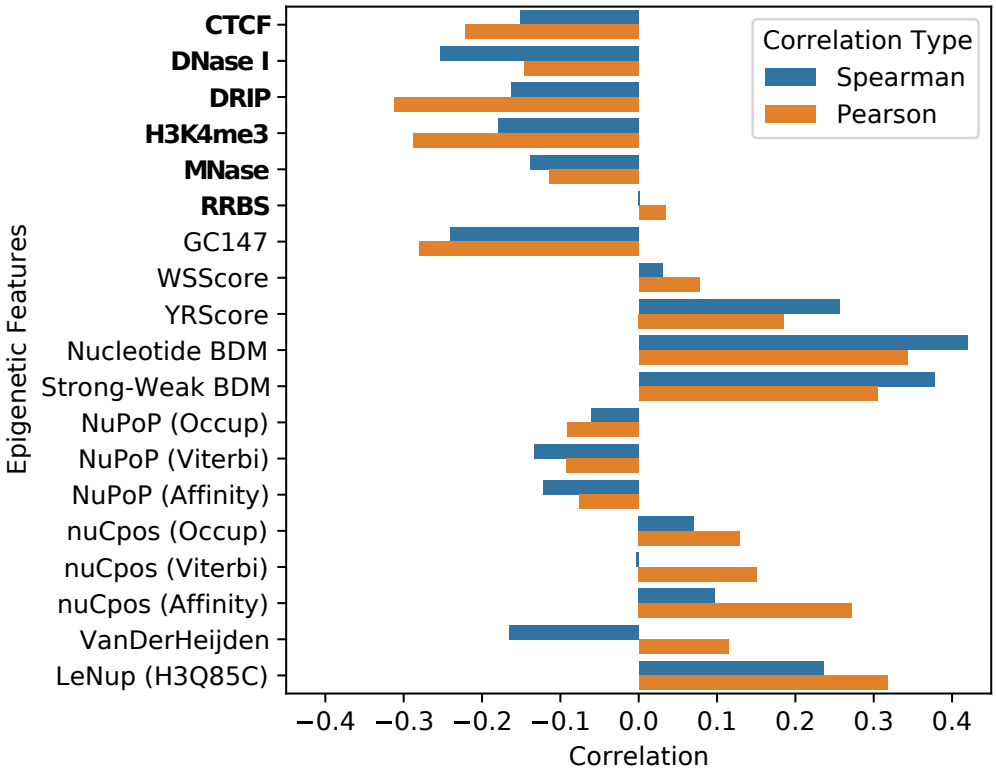


Figure A.10: Bar plot showing Spearman and Pearson correlations between 19 epigenetic features and SpCas9 on-target cleavage activities for all cell lines that contribute more than 1% to the crisprSQL dataset. The 19 epigenetic features consists of six experimental epigenetic features (bolded) and 13 nucleosome organization-related scores.

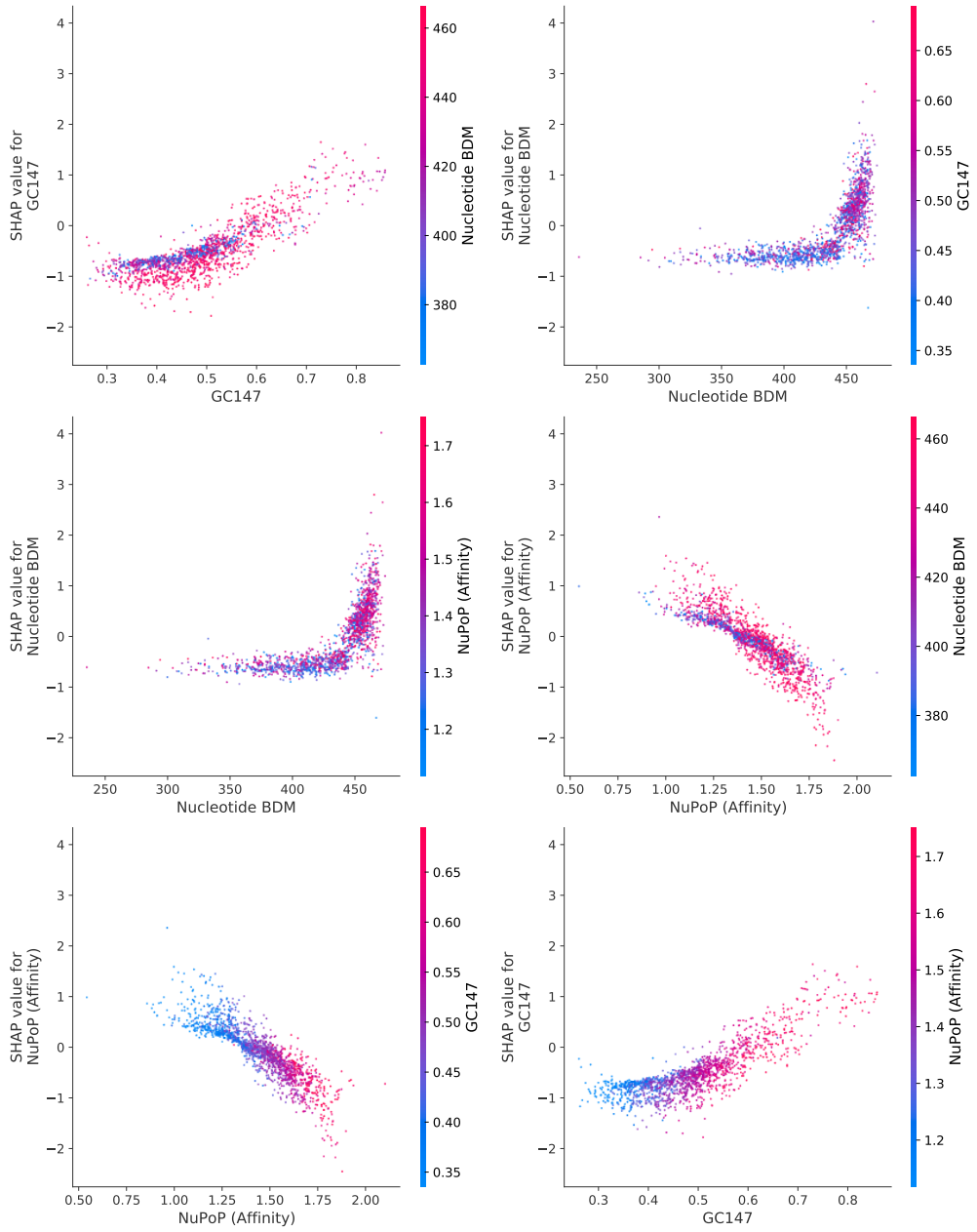


Figure A.11: SHAP dependency plots for GC147, Nucleotide BDM and NuPoP (Affinity) for XGBoost model.

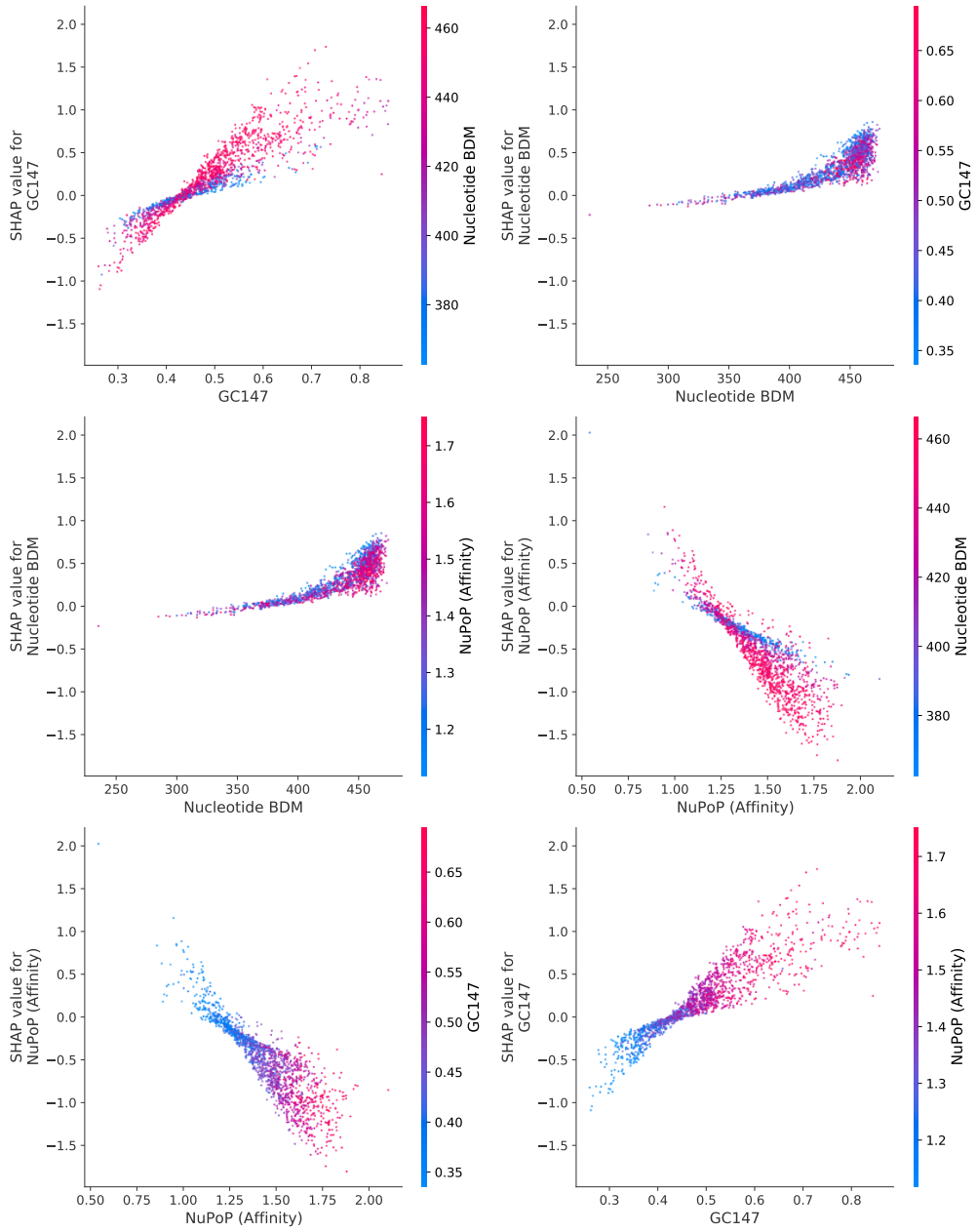


Figure A.12: SHAP dependency plots for GC147, Nucleotide BDM and NuPoP (Affinity) for CNN model.

A.2 Nucleosome organization-related tools

This section provides additional background on some of the nucleosome organization-related tools used in this study.

A.2.1 GC Content/GC147

Mathematically, the GC content of a 147 bp nucleosomal sequence s is defined by

$$GC147(s) = \frac{1}{147}\sum_{i=0}^{147} \mathbb{1}_{s_i \in \{G,C\}}.$$

GC content significantly correlates with *in vitro* nucleosome occupancy in budding yeast [304, 207]. Specifically, it was shown that GC content was the dominant feature in a linear model of nucleosome occupancy based on GC content and 14 other DNA sequence-related input features [207]. Nonetheless, GC content is not sufficient for *in vitro* nucleosome occupancy prediction, since it is not indicative of nucleosome occupancy levels in GC-rich regions *in vitro* [305].

A.2.2 W/S and YR schemes

The W/S scheme is based on the well-established DNA sequence pattern where weak-weak (WW) and strong-strong (SS) dinucleotides are periodically located on the histone octamer-facing minor and major grooves, respectively. Mathematically, W/S scheme is defined as

$$\begin{aligned}\mathrm{WSScore}(s) = &\sum_{s\in\text{minor sites}} C(WW,s)\\ &+ \sum_{s\in\text{major sites}} C(SS,s)\\ &- \sum_{s\in\text{minor sites}} C(SS,s)\\ &- \sum_{s\in\text{major sites}} C(WW,s)\end{aligned}$$

where $W \in \{A,T\}$, $S \in \{C,G\}$, minor sites $= \{s \in \mathbb{Z} | -6 \leq s \leq 6\}$, major sites $= \{s+0.5 | s \in \mathbb{Z}, -7 \leq s \leq 6\}$ and $C(\texttt{type}, s)$ denotes the number of `type` nucleotides in superhelical location (SHL) s.

The YR scheme is based on a weighted sum of GC, YR, YYRR and RYRY counts in the different sites. Further detailed descriptions on how the YR scheme predicts translational positioning can be found in [208] and [209].

A.2.3 Van Der Heijden algorithm

Based on the dinucleotide wedge model [306], the likelihood ratio for each base pair position is given by

$$P_{\mathrm{nuc}}(S) = 4^{|S|-1}\prod_{s=0}^{|S|-1} P_{\mathrm{dinuc}}(s, S[s, s+1])$$

where S is the sequence with $|S| \approx 147$ centered on the base position. In addition, the position-dependent dinucleotide probabilities are defined by

$$P_{\text{dinuc}}(s,d) = \begin{cases} 0.25 + B\sin(\frac{2\pi s}{p}) & \text{if } d \in \{AA, TA, TT\} \\ 0.25 + \frac{B}{3}\sin(\frac{2\pi s}{p}) & \text{if } d \in \{GA, GG, GT\} \\ 0.25 - B\sin(\frac{2\pi s}{p}) & \text{if } d \in \{GC, TC, TG\} \\ 0.25 - \frac{B}{3}\sin(\frac{2\pi s}{p}) & \text{if } d \in \{AC, AG, AT\} \\ 0.25 & \text{otherwise} \end{cases}$$

where B and p are the amplitude and period of the dinucleotide frequencies, respectively. Using the likelihood ratios, an energy landscape can be derived. We can then apply the algorithm required for solving Percus's equation [307] in order to generate the nucleosome positioning scores. Nucleosome occupancy values can then be obtained by applying a convolution operation with a 147 bp uniform filter. To determine the algorithm's hyperparameters, a Levenberg–Marquadt routine [308] can be used for fitting periodicity p and chemical potential μ to experimental data. In particular, μ is a hyperparameter used when computing the solution to Percus's equation. More details on VanDerHeijden can be found in [212].

A.2.4 Block Decomposition Method-based measures

Block Decomposition Method (BDM) is a training-free method for approximating the algorithmic complexity of sequences. Mathematically, BDM is founded on the Coding theorem method [309, 310], which relates algorithmic (Kolmogorov-Chaitin) complexity [311] with algorithmic probability [312]. Specifically, BDM approximates algorithmic complexity and Shannon entropy for short and long sequences, respectively [205]. Since DNA sequences can easily be represented as a string, BDM scores can readily be computed for DNA sequences.

A.2.5 NuPoP

NuPoP uses a dHMM and a Hidden Markov Model (HMM) for modelling 147 bp nucleosomal and linker DNA sequences, respectively. Training data for both models consist of yeast nucleosomal and non-nucleosomal sequences derived from MNase-seq. Both models are used for computing log likelihood ratios, which can be seen as histone binding affinity (HBA) scores. Computationally, the HBA score at position i is given by $\log \frac{P_N(S_i)}{G_L(S_i)}$ where S_i is the 147 bp sequence centered at position i. P_N and G_L indicate the probability that the S_i is a nucleosomal and linker sequence, respectively. Since linker sequences cannot be too long, NuPoP sets a maximum linker sequence length to 500bp for the dHMM. Using the HBA scores, the forward and backward algorithms can then be used for computing the nucleosome occupancy scores. A Viterbi score can also be computed, which predicts whether a specified nucleotide is located in nucleosomal or linker DNA. More details on the algorithm can be found in [210].

Appendix B

Supplementary materials for "Critical assessment of 3D nanoenvironment-based rational descriptors pertinent to CRISPR-Cas9 off-target cleavage activity"

B.1 STING descriptors

Here we provide a brief description of STING descriptors.

B.1.1 Accessibility

The Amino acid accessibility is calculated according to SurfV [1] program. We calculated five different values for accessibility: i) for the protein chain in isolation, ii) for the protein chain in complex with whatever other chain (if) present in the same PDB file and finally, iii) a relative accessibility, iv) a difference between accessibility in isolation and in complex (interface residues), v) a buried surface area (bsa). Details described at `http://sms.cbi.cnptia.embrapa.br/SMS/STINGm/help/solvent_accessible_area.html`. Other flavors of accessibility, such as calculated by Surface Racer [313] (a program already used by Blue Star STING for the calculation of Curvature) and NACCESS© [2] are also available at the STING RDB2. Additionally, the Shrake-Rupley accessibility (NSC) is also available for analysis [3]. These algorithms calculate an approximate accessible surface and therefore the corresponding accessibility is also an estimation; this is mainly due to the fact that obtaining precise surface description by those methods is computationally prohibitive, so different algorithms can result in different values of accessible area. Consequently, having all those accessible area values obtained using several algorithms allows one to compare and even define a consensus between the accessibility values of an atom or a residue.

List of accessibility descriptors:

1. Accessibility in isolation using SurfV, NACCESS and NSC: `acc_isol_surfv`, `acc_isol_naccess` and `acc_isol_nsc`,

2. Accessibility in complex using SurfV, NACCESS and NSC: `acc_complex_surfv`, `acc_complex_naccess` and `acc_complex_nsc`,

3. Relative accessibility (RSA) calculated as the ratio between the accessibility in isolation and the absolute solvent accessible area for specific residue type (acc^t_{max}) given in `http://sms.cbi.cnptia.embrapa.br/SMS/STINGm/help/solvent_accessible_area.html`: `acc_rsa_surfv`, `acc_rsa_naccess` and `acc_rsa_nsc`

$$acc\_rsa = \frac{acc\_isol}{acc^t_{max}} \tag{B.1}$$

4. Difference between accessibility in isolation and in complex (IFR): `acc_ifr_surfv`, `acc_ifr_naccess` and `acc_ifr_nsc`. Residues which have IFR lesser than accessibility in isolation are present in the protein's interface.

5. Buried Surface Area (BSA) is a measure of the size of the interface in a protein-protein complex. BSA is calculated as the ratio between the interface accessible area of a residue and its accessible area in isolation: `acc_bsa_surfv`, `acc_bsa_naccess`, `acc_bsa_nsc`

$$acc\_bsa = \frac{acc\_ifr}{acc\_isol} \tag{B.2}$$

The absolute (or maximum) accessibility is calculated for each amino acid type using four arbitrarily selected protein structures from PDB, where the selected amino acid type (one of 20) must be located at the C-terminal end. Then, we edited these structures so that the C-terminal residue would be left isolated in the "vacuum" (all other amino acid residues were deleted from the file). The SurfV software is then used to calculate the accessible surface area of the edited structure and among those 4 structures (for each of the 20 amino acids) we identified the maximum and minimum value for the ASA. The difference (rounded to the higher integer number) between the two values (taken percentwise) was added to the higher of two ASA values. We believe that this is much more convenient than taking the tabular data available in the literature, as the former one is a more realistic approach, based on the real experimental data within the environment of interest — a protein crystal.

B.1.2 Contact energy density

Amino acid contacts in terms of atomic interactions are essential factors to be considered in the analysis of the structure of a protein and its complexes. Residue-residue contacts are calculated according to the description given by Mancini et al. [201]. Contact Energy Density (CED) of internal protein contacts is then defined as the sum of contact energies (`https://www.cbi.cnptia.embrapa.br/SMS/STINGm/help/energy_contacts_table.html`) of the contacts established within a given sphere, among residues belonging to the same protein chain, and then divided by the volume of the probe sphere. Additionally, we also calculate the CED descriptors using the protein complex which include interchain contacts (CED IFR). For each residue in a protein we calculate 10 CED descriptors (five using only intrachain contacts and five also including interchain contacts) using five different radii for the probe sphere (3, 4, 5, 6 and 7Å).

$$CED_i = \frac{\sum_{c_t \in C^r_i} E(t)}{V(r)} \tag{B.3}$$

where C_i is the set of contacts of residue i within the probe sphere of radius r, c is a contact of type t, $E(t)$ is the energy for the contact of type t, and $V(r)$ is the volume of the probe sphere.

List of CED descriptors:

1. CED using only internal contacts with probe spheres of radius 3, 4, 5, 6, 7Å centered at residue's Cα atom: `ced_CA_3`, `ced_CA_4`, `ced_CA_5`, `ced_CA_6` and `ced_CA_7`,
2. CED using only internal contacts with probe spheres of radius 3, 4, 5, 6, 7Å centered at residue's LHA atom: `ced_LHA_3`, `ced_LHA_4`, `ced_LHA_5`, `ced_LHA_6` and `ced_LHA_7`,
3. CED including interchain contacts with probe spheres of radius 3, 4, 5, 6, 7Å centered at residue's Cα atom: `ced_CA_IFR_3`, `ced_CA_IFR_4`, `ced_CA_IFR_5`, `ced_CA_IFR_6` and `ced_CA_7`,
4. CED including interchain contacts with probe spheres of radius 3, 4, 5, 6, 7Å centered at residue's LHA atom: `ced_LHA_IFR_3`, `ced_LHA_IFR_4`, `ced_LHA_IFR_5`, `ced_LHA_IFR_6` and `ced_LHA_7`,

B.1.3 Cross link order

Cross Links are defined as contacts (from five types described above under RESIDUE CONTACTS) established among residues that are far apart in the protein primary sequence, but are close in its 3D fold. The order of cross link is defined as a number of such contacts (cross-links) established among independent stretches of the protein sequence (the size of which varies from 15, 20 to 30 amino acids). Only a single occurrence is counted for the Cross Link Order for a given amino acid residue, even though several such contacts could be observed "aiming" at the same "contacted" sequence stretch. In other words, a central amino acid can make more than one contact with the targeted sequence stretch and each one of those can be established with a different amino acid belonging to that same stretch of probing sequence size (15, 20 or 30 AAs long).

The higher the order, the more important that residue must be for the protein folding/stability/ binding. This specific STING descriptor is calculated by varying three input parameters:

- the size of the sequence stretch separating the residues in contact (15, 20 or 30 AA's long),
- the radius size of the probing sphere within which the contacts are counted (3.5, 5 and 8.5 Å), and
- The center of the probing sphere (being either C-α, C-β or Last Heavy Atom in the side chain).

List of CLO descriptors:

1. CLO with probe sphere centered at the C-α for three different stretch lengths (15, 20, and 30 AA's) and three different radii (3.5, 5, and 8.5 Å): `clo_35_15_CA`, `clo_5_15_CA`, `clo_85_15_CA`, `clo_35_20_CA`, `clo_5_20_CA`, `clo_85_20_CA`, `clo_35_30_CA`, `clo_5_30_CA`, `clo_85_30_CA`,
2. CLO with probe sphere centered at the C-β for three different stretch lengths (15, 20, and 30 AA's) and three different radii (3.5, 5, and 8.5 Å): `clo_35_15_CB`, `clo_5_15_CB`, `clo_85_15_CB`, `clo_35_20_CB`, `clo_5_20_CB`, `clo_85_20_CB`, `clo_35_30_CB`, `clo_5_30_CB`, `clo_85_30_CB`,

3. CLO with probe sphere centered at the LHA for three different stretch lengths (15, 20, and 30 AA's) and three different radii (3.5, 5, and 8.5 Å): `clo_35_15_LHA`, `clo_5_15_LHA`, `clo_85_15_LHA`, `clo_35_20_LHA`, `clo_5_20_LHA`, `clo_85_20_LHA`, `clo_35_30_LHA`, `clo_5_30_LHA`, `clo_85_30_LHA`,

B.1.4 Cross presence order

Cross Presence Order (CPO) is defined as a "presence" within a probing sphere (centered at a given residue) of any residue that is far apart in the protein primary sequence from the central residue, but is close in its 3D fold. Remaining details are equivalent to those described above under cross link order.

List of CPO descriptors:

1. CPO with probe sphere centered at the C-α for three different stretch lengths (15, 20, and 30 AA's) and three different radii (3.5, 5, and 8.5 Å): `cpo_35_15_CA`, `cpo_5_15_CA`, `cpo_85_15_CA`, `cpo_35_20_CA`, `cpo_5_20_CA`, `cpo_85_20_CA`, `cpo_35_30_CA`, `cpo_5_30_CA`, `cpo_85_30_CA`,
2. CPO with probe sphere centered at the C-β for three different stretch lengths (15, 20, and 30 AA's) and three different radii (3.5, 5, and 8.5 Å): `cpo_35_15_CB`, `cpo_5_15_CB`, `cpo_85_15_CB`, `cpo_35_20_CB`, `cpo_5_20_CB`, `cpo_85_20_CB`, `cpo_35_30_CB`, `cpo_5_30_CB`, `cpo_85_30_CB`,
3. CPO with probe sphere centered at the LHA for three different stretch lengths (15, 20, and 30 AA's) and three different radii (3.5, 5, and 8.5 Å): `cpo_35_15_LHA`, `cpo_5_15_LHA`, `cpo_85_15_LHA`, `cpo_35_20_LHA`, `cpo_5_20_LHA`, `cpo_85_20_LHA`, `cpo_35_30_LHA`, `cpo_5_30_LHA`, `cpo_85_30_LHA`,

B.1.5 Curvature

The curvature value for each amino acid is calculated using the Surface Racer [313] program. Surface Racer calculates the curvature value first for each atom. Then, a curvature of residues is obtained as an average of the surface atoms' curvatures. The curvature is defined at the atomic level, that is, each atom of the protein is assigned a curvature value corresponding to the region where it is located on the molecule's surface. A negative value is assigned to atoms in concave regions, positive for atoms in convex regions and the value zero to the buried atoms (atoms not at the protein surface). To assign values to the amino acid residues of each protein, Blue Star STING calculates the mean curvature considering only the residues on the protein's surface (curvature $\neq 0$). The curvature description is calculated for the chain in isolation and in complex with other chains (if present).

Using Surface Racer program we also calculate the accessible surface area (ASA) and molecular surface area (MSA) for each residue:

List of descriptors produced by Surface Racer:

1. Curvature in isolation: `curvature_isol`
2. Curvature in complex: `curvature_complex`
3. Accessible Surface Area (ASA) in isolation: `asa_isol`,
4. Accessible Surface Area (ASA) in complex: `asa_complex`,
5. Molecular Surface Area (MSA) in isolation: `msa_isol`,
6. Molecular Surface Area (MSA) in complex: `msa_complex`,

B.1.6 Density and sponge

The Density descriptor is the sum of total or partial atomic masses of atoms within a probe sphere divided by the volume of such sphere. Following the same kind of approach used to calculate the Density descriptor, but, instead of adding the total or partial mass of the atoms inside the spherical probe, in the Sponge descriptor the volume occupied by each atom is added (using the radius of van der Waals and disregarding the overlap volumes). This volume is then subtracted from and normalized by the volume of the spherical probe, resulting in a measure of the empty space in the nanoenvironment around each residue. Similar to the density calculation, sponge also introduces the same bias for those atoms located on the surface of the molecule. In the same way as the density descriptor, the Blue Star STING has pre-calculated 20 types of variations for Density and Sponge, resulting from the use of a sphere probes with variable radii — from 3 Å to 7 Å, centered on the α-carbons and LHA of each protein residue, and centered at P and C4 atoms of nucleotide residues. In addition, chains in isolation and in complex with other chains present in a given PDB file are considered.

List of Density/Sponge descriptors:

1. Density in isolation for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at Cα: `density_CA_3`, `density_CA_4`, `density_CA_5`, `density_CA_6`, `density_CA_7`,
2. Density in isolation for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at LHA: `density_LHA_3`, `density_LHA_4`, `density_LHA_5`, `density_LHA_6`, `density_LHA_7`,
3. Density in complex for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at Cα: `density_CA_3_IFR`, `density_CA_4_IFR`, `density_CA_5_IFR`, `density_CA_6_IFR`, `density_CA_7_IFR`,
4. Density in complex for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at LHA: `density_LHA_3_IFR`, `density_LHA_4_IFR`, `density_LHA_5_IFR`, `density_LHA_6_IFR`, `density_LHA_7_IFR`,
5. Density in isolation for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at P atom of a nucleotide: `density_P_3`, `density_P_4`, `density_P_5`, `density_P_6`, `density_P_7`,
6. Density in isolation for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at C4: `density_C4_3`, `density_C4_4`, `density_C4_5`, `density_C4_6`, `density_C4_7`,
7. Density in complex for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at P atom of a nucleotide: `density_P_3_IFR`, `density_P_4_IFR`, `density_P_5_IFR`, `density_P_6_IFR`, `density_P_7_IFR`,
8. Density in complex for probe sphere radius of 3, 4, 5, 6 and 7 Å centered at C4 atom of a nucleotide: `density_C4_3_IFR`, `density_C4_4_IFR`, `density_C4_5_IFR`, `density_C4_6_IFR`, `density_C4_7_IFR`,

B.1.7 Electrostatic potential

Electrostatic Potential (EP) is calculated using the program Delphi [314] according to the modifications done by Walter Rocchia and Goran Neshich [315]. The EP value is calculated on a per atom basis and then reported for all eligible PDB format files (including those containing modeled protein structures) in a residue-by-residue fashion. Four pre-calculated categories are stored for each residue:

1. EP at the residue's alpha carbon (CA) atom (`ep_CA`),
2. EP value at the side-chain's last heavy atom (LHA) of amino acid residue (`ep_LHA`),
3. average EP value over all amino acid atoms (`ep_average`), and
4. EP value averaged over the patch of the molecular surface that is attributable to that particular amino acid (`ep_surface`).

It is worth noting that the whole nanoenvironment EP plays a major corrective role in the four final EP reported values.

B.1.8 Entropy density

The Entropy Density descriptor is similar to the Contact Energy Density descriptor, but instead of the summing contacts energy values within a probe sphere, we use the relative entropy calculated according to HSSP [316]. Then the sum of relative entropies is divided by the volume of the probe sphere. Entropy in this case is referred to as disorder observable at certain location in alignment of homologous primary sequences. The Entropy Density is calculated by centering the probe sphere at the C-α and LHA atoms of each residue, and considering both the protein chain in isolation and in complex.

List of Entropy Density descriptors:

1. Probe sphere of radii 3, 4, 5, 6 and 7 Å centered at C-α for protein chain in isolation: `entd_CA_3`, `entd_CA_4`, `entd_CA_5`, `entd_CA_6`, `entd_CA_7`,
2. Probe sphere of radii 3, 4, 5, 6 and 7 Å centered at LHA for protein chain in isolation: `entd_LHA_3`, `entd_LHA_4`, `entd_LHA_5`, `entd_LHA_6`, `entd_LHA_7`,
3. Probe sphere of radii 3, 4, 5, 6 and 7 Å centered at C-α for protein chain in complex: `entd_CA_3_IFR`, `entd_CA_4_IFR`, `entd_CA_5_IFR`, `entd_CA_6_IFR`, `entd_CA_7_IFR`,
4. Probe sphere of radii 3, 4, 5, 6 and 7 Å centered at LHA for protein chain in complex: `entd_LHA_3_IFR`, `entd_LHA_4_IFR`, `entd_LHA_5_IFR`, `entd_LHA_6_IFR`, `entd_LHA_7_IFR`,

B.1.9 Graph descriptor

Protein chains can be represented as undirected graphs where the set of vertices is composed of a protein chain's amino acid residues (or atoms), while the edges of the graph represent interactions between these residues (or atoms) [317, 318, 319, 320]. In Blue Star STING, the amino acid residues of a protein were used as a set of vertices, and the set of edges is defined using previously calculated interatomic contacts. From a graph, it is possible to obtain several metrics and measures that extract and describe the behavior of protein chains as networks of interactions between amino acid residues. Representing geometric and topological properties of protein chains, these metrics can be considered as structural descriptors of proteins, since the graphs are constructed based on structural information of the protein chains.

List of Graph descriptors (for a definition of the graph metrics see [321, 322, 323]):

1. Eccentricity: (`eccentricity`),
2. Radiality Centrality: `radiality_centrality`,

3. Local Closeness: `local_closeness`,
4. Dice similarity: `dice_similarity`,
5. Mean Neighbor Degree (MND): `mean_neighbor_degree`,
6. Local average centrality (LAC): `lac`,
7. Density of Maximum Neighborhood Component (DMNC): `dmnc`,
8. Closeness: `closeness`,
9. Cluster coefficient: `cluster_coefficient`,
10. Degree: `degree`,
11. Betweenness: `betweenness`,
12. Random walk betweenness: `random_walk_betweenness`,
13. Bary center: `bary_center`,
14. Page rank: `page_rank`,
15. Bottleneck: `bottle_neck`

B.1.10 Hydrophobicity

Blue Star STING calculates hydrophobicity using the hydrophobicity scales defined by RADZICKA & WOLFENDEN [324] and KYTE & DOOLITTLE [325]. The Hydrophobicity of an amino acid residue i (where i is the residue sequence position in the protein's primary structure), of type t, is calculated using the value stipulated in above mentioned scales, weighted by the relative accessibility to the solvent (acc_{max}). We calculate hydrophobicity for each residue using the accessibilities for the protein chain in isolation and in complex.

$$Hydrophobicity_i = \frac{acc_i}{acc^t_{max}} * Hydrophobicity^S_t \tag{B.4}$$

where acc_i is the accessibility of residue i (in isolaiton or in complex), acc^t_{max} is the absoluute accessibility for the residue of type t and $Hydrophobicity^S_t$ is the hydrophobicity for the residue of type t as defined in the scale S.

List of hydrophobicity descriptors:

1. Hydrophobicity in isolation using the RADZICKA & WOLFENDEN scale: `hydro_radzicka_isol_surfv`, `hydro_radzicka_isol_naccess` and `hydro_radzicka_isol_nsc`,
2. Hydrophobicity in complex using the RADZICKA & WOLFENDEN scale: `hydro_radzicka_complex_surfv`, `hydro_radzicka_complex_naccess` and `hydro_radzicka_complex_nsc`,
3. Hydrophobicity in isolation using the KYTE & DOOLITTLE scale: `hydro_kite_dolitte_isol_surfv`, `hydro_kite_dolitte_isol_naccess` and `hydro_kite_dolitte_isol_nsc`,
4. Hydrophobicity in complex using the KYTE & DOOLITTLE scale: `hydro_kite_dolitte_complex_surfv`, `hydro_kite_dolitte_complex_naccess` and `hydro_kite_dolitte_complex_nsc`,

B.1.11 Residue contacts

Amino acid contacts in terms of atomic interactions are essential factors to be considered in the analysis of a protein's structure and its complexes. Residue-residue contacts are calculated according to description given in Mancini et al. [201]. Contact types considered in STING RDB are:

1. Hydrophobic interactions (energy: 0.6 Kcal/mol),
2. Hydrogen Bonding (energy: 2.6 Kcal/mol),
3. Aromatic Stacking (energy: 1.5 Kcal/mol),
4. Salt bridging (energy: 10.0 Kcal/mol),
5. Cysteine-bridging (energy: 85.0 Kcal/mol).

B.1.12 Secondary structure

In Blue Star STING, there are three different secondary structure assignments for an amino acid residue, obtained by consulting information contained in the PDB files themselves (when available) and those calculated by softwares DSSP [326] and STRIDE [327]. Often, the types of secondary structures, as well as the initial and final amino acid position in addition to sizes of particular secondary structure elements (SSE), diverge between the different classifications. Thus, with the presence of values/descriptions in STING RDB coming from three different sources, it is possible to obtain a consensus and find regions with more reliable secondary structures assignments. Noteworthy is the observation that some protein districts appear to have preferences to specific secondary structure configuration in order to perform certain function (structural or enzymatic). Each program (DSSP and STRIDE) produces different outputs, which we store in STING RDB2. In order to make them comparable we provide a mapping between the program's secondary structure encoding schema to a common schema: **H** (alpha helix); **G** (310 helix); **I** (Pi Helix); **E** or **D** (extended strand in parallel and/or anti-parallel B-sheet conformation); **B** or **b** (isolated B-bridge); **T** (turn); **C** (coil); **S** (bend).

List of descriptors produced by DSSP:

1. Secondary structure: one of the codes in the common schema (`secondary_structure`),
2. Kappa: virtual bond angle (bend angle) defined by the three Cα atoms of residues $i-2$, i, $i+2$. Used to define bend (structure code **S**) (`kappa`),
3. Dihedral angles ϕ and ψ (`psi` and `psi`),
4. Accessibility: DSSP calculated accessibility (`accessibility`)

List of descriptors produced by STRIDE:

1. Secondary structure: one of the codes in the common schema (`secondary_structure`),
2. Secondary structure elements: delimitation of starting and ending residues of a SS element (consecutive residues forming a major SS) - first residue in a SS element (`>`), last residue in a SS element (`<`) and a residue within a SS element (`=`),
3. Dihedral angles ϕ and ψ (`psi` and `psi`),
4. Accessibility: DSSP calculated accessibility (`accessibility`)

B.1.13 Side chain orientation

The side chain orientation [328] is calculated for each amino acid residue in a protein chain as an angle formed between two vectors: Cα-CENTROID and Cα-LHA. The Cα-CENTROID is the vector from the C-α atom of an amino acid residue to the center of mass of a specific region (probe sphere), and the Cα-LHA is the vector from the C-α atom to the Last Heavy Atom of that same amino acid residue. Then, for each amino acid residue, we calculate how much their side chains (Cα-LHA vector) deviate from the vector pointing to the center of mass of the probing sphere. Additionally, we also calculate the average angle of all amino acid residues found within the probing sphere. Finally, we subtract that angle calculated for each amino acid separately, from the average one, giving us a description of how divergent or convergent the side chain of any residue is compared to its neighbors.

List of Side Chain Orientation descriptors:

1. Side chain orientation angle for probe sphere of radii 3, 4, 5, 6, and 7 Å: `side_chain_angle_3`, `side_chain_angle_4`, `side_chain_angle_5`, `side_chain_angle_6`, `side_chain_angle_7`,
2. Side chain orientation using average angle for probe sphere of radii 3, 4, 5, 6, and 7 Å: `side_chain_average_angle_3`, `side_chain_average_angle_4`, `side_chain_average_angle_5`, `side_chain_average_angle_6`, `side_chain_average_angle_7`,
3. Side chain orientation angle of neighbor atoms for probe sphere of radii 3, 4, 5, 6, and 7 Å: `neighbors_side_chain_angle_3`, `neighbors_side_chain_angle_4`, `neighbors_side_chain_angle_5`, `neighbors_side_chain_angle_6`, `neighbors_side_chain_angle_7`

B.1.14 Solvation (energy)

The solvation energy corresponds to the energy of atom bonds established between the solute and the solvent. In case the solvent is water, this is also called the free energy of hydration. From the relative solvent accessible area of each protein atom, it is possible to calculate approximately the free energy of hydration that originates from the interactions between such atoms and the water molecules (solvent) [329]). The free energy of hydration of the i-th atom of an amino acid residue is calculated as the product of its experimentally determined atomic solvation parameter (g_i) and the relative solvent accessible area (RSA). The sum of the free energies of hydration of the atoms that make up the residue r and the atoms in the vicinity of r, normalized by the sum of the relative areas accessible to the solvent of all atoms considered, gives the solvation energy of the residue r (G_r) with the formula:

$$G_r = \frac{\sum_i g_i ASA_{relative}}{\sum_i ASA_{relative}} \tag{B.5}$$

List of Solvation descriptors:

1. Solvation for pobre sphere of radii 3, 4, 5, 6, and 7 Å: `solvation_3`, `solvation_4`, `solvation_5`, `solvation_6`, `solvation_7`

B.1.15 Unused contacts

Each residue can make certain (maximum) number of contacts. The difference between the maximum number of contacts and the contacts established is defined as "unused contacts". We maintain a table describing the maximum number of interatomic contacts identified for each of the 20 amino acid types (`http://www.cbi.cnptia.embrapa.br/SMS/STINGm/help/table_of_max_number_contacts.html`). The contacts are classified by the contact type (and residue type) and were extracted only from those PDB files containing the structures resolved by the X-ray crystallography and having the resolution better or equal to 2.0 Å. Furthermore, structures with identified double occupancy atoms, were not considered. This table is consulted with every PDB update and if necessary, numbers in the table are updated to reflect possible changes in occurrence of maximum number of contacts for each contact type and each residue type. When the table is updated the Unused Contacts descriptor is recalculated for all structures in the PDB.

List of Unused Contacts descriptors:

1. Number of unused contacts per contact type:
   - Hydrophobic: `hydrophobic_uc`,
   - Charge attractive: `charge_attr_uc`,
   - Charge repulsive: `charge_repu_uc`,
   - H-Bond between atoms of the residues' main chains, including zero, one or two intermediate waters: `hb_mm_uc`, `hb_mwm_uc` and `hb_mwwm_uc`,
   - H-Bond between atoms of the residues' side chains, including zero, one or two intermediate waters: `hb_ss_uc`, `hb_sws_uc` and `hb_swws_uc`,
   - H-Bond between atoms of the residues' main chain and side chain , including zero, one or two intermediate waters: `hb_ms_uc`, `hb_mws_uc` and `hb_mwws_uc`,
   - Aromatic: `aromatic_uc`,
   - Disulfide bridge: `ss_bond_uc`
2. Energy of unused contacts per contact type:
   - Hydrophobic: `hydrophobic_uc_energy`,
   - Charge attractive: `charge_attr_uc_energy`,
   - Charge repulsive: `charge_repu_uc_energy`,
   - H-Bond between atoms of the residues' main chains, including zero, one or two intermediate waters: `hb_mm_uc_energy`, `hb_mwm_uc_energy` and `hb_mwwm_uc_energy`,
   - H-Bond between atoms of the residues' side chains, including zero, one or two intermediate waters: `hb_ss_uc_energy`, `hb_sws_uc_energy` and `hb_swws_uc_energy`,
   - H-Bond between atoms of the residues' main chain and side chain , including zero, one or two intermediate waters: `hb_ms_uc_energy`, `hb_mws_uc_energy` and `hb_mwws_uc_energy`,
   - Aromatic: `aromatic_uc_energy`,
   - Disulfide bridge: `ss_bond_uc_energy`

B.1.16 Weighted contact number

The Weighted Contact Number (WCN) is a measure of backbone flexibility of amino acid residues [328]. For each amino acid residue in the protein chain we calculated the WCN and average WCN according to the following equations:

$$WCN_i = \sum_{j \neq i} \frac{1}{r_{ij}^2} \tag{B.6}$$

where j is any other residues in the protein chain and r_{ij}^2 is the squared distance between the Cα atoms of residue i and j, and

$$\overline{WCN_i} = \sum_{j \in k} \frac{z_j}{K} \tag{B.7}$$

where k are the nearest neighbors of residue i (square euclidean distance between Cα atoms), z_j is the normalized WCN of residue j (z-score) and K is the number of nearest neighbors residue ($|k| \leq K$).

List of Weighted Contact Number descriptors:

1. Weighted contact number: `weighted_contact_number`,
2. Average Weighted contact number for layers (k) of 2, 3, 4 and 5: `avg_weighted_contact_number_k_2`, `avg_weighted_contact_number_k_3`, `avg_weighted_contact_number_k_4`, `avg_weighted_contact_number_k_5`

B.1.17 Neighbor descriptors

In addition to the previously described descriptors, we calculated a class of descriptors called *Neighbor Descriptors*. The Neighbor Descriptors (ND) are calculated by performing an aggregation of a *base descriptor* using a neighborhood definition.

Weighted neighbor average descriptor

This descriptor is a type of a Neighbor Descriptor (ND) inspired by the work of POROLLO & MELLER [330]. The Weighted Neighbor Average or simply WNA were calculated and stored. There are two Weighted Neighbor Average (WNA) descriptors for each *base descriptor* in Blue Star STING. The first of them uses values of the relative accessibility (RSA) for the neighboring residues used for adding specific weight to the base descriptors (therefore yielding the WNASurface), and the second one uses the inverse of the distance between the central residue and its neighbors (WNADistance). To define the neighborhood for a selected residue, a sphere of 15 Å radius centered on residue's α-carbon was used. In the case of WNASurface descriptors that use relative accessibility as a weighting factor, only residues with a relative area accessible to the solvent greater than 5% are considered. The neighbor descriptors are calculated for all numerical descriptors in Blue Star STING.

For each *base descriptor* d_i of a residue i the WNASurface descriptor is calculated as follows:

$$WNASurface_i = \frac{\sum_{j \in N_i \wedge acc\_rsa_j > 5\%} acc\_rsa_j * d_j}{V(r)} \tag{B.8}$$

where N_i is the set of neighbor residues of the residue i (including i itself), acc_rsa_j is the RSA of residue j, d_j is the value of the *base descriptor* for residue j and $V(r)$ is the

volume of the sphere (i.e., the neighborhood). In this work we use $r = 15\,\text{Å}$.

The WNADistance descriptor is calculate as:

$$WNADistance_i = \frac{\sum_{j \in N_i} \frac{d_j}{D_{i,j}}}{V(r)} \tag{B.9}$$

where $D_{i,j}$ is the euclidean distance between residues i and j.

Sliding window neighbor descriptor

The Sliding Window (SW) uses the primary structure to define the neighborhood of a residue. For each *base descriptor* d_i of a residue i the SW descriptor is calculated as follow:

$$SW_i = \frac{\sum_{(i-L/2) \leq j \leq (i+L/2)} d_i}{L} \tag{B.10}$$

where L is the window length and j is a sequence neighbor of residue i.

STING RDB2 stores SW descriptors calculated using four window lengths: 3, 5, 7, and 9 amino acids. The SW descriptors are calculated for all numerical descriptors and they are named in the form `base_descriptor_name_SW_`L.

Graph neighbor descriptor

In the same way as the graphs are constructed to represent protein chains, they can be used to define a neighborhood. The neighborhood can then be used for calculating neighbor descriptors.

For this, a maximum value is defined for the neighborhood size (K), representing the number of edges or the length of the shortest path between a central residue and its neighbors. That is, for $K = 1$, only the vertices immediately adjacent to the central atom are considered as neighbors. In the case of $K = 2$, the adjacent neighbors plus the neighbors of these neighbors are considered, and so on for larger values of K. Therefore, let r_i be any amino acid residue and f_i be a *base descriptor* for this residue, then the Graph Neighbor (GN) descriptor can be calculated for the neighborhood (g_{f_i}) from the graph $G(V, E)$, where V is the set of vertices or amino acid residues and E is the set of edges or contacts. Considering as "neighbors" of the vertex r_i, those amino acid residues that were visited by a minimum path of a maximum length (equal to K), where $d_{i,j}$ is the distance (number of edges) between the residue and its neighbor j, the calculation of g_{f_i} is performed considering different weights for the neighboring layers ($k = 1, 2, 3, ..., K$):

$$g_{f_i} = f_i + \sum_{k=1}^{K} \frac{\sum_{\forall j | d(i,j)<k} f_j}{k^2} \tag{B.11}$$

The STING RDB2 stores pre-calculated GN descriptor using $K = 6$, and the descriptors are named in the form: `base_descriptor_name_GN`.

B.1.18 All descriptors

For all descriptors used in this work, we used STING SDLg (Sting Data Library generator) to calculate them in all possible variants (meaning, using all values for variables used to calculate each one of them) and to apply batch calculations on sgRNA-DNA-Cas9 complexes modeled in the molecular dynamics simulations as described previously.

B.2 Supplementary methods

B.2.1 SHAP model interpretation

We compute feature counts and SHAP importances of parent descriptor classes for each residue cluster by stratifying features into their respective residue clusters.

Cas9 residues may be on the surface of the isolated protein, the surface of the Cas9 complex, and/or the interface between Cas9 and non-Cas9 components in the complex. However, unlike the previous properties, these properties dynamically change across different PDB snapshots. For example, a residue may be on the interface in some snapshots but not in others. Based on the above, we compute:

- the average number of surface residues $\frac{1}{|D|}\sum_{i=1}^{|D|}\alpha_r^{(i)}$ and non-surface residues $\frac{1}{|D|}\sum_{i=1}^{|D|}(1-\alpha_r^{(i)})$
- the SHAP importance of surface residues $I_{\text{surface}} = \frac{1}{|D|}|\sum_{r\in R}(\alpha_r^{(i)}\sum_{j\in F(r)}\phi_j^{(i)})|$ and non-surface residues $I_{\text{non-surface}} = \frac{1}{|D|}|\sum_{r\in R}((1-\alpha_r^{(i)})\sum_{j\in F(r)}\phi_j^{(i)})|$
- the average number of interface residues $\frac{1}{|D|}\sum_{i=1}^{|D|}\alpha_{r,IFR}^{(i)}$ and non-interface residues $\frac{1}{|D|}\sum_{i=1}^{|D|}(1-\alpha_{r,IFR}^{(i)})$; and
- the SHAP importance of interface residues $I_{\text{interface}} = \frac{1}{|D|}\sum_{r\in R}(\alpha_{r,IFR}^{(i)}\sum_{j\in F(r)}\phi_j^{(i)})|$ and non-interface residues $I_{\text{non-surface}} = \frac{1}{|D|}|\sum_{r\in R}((1-\alpha_{r,IFR}^{(i)})\sum_{j\in F(r)}\phi_j^{(i)})|$

where:

- R is the set of residues in STING_CRISPR;
- $F(r)$ is the set of features with residue r;
- $\alpha_r^{(i)} = 1$ if residue r is a surface residue in PDB snapshot i, and 0 otherwise; and
- $\alpha_{r,IFR}^{(i)} = 1$ if residue r is an interface residue in PDB snapshot i, and 0 otherwise.

We also compute the fraction of snapshots in which a residue is a surface or an interface residue, which are given by $\frac{1}{|D|}\sum_{i=1}^{|D|}\alpha_r^{(i)}$ and $\frac{1}{|D|}\sum_{i=1}^{|D|}\alpha_{r,IFR}^{(i)}$, respectively. We repeat the above calculations for the three accessibility tools SurfV [1], NACCESS [2] and NSC [3]. We use bar plots to visualize the above SHAP importances, and use a heatmap to visualize the fractions.

To quantify the importance of HPR for predicting CRISPR-Cas9 cleavage activity, we repeat the same training procedure, but considered the $1361\dot{1}671 = 2274231$ features spanning Cas9 residues $3-1363$ in place of the 380 HPRs, thereby obtaining a different model STING_CRISPR_ALL. HPRs are not always proximal to the heteroduplex in all PDB snapshots. Because of this, using STING_CRISPR_ALL, we calculate:

- feature counts of HPRs $\frac{1}{|D|}\sum_{i=1}^{|D|}\delta_{r,HPR}^{(i)}$ and non-HPRs $\frac{1}{|D|}\sum_{i=1}^{|D|}(1-\delta_{r,HPR}^{(i)})$;
- SHAP importance of HPRs $\frac{1}{|D|}\sum_{i=1}^{|D|}|\sum_{r\in R}\delta_{r,HPR}^{(i)}(\sum_{j\in F(r)}\phi_j^{(i)})|$; and
- SHAP importance of non-HPRs $\frac{1}{|D|}\sum_{i=1}^{|D|}|\sum_{r\in R}(1-\delta_{r,HPR}^{(i)})(\sum_{j\in F(r)}\phi_j^{(i)})|$,

where $\delta^{(i)}_{r,HPR} = 1$ if residue r is a HPR in PDB snapshot i, and 0 otherwise, and $F(r)$ is the set of features with residue r, and use heatmaps to visualize the importances.

We also project residue importances onto the heteroduplex bases to measure their importances. Namely, for a given heteroduplex base b, we can quantify its importance I_b by the following equation: $I_b = \sum_{i=1}^{|D|} |\sum_{r \in R(b)} d_r^{(i)} (\sum_{j \in F(r)} \phi_j^{(i)})|$, where $R(b)$ denotes the set of residues whose α-carbon atom is 3-7 Å away from the C4' atom of base b. Similarly, we can also quantify the SHAP importance of residue-base pairs by calculating $I_{b,r} = \sum_{i=1}^{|D|} |d_r^{(i)} (\sum_{j \in F(r)} \phi_j^{(i)})|$.

B.2.2 Raw data

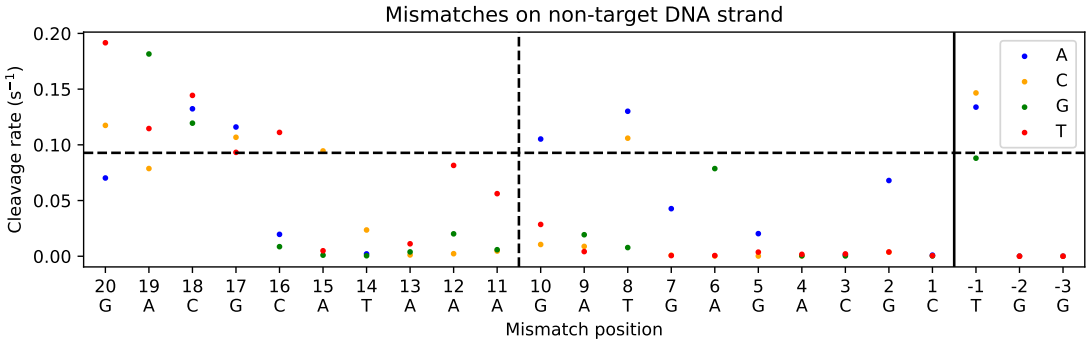


Figure B.1: CRISPR-Cas9 (off-)target cleavage activity for on- and off-target interfaces listed in Table B.1, ultimately from Jone Jr. et al. [156]. The nucleotide sequence under the x-axis shows the nucleotides in the on-target interface's non-target strand, and the dots' colors show the resulting mutated nucleotide at the particular nucleotide position. The horizontal dotted line shows the on-target's activity, and the vertical dotted line separates PAM-distal positions (+20 to +11) from PAM-proximal positions (+10 to +1).

Target Type	CMUT	Target Site (Non-Target Strand)	Mutation	CRISPR-Cas9 Cleavage Activity
On-target	CMUT1	GACGCATAAAGATGAGACGCTGG	None	0.0928059068718
Off-target	CMUT2	G**C**CGCATAAAGATGAGACGCTGG	A19C	0.0787201649863
	CMUT8	G**G**CGCATAAAGATGAGACGCTGG	A19G	0.181542289862
	CMUT4	G**T**CGCATAAAGATGAGACGCTGG	A19T	0.114618189415
	CMUT5	GA**A**GCATAAAGATGAGACGCTGG	C18A	0.132345410767
	CMUT3	GA**G**GCATAAAGATGAGACGCTGG	C18G	0.119446527474
	CMUT7	GA**T**GCATAAAGATGAGACGCTGG	C18T	0.14436947128
	CMUT10	GAC**A**CATAAAGATGAGACGCTGG	G17A	0.115995295517
	CMUT13	GAC**C**CATAAAGATGAGACGCTGG	G17C	0.106781628351
	CMUT19	GAC**T**CATAAAGATGAGACGCTGG	G17T	0.0932489445754
	CMUT21	GACG**A**ATAAAGATGAGACGCTGG	C16A	0.0196421394474
	CMUT9	GACG**G**ATAAAGATGAGACGCTGG	C16G	0.00859648371669
	CMUT12	GACG**T**ATAAAGATGAGACGCTGG	C16T	0.111148283845
	CMUT29	GACGC**C**TAAAGATGAGACGCTGG	A15C	0.0945242769362
	CMUT23	GACGC**G**TAAAGATGAGACGCTGG	A15G	0.000971813092723
	CMUT27	GACGC**T**TAAAGATGAGACGCTGG	A15T	0.00493182712775
	CMUT17	GACGCA**A**AAAGATGAGACGCTGG	T14A	0.00208172468785
	CMUT30	GACGCA**C**AAAGATGAGACGCTGG	T14C	0.0235738673331
	CMUT28	GACGCA**G**AAAGATGAGACGCTGG	T14G	0.000458729016148
	CMUT25	GACGCAT**C**AAGATGAGACGCTGG	A13C	0.00116725554782
	CMUT20	GACGCAT**G**AAGATGAGACGCTGG	A13G	0.00389589061704
	CMUT26	GACGCAT**T**AAGATGAGACGCTGG	A13T	0.0111729840733
	CMUT18	GACGCATA**C**AGATGAGACGCTGG	A12C	0.00228748213652
	CMUT14	GACGCATA**G**AGATGAGACGCTGG	A12G	0.0201204142155
	CMUT24	GACGCATA**T**AGATGAGACGCTGG	A12T	0.0815231184044
	CMUT22	GACGCATAA**C**GATGAGACGCTGG	A11C	0.00465650328521
	CMUT16	GACGCATAA**G**GATGAGACGCTGG	A11G	0.00586891296755
	CMUT11	GACGCATAA**T**GATGAGACGCTGG	A11T	0.0561840566924
Off-target (unused)	CMUT6	**A**ACGCATAAAGATGAGACGCTGG	G20A	0.0702299155412
	CMUT15	**T**ACGCATAAAGATGAGACGCTGG	G20T	0.191606618035

Table B.1: The 30 (1 on-target and 29 off-target) CRISPR-Cas9 guide-target interfaces initially considered in this study. Sorted by mismatch position, 27 of the 29 off-target interfaces consist of single mismatches +19 to +11 nucleotides away from the PAM. Cleavage activity values are extracted from the column "wtCas9_cleave_rate_log" within Supplementary File 2 of Jones Jr. et al. [156]. All-atom molecular dynamic trajectories are not produced for CMUT6 and CMUT15, and thus they are discarded when creating the machine learning dataset.

B.2.3 Heteroduplex-proximal residues

This study considers six sets of sgRNA-TS heteroduplex-proximal residues (HPRs): one for each fold during five-fold cross validation and one for the train-test split. In summary, all six HPR sets share the following Cas9 residues: 13, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 78, 136, 139, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 218, 249, 260, 263, 265, 266, 267, 268, 269, 270, 271, 301, 317, 321, 324, 362, 364, 365, 366, 367, 368, 369, 370, 371, 374, 396, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 414, 415, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 460, 461, 462, 463, 464, 465, 475, 478, 488, 489, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 503, 506, 507, 508, 509, 510, 511, 515, 516, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 536, 537, 538, 539, 557, 558, 559, 560, 561, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 624, 625, 626, 627, 628, 631, 655, 656, 657, 658, 659, 660, 661, 662, 663, 666, 667, 683, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 705, 708, 709, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 737, 761, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 802, 803, 804, 806, 807, 808, 809, 810, 812, 816, 833, 834, 835, 836, 837, 838, 839, 840, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 859, 860, 861, 862, 864, 866, 867, 868, 869, 893, 895, 896, 908, 913, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 936, 937, 941, 948, 949, 951, 955, 956, 957, 958, 959, 960, 961, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1038, 1039, 1106, 1107, 1108, 1109, 1110, 1111, 1122, 1134, 1135, 1136, 1138. Each HPR set additionally have the following residues:

- Fold 0 (15 extra residues): 261, 416, 512, 621, 629, 710, 762, 813, 858, 863, 870, 910, 934, 938, 940;
- Fold 1 (16 extra residues): 217, 261, 416, 490, 512, 621, 629, 710, 762, 813, 858, 863, 870, 910, 934, 940;
- Fold 2 (13 extra residues): 217, 261, 416, 490, 621, 629, 710, 762, 813, 858, 870, 910, 938;
- Fold 3 (13 extra residues): 217, 261, 416, 490, 512, 629, 762, 858, 863, 910, 934, 938, 940;
- Fold 4 (11 extra residues): 217, 490, 512, 621, 710, 813, 863, 870, 934, 938, 940
- Train-test split (17 extra residues): 217, 261, 416, 490, 512, 621, 629, 710, 762, 813, 858, 863, 870, 910, 934, 938, 940

In terms of counts, HPRs for the five folds have 378, 379, 376, 376 and 374 residues, respectively, and the train-test split HPR has 380 residues.

B.2.4 STING_CRISPR: an ExtraTrees model with Cas9 STING features

Table B.3 lists all 21 features grouped by STING descriptors. In total, there are 9 unique descriptor classes, 16 unique descriptors and 15 unique amino acids (136, 271, 317, 406, 730, 731, 732, 733, 734, 837, 838, 839, 925, 1015, 1016) among the amino acid-specific STING descriptor features.

Parent descriptor class (neighbor aggregations)	No. of descriptors	No. of features
Accessibility	15	5700
Cross Link Order (GN, SW, WNA, VD)	216 (= 27 + 108 + 54 + 27)	82080
Cross Presence Order (GN, SW, WNA, VD)	216 (= 27 + 108 + 54 + 27)	82080
Curvature (GN, SW, WNA, VD)	96 (= 12 + 48 + 24 + 12)	36480
Density (GN, SW, WNA, VD)	160 (= 20 + 80 + 40 + 20)	60800
Sponge (GN, SW, WNA, VD)	160 (= 20 + 80 + 40 + 20)	60800
Contact Energy Density (GN, SW, WNA, VD)	160 (= 20 + 80 + 40 + 20)	60800
DSSP	15	5700
Stride	13	4940
Electrostatic Potential (GN, SW, WNA, VD)	32 (= 4 + 16 + 8 + 4)	12160
Entropy Density (GN, SW, WNA, VD)	160 (= 20 + 80 + 40 + 20)	60800
Graph Descriptor (GN, SW, WNA, VD)	128 (= 16 + 64 + 32 + 16)	48640
Hydrophobicity	12	4656
Residue Contact (GN, SW, WNA, VD)	72 (= 9 + 36 + 18 + 9)	27360
Side Chain Orientation (GN, SW, WNA, VD)	120 (= 15 + 60 + 30 + 15)	45600
Solvation (GN, SW, WNA, VD)	40 (= 5 + 20 + 10 + 5)	15200
Unused Contacts (GN, SW, WNA, VD)	16 (= 2 + 8 + 4 + 2)	6080
Weighted Contact Number (GN, SW, WNA, VD)	40 (= 5 + 20 + 10 + 5)	15200
Total	1671	634980

Table B.2: Number of descriptors and features generated from the 60 STING descriptor classes used for characterizing CRISPR-Cas9's internal protein nanoenvironment in this study. The left column lists the 17 parent descriptor classes and their corresponding list of relevant neighbor aggregation methods (GN = Graph Neighbors, SW = Sliding Window, WNA = Weighted Neighbor Average, VD = Voronoi Diagram). The middle column shows the total number of descriptors for each parent descriptor class, with a breakdown of the count enclosed in parentheses if neighbor descriptors are used instead. The right column shows the number of residue-resolved features considered as ML input features for each parent descriptor class when building STING_CRISPR. For STING_CRISPR, we use 380 sgRNA-target strand DNA heteroduplex-proximal residues in order to model the CRISPR-Cas9 protein 3D nanoenvironment close to the sgRNA-target strand DNA heteroduplex. As a result, numbers on the right column is calculated by multiplying the number of descriptors by 380, the number of heteroduplex-proximal residues. In total, there are 634980 amino acid-resolved features considered in this study.

Parent descriptor class	Neighbour Type	Descriptor name	CRISPR-Cas9 residue(s)
Accessibility	-	acc_ifr_surfv	730
Accessibility	-	acc_isol_surfv	837
Contact Energy Density	VD	ced_CA_4_VD	837
Contact Energy Density	WNA	ced_LHA_4_WNADist	164
Cross Presence Order	SW	cpo_85_30_CB_SW_3	730
Cross Presence Order	WNA	cpo_85_30_CB_WNADist	136
Cross Presence Order	WNA	cpo_85_15_CB_WNADist	908
Density	GN	density_CA_7_IFR_GN	415
Density	SW	density_LHA_6_SW_3	408
Density	SW	density_LHA_7_SW_7	734
Density	SW	density_CA_4_SW_3	919
Density	SW	density_LHA_6_SW_3	1017
Density	VD	density_LHA_5_IFR_VD	402
Density	VD	density_CA_7_VD	732
Electrostatic Potential	GN	ep_average_GN	317
Electrostatic Potential	WNA	ep_average_WNADist	839
Entropy Density	SW	entd_CA_5_SW_5	268
Entropy Density	VD	entd_CA_5_VD	837
Entropy Density	WNA	entd_CA_3_WNADist	415
Entropy Density	WNA	entd_CA_6_WNADist	838
Graph Descriptor	SW	betweenness_SW_7	1122
Graph Descriptor	VD	closeness_VD	728
Side Chain Orientation	SW	side_chain_angle_3_SW_7	411
Side Chain Orientation	VD	neighbors_side_chain_angle_3_VD	733
Solvation	VD	solvation_4_VD	1010
Sponge	SW	sponge_CA_6_IFR_SW_5	408
Sponge	SW	sponge_CA_7_SW_5	1016
Weighted Contact Number	GN	avg_weighted_contact_number_k_5_GN	732
Weighted Contact Number	SW	weighted_contact_number_SW_9	733
Weighted Contact Number	SW	avg_weighted_contact_number_k_3_SW_9	1025

Table B.3: The 30 input features used in STING_CRISPR. Features are grouped by descriptor classes (in alphabetical order), and subsequently sorted in ascending CRISPR-Cas9 residue numbers.

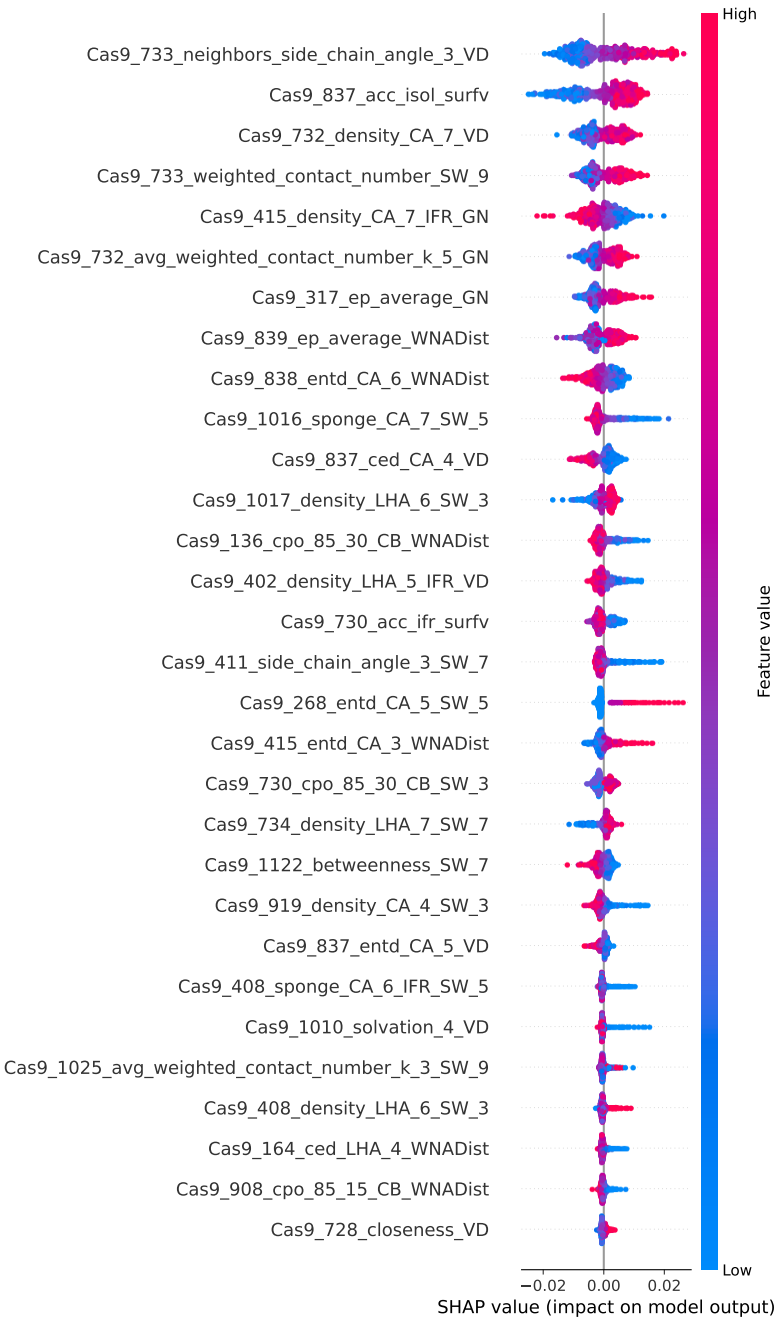


Figure B.2: SHAP summary plot for the 30 input features in STING_CRISPR for all 672 PDB snapshots.

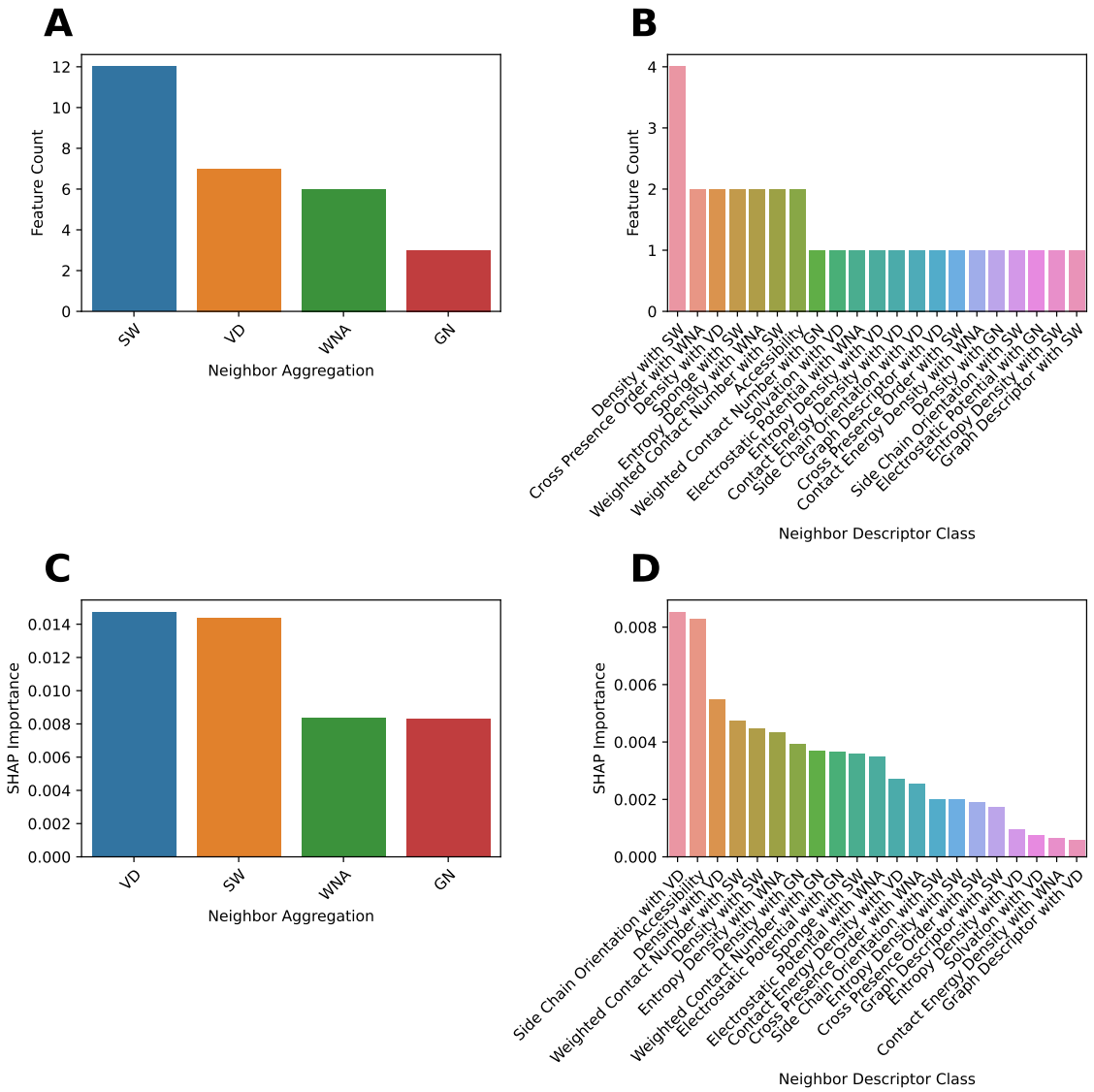


Figure B.3: (A, B) The number of features in STING_CRISPR present for each neighbor aggregation method (A) and descriptor class (B), sorted by decreasing feature counts. (Bottom) SHAP importance of neighbor aggregation methods (C) and descriptor classes (D) in STING_CRISPR, sorted by descending SHAP importance.

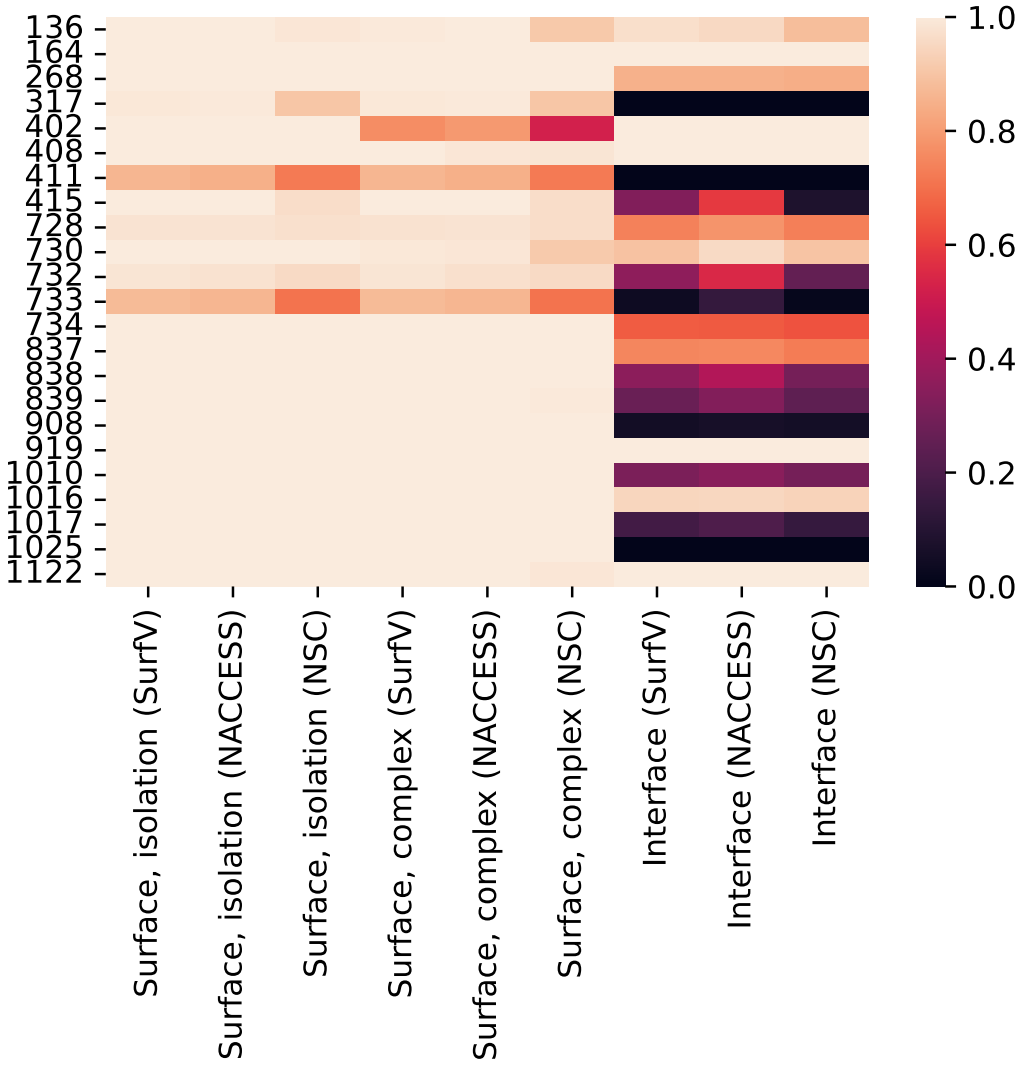


Figure B.4: Fraction of the 672 snapshots where a residue in STING_CRISPR is a surface residue in isolation, a surface residue in complex, or is on the interface when accessibility is defined by either SurfV [1], NACCESS [2] or NSC [3].

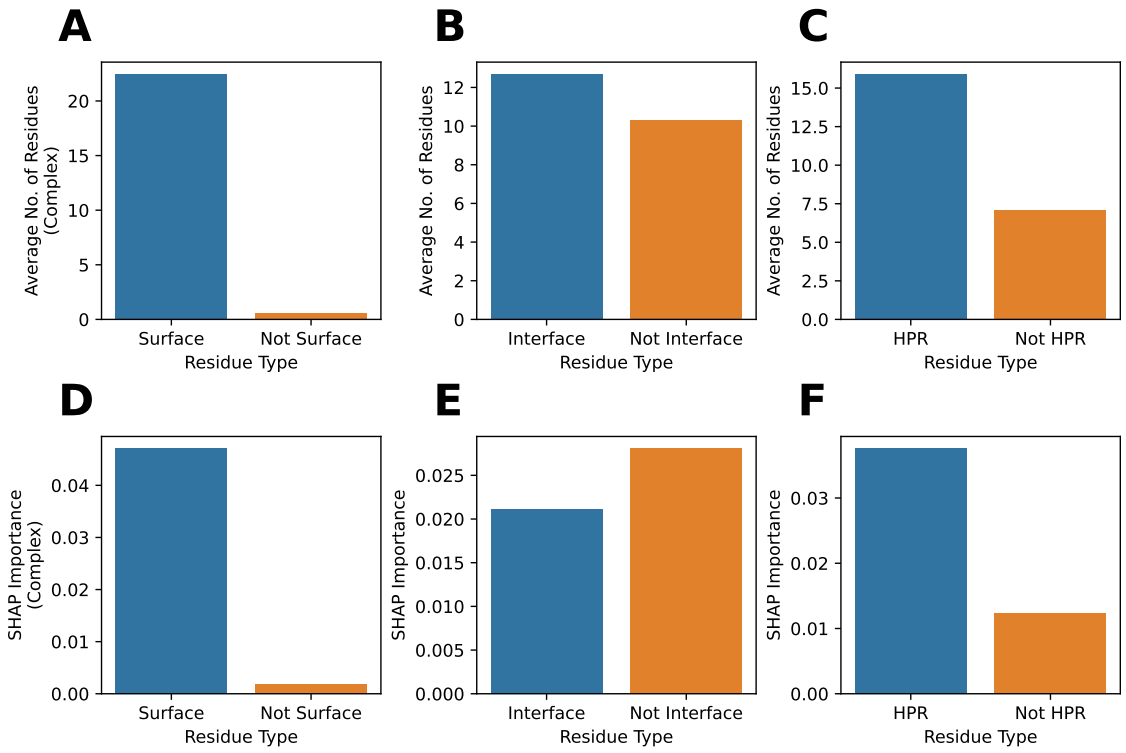


Figure B.5: Average number of residues (top) and SHAP importances (bottom) of surface vs. non-surface residues in complex (left), interface vs. non-interface residues (middle), and heteroduplex-proximal residues vs. non-heteroduplex-proximal residues (right). Plots A and C use SurfV [1] to determine whether a residue is on the complex's surface or the interface. Results similar plots A and C are obtained when using other tools (NACCESS [2] and NSC [3]) for measuring solvent accessible area, and/or when categorizing by surface vs. non-surface residues in isolation.

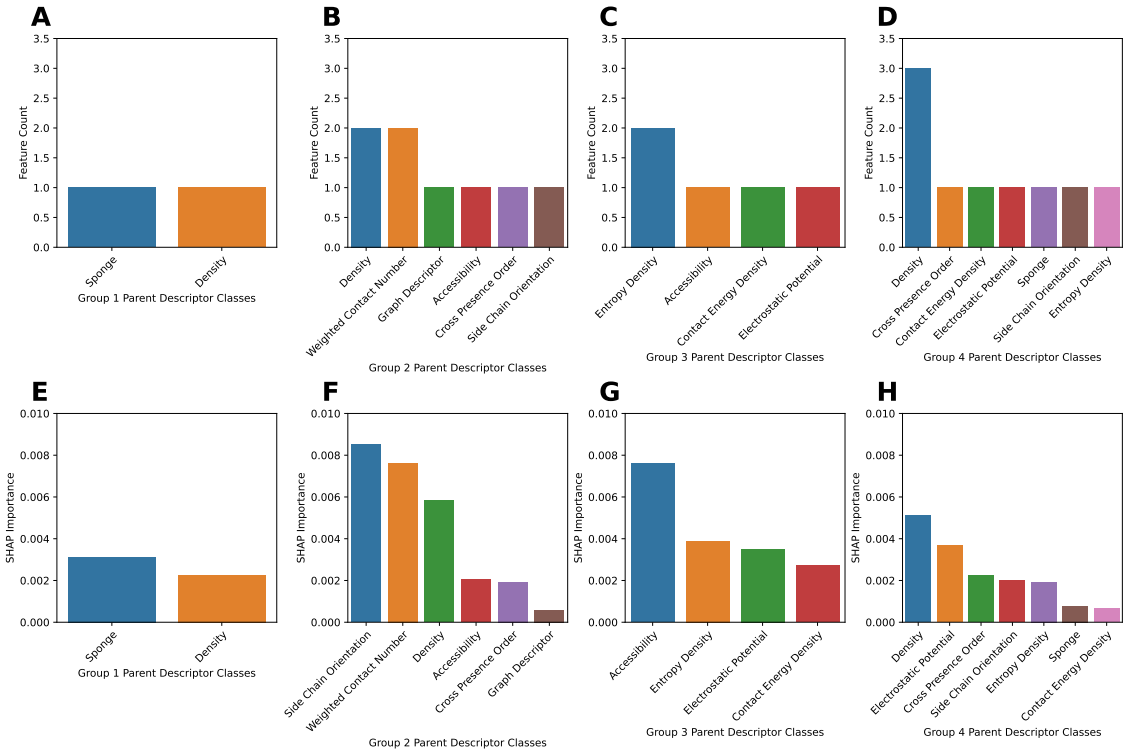


Figure B.6: Analysis of 4 residue groups (group 1: 1016, 1017; group 2: 728, 730, 732-734; group 3: 837-839; group 4: 136, 164, 317, 402, 408, 411, 415) and other residues (908, 919, 268, 1122, 1010 and 1025) identified by STING_CRISPR. (Top) Feature counts (left) and SHAP importance (right) of the 4 residue groups and other residues. (Middle) Feature counts of parent descriptor classes for the 4 residue groups. (Bottom) SHAP importance of parent descriptor classes for the 4 residue groups.

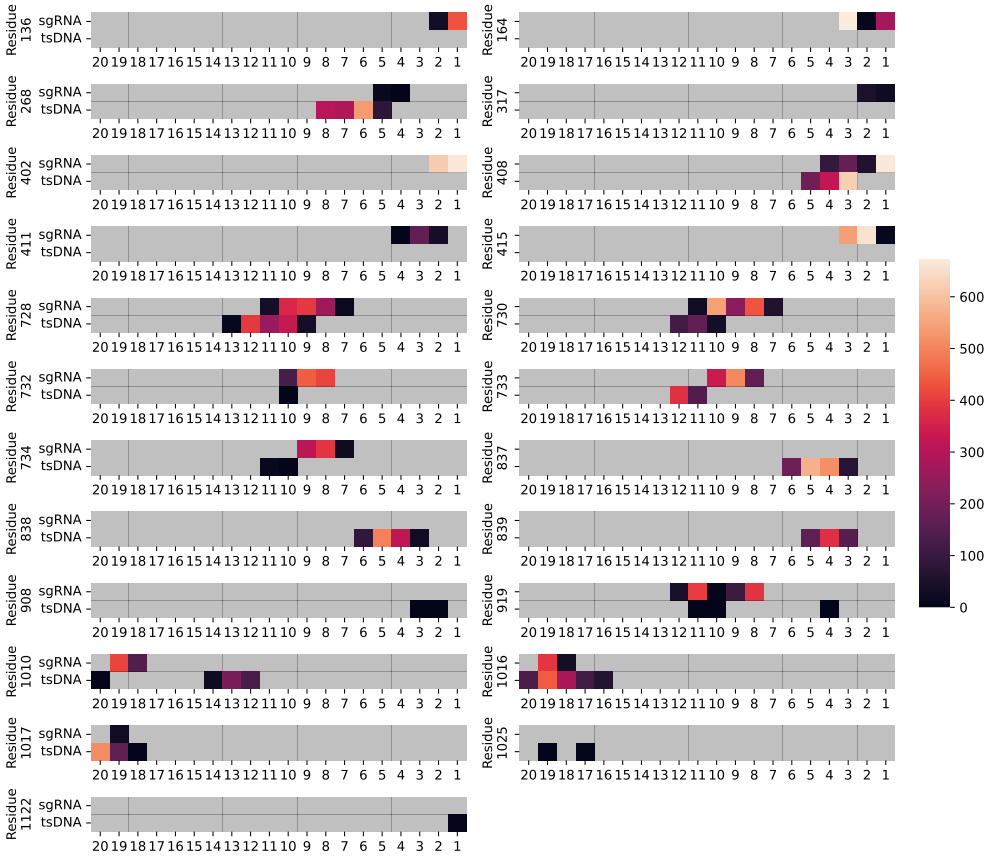


Figure B.7: Number of PDB snapshots where a specific amino acid residue has its α-carbon atom $3-7$Å away from a specified sgRNA or TS nucleotide's C4' atom, for the 20 CRISPR-Cas9 residues in STING_CRISPR. The maximum count for a given heatmap cell is 672. Grey cells indicate a count of zero.

B.2.5 sgRNA-target DNA strand heteroduplex stability

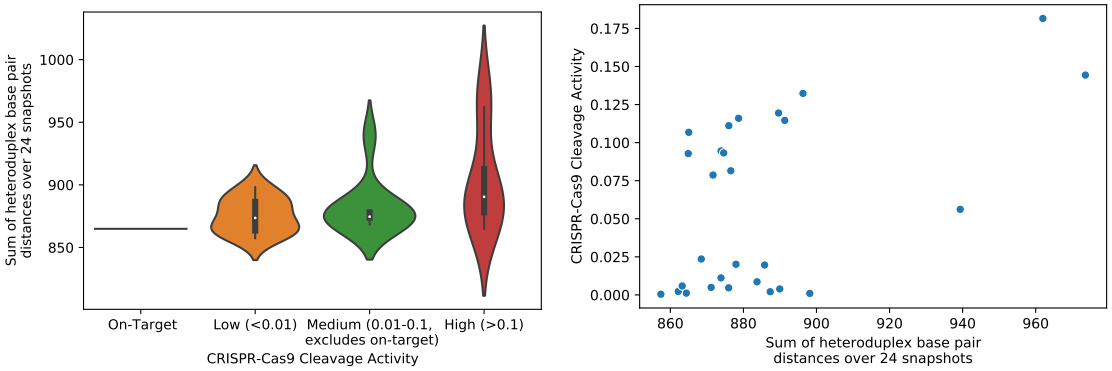


Figure B.8: (Left) Sum of the 19 PAM-proximal base pair distances in the sgRNA-TS heteroduplex for the 28 guide-target interfaces, categorized by activity level. On-target refers to the MD trajectory with no base pair mismatches in the heteroduplex. Low, medium and high refer to MD trajectories with CRISPR-Cas9 off-target cleavage activity < 0.01, $0.01 - 0.1$ (excluding on-target interface), and > 0.1, respectively. (Right) Scatter plot of the sum of the 19 heteroduplex base pair distances in the sgRNA-TS heteroduplex versus CRISPR-Cas9 off-target cleavage activity levels, with Spearman and Pearson correlations 0.418 and 0.503, respectively.

B.2.6 Holding out trajectories as test sets

Test sets for each fold of the five-fold cross validation was constructed by binning snapshots associated with the trajectory with the nth lowest cleavage activity in to the test partition of fold $n \mod 5$, and into the training partition in the other folds. We then used these train-test splits to train linear, ridge, XGBoost, extra trees, and LightGBM regression models, all using default parameters. After model training, we compared the distribution of test squared errors between trained LightGBM models in each of the 5 folds and STING-CRISPR on the last four snapshots of the 5-fold test trajectories.

Appendix C

Supplementary materials for 'DeepEmbCas9: Cas9 coevolution and sgRNA structural information for CRISPR-Cas9 cleavage activity prediction"

C.1 Supplementary methods

C.1.1 Model comparisons

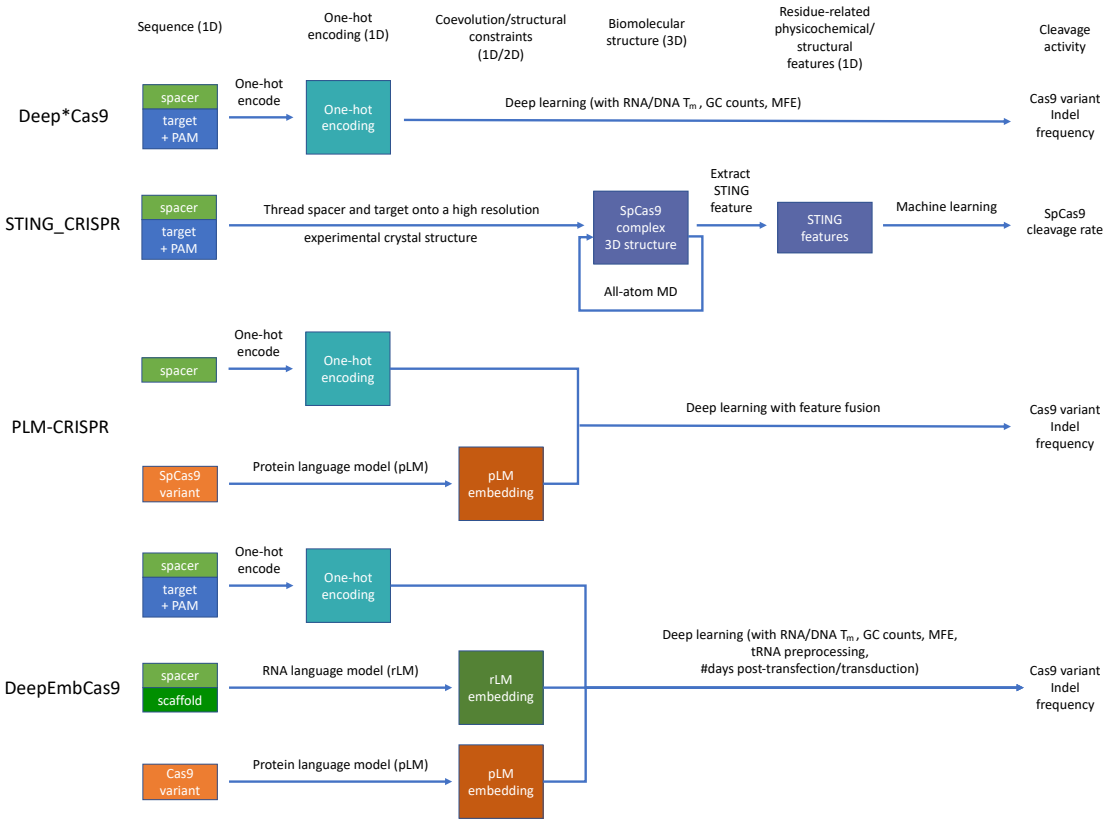


Figure C.1: ML/DL model comparison between individual Cas9 cleavage activity tools (i.e., DeepSpCas9 [5], DeepHF [4], DeepxCas9 [6], DeepSpCas9-NG [6], DeepSpCas9variants [7], DeepSmallCas9 [8], DeepSniper [10], DeepCas9variants [9]), STING_CRISPR, PLM-CRISPR and DeepEmbCas9. A single ML/DL model (DeepEmbCas9) is built for 40 Cas9 variants, while ML/DL-based individual Cas9 cleavage activity models (Deep*Cas9) are built for each. PLM_CRISPR only considers 7 SpCas9 variants (including wild type SpCas9). Owing to limited computational resources, STING_CRISPR can only predict wild type SpCas9 cleavage activity for an extremely limited set of guide-target interfaces.

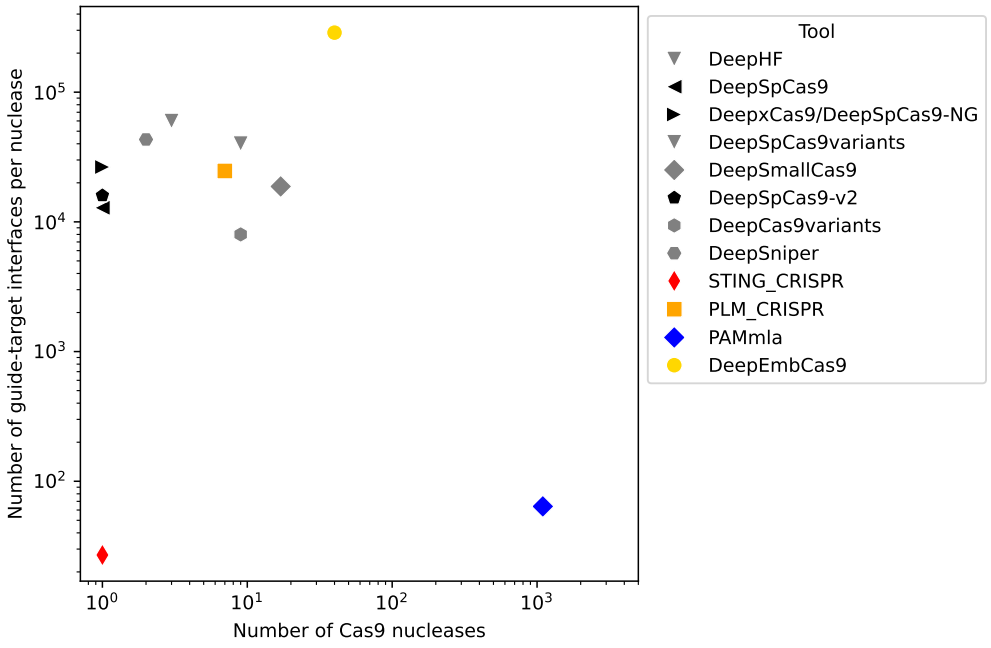


Figure C.2: Number of Cas9 nucleases and average number of guide-target interfaces per nuclease used for training DeepHF, DeepSpCas9, DeepxCas9, DeepSpCas9-NG, DeepSPCas9variants, DeepSmallCas9, DeepSpCas9-v2, DeepCas9variants, DeepSniper, STING_CRISPR, PLM_CRISPR, PAMmla and DeepEmbCas9.

C.1.2 Dataset

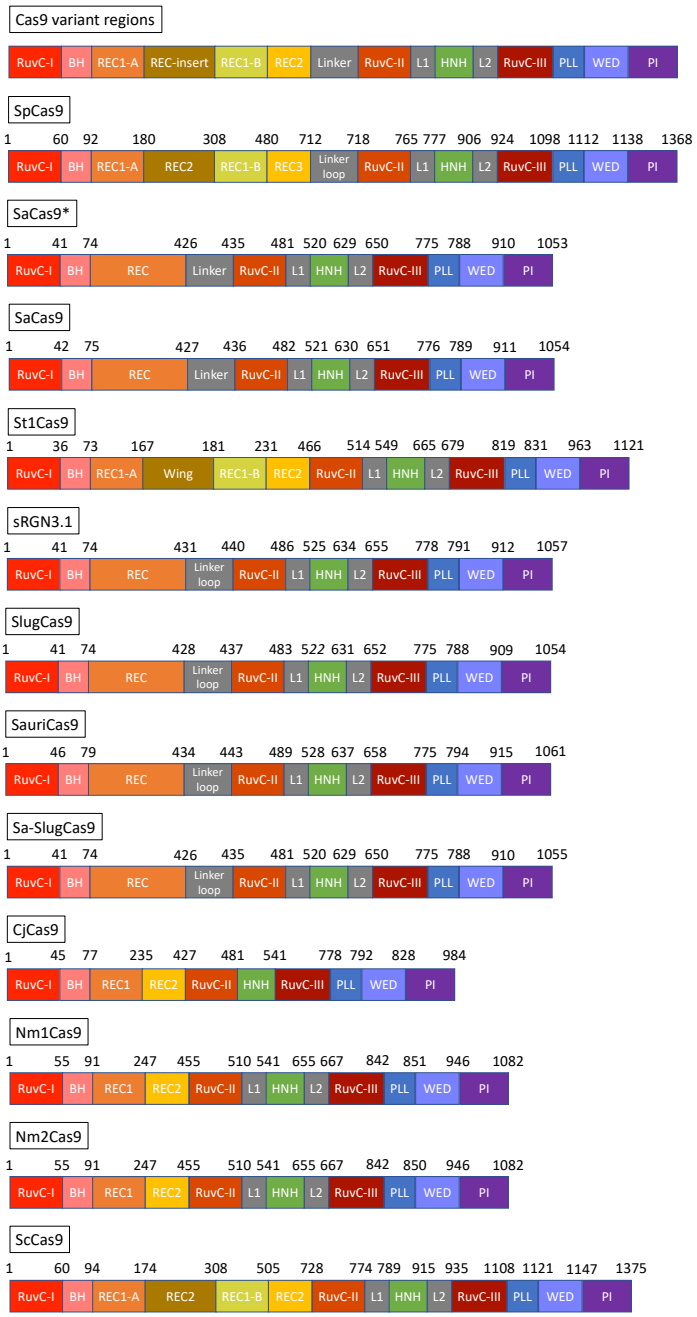


Figure C.3: Cas9 regions used for partitioning Cas9 variants. Top row shows the list of Cas9 regions, and the subsequent rows show the partitioning for the base Cas9 nucleases SpCas9, SaCas9*, SaCas9, St1Cas9, sRGN3.1, SlugCas9, SauriCas9, Sa-SlugCas9, CjCas9, Nm1Cas9, Nm2Cas9 and ScCas9. Identical partitionings are used for Cas9 variants with the same base nuclease.

Dataset column	Description
Study	Publication(s) where the datapoint(s) were collected
Library	The oligonucleotide pool containing the datapoint(s)
Table	Supplementary table/data in the study containing the datapoint(s)
Sheet	Datasheet name within the Excel file containing the datapoint(s)
src_idx	Row number within the Excel datasheet
n_data	Total number of biological/technical replicates across all studies
Partition	Training or test datapoint(s)
Partition (source)	Data partition(s) for each indel frequency label listed in "Background subtracted indel frequencies (%)"
Barcode	Barcode(s) for each experiment in each study
Spacer sequence (raw)	The spacer sequence without N-padding (typically $\geq$ 20nt)
Target context sequence (raw)	The target sequence with 5' upstream and 3' downstream context without N-padding (typically $\geq$ 30nt)
Spacer sequence	The spacer sequence with N-padding (40nt)
Target context sequence	The target sequence with 5' upstream and 3' downstream context, with N-padding (40nt)
Variant	Name of the Cas9 nuclease with nuclear localization signal (NLS), FLAG tag and P2A peptide
Nuclease	Name of the Cas9 nuclease only
gRNA scaffold	Name of the guide RNA scaffold
Day	The number of days post-transfection/transduction prior to genomic DNA isolation from edited cells
tRNA feature	Binary feature indicating the use of a $tRNA^{Gln}$-N20 sgRNA
Background subtracted indel frequencies (%)	Sets of indel frequencies for each study listed in "Study"
Mean background subtracted indel frequency (source, %)	Replicate-weighted mean of indel frequencies for each study
Mean background subtracted indel frequency (%)	Replicate-weighted mean indel frequency

Table C.1: List of column names and descriptions for the curated CRISPR-Cas9 indel frequency dataset.

No. of mismatches	No. of datapoints
0	782201
1	858316
2	71367
3	33802
4	1000

Table C.2: Number of guide-target mismatches in the Cas9 variant indel frequency dataset.

gRNA scaffold	Repeat-antirepeat length	tracrRNA (excluding antirepeat) length	poly(U) tail length
SpCas9 scaffold 1	30	46	6
SpCas9 scaffold 1 (5T)	30	46	5
SpCas9 scaffold 2	40	46	6
SaCas9 scaffold 1	34	42	6
SaCas9 scaffold 2	42	42	6
SaCas9 scaffold 3	34	42	6
NmCas9 scaffold 1	52	69	6
NmCas9 scaffold 2	40	61	6
NmCas9 scaffold 3	36	61	6
CjCas9 scaffold 1	28	45	6
CjCas9 scaffold 2	26	45	6
St1Cas9 scaffold 1	79	46	6
St1Cas9 scaffold 2	79	46	6
St1Cas9 scaffold 3	37	46	6
St1Cas9 scaffold 4	32	46	6
St1Cas9 scaffold 5	34	46	6

Table C.3: Length of sgRNA regions for the 16 gRNA scaffolds in this study.

C.1.3 Protein sequences

Codon sequences for SpCas9-NLS-FLAG-P2A (Addgene, #52962), eSpCas9(1.1)-NLS-FLAG-P2A (Addgene, #138555), SpCas9-HF1-NLS-FLAG-P2A (Addgene, #138556), HypaCas9-NLS-FLAG-P2A (Addgene, #138557), evoCas9-NLS-FLAG-P2A (Addgene, #138558), xCas9-NLS-FLAG-P2A (Addgene, #138565), Sniper-Cas9-NLS-FLAG-P2A (Addgene, #138559), VQR-NLS-FLAG-P2A (Addgene, #138560), VRER-NLS-FLAG-P2A (Addgene, #138561), VRQR-NLS-FLAG-P2A (Addgene, #138562), VRQR-HF1-NLS-FLAG-P2A (Addgene, #138563), QQR1-NLS-FLAG-P2A (Addgene, #138564), SpCas9-NG-NLS-FLAG-P2A (Addgene, #138566), Sniper2L-NLS-FLAG-P2A (Addgene, #193856), Sniper2P-NLS-FLAG-P2A (Addgene, #193857) were obtained from their respective Addgene plasmids. Codon sequences for the small Cas9 variants (including NLS-SpCas9-NLS-FLAG-P2A) in Seo et al. [8] were obtained from Supplementary Note 1 of the study, and codon sequences from nucleases in Kim, Choi et al. [9] were computationally derived from the gBlocks Gene Fragments and PCR primers listed in Supplementary Table 9 of the study. Biopython [247] was then used to computationally derive protein sequences from codon sequences. The protein sequence of NLS-SaCas9*-NLS-FLAG was derived from NLS-SaCas9-NLS-FLAG by removing the glycine residue located at the start of SaCas9's RuvC-I subdomain.

Nuclease	Base nuclease	Mutations	Primary PAM
SpCas9	SpCas9	WT	NGG
eSpCas9(1.1)	SpCas9	K848A/K1003A/R1060A	NGG
SpCas9-HF1	SpCas9	N497A/R661A/Q695A/Q926A	NGG
HypaCas9	SpCas9	N692A/M694A/Q695A/H698A	NGG
evoCas9	SpCas9	M495V/Y515N/K526E/R661Q	NGG
xCas9	SpCas9	A262T/R324L/S409I/E480K/E543D/M694I/E1219V	NG, GAA, GAT
Sniper-Cas9	SpCas9	F539S/M763I/K890N	NGG
VQR	SpCas9	D1135V/R1335Q/T1337R	NGAN, NGCG
VRER	SpCas9	D1135V/G1218R/R1335E/T1337R	NGCG
VRQR	SpCas9	D1135V/G1218R/R1335Q/T1337R	NGAH
VRQR-HF1	SpCas9	N497A/R661A/Q695A/Q926A/D1135V/G1218R/R1335Q/T1337R	NGAH
QQR1	SpCas9	G1218R/N1286Q/I1331F/D1332K/R1333Q/R1335Q/T1337R	NAAG
SpCas9-NG	SpCas9	L1111R/D1135V/G1218R/E1219F/A1322R/R1335V/T1337R	NG
sRGN3.1	sRGN3.1	WT	NNGG
SlugCas9	SlugCas9	WT	NNGG
SaCas9	SaCas9	WT	NNGRRT
SauriCas9	SauriCas9	WT	NNGG
Sa-SlugCas9	Sa-SlugCas9	WT	NNGG
SaCas9*	SaCas9*	WT	NNGRRT
SaCas9-KKH	SaCas9	E799K/N985K/R1032H	NNNRRT
eSaCas9	SaCas9	R516A/Q517A/R671A/G672A	NNGRRT
efSaCas9	SaCas9	N277D	NNGRRT
SauriCas9-KKH	SauriCas9	Q804K/Y989K/R1036H	NNRG
SlugCas9-HF	SlugCas9	R263A/N431A/T437A/R672A	NNGG
SaCas9-HF	SaCas9	R262A/N430A/N436A/R671A	NNGRRT
SaCas9-KKH-HF	SaCas9	R262A/N430A/N436A/R671A/E799K/N985K/R1032H	NNNRRT
St1Cas9	St1Cas9	WT	NNRGAA
Nm1Cas9	Nm1Cas9	WT	NNNNGATT
enCjCas9	CjCas9	L74Y/D916K	NNNVRYAC
CjCas9	CjCas9	WT	NNNVRYAC
Nm2Cas9	Nm2Cas9	WT	NNNNCCA
SpCas9-NRRH	SpCas9	I322V/S409I/E427G/R654L/R753G/R1114G/D1135N/V1139A/D1180G/E1219V/Q1221H/A1320V/R1333K	NRRH
SpCas9-NRTH	SpCas9	I322V/S409I/E427G/R654L/R753G/R1114G/D1135N/D1180G/G1218S/E1219V/Q1221H/P1249S/E1253K/P1321S/D1332G/R1335L	NRTH
SpCas9-NRCH	SpCas9	I322V/S409I/E427G/R654L/R753G/R1114G/D1135N/E1219V/D1332N/R1335Q/T1337N/S1338T/H1349R	NRCH
SpG	SpCas9	D1135L/S1136W/G1218K/E1219Q/R1335Q/T1337R	NGN
SpRY	SpCas9	A61R/L1111R/D1135L/S1136W/G1218K/E1219Q/N1317R/A1322R/R1333P/R1335Q/T1337R	NRN > NYN
Sc++	Sc++	I365A/G366D/I367K/H369L/T373S/T374G/Q379E/T1227K	NNG
Sniper2L	SpCas9	F539S/M763I/K890N/E1007L	NGG
Sniper2P	SpCas9	F539S/M763I/K890N/E1007P	NGG

Table C.4: List of 39 Cas9 nucleases considered in this study, with WT denoting wild type. All SpCas9 variants and Sc++ appear as Cas9-NLS-FLAG-P2A, and all Small Cas9s appear as NLS-Cas9-NLS-FLAG-P2A in the dataset, except for SpCas9, which appears as NLS-Cas9-NLS-FLAG-P2A in [8] and as Cas9-NLS-FLAG-P2A in all other studies, thus resulting in 40 Cas9 proteins observed in the dataset.

C.1.4 Input feature encodings

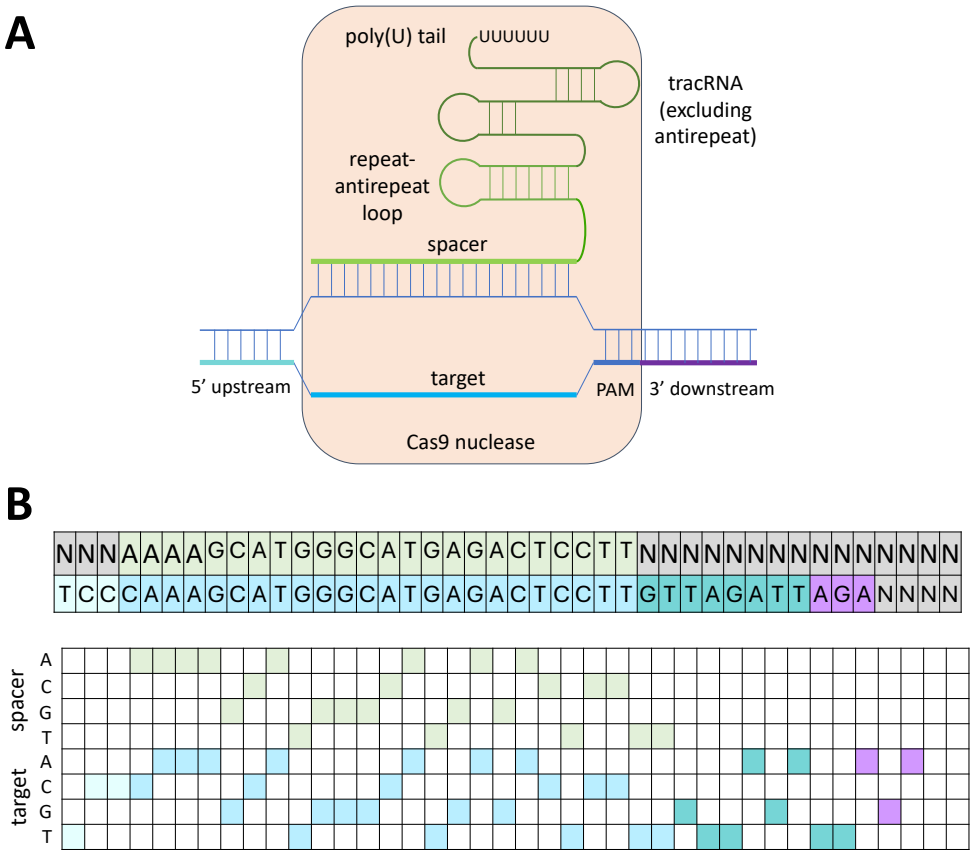


Figure C.4: (A) Schematic representation of the guide-target-Cas9 variant R-loop complex. (B) Unified guide-target interface for an example mismatched Nm1Cas9 interface (top) and its one-hot encoding (bottom).

C.1.5 Mean-variance estimation

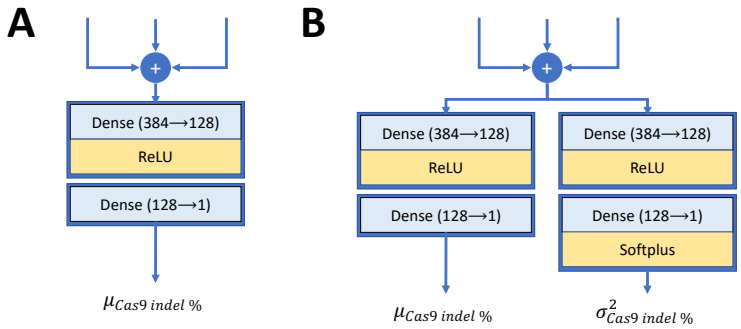


Figure C.5: (A) Deterministic (mean) output head used in DeepEmbCas9 and DeepEmbCas9_naive (B) Mean and variance output heads used in DeepEmbCas9-MVE and DeepEnsEmbCas9.

C.2 Supplementary results

C.2.1 In-distribution performance

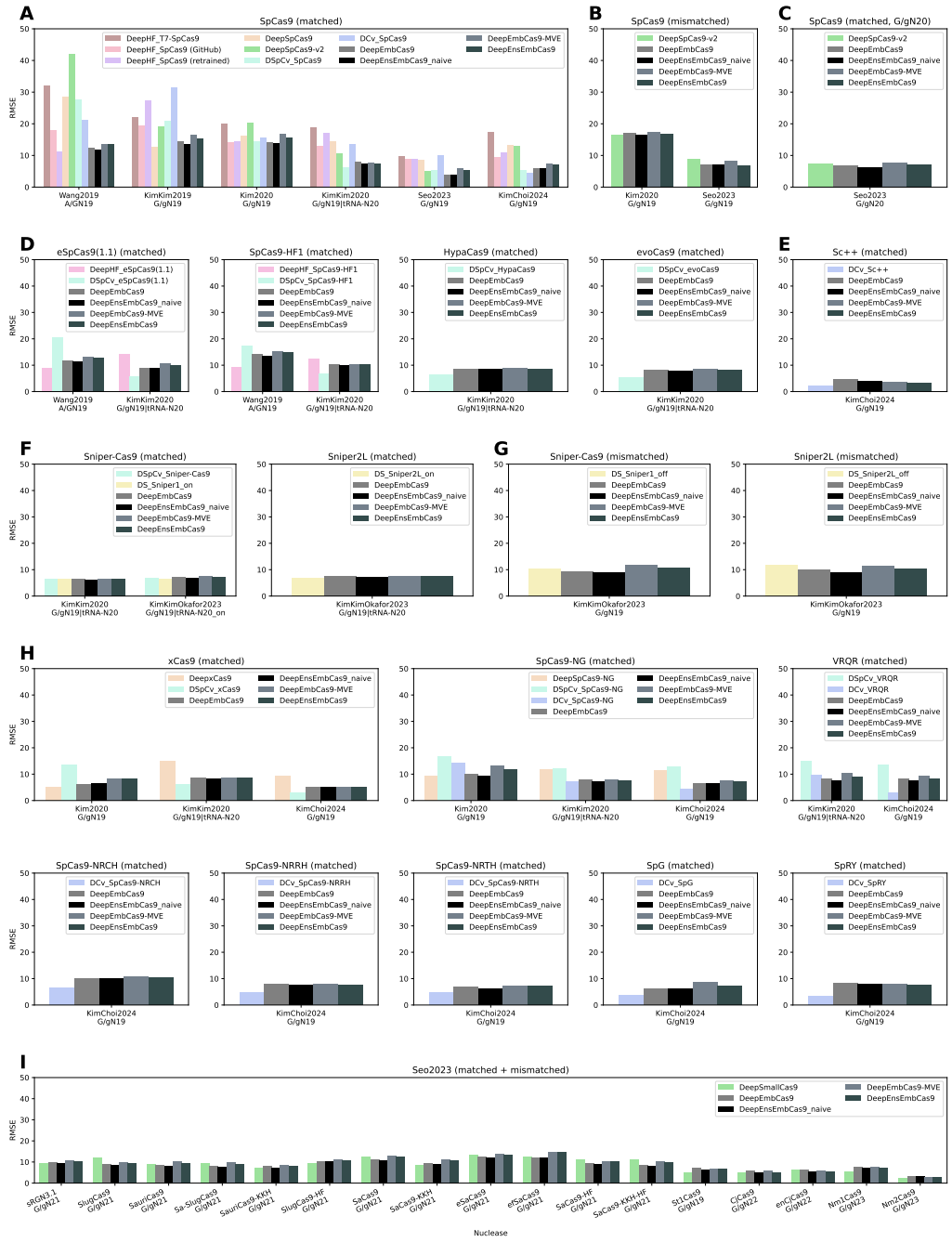


Figure C.6: Benchmark test RMSE correlation comparison for DeepEmbCas9, DeepEnsEmbCas9_naive, DeepEmbCas9-MVE and DeepEnsEmbCas9 against DeepHF, DeepSpCas9, DeepxCas9, DeepSpCas9-NG, DeepSpCas9variants, DeepSmallCas9, DeepSpCas9-v2, DeepCas9variants and DeepSniper across 39 Cas9 nucleases. The test sets consist of (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} wild type SpCas9 interfaces; (C) matched G/gN_{20} wild type SpCas9 interfaces; (D,E,F) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for 6 increased-fidelity SpCas9 variants and Sc++; (G) mismatched G/gN_{19} interfaces for Sniper variants; (H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for 8 PAM-altered SpCas9 variants; and (I) (mis)matched interfaces for 17 wild type or engineered small Cas9 nucleases.

In-distribution performance comparisons of DeepEmbCas9, DeepEnsEmbCas9 and DeepEmbCas9-MVE

Among the 51 benchmark test sets, DeepEmbCas9 attains higher Spearman correlation than all individual activity prediction tools on 10 test sets(Figure 4.4, gray bars), namely 3 mismatched G/gN$_{19}$ interface test sets for SpCas9 and Sniper-Cas9 (Figures 4.4B and G) and 7 small Cas9 test sets (2 SlugCas9 variants, 4 SaCas9 variants and SauriCas9) from Seo et al. [8] (Figure 4.4I). As for the remaining 41 test sets, DeepEmbCas9 has an average Spearman correlation drop of 4.80×10^{-2} compared to the best-performing individual activity prediction tools, with the test set containing (mis)matched G/gN$_{23}$ Nm2Cas9 interfaces from Seo et al. [8] yielding the largest Spearman drop of 0.175 (DeepSmallCas9's 0.559 vs. DeepEmbCas9 0.383). Among test sets outside of the 51 benchmark test sets which lack proper baselines, DeepEmbCas9 attains 0.451-0.917 Spearman correlation in 9 out of 10 test sets (Figures C.10 rows 2-3, C.11, C.15, C.21 row 3, C.22 row 3 and C.23 row 3).

Among the 51 benchmark test sets, DeepEnsEmbCas9 attains higher Spearman correlation than all individual activity prediction tools on 8 test sets (Figure 4.4, dark slate gray bars), which consist of the test set containing matched G/gN$_{19}$ SpCas9 interfaces from Kim et al. [6] (Figure 4.4A), 3 mismatched G/gN$_{19}$ interface test sets for SpCas9 and Sniper-Cas9 (Figures 4.4B and G), and 3 small Cas9 test sets (SlugCas9, Sa-SlugCas9 and enCjCas9) from Seo et al. [8] (Figure 4.4I). As for the remaining 43 test sets, DeepEnsEmbCas9 has an average Spearman correlation drop of 4.13×10^{-2} compared to the best-performing individual activity prediction tools, with the test set containing matched A/GN$_{19}$ SpCas9-HF1 interfaces from Wang et al. [4] yielding the largest Spearman drop of 0.194 (DeepHF_SpCas9-HF1's 0.881 and DeepEnsEmbCas9's 0.687). Among test sets outside of the 51 benchmark test sets which lack proper baselines, DeepEnsEmbCas9 attains 0.414-0.916 Spearman correlation in 9 out of 10 test sets (Figures C.10 rows 2-3, C.11, C.15, C.21 row 3, C.22 row 3 and C.23 row 3).

Among the 51 benchmark test sets, DeepEmbCas9-MVE attains higher Spearman correlation than all individual activity prediction tools on 3 test sets (Figure 4.4, slate gray bars), specifically the test set with matched G/gN$_{19}$ SpCas9 interfaces from Kim et al. [6] and 2 test sets with mismatched G/gN$_{19}$ SpCas9 interfaces (Figure 4.4A-B). As for the remaining 48 test sets, DeepEmbCas9-MVE has an average Spearman correlation drop of 5.40×10^{-2} compared to the best-performing individual activity prediction tools, with the test set containing matched G/gN$_{19}$ SpRY interfaces from Kim, Choi et al. [9] yielding the largest Spearman drop of 0.221 (DCv_SpRY's 0.934 vs. DeepEmbCas9-MVE 0.713). Among test sets outside of the 51 benchmark test sets which lack proper baselines, DeepEmbCas9-MVE attains 0.412-0.911 Spearman correlation in 9 out of 10 test sets (Figures C.10 rows 2-3, C.11, C.15, C.21 row 3, C.22 row 3 and C.23 row 3).

Detailed analysis of DeepEmbCas9's in-distribution performance

DeepEmbCas9 trained using ESM-C-600M and BEACON-B embeddings yields Spearman and RMSE metrics comparable to those of individual activity prediction tools corresponding to the test set's study on 51 benchmark test sets from the 6 studies considered (Figures 4.4 and C.6). Regarding matched SpCas9 interfaces (Figures 4.4A) with A/GN$_{19}$ sgRNAs from Wang et al. [4], DeepEmbCas9 (0.788) attain higher Spearman correlation for all SpCas9 activity prediction tools except DeepHF_SpCas9 (retrained, 0.821; GitHub, 0.797) (Figure C.7 row 1). On matched G/g$_{19}$ SpCas9 test interfaces from Kim Kim et al. [5], DeepEmbCas9 (0.692) attains higher Spearman correlation for all SpCas9 activity

prediction tools except DeepSpCas9 (0.773) and DeepHF (GitHub, 0.713; Figure C.8). On matched G/g_{19} SpCas9 test interfaces from Kim et al. [6], DeepEmbCas9 (0.905) attains similar Spearman correlation to DeepSpCas9variants (abbreviated DSpCv_SpCas9), and higher Spearman correlation for all other SpCas9 activity prediction tools (Figure C.9 row 1). On matched G/g_{19} and tRNA^{Gln}-N_{20} SpCas9 test interfaces from Kim, Kim et al. [7], DeepEmbCas9 (0.927) attains higher Spearman correlation for all SpCas9 activity prediction tools except DeepSpCas9variants (0.937; Figure C.12 row 1). On matched G/g_{19} SpCas9 test interfaces from Seo et al. [8], DeepEmbCas9 (0.710) attains higher Spearman correlation for all SpCas9 activity prediction tools except DeepCas9variants (abbreviated DCv_SpCas9, 0.776), DeepSpCas9variants (0.766), and DeepSpCas9-v2 (0.732; Figure C.17 row 1). On matched G/g_{19} SpCas9 test interfaces from Kim, Choi et al. [9], DeepEmbCas9 (0.733) attains higher Spearman correlation for all SpCas9 activity prediction tools except DeepCas9variants (0.762) and DeepSpCas9-v2 (0.744; Figure C.17 row 1).

Regarding mismatched G/gN_{19} SpCas9 test interfaces, DeepEmbCas9 (0.778 and 0.906) surpasses DeepSpCas9-v2 (0.773 and 0.874) in Spearman correlation for test datasets from Kim et al. [6] and Seo et al. [8](Figures 4.4B, C.10 row 1 and C.16 row 2). When combining matched and mismatched G/gN_{19} SpCas9 interfaces in Kim et al. [6], DeepEmbCas9 (0.889) has higher Spearman correlation than DeepSpCas9-v2 (0.862; Figure C.11 row 1). When combining matched and mismatched G/g/N_{19} SpCas9 interfaces in Seo et al. [8], DeepEmbCas9 (0.808) attains lower Spearman correlation than DeepSpCas9-v2 (0.823; Figure C.16 row 3). On matched GN_{20} SpCas9 interfaces from Seo et al. [8], DeepEmbCas9 (0.345) has lower Spearman correlation compared to DeepSpCas9-v2 (0.358; Figures 4.4C and C.16 row 4).

Next, we assess performances for 4 increased-fidelity SpCas9 variants and Sc++ (Figure 4.4D-E). Regarding eSpCas9(1.1) and SpCas9-HF1 on matched A/GN_{19} interfaces from Wang et al. [4], DeepEmbCas9 (0.849 and 0.718) attains higher Spearman correlation than DeepSpCas9variants (0.449 and 0.511) but not DeepHF (0.886 and 0.881; Figure C.7 rows 2-3). On matched G/gN_{19} and tRNA^{Gln}-N_{20} eSpCas9(1.1) and SpCas9-HF1 interfaces from Kim, Kim et al. [7], DeepEmbCas9 (0.849 and 0.790) attains higher Spearman correlation than DeepHF (0.759 and 0.725) but not DeepSpCas9variants (0.877 and 0.821; Figure C.12 rows 2-3). For HypaCas9 and evoCas9 on matched G/g_{19} interfaces from Kim, Kim et al. [7], DeepEmbCas9 attains lower Spearman correlation than DeepSpCas9variants (0.833 vs. 0.855 for HypaCas9; 0.587 vs. 0.647 for evoCas9; Figures C.13 rows 1-2). For Sc++ on matched G/g_{19} interfaces from Kim, Choi et al. [9], DeepEmbCas9 (0.573) attains lower Spearman correlation than DeepCas9variants (0.624; Figures 4.4E and C.17 row 2).

We then examine performances for Sniper-Cas9 and Sniper2L on matched (Figure 4.4F) and mismatched (Figure 4.4G) interfaces. For Sniper-Cas9, on matched G/gN_{19} and tRNA^{Gln}-N_{20} interfaces from Kim, Kim et al. [7], DeepEmbCas9 (0.931) attains similar Spearman correlation to DeepSpCas9variants (0.936) and DeepSniper (0.935; Figure C.13 row 3). On matched G/gN_{19} and tRNA^{Gln}-N_{20} Sniper-Cas9 interfaces from Kim, Kim, Okafor et al. [10], DeepEmbCas9 (0.926) also has similar Spearman correlation to DeepSpCas9variants (0.929) and DeepSniper (0.936; Figure C.21 row 1). For Sniper2L, on matched G/gN_{19} and tRNA^{Gln}-N_{20} interfaces from Kim, Kim, Okafor et al. [10], DeepEmbCas9 attains similiar Spearman correlation (0.923) to DeepSniper (0.931; Figure C.21 row 2). Combining matched and mismatched interfaces from Kim, Kim, Okafor et al. [10], DeepEmbCas9 (0.925 and 0.922) attains similar Spearman correlation to DeepSniper (0.929 and 0.924) for Sniper-Cas9 and Sniper2L, respectively (Figure C.23 rows 1-2).

We subsequently examine performances for xCas9, SpCas9-NG, and 6 other PAM-altered SpCas9 variants (Figure 4.4H). For xCas9, on matched G/gN$_{19}$ interfaces from Kim et al. [6], DeepEmbCas9 (0.884) attains lower Spearman correlation than DeepxCas9 (0.913) and DeepSpCas9variants (0.886; Figure C.9 row 2). On matched xCas9 interfaces from Kim, Kim et al. [7] and Kim, Choi et al. [9], DeepEmbCas9 (0.877 and 0.717) attains higher Spearman correlation than DeepxCas9 (0.844 and 0.704), but not for DeepSpCas9variants (0.927 and 0.740; Figure C.14 row 1 and C.18 row 1).

For SpCas9-NG, on matched G/gN$_{19}$ interfaces from Kim et al. [6], DeepEmbCas9 (0.879) attains higher Spearman correlation than DeepSpCas9variants (0.722) and DeepCas9variants (0.713), but not for DeepSpCas9-NG (0.904; Figure C.9 row 3). On matched SpCas9-NG interfaces for test datasets from Kim, Kim et al. [7] and Kim, Choi et al. [9], DeepEmbCas9 (0.890 and 0.891) attains higher Spearman correlation than DeepSpCas9-NG (0.817 and 0.868) and DeepSpCas9variants (0.799 and 0.722), but not for DeepCas9variants (0.903 and 0.935; Figure C.14 row 2 and C.18 row 2).

Moving to SpCas9-VRQR (abbreviated as VRQR), on matched VRQR interfaces for test datasets from Kim, Kim et al. [7] and Kim, Choi et al. [9], DeepEmbCas9 (0.889 and 0.742) attains higher Spearman correlation than DeepCas9variants (0.941 and 0.772), but not for DeepSpCas9variants (0.799 and 0.717; Figures C.13 row 3 and C.18 row 3). As for matched interfaces for the other 5 SpCas9 variants, DeepEmbCas9 attains lower Spearman correlation than DeepCas9variants (0.848 vs. 0.936 for SpCas9-NRCH, 0.897 vs. 0.955 for SpCas9-NRRH, 0.890 vs. 0.935 for SpCas9-NRTH, 0.862 vs. 0.903 for SpG, and 0.764 vs. 0.934 for SpRY; Figures C.19 and C.20).

As for the (mis)matched small Cas9 variant interfaces, DeepEmbCas9 attains higher Spearman correlation than DeepSmallCas9 for SlugCas9 (0.922 vs. 0.916), SauriCas9 (0.914 vs. 0.905), Sa-SlugCas9 (0.922 vs. 0.901), SaCas9 (0.904 vs. 0.901), eSaCas9 (0.860 vs. 0.854), efSaCas9 (0.865 vs. 0.860), and SaCas9-KKH-HF (0.862 vs. 0.828), but not for sRGN3.1 (0.908 vs. 0.916), SauriCas9-KKH (0.848 vs. 0.855), SlugCas9-HF (0.789 vs. 0.810), SaCas9-KKH (0.902 vs. 0.928), SaCas9-HF (0.847 vs. 0.851), St1Cas9 (0.789 vs. 0.890), CjCas9 (0.791 vs. 0.858), enCjCas9 (0.765 vs. 0.819), Nm1Cas9 (0.780 vs. 0.876), and Nm2Cas9 (0.384 vs. 0.559; Figures C.24, C.25 and C.26).

DeepEmbCas9 also attains high generalization performance on test sets without baselines. On mismatched G/gN$_{19}$ interfaces from Kim et al. [6], DeepEmbCas9 attains 0.538 and 0.779 Spearman correlation for xCas9 and SpCas9-NG, respectively (Figure C.10 rows 2-3). Combining matched and mismatched G/gN$_{19}$ interfaces from Kim et al. [6], DeepEmbCas9 attains 0.576 and 0.797 Spearman correlation for xCas9 and SpCas9-NG, respectivley (Figure C.11). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ interfaces from Kim, Kim et al. [7], DeepEmbCas9 has Spearman correlations 0.643, 0.522, 0.451 and 0.200 for VQR, VRER, VRQR-HF1 and QQR1, respectively (Figure C.15). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ and mismatched G/gN$_{19}$ Sniper2P interfaces from Kim, Kim, Okafor et al. [10], DeepEmbCas9 attains 0.917 and 0.833 Spearman correlation performance, respectively (Figures C.21 row 3 and C.22 row 3). Combining matched and mismatched Sniper2P interfaces, DeepEmbCas9 attains 0.912 Spearman correlation (Figures C.23 row 3).

C.2.2 Leave-one-nuclease-out extrapolation performance

Further leave-one-nuclease-out extrapolation performance

DeepEmbCas9_omit attains higher Spearman correlation than all individual activity prediction tools on 11 out of 48 benchmark test sets, namely 1 mismatched SpCas9 interface

test set from Kim et al. [6] (Figure C.9 row 1), 2 matched xCas9 interface test sets from Kim et al. [6] and Kim, Kim et al. [7] (Figures C.9 row 2 and C.14 row 1), 2 matched SpCas9-NRCH and SpCas9-NRTH interface test sets from Kim, Choi et al. [9] (Figure C.19 rows 1 and 3), 1 mismatched Sniper2L interface test set from Kim, Kim, Okafor et al. [10], and and 5 small Cas9 test sets (3 SaCas9 variants, SauriCas9-KKH, and SlugCas9-HF) from Seo et al. [8] (Figures C.24-C.26). As for the remaining 37 test sets, DeepEmbCas9_omit has an average Spearman performance drop of 6.65×10^{-2} compared to the best-performing individual activity prediction tools not trained on the test sets' nucleases, with the test set containing matched G/gN$_{19}$ Sc++ interfaces from Kim, Choi et al. [9] yielding the largest Spearman drop of 0.275 (DSpCv_Sniper-Cas9's 0.554 vs. DeepEmbCas9_omit's 0.279; Figure C.17 row 2).

Among the test sets with extrapolation baselines and not in the 51 benchmark test sets (excluding Kim, Kim et al. [7]'s QQR1 test set), DeepEmbCas9_omit outperforms all individual activity prediction tools on 4 out of 14 test sets, namely 2 mismatched xCas9 and SpCas9-NG interface test sets from Kim et al. [6] (Figure C.10 rows 2-3) and 2 (mis)matched xCas9 and SpCas9-NG interface test sets from Kim et al. [6] (Figure C.11 rows 2-3). As for the remaining 10 test sets, DeepEmbCas9_omit has an average Spearman performance drop of 1.78×10^{-2} compared to the best-performing individual activity prediction tools, with the test set containing matched VQR G/gN$_{19}$ interfaces from Kim, Kim et al. [6] yielding the largest Spearman drop of 6.09×10^{-2} (DCv_VRQR's 0.691 vs. DeepEnsEmbCas9_naive_omit 0.630).

DeepEnsEmbCas9_omit attains higher Spearman correlation than all individual activity prediction tools on 9 out of 48 benchmark test sets, namely 2 mismatched G/gN$_{19}$ SpCas9 interface test sets (Figures C.10 row 1 and C.16 row 2), 1 matched eSpCas9(1.1) interface test set from Kim, Kim et al. [7] (Figure C.12 row 2), 2 matched xCas9 interface test sets from Kim et al. [6] and Kim, Kim et al. [7] (Figures C.9 row 2 and C.14 row 1), 1 mismatched Sniper2L interface test sets from Kim, Kim, Okafor et al. [10] (Figure C.22 rows 2), and 3 small Cas9 test sets (SaCas9, SlugCas9-HF and Nm1Cas9) from Seo et al. [8] (Figures C.24-C.26). As for the remaining 39 test sets, DeepEnsEmbCas9_omit has an average Spearman performance drop of 5.39×10^{-2} compared to the best-performing individual activity prediction tools not trained on the test sets' nucleases, with the test set containing (mis)matched G/gN$_{21}$ Sa-SlugCas9 interfaces from Seo et al. [8] yielding the largest Spearman drop of 0.206 (DeepSmallCas9_SauriCas9's 0.884 vs. DeepEnsEmbCas9_omit's 0.678; Figure C.25 row 4).

Among the test sets with extrapolation baselines and not in the 51 benchmark test sets (excluding Kim, Kim et al. [7]'s QQR1 test set), DeepEnsEmbCas9_omit outperforms all individual activity prediction tools on 3 out of 14 test sets, namely the mismatched xCas9, (mis)matched SpCas9 and (mis)matched xCas9 interface test sets from Kim et al. [6]. As for the remaining 11 test sets, DeepEnsEmbCas9_omit has an average Spearman performance drop of 0.120 compared to the best-performing individual activity prediction tools, with the test set containing mismatched G/gN$_{19}$ SpCas9-NG interfaces from Kim et al. [6] yielding the largest Spearman drop of 0.576 (DeepSpCas9-v2's 0.603 vs. DeepEnsEmbCas9_omit's 0.0274; Figure C.10 row 3).

DeepEmbCas9-MVE_omit attains higher Spearman correlation than all individual activity prediction tools on 6 out of 48 benchmark test sets, namely 1 mismatched G/gN$_{19}$ SpCas9 interface test sets from Kim et al. [6] (Figures C.10 row 1), 3 matched xCas9 interface test sets (Figures C.9 row 2, C.14 row 1 and C.18 row 1), and 2 small Cas9 test sets (SaCas9 and Nm1Cas9) from Seo et al. [8] (Figures C.24 and C.26). As for the remaining 42 test sets, DeepEmbCas9-MVE_omit has an average Spearman performance drop

of 7.11×10^{-2} compared to the best-performing individual activity prediction tools not trained on the test sets' nucleases, with the test set containing matched G/gN$_{19}$ SpCas9-NG interfaces from Kim et al. [6] yielding the largest Spearman drop of 0.506 (DCv_SpG's 0.786 vs. DeepEmbCas9-MVE_omit's 0.279; Figure C.9 row 3).

Among the test sets with extrapolation baselines and not in the 51 benchmark test sets (excluding Kim, Kim et al. [7]'s QQR1 test set), DeepEmbCas9-MVE_omit outperforms all individual activity prediction tools on 3 out of 14 test sets, namely the mismatched xCas9, (mis)matched SpCas9 and (mis)matched xCas9 interface test sets from Kim et al. [6]. As for the remaining 11 test sets, DeepEnsEmbCas9_omit has an average Spearman performance drop of 0.120 compared to the best-performing individual activity prediction tools, with the test set containing mismatched G/gN$_{19}$ SpCas9-NG interfaces from Kim et al. [6] yielding the largest Spearman drop of 0.995 (DeepSpCas9-v2's 0.603 vs. DeepEnsEmbCas9_omit's -0.392; Figure C.10 row 3).

DeepEmbCas9 extrapolates to unseen Cas9 variants

Deep(Ens)EmbCas9_omit has decent leave-Cas9-nuclease-out performance compared to benchmark test sets (Figure C.7-C.26). Similar trends are observed when using the entire dataset for train-test splits (Figure C.27 for Spearman, Figure C.28 for Pearson).

On matched A/GN$_{19}$ SpCas9 interfaces from Wang et al. [4], DeepEmbCas9_omit (0.677) attains higher Spearman correlation than all non-SpCas9 activity tools except for DeepHF_eSpCas9(1.1) (0.707) and DeepHF_SpCas9-HF1 (0.690; Figure C.7 row 1). On matched A/GN$_{19}$ SpCas9-HF1 interfaces from Wang et al. [4], DeepEmbCas9_omit (0.671) attains higher Spearman correlation than all non-SpCas9 activity tools except for DeepHF_eSpCas9(1.1) (0.769) and DeepHF_SpCas9 (GitHub) (0.680; Figure C.7 row 2). On matched A/GN$_{19}$ eSpCas9(1.1) interfaces from Wang et al. [4], DeepEmbCas9_omit (0.725) attains higher Spearman correlation than all non-SpCas9 activity tools except for DeepHF_SpCas9-HF1 (0.777) and DeepHF_SpCas9 (GitHub) (0.730; Figure C.7 row 3).

On matched G/gN$_{19}$ SpCas9 interfaces from Kim et al. [5], DeepEmbCas9_omit (0.601) attains higher Spearman correlation than all non-SpCas9 activity tools apart from DeepHF_eSpCas9(1.1) (0.693), DeepHF_SpCas9-HF1 (0.674) and DeepxCas9 (0.611; Figure C.8).

On matched G/gN$_{19}$ SpCas9 interfaces from Kim et al. [6], DeepEmbCas9_omit (0.859) attains higher Spearman correlation than all non-SpCas9 activity tools apart from DeepSniper's Sniper1_on (0.909), DeepSniper's Sniper2L_on (0.905), and DeepSpCas9variants's Sniper-Cas9 model (0.900; Figure C.9 row 1). On matched G/gN$_{19}$ xCas9 interfaces from Kim et al. [6], DeepEmbCas9_omit (0.850) attains higher Spearman correlation than all non-xCas9 activity tools (Figure C.9 row 1). On matched G/gN$_{19}$ SpCas9-NG interfaces from Kim et al. [6], DeepEmbCas9_omit (0.765) attains higher Spearman correlation than all non-SpCas9-NG activity tools except for DeepCas9variants's SpG model (0.786) and DeepxCas9 (0.772; Figure C.9 row 3).

On mismatched G/gN$_{19}$ SpCas9 interfaces from Kim et al. [6], DeepEmbCas9_omit (0.528) attains higher Spearman correlation than DeepSniper's Sniper1 (0.295) and Sniper2L (0.250) models (Figure C.10 row 1). On mismatched G/gN$_{19}$ xCas9 interfaces from Kim et al. [6], DeepEmbCas9_omit (0.558) attains higher Spearman correlation than DeepSpCas9-v2 (0.525), DeepSniper's Sniper1 model (0.368) and DeepSniper's Sniper2L model (0.335; Figure C.10 row 2). On mismatched G/gN$_{19}$ SpCas9-NG interfaces from Kim et al. [6], DeepEmbCas9_omit (0.745) attains higher Spearman correlation than DeepSpCas9-v2 (0.603), DeepSniper's Sniper1 model (0.461) and DeepSniper's Sniper2L model (0.413; Figure C.10 row 3).

Combining matched and mismatched SpCas9 interfaces from Kim et al. [6], DeepEmbCas9_omit (0.823) has lower Spearman correlation than DeepSniper's Sniper1 (0.846) and Sniper2L (0.837) models (Figure C.11 row 1). As for (mis)matched xCas9 interfaces from Kim et al.[6], DeepEmbCas9_omit (0.594) attain higher Spearman correlation than DeepSpCas9-v2 (0.550), DeepSniper's Sniper1 model (0.404) and DeepSniper's Sniper2L model (0.379; Figure C.11 row 2). Such is also the case for SpCas9-NG interfaces from Kim et al.[6], with DeepEmbCas9_omit (0.7580) surpassing DeepSpCas9-v2 (0.612), DeepSniper's Sniper1 model (0.479) and DeepSniper's Sniper2L model (0.441; Figure C.11 row 3).

On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ SpCas9 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.903) attains higher Spearman correlation than all non-SpCas9 models except for DeepSniper's Sniper1_on model (0.935), DeepSpCas9variants's Sniper-Cas9 model (0.931) and DeepSniper's Sniper2L_on model (0.931; Figure C.12 row 1). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ eSpCas9(1.1) interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.850) attains higher Spearman correlation than all non-eSpCas9(1.1) models except for DeepSniper's Sniper2L_on model (0.858) and DeepSpCas9variants's Sniper-Cas9 model (0.852; Figure C.12 row 2). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ SpCas9-HF1 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.767) attains higher Spearman correlation than all non-SpCas9-HF1 models except for DSpCv_eSpCas9(1.1) (0.829), DSpCv_HypaCas9 (0.814), DS_Sniper2L_on (0.798), DSpCv_Sniper-Cas9 (0.788), DS_Sniper1_on (0.780) and DCv_SpCas9 (0.776; Figure C.12 row 3).

On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ HypaCas9 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.833) attains higher Spearman correlation than all non-HypaCas9 models except for DeepSniper's Sniper2L_on model (0.839; Figure C.13 row 1). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ evoCas9 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.576) attains higher Spearman correlation than all non-evoCas9 models except DSpCv_eSpCas9(1.1) (0.628), DSpCv_SpCas9-HF1 (0.606), DSpCv_HypaCas9 (0.590), DeepHF_eSpCas9(1.1) (0.582), DeepHF_SpCas9-HF1 (0.581) and DS_Sniper2L_on (0.576; Figure C.13 row 2). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ Sniper-Cas9 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.920) attains higher Spearman correlation than all non-Sniper-Cas9 models except for DCv_SpCas9 (0.935), DSpCv_SpCas9 (0.935) and DS_Sniper2L_on (0.934; Figure C.13 row 3).

On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ xCas9 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.829) surpasses all non-xCas9 models in Spearman correlation (Figure C.14 row 1). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ SpCas9-NG interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.805) attains higher Spearman correlation than all non-SpCas9-NG models apart from DSpCv_VRQR (0.919), DCv_SpG (0.889) and DCv_VRQR (0.810; Figure C.14 row 2). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ VRQR interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.812) surpasses all non-VRQR models in Spearman correlation, except for DSpCv_SpCas9-NG (0.928) and DCv_SpG (0.861; Figure C.14 row 3).

On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ QQR1 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit attains 0.160 Spearman correlation (Figure C.15 row 1). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ VQR interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.630) attains higher Spearman correlation for all non-VQR models except for DCv_VRQR (0.691), DSpCv_SpCas9-NG (0.655), DCv_SpCas9-NG (0.637) and DCv_SpG (0.635; Figure C.15 row 2). On matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ VRER interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.512) attains higher Spearman correlation for all non-VRER model except for DCv_SpG (0.523) and DSpCv_SpCas9-NG (0.515; Figure C.15

row 3). On matched G/gN$_{19}$ and tRNA^{Gln}-N$_{20}$ VRQR-HF1 interfaces from Kim, Kim et al. [7], DeepEmbCas9_omit (0.421) surpasses all non-VRQR-HF1 models in Spearman correlation, apart from DSpCv_SpCas9-NG (0.455), DCv_VRQR (0.455), DCv_SpG (0.442), DSpCv_HypaCas9 (0.439), DSpCv_evoCas9 (0.435) and DCv_SpCas9-NG (0.428; Figure C.15 row 4).

On matched G/gN$_{19}$ SpCas9 interfaces from Seo et al. [8], DeepEmbCas9_omit (0.734) attains higher Spearman correlation than all non-SpCas9 models apart from DS_Sniper1_on (0.748), DS_Sniper2L_on (0.747) and DSpCv_Sniper-Cas9 (0.747; Figure C.16 row 1). On mismatched G/gN$_{19}$ SpCas9 interfaces from Seo et al. [8], DeepEmbCas9_omit (0.891) attains higher Spearman correlation than DS_Sniper2L_off (0.875), but not DS_Sniper1_off (0.892; Figure C.16 row 1). Combining matched and mismatched G/gN$_{19}$ SpCas9 interfaces from Seo et al. [8], DeepEmbCas9_omit (0.812) attains lower Spearman correlation than DS_Sniper1 (0.822) and DS_Sniper2L (0.822; Figure C.16 row 3). On matched G/gN$_{20}$ SpCas9 interfaces, DeepEmbCas9_omit has 0.184 Spearman correlation (Figure C.16 row 4).

Looking at test datasets from Kim, Choi et al. [9] with matched G/gN$_{19}$ interfaces, for SpCas9, DeepEmbCas9_omit (0.670) attains higher Spearman correlation for all non-SpCas9 models except for DSpCv_Sniper-Cas9 (0.728) and DS_Sniper2L_on (0.722; Figure C.17 row 1). For Sc++, DeepEmbCas9_omit attains 0.279 Spearman correlation (Figure C.17 row 2).

For xCas9, DeepEmbCas9_omit (0.656) attains higher Spearman correlation than all non-xCas9 models apart from DCv_SpCas9-NRRH (0.699), DCv_SpCas9-NRCH (0.696), DeepSpCas9-v2 (0.686), DCv_SpCas9-NRTH (0.670), DCv_SpCas9-NG (0.665), DCv_SpG (0.664), DSpCv_VRQR (0.663) and DCv_SpCas9 (0.659; Figure C.18 row 1). For SpCas9-NG, DeepEmbCas9_omit (0.801) attains higher Spearman correlation than all non-SpCas9-NG models apart from DSpCv_VRQR (0.927) and DCv_SpG (0.867; Figure C.18 row 2). For VRQR, DeepEmbCas9_omit (0.668) attains higher Spearman correlation than all non-VRQR models apart from DSpCv_SpCas9-NG (0.758), DCv_SpCas9-NG (0.735), DCv_SpG (0.733) and DeepSpCas9-NG (0.687; Figure C.18 row 3).

For SpCas9-NRCH/SpCas9-NRTH, DeepEmbCas9_omit (0.801/0.854) attains higher Spearman correlation than all non-SpCas9-NRCH/non-SpCas9-NRTH models (Figure C.19 rows 1 and 3). For SpCas9-NRRH, DeepEmbCas9_omit (0.840) attains higher Spearman correlation than all non-SpCas9-NRRH models except for DCv_SpCas9-NRTH (0.849; Figure C.19 row 2).

For SpG, DeepEmbCas9_omit (0.736) attains higher Spearman correlation than all non-SpG models except for DCv_SpCas9-NG (0.870), DSpCv_VRQR (0.852), DeepSpCas9-NG (0.820), DCv_VRQR (0.779), DeepxCas9 (0.741; Figure C.20 row 1). For SpRY, DeepEmbCas9_omit (0.729) attains higher Spearman correlation than all non-SpRY models except for DCv_SpCas9-NG (0.770) and DCv_SpG (0.747; Figure C.20 row 2).

We next look at test datasets from Kim, Kim, Okafor et al. [10]. On matched G/gN$_{19}$ and tRNA^{Gln}-N$_{20}$ interfaces, for Sniper-Cas9, DeepEmbCas9_omit (0.918) attains higher Spearman correlation than all non-Sniper-Cas9 models except for DSpCv_SpCas9 (0.933), DS_Sniper2L_on (0.933) and DCv_SpCas9 (0.930; Figure C.21 row 1). For Sniper2L, DeepEmbCas9_omit (0.917) attains higher Spearman correlation than all non-Sniper2L models except for DS_Sniper1_on (0.928), DSpCv_Sniper-Cas9 (0.924), DSpCv_SpCas9 (0.923) and DCv_SpCas9 (0.922; Figure C.21 row 2). For Sniper2P, DeepEmbCas9_omit (0.914) attains higher Spearman correlation than all non-Sniper2P models except for DS_Sniper1_on (0.929), DS_Sniper2L_on (0.925), DSpCv_SpCas9 (0.925), DSpCv_Sniper-Cas9 (0.917), and DCv_SpCas9 (0.916; Figure C.21 row 3).

Regarding mismatched G/gN$_{19}$ interfaces, for Sniper-Cas9, DeepEmbCas9_omit (0.876) attains higher Spearman correlation than DeepSpCas9-v2 (0.866), but not DS_Sniper2L_off (0.881; Figure C.22 row 1). For Sniper2L, DeepEmbCas9_omit (0.872) attains higher Spearman correlation than DS_Sniper1_off (0.880) and DeepSpCas9-v2 (0.848; Figure C.22 row 2). For Sniper2P, DeepEmbCas9_omit (0.862) attains higher Spearman correlation than DS_Sniper1_off (0.840) and DeepSpCas9-v2 (0.838), but not for DS_Sniper2L_off (0.865; Figure C.22 row 3).

Combining matched and mismatched interfaces from Kim, Kim, Okafor et al. [10], for Sniper-Cas9, DeepEmbCas9_omit (0.917) attains higher Spearman correlation than DeepSpCas9-v2 (0.875) but not for DS_Sniper2L (0.925; Figure C.23 row 1). For Sniper2L, DeepEmbCas9_omit (0.918) attains higher Spearman correlation than DeepSpCas9-v2 (0.868) but not for DS_Sniper1 (0.922; Figure C.23 row 2). For Sniper2P, DeepEmbCas9_omit (0.918) attains higher Spearman correlation than DeepSpCas9-v2 (0.868) but not for DS_Sniper2L (0.924) and DS_Sniper1 (0.922; Figure C.23 row 3).

We next examine small Cas9 nuclease test datasets from Seo et al. [8]. Looking at test sets with G/gN$_{21}$ interfaces, for SaCas9/efSaCas9/SaCas9-HF, DeepEmbCas9_omit (0.815/0.855/0.831) attains higher Spearman correlation than all non-SaCas9/non-efSaCas9/non-SaCas9-HF models, respectively (Figure C.24 rows 1, 3 and 4). For eSaCas9, DeepEmbCas9_omit (0.852) attains higher Spearman correlation than all non-eSaCas9 models except for DeepSmallCas9_efSaCas9 (0.858; Figure C.24 row 2). For SaCas9-KKH, DeepEmbCas9_omit (0.789) attains higher Spearman correlation than all non-SaCas9-KKH models except for DeepSmallCas9_SaCas9-KKH-HF (0.831; Figure C.24 row 5). For SaCas9-KKH-HF, DeepEmbCas9_omit (0.833) attains higher Spearman correlation than all non-SaCas9-KKH-HF models except for DeepSmallCas9_SaCas9-KKH (0.883; Figure C.24 row 6).

For sRGN3.1, DeepEmbCas9_omit (0.870) attains higher Spearman correlation than all non-sRGN3.1 models except for DeepSmallCas9_SlugCas9 (0.894) and DeepSmallCas9_SauriCas9 (0.874; Figure C.25 row 1). For SlugCas9, DeepEmbCas9_omit (0.901) attains higher Spearman correlation than all non-SlugCas9 models except for DeepSmallCas9_sRGN3.1 (0.915; Figure C.25 row 2). For SauriCas9, DeepEmbCas9_omit (0.846) attains higher Spearman correlation than all non-SauriCas9 models except for DeepSmallCas9_Sa-SlugCas9 (0.893), DeepSmallCas9_SauriCas9-KKH (0.862) and DeepSmallCas9_sRGN3.1 (0.848; Figure C.25 row 3). For Sa-SlugCas9, DeepEmbCas9_omit (0.688) attains lower Spearman correlation than all non-Sa-SlugCas9 models (Figure C.25 rows 4-6). For SauriCas9-KKH/SlugCas9-HF, DeepEmbCas9_omit (0.817/0.783) attains higher Spearman correlation than all non-SauriCas9-KKH/non-SlugCas9-HF models (Figure C.25 rows 4-6).

Looking at non-G/gN$_{21}$ nucleases, for G/gN$_{19}$ St1Cas9 interfaces, DeepEmbCas9_omit attains 0.068 Spearman correlation (Figure C.26 row 1). For G/gN$_{22}$ CjCas9 interfaces, DeepEmbCas9_omit (0.708) attains lower Spearman correlation than DeepSmallCas9_enCjCas9 (0.820; Figure C.26 row 2). For G/gN$_{22}$ enCjCas9 interfaces, DeepEmbCas9_omit (0.741) attains lower Spearman correlation than DeepSmallCas9_CjCas9 (0.887; Figure C.26 row 3). For G/gN$_{23}$ Nm1Cas9 interfaces, DeepEmbCas9_omit (-0.049) attains lower Spearman correlation than DeepSmallCas9_Nm2Cas9 (0.193; Figure C.26 row 4). For G/gN$_{22}$ and G/gN$_{23}$ Nm2Cas9 interfaces, DeepEmbCas9_omit attains -0.007 Spearman correlation (Figure C.26 row 5).

Performance is varied when leaving one gRNA scaffold out for testing (Figures C.29 and Figure C.30 for Spearman and Pearson correlations, respectively).

C.2.3 Per-nuclease in-distribution and extrapolation plots

SpCas9 variants and Sc++

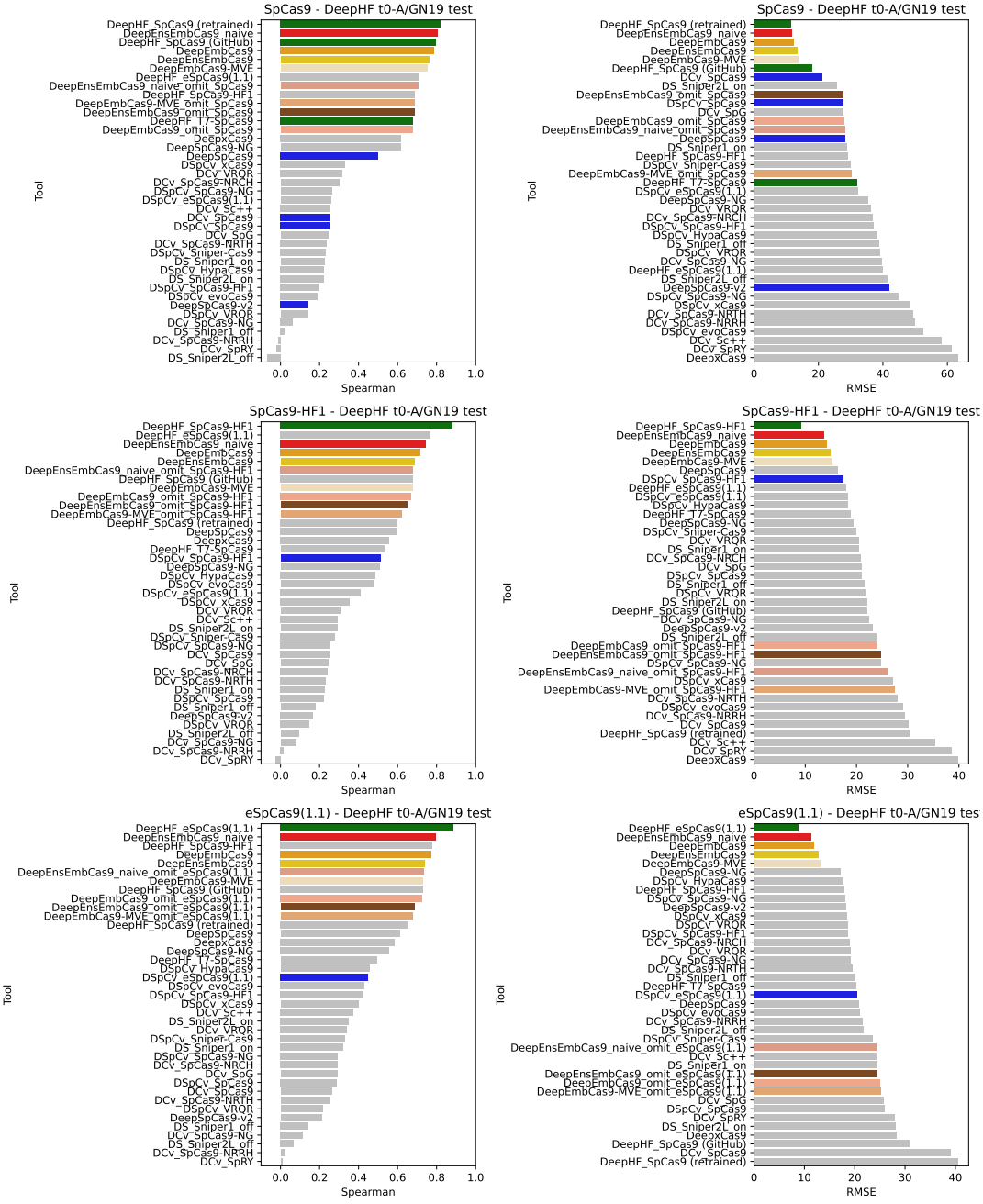


Figure C.7: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched A/GN$_{19}$ wild type SpCas9 (top), SpCas9-HF1 (middle) and eSpCas9(1.1) (bottom) interfaces from Wang et al. [4], where green and blue bars denote DeepHF and other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease, respectively.

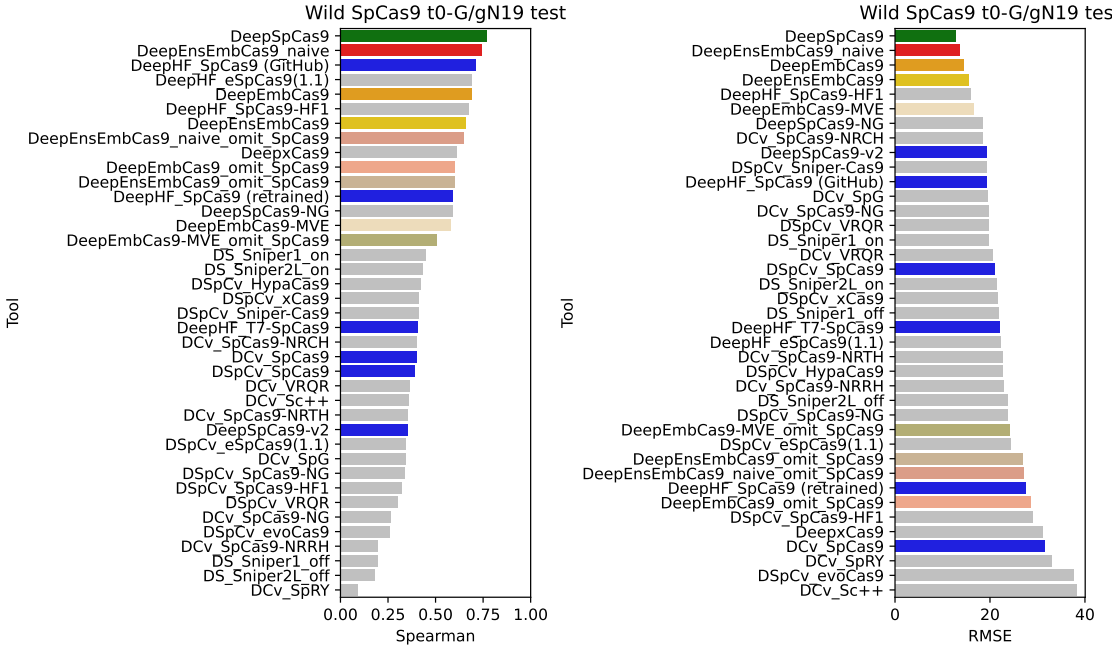


Figure C.8: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} wild type SpCas9 interfaces from Kim, Kim et al. [5], where green and blue bars denote DeepSpCas9 and other individual Cas9 cleavage activity tools trained on matched wild type SpCas9 interfaces, respectively.

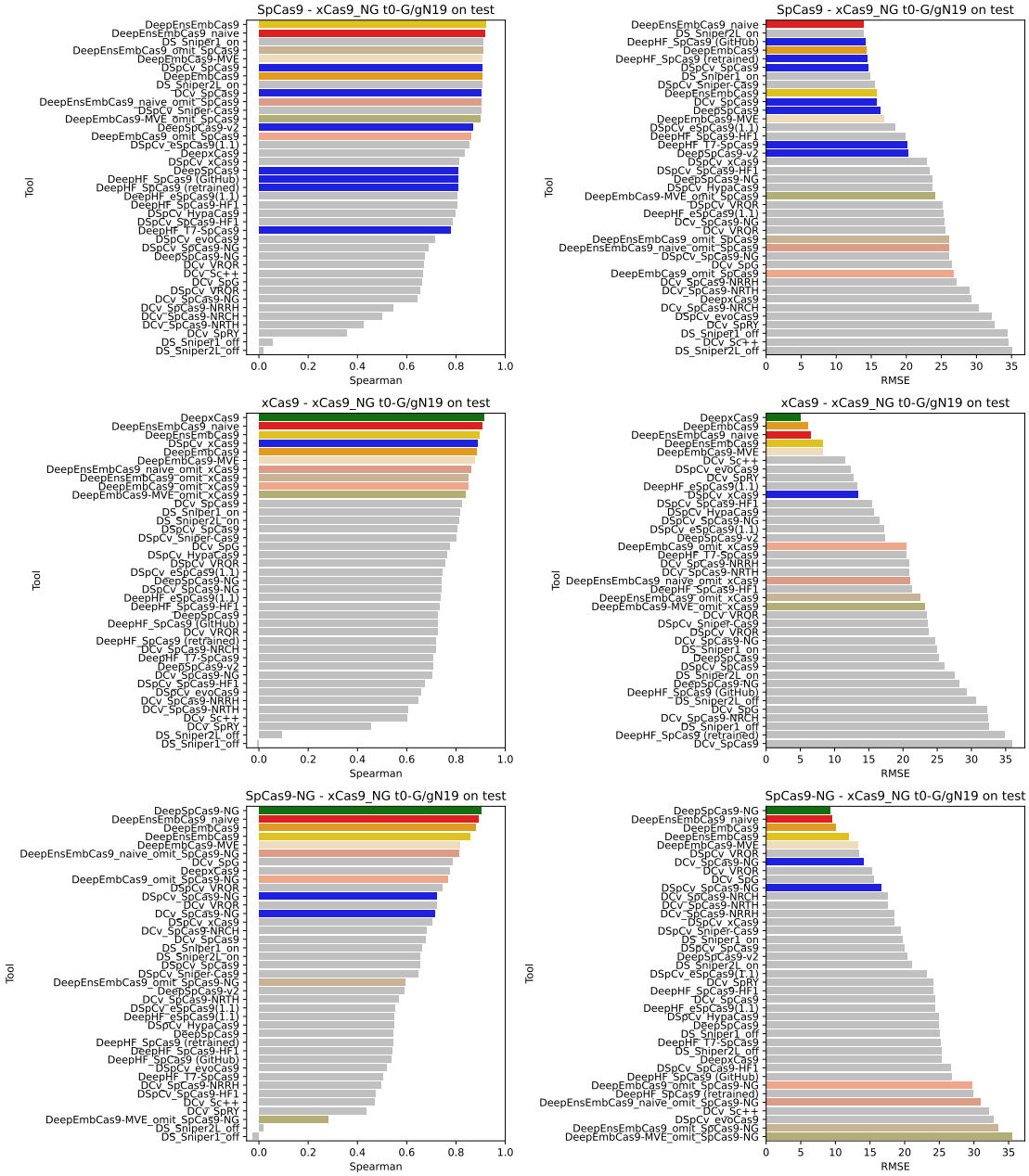


Figure C.9: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} wild type SpCas9 (top), xCas9 (middle) and SpCas9-NG (bottom) interfaces from Kim et al. [6], where green bars denote DeepxCas9 (middle row) and DeepSpCas9-NG (bottom row), and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

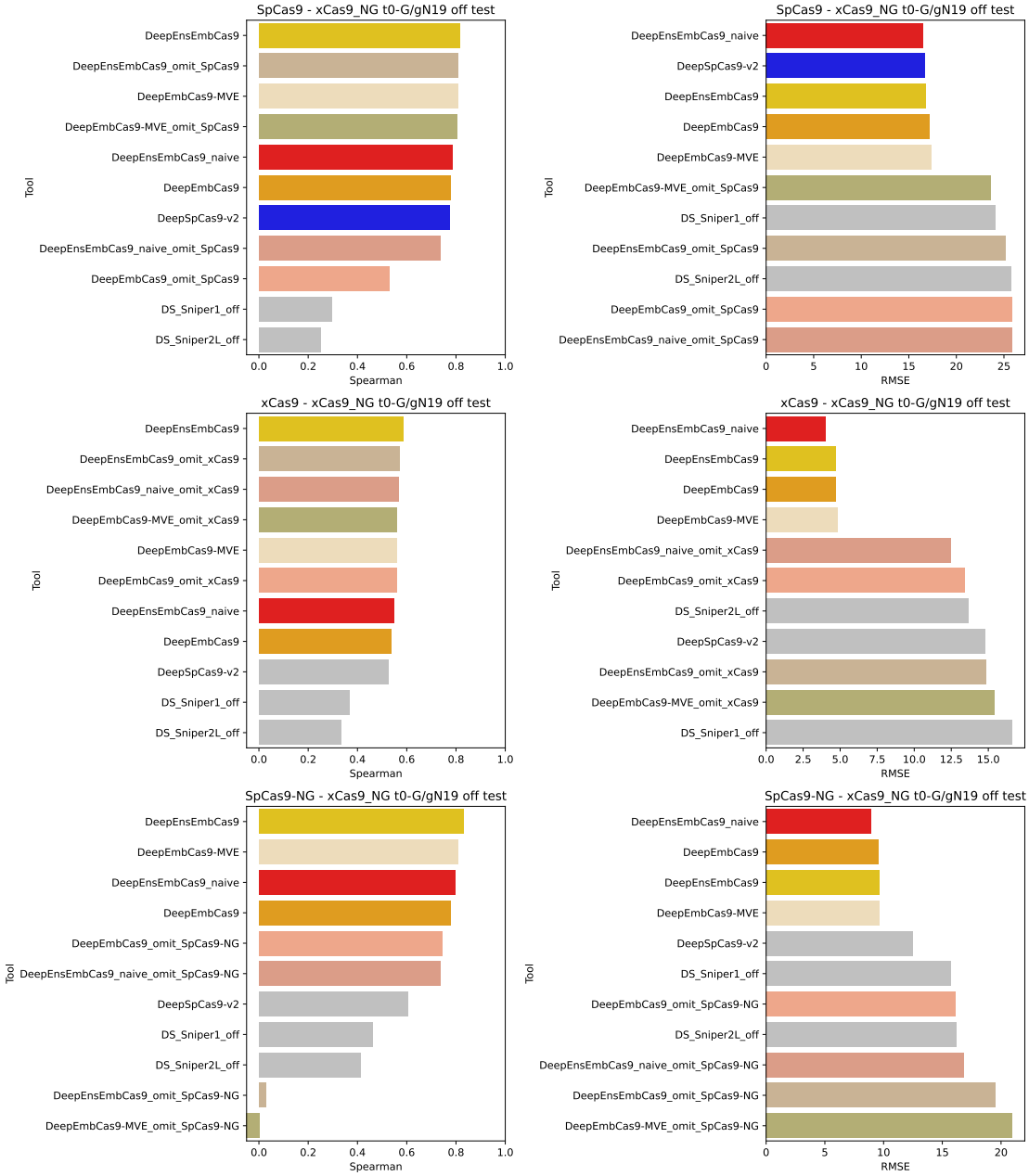


Figure C.10: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for mismatched G/gN$_{19}$ wild type SpCas9 (top), SpCas9-NG (middle) and xCas9 (bottom) interfaces from Kim et al. [6], where blue bars denote individual Cas9 cleavage activity tools trained on mismatched interfaces of the test nuclease.

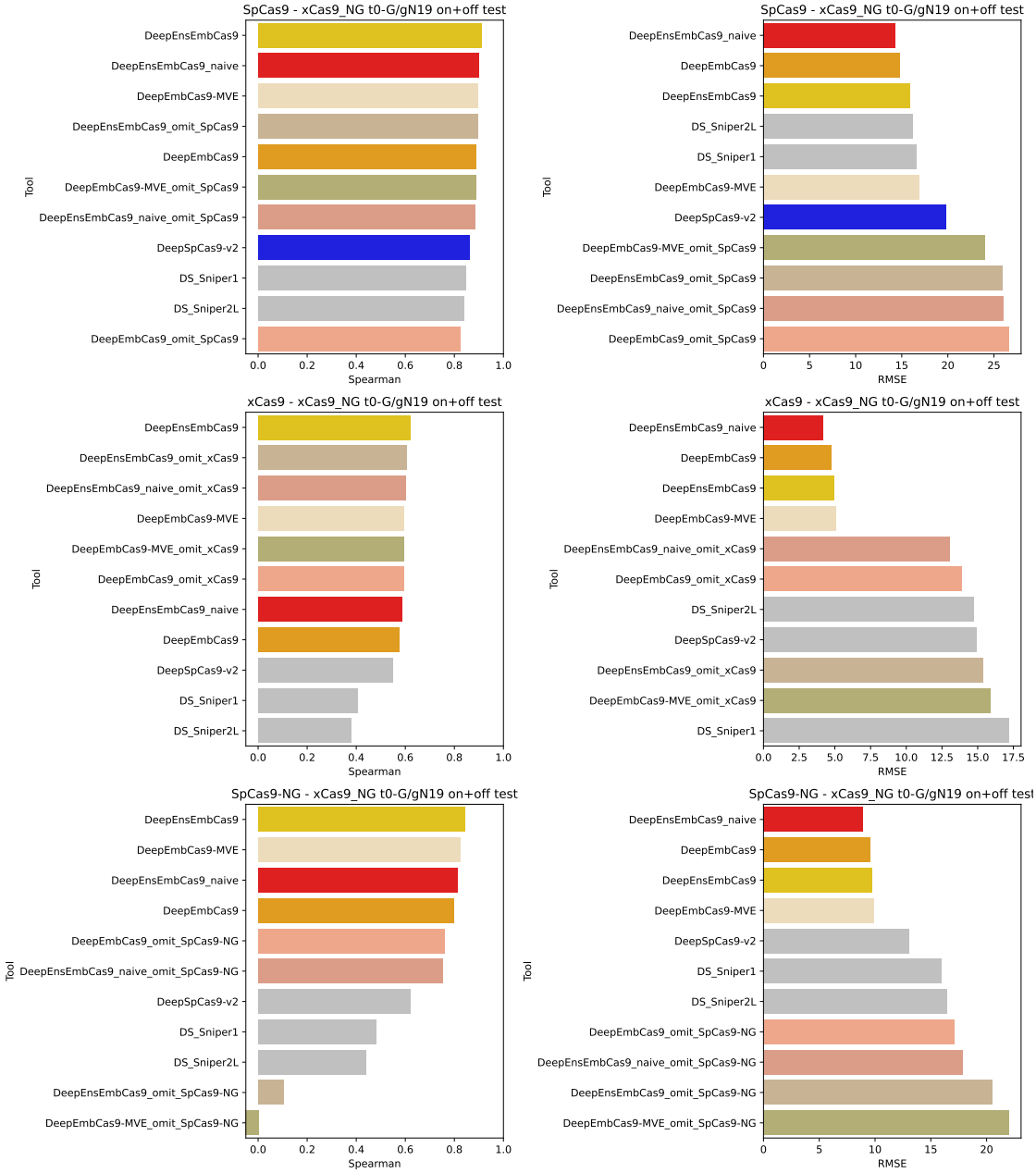


Figure C.11: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for (mis)matched G/gN$_{19}$ wild type SpCas9 (top), SpCas9-NG (middle) and xCas9 (bottom) interfaces from Kim et al. [6], where blue bars denote individual Cas9 cleavage activity tools trained on (mis)matched interfaces of the test nuclease.

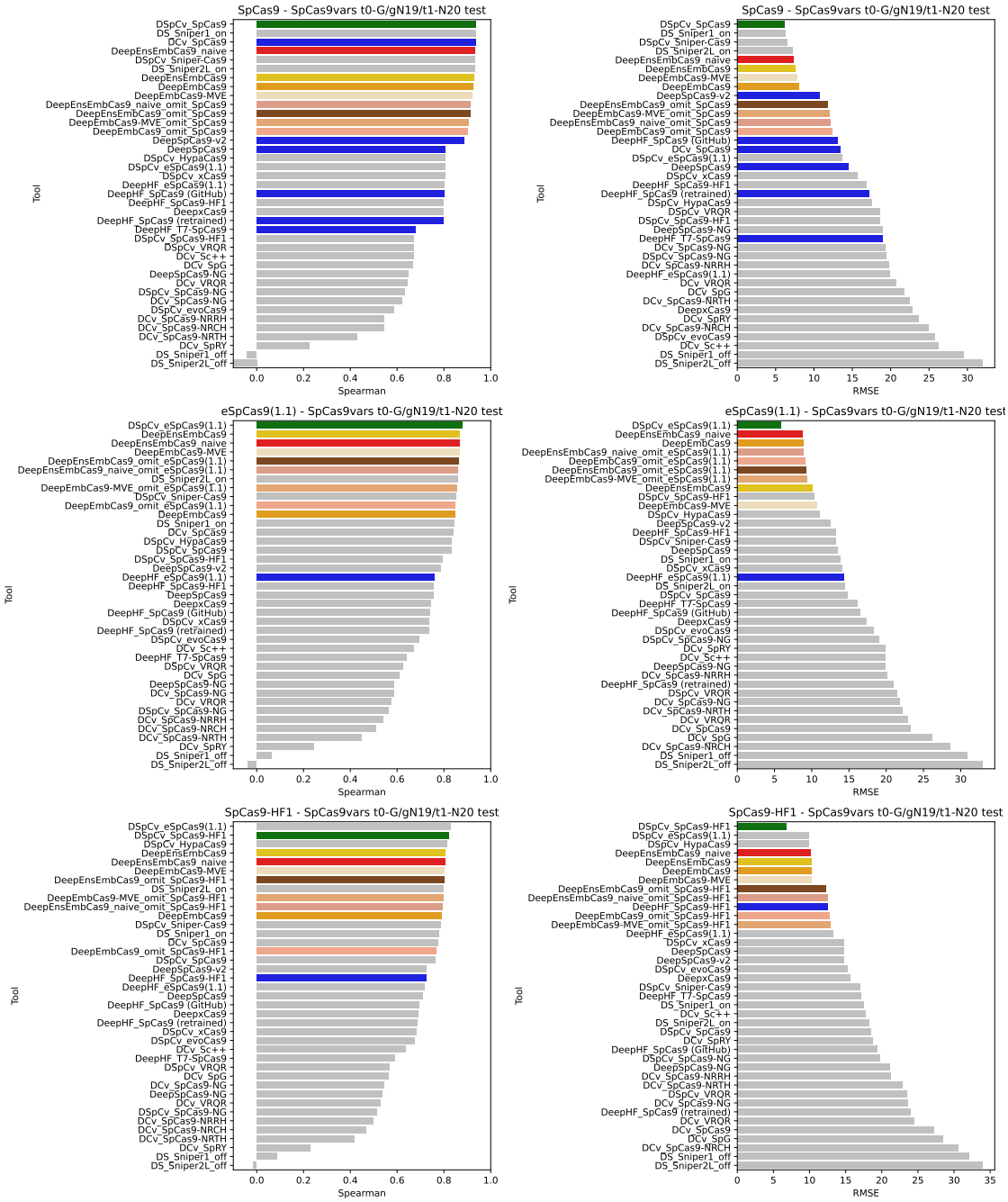


Figure C.12: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 (top), eSpCas9(1.1) (middle) and SpCas9-HF1 (bottom) interfaces from Kim, Kim et al. [7], where green bars denote DeepSpCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

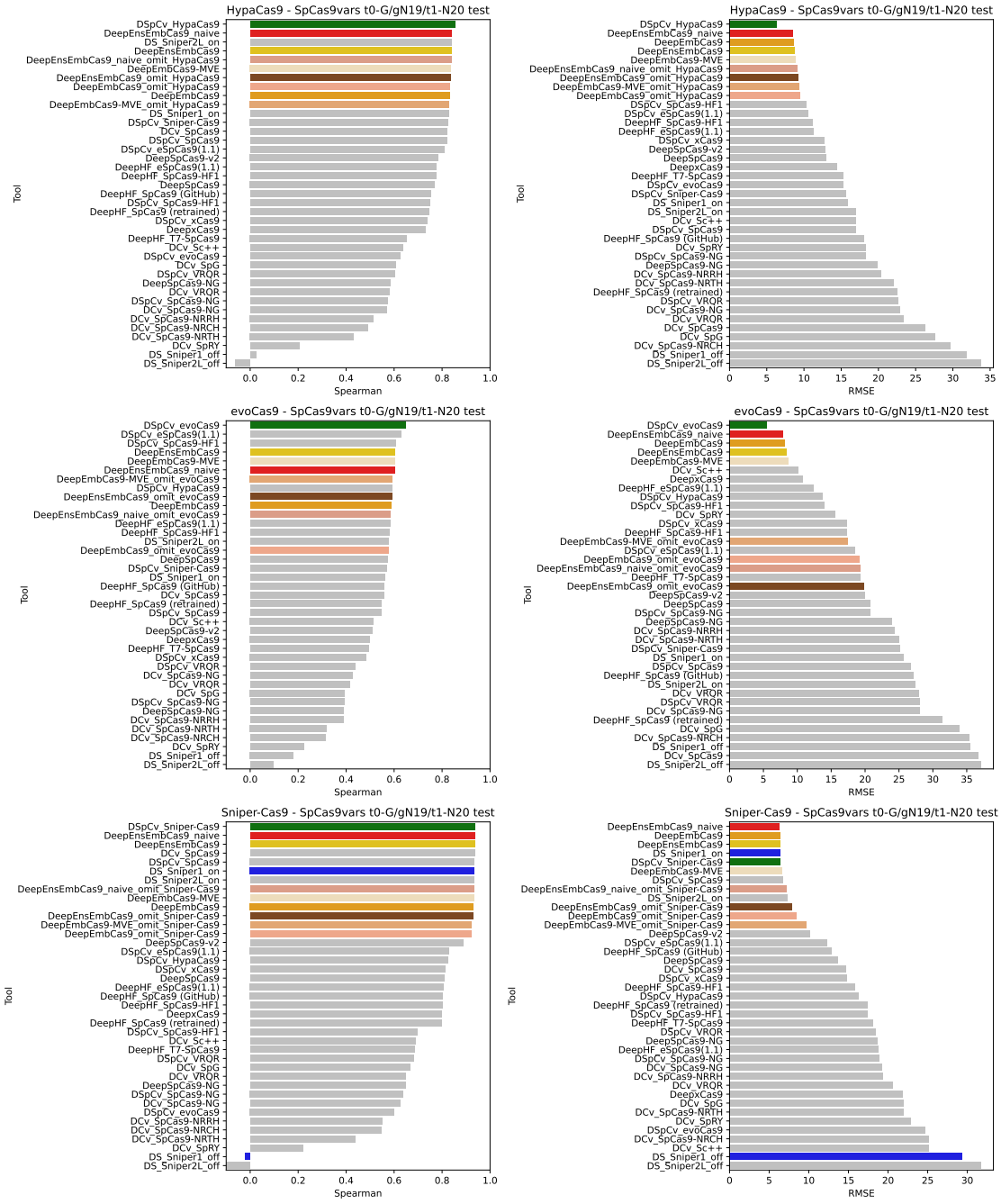


Figure C.13: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9 (top), evoCas9 (middle) and Sniper-Cas9 (bottom) interfaces from Kim, Kim et al. [7], where green bars denote DeepSpCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

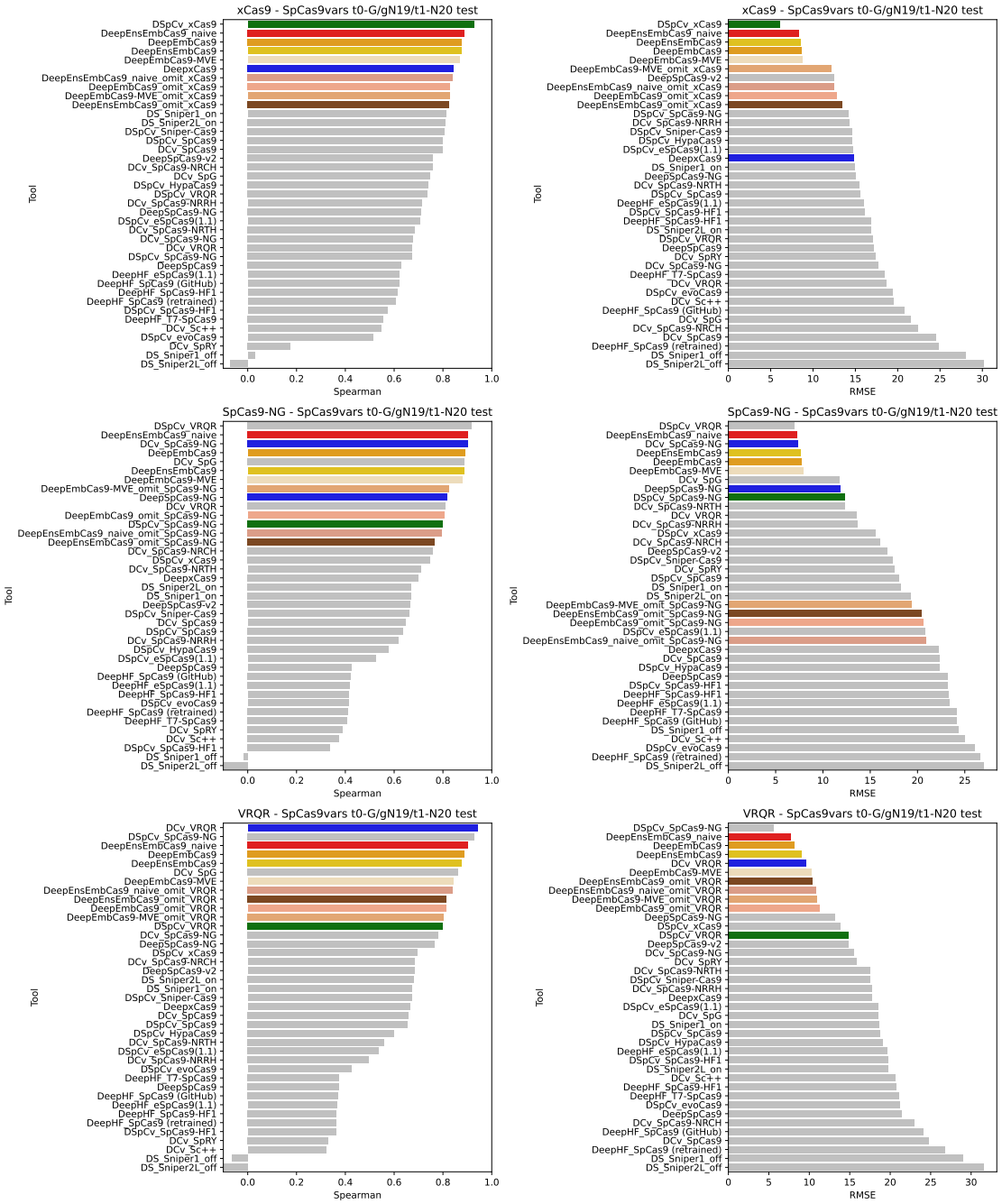


Figure C.14: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ xCas9 (top), SpCas9-NG (middle) and VRQR (bottom) interfaces from Kim, Kim et al. [7], where green bars denote DeepSpCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

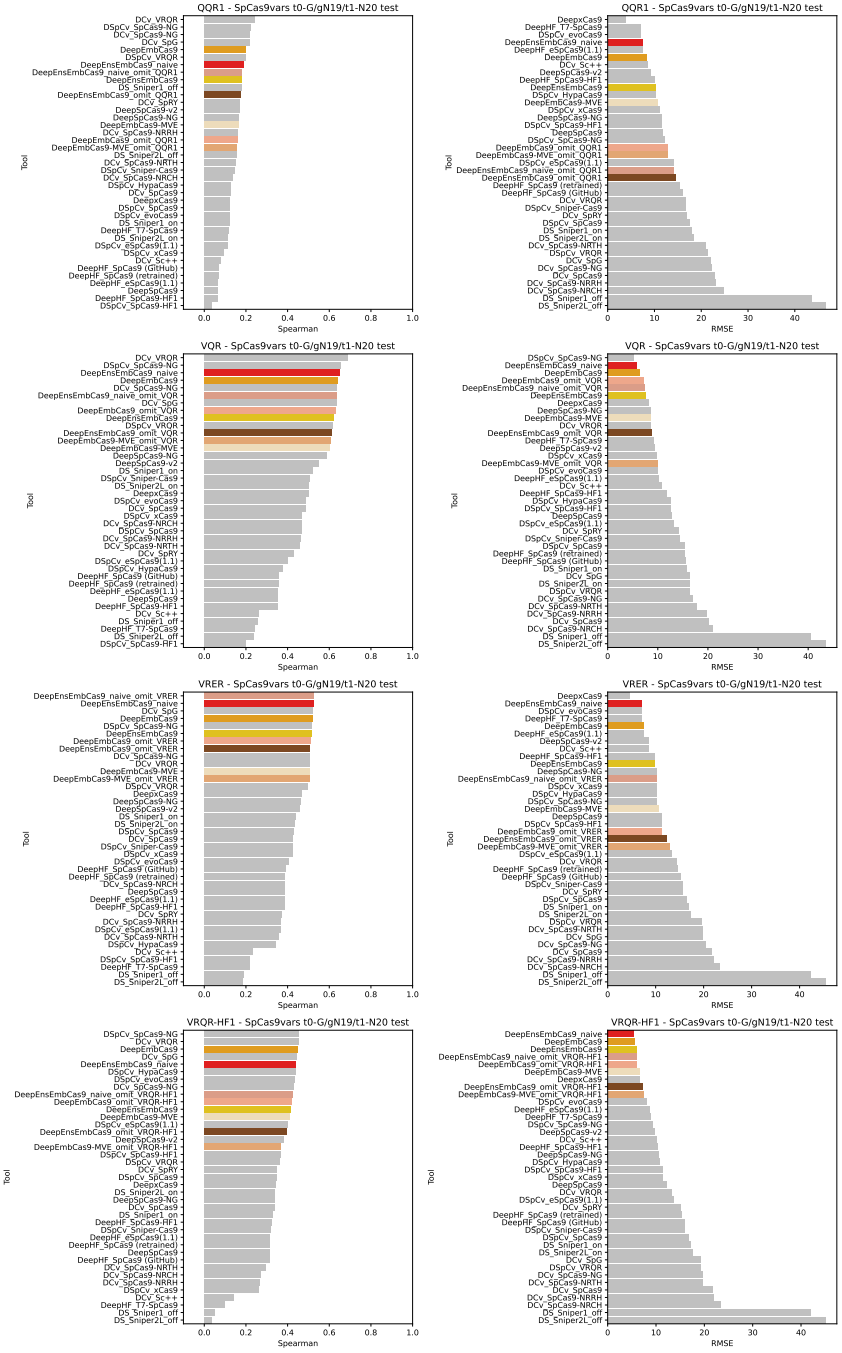


Figure C.15: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} QQR1 (row 1), VQR (row 2), VRER (row 3) and VRQR-HF1 (row 4) interfaces from Kim, Kim et al. [7], where green bars denote DeepSpCas9variants.

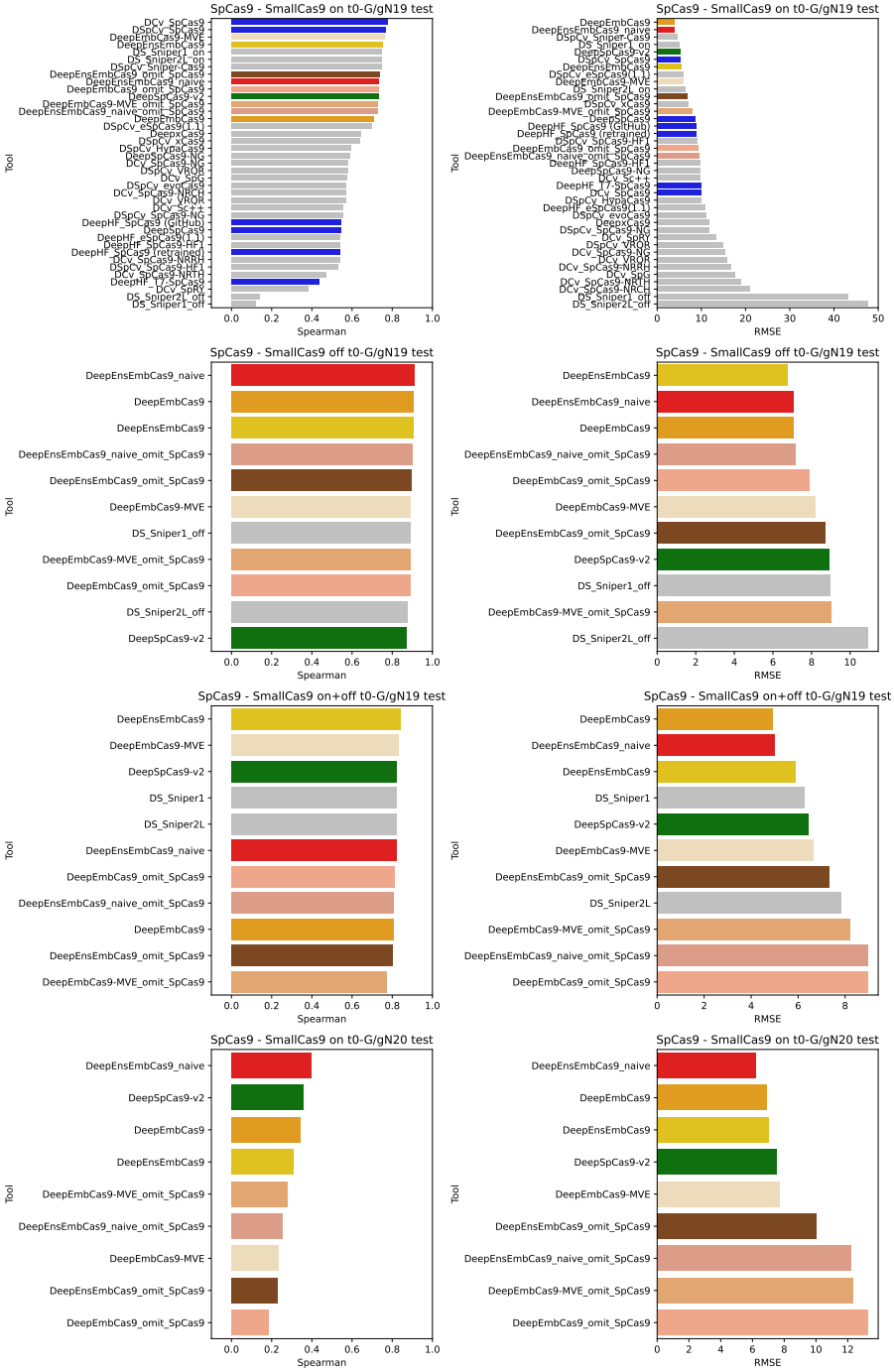


Figure C.16: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for wild type SpCas9 (specifically NLS-SpCas9-NLS-FLAG-P2A) with matched G/gN$_{19}$ (row 1), mismatched G/gN$_{19}$ (row 2), (mis)matched G/gN$_{19}$ (row 3) and matched G/gN20 interfaces (row 4) from Seo et al. [8], where green bars denote DeepSpCas9-v2, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

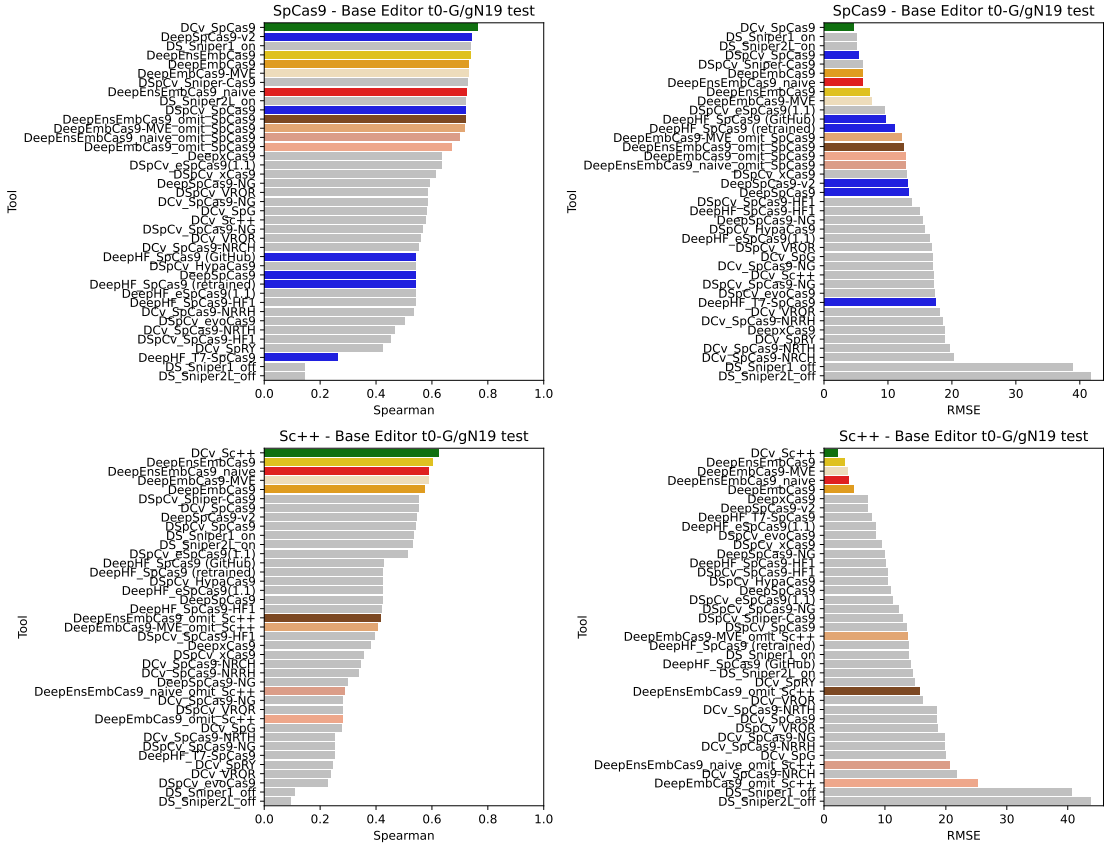


Figure C.17: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} SpCas9 (top) and Sc++ (bottom) interfaces from Kim, Choi et al. [9], where green bars denote DeepCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

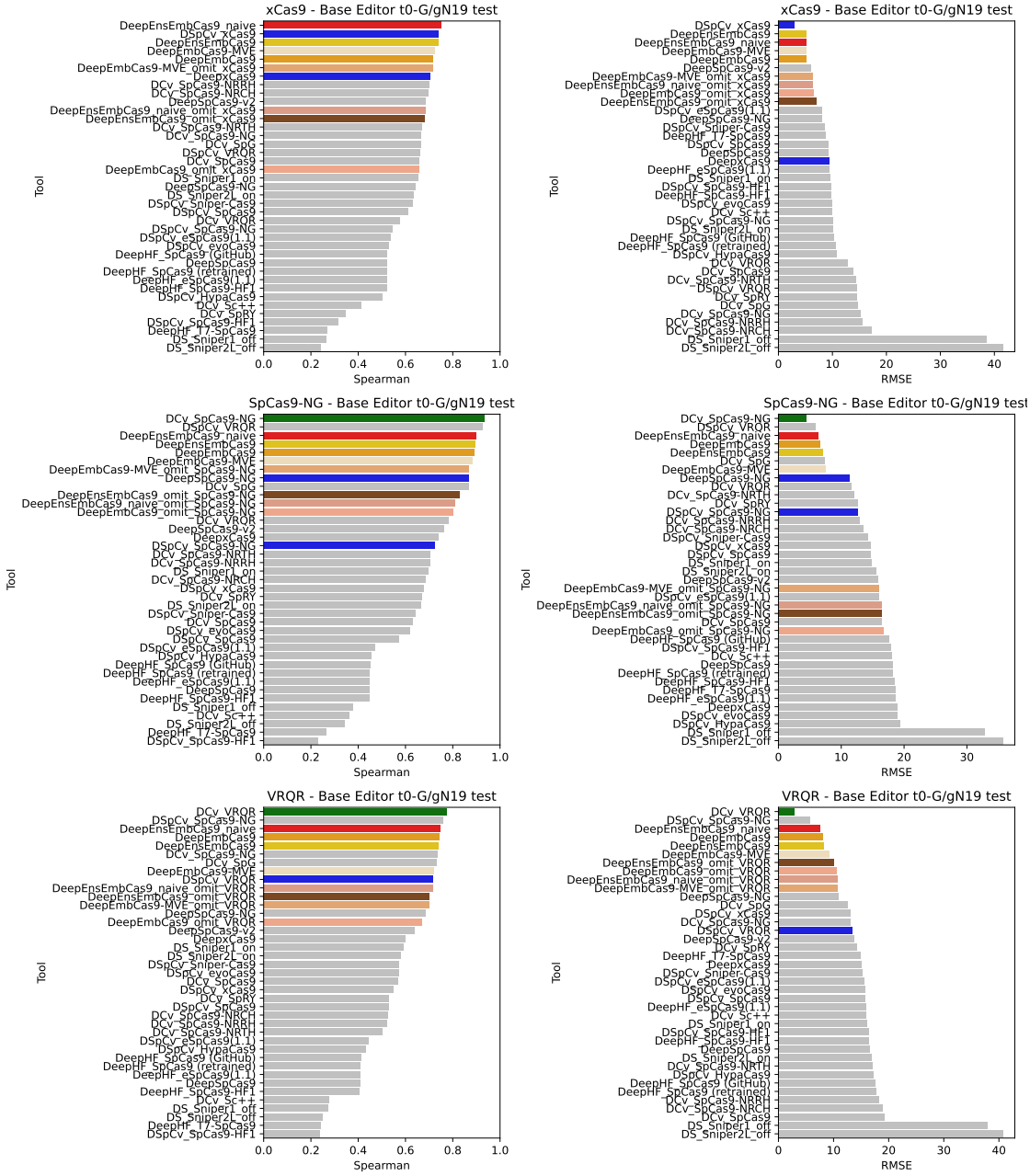


Figure C.18: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN$_{19}$ xCas9 (row 1), SpCas9-NG (row 2) and VRQR (row 3) interfaces from Kim, Choi et al. [9], where green bars denote DeepCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

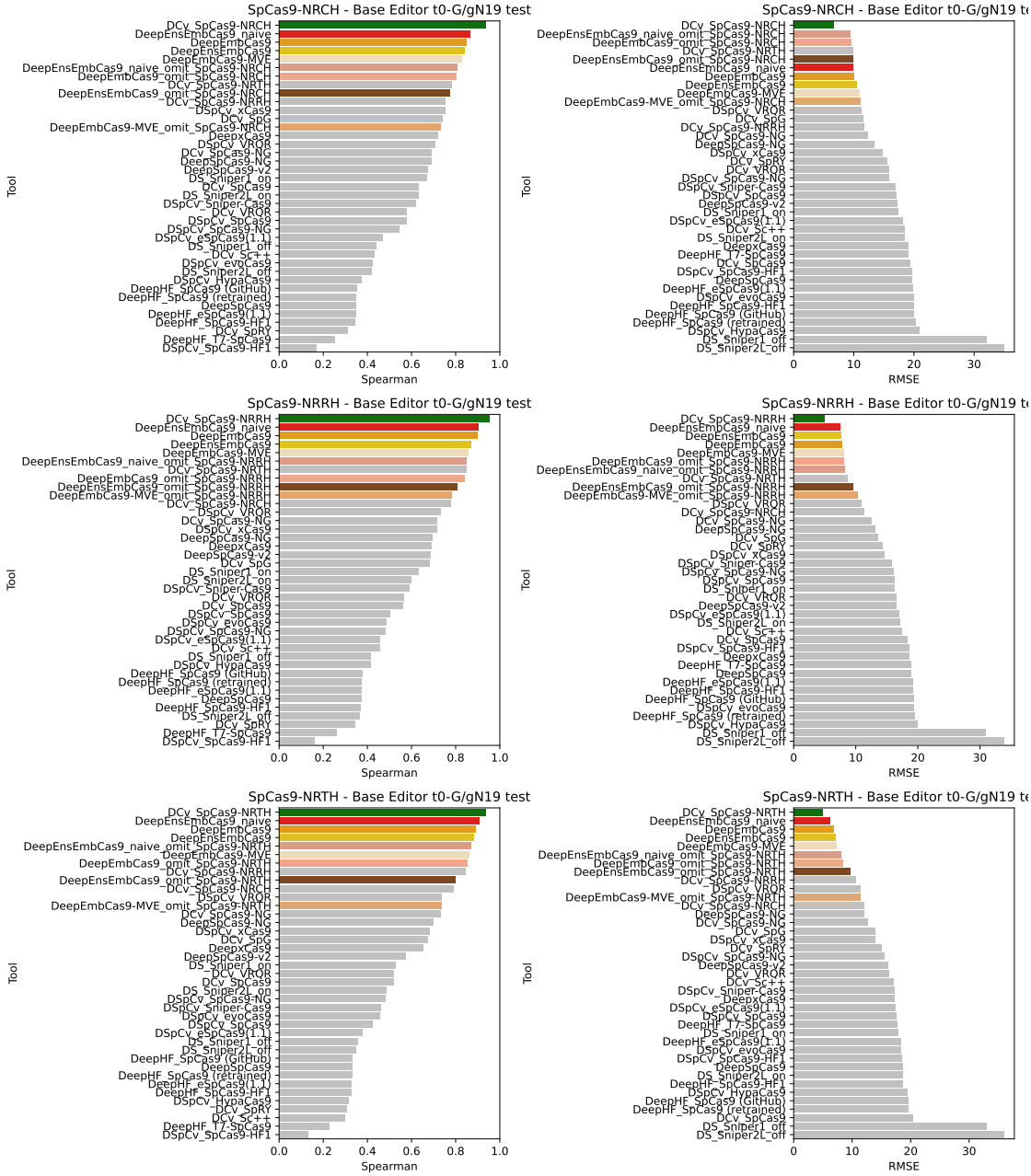


Figure C.19: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN_{19} SpCas9-NRCH (row 1), SpCas9-NRRH (row 2) and SpCas9-NRTH (row 3) interfaces from Kim, Choi et al. [9], where green bars denote DeepCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

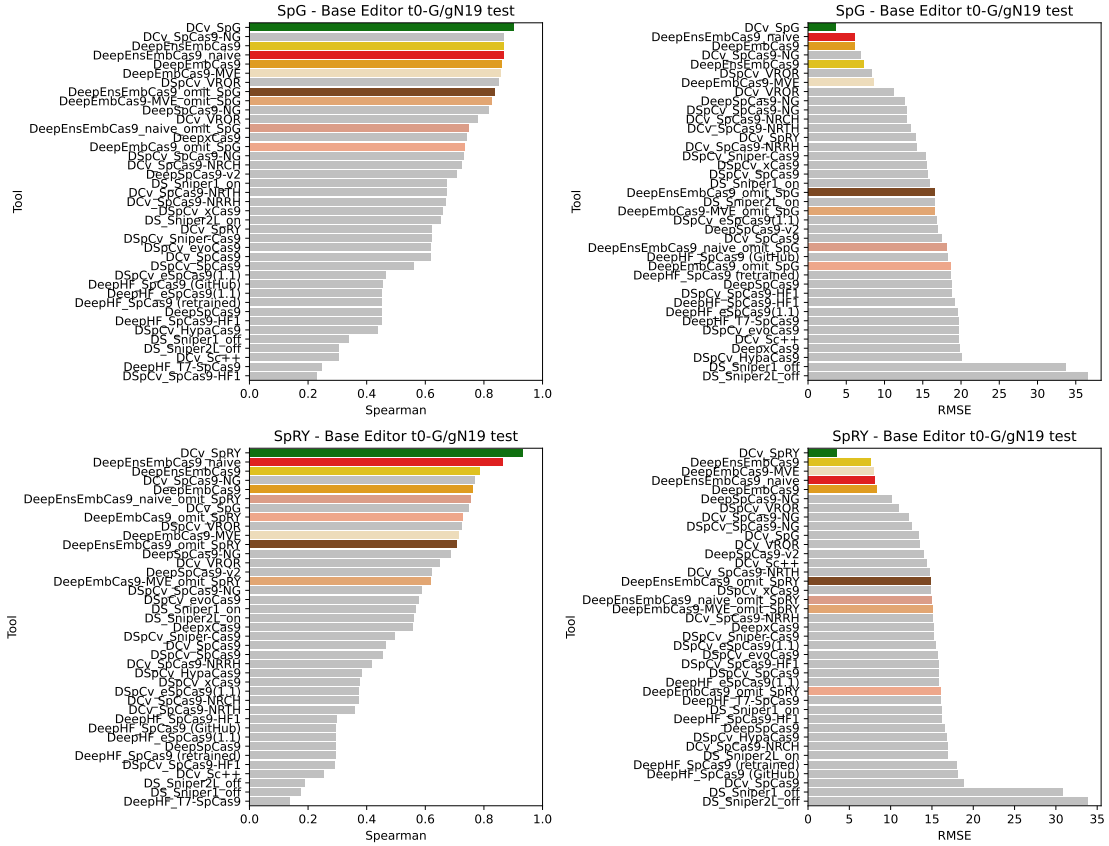


Figure C.20: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN$_{19}$ SpG (top) and SpRY (bottom) interfaces from Kim, Choi et al. [9], where green bars denote DeepCas9variants, and blue bars denote other individual Cas9 cleavage activity tools trained on matched interfaces of the test nuclease.

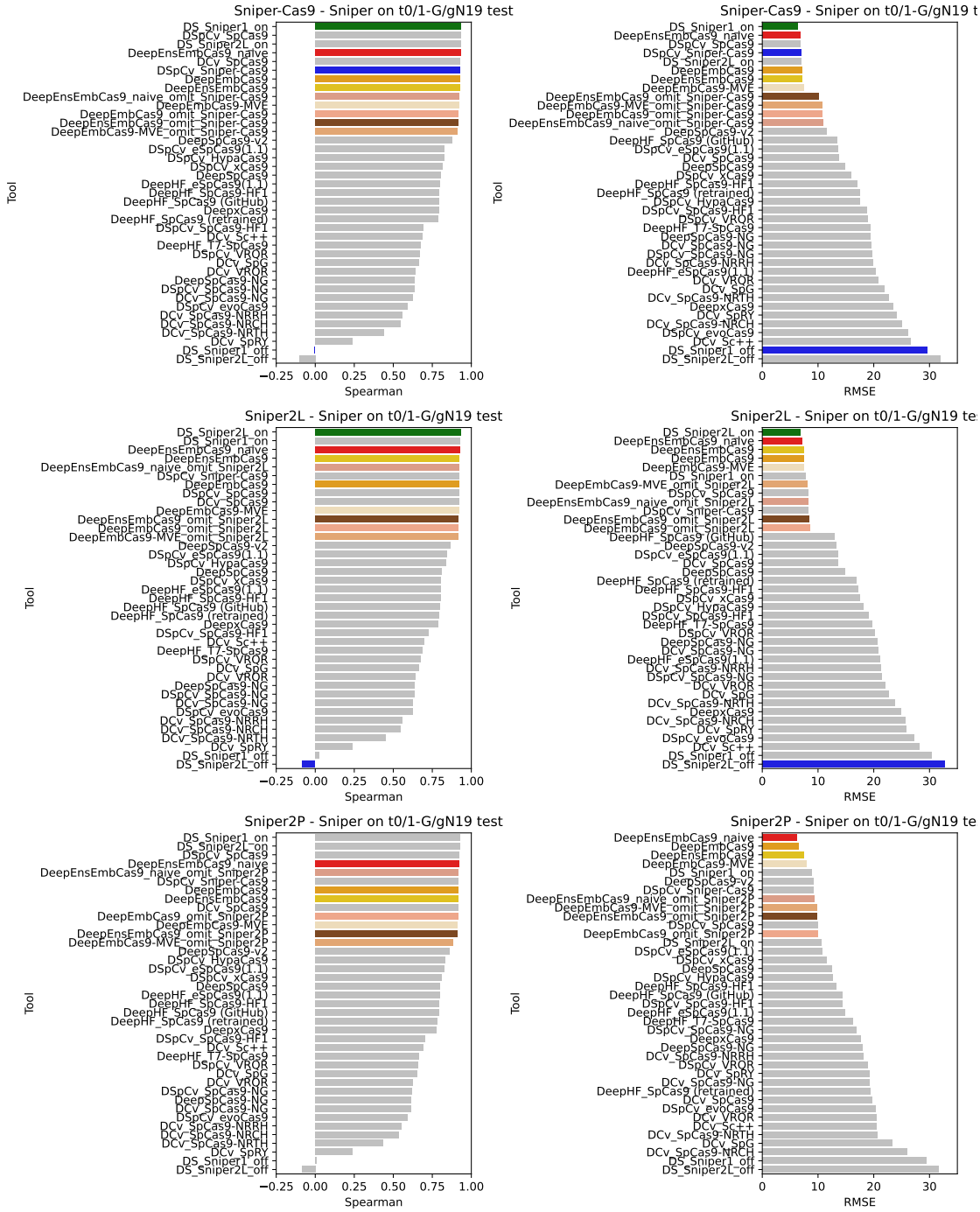


Figure C.21: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for matched G/gN$_{19}$ and tRNA$^{\text{Gln}}$-N$_{20}$ Sniper-Cas9 (top), Sniper2L (middle) and Sniper2P (bottom) interfaces from Kim, Kim and Okafor et al. [10], where green bars denote DeepSniper's Sniper1_on (top) and Sniper2L_on (middle), and blue bars denote other individual Cas9 cleavage activity tools trained on interfaces of the test nuclease.

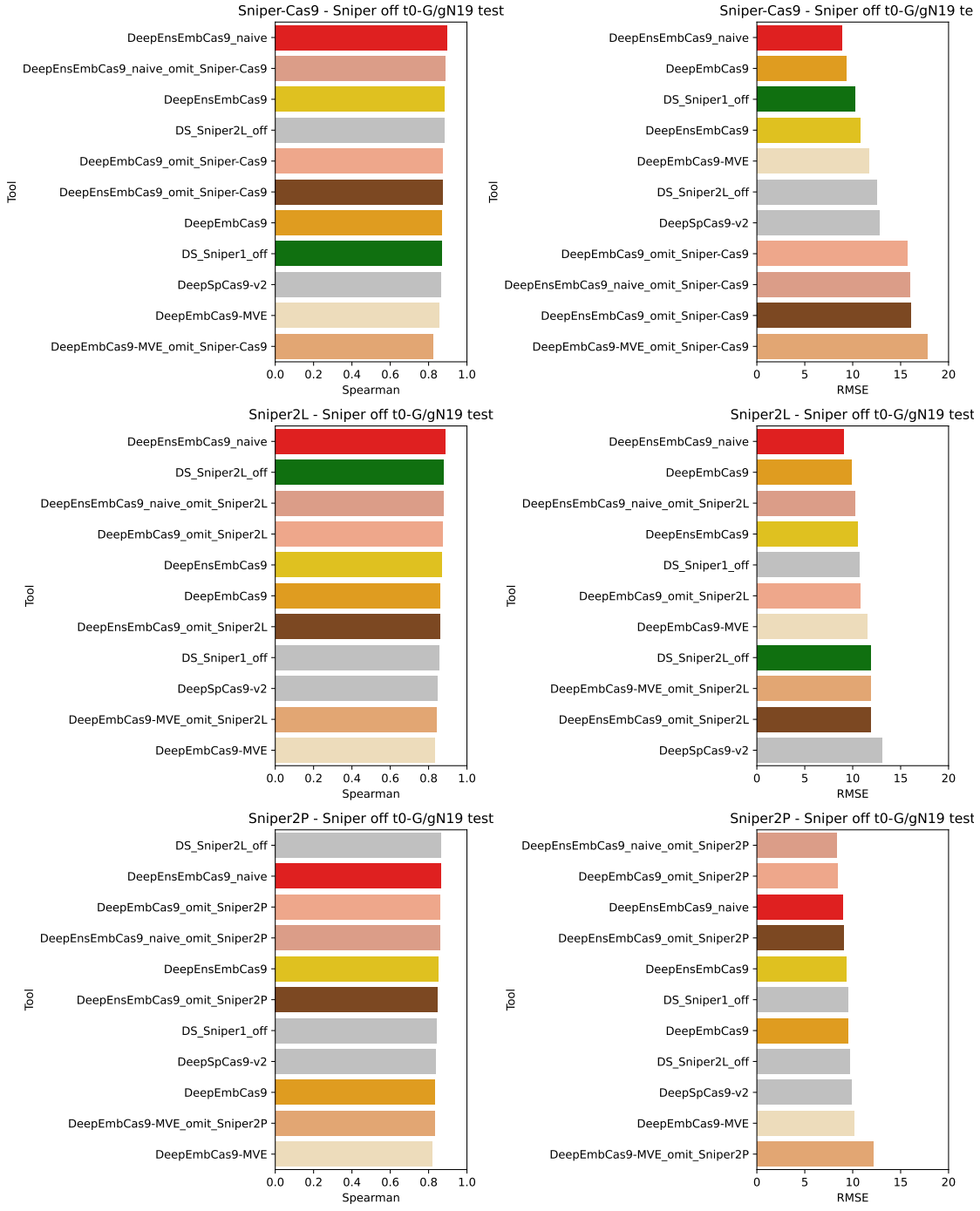


Figure C.22: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for mismatched G/gN$_{19}$ Sniper-Cas9 (top), Sniper2L (middle) and Sniper2P (bottom) interfaces from Kim, Kim and Okafor et al. [10], where green bars denote DeepSniper's Sniper1_off (top) and Sniper2L_off (middle), and blue bars denote other individual Cas9 cleavage activity tools trained on interfaces of the test nuclease.

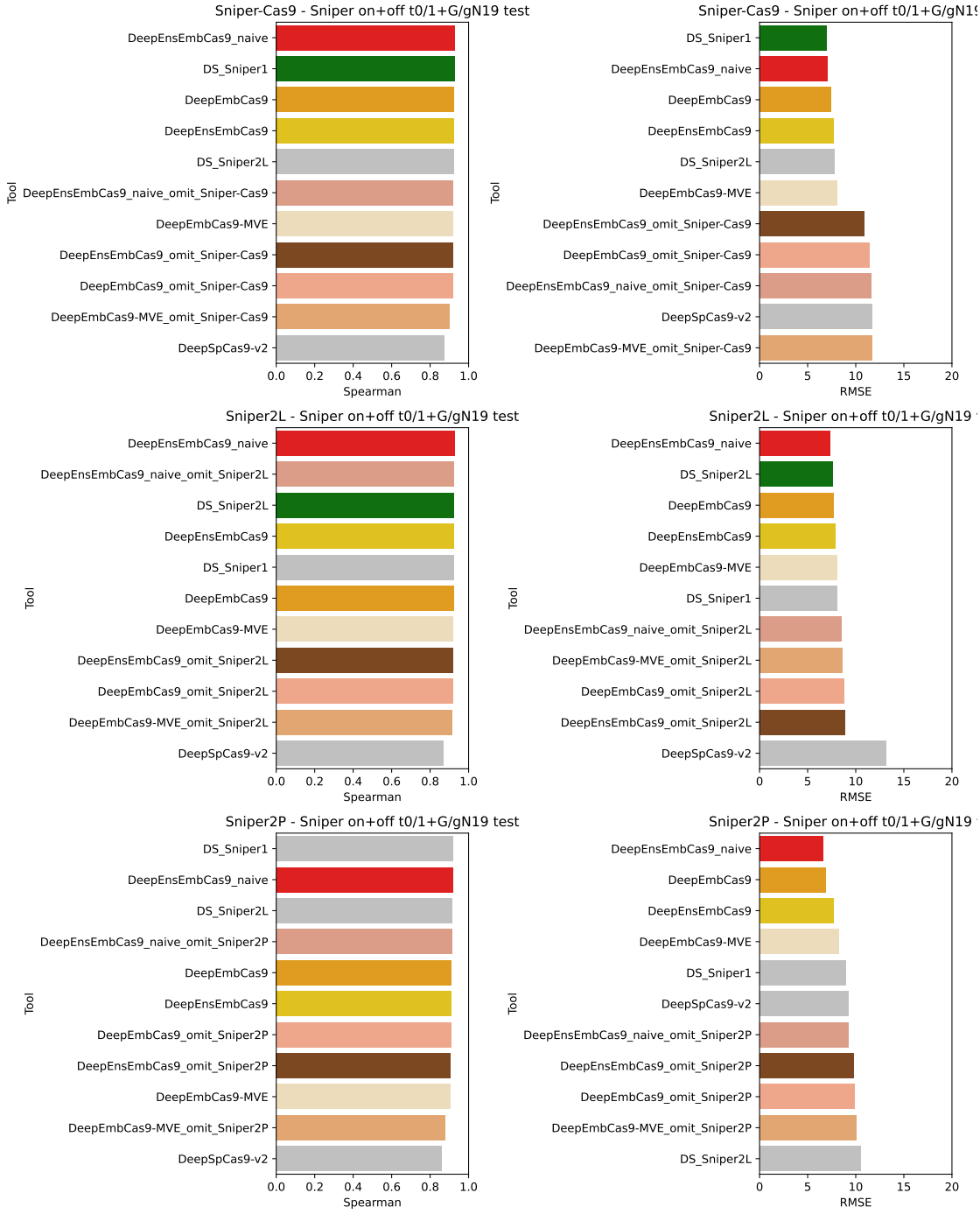


Figure C.23: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9_naive (red), DeepEmbCas9-MVE (wheat-colored), DeepEnsEmbCas9 (gold), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for mismatched G/gN$_{19}$ Sniper-Cas9 (top), Sniper2L (middle) and Sniper2P (bottom) interfaces from Kim, Kim and Okafor et al. [10], where green bars denote DeepSniper, and blue bars denote other individual Cas9 cleavage activity tools trained on interfaces of the test nuclease.

Small Cas9 variants

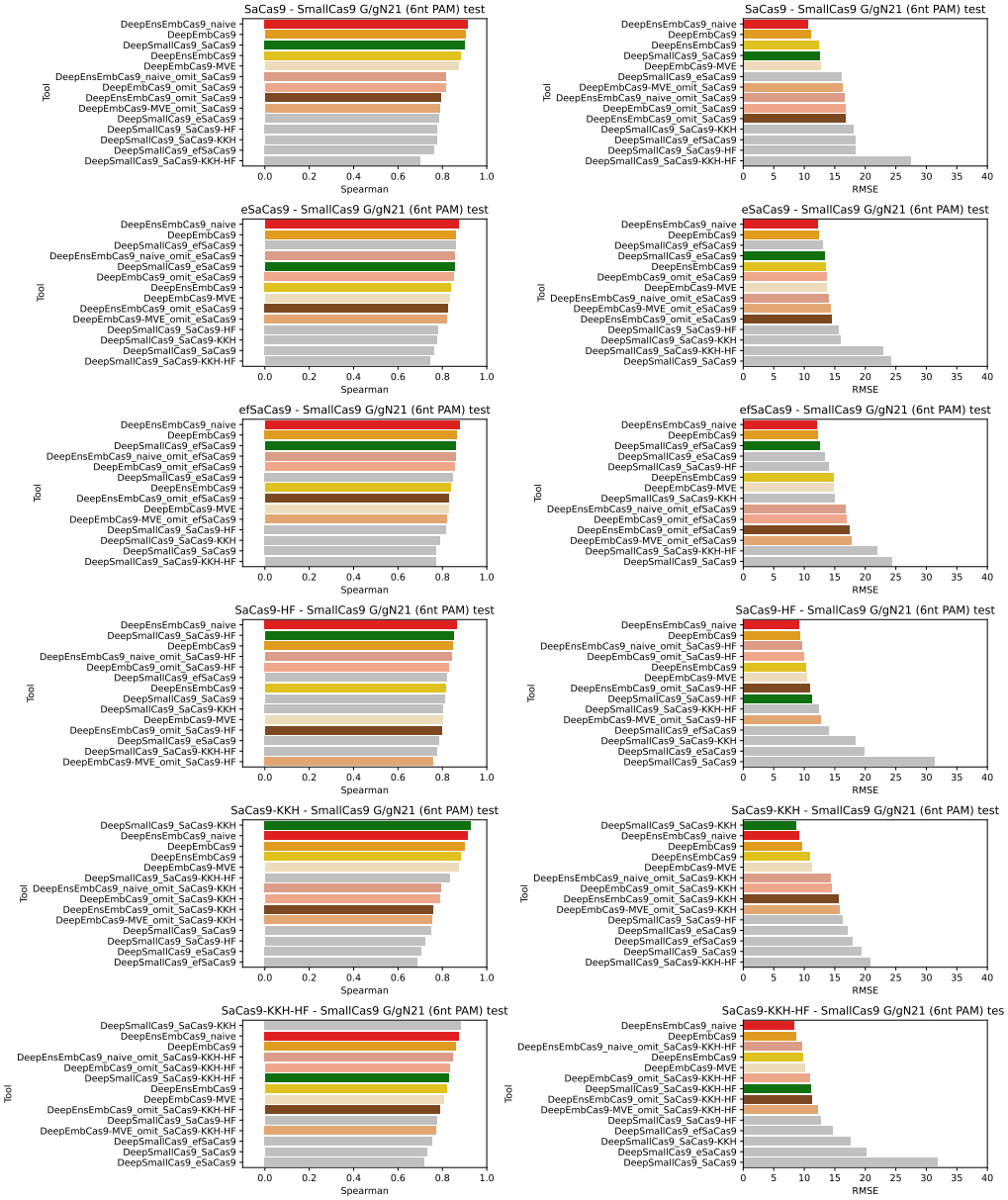


Figure C.24: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9 (red), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for (mis)matched G/gN$_{21}$ wild type and engineered SaCas9 interfaces from Seo et al. [8], where green bars denote DeepSmallCas9.

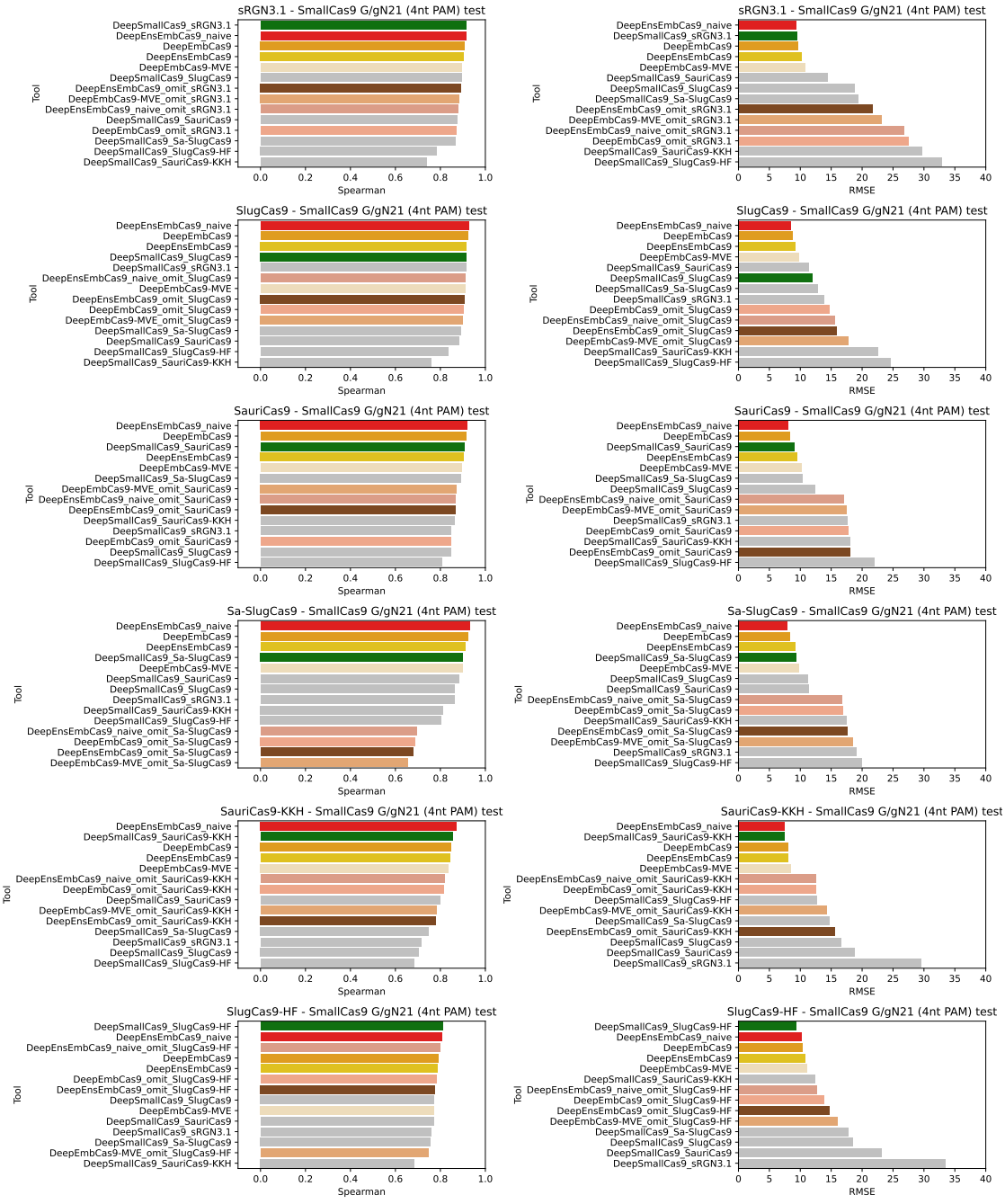


Figure C.25: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9 (red), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for (mis)matched G/gN$_{21}$ wild type and engineered SlugCas9/sRGN3.1/SauriCas9 interfaces from Seo et al. [8], where green bars denote DeepSmallCas9.

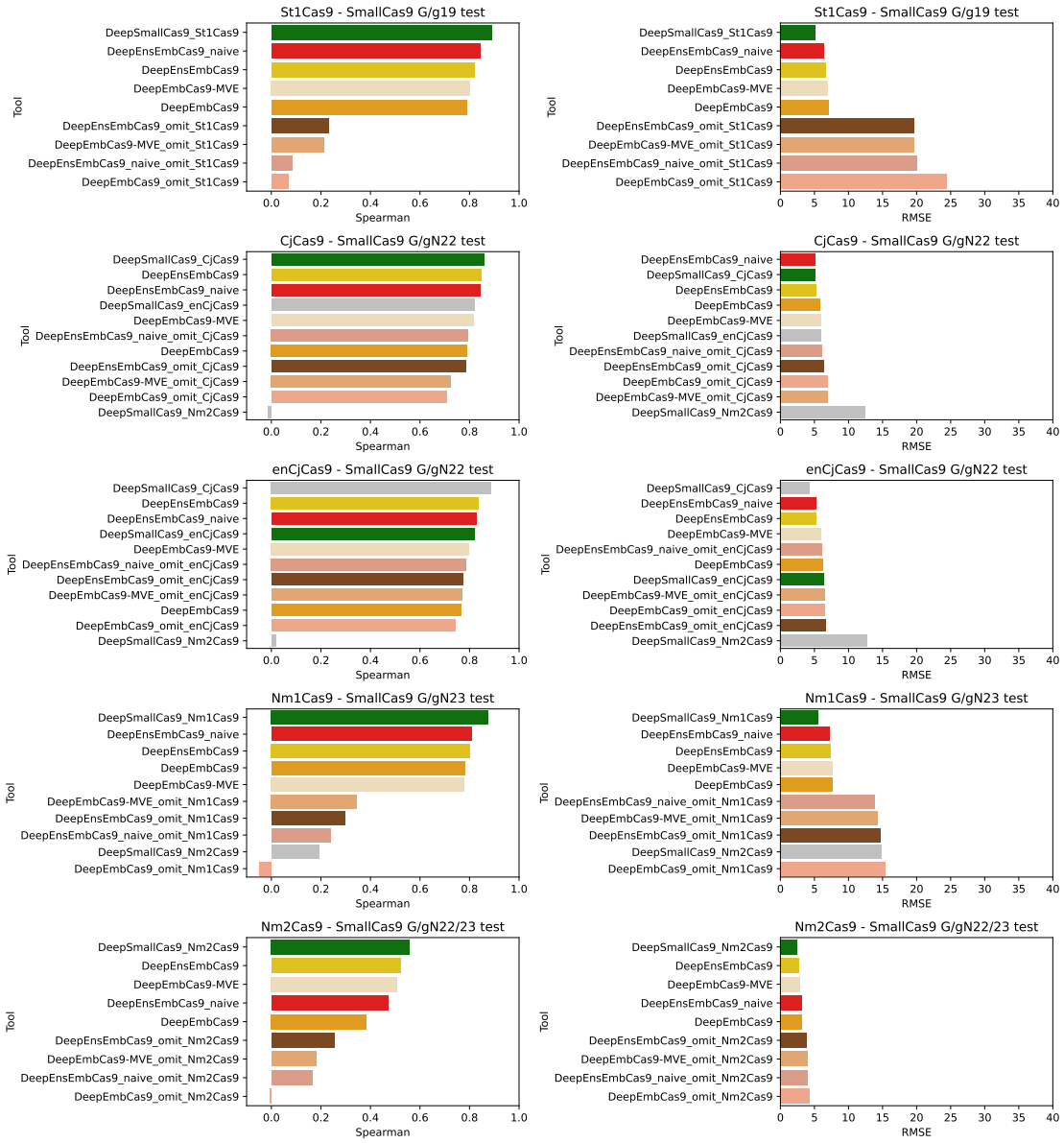


Figure C.26: Benchmark test Spearman correlation (left) and RMSE (right) comparisons for DeepEmbCas9 (orange), DeepEnsEmbCas9 (red), DeepEmbCas9_omit (dark salmon) and DeepEnsEmbCas9_omit (brown) against relevant individual Cas9 cleavage activity tools for (mis)matched G/gN$_{21}$ G/gN$_{19}$ St1Cas9 (row 1), G/gN$_{22}$ CjCas9 (row 2), G/gN$_{22}$ enCjCas9 (row 3), G/gN$_{23}$ Nm1Cas9 (row 4) and G/gN$_{22-23}$ Nm2Cas9 (row 5) interfaces from Seo et al. [8], where green bars denote DeepSmallCas9.

C.2.4 Extrapolation performance on whole dataset

pLM	rLM	Guide-target 5-fold CV	Cas9 variants 5-fold CV	gRNA scaffold LOOCV	Overall
ESM-C-300M	BEACON-B	0.8989 ± 0.0026	0.6993 ± 0.0944	**0.5679 ± 0.1396**	**0.7220 ± 0.1666**
ESM-C-600M	RiNALMo	0.8988 ± 0.0021	**0.7200 ± 0.0767**	0.5470 ± 0.1624	0.7219 ± 0.1759
ESM-C-600M	RNA-FM	**0.9017 ± 0.0011**	0.6865 ± 0.0842	0.5670 ± 0.1411	0.7184 ± 0.1696
ESM-C-300M	RNA-FM	0.8986 ± 0.0016	0.6996 ± 0.1040	0.5548 ± 0.1325	0.7177 ± 0.1726
ESM-C-600M	BEACON-B512	0.8991 ± 0.0012	0.7016 ± 0.0674	0.5474 ± 0.1393	0.7160 ± 0.1763
ESM-C-300M	RiNALMo	0.8988 ± 0.0010	0.6991 ± 0.0811	0.5443 ± 0.1755	0.7141 ± 0.1777
ESM-C-300M	BEACON-B512	0.8963 ± 0.0009	0.6899 ± 0.0976	0.5524 ± 0.1493	0.7129 ± 0.1731
ESM-C-600M	evo-1-8k	0.8980 ± 0.0007	0.7085 ± 0.0698	0.5247 ± 0.1636	0.7104 ± 0.1866
ESM-C-6B	BEACON-B	0.8951 ± 0.0016	0.7047 ± 0.0957	0.5302 ± 0.1634	0.7100 ± 0.1825
ESM-C-600M	BEACON-B	0.9013 ± 0.0016	0.6973 ± 0.0607	0.5289 ± 0.1650	0.7092 ± 0.1865
ESM-C-6B	RNA-FM	0.8977 ± 0.0031	0.6965 ± 0.0775	0.5306 ± 0.1972	0.7083 ± 0.1839
ESM-C-6B	BEACON-B512	0.8919 ± 0.0018	0.6937 ± 0.0892	0.5334 ± 0.1674	0.7064 ± 0.1796
ESM-C-300M	evo-1-8k	0.8947 ± 0.0009	0.6985 ± 0.0876	0.5228 ± 0.1294	0.7053 ± 0.1860
ProtT5	evo-1-8k	0.8898 ± 0.0033	0.6725 ± 0.1078	0.5456 ± 0.1433	0.7026 ± 0.1741
ESM-C-6B	RiNALMo	0.8947 ± 0.0020	0.6843 ± 0.0856	0.5242 ± 0.1808	0.7011 ± 0.1858
Ankh-large	RNA-FM	0.8914 ± 0.0026	0.6593 ± 0.1050	0.5438 ± 0.1577	0.6982 ± 0.1770
Ankh-large	BEACON-B	0.8919 ± 0.0016	0.6600 ± 0.1020	0.5341 ± 0.1781	0.6953 ± 0.1815
ProtT5	RNA-FM	0.8950 ± 0.0016	0.6418 ± 0.1175	0.5470 ± 0.1636	0.6946 ± 0.1799
Ankh-large	BEACON-B512	0.8873 ± 0.0015	0.6608 ± 0.0943	0.5323 ± 0.1484	0.6935 ± 0.1797
ProtT5	RiNALMo	0.8934 ± 0.0033	0.6687 ± 0.1049	0.5168 ± 0.2108	0.6930 ± 0.1895
ProtT5	BEACON-B512	0.8890 ± 0.0022	0.6504 ± 0.0957	0.5379 ± 0.1812	0.6924 ± 0.1793
ProtT5	BEACON-B	0.8927 ± 0.0026	0.6496 ± 0.1228	0.5223 ± 0.2016	0.6882 ± 0.1882
Ankh-large	evo-1-8k	0.8903 ± 0.0013	0.6552 ± 0.1163	0.5178 ± 0.1452	0.6878 ± 0.1884
Ankh-large	RiNALMo	0.8913 ± 0.0024	0.6503 ± 0.1159	0.5182 ± 0.2031	0.6866 ± 0.1892
ESM-C-6B	evo-1-8k	0.8898 ± 0.0013	0.6747 ± 0.0997	0.4879 ± 0.1958	0.6841 ± 0.2011
gLM2-650M	BEACON-B512	0.8193 ± 0.0024	0.6447 ± 0.0730	0.4594 ± 0.1553	0.6411 ± 0.1800
gLM2-650M	RNA-FM	0.8251 ± 0.0035	0.6386 ± 0.0512	0.4514 ± 0.2073	0.6384 ± 0.1869
ESM3	RNA-FM	0.8239 ± 0.0042	0.6325 ± 0.1043	0.4531 ± 0.1879	0.6365 ± 0.1854
gLM2-650M	evo-1-8k	0.8193 ± 0.0031	0.6614 ± 0.0645	0.4251 ± 0.1809	0.6353 ± 0.1984
gLM2-650M	RiNALMo	0.7964 ± 0.0725	0.6252 ± 0.0477	0.4705 ± 0.1621	0.6307 ± 0.1630
gLM2-650M	BEACON-B	0.8239 ± 0.0043	0.5894 ± 0.0987	0.4675 ± 0.1719	0.6269 ± 0.1811
ESM3	BEACON-B512	0.8007 ± 0.0415	0.6301 ± 0.1038	0.4127 ± 0.2113	0.6145 ± 0.1944
ESM3	RiNALMo	0.7954 ± 0.0671	0.6136 ± 0.1152	0.4262 ± 0.1379	0.6117 ± 0.1846
ESM3	evo-1-8k	0.8225 ± 0.0021	0.5896 ± 0.1003	0.4191 ± 0.1518	0.6104 ± 0.2025
ESM3	BEACON-B	0.7397 ± 0.0826	0.5351 ± 0.0827	0.3915 ± 0.1875	0.5554 ± 0.1750

Table C.5: Test Spearman correlation of the 30 pLM-rLM embedding combinations (arising from 6 pLM (Ankh-large, ESM3, ESM-C-300M, ESM-C-600M, ESM-C-6B, gLM2-650M, ProtT5) and 5 rLM (BEACON-B, BEACON-B512, RNA-FM, RiNALMo, evo-1-8k) embeddings) considered for DeepEmbCas9 across three tasks — guide-target 5-fold cross validation (CV), Cas9 variants 5-fold cross validation and gRNA scaffold leave-one-out cross validation (LOOCV). The pLM-rLM combinations are ranked by decreasing "Overall" score, which denotes the average between the mean performances in the three tasks.

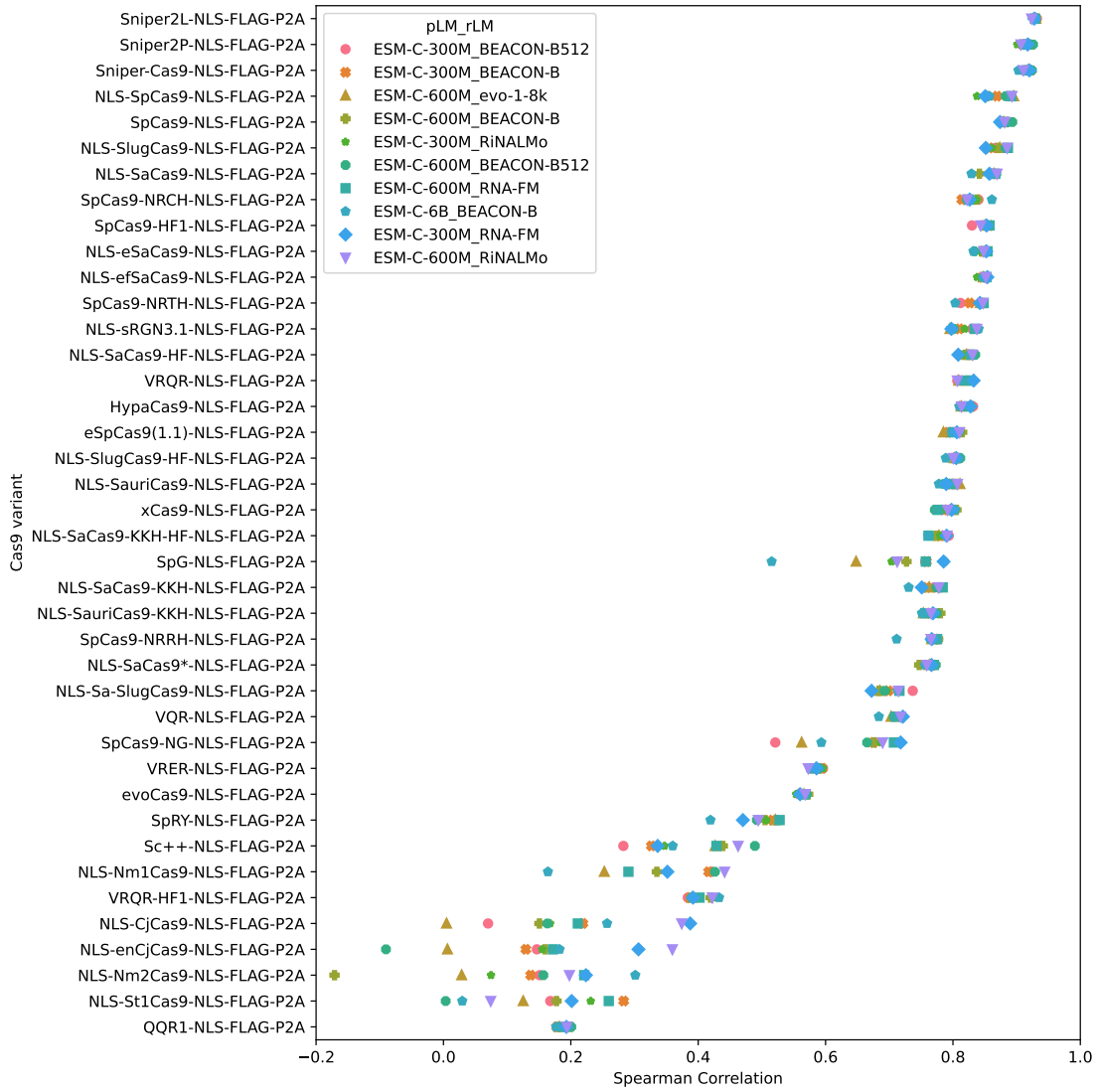


Figure C.27: Test Spearman correlations when holding out data associated with one Cas9 variant for testing for the 10 pLM-rLM combinations with the highest "Overall" score in Table C.5 (see list of Cas9 mutations in Table C.4), with Cas9 variants roughly sorted in descending Spearman correlation.

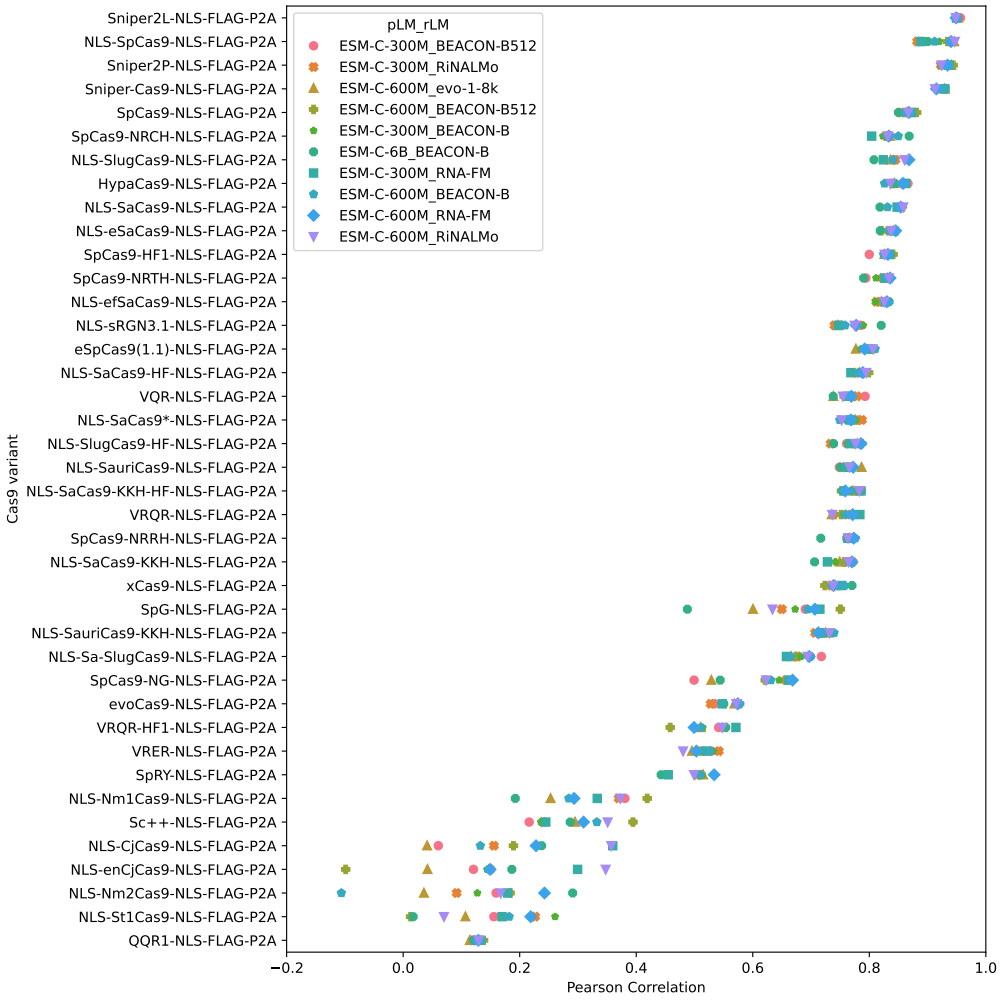


Figure C.28: Test Pearson correlation of the 18 pLM-rLM embedding combinations (arising from 6 pLM (Ankh-large, ESM3, ESM-C-300M, ESM-C-600M, ESM-C-6B, gLM2-650M, ProtT5) and 3 rLM (BEACON-B, BEACON-B512, RNA-FM) embeddings) considered for DeepEmbCas9 across three tasks — guide-target 5-fold cross validation (CV), Cas9 variants 5-fold cross validation and gRNA scaffold leave-one-out cross validation (LOOCV). The pLM-rLM combinations are ranked by decreasing "Overall" score, which denotes the average between the mean performances in the three tasks.

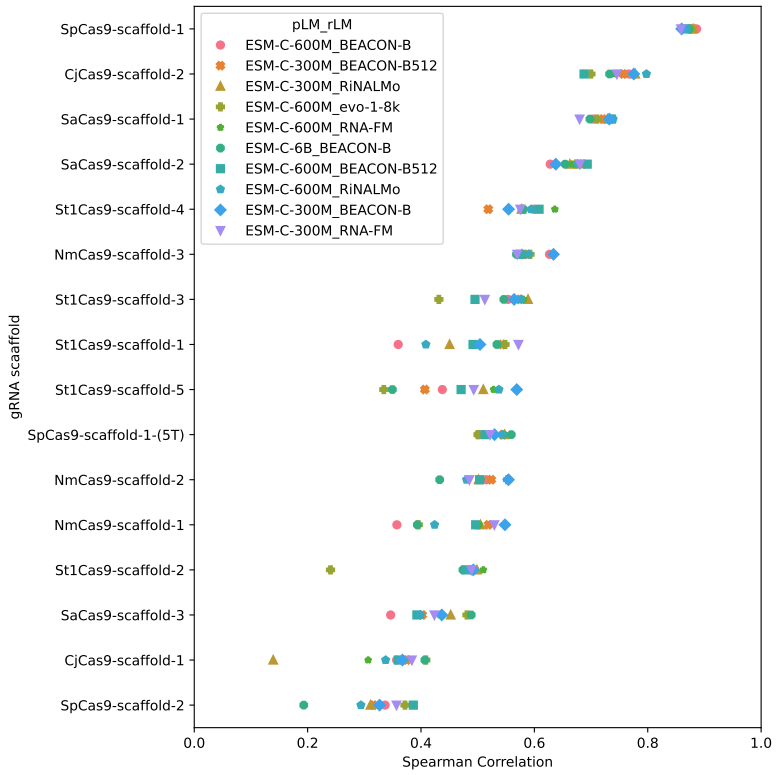


Figure C.29: Test Spearman correlations when holding out data associated with one gRNA scaffold for testing for the 10 pLM-rLM combinations with the highest "Overall" score in Table C.5, with Cas9 variants roughly sorted in descending Spearman correlation.

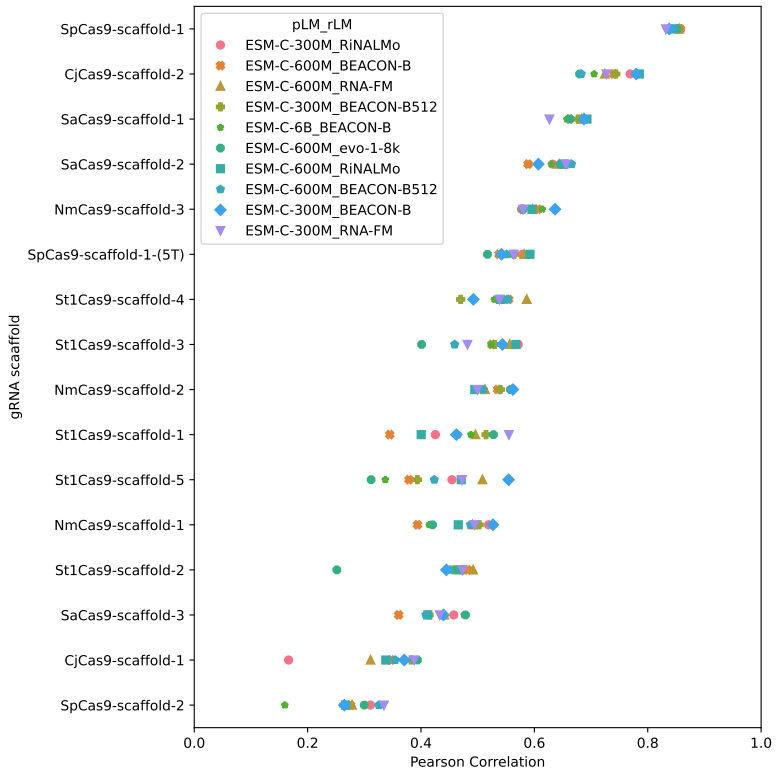


Figure C.30: Test Pearson correlations when holding out data associated with one Cas9 variant for testing for 10 pLM-rLM combinations with the highest "Overall" score in Table C.5 (see list of Cas9 mutations in Table C.4).

C.2.5 In-distribution calibration

Quantile calibration plots - DeepEnsEmbCas9

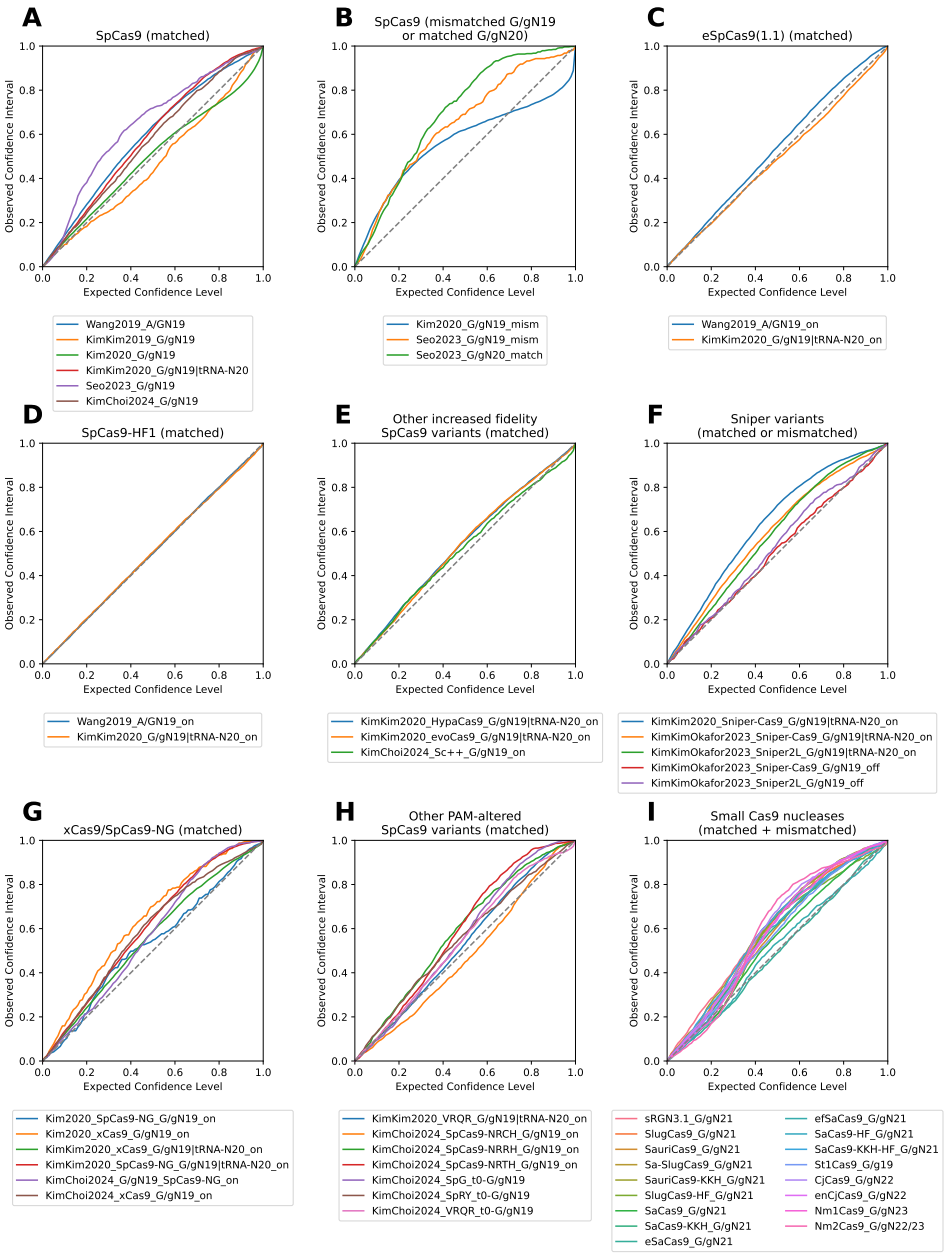


Figure C.31: Confidence interval-based calibration curves for DeepEnsEmbCas9, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Quantile calibration plots - DeepEmbCas9-MVE

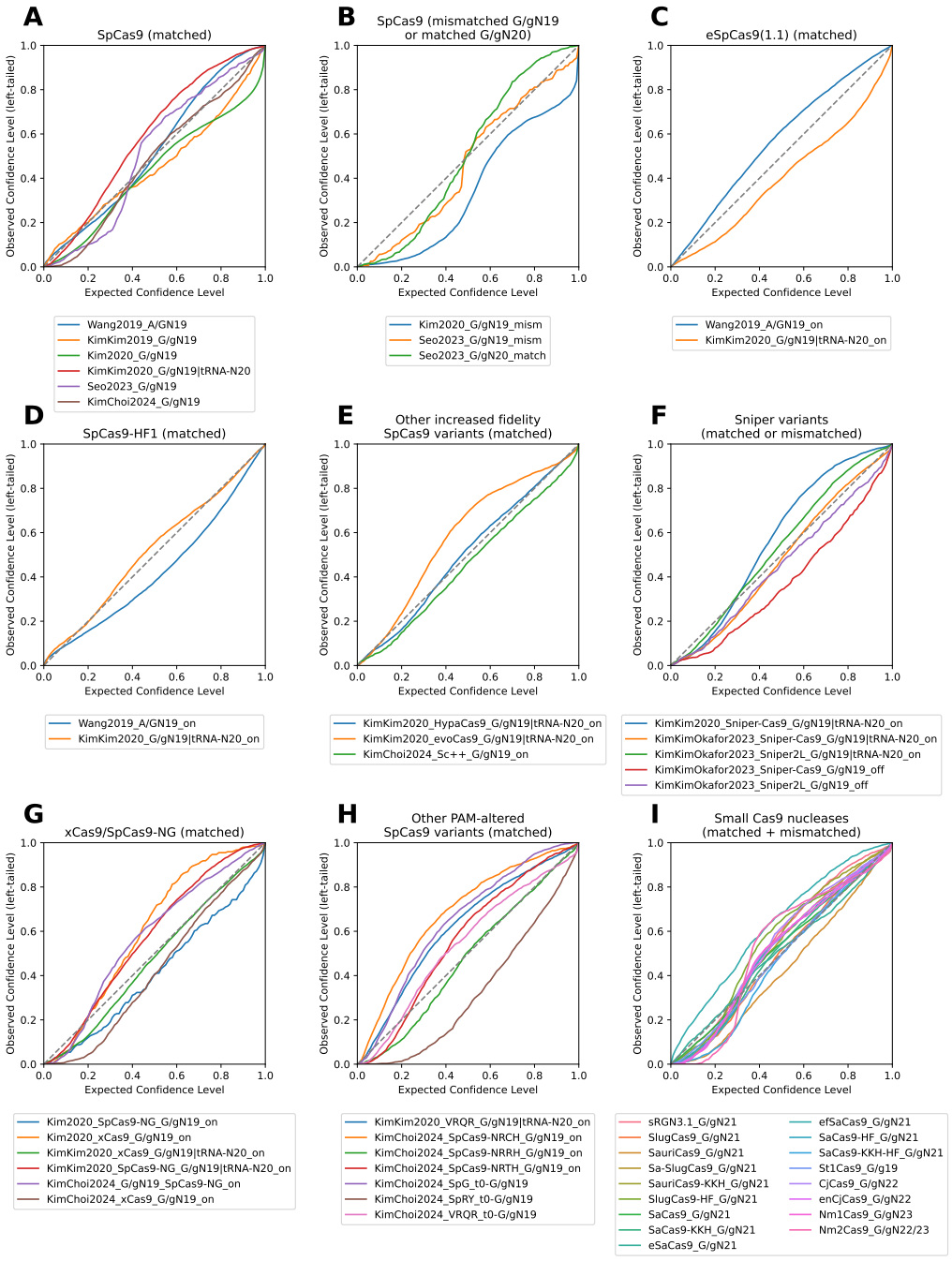


Figure C.32: Quantile calibration plots for DeepEmbCas9-MVE, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

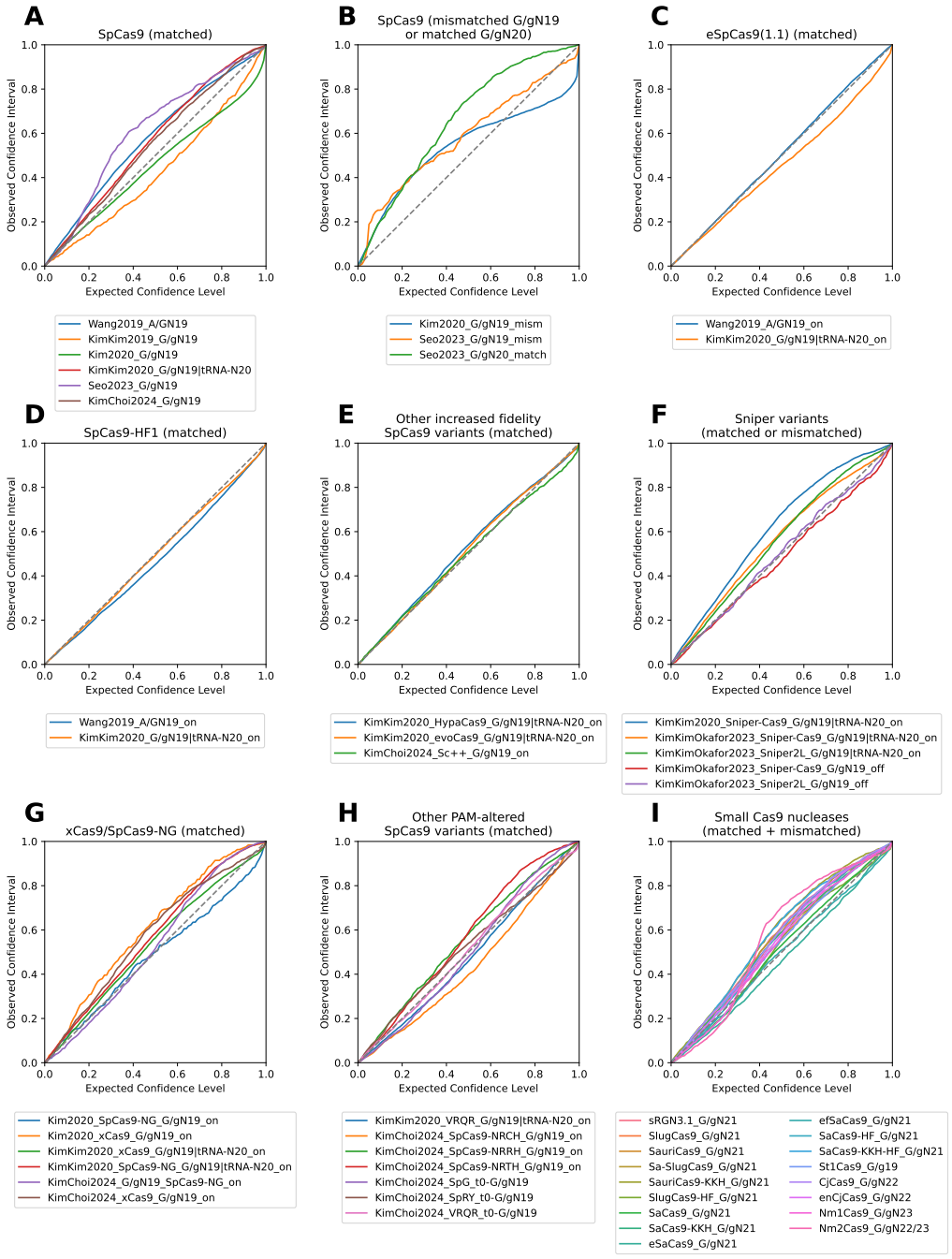


Figure C.33: Confidence interval-based calibration curves for DeepEmbCas9-MVE, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Quantile calibration plots - DeepEnsEmbCas9_naive

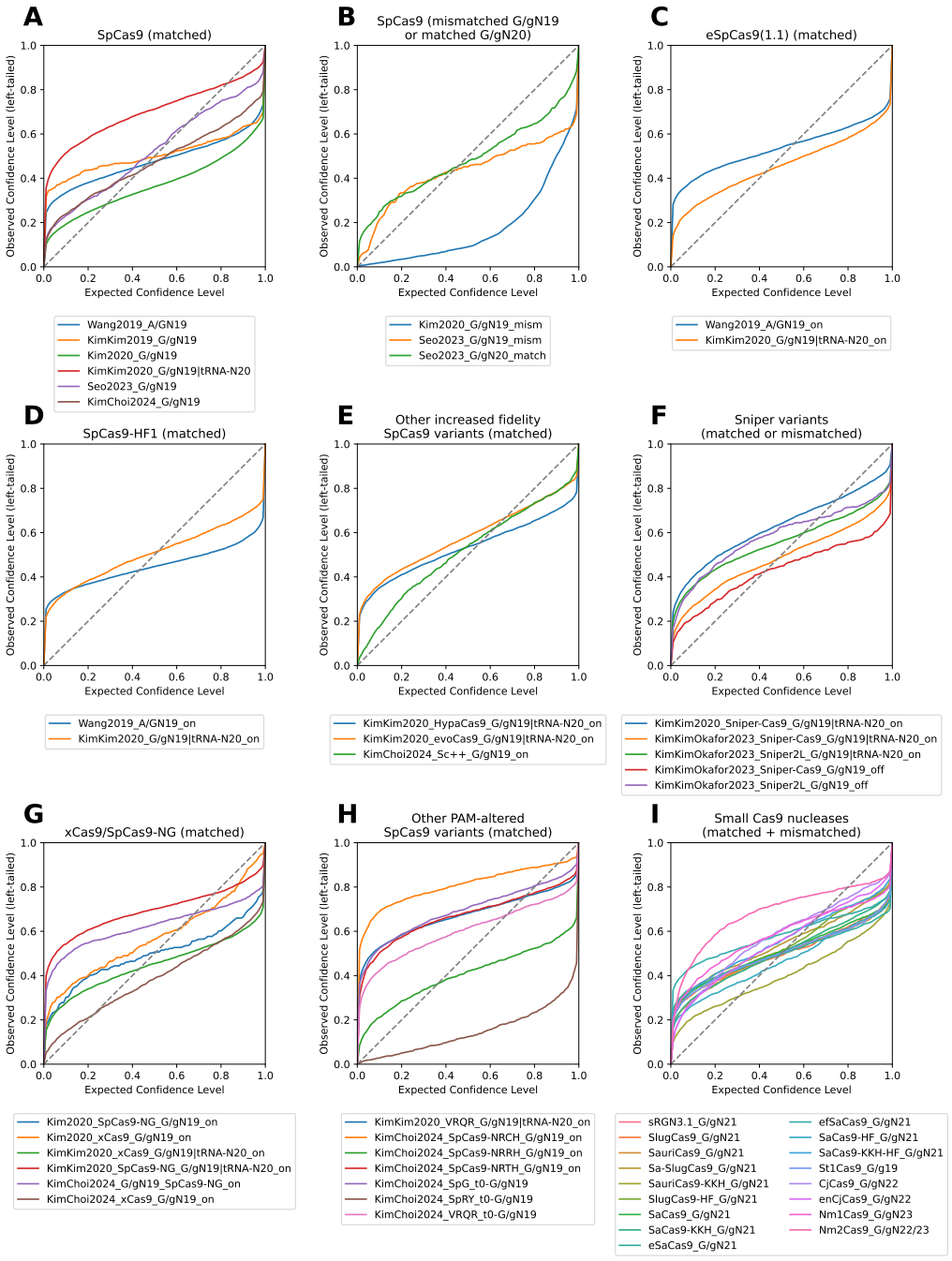


Figure C.34: Quantile calibration plots for DeepEnsEmbCas9_naive, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

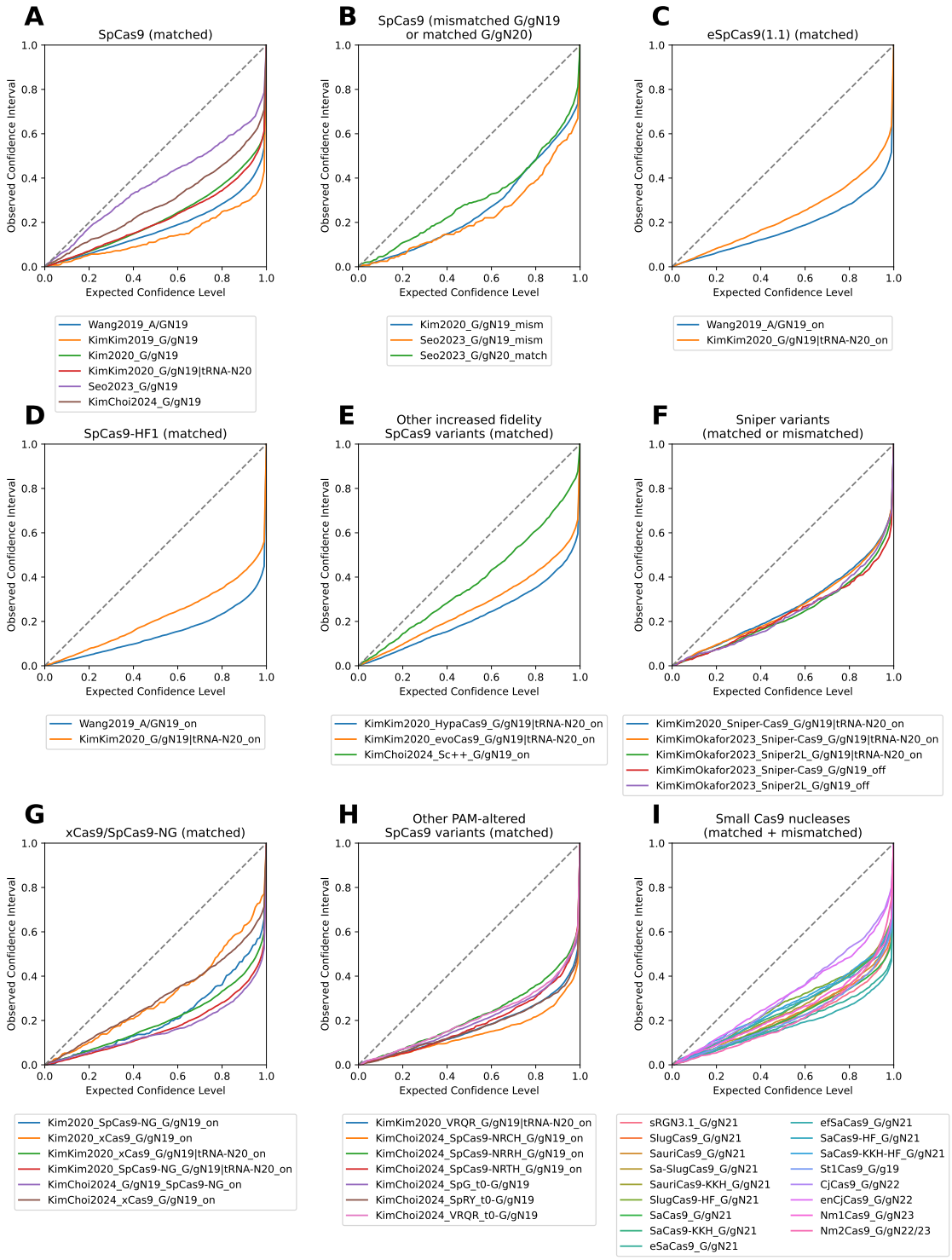


Figure C.35: Confidence interval-based calibration curves for DeepEnsEmbCas9_naive, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Quantile calibration error

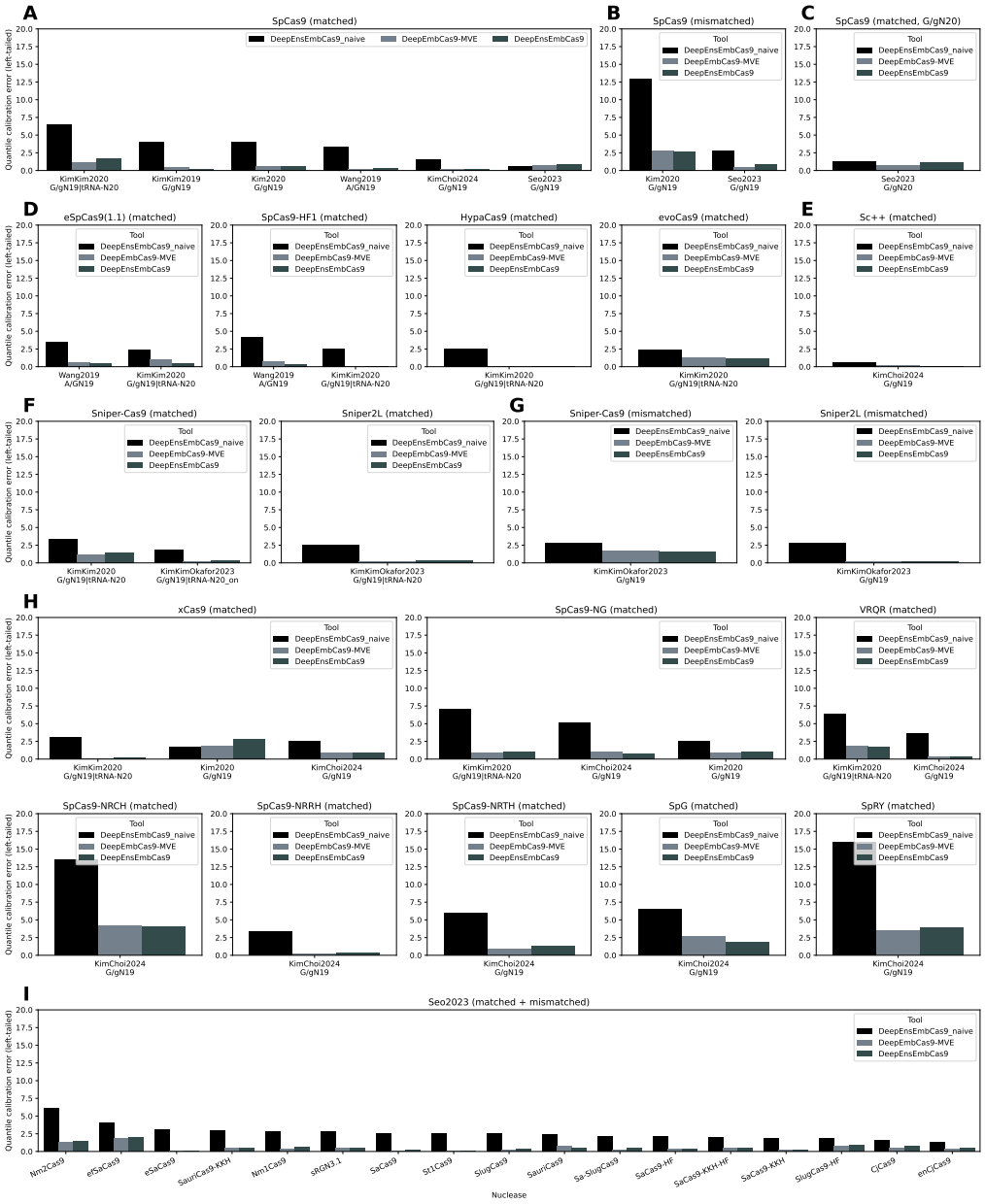


Figure C.36: Quantile calibration errors for DeepEnsEmbCas9_naive, DeepEmbCas9-MVE and DeepEnsEmbCas9, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

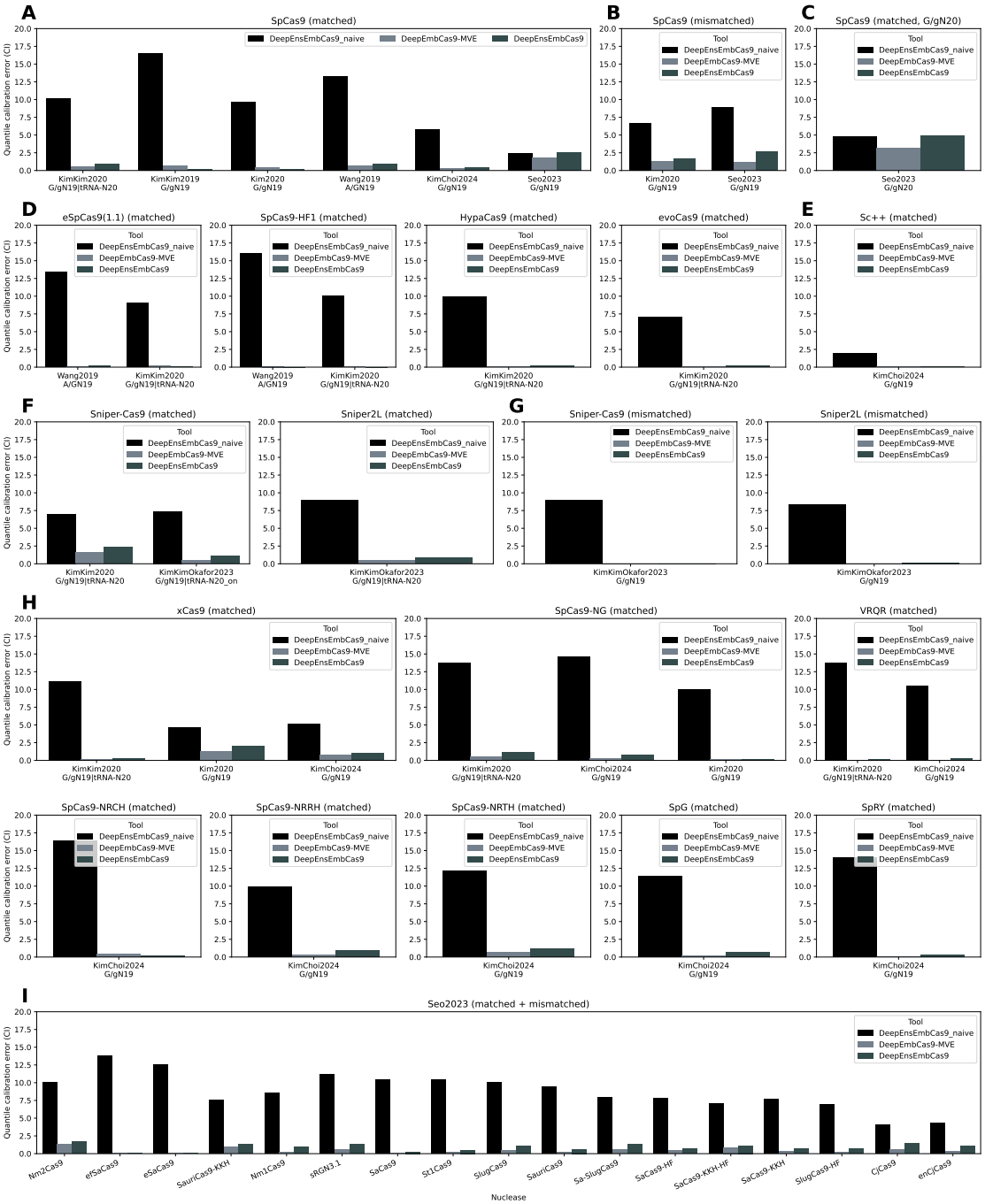


Figure C.37: Confidence interval-based quantile calibration errors for DeepEnsEmbCas9_naive, DeepEmbCas9-MVE and DeepEnsEmbCas9, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

C.2.6 Per-nuclease extrapolation calibration

Quantile calibration plots - DeepEnsEmbCas9_omit

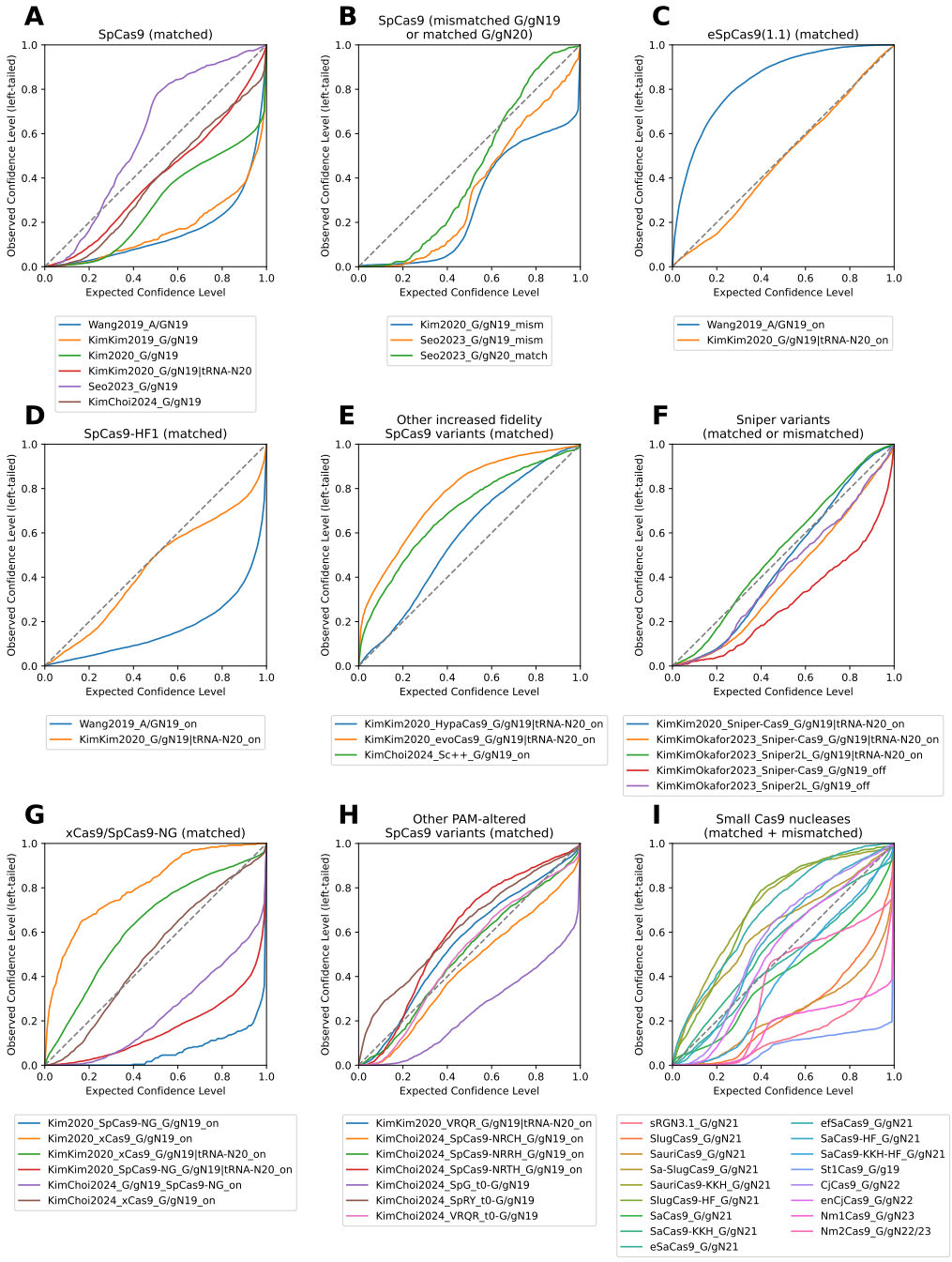


Figure C.38: Quantile calibration plots for DeepEnsEmbCas9_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

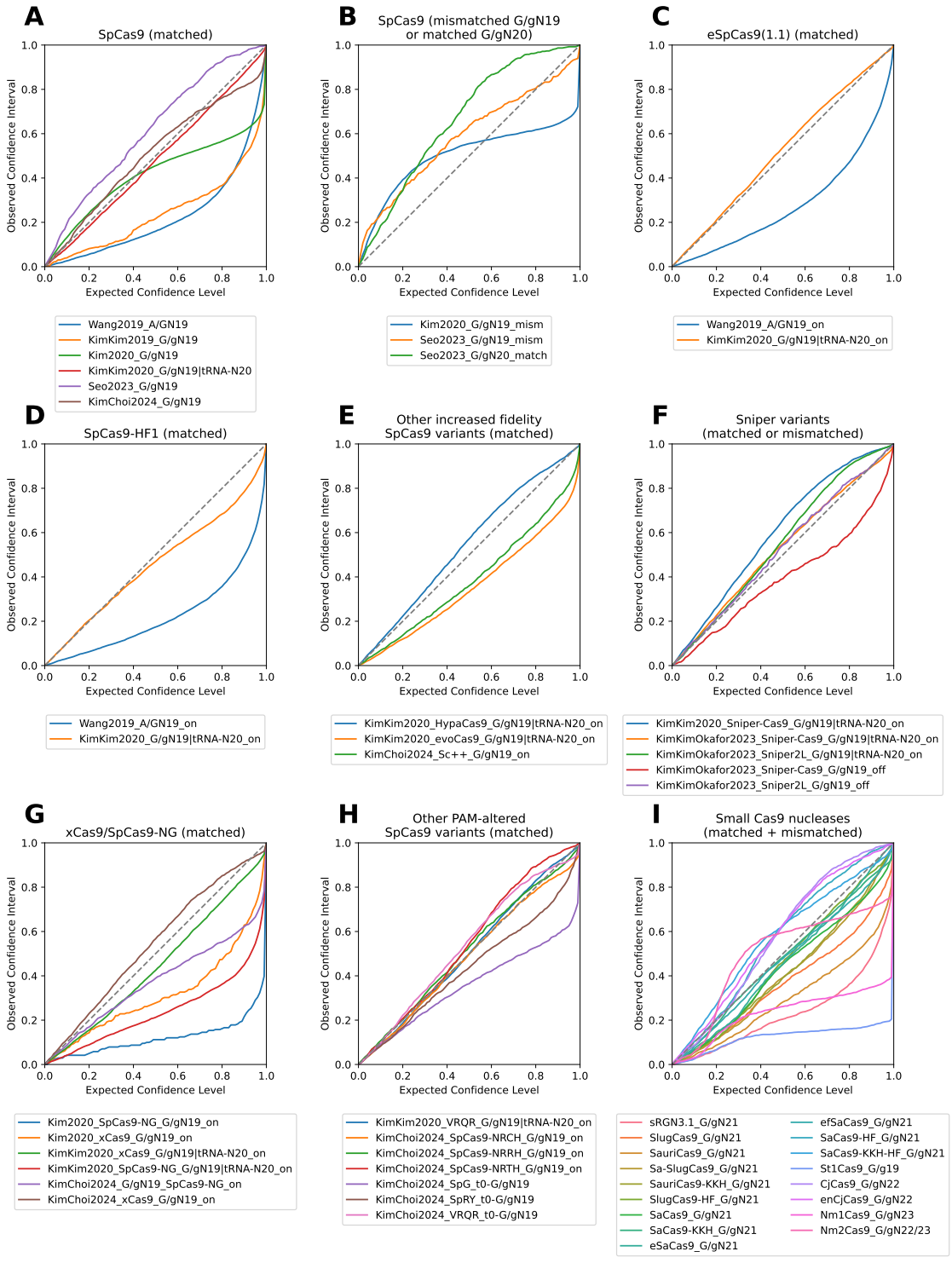


Figure C.39: Confidence interval-based calibration curves for DeepEnsEmbCas9_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Quantile calibration plots - DeepEmbCas9-MVE_omit

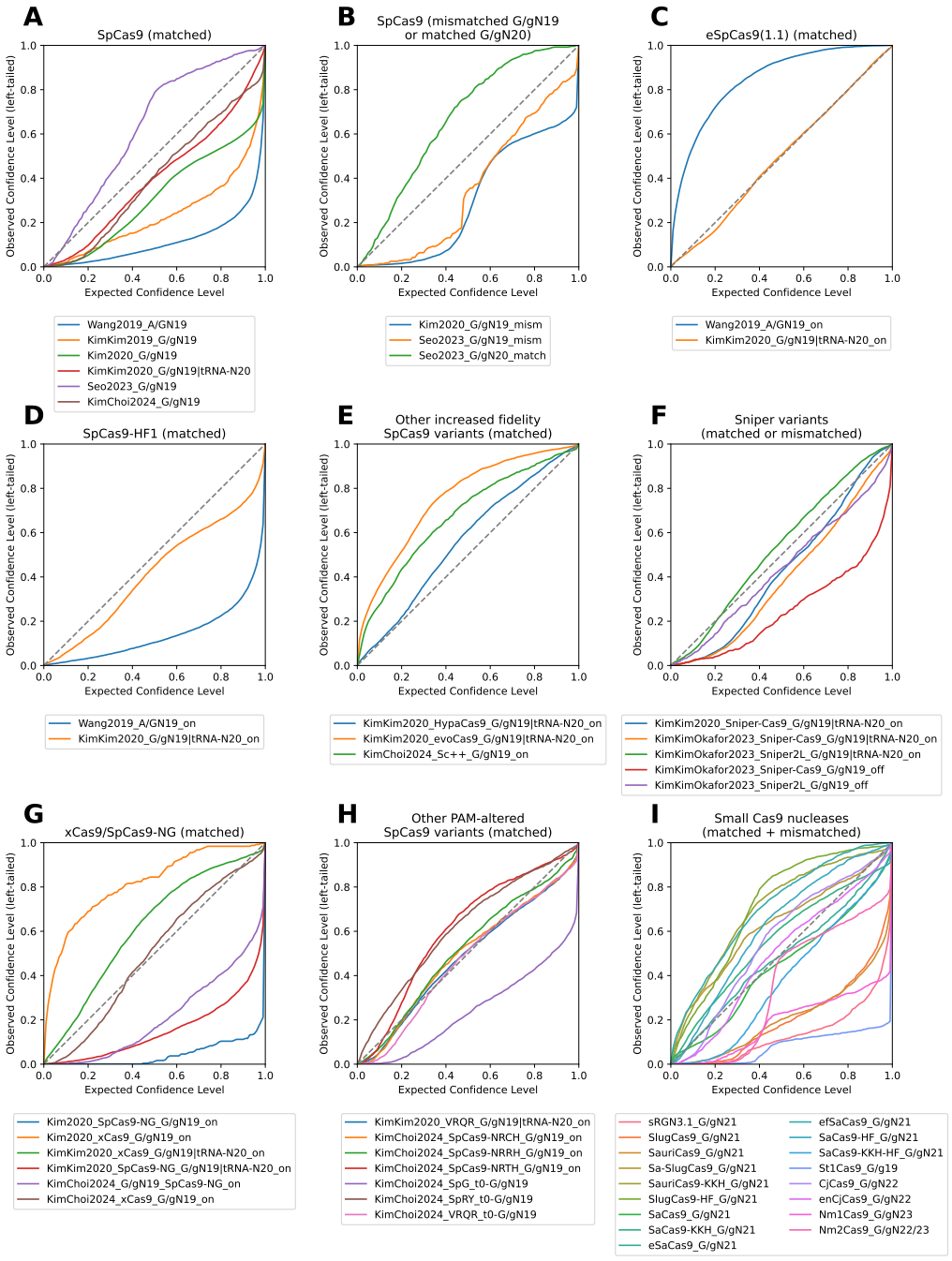


Figure C.40: Quantile calibration plots for DeepEmbCas9-MVE_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

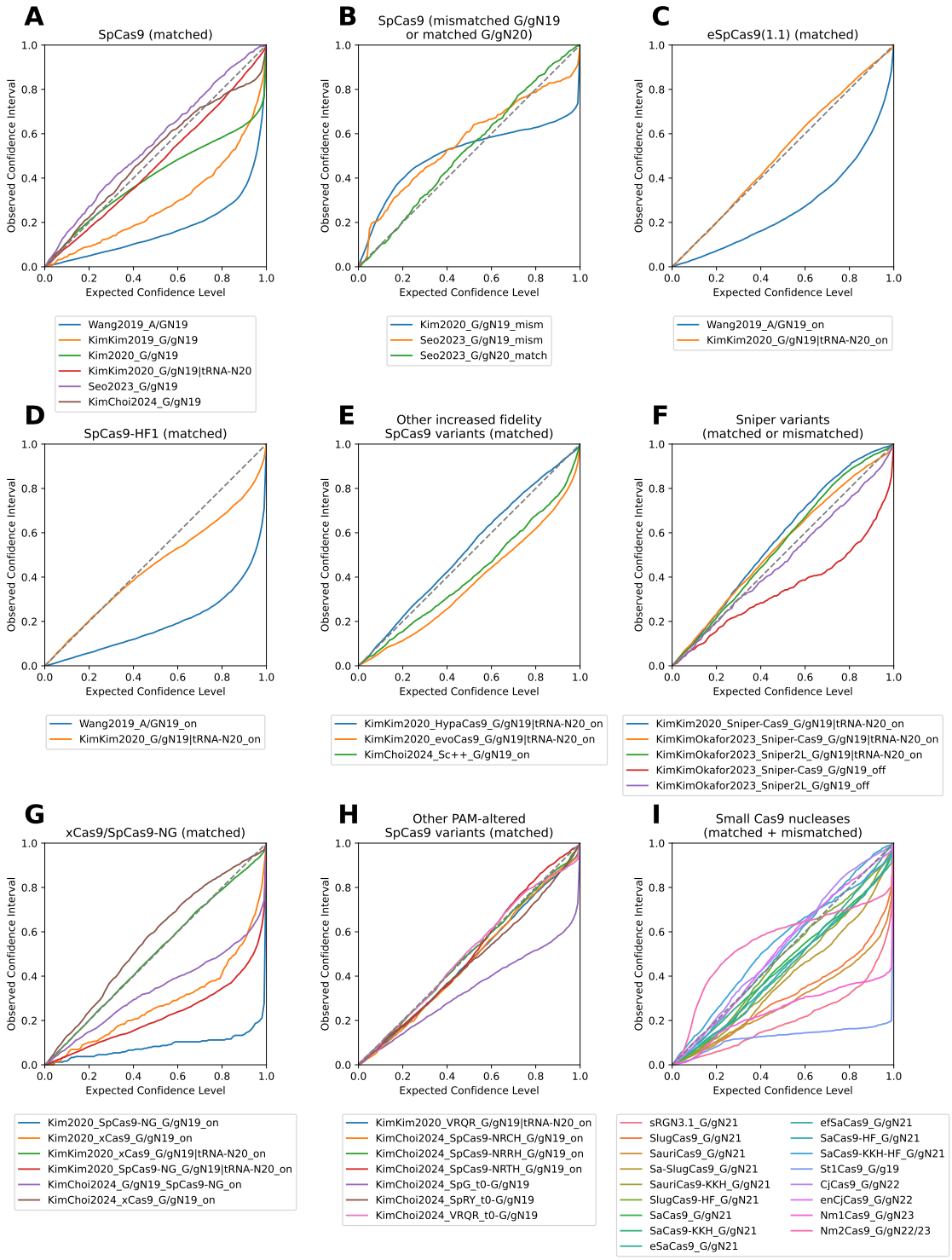


Figure C.41: Confidence interval-based calibration curves for DeepEmbCas9-MVE_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Quantile calibration plots - DeepEnsEmbCas9_naive_omit

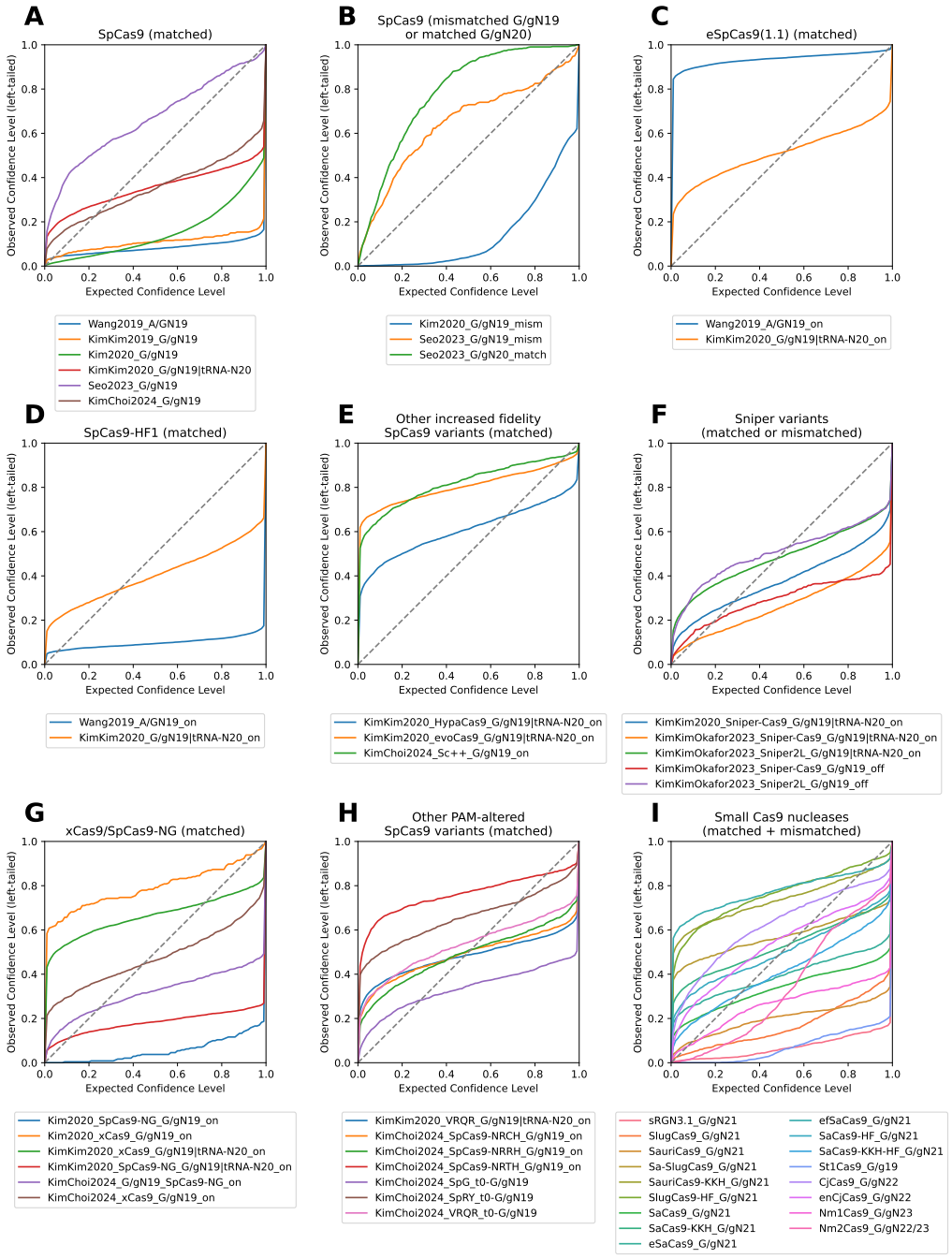


Figure C.42: Quantile calibration plots for DeepEnsEmbCas9_naive_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

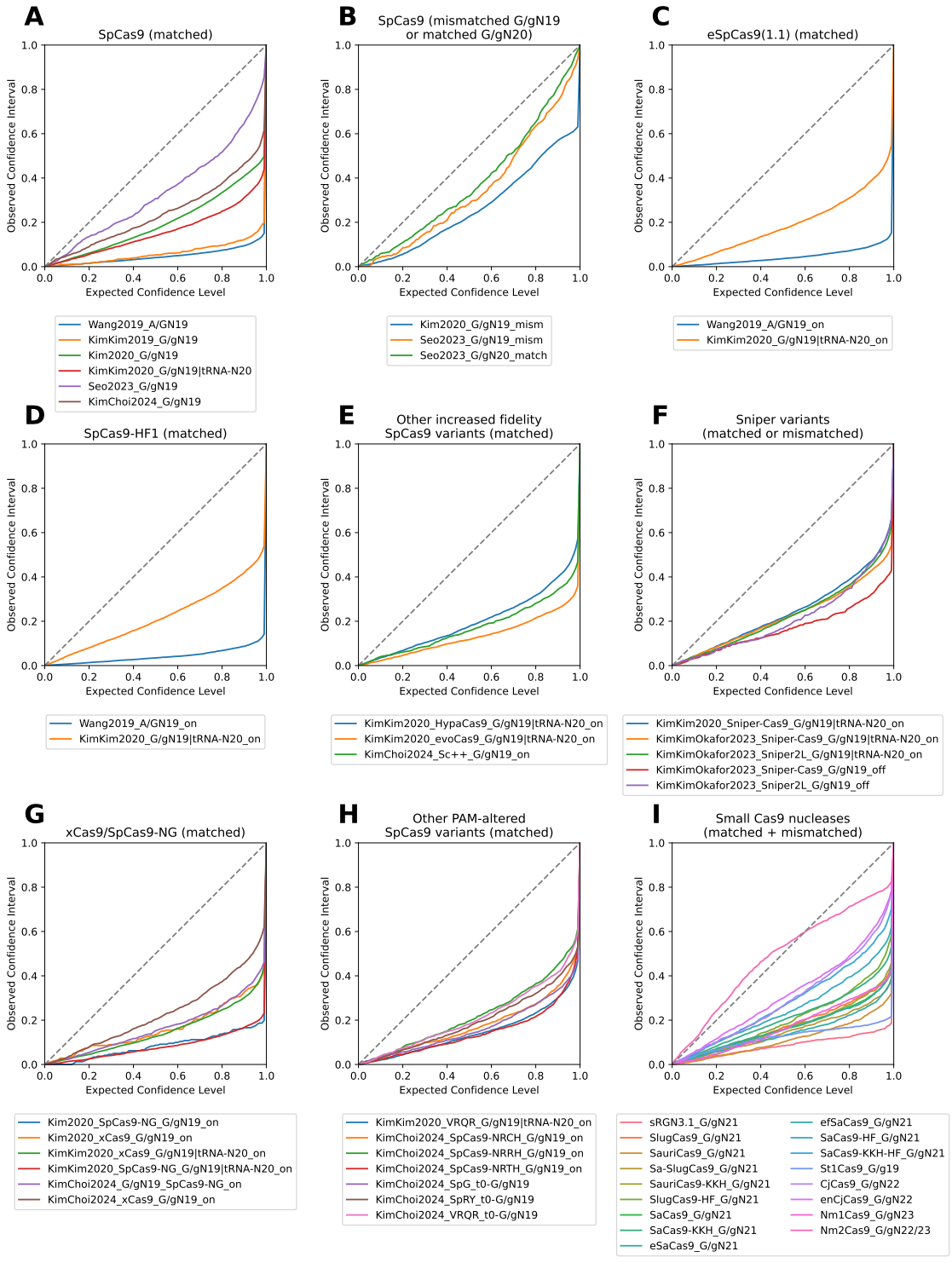


Figure C.43: Confidence interval-based calibration curves for DeepEnsEmb-Cas9_naive_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Quantile calibration error

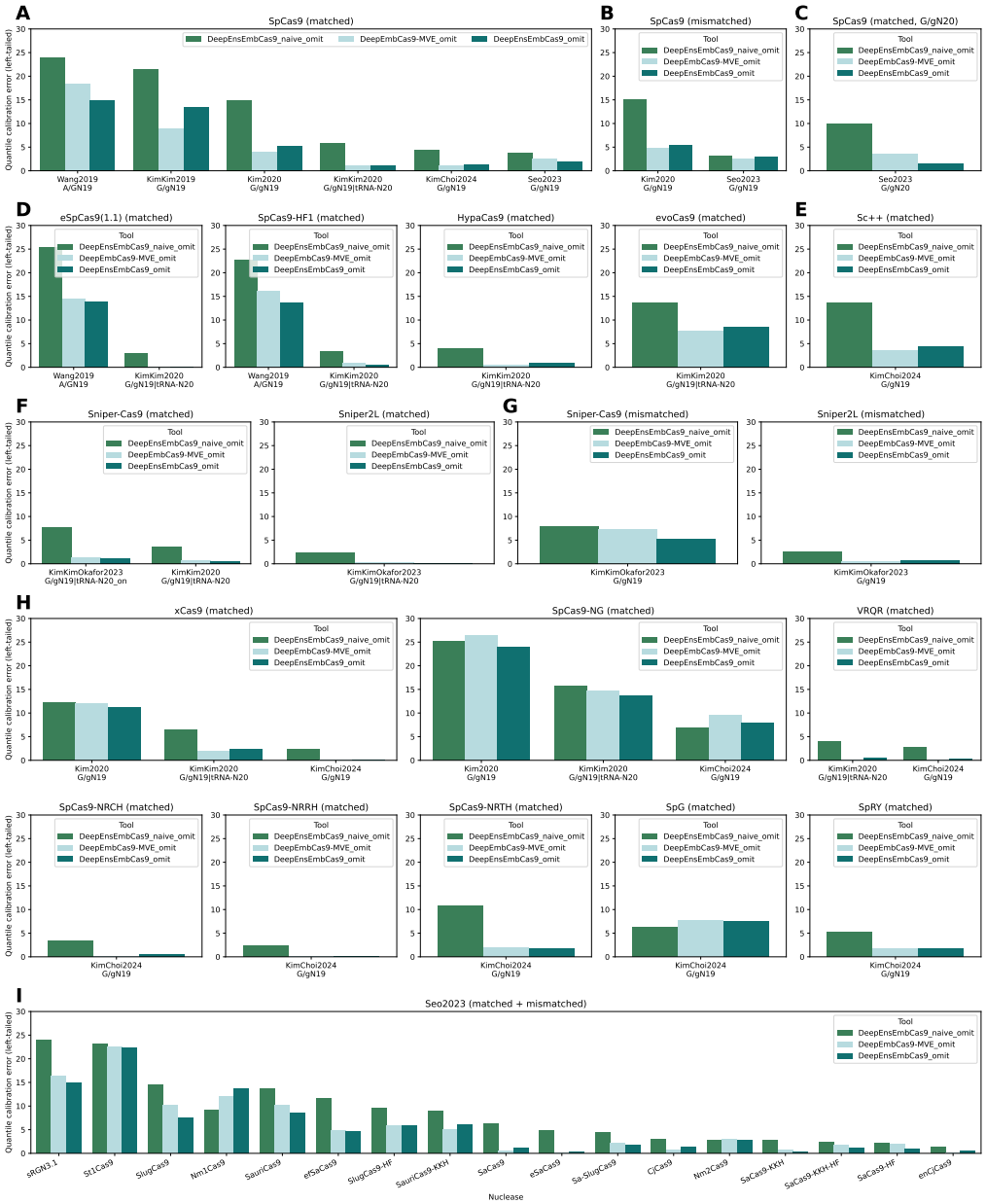


Figure C.44: Quantile calibration errors for DeepEnsEmbCas9_naive_omit, DeepEmbCas9-MVE_omit and DeepEnsEmbCas9_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

Figure C.45: Confidence interval-based quantile calibration errors for DeepEnsEmbCas9_naive_omit, DeepEmbCas9-MVE_omit and DeepEnsEmbCas9_omit, conditioned on (A) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} wild type SpCas9 interfaces; (B) mismatched G/gN_{19} and matched G/gN_{20} wild type SpCas9 interfaces; (C) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} eSpCas9(1.1) interfaces; (D) matched AN_{19}, G/gN_{19} and $tRNA^{Gln}$-N_{20} SpCas9-HF1 interfaces; for matched G/gN_{19} and $tRNA^{Gln}$-N_{20} HypaCas9/evoCas9 and G/gN_{19} Sc++ interfaces; (F) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} and mismatched G/gN_{19} interfaces for 2 Sniper variants; (G,H) matched G/gN_{19} and $tRNA^{Gln}$-N_{20} interfaces for xCas9/SpCas9-NG (G) and 6 other PAM-altered SpCas9 variants (H); and (I) matched and mismatched interfaces for 17 wild type or engineered small Cas9 nucleases.

C.2.7 Model Interpretation

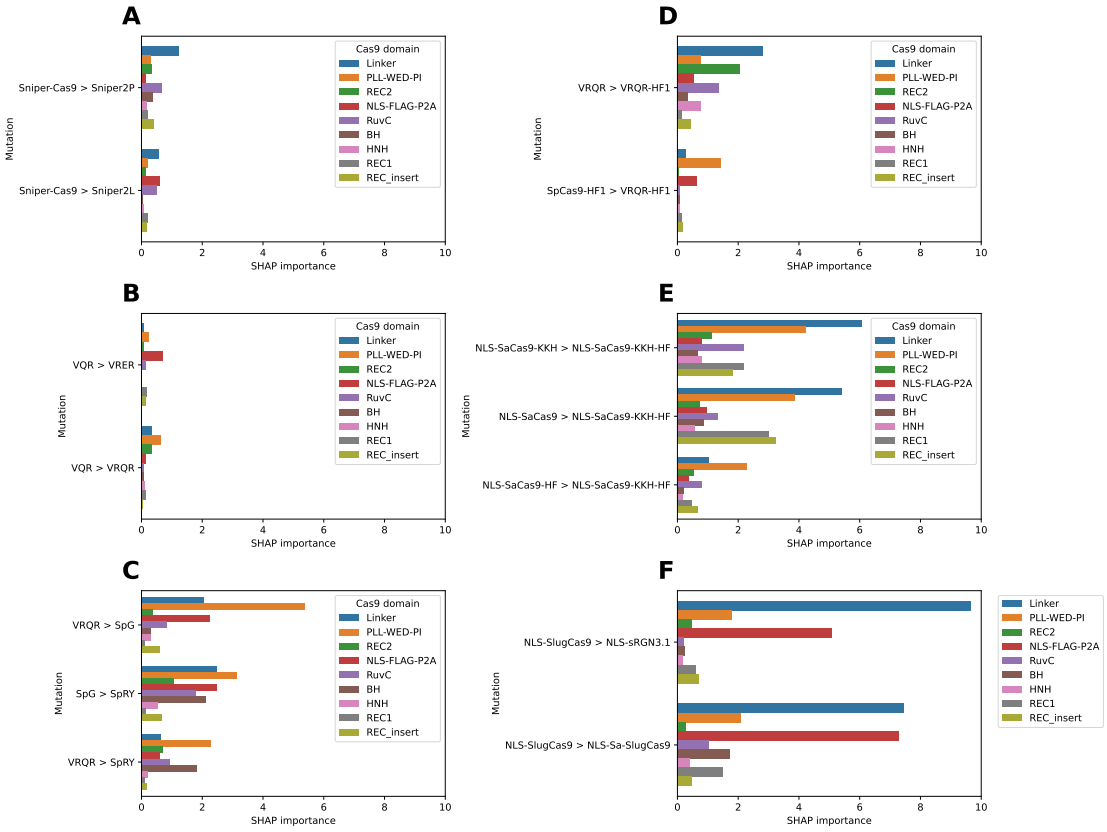


Figure C.46: Important Cas9 domains driving DeepEmbCas9's change in predicted activity when (A) mutating Sniper-Cas9 into Sniper2 variants; (B) mutating VQR into VRER and VRQR; (C) mutating into PAM-relaxed/PAMless SpCas9 variants; (D) mutating to VRQR-HF1 from VRQR and SpCas9-HF1; (E) mutating SaCas9-KKH, SaCas9 and SaCas9-HF to SaCas9-KKH-HF; and (F) mutating SlugCas9 into sRGN3.1 and Sa-SlugCas9.

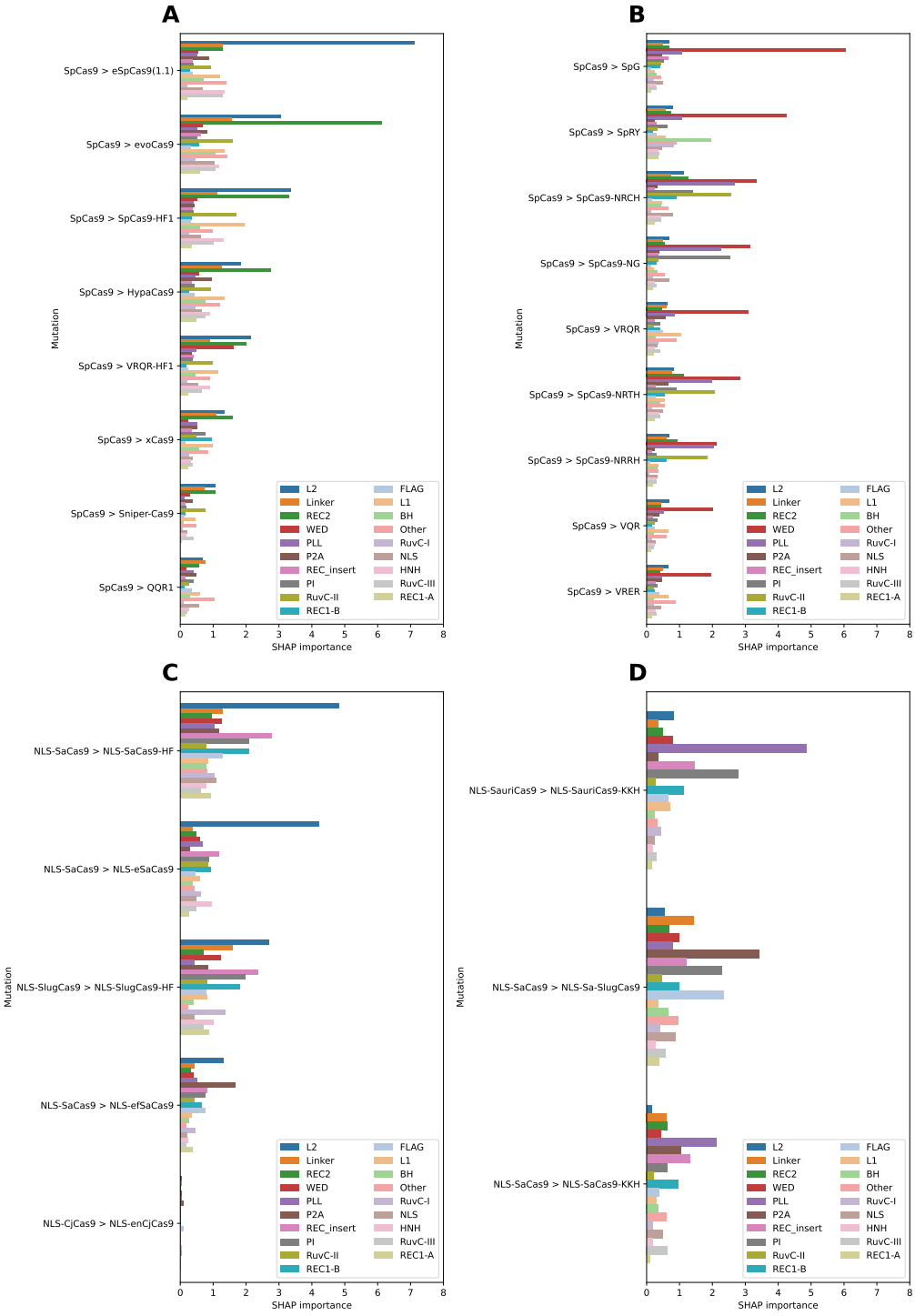


Figure C.47: CRISPR-Cas9 Cas9 regions driving DeepEmbCas9's change in predicted activity when introducing residue mutations in Cas9. (A,B) Cas9 region importances for SpCas9 variants with (A) and without (B) D1135 mutations. (C,D) Cas9 region importances for small Cas9 variants with (C) increased fidelity and (D) PAM-altering variants.

Feature group	Description of features	No. of features
spacer + spacer_MFE + spacer_GCcount	spacer one-hot encoding	4×42
	spacer MFE	1
	spacer GC count	1
	sgRNA rLM embedding for spacer region	768
upstream + protospacer + protospacer_Tm	5' upstream and protospacer target one-hot encoding	4×27
	protospacer DNA melting temperatures	21
	protospacer GC count	1
PAM + downstream + PAM_downstream_Tm	PAM and 3' downstream one-hot encoding	4×15
	PAM melting temperatures	6
	3' downstream DNA melting temperatures	9
tRNA preprocessing	tRNA feature	1
Day	Day	1
Cas9_ESM-C-600M_RuvC-I	Cas9 pLM embedding features for the RuvC-I region	960
Cas9_ESM-C-600M_BH	Cas9 pLM embedding features for BH region	960
Cas9_ESM-C-600M_REC1-A	Cas9 pLM embedding features for the REC1-A region	960
Cas9_ESM-C-600M_REC_insert	Cas9 pLM embedding features for the REC_insert region	960
Cas9_ESM-C-600M_REC1-B	Cas9 pLM embedding features for the REC1-B region	960
Cas9_ESM-C-600M_REC2	Cas9 pLM embedding features for the REC2 region	960
Cas9_ESM-C-600M_Linker	Cas9 pLM embedding features for the Linker region	960
Cas9_ESM-C-600M_RuvC-II	Cas9 pLM embedding features for the RuvC-II region	960
Cas9_ESM-C-600M_L1	Cas9 pLM embedding features for the L1 region	960
Cas9_ESM-C-600M_HNH	Cas9 pLM embedding features for the HNH region	960
Cas9_ESM-C-600M_L2	Cas9 pLM embedding features for the L2 region	960
Cas9_ESM-C-600M_RuvC-III	Cas9 pLM embedding features for the RuvC-III region	960
Cas9_ESM-C-600M_PLL	Cas9 pLM embedding features for the PLL region	960
Cas9_ESM-C-600M_WED	Cas9 pLM embedding features for the WED region	960
Cas9_ESM-C-600M_PI	Cas9 pLM embedding features for the PI region	960
Cas9_ESM-C-600M_NLS	Cas9 pLM embedding features for the NLS region	960
Cas9_ESM-C-600M_FLAG	Cas9 pLM embedding features for the FLAG region	960
Cas9_ESM-C-600M_P2A	Cas9 pLM embedding features for the P2A region	960
Cas9_ESM-C-600M_Other	Cas9 pLM embedding features for the Other region	960
sgRNA_BEACON-B_repeat-antirepeat	sgRNA rLM embedding features for the repeat-antirepeat region	960
sgRNA_BEACON-B_tracrRNA-rest	sgRNA rLM embedding features for the repeat-antirepeat region	960
sgRNA_BEACON-B_polyT	sgRNA rLM embedding features for the polyT region	960

Table C.6: List and descriptions of fine resolution CRISPR-Cas9 complex component feature groups used in SHAP importance analysis.

Feature Group	Features	No. of features
spacer	spacer one-hot encoding	4×42
	spacer MFE	1
	$\frac{1}{2}\times$ sgRNA MFE	0.5
	spacer GC count	1
	sgRNA rLM embedding for spacer region	768
target	target context sequence one-hot encoding	4×42
	DNA melting temperature features	36
	protospacer GC count	36
Cas9	Cas9 pLM embedding for all Cas9 regions	
scaffold	sgRNA rLM embedding for repeat-antirepeat region	768
	sgRNA rLM embedding for tracrRNA-rest region	768
	sgRNA rLM embedding for polyT region	768
	$\frac{1}{2}\times$ sgRNA MFE	0.5
tRNA preprocessing	tRNA feature	1
Day	Day	1

Table C.7: List and descriptions of coarse resolution CRISPR-Cas9 complex component feature groups used in SHAP importance analysis.

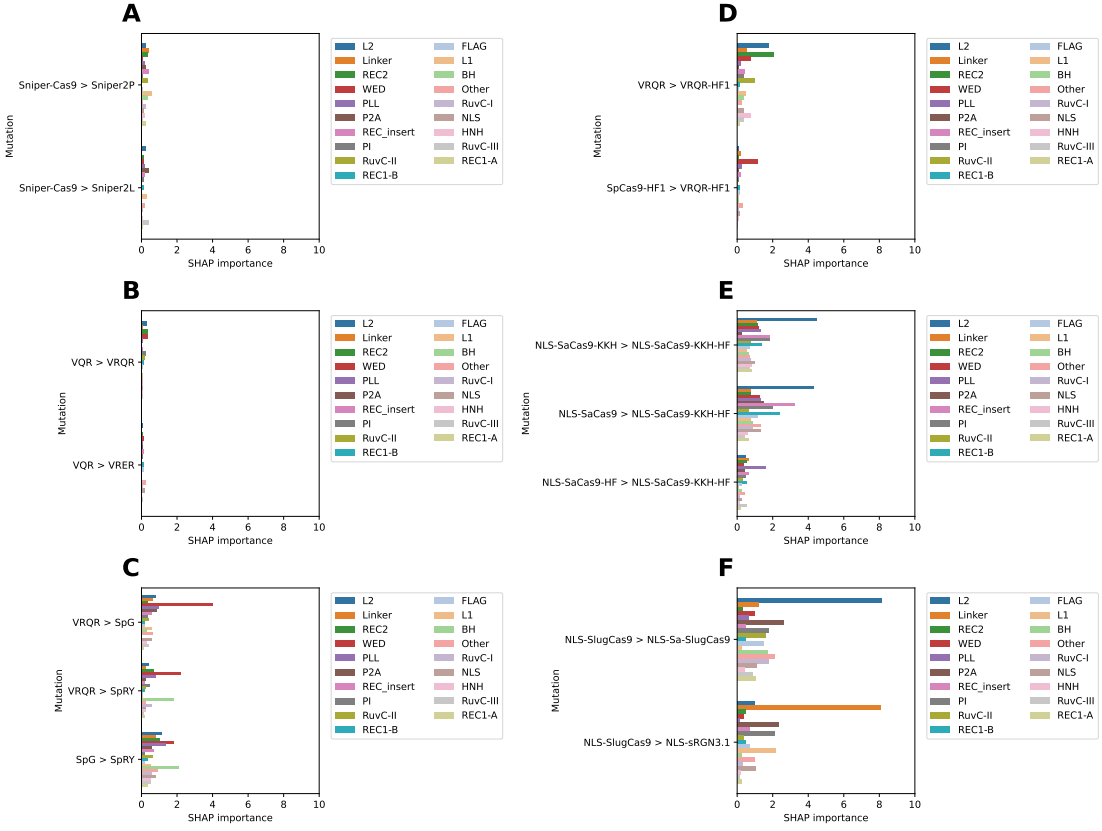


Figure C.48: Important Cas9 regions driving DeepEmbCas9's change in predicted activity when (A) mutating Sniper-Cas9 into Sniper2 variants; (B) mutating VQR into VRER and VRQR; (C) mutating into PAM-relaxed/PAMless SpCas9 variants; (D) mutating to VRQR-HF1 from VRQR and SpCas9-HF1; (E) mutating SaCas9-KKH, SaCas9 and SaCas9-HF to SaCas9-KKH-HF; and (F) mutating SlugCas9 into sRGN3.1 and Sa-SlugCas9.

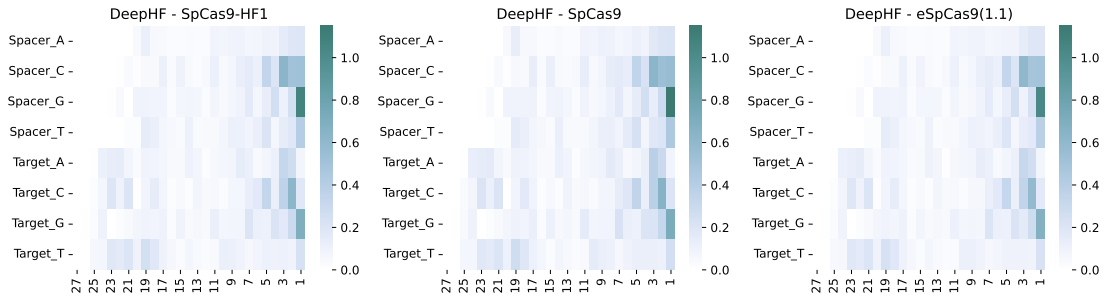


Figure C.49: SHAP importance of spacer-target one-hot encoding features in driving DeepEmbCas9's change in predicted activity for Cas9 variants from Wang et al. [4].

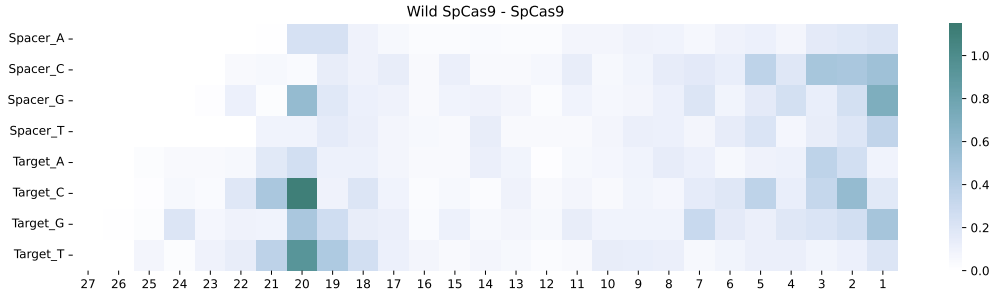


Figure C.50: SHAP importance of spacer-target one-hot encoding features in driving DeepEmbCas9's change in predicted activity for Cas9 variants from Kim, Kim et al. [5].

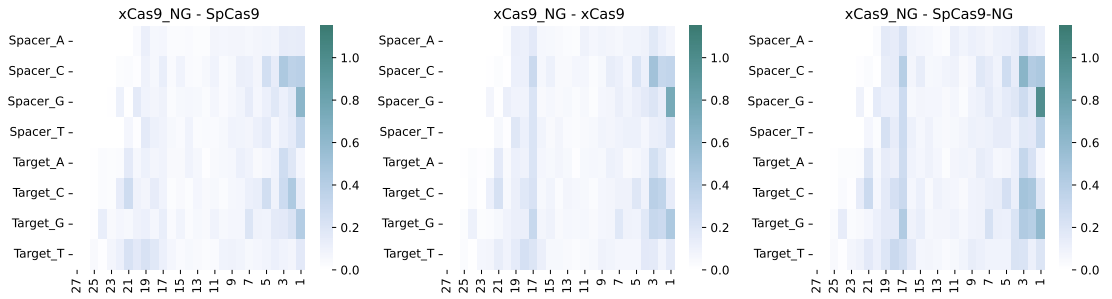


Figure C.51: SHAP importance of spacer-target one-hot encoding features in driving DeepEmbCas9's change in predicted activity for Cas9 variants from Kim et al. [6].

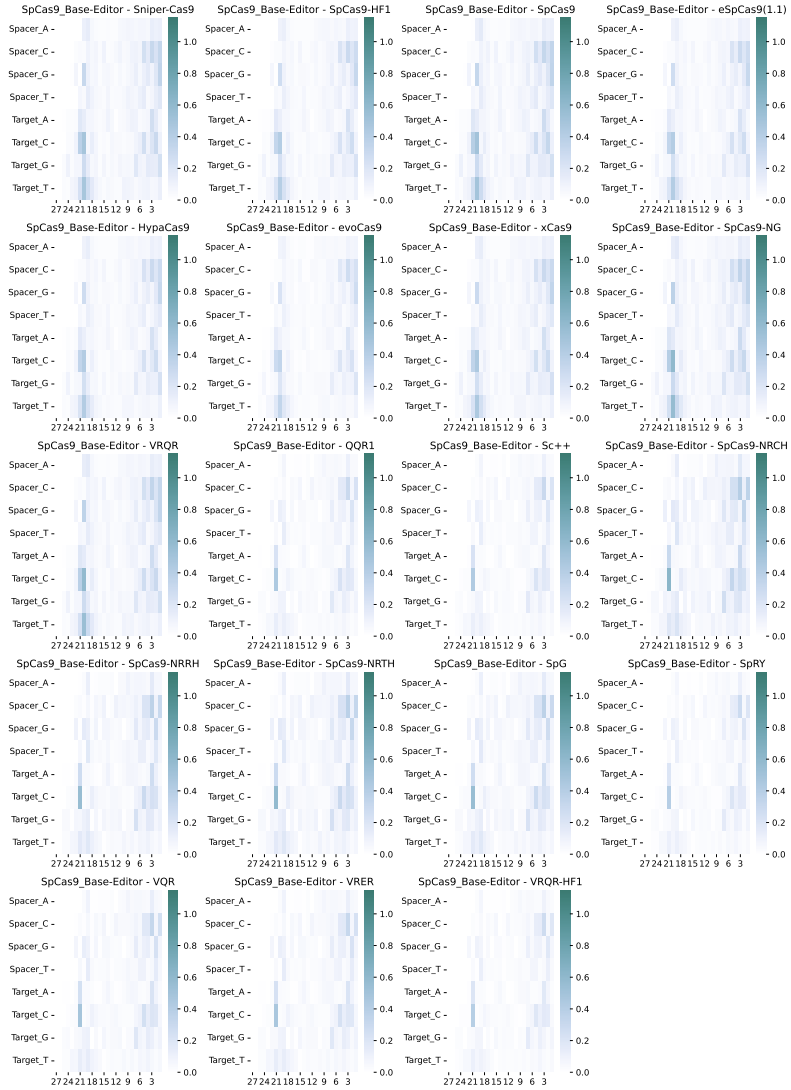


Figure C.52: SHAP importance of spacer-target one-hot encoding features in driving DeepEmbCas9's change in predicted activity for Cas9 variants from Kim, Kim et al. [7] and Kim, Choi et al. [9].

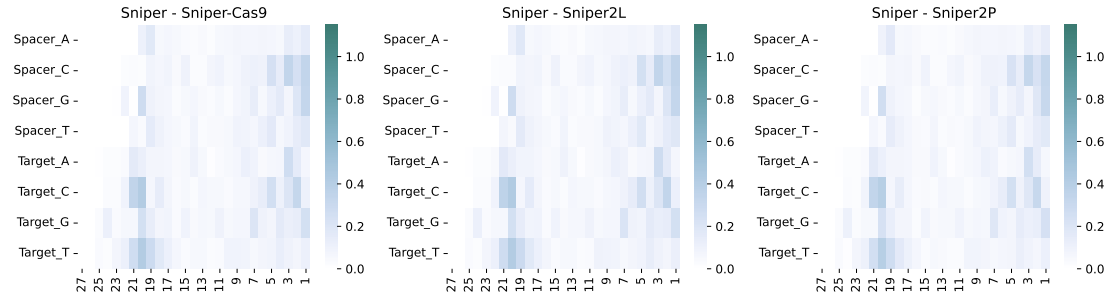


Figure C.53: SHAP importance of spacer-target one-hot encoding features in driving DeepEmbCas9's change in predicted activity for Cas9 variants from Kim, Kim, Okafor et al. [9].

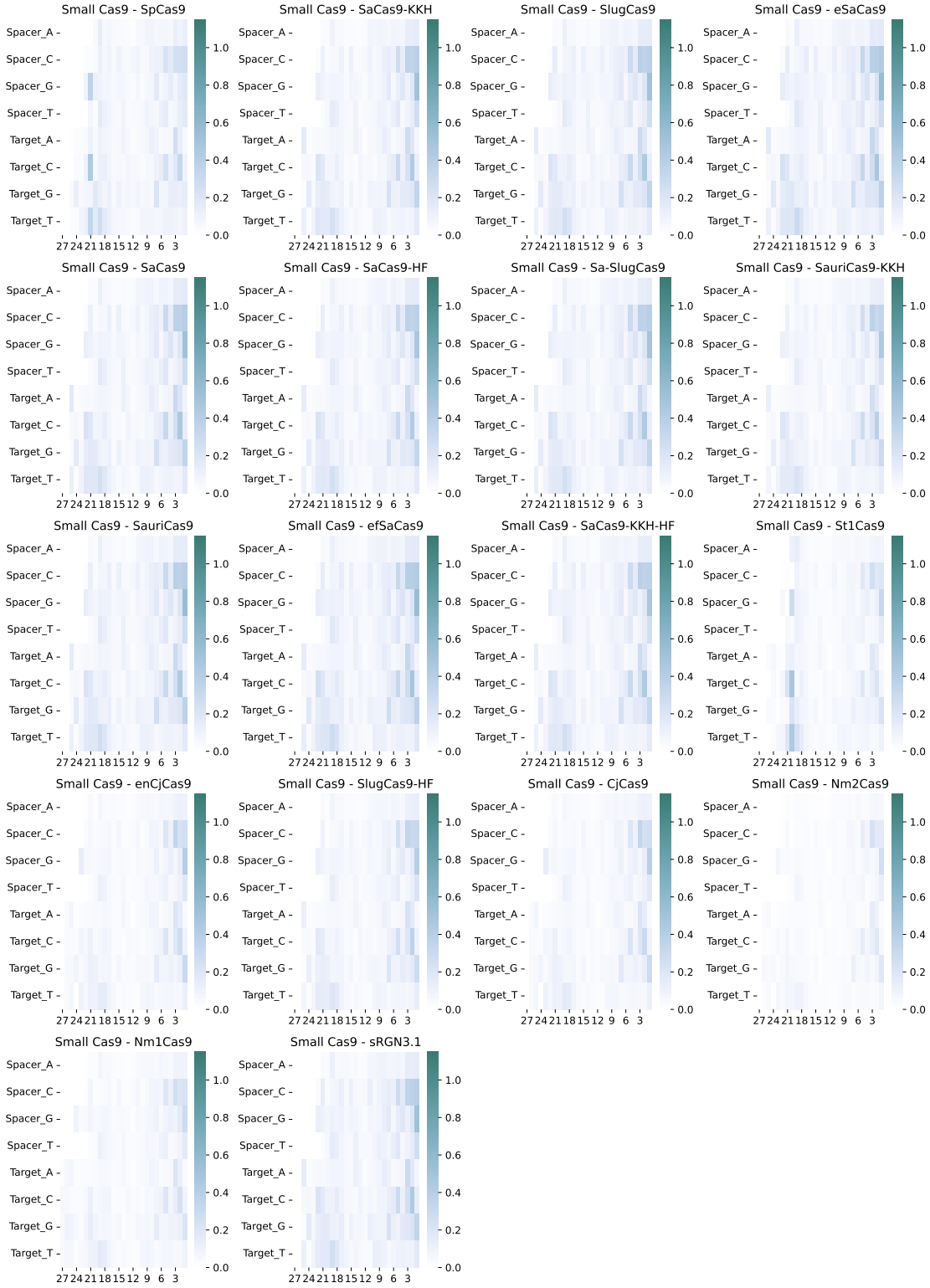


Figure C.54: SHAP importance of spacer-target one-hot encoding features in driving DeepEmbCas9's change in predicted activity for Cas9 variants from Kim, Kim, Okafor et al. [9].

Appendix D

Other published work

D.1 Physically-inspired modelling of CRISPR-Cas9 cleavage activity prediction

I have contributed to the research article "piCRISPR: Physically informed deep learning models for CRISPR/Cas9 off-target cleavage prediction", which combines the nucleosomal scores produced in Chapter 2 with spacer-target nucleotides features and R-loop formation energy scores for SpCas9 on- and off-target activity prediction using data from the crisprSQL database [54]. The publication is reprinted in the forthcoming pages.

Artificial Intelligence in the Life Sciences 3 (2023) 100075

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Research Article

piCRISPR: Physically informed deep learning models for CRISPR/Cas9 off-target cleavage prediction

Florian Störtz, Jeffrey K. Mak, Peter Minary*

Department of Computer Science, University of Oxford, Parks Road, Oxford OX1 3QD, UK

ARTICLE INFO

Keywords:
CRISPR
Cas9
Deep learning
Cleavage prediction
Nucleosome organisation

ABSTRACT

CRISPR/Cas programmable nuclease systems have become ubiquitous in the field of gene editing. With progressing development, applications in *in vivo* therapeutic gene editing are increasingly within reach, yet limited by possible adverse side effects from unwanted edits. Recent years have thus seen continuous development of off-target prediction algorithms trained on *in vitro* cleavage assay data gained from immortalised cell lines. It has been shown that in contrast to experimental epigenetic features, computed physically informed features are so far underutilised despite bearing considerably larger correlation with cleavage activity. Here, we implement state-of-the-art deep learning algorithms and feature encodings for off-target prediction with emphasis on *physically informed* features that capture the biological environment of the cleavage site, hence terming our approach piCRISPR. Features were gained from the large, diverse crisprSQL off-target cleavage dataset. We find that our best-performing models highlight the importance of sequence context and chromatin accessibility for cleavage prediction and compare favourably with literature standard prediction performance. We further show that our novel, environmentally sensitive features are crucial to accurate prediction on sequence-identical locus pairs, making them highly relevant for clinical guide design. The source code and trained models can be found ready to use at github.com/florianst/picrispr.

Introduction

The clustered regularly interspaced short palindromic repeats (CRISPR) sequence family was first described in *E. coli* in 1987 [1], but it took until 2007 to recognise it as a part of the viral defense system of most archaea and bacteria [2]. Exogenous viral DNA is cleaved off by specialised nuclease enzymes, coded for on genomic regions which are often adjacent to CRISPR and hence named CRISPR-associated (Cas). Cleaved-off regions are subsequently incorporated into the CRISPR sequences, which act as a viral history of the respective cell, stabilised by the palindromic nature of their saved states which results in stable secondary structures [3]. From there they can be transcribed to crRNA and invading copies of them can subsequently be rendered inactive. Researchers have used this ability for programmable genome editing in many eukaryotic species, complementing strategies such as zinc-finger nucleases (ZFNs, [4]) and transcription activator-like effector nucleases (TALENs, [5]).

We concentrate on the effects of the wild-type Cas9 protein gained from *Streptococcus pyogenes*. The crRNA which is originally responsible for recognition of a 20bp viral sequence forms an active complex with the tracrRNA, called single guide RNA (sgRNA), of about 50bp length [6]. Homology of the crRNA part with a 20bp region in the genome results in annealing of the sgRNA with one strand of this region, which we call 'target strand'. Binding happens when the interaction of the 3bp protospacer-adjacent motif (PAM) on the opposite, non-target strand with the Cas9 protein is favourable [7]. For *S. pyogenes* Cas9, this is the case for an 'NGG' PAM where N stands for an arbitrary nucleobase (A, T, C, G).

Tertiary DNA structure, such as nucleosome octamers, can occlude or expose different regions of DNA and hinder Cas9 access [8]. After binding has taken place, nuclease-active enzymes within Cas9 can cleave the double-stranded DNA 3bp upstream of the PAM. Due to the stochastic, energy-driven nature of both the binding and the cleavage process, we expect a distribution of cuts over the whole genome, including undesired off-target effects which could possibly have catastrophic consequences, such as knocking out tumor suppressor genes like p53 and Rb [9].

We noticed that repositories of off-target cleavage data contain a significant amount of data points which match in both guide and (off-)target sequence and differ only in the biological environment of the respective loci (see Fig. 1). Capturing this environment is therefore instrumental in providing accurate predictions of cleavage activity. We recently found that computed nucleosome organisation-related features correlate better with cleavage frequency values than experimental epigenetic markers (Deoxyribonuclease-I hypersensitive sites sequencing

* Corresponding author.
E-mail address: peter.minary@cs.ox.ac.uk (P. Minary).

https://doi.org/10.1016/j.ailsci.2023.100075
Received 29 December 2022; Received in revised form 2 April 2023; Accepted 30 April 2023
Available online 15 May 2023

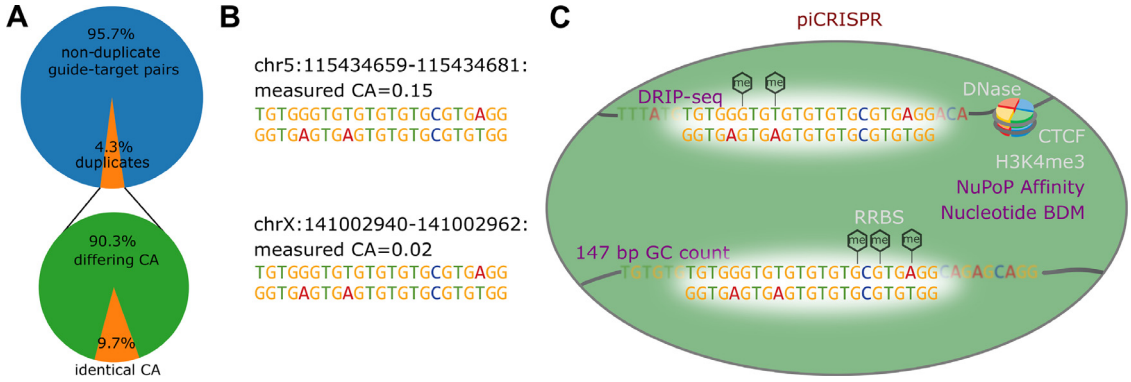


Fig. 1. **A** The crisprSQL dataset contains an appreciable amount of perfect guide-target duplicates. We only consider data gained from human cell lines and putative off-targets which we generated based on sequence similarity. 8922 of 230,274 data points have at least one guide-target duplicate within this set which differs in cleavage activity (CA). As the example from our dataset in panel **B** shows, such a pair looks identical to purely sequence-based prediction algorithms. They might therefore not predict dangerous off-target effects. **C** piCRISPR remedies this by taking into account the biological environment of the cleavage site based on a range of features beyond guide and (off-)target sequence. So far, prediction algorithms have used features related to chromatin organisation (CTCF, [20]), chromatin accessibility (DNase-Seq), DNA methylation (RRBS) and histone methylation (H3K4me3, [21]). We on the other hand use features pertaining to the 147 bp sequence context around each (off-)target nucleotide: GC count, sequence complexity (BDM, [19]) and nucleosome positioning information (NuPoP, [8]) which introduce unprecedented sensitivity to the biological environment of the cleavage site. Using these, piCRISPR can correctly rank the two example loci given here.

/ DNase, reduced representation bisulfite sequencing / RRBS, CCCTC-binding factor / CTCF, histone-3 lysine-4 trimethylation / H3K4me3) [10] which have heretofore been the literature standard for cleavage prediction models. These computed features also surpassed epigenetic markers in terms of their feature importance in preliminary cleavage activity prediction models which had access to both computed and experimental epigenetic features. We therefore aim to make full use of this novel class of features by embedding them in a rich feature set, including DNA/RNA sequence and context sequence-based features.

With a considerable amount of cleavage prediction algorithms present in literature [11–15], we present here a choice of two model architectures, two encodings and two sets of features, yielding a total of eight combinations. We scrutinise these according to both prediction performance and interpretability. Besides improving prediction accuracy and capturing off-target effects that might so far have gone unnoticed, this will also generate insight into the biological environment that influences CRISPR cleavage.

Methods

Data Source

In order to achieve maximum transparency and comparability, we use guide-target pairs from the crisprSQL dataset [16] curated by our group. It is a collection of 17 base-pair resolved off-target cleavage studies on Sp-Cas9, comprising 25,632 data points and is larger than most datasets used to train prediction algorithms to date. It contains data on various cell lines, mainly U2OS, HEK293 and K562. We have chosen to use version 26/05/2020 of the database which does not include T-cell data from Lazzarotto *et al.* [17] in order to avoid introducing a considerable cell line imbalance. Furthermore, the evaluation of our modelling on-target datasets is beyond the scope of this work due to their different underlying experimental techniques and cleavage quantification measures.

Experimental data points containing guide and target loci, sequence, cell line, assay type and cleavage frequency have been completed and enriched by sequence context as well as the `CRISPRoff` score, an empirical estimate of the (off-)target binding energy [18].

Besides these established features, we propose the usage of nucleosome organisation-related features [10] which add an unprecedented level of sensitivity towards the biological environment of the cleavage site (see Fig. 1C). In that publication, we trained a preliminary cleavage prediction model on 13 distinct nucleosome organisation-related scores all based on the 147 bp context around each (off-)target nucleotide (see Supplementary Material) as well as the four literature-standard epigenetic markers named above. We here include the three scores that showed the highest feature importance there: GC count, Nucleotide BDM [19] and NuPoP Affinity [8].

GC count refers to the relative proportion of G and C nucleotides within the 147 base pair window centred around a given target nucleotide. Nucleotide BDM refers to a training-free method that approximates the algorithmic complexity of a DNA sequence. Low values of Nucleotide BDM have been shown to correlate with proximity to nucleosome dyad positions [19]. NuPoP refers to a duration Hidden Markov Model trained to predict the base-pair specific nucleosome affinity of a given (off-)target sequence. For the precise calculation of these three features, we refer to [10].

Data Augmentation

In order to increase the size of the training set, we extend it by those putative off-target sites along the respective genome which had fewer than seven mismatches to each respective guide sequence, omitting the (off-)target locus itself. It was ensured that the protospacer adjacent motif (PAM) at the very end of the guide sequence was either the canonical5′-NGG-3’ characteristic of SpCas9 [22], or one of the noncanonical forms5′-NGA-3’ and5′-NAG-3’ observed in [23]. If a genome-wide off-target detection method has not detected cleavage at a locus within the genome that satisfies these criteria, we deem the cleavage activity at this point to be zero. This yielded 310,142 total guide-target pairs, making the complete data set highly imbalanced. Sticking with the convention in literature, we refer to this process of extending the number of data points as *data augmentation*. For this work, we concentrated on the 251,854 data points originating from a human cell line or synthetic human DNA.

Labels For classification, we define the negative class as all data points with cleavage activity (CA) values below the lowest reported assay accuracy of 10^{-5}, combined with the set of putative off-targets. In order to achieve comparability between different studies for regression tasks, we perform a nonlinear Box–Cox transformation [24] to transform the cleavage rates to approximate a Gaussian with zero mean and variance $\sigma^2 = 2$, similar to the approach in [25] and [13]. Cleavage activity values below the lowest reported assay accuracy of 10^{-5} as well as putative off-targets were set to $-2\sigma^2 = -4$, and transformed values clipped to the $[-4, 4]$ range. This is an empirical choice based on the shape of the resulting distributions.

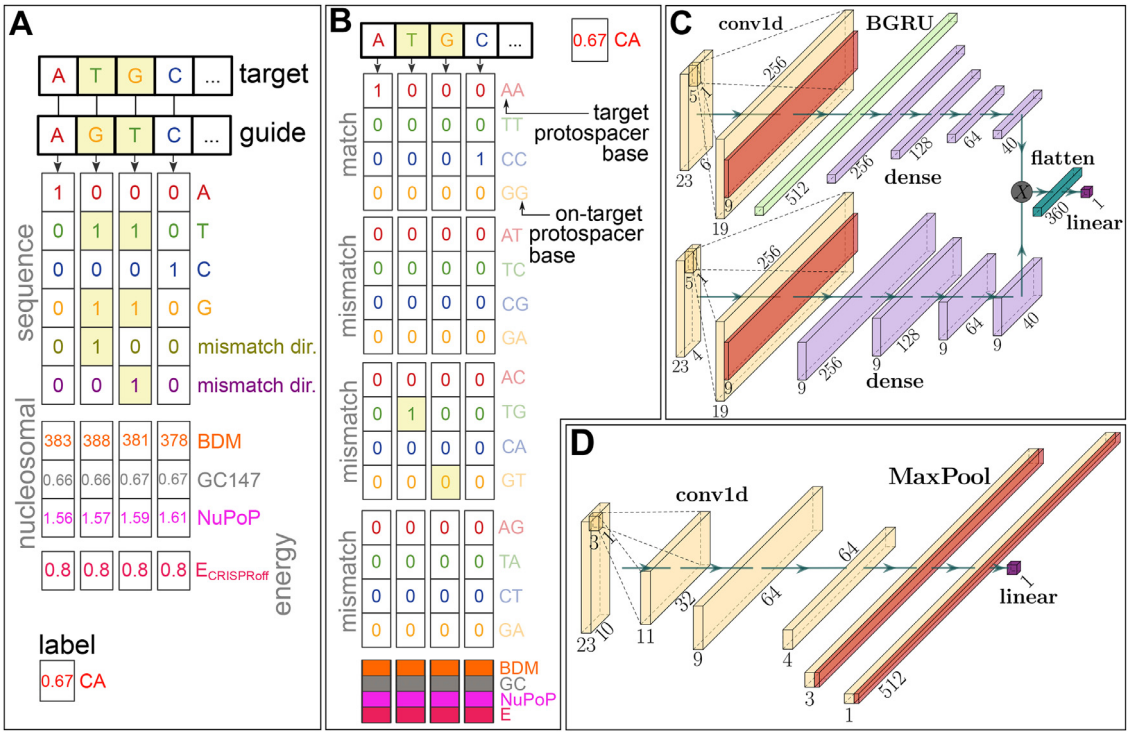


Fig. 2. Overview of our two feature encodings and model architectures. **A** 6×23 encoding: (Off-)target and guide (on-target protospacer) sequences are one-hot encoded and copied together using a bitwise OR operation. In order to make this encoding lossless, two channels are added that encode which nucleotide is on the guide and which on the target, respectively (in the case of a mismatched interface). This is identical to the encoding used in [13]. The nucleosome positioning channels (147 bp GC count, Nucleotide BDM [19], NuPoP Affinity [8]) of the target as well as the `CRISPRoff` free energy estimate score [18,27] are concatenated to this matrix. The Box–Cox transformed cleavage activity (CA) is used as a label. **B** The 16×23 encoding uses a 16-letter alphabet which explicitly contains information about the precise nature of the mismatch. **C** Bidirectional gated recurrent unit architecture as used in our RNN model, modelled after the network in [28]. Upper and lower arm of the network contain the sequence and nucleosomal/energy information, respectively. **D** Convolutional neural network architecture used in our CNN model, comparable with the model in [11]. Dimensions in both model architectures are valid for the 6×23 encoding (panel A).

Feature Encoding

We employ two different feature encoding schemes which occupy different points in the tradeoff between sparsity and interpretability. The first was introduced in [13] and consists of one-hot encoded representations of the guide and target sequence which have been combined using a bitwise OR operation. In order to make up for the loss of information that this operation causes in terms of mismatches, two additional channels are added containing information about the directionality of the bases involved in the mismatch, i.e. which of the two entries describes the target and which the guide nucleotide. Note that the guide nucleotide is first translated into its corresponding target protospacer nucleotide. We call this encoding the 6×23 encoding based on the resulting shape of the sequence matrix (see Fig. 2A).

Based on the energy-driven nature of binding and cleavage, we hypothesise that mismatched interfaces affect binding in a totally different way than matched interfaces. This has so far not been recognised in detail by off-target prediction algorithms. Since the 6×23 encoding contains the information about the precise nature of a given interface only implicitly, we decided to include a further encoding which does so explicitly. This uses a one-hot representation using the 16 letter cross product between guide and (off-)target nucleotide, and is hence termed the 16×23 encoding (see Fig. 2B). This is similar to the encoding scheme in [26].

Both matrices are then concatenated with a matrix of base-pair resolved nucleosomal features, as well as the `CRISPRoff` value of the given target-guide interface repeated along the sequence axis. Exploring latent representations of guide or target is not within the scope of this work, given that it further complicates comparison between models.

Model Architectures

Literature contains a wealth of model architectures commonly used to predict CRISPR cleavage. Currently, successful model architectures for learning-based cleavage prediction fall in one of three categories [29]: tree-based methods [25], convolutional neural networks (CNN, [11,12]) and recurrent neural networks (RNN, [13,15,28]). We take successful CNN and RNN architectures present in the field and adapt them to the off-target prediction task using various encodings of the features described above.

Our CNN model is comparable to the architecture described in [11]. There, the outputs of two separate, convolutional layer-based encoders for guide and (off-)target are concatenated channel-wise (forming the Siamese part of the network) and serve as input for a convolutional classifier (the conjoined part). Since both encodings scrutinised in this publication combine guide and target sequences, we only utilise 1 arm of this Siamese network (see Fig. 2D). We have made various adjustments to this architecture based on training stability and validation set performance (see the Supplementary Text).

Our RNN architecture is modelled after the bidirectional gated recurrent unit (BGRU) on-target prediction model from [28]. Here, a BGRU layer is used to make use of the relevant longer-range dependencies between sequence features that would go unnoticed by a CNN of manageable kernel size. In order to make this type of architecture usable for off-target prediction, we feed a combination of guide and target sequence as described above into the sequence arm of the network, and the nucleosomal and energy features in a separate arm (see Fig. 2C). Layer dimensions in both arms were adjusted to the shapes of their respective inputs.

Model Training & Evaluation

Given the imbalance of validated/measured and non-validated/augmented data points, we employ a bootstrapping strategy as suggested in [30], where training batches on average contain equal numbers of both classes. For regression (classification), early stopping is based on the mean squared error (binary cross-entropy) loss on half of the test set, where the other half is reserved for evaluation.

The CNN models are trained in the same way, with hyperparameters of `batchnorm_momentum`=0.01, Gaussian noise with $\mu = 0, \sigma = 0.01$ and Adam learning rate 10^{-3}.

The RNN models are trained for 100 epochs, where batches of 10,000 points are sampled each epoch out of a class-balanced sample of 50,000 data points. We replicate the transfer learning approach taken in [28] with adjustments to increase training stability and generalisation performance as detailed in the Supplementary Text. Dropout probability was 0.2 and the Adam learning rate was 10^{-3}.

Testing Scenario 1: held out studies

In this scenario we hold out studies [31–33] from the training set. These studies have not been included in the training set for the state-of-the-art off-target prediction algorithm CRISPR-Net [13], such that they remain an independent test set to compare CRISPR-Net and piCRISPR side by side. The inherent class imbalance in this test set is 1:103.96. 22% of the unique guides within the training set have at least one corresponding guide in the test set with five or fewer mismatches, indicating a satisfactory independence between training and test set.

Testing Scenario 2: literature comparison

In this scenario we use the CIRCLE-seq [34] dataset as the held out test set, as was done in [15]. The exact test set has been replicated using the code provided by the authors, such that comparison values could be taken straight from publication [15]. Nucleosomal and empirical energy data was filled in using the crisprSQL dataset.

Testing Scenario 3: set of duplicate pairs

In this scenario, we scrutinise our hypothesis that an environmentally sensitive feature set is fit to not only increase prediction performance overall, but especially for given groups of identical guide-target sequence pairs. To this end we calculate two quantities: First, the mean squared error (MSE) between the predicted regression scores and the ground truth cleavage frequencies within each of the 2703 groups. Second, the average proportion of the true cleavage activity difference for two points within a given group which the model predicts. This is zero for purely sequence-based models and unity for an ideal predictor. This quantifies how faithful a model is to the differences in biological environment for a given pair. In order to emphasise small deviations which preserve the rank of predicted cleavage activities, we use the cubic root as a sign-preserving nonlinearity and term this quantity *relative difference*. We consider the resulting distributions of both of these quantities for different feature sets.

Model Explanation

We obtain feature importances using the model-agnostic Shapley Additive Explanations (SHAP) library [35]. Since piCRISPR wraps the feature encoding inside a given model, we retain full explainability of input features even for non-invertible encodings. In this way, using the two encodings detailed above, we obtain an unprecedented, context-sensitive resolution of sequence-based features.

Sticking with the convention set by the SHAP library [35], we calculate global SHAP values as the mean of the absolute value of SHAP values across data points in the explanation set, which is a random subset of 500 points from the held out test set. In order to show not only the magnitude but the direction in which a given feature influences the model's prediction, we multiply each feature's global SHAP value by the sign of the average SHAP value of all data points whose value is larger than the median of that feature.

Command line usage of our models

We have implemented a command line interface with which piCRISPR predictions can readily be obtained. For maximum usability, the model automatically uses default feature values in case a certain feature was not provided, thereby lowering prediction performance (see Figure S4). The default value of a given feature is defined as the average feature value of the set of those crisprSQL data points which lie within a 20% interval around the mean SHAP value. This means that high-accuracy piCRISPR predictions can be obtained in a user-friendly way, even when providing only guide and (off-)target sequence. Our online repository contains hands-on examples on this.

Results & Discussion

Testing Scenario 1

Fig. 3 shows the regression and classification performance of our piCRISPR-implemented models, with the 6×23 RNN model yielding the highest benchmarks. As mentioned in [30], the area under precision-recall curve (AU-PRC) is a much more suitable measure than the area under receiver operating curve (AU-ROC) for off-target prediction, since

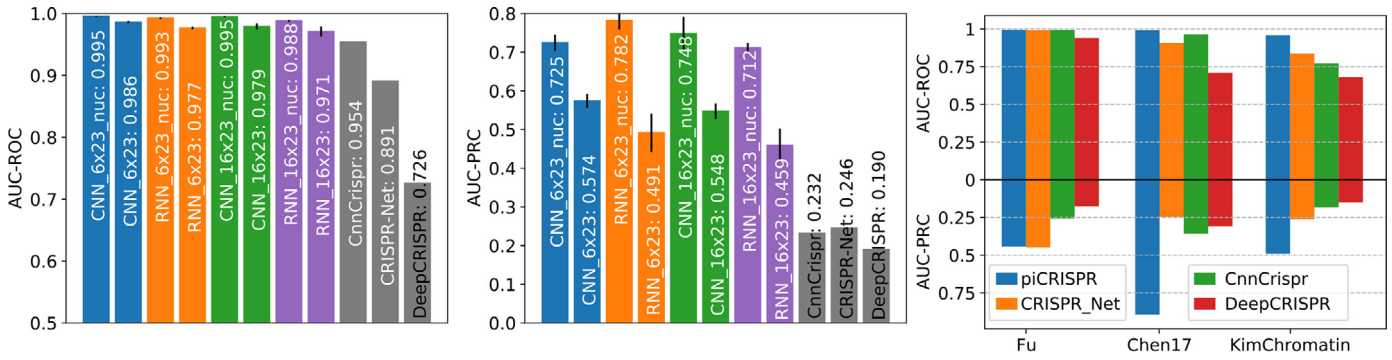


Fig. 3. Comparison of piCRISPR models with published algorithms. All models were tested on held out studies [31–33] (testing scenario 1). Non-validated data points have been oversampled in the test set to match the class imbalance of 1:79.35 found in the dataset I-1 from [13]. piCRISPR models have been trained on the remaining data points within the crisprSQL data set. **Left two panels**: Comparison with three published off-target prediction algorithms [11,13,26] that were run on this test set. Within a model family of the same colour, the model labelled "nuc" contains nucleosomal features whereas the other does not. piCRISPR training and testing have been repeated 5 times to obtain mean and standard deviation as shown. For the underlying ROC and PRC curves see Figure S1. **Right panel**: AUC-ROC and AUC-PRC benchmarks for the RNN 6×23 model with nucleosomal features, resolved by individual study within the held out test set.

in clinical application, false negatives have far more adverse effects than false positives. The addition of the nucleosomal features considerably improves model performance according to all benchmarks, supporting our hypothesis that nucleosomal features can serve as a key ingredient to cleavage prediction.

A direct comparison with prediction results obtained from the published versions of CnnCrispr, CRISPR-Net and DeepCRISPR on the identical held out test set shows that piCRISPR achieves higher classification benchmarks in terms of areas under ROC and PRC curve for all three individual studies contained in the test set, except for study [31] for which piCRISPR and CRISPR-Net achieve comparable benchmarks.

Testing Scenario 2

Fig. 6 shows that when testing on the CIRCLE-seq dataset [34], piCRISPR performance drops slightly as compared to testing scenario 1. Especially RNN models generalise slightly worse to this study. We still observe that nucleosomal features enhance the performance of a model, and piCRISPR still outperforms both CRISPR-IP and CRISPR-Net models according to area under PRC curve.

Testing Scenario 3

Table 1 shows that the model performance, measured by the mean squared error of predictions within a group of data points that share both guide and target sequence, is considerably improved by introducing features beyond sequence information (left column). The resulting distribution of MSEs is shown in Figure S9.

Looking at the relative pairwise difference, we observe that introducing features beyond sequence leads to an increase of the average proportion of true cleavage frequency differences between points of differing biological environment which is captured by the model. This is true for both DeepCRISPR and piCRISPR. Whilst the full feature set in piCRISPR achieves the highest proportion in comparison, it is the piCRISPR sequence-only model that achieves the lowest overall mean squared error. This indicates that a low mean squared error does not necessarily go hand in hand with the model drawing the correct conclusions from environmentally sensitive features. This can be seen as well when considering the comparably small relative pairwise difference that is recovered by the DeepCRISPR model from the literature-standard epigenetics channels to which it has access.

Feature importance

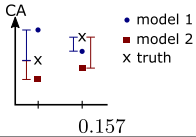
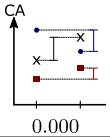
Due to its comparatively stronger prediction benchmarks between testing scenarios 1 and 2, we use the 16×23 CNN classification model in testing scenario 1 to extract feature importance values of unprecedented resolution. Fig. 4 shows that the model draws on sequence features which stem from mismatched interfaces differently than on those from matched interfaces, supporting our hypothesis that this differentiation is not only physically indicated but also backed by the model's behaviour. Global SHAP values suggest that the preference of the variable PAM nucleotide at position 21 is contingent on the specific sgRNA–DNA interface formed. We recover the preference for cytosine at position 17 [11,29,36] as well as position 20 [11,28,29] found in literature for matched interfaces. However, for mismatched interfaces, cytosine is disfavoured. Whilst we cannot recover a strong preference for the variable PAM nucleotide at position 21 for matched interfaces, we observe the preference for guanine reported in literature [11,28,36] for mismatched interfaces. This supports the notion that a concentration on guide-target interfaces rather than pure base identities is necessary for off-target prediction, and that deeper insight is required than the notion of a preferred base at a specific position. It therefore appears necessary to consider mismatch interfaces together with sequences in the desired genome, not just the sequence of the putative guide, for sgRNA design.

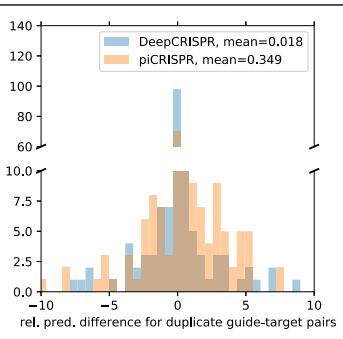
Note that due to the low prevalence of non-NGG PAMs in our dataset, as has been our choice when augmenting it with putative off-targets, the model attributes little importance to the two5′ GG base pairs. We observe the blind spot of mismatch discrimination by the REC3 domain of Cas9 around nucleotide 7 (see also Figure S5) which has been reported in a recent cryo-EM structural study [37] and results in reduced importance of sequence features pertaining mismatched interfaces in this region. At nucleotides 3–5 and 9–11, where mismatch detection by the REC3 domain of Cas9 is high, we observe a mismatch-induced reduction in cleavage activity. We further observe a PAM-distal 'preference zone' and a PAM-proximal 'avoiding zone' of mismatches when averaging over feature importance values by nucleotide, which has been observed in computational [11] as well as cryo-EM [37] studies.

The model draws heavily on the empirical energy estimate feature $E_{\mathrm{CRISPRoff}}$ which yields the largest global SHAP value. We further observe a considerable correlation between its value and the SHAP value attributed to it by the model (Figure S8). An energy score of $E_{\mathrm{CRISPRoff}} =$

Table 1

Benchmark quantities gained on the subset of duplicate guide-target sequence pairs (testing scenario 3) using our 6×23 CNN model as well as the CRISPR-Net [13] and DeepCRISPR [11] models for comparison. For piCRISPR, we also give a sequence-only version of the model in which nucleosome organisation related features and the empirical energy estimate have been set to a default value across all data points. **Left column**: mean squared error (MSE) between predicted cleavage score and ground truth cleavage activity, averaged over all groups of identical guide-target sequence pairs. **Right column**: How faithful a model is to the differences in biological environment for a given pair within such a group is measured by the average proportion of the true cleavage activity difference which the model predicts. This is zero for purely sequence-based models and unity for an ideal predictor. To emphasise small deviations which preserve the rank of predicted cleavage activities, we use the cubic root as a sign-preserving nonlinearity and term this quantity *relative difference*. **Right panel**: Example distributions of relative pairwise difference for the two models. All underlying distributions are shown in Figure S9.

		$(\mathrm{pred}_i - \mathrm{truth}_i)^2$ prediction MSE	$\left(\frac{\mathrm{pred}_i - \mathrm{pred}_j}{\mathrm{truth}_i - \mathrm{truth}_j}\right)^{\frac{1}{3}}$ rel. pairwise difference
CRISPR-Net	sequence-only	 0.157	 0.000
DeepCRISPR	epigenetic	0.019	0.018
piCRISPR	sequence-only	**0.012**	0.000
	full feature set	0.101	**0.349**



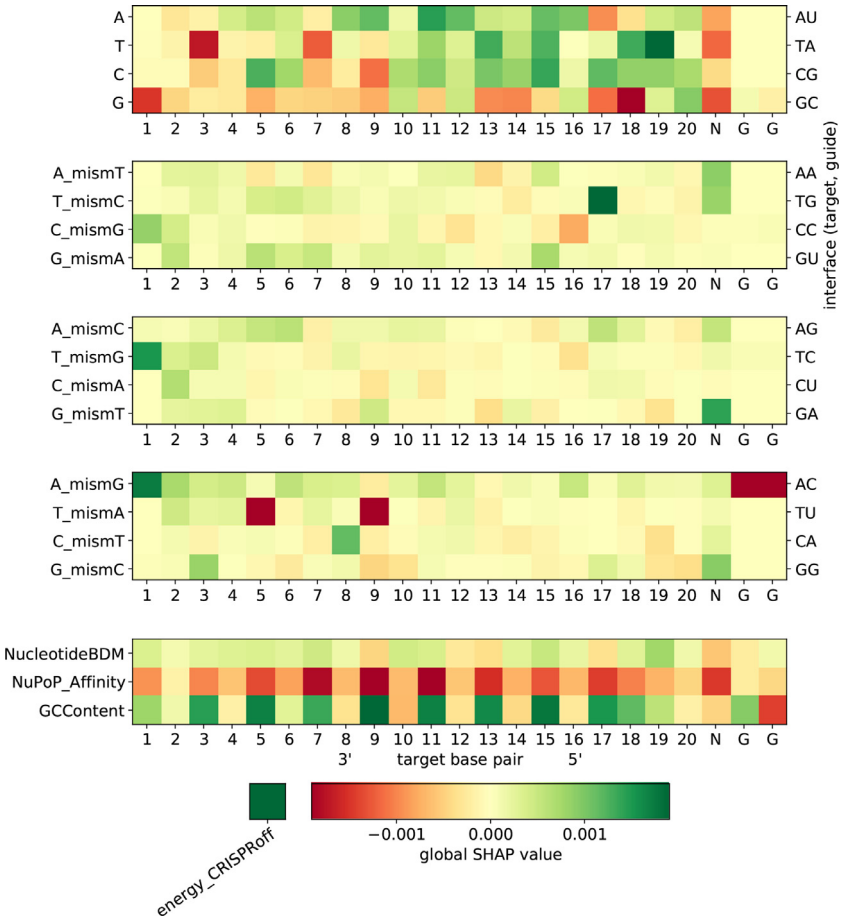


Fig. 4. Global SHAP values for the 16×23 CNN classification model. Negative global SHAP values (red) indicate an average predicted decrease in guide activity for the respective feature. Mismatch channels (middle three heatmaps) can be represented by the (off-)target and on-target protospacer nucleotides (left vertical axis) as well as the physical base pair interfaces (right vertical axis), such that `A_mismT` describes all configurations in which an adenine on the target strand faces an adenine on the sgRNA. The bottom heatmap visualises the influence of our chosen set of nucleosomal organisation features on cleavage activity. A bar representation of this can be found in Figure S5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

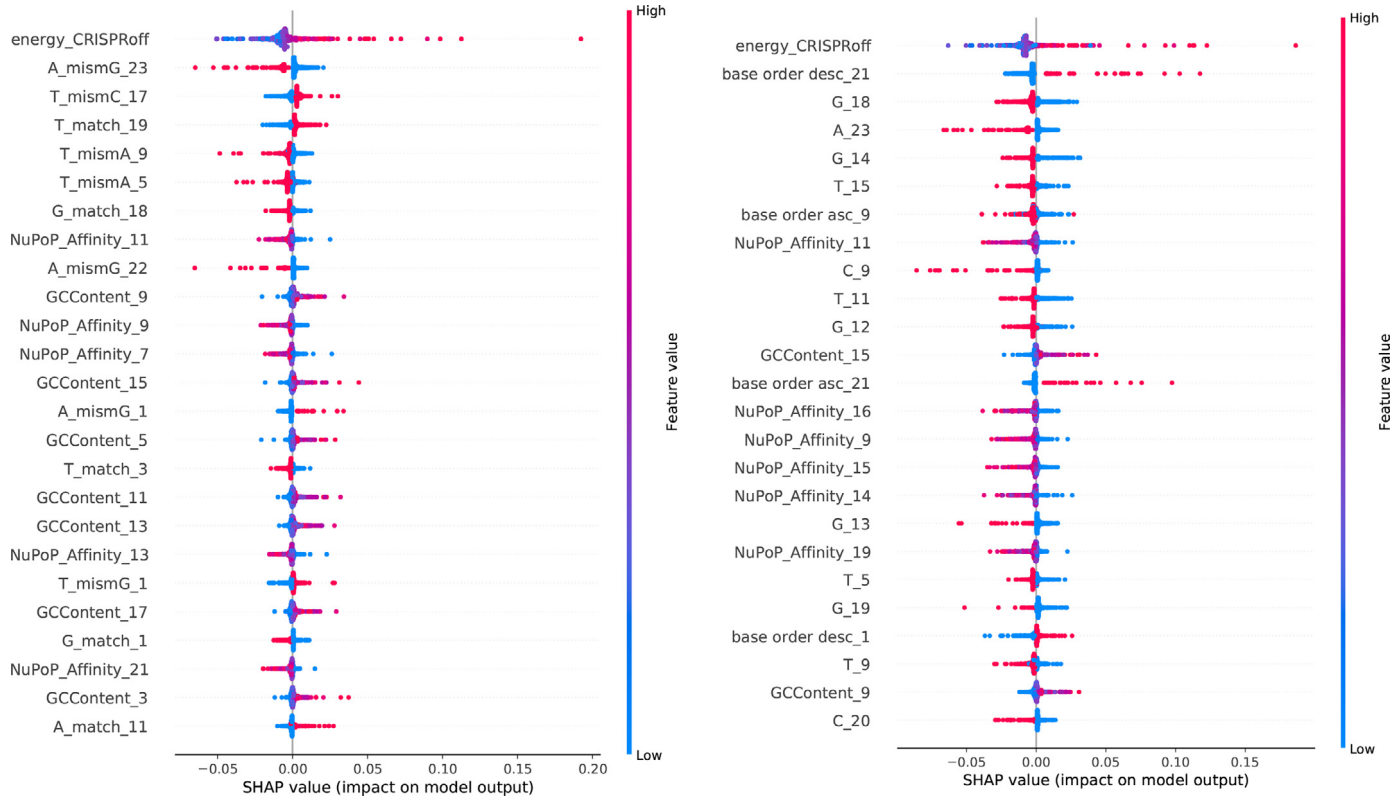


Fig. 5. Base-pair resolved SHAP values for the 16×23 (left panel, see Fig. 4) and 6×23 (right panel, see Figure S7) CNN classification models. SHAP values have been obtained on the held out studies [31–33] from the crisprSQL dataset. Note that high values of the NuPoP Affinity feature (red dots), i.e. highly positioned nucleosomes, always influence the model towards reduced cleavage activity (negative SHAP value). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

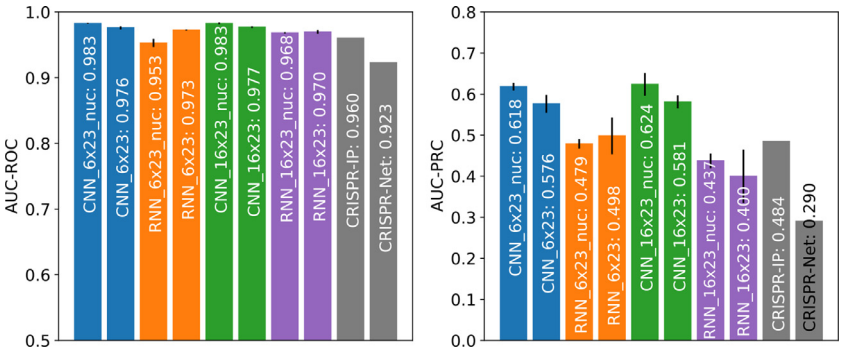


Fig. 6. Comparison between piCRISPR, CRISPR-Net [13] and CRISPR-IP [15], with comparison values for the latter two models taken from [15]. All models were tested on a the subset of the CIRCLE-seq dataset [34] as given in [15] (testing scenario 2). Note the slightly reduced performance of the RNN models compared to testing scenario 1 (Fig. 3).

−1.15 has a neutral influence on cleavage activity in the 16×23 CNN model, with higher (lower) values yielding larger (smaller) SHAP values, i.e. a positive (negative) influence on cleavage activity.

When considering nucleosome positioning-related feature channels, we see that the 147 bp GC content around each nucleotide has a net positive influence on cleavage activity. Similar to the argument in [10], this can be attributed to the increased bendability of GC-rich DNA [38] which is beneficial to Cas9 sequence readout during binding [39]. We further observe that the NuPoP Affinity score ranks higher in terms of global SHAP value than most sequence features. The negative influence of nucleosome affinity can be explained by the reduced accessibility of high-affinity DNA regions, and is observed strongly between nucleotides 5 and 19. This effect has been observed in [10] as well. We further observe an overall negative influence of low Nucleotide BDM values on cleavage activity, supporting what has been observed in preliminary, non-sequence based models in [10].

This also demonstrates the importance of nucleosome-related features for cleavage prediction, and also supports the notion of chromatin accessibility influencing cleavage activity found in [40]. To our knowledge, this strong effect of a more than 10 bp wide sequence context on genome-wide off-target cleavage prediction has not been demonstrated yet. Hints of it have been seen only for smaller contexts and on-target efficacy prediction [41,42]. In addition, our findings present an unprecedented example in which information in the 147 bp sequence context has considerable impact on the model.

A similar analysis for the 6×23 CNN model can be found in Figure S6 and for the 16×23 RNN model in Figure S7. Figure 5 shows the underlying SHAP values for both CNN classification models. Note that within the nucleosomal feature class, the RNN models attribute more importance to the Nucleotide BDM feature than the CNN models scrutinised here. This could in part explain their slight difference in performance between testing scenarios 1 and 2.

Conclusion

Through the introduction of a new feature class and the careful adjustment of model architectures, we have identified a set of models which match the performance of state-of-the-art off-target cleavage prediction algorithms in direct comparison. All models are highly influenced by nucleosome organisation-related features such as histone binding affinity, which emphasises the importance of capturing the biological environment around the cleavage site when modelling cleavage activity. Our approach has shown that these computed physically informed features are fit to enhance the predictive power of cleavage prediction models and to replace experimental epigenetic markers in future modelling efforts. We have further provided an accessible, user-friendly command line interface that allows users of various disciplines to utilise all our models, even without providing a complete set of features. This all paves the way towards the prediction of off-target sites which would so far have gone unnoticed.

Our environmentally sensitive set of features reveals several novel, promising pathways towards further improvement of off-target cleavage prediction. Going forward, it could be fruitful to increase model complexity, e.g. using a 2D convolutional kernel to capture interaction between features of a single nucleotide. A 2D convolution kernel would be able to capture the base-pair resolved interaction between sequence and nucleosomal markers as well as between sequence k-mers. Further than this, our multimodal data could be fused at different stages, such that sequence, nucleosomal and energy features interact at different levels of representation of each other.

We further envision to replace the epigenetic information of the guide, which so far only copies the epigenetic information of the target DNA. This is clearly an unphysical choice. Given that a synthetic sgRNA does by design not carry epigenetic markers, a one-hot encoded dot-bracket representation of the sgRNA folding would be a more suitable choice to capture its informative properties.

Funding

This research was funded in whole or in part by the Biotechnology and Biological Sciences Research Council (BBSRC) [BB/M011224/1, BB/S507593/1]. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission. Some of the presented results have been obtained using the University of Oxford Advanced Research Computing (ARC) facility (https://doi.org/10.5281/zenodo.22558). The authors declare no conflict of interest.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset, source code and trained models can be found ready to use at github.com/florianst/picrispr.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ailsci.2023.100075

References

[1] Ishino, et al. J Bacteriol 1987;169:5429.
[2] Horvath, et al. Science 2010;327:167.
[3] Kunin, et al. Genome Biol 2007;8:R61.
[4] Urnov, et al. Nat Rev Genet 2010;11:636.
[5] Joung, et al. Nat Rev Mol Cell Biol 2013;14:49.
[6] Ran, et al. Nat Protoc 2013;8:2281.

[7] Wang, et al. Annu Rev Biochem 2016;85:227.
[8] Xi, et al. BMC Bioinformatics 2010;11:346.
[9] Ozaki, et al. Cancers (Basel) 2011;3:994.
[10] Mak, et al. BMC Genomics 2022;23:805.
[11] Chuai, et al. Genome Biol 2018;19:80.
[12] Liu, et al. PLoS Comput Biol 2019;15:e1007480.
[13] Lin, et al. Adv Sci 2020;7:1903562.
[14] Charlier, et al. Bioinformatics 2021;37:2299.
[15] Zhang, et al. Comput Struct Biotechnol J 2022;20:650.
[16] Störtz, et al. Nucleic Acids Res 2020;49:855.
[17] Lazzarotto, et al. Nat Biotechnol 2020;38:1317.
[18] Alkan, et al. Genome Biol 2018;19:177.
[19] Zenil, et al. Nucleic Acids Res 2019;47:e129.
[20] Franco, et al. Biol Reprod 2014;91.
[21] Sims, et al. Trends Genet 2003;19:629.
[22] Anders, et al. Nature 2014;513:569.
[23] Kim, et al. Nat Methods 2015;12:237.
[24] Box, et al. J R Stat Soc B 1964;26:211.
[25] Listgarten, et al. Nat Biomed Eng 2018;2:38.
[26] Liu, et al. BMC Bioinformatics 2020;21:1.
[27] Gruber, et al. Nucleic Acids Res 2008;36:W70.
[28] Zhang, et al. Comput Struct Biotechnol J 2020;18:344.
[29] Konstantakos, et al. Nucleic Acids Res 2022;50:3616.
[30] Gao, et al. Brief Bioinform 2020;21:1448.
[31] Fu, et al. Nat Biotechnol 2013;31:822.
[32] Kim, et al. Genome Res 2018;28:1894.
[33] Chen, et al. Nature 2017;550:407.
[34] Tsai, et al. Nat Methods 2017;14:607.
[35] Lundberg, et al. Adv Neural Inf Process Syst 2017;30:4765.
[36] Doench, et al. Nat Biotechnol 2014;32:1262.
[37] Bravo, et al. Nature 2022;603:343.
[38] Vinogradov, et al. Nucleic Acids Res 2003;31:1838.
[39] Cofsky, et al. Nat Struct Mol Biol 2022;29:395.
[40] Dhanjal, et al. Genomics 2020;112:3609.
[41] Xu, et al. Genome Res 2015;25:1147.
[42] Boyle, et al. Sci Adv 2021;7:5496.