UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UTILISATION DE MÉTHODES D'APPRENTISSAGE PROFOND POUR LA PRÉDICTION DES EFFETS CIBLE ET HORS CIBLE DANS LA TECHNOLOGIE CRISPR-CAS9

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE

PAR

ZEINAB SHERKATGHANAD

MARS 2026

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Professor Vladimir Makarenkov from Université du Québec à Montréal (UQAM), for his continuous encouragement and motivation throughout my PhD journey. I deeply appreciate the efforts, valuable ideas, and financial support he provided, which made my research productive and intellectually stimulating. His expert supervision and guidance were instrumental in shaping both the research direction and the methodology of this thesis.

I would also like to thank my colleagues and friends at the Department of Computer Science at UQAM for their immense support during my studies. In particular, I am grateful to Dr. Moloud Abdar for his assistance with research materials, helpful discussions, and insightful comments. His support was invaluable in refining the methodology and developing key ideas throughout this work.

This research was supported by funding from Natural Sciences and Engineering Research Council of Canada (NSERC), which I gratefully acknowledge.

Special thanks are addressed to my family—my husband, my daughter, and my parents—for their unwavering spiritual support, understanding, and encouragement throughout this multi-year adventure. Their love and belief in me have been a constant source of strength.

Zeinab Sherkatghanad

Montreal, September, 2025

DÉDICACE

Dedicated to my beloved family,
whose love, support, and encouragement
have guided me throughout this journey.

PREFACE

This thesis presents the application of deep learning (DL) and machine learning (ML) techniques to improve the reliability, efficiency, and safety of genome editing technologies, with a particular focus on the Clustered Regularly Interspaced Short Palindromic Repeats and CRISPR-associated protein 9 (CRISPR/Cas9) systems. We proposed new deep learning methods and computational frameworks for improving the predictive accuracy and reliability of CRISPR/Cas9 genome editing applications. When Prof. Makarenkov agreed to become my supervisor, he proposed that I focus my research on the development of machine learning-based approaches to address critical challenges in genome editing, particularly the prediction and reduction of off-target effects. Prof. Makarenkov explained that while CRISPR/Cas9 is a powerful tool for genetic engineering, it is prone to unintended edits at non-target sites, which presents significant obstacles for clinical and research applications. This challenge, combined with the increasing availability of CRISPR benchmark datasets, opened an important research avenue to develop computational methods capable of enhancing the safety and accuracy of CRISPR predictions. As I began my literature review, I realized the urgent need for methods that could not only predict guide RNA efficiency but also provide reliable estimates of off-target activity. Although numerous data-driven models have been proposed for off-target prediction, their applicability has often been constrained by limited data availability, the absence of robust Uncertainty Quantification (UQ), and poor generalizability across diverse datasets. Together with Prof. Makarenkov, we formulated several key research questions focused on addressing these limitations using novel techniques, including uncertainty estimation method, and similarity-based Transfer Learning (TL).

In the first paper, published in *Briefings in Bioinformatics*, we reviewed the current landscape and predictive challenges inherent in CRISPR/Cas9 off-target activity. This work provided a comprehensive overview of existing computational methods, highlighting the limitations of current off-target prediction models. Building on this foundation, we present an overview and comparative analysis of traditional machine learning and deep learning approaches applied to CRISPR/Cas9. We discuss existing datasets and recent advances in encoding sgRNA–DNA sequences. Our study further explores the most popular deep learning neural network architectures employed in CRISPR/Cas9 prediction tasks. Finally, we summarize the remaining challenges and propose potential avenues for future research aimed at improving both on-target and off-target prediction accuracy. This work led to the writing of my second article, published in *Knowledge-Based Systems (2025)*, introduced BayTTA, a novel uncertainty estimation framework that leverages Bayesian Model Averaging (BMA) for optimized Test-Time augmentation (TTA). BayTTA was designed to improve the robustness of neural

network predictions by integrating uncertainty quantification into the inference process. Building upon the results of the second paper, we developed a third article, which was submitted to *PLOS Computational Biology*. In this work, we proposed a novel similarity-based transfer learning approach that integrates similarity based methods to enhance the effectiveness of transfer learning for CRISPR-Cas9 off-target prediction.

This PhD thesis encompasses multidisciplinary material that will be of interest to statisticians, bioinformaticians, and life scientists. The Introduction chapter does not provide a comprehensive literature review, as this is covered extensively in the first article (Chapter I). The thesis is structured to include an introductory chapter, three manuscript chapters corresponding to the research articles, and a concluding chapter.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ALGORITHMES

LIST OF ABBREVIATIONS AND ACRONYMS

CRISPR	Clustered Regularly Interspaced Short Palindromic Repeats
sgRNA	single-guide RNA
HTS	High-throughput screening
HCS	high-content screening
CIRCLE-Seq	Circularization for In vitro Reporting of CLeavage Effects by sequencing
GUIDE-Seq	Genome-wide unbiased identification of DSBs enabled by sequencing
HTGTS	High-throughput genome-wide translocation sequencing
DeepHF	Deep learning for High-Fidelity Cas9
ML	Machine Learning
DL	Deep Learning
AUROC	Area under the receiver operating characteristic curve
AUPRC	Area under the precision recall curve
ROC	Receiver Operating Characteristic
FNN	Feedforward Neural Networks
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
GRU	Gated Recurrent Unit
MLP	Multilayer Perceptron
LR	Logistic Regression
RF	Random Forest
GBRT	Gradient-boosted regression tree
SVM	Support Vector machine
MCC	Matthews Correlation Coefficient
GB	Gradient boosting
NN	Neural networks
KNN	k-nearest neighbors
GTB	Gradient Tree Boosting

DT	Decision Tree
BRR	Bayesian Ridge regression
nDCG	Normalized discounted cumulative gain
LSTM	Long Short-Term Memory
BLSTM	Bidirectional Long Short-Term Memory
DCDNN	Deep Convolutional Denoising Neural Network
LRCN	Long-term Recurrent Convolutional Network
SVR	Support Vector Regression
TTA	Test Time Augmentation
BMA	Bayesian Model Averaging
BayTTA	Bayesian-based TTA
UQ	Uncertainty quantification
ROI	Region of interest
TL	Transfer Learning

RÉSUMÉ

La technologie CRISPR-Cas9 (Clustered Regularly Interspaced Short Palindromic Repeats) a révolutionné l'édition génomique en permettant des modifications précises et efficaces des séquences d'ADN. Dans ce contexte, une prédiction précise des effets cible et des activités hors cible est essentielle pour améliorer la sécurité et l'efficacité de l'édition génomique.

Malheureusement, l'application étendue de CRISPR-Cas9 est remise en question par des effets hors cible non intentionnels, susceptibles de compromettre l'intégrité du génome et de limiter son utilisation clinique. Des modèles récents basés sur l'analyse des données et s'appuyant sur l'apprentissage profond montrent des résultats prometteurs avec le nombre croissant de données génomiques. En termes de prédiction des effets hors cible, ils surpassent généralement les méthodes existantes basées sur une fonction de score. De plus, des travaux récents ont démontré l'efficacité de l'encodage « one-hot » des séquences d'ARN guide et d'ADN ciblée, associée à l'enzyme Cas9, qui sont employées par CRISPR-Cas9. Ces données encodées peuvent ensuite être efficacement utilisées comme les données d'entrée des réseaux neuronaux convolutifs ou récurrents pour prédire des événements de clivage hors cible. L'amélioration des prédictions relatives aux effets cible et hors cible dans l'édition du génome reste l'une des principales préoccupations des chercheurs et des cliniciens, car elles impactent directement la fiabilité et la sécurité des applications basées sur la technologie CRISPR. Il est important de noter que les modèles d'apprentissage profond utilisent des milliers de paramètres, nécessitant un nombre important d'échantillons dans les jeux de données CRISPR-Cas9. Bien que ces modèles marquent une avancée importante, leur application pratique reste quand même limitée en raison de la pénurie de données, de l'indisponibilité de méthodes adaptées de quantification de l'incertitude et de leur faible généralisabilité à travers des ensembles de données hétérogènes.

Cette thèse, organisée par article, vise à surmonter les défis mentionnés grâce à nos trois contributions principales. Le premier article, publié dans Briefings in Bioinformatics, propose une revue de littérature complète des modèles d'apprentissage automatique traditionnels et d'apprentissage profond existants pour une prédiction d'efficacité cible et d'activité hors cible en CRISPR-Cas9, soulignant leurs avantages et limites principaux. Nous examinons également les stratégies existantes de l'encodage des séquences et discutons de la disponibilité des données, mentionnant les défis à relever et les orientations de recherches futures. Le deuxième article, publié dans Knowledge-Based Systems, présente BayTTA - un cadre d'estimation de l'incertitude, qui exploite un modèle bayésien basé sur les moyennes (BMA - Bayesian Model Averaging) et une technique d'optimisation reliée (TTA - Test-Time Augmentation). Nous générons une liste de prédictions associée à différentes variations des données d'entrée créées par TTA. Ensuite, nous utilisons BMA pour combiner les prédictions pondérées par les probabilités a posteriori respectives. BayTTA permet de prendre en compte l'incertitude du modèle, améliorant ainsi la robustesse prédictive et fournissant des estimations d'incertitude interprétables. Nous évaluons les performances de BayTTA sur divers jeux de données publiques, dont trois jeux de données d'images médicales et deux jeux de données CRISPR/Cas9 bien connus, CRISPOR et GUIDE-seq. Le troisième article, soumis à PLOS Computational Biology (révisions mineures restantes), propose une nouvelle approche d'apprentissage par transfert basée sur la similarité pour prédire des effets hors cible dans la technologie CRISPR-Cas9. L'apprentissage par transfert s'est révélé un outil puissant pour améliorer la précision prédictive dans les tâches complexes, en particulier dans les scénarios où les données sont limitées ou déséquilibrées. Notre étude explore l'utilisation de la pré-évaluation basée sur la similarité entre les jeux de données comme méthodologie pour identifier les données sources optimales pour l'apprentissage par transfert. Nous répondons ainsi au double défi de l'appariement efficace des jeux de données source-cible lors de l'apprentissage par transfert et de la prédiction des effets hors cible dans

CRISPR-Cas9. En intégrant des méthodes basées sur la similarité, notre approche améliore la généralisation et la précision prédictive pour des petits jeux de données d'édition génomique.

Ensemble, ces contributions font progresser l'application de l'apprentissage automatique traditionnelle et de l'apprentissage profond dans l'édition du génome en améliorant la précision des prédictions des effets hors cible et en quantifiant efficacement l'incertitude de ces prédictions.

Mots-clés : CRISPR-Cas9, Édition du génome, Apprentissage automatique, Apprentissage profond, Sur cibles, Hors cibles, Estimation de l'incertitude, Apprentissage par transfert.

SUMMARY

Clustered Regularly Interspaced Short Palindromic Repeats and its-associated protein 9 (CRISPR/Cas9) technology has revolutionized genome editing by enabling precise and efficient modifications of DNA sequences. In this context, accurate prediction of both on-target efficiency and off-target activity is essential to improve the safety and effectiveness of genome editing.

The widespread application of CRISPR/Cas9 is challenged by unintended off-target effects, which can compromise genome integrity and limit clinical translation. The modern data-driven models that rely on deep learning show promising results with the growing number of CRISPR-Cas9 data ; they typically outperform existing scoring methods in terms of off-target prediction. Recent works have often demonstrated the effectiveness of one-hot sequence encoding, often combined with convolutional or recurrent neural networks, to predict off-target cleavage events. Improving the trustworthiness of predictions related to on- and off-target effects in genome editing remains one of the main concerns for both researchers and clinicians, as it directly impacts the reliability and safety of CRISPR-based applications. Deep learning models employ thousands of parameters, requiring a substantial number of samples in CRISPR-Cas9 datasets. While these models mark an important step forward, their application if genome editing remains limited because of data scarcity, unavailability of adapted uncertainty quantification methods, and poor generalizability across heterogeneous datasets.

This thesis, organized by publication, seeks to overcome those challenges through our three main contributions. The first article, published in *Briefings in Bioinformatics*, provides a comprehensive review of existing traditional machine learning and deep learning models for CRISPR/Cas9 on-target efficiency and off-target activity predictions, highlighting their limitations in handling data scarcity, model uncertainty, and generalizability across datasets. We further examine sequence encoding strategies, data availability, and neural network architectures applied in CRISPR/Cas9, outlining open challenges and future research directions. The second article, published in *Knowledge-Based Systems*, introduces BayTTA, an uncertainty estimation framework that leverages Bayesian Model Averaging (BMA) for optimized test-time augmentation (TTA). We generate a prediction list associated with different variations of the input data created through TTA. Then, we use BMA to combine predictions weighted by the respective posterior probabilities. BayTTA allows one to take into consideration model uncertainty, and thus improves predictive robustness and provides interpretable uncertainty estimates. We evaluate the performance of BayTTA on various public data, including three medical image datasets and two well-known CRISPR/Cas9 datasets, CRISPOR and GUIDE-seq. The third article, submitted to *PLOS Computational Biology* (minor revisions remaining), proposes a novel similarity-based transfer learning approach for CRISPR/Cas9 off-target prediction. Transfer learning has emerged as a powerful tool for enhancing predictive accuracy in complex tasks, particularly in scenarios where data is limited or imbalanced. This study explores the use of similarity-based pre-evaluation as a methodology to identify optimal source datasets for transfer learning, addressing the dual challenge of efficient source-target dataset pairing and off-target prediction in CRISPR-Cas9. By integrating similarity-based methods, this approach enhances the effectiveness of transfer learning, improving generalization and predictive accuracy on small genome-editing datasets.

Together, these contributions advance the application of machine learning and deep learning in genome editing by improving prediction accuracy and quantifying uncertainty, both of which are critical for minimizing off-target effects and accelerating the safe development of CRISPR-based therapeutics.

KEYWORDS : CRISPR-Cas9, Genome Editing, Machine Learning, Deep Learning, On-Targets, Off-Targets, Uncertainty Estimation, Transfer Learning.

INTRODUCTION

The Clustered Regularly Interspaced Short Palindromic Repeats and CRISPR-associated protein 9 (CRISPR/Cas9) gene-editing platform originates from the bacterial CRISPR-Cas9 adaptive immune defense system. In this system, CRISPR sequences represent clusters of DNA within prokaryotic genomes that are derived from fragments of invading bacteriophage DNA, thereby providing a molecular record of past infections. Rapid advancement of genome editing technologies, particularly CRISPR/Cas9 system, has revolutionized biological research and opened new possibilities for therapeutic applications.

Due to its remarkable simplicity, high efficiency, and broad versatility, the CRISPR/Cas9 system has rapidly become the method of choice for performing targeted genome modifications across a wide variety of organisms, ranging from microorganisms to complex eukaryotes. Its ability to introduce precise double-strand breaks at virtually any desired genomic locus has revolutionized biological research and opened new possibilities for therapeutic interventions. Despite these advantages, the precision of CRISPR/Cas9 remains a major concern. In particular, unintended off-target cleavage events can lead to undesirable genetic alterations, which may compromise the validity of experimental findings and pose serious risks in clinical applications. Therefore, mitigating off-target activity while maintaining high on-target efficiency is of central importance. Addressing this challenge requires the development of reliable and accurate computational models that are capable of simultaneously predicting guide RNA activity and assessing its potential off-target interactions. Such predictive frameworks are crucial not only for improving experimental reproducibility but also for ensuring the safe and effective application of CRISPR/Cas9 in therapeutic contexts.

Parallel to the evolution of genome editing tools, the increasing availability of high-throughput screening (HTS) and high-content screening (HCS) platforms has enabled the generation of large-scale biological datasets. These technologies allow researchers to systematically evaluate the effects of thousands of guide RNAs or compounds, producing rich data that can be used to train predictive models. The integration of such data with advanced machine learning (ML) and deep learning (DL) techniques offer a unique opportunity to improve the reliability and generalizability of CRISPR/Cas9 activity prediction.

Over the past decade, data-driven methods have increasingly outperformed traditional scoring-based approaches such as the MIT CRISPR Design Tool2 [Ran et al., 2013], CCTop [Stemmer et al., 2015], CRISPR Design [Hsu et al., 2013], E-CRISP [Heigwer et al., 2014], and CHOPCHOP [Montague et al., 2014] in both on-

and off-target prediction. While conventional methods often reach a performance plateau regardless of dataset size, deep learning models can continuously improve as more training data become available. This scalability has positioned deep learning as a cornerstone for next-generation genome editing prediction tools. Recent research in clinical CRISPR/Cas9 applications has shifted towards refining these data-driven models to achieve three key objectives :

- Increasing on-target editing efficiency,
- Reducing off-target cleavage events,
- Optimizing both objectives simultaneously for therapeutic reliability.

Furthermore, incorporating Uncertainty quantification (UQ) into predictive models can provide valuable confidence measures, enabling more informed experimental design and decision-making. Transfer learning (TL) strategies, leveraging related datasets and prior models, can further enhance predictive accuracy in data-scarce scenarios, thus expanding the applicability of computational CRISPR prediction tools across diverse contexts.

This thesis addresses these challenges by developing and evaluating an uncertainty estimation-based framework and transfer learning approaches for CRISPR/Cas9 activity prediction.

0.1 Evolution of Genome Editing Tools

Advances in the area of genome editing (also called gene editing) in the 2010s revolutionized molecular biology, genetics and bio-medicine. Genome editing refers to a set of techniques that enable targeted modifications of the DNA in living organisms. Genome editing techniques allow precise manipulation, deletion and insertion of sequence fragments within the DNA of living organisms. In recent years, three effective types of genome editing toolsets, called Zinc Finger Nucleases (ZFNs)[Esvelt and Wang, 2013], Transcription Activator-Like Effector Nucleases (TALENs), and Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR), have been developed to study the process of target modifications in gene sequences [Esvelt and Wang, 2013, Puchta and Fauser, 2013, Barrangou and Doudna, 2016, Manghwar et al., 2019, Bogdanove et al., 2018].

ZFNs are synthetic proteins combining zinc finger DNA-binding domains with the FokI nuclease, allowing targeted double-strand breaks [Carroll, 2011]. Each zinc finger recognizes a specific 3-base-pair sequence, enabling modular assembly for site-specific targeting. Despite their versatility, ZFNs are challenging to design

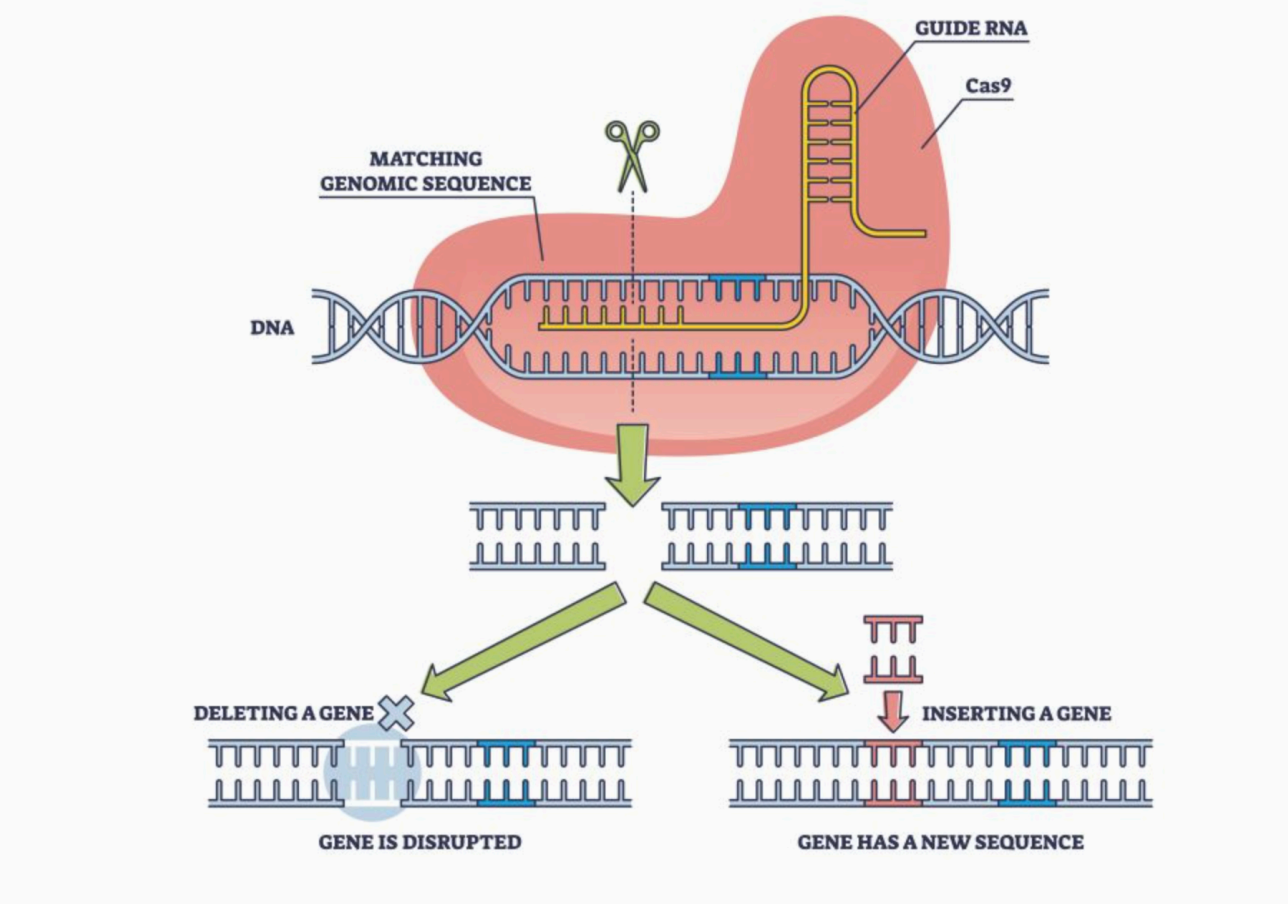

Figure 0.1 – Schematic view of CRISPR/Cas9 gene editing system ; source : https ://labassociates.com/crispr-a-gene-editing-tool.

and often limited by off-target effects and cytotoxicity. TALENs are engineered proteins composed of TALE repeat arrays that bind to specific DNA sequences, fused to a FokI nuclease domain. Each TALE repeat corresponds to a single nucleotide, offering high specificity and straightforward customization. TALENs are easier to design than ZFNs but are large in size, which can complicate their cellular delivery.

Highly effective CRISPR/Cas9 (Clustered Regularly Interspaced Short Palindromic Repeats and CRISPR-associated protein 9) gene editing system, co-invented in 2012 by Emmanuelle Charpentier and Jennifer Doudna [Jinek et al., 2012], has been used in various fields, ranging from basic research on genetic therapies at the cellular level to applied biomedical research [Jinek et al., 2012, Cho et al., 2013, Cong et al., 2013, Mali et al., 2013, Chang et al., 2013, Hsu et al., 2014]. CRISPR/Cas9 demonstrated important clinical potential for treating human diseases such as cancer and genetic disorders [Kang et al., 2017, Liang et al., 2015, Ma et al., 2017], for plant genetic engineering [Liu et al., 2017, Tang et al., 2016, Raitskin and Patron, 2016], and for animal disease treatment [Zarei et al., 2019, Wang et al., 2013].

0.2 Data description and sgRNA-DNA sequence encoding

Benchmark data sets in CRISPR/Cas9 research can be broadly divided into three categories : (i) off-targets only, (ii) on-targets only, and (iii) combined off- and on-targets. The increasing availability of large, high-quality data sets enables more comprehensive modeling of CRISPR/Cas9 activity, allowing machine learning and deep learning methods to leverage both sequence-intrinsic features and broader cellular context. Despite these advances, systematic comparisons between different experimental data sets reveal notable inconsistencies. This highlights the need for further investigation to determine the most robust and generalizable predictors of guide performance, including both on-target efficiency and off-target specificity.

Before building data-driven models for on- and off-target prediction in CRISPR/Cas9, sgRNA-DNA sequence data must be pre-processed into a format readable by computational models. Data pre-processing, also called data encoding, converts nucleotide sequences into numerical representations that can be used as model inputs. Two widely used encoding strategies in genome editing are one-hot encoding and word embedding. In one-hot encoding, each nucleotide (A, C, G, T) is represented as a unique binary vector, while word embedding techniques, inspired by natural language processing, map k-mer substrings of sgRNA-DNA sequences to dense vector representations that capture contextual relationships [Mikolov et al., 2013].

Recent works, such as DeepCRISPR [Chuai et al., 2018], Lin et al. [Lin and Wong, 2018, Lin et al., 2020],

and Charlier et al. [Charlier et al., 2021], have demonstrated the effectiveness of one-hot encoding, often combined with convolutional or recurrent neural networks, to predict off-target cleavage events. Advanced one-hot schemes also incorporate mismatches, insertions, deletions, or DNA/RNA bulges to enhance the input representation and enable state-of-the-art predictive performance. Word embedding approaches, implemented with methods such as GloVe or Transformer architectures [Liu et al., 2019, 2020b, Zhang et al., 2021], offer an alternative by encoding sgRNA sequences in a vector space where similar sequences are placed closer together, capturing both sequence patterns and functional similarity.

Beyond sequence encoding, additional features can further improve predictive accuracy for both machine learning and deep learning models. Traditional machine learning approaches rely heavily on feature engineering, including position-specific nucleotide composition, GC content, mismatch counts, n-gapped dinucleotides, thermodynamic properties, RNA secondary structure, and amino acid-related characteristics [Doench et al., 2014, 2016, Rahman and Rahman, 2017, Schoonenberg et al., 2018, Dhanjal et al., 2020, Hiranniramol et al., 2020, He et al., 2021]. Deep learning models, in contrast, can automatically learn and extract informative features from encoded sequences, potentially reducing the need for manual feature design. Nevertheless, comparisons across different experimental data sets reveal substantial discordance in feature importance and predictive relevance, highlighting the ongoing need to identify generic explanatory features that robustly govern both guide efficiency (on-target activity) and specificity (off-target effects). This underscores the value of integrating high-quality data, careful pre-processing, and systematic feature analysis in building reliable CRISPR-Cas9 predictive models.

0.3 Mechanism and Applications of CRISPR/Cas9

In the CRISPR/Cas9 gene-editing system, the Cas9 nuclease is guided by a synthetic single-guide RNA (sgRNA) to a specific genomic locus, where it introduces a double-strand break. This enables the removal of targeted genes or the insertion of new sequences in vivo [Bak et al., 2018]. The sgRNA, engineered for type II CRISPR/Cas9 systems, determines target specificity by recognizing a complementary DNA sequence and directing Cas9 to cleave at the intended site. For successful cleavage, a short DNA motif known as the Protospacer Adjacent Motif (PAM), typically three to five nucleotides in length, must be present immediately downstream of the target site [Shah et al., 2013].

Despite its precision, the CRISPR/Cas9 system is susceptible to unintended cleavage at genomic sites that share partial sequence similarity with the target, referred to as off-target effects [Zhang et al., 2015, Chen

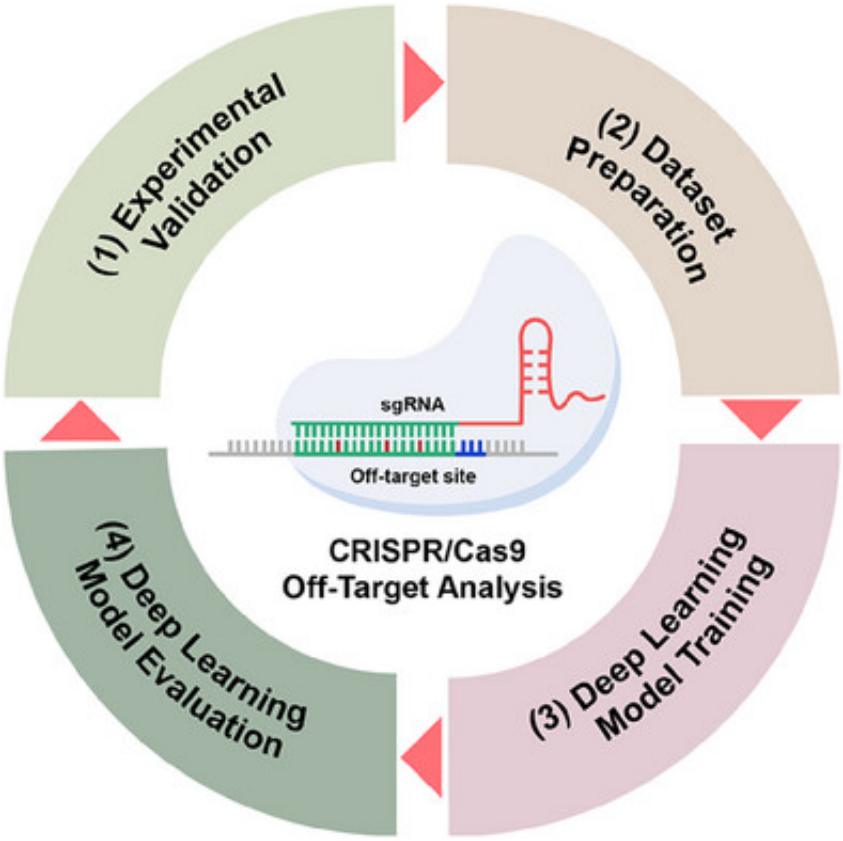

Figure 0.2 – Overview of deep learning models for CRISPR/Cas9 off-target prediction ; Figure taken from Cao et al. (2025), source : https ://onlinelibrary.wiley.com/doi/abs/10.1002/smtd.202500122.

et al., 2017, Kim et al., 2015]. These off-target modifications raise significant safety concerns for clinical and therapeutic applications in humans. Therefore, a primary challenge in refining CRISPR/Cas9 technology is to enhance its on-target activity (editing efficiency) while reducing off-target activity (editing specificity).

In recent years, data-driven approaches have gained significant attention as a powerful alternative to conventional scoring-based prediction tools for CRISPR/Cas9 guide design. Traditional methods, such as the MIT CRISPR Design Tool [Ran et al., 2013], the CCTop algorithm [Stemmer et al., 2015], CRISPR Design [Hsu et al., 2013], E-CRISP [Heigwer et al., 2014], and CHOPCHOP [Montague et al., 2014], have been widely used but remain limited in scalability. Specifically, their predictive accuracy does not necessarily improve as the size of available training data increases. In contrast, data-driven approaches, particularly those leveraging deep learning, inherently benefit from larger datasets, thereby continuously enhancing their predictive power. Current state-of-the-art research in the development of clinical CRISPR/Cas9 applications focuses on improving these models with maximizing on-target activity, reducing off-target effects, and achieving a balance between efficiency and specificity. An overview of representative deep learning models applied to CRISPR/Cas9 off-target prediction is illustrated in Figure 0.2.

0.4 Uncertainty Quantification

Deep learning has fundamentally transformed diverse fields enabling models to achieve unprecedented levels of accuracy in complex tasks. However, despite these advances, deep learning models remain largely opaque, often functioning as “black boxes” that provide predictions without clear indications of their confidence or reliability. This lack of transparency presents significant challenges, particularly in high-stakes domains such as medical diagnosis, autonomous driving, and financial forecasting, where errors can lead to catastrophic outcomes [Hoffmann et al., 2021, Mazoure et al., 2022, Abdar et al., 2023].

Uncertainty quantification (UQ) plays a critical role in enhancing the reliability of predictions made by trained neural networks. Widely adopted across diverse research domains, UQ holds significant promise for applications in genome editing. These uncertainties are typically categorized into two types : Aleatoric uncertainty, which captures the randomness present in the data, and Epistemic uncertainty, which reflects the lack of knowledge or the model’s inability to generalize to new, unseen data.

Several well-known methods exist for measuring uncertainty, such as Monte Carlo dropout [Gal and Ghahramani, 2016], Variational Inference [Wang and Van Hoof, 2020, Rudner et al., 2022], Deep Ensembles

[D'Angelo and Fortuin, 2021], Test-Time Data Augmentation (TTA) [Wang et al., 2019d], and Bayesian Deep Ensembles [He et al., 2020].

Understanding and quantifying these uncertainties is crucial to improving model robustness, supporting informed decision-making under uncertainty, and optimizing performance in complex and dynamic environments. This technique is particularly relevant for improving the trustworthiness of predictions related to on- and off-target effects in genome editing. Kirillov et al. [Kirillov et al., 2022] were among the first to incorporate uncertainty into final predictions. Their method offers interpretable assessments of Cas9-gRNA and Cas12a-gRNA specificity using deep kernel learning, outputting cleavage efficiency estimates along with confidence intervals.

0.5 Transfer Learning in Genome Editing Prediction

Transfer learning is a powerful technique for situations where deep learning models must be developed using datasets of limited size. Rather than training a network entirely from scratch, this approach reuses knowledge acquired by a model trained on one task to aid in solving a different yet related task [Caruana, 1997]. Widely adopted across numerous domains, transfer learning enables the development of complex neural networks even in the presence of scarce labeled data. By leveraging representations learned from large and diverse datasets, it can substantially reduce training time and enhance model generalization, making it particularly valuable in fields where data collection is expensive, time-consuming, or technically challenging.

Transfer learning has become an essential strategy in modern deep learning, particularly in domains where large-scale labeled datasets are unavailable or impractical to obtain. Instead of training a neural network entirely from scratch, transfer learning reuses knowledge acquired by a model trained on a related source task to accelerate and improve learning on a target task [Caruana, 1997]. This paradigm allows researchers to leverage the representational power of models trained on massive, diverse datasets, thereby addressing the common challenge of limited sample sizes in specialized application areas. By doing so, transfer learning not only reduces the computational and data requirements but also mitigates overfitting, leading to improved generalization performance.

In the context of CRISPR/Cas9 genome editing, acquiring extensive, high-quality labeled datasets is often hindered by the experimental cost, complexity, and time required for validation of both on- and off-target effects. Although some CRISPR-Cas9 benchmark datasets for on- and off-target prediction are currently

available [Haeussler et al., 2016], the number of samples they contain is often insufficient to achieve accurate deep learning predictions.

Recent studies have explored transfer learning (TL) as a strategy to enhance off-target prediction in CRISPR-Cas9 systems. For example, Lin and Wong [2018] and Charlier et al. [2021] trained models on a large CRISPOR dataset (18,236 samples) and transferred the learned knowledge to a much smaller GUIDE-Seq dataset (430 samples). Similarly, Elkayam et al. [Elkayam and Orenstein, 2022] introduced *DeepCRISTL*, which was pretrained on high-throughput datasets comprising more than 150,000 gRNAs and subsequently fine-tuned through TL on small-scale functional or endogenous datasets. Zhang et al. [2020a] developed *C-RNNCrispr*, a hybrid CNN-RNN framework for sgRNA activity prediction, pretrained on benchmark data and adapted with small datasets using TL. In related work, Zhang et al. [2021] proposed two attention-based CNN models, *CRISPR-ONT* and *CRISPR-OFFT*, applying TL for sgRNA specificity prediction in limited cellular contexts.

Different CRISPR-Cas9 datasets are often collected under distinct laboratory conditions and equipment, which leads to variations in data patterns and distributions. Consequently, the prediction accuracy achieved by a transfer learning technique on a given target dataset depends strongly on the degree of similarity between this target dataset and the source dataset used during pretraining. In other words, employing source and target datasets that are substantially different is unlikely to yield satisfactory prediction results for the target data. This paper specifically addresses the challenge of evaluating and ensuring similarity between source and target datasets in transfer learning experiments for CRISPR-Cas9 off-target prediction.

0.6 Thesis content

My thesis is organized by articles. It includes an Introduction, three article chapters (Chapters I-III) and a Conclusion.

The Introduction provides the necessary background on CRISPR/Cas9 genome editing, outlines the key challenges such as off-target effects, and presents computational approaches including data-driven and deep learning models. It also defines the research objectives and explains the overall structure of the thesis.

Chapter I reviews recent studies on the effectiveness of artificial intelligence methods for on- and off-target activity prediction related to CRISPR/Cas9. Over the last few years, data-driven methods emerged as a new modelling approach that outperforms the common types of genome editing toolsets. One of the main

advantages of modern data-driven models that rely on deep learning is their ability to improve predictive performance as the number of input data grows. Due to the fast-developing computing power provided by graphic processing units (GPUs), deep learning algorithms have received increasing attention in research works year by year. Deep learning applications across all research fields have recently gained popularity due to easier access to data, boosted computing power, and recent progress in deep neural networks. In contrast to conventional machine learning models, deep learning models can automatically learn sequence features by generating new internal features that are crucial for accurate outcome predictions (see, for example, the convolutional deep learning models used by Lin et al. [Lin et al., 2020] and Shrawgi et al. [Shrawgi and Sisodia, 2019], and the discussion therein). Deep neural networks are at the core of deep learning. They are capable to learn complex patterns from the data using multiple layers of neurons connected together. Due to the fast-growing techniques of data collection and storage tools, CRISPR/Cas9 benchmark data sets are emerging, and large data are becoming increasingly available to the research community. The benchmark data are available for researchers on our GitHub repository at the following URL address : `https://github.com/dagrate/public_data_crisprCas9`.

Chapter II introduces a novel model called BayTTA : Uncertainty-aware medical image classification with optimized test-time augmentation using Bayesian model averaging. BayTTA propose a Bayesian Model Averaging (BMA)-based framework for optimizing test-time augmentation (TTA). The approach involves generating multiple predictions using augmented variants of the input, then combining them through BMA, with each prediction weighted by its posterior probability. This enables the integration of model uncertainty, ultimately improving the robustness and accuracy of predictions. The model's effectiveness is demonstrated across multiple datasets, including three medical image classification tasks (skin cancer, breast cancer, and chest X-ray images) and two prominent genome editing datasets : CRISPOR and GUIDE-seq. The source code is freely available at : `https://github.com/Z-Sherkat/BayTTA`.

Chapter III presents a similarity-based approach aimed at enhancing the effectiveness of transfer learning for improving CRISPR-Cas9 off-target predictions. In this context, a model trained on a dataset for one specific application can be adapted to make predictions on a different dataset within a similar domain. For example, Lin et al. [Lin and Wong, 2018] and Charlier et al. [Charlier et al., 2021] have successfully applied transfer learning techniques to predict off-target CRISPR sequences using relatively small datasets. Our proposed method builds upon this foundation by incorporating sequence similarity information to further improve model generalizability and prediction accuracy across different CRISPR-Cas9 datasets. The code and data used

in this study are freely available at : `https://github.com/dagrate/transferlearning_offtargets`.

The Conclusion summarizes the main findings from the three article chapters, highlights their contributions to improving prediction accuracy, discusses the broader implications for CRISPR/Cas9 applications, and outlines future research directions.

CHAPITRE 1

USING TRADITIONAL MACHINE LEARNING AND DEEP LEARNING METHODS FOR ON- AND OFF-TARGET PREDICTION IN CRISPR/CAS9 : A REVIEW, BRIEFINGS IN BIOINFORMATICS

This chapter is a reproduction of the following article : Zeinab Sherkatghanad, Moloud Abdar, Jeremy Charlier, Vladimir Makarenkov, "Using traditional machine learning and deep learning methods for on- and off-target prediction in CRISPR/Cas9 : a review, Briefings in Bioinformatics", Volume 24, Issue 3, May 2023, bbad131,

1.1 Abstract

1.1.1 Motivation

CRISPR/Cas9 (Clustered Regularly Interspaced Short Palindromic Repeats and CRISPR-associated protein 9) is a popular and effective two-component technology used for targeted genetic manipulation. It is currently the most versatile and accurate method of gene and genome editing, which benefits from a large variety of practical applications. For example, in biomedicine it has been used in research related to cancer, virus infections, pathogen detection and genetic diseases. Recent CRISPR/Cas9 research is based on data-driven models for on- and off-target prediction as a cleavage may occur at non-target sequence locations. Currently, conventional machine learning and deep learning methods are applied on a regular basis to accurately predict the sgRNA (single-guide RNA) on-target knockout efficacy and off-target profile.

1.1.2 Results

We present an overview and a comparative analysis of traditional machine learning and deep learning models used in CRISPR/Cas9. We highlight the key research challenges and directions associated with target activity prediction. We discuss some recent advances in the sgRNA-DNA sequence encoding used in state-of-the-art on- and off-target prediction models. Furthermore, we present the most popular deep learning neural network architectures used in CRISPR/Cas9 prediction models. Finally, we summarize existing challenges and discuss possible future investigations in the field of on- and off-target prediction.

1.1.3 Conclusions

Our paper provides valuable support for academic and industrial researchers interested in the application of machine learning methods in the field of CRISPR/Cas9 genome editing.

1.1.4 Availability and Implementation

The benchmark data are available for researchers on our GitHub repository at the following URL address : https ://github.com/dagrate/public_data_crisprCas9.

1.2 Introduction

Advances in the area of genome editing (also called gene editing) in the 2010s revolutionized molecular biology, genetics and biomedicine. Genome editing techniques allow precise manipulation, deletion and insertion of sequence fragments within the DNA of living organisms. In recent years, three effective types of genome editing toolsets, called Zinc Finger Nucleases (ZFNs)[Esvelt and Wang, 2013], Transcription Activator-Like Effector Nucleases (TALENs), and Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR), have been developed to study the process of target modifications in gene sequences [Esvelt and Wang, 2013, Puchta and Fauser, 2013, Barrangou and Doudna, 2016, Manghwar et al., 2019, Bogdanove et al., 2018].

Highly effective CRISPR/Cas9 gene editing system, co-invented in 2012 by Emmanuelle Charpentier and Jennifer Doudna [Jinek et al., 2012], has been used in various fields, ranging from basic research on genetic therapies at the cellular level to applied biomedical research [Jinek et al., 2012, Cho et al., 2013, Cong et al., 2013, Mali et al., 2013, Chang et al., 2013, Hsu et al., 2014]. CRISPR/Cas9 demonstrated important clinical potential for treating human diseases such as cancer and genetic disorders [Kang et al., 2017, Liang et al., 2015, Ma et al., 2017], for plant genetic engineering [Liu et al., 2017, Tang et al., 2016, Raitskin and Patron, 2016], and for animal disease treatment [Zarei et al., 2019, Wang et al., 2013]. Fig. 1.1 presents a schematic view of the CRISPR/Cas9 gene editing system and its practical applications.

The CRISPR/Cas9 genetic engineering system is an adapted version of the bacterial CRISPR-Cas9 antiviral defense system. CRISPR is a family of DNA sequences present in prokaryotic genomes that stems from DNA fragments of bacteriophages which had infected prokaryotic genomes in the past. These DNAs are used as antiviral defense elements to recognize and eliminate DNA from similar bacteriophages during eventual

infections [Barrangou et al., 2007]. Cas9 is a type of nuclease enzyme that uses CRISPR sequences as a guide to identify and cleave specific DNA fragments that are complementary to a given CRISPR sequence.

In the CRISPR/Cas9 gene editing system, the Cas9 nuclease combined with a guide RNA (gRNA) is delivered into a cell, allowing the cell's genome to be cut in a specific location, some targeted genes to be removed from it and some other added to it in vivo [Bak et al., 2018]. Guide RNA, or artificially programmed single guide RNA (sgRNA) used in type II CRISPR/Cas9 systems, is responsible for identifying the target DNA sequence in the cell's genome and ensuring that the cutting takes place at the desired sequence location. The Protospacer-Adjacent Motif (PAM), located at the end of the DNA target site, is a short three-to-five nucleobase sequence serving as the binding signal of the Cas protein [Shah et al., 2013].

The CRISPR/Cas9 system is nonetheless prone to unintended off-targets : a cleavage may occur at non-target locations [Zhang et al., 2015, Chen et al., 2017, Kim et al., 2015]. Thus, the safety aspect of the use of CRISPR/Cas9 on humans remains an open issue. The main challenge in the effective application of the CRISPR/Cas9 system is to maximize on-target activity (i.e., guide efficiency) and minimize the number of potential off-targets (i.e., guide specificity).

Over the last few years, data-driven machine learning methods emerged as a new modeling approach which outperforms the common scoring prediction methods, such as MIT CRISPR Design Tool2 [Ran et al., 2013], CCTop algorithm [Stemmer et al., 2015], CRISPR Design [Hsu et al., 2013], E-CRISP [Heigwer et al., 2014], and CHOPCHOP [Montague et al., 2014], in terms of on- and off-target prediction. One of the main drawbacks of the latter methods is the lack of capacity to increase the prediction accuracy when the number of samples increases. In contrast, one of the main advantages of modern data-driven models that rely on deep learning is their ability to improve the predictive performance as the number of samples grows. Current state-of-the-art research aiming at designing robust clinical CRISPR/Cas9 applications looks for enhancing data-driven models by : (i) increasing on-target efficiency, (ii) improving off-target specificity, and (iii) simultaneously maximizing on-target activity and minimizing off-target effects.

Among recent reviews and comparative studies discussing the use of computational methods for evaluating guide RNA efficiency and predicting guide RNA specificity, we need to mention the following works. Wang et al. [2019b] highlighted new advances in CRISPR-Cas systems in terms of RNA targeting, tracking, and editing. The authors compared Cas protein-based technologies with traditional technologies intended for

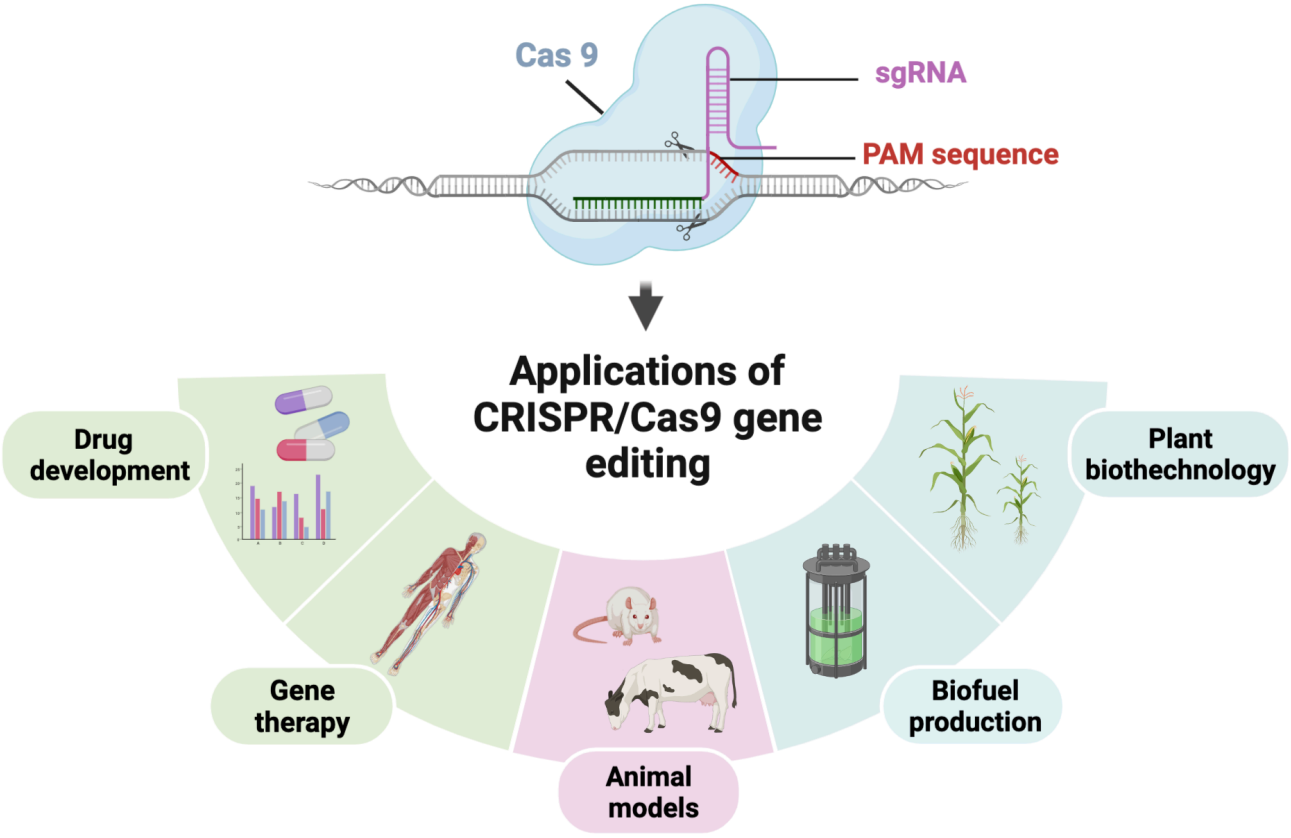

Figure 1.1 – Schematic view of CRISPR/Cas9 gene editing system and its practical applications.

these goals. Liu et al. [2020a] provided an up-to-date overview of computational methods for guide RNA design, including web-based platforms, to help researchers to select optimal tools for their CRISPR-Cas experiments. However, among 14 methods for evaluating gRNA efficiency and 13 methods for predicting gRNA specificity considered by Liu et al. only one of them employs deep learning. Chen et al. [2020b] first briefly reviewed the main properties of CRISPR systems and their use in genome editing. Then, the authors discussed feasible methods for detecting potential off-targets during CRISPR/Cas9 genetic manipulations. Yan et al. [2020] compared 17 in silico off-target prediction tools in order to evaluate their genome-wide CRISPR performances, and introduced an integrated Genome-Wide Off-target cleavage Search (iGWOS) platform designed for optimal genome-wide off-target predictions. The main goal of the study by Yaish et al. [2022] was to systematically evaluate data pre-processing and formulation of the CRISPR off-target prediction problem. The authors pointed out that data transformation is a crucial data pre-processing step that should be applied prior to model training. They highlighted the importance of considering as model's features inactive off-target sites and the number of mismatches between gRNAs and their off-target sites. Moreover, Yaish et al. introduced predictive off-target in cellula models based on gradient boosting (i.e., the XGBoost decision tree-based ensemble learning framework was implemented) and compared them to state-of-the-art off-target prediction methods. The paper of O'Brien et al. [2021] presents the main machine learning approaches and pitfalls in the context of CRISPR-Cas9 experiments. The authors consider the related computational problems, including algorithm choice, accuracy overestimation, and data interoperability. They concluded that thanks to the broad availability of machine learning-based tools, in silico optimization can successfully replace in vitro CRISPR-Cas9 designs. Thus, algorithmic solutions can be used for maximizing gRNA editing efficiency and minimizing gRNA specificity. Finally, a recent review of Konstantakos et al. [2022b] addresses the problem of CRISPR–Cas9 gRNA efficiency prediction and evaluates the role of deep learning in this context. The authors discussed the main computational approaches for on-target activity prediction, focusing on the selection of optimal features and algorithms. In their comparative experiments, Konstantakos et al. assessed the performances of 10 deep learning and one conventional machine learning methods on six benchmark data sets, and provided recommendations for their use. The authors pointed out that the existing on-target prediction approaches still have some flaws, including their sensitivity to data heterogeneity, unclear decision making mechanism, and inability to produce general gRNA design rules. Some recent works in the field include [Pacesa et al., 2024, Eggers et al., 2024, Villiger et al., 2024].

In this review paper, we provide a summary of studies that have examined the effectiveness of artificial intelligence methods for on- and off-target activity prediction related to CRISPR/Cas9. In contrast to some

previous reviews [Liu et al., 2020a, Yan et al., 2020, Konstantakos et al., 2022b, Naeem et al., 2020, Almutiri et al., 2022, Wilson et al., 2018, Wang et al., 2020b, Newman et al., 2020], our study discusses the use of both traditional machine learning and deep learning methods, focusing on the latest state-of-the-art on- and off-target prediction models. We describe the main advantages and disadvantage of existing prediction models, while highlighting noticeable progress that has been made in sequence encoding.
Our main contributions are as follows :

— To the best of our knowledge, this is the first comprehensive review of both traditional machine learning and deep learning methods, used for both on-target (guide efficiency) and off-target (guide specificity) outcome prediction in CRISPR/Cas9 gene editing ;

— A description of the benchmark data sets used for on- and off-target prediction in CRISPR/Cas9 is provided. Most of the discussed data sets were curated and made available for researchers on our GitHub repository ;

— The main sequence encoding techniques used in CRISPR/Cas9, applying to both traditional machine learning and deep learning methods, are discussed along with their specific properties ;

— The main deep learning models used for on- and off-target prediction in CRISPR/Cas9 are presented and their properties and limitations are highlighted ;

— Research challenges and avenues of future investigation regarding the application of traditional machine learning and deep learning methods in the field of CRISPR/Cas9 gene editing are discussed.

1.3 Data description

In this section, we present the most popular CRISPR/Cas9 benchmark data sets used in the literature for on- and off-target prediction. These data sets can be divided into three categories : data sets including off-targets only, on-targets only, and both off- and on-targets.

The first group of benchmark data sets consists of off-targets. GUIDE-seq was one of the first off-target data repositories, based on the results of the GUIDE-seq technique developed by Tsai et al. [2015]. It can serve as an accurate framework for genome-wide identification of off-target effects. The sgRNAs used in GUIDE-seq target the following sites : VEGFA site 1, VEGFA site 2, VEGFA site 3, FANCF, HEK293 site 2, HEK293 site 3, and HEK293 site 4, in which 28 off-targets with a minimum modification frequency of 0.1 were identified (among 403 potential off-targets).

The CIRCLE-Seq (Circularization for In vitro Reporting of CLeavage Effects by sequencing) screening strategy

introduced by Tsai et al. [2017] was used to analyze the related data set that includes gRNA-DNA pairs for 10 gRNA sequences with the corresponding mismatch, insertion, and deletion information ; 7,371 of these sequence pairs were identified as active off-targets.

Cameron et al. [2017] proposed the SITE-Seq biochemical method that uses Cas9 programmed with sgRNAs to recognize cut sites within genomic DNA. The related data set contains sgRNA-DNA sequence pairs for 9 guide sequences ; 3,767 of these sequence pairs correspond to active off-targets. Abadi et al. [2017] collected a training data set based on three genome-wide methods for unbiased CRISPR-Cas9 cleavage sites profiling, which are as follows : (i) Genome-wide unbiased identification of DSBs enabled by sequencing (GUIDE-Seq) [Tsai et al., 2015, Kleinstiver et al., 2016], (ii) High-throughput genome-wide translocation sequencing (HTGTS) [Frock et al., 2015], and (iii) Breaks labeling, enrichment on streptavidin and next-generation sequencing (BLESS) [Ran et al., 2015, Slaymaker et al., 2016]. The resulting data set was assembled from the five following studies : [Tsai et al., 2015, Kleinstiver et al., 2016, Frock et al., 2015, Ran et al., 2015, Slaymaker et al., 2016]. It includes 33 collections of sgRNAs with their respective targets. Altogether, these sgRNAs cleaved 872 genomic targets across human genome. Lazzarotto et al. [2020] applied their CHANGE-seq automatable tagmentation-based method to analyse the related in vitro Cas9 genome-wide nuclease activity data set. CHANGE-seq was carried out to analyze 110 sgRNA targets across 13 therapeutically relevant loci in human primary T cells. A total of 201,934 off-target sites were identified with variable numbers of off-target sites, ranging from 19 to 61,415, for an individual sgRNA.

The second group of benchmark data sets consists of on-targets. Wang et al. [2014] used a practical library containing $73,000$ sgRNAs to generate knockout collections and to investigate screens in human cell lines HL-60 and KBM7. The authors tested both ribosomal and non-ribosomal protein coding genes with all possible sgRNAs gathered in the library. Koike-Yusa et al. [2014] conducted their investigation on a data set that consisted of $87,897$ gRNAs targeting $19,150$ mouse protein-coding genes. They designed genome-wide mutant mouse embryonic stem cell libraries to identify unknown host factors that modulate toxin susceptibility. The Doench V1 data set [Doench et al., 2014] consists of 1,831 guides targeting three human (CD13, CD15, and CD33) and six mouse (Cd5, Cd28, H2-K, Cd45, Thy1, and Cd43) genes, all producing cell-surface markers which could be assayed by flow cytometry. GenomeCRISPR is a well-formatted data repository, organized by Rauscher et al. [2016], which was designed for high-throughput CRISPR screening studies. GenomeCRISPR contains over 550,000 sgRNAs on-targets derived from 84 different experiments. Wang et al. [2019a] used the DeepHf (Deep learning for High-Fidelity Cas9) method to perform a genome-scale screen

measuring gRNA activity of two highly specific SpCas9 variants (eSpCas9(1.1) and SpCas9-HF1), and a wild-type SpCas9 (WT-SpCas9) in human cells. The obtained data set contains indel rates for over 50,000 gRNAs for each nuclease, covering about 20,000 genes. It is the largest gRNA on-target activity data set reported to date for mammalian cells. Kim et al. [2019] generated a data set of SpCas9 activities at 12,832 target sequences from a human cell library using the deep learning–based DeepSpCas9 model. The DeepSpCas9 target sequences were chosen from the human genome and synthetic sequences without using any information related to the activity of the associated sgRNAs. The sgDesigner data is a unique plasmid target library expressed in human cells that was used by Hiranniramol et al. [2020] for experimental quantification of sgRNA CRISPR/Cas9 efficiency. A pool of 12,472 oligonucleotides was used to train a machine learning algorithm for assay design.

The third group of benchmark data sets considered in the literature includes data containing both off- and on-target information. First, we need to mention the RESistance assays (RES) data set, i.e., Doench V2, made available by Doench et al. [2016]. It consists of 2,549 unique guides targeting eight genes (i.e. CCDC101, MED12, TADA2B, TADA1, HPRT, CUL3, NF1, and NF2) from Human A375 cells.

The well-known CRISPOR database organized and maintained by Haeussler et al. [Haeussler et al., 2016] aggregates different public data sets that have been widely used to quantify on-target guide efficiency and detect off-target cleavage sites, including : Wang-Xu et al. data set (2,076 guides targeting 221 genes in Human HL-60 cells) [Wang et al., 2014, Xu et al., 2015], Koike-Yusa et al. data set [Koike-Yusa et al., 2014], Doench V1 and V2 data sets [Doench et al., 2014, 2016], Hart et al. data set (4,239 guides targeting 829 genes in Human Hct116 cells) [Hart et al., 2015], Z_fish MM data set (1,020 guides targeting 128 genes in Zebrafish genome) [Moreno-Mateos et al., 2015], Z_fish VZ data set (102 guides targeting different genes in Zebrafish genome) [Varshney et al., 2015], Z_fish GZ data set (111 guides targeting different genes in Zebrafish genome) [Gagnon et al., 2014], Drosophila data set [Ren et al., 2014], Chari et al. data set (1234 guides targeting Human 293T cells) [Chari et al., 2015], Ciona data set (72 guides targeting different genes in Ciona genome) [Gandhi et al., 2016], Farboud et al. data set (50 guides targeting different genes in Caenorhabditis elegans genome) [Farboud and Meyer, 2015]. Moreover, Haeussler et al. [2016] developed the CRISPOR web tool (available at : crispor.org) that is intended to design, evaluate and clone guide sequences for the CRISPR/Cas9 system. This web tool incorporates several on- and off-target scoring algorithms. It also displays pre-calculated results for all human exons from the UCSC Genome Browser tracks. On the first page of crispor.org, the user enters three pieces of information : (i) a single genomic sequence, typically an exon under 2300 bp ; (ii) a genome (from the list of more than 150 genomes, including plants and emerging model organisms) ; (iii) a PAM motif.

The main output of CRISPOR is a web page that shows the annotated input sequence and the list of possible guides in the input sequence. Furthermore, CRISPOR also generates a list of primers related to a selected guide. The related GitHub data repository organized by Haeussler comprises direct links to 22 experimental data sets along with some necessary data conversion scripts written in Python.

Munoz et al. [2016] designed the CRISPR tiling library, which is a large tiling-sgRNA data set containing data for 139 genes with an average of 364 sgRNAs/gene for three cancer cell lines DLD1, RKO, and NCI-H1299. Furthermore, the DeepCRISPR data set generated by Chuai et al. [2018] includes approximately 0.68 billion sgRNA sequences derived from 13 human cell lines, including HEK293, MCF-7, K562, HL60, NB4, BE2C, Caco-2, GM06990, Hela, HCT116, LNCap, HepG2, and GM12878. This large data set comprises epigenetic information for different cell types, providing a unified feature space which combines the data from various experiments and cell types. The related DeepCRISPR software includes the models for both sgRNA on-target knockout efficacy and genome-wide off-target cleavage profile prediction.

In Table 1.1, we present a summary of the main features of the CRISPR/Cas9 benchmark data sets used for on- and off-target prediction, including the original article describing the data set in question, the URL link to the data set, and the prediction target. We curated the benchmark data reported in Table 1.1 and made them available for researchers on our GitHub repository at the following URL address : https ://github.com/dagrate/public_data_crisprCas9. Moreover, several data sets presented here have been one-hot encoded and prepared for use in machine learning and deep learning experiments. The related Python scripts have been also made available to the scientific community.

In conclusion, we think that using the latest benchmark data containing large amounts of samples and features is likely to facilitate future work in CRISPR/Cas9, since such data can provide wide and complete coverage of intrinsic properties of both off- and on-targets under study, and thus be effectively exploited by state-of-the-art machine learning and deep learning methods.

Table 1.1 – A summary of the most popular CRISPR/Cas9 benchmark data sets and databases used for on- and off-target prediction.

Source	Year	Data description	Target	Data link
Wang et al. data [Wang et al., 2014]	2014	A library containing $73,000$ sgRNAs	On-targets	https ://www.ncbi.nlm.nih.gov/pmc/articles/

				PMC3972032/#SD2
Koike-Yusa et al. data [Koike-Yusa et al., 2014]	2014	$87,897$ gRNAs targeting $19,150$ mouse protein-coding genes	On-targets	Deposited at the European Nucleotide Archive under accession number ERP003292.
Doench V1 data [Doench et al., 2014]	2014	1,831 guides targeting three human (CD13, CD15, and CD33) and six mouse genes (Cd5, Cd28, H2-K, Cd45, Thy1, and Cd43)	On-targets	www.broadinstitute.org/rnai/public/analysis-tools/sgrna-design
GUIDE-seq data [Tsai et al., 2015]	2015	CRISPR RNA-guided nucleases (RGNs) from two human cell lines : U2OS and HEK293 ; different sites such as VEGFA sites 1, 2 and 3, and HEK293 sites 2, 3 and 4 were studied	Off-targets	https ://github.com/tsailabSJ/guideseq
Doench V2 data [Doench et al., 2016]	2016	2,549 unique guides targeting eight genes (CCDC101, MED12, TADA2B, TADA1, HPRT, CUL3, NF1, and NF2) from human A375 cells	Off-targets and on-targets	https ://www.nature.com/articles/nbt.3437
CRISPOR program + data repository [Haeussler et al., 2016]	2016	Aggregate data for more than 150 genomes, including the following public data sets : Wang-Xu [Wang et al., 2014, Xu et al., 2015], Koike-Yusa [Koike-Yusa et al., 2014], Doench V1 and V2 [Doench et al., 2014, 2016], Hart [Hart et al., 2015], Z_fish MM [Moreno-Mateos et al., 2015], Z_fish VZ [Varshney et al., 2015], Z_fish GZ [Gagnon et al., 2014], Drosophila [Ren et al., 2014], Chari [Chari et al., 2015], Ciona [Gandhi et al., 2016] Farboud [Farboud and Meyer, 2015]	Off-targets and on-targets	http ://crispor.org + https ://github.com/maximilianh/crisporPaper/tree/master/effData#readme
GenomeCRISPR database [Rauscher et al., 2016]	2016	Aggregate data for more than 550,000 sgRNAs derived from 84 experiments	On-targets	http ://genomecrispr.org
CIRCLE-Seq data [Tsai et al., 2017]	2017	Contains mismatch, insertion, and deletion information, and includes sgRNA-DNA pairs from 10 guide sequences, 7,371 of which are off-targets (430 with bulges)	Off-targets	https ://github.com/tsailabSJ/circleseq
SITE-Seq data [Cameron et al., 2017]	2017	gRNA-DNA pairs from nine guide sequences, 3,767 of which are active off-targets (no bulges)	Off-targets	https ://experiments.springernature.com/articles/10.1038/nmeth.4284

Abadi et al. [Abadi et al., 2017]	2017	A data set based on three genome-wide methods for unbiased CRISPR-Cas9 cleavage sites profiling : GUIDE-Seq, HTGTS, and BLESS. It includes 33 collections of sgRNAs with their respective targets	Off-targets	https ://journals.plos.org/ploscompbiol/article ?id= 10.1371/journal.pcbi.1005807
DeepCRISPR [Chuai et al., 2018] platform	2018	Includes approximately 0.68 billion sgRNA sequences derived from 13 human cell lines	Off-targets and on-targets	https ://github.com/bm2-lab/DeepCRISPR
DeepHf data [Wang et al., 2019a]	2019	Includes indel rates of over 50,000 gRNAs for each nuclease, covering about 20,000 genes. It is the largest gRNA on-target activity set reported for mammalian cells	On-targets	http ://www.DeepHF.com
DeepSpCas9 data [Kim et al., 2019]	2019	A dataset of SpCas9 activities at 12,832 integrated target sequences for a human cell library	On-targets	http ://deepcrispr.info/DeepSpCas9
sgDesigner data [Hiranniramol et al., 2020]	2020	A unique plasmid library expressed in human cells was used to quantify the potency of thousands of CRISPR/Cas9 sgRNAs (a pool of 12,472 oligonucleotides was analyzed)	On-targets	https ://academic.oup.com/bioinformatics/article/36/9/2684/5714741 ?login= false#supplementary-data
CHANGE-seq data [Lazzarotto et al., 2020]	2020	110 sgRNA targets across 13 therapeutically relevant loci in human primary T-cells were studied to identify 201,934 off-target sites across the human genome	Off-targets	https ://github.com/tsailabSJ/changeseq

1.4 sgRNA-DNA sequence encoding

Before building Artificial Intelligence (AI) models intended for on- and off-target prediction, the sgRNA-DNA sequence data must be pre-processed to be used as input. Data pre-processing, or data encoding, allows converting the sgRNA-DNA sequences of letters into sequences of numbers that AI models can read and interpret to build their predictions. Data pre-processing is an important milestone when trying to boost the predictive performance of AI models. The two most popular encoding techniques used in CRISPR-Cas9 are : (a) One-hot encoding and (b) Word embedding.

Figure 1.2 highlights the differences between the two techniques. In one-hot encoding, each possible channel

A, C, G or T is represented by a one-hot vector such as [1,0,0,0], [0,1,0,0], [0,0,1,0], and [0,0,0,1]. In embedding, a particular word, or string, is represented using a unique vector representation. A sgRNA-DNA sequence, which can be subdivided into substrings of length k, called k-mers, can be thus transformed into a vector representation thanks to word embedding. The most popular embedding technique is Word2Vec [Mikolov et al., 2013]. This natural language processing technique relies on the use of neural networks. In this review, we first discuss some recent papers that use one-hot encoding schemes in CRISPR-Cas9, followed by a brief overview of papers dealing with word embedding, and by the section presenting further sequence characteristics often used as additional explanatory features in ML and DL models.

It is worth noting that most of the sequence encoding schemes discussed in this section apply to both traditional machine learning and deep learning methods. For example, the recent works of Lin et al., Wang et al., Lin et al., and Charlier et al. [Lin and Wong, 2018, Wang et al., 2019a, Lin et al., 2020, Charlier et al., 2021] use the same sequence encoding schemes to provide the input of both machine learning and deep learning algorithms compared in these papers.

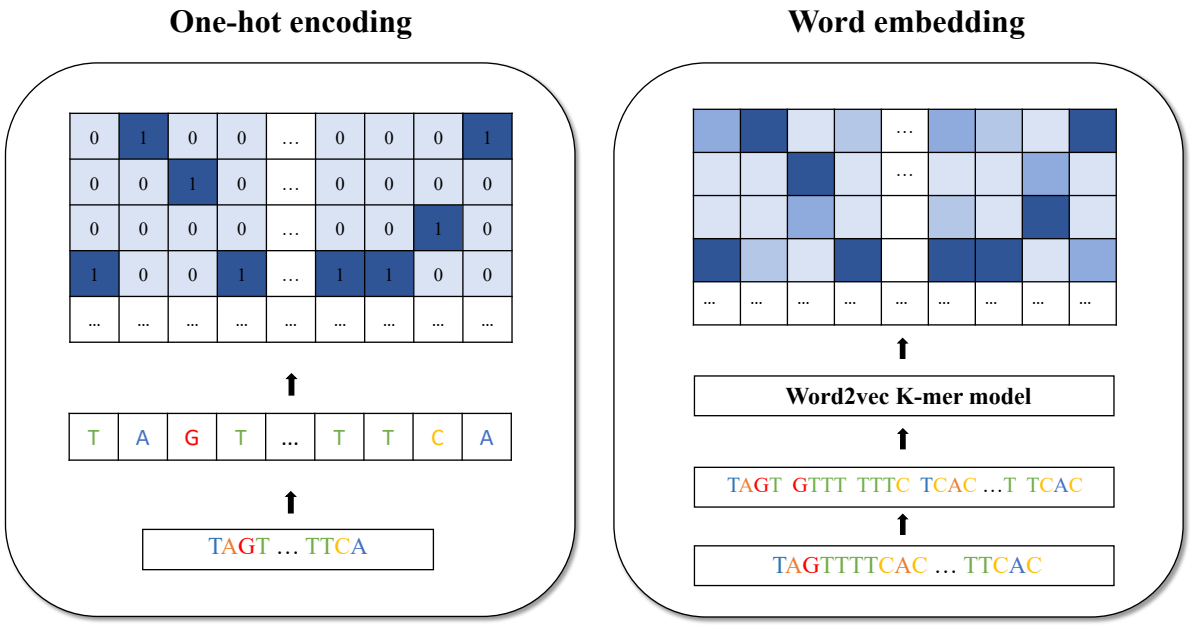

Figure 1.2 – Two sequence encoding models used in CRISPR/Cas9 : one-hot encoding and word embedding.

In this paragraph, we present recent papers using novel one-hot encoding techniques in the field of genome editing. When applying a one-hot encoding, each sgRNA-DNA sequence pair of length L is encoded in a

one-hot matrix of four rows and L columns. Each row corresponds to the nucleotide type, i.e. A, C, G, and T. Each base in the sgRNA and the target DNA is then encoded in the form of a one-hot vector according to one of various methods.

Chuai et al. [2018] proposed the deepCRISPR model for sgRNA on- and off-target prediction. DeepCRISPR relies on a deep convolutionary denoising neural network and one-hot data pre-processing. The nucleotide sequence is a 20-bp sgRNA sequence with an NGG PAM across the human genome. It is represented by four channels, (A, C, G, and T), and each epigenetic feature is considered as one channel. Thus, the encoded matrix used by Chuai et al. is of size $(4 + n) \times 23$, where 4 corresponds to the number of channels and n to the number of epigenetic features.

Lin and Wong [2018] introduced a one-hot sequence encoding method that converts each sgRNA-DNA sequence pair into a matrix to be used as a convolutional input. In their method the four channels are used to represent both sgRNA and target DNA. Thus, each character in the sgRNA and target DNA sequences is encoded as a single one-hot vector. Consequently, every sgRNA-DNA sequence pair is encoded in a matrix of size 4×23, where 23 corresponds to the 3-bp PAM adjacent to the 20 bases. The use of such 4×23 input matrices allowed the authors to apply for the first time deep FNNs (Feedforward Neural Networks) and deep CNNs (Convolutional Neural Networks) for off-target prediction in CRISPR-Cas9 gene editing.

Charlier et al. [2021] described a different novel one-hot encoding method. Their main idea was to build a data encoding procedure that relies on a bijective mapping for sgRNA-DNA sequence pairs. It allows for encoding, and decoding, of the sgRNA-DNA sequence pairs without any information loss that can occur in the encoding scheme adopted in [Lin and Wong, 2018]. Specifically, Charlier et al. combined a 4×23 matrix used for sgRNA encoding and a 4×23 matrix used for DNA encoding, resulting in a 8×23 matrix used as a convolutional input. The authors applied FNNs, CNNs, and RNNs (Recurrent Neural Networks) to generate accurate off-target predictions.

Lin et al. [2020] have recently introduced an encoding technique capable of incorporating base mismatch, missing base (RNA bulge or insertion), and extra-base (DNA bulge or deletion) in off-target sites. Each sequence pair was considered as a fixed length vector with the following five-bit channel : (A, C, G, T, _). Additionally, the authors considered a two-bit direction channel used to identify the indel and mismatch directions. Thus, a combined seven-bit channel, encoded as seven one-hot encoded vectors, allowed them to

take into account not only sgRNA-DNA sequence mismatches, but insertions and deletions as well. Precisely, Lin et al. [2020] used a 7×23 matrix encoding scheme (see Fig. 1.3), where 23 is the length of the sgRNA-DNA sequence pairs. This encoding scheme is a perfect example of feature engineering, i.e., new feature construction process that is explicitly defined and manually or automatically applied. Feature engineering is common in machine learning. Moreover, some machine learning methods, e.g., SVMs (Support Vector Machine), incorporate feature engineering as part of their operation [Heaton, 2018]. In the case of the sgRNA-DNA sequence encoding proposed by Lin et al. the created seven-bit-long features allow one to take into account all possible correspondences existing between the original sgRNA-DNA pairs of features taking the values (A, C, G, T, _). This innovative encoding scheme was used with different deep learning models for off-target prediction on CIRCLE-Seq and GUIDE-Seq data sets, and demonstrated state-of-the-art prediction performance.

Zhang et al. [2020c] designed an encoding scheme consisting of a matrix of size $20 \times L$, with L being the sequence length. In the encoding process, the authors used a four-bit channel (A, C, G, T) for sgRNA encoding, a four-bit channel (A, C, G, T) for DNA encoding, and a twelve-bit channel to one-hot encode all possible mismatches. They then regrouped the three corresponding matrices, resulting in a final matrix of size $20 \times L$. This extended matrix was then used for data augmentation to reduce the class imbalance between off-targets and on-targets, while a CNN model was used for on-target activity prediction.

Zhang and Jiang [2022] proposed another encoding scheme with a similar objective to incorporate mismatch, DNA and RNA bulge information into different off-target prediction models. The authors first considered a four-bit channel (A, C, G, T) and a one-hot vector encoding scheme. Furthermore, they used a two-bit channel to indicate a base deletion on RNA and DNA, and another one-bit function channel to indicate if the location is part of the guide sequence (0) or the PAM sequence (1). The encoded matrix was thus of size 7×23. Finally, an "OR" operation was carried out to indicate when two bases in a base pair were identical. Zhang et al. tested their encoding scheme with different FNN, CNN, and RNN models. They demonstrated performance on par with state-of-the-art.

Overall, among the different one-hot encoding schemes found in the literature, the highest potential has been recently demonstrated by those relying on the use of indels.

The second common data pre-processing strategy used by several researchers is a Natural Language Pro-

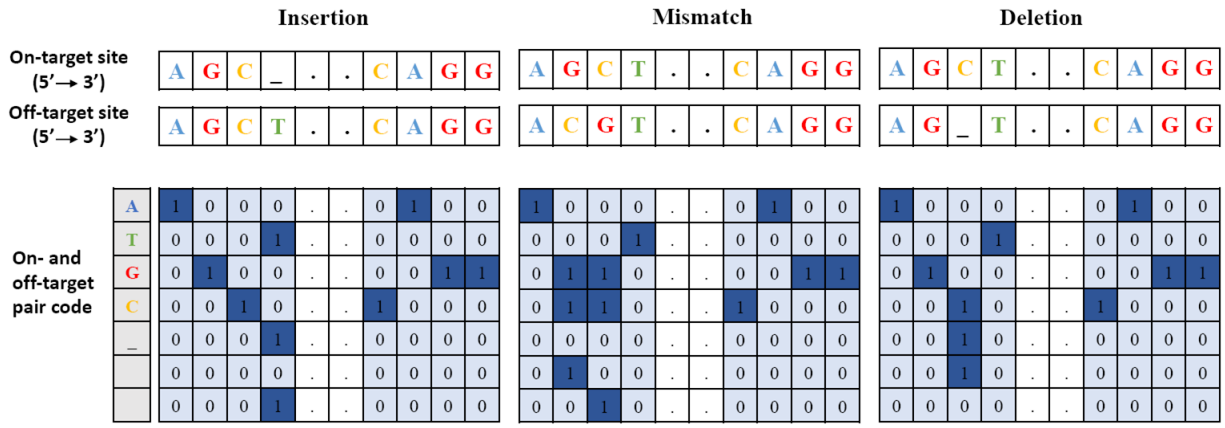

Figure 1.3 – A novel effective sgRNA-DNA one-hot sequence encoding scheme used by Lin et al. [2020]. A seven-bit encoding example is shown. Here, "_" symbol indicates the DNA or RNA bulge position. Each sgRNA-DNA sequence pair is encoded as a fixed-length seven-row matrix that includes a five-bit character channel (A, G, C, T, _) and a two-bit direction channel. The five-bit channel is used to encode the on- and off-target site nucleotides, whereas the direction channel is used to indicate the mismatch and indel locations.

cessing (NLP) technique, called word embedding. The idea of applying word embedding on sgRNAs is that off-targets are encoded closer to each other in the vector space than are on-targets.

Liu et al. [2019] combined the word embedding with a transformer to convey sgRNA sequences to a deep neural network model consisting of CNNs and FNNs. The authors demonstrated that the word embedding approach had similar predictive performance as the latest one-hot encoding-based deep learning models. Later on, Liu et al. [2020b] proposed to use a trained unsupervised learning algorithm, GloVe [Pennington et al., 2014], designed to aggregate word-to-word occurrence statistics outputting linear substructures of the word vector space. The authors applied GloVe to convert sgRNA sequences into substructures of the word vector space. They forwarded the sgRNA word vectors to a bidirectional LSTM and a CNN with five convolutional layers to predict the sgRNA off-target propensity, and demonstrated state-of-the-art predictive performance of their models.

Zhang et al. [2021] thus proposed to label-encode and word-embed sgRNA sequences. Each sgRNA sequence was transformed into a numerical vector using the Tokenizer module from the Keras library [Chollet et al., 2015]. Encoded sequences were then passed to pooling and convolutional layers, and to three convolutional layers to obtain sgRNA cleavage efficiency predictions. Using word-embedding techniques from NLP is fairly

recent, but we are confident that it opens new perspectives for future work in the field of genome editing.

One-hot encoding and word embedding are not the only ways to represent a sequence in machine learning models. The majority of conventional machine learning models used in CRISPR-Cas9 consider additional sequence features such as position-specific features, different counts, and structural/thermodynamic characteristics to best capture sequence information (see Table 1.2).

In contrast to conventional machine learning models, deep learning models can automatically learn the sequence features by generating new internal features that are crucial for accurate outcome predictions (see, for example, the convolutional deep learning models used by Lin et al. [2020] and Shrawgi and Sisodia [2019], and the discussion therein).

Here, we briefly recall some important works in the field and highlight sequence features that helped to boost the performance of the related machine learning models.

Doench et al. [2014], Xu et al. [2015] and Peng et al. [2018] identified several sgRNA and target DNA features allowing one to improve the prediction results, including position-specific nucleotide composition for individual (order 1) nucleotides (A/C/T/G) and pairwise (order 2) nucleotides (AA/AT/AG/...) in 30mer sequences (i.e. the 20mer guide plus context on either side), and GC counts for each guide sequence.

Afterwards, Doench et al. [2016] used thermodynamic features and position-independent features in addition to previously considered position-specific features and the GC counts (Doench et al. [2014]). Position-independent features included individual nucleotide counts (order 1) and adjacent pairwise nucleotide counts (order 2), ignoring their position in the sgRNA. Thermodynamic features were computed from the melting temperatures of the DNA version of the RNA guide sequence, or portions thereof, using the Biopython Tm_staluc function [Doench et al., 2016]. Rahman and Rahman [2017] considered position-specific features, position-independent features, as well as sgRNAs secondary structures to create additional features for their ML models.

The structural features used by Rahman et al., which are also known as thermodynamic properties, included Minimum Free Energy (MFE), the most favourable thermodynamic RNA-RNA interaction energy, local pair probabilities, and specific sgRNA heat parameters.

In their CRISPRO experiments, Schoonenberg et al. [2018] considered several categorical and numerical features. The categorical features used in this work include targeted amino acids 1 and 2, domain occupancy status (InterPro), exon multiple of 3, the ability of targeted transcript to escape nonsense-mediated decay, single nucleotide and dinucleotide positional identities within sgRNA spacer, and orientation of sgRNA relative to gene. All categorical features were one-hot encoded. Numerical features considered by Schoonenberg et al. include the PROVEAN (Protein Variation Effect Analyzer) deletion score of the targeted amino acids 1 and 2, position in the gene, predicted disorder score of amino acids 1 and 2, GC content of the 20-mer guide, length of the targeted exon, and off-target score of the guide RNA.

Chen and Tran [2019] used two categorical features, i.e. the mismatched bases on both the guide RNA (called "Ref allele") and donor DNA (called "Alt allele"), and one numerical feature, i.e. mismatch position relative to the guide RNA. These features were then converted into binary vectors using one-hot encoding and used as input of several traditional ML models tested by the authors.

Muhammad Rafid et al. [2020] proposed an accurate SVM-based tool using sequence-based features only. The authors experimented with three types of features, including position-independent features, position-specific features, and n-gapped dinucleotides (nGD). The nGD features count the number of times that two given nucleotides appear at a certain distance in a sgRNA sequence. Dhanjal et al. [2020] explored 11 categories of sequence features that possibly govern the specificity of sgRNAs. The main finding of their work consists in the identification of the four most important sequence features, including accessibility of target sequence in the genome, mismatch count between the off-target and target sequences, position-specific occurrence of nucleotides in the spacer and regions flanking it on both sides, and, finally, GC count in the target and off-target sequences.

Hiranniramol et al. [2020] used sequence and structural features of the most and the least potent sgRNAs to train their sgDesigner model. Significance levels for numerical features were computed using Student's t-test, and for binary features using χ^2 test. The authors considered only the high- and low-efficiency sgRNAs groups to emphasize the most predictive features affecting sgRNA efficiency. He et al. [2021] demonstrated that sequence-specific sgRNA activity, frameshift probability, and amino acid features could significantly improve the selection of efficient sgRNAs in protein knockouts. They highlighted the importance of amino acid sensitivity as one of the critical factors that govern the efficiency prediction, in addition to the use of effective sequence models that predict sgRNA activity.

In conclusion, we need to point out that feature comparison between different experiential CRISPR-Cas9 data sets uncover substantial discordance and further research is warranted to identify the most significant generic predictors (i.e. explanatory features) in case of both guide efficiency (i.e. on-target activity) and guide specificity (i.e. off-target effects).

1.5 Traditional machine learning models and their application in CRISPR/Cas9

In this section, we present different conventional machine learning models for on- and off-target prediction found in the genome editing literature related to CRISPR/Cas9. The presentation is organized based on the target categories : (1) off-target prediction only, (2) on-target prediction only, and finally (3) both on- and off-target prediction. Within each category, the works follow chronological order.

First, we discuss papers dealing with off-target activity prediction. Abadi et al. [2017] proposed the CRISPR Target Assessment (CRISTA) algorithm that relies on a random forest ensemble machine learning framework to determine the propensity of a genomic site to be cleaved by a given sgRNA. The authors determined that the system attributes representing spatial structure and rigidity of the entire genomic site as well as those related to the PAM region have the main impact on the prediction capabilities.

Peng et al. [2018] were among the first authors to capitalize on the recent advances in CRISPR/Cas9 data availability. They experimented with two positive sample sets, comprising both on- and off-targets. The first of them contains 215 sequence pairs related to 29 sgRNAs' on-target and off-target editing sites. The second data set includes 527 sequence pairs obtained by using high-throughput sequencing techniques - Digenome-seq [Kim et al., 2015], GUIDE-seq [Tsai et al., 2015], HTGTS [Frock et al., 2015], CIRCLE-seq [Tsai et al., 2017], and multiplex Digenome-seq [Kim et al., 2016]. The authors randomly under-sampled the data to compensate for the class imbalance between on- and off-targets. Then, they trained an ensemble SVM classifier to detect the off-target sites. The authors demonstrated the ability of their model to outperform state-of-the-art predictive methods by aggregating larger numbers of sgRNA-DNA sequence pairs. The work of Peng et al. opened new directions for data aggregation in the field of genome editing.

Chen and Tran [2019] generated the CRISPEY data set consisting of 23,936 samples, each of which contains a 20-nucleotide gRNA sequence and a 100-basepair donor DNA sequence. In this data set, 306 samples are labeled as effect samples and 23,630 samples are labeled as no-effect samples. To predict eventual off-targets, the authors applied three conventional machine learning algorithms including Logistic regression,

SVM [Burges, 1998], and random forest [Breiman, 2001] as well as a simple deep neural network (DNN). The SVM model with 64% recall rate and the Logistic regression with 94% accuracy provided the best prediction results overall.

Zhang et al. [2019b] explored the ensemble learning potential for off-target prediction by synergizing multiple tools. The input of their ensemble learning model included five scores calculated by the following scoring methods - CCTop [Stemmer et al., 2015], MIT Website [Hsu et al., 2013], CFD [Doench et al., 2014], MIT [Haeussler et al., 2016], and Cropit [Singh et al., 2015] as well as evolutionary conservation data and Chromatin state segmentation data. The authors considered an imbalanced data set containing 25,332 putative off-target DNA sequences, with 152 verified positive off-targets, and the other being negative. They compared the five following machine learning algorithms - AdaBoost [Freund et al., 1996], random forest, a multi-layer perceptron [Bishop et al., 1995], SVM, and decision trees [Quinlan, 1987]. In their experiments, Zhang et al. demonstrated that the ensemble-based AdaBoost algorithm was able to outperform the other predictive algorithms in terms of the area under the precision recall curve (AUPRC) and the area under the receiver operating characteristic curve (AUROC) metrics. Lazzarotto et al. [2020] proposed an approach targeting the fast pace of changes in genome editing with a scalable, automatable tagmentation-based model for estimating the genome-wide Cas9 in vitro activity. Their CHANGE-Seq model was designed to better understand the specificity of genome editors. In their experiments, Lazzarotto et al. used the encoded one-dimensional vectors to train a gradient tree boosting model to predict off-target activities. The authors highlighted the importance of the protospacer and the PAM position to ensure accurate off-target predictions. Moreover, they showed that CHANGE-Seq generally outperforms the well-known GUIDE-seq [Tsai et al., 2015] and CIRCLE-seq [Tsai et al., 2017] models.

Second, we present a summary of recent papers addressing the problem of machine learning prediction of on-target activities. Wang et al. [2014] were among the first authors to use SVMs to predict sgRNA efficacy. The authors used log2 fold change of sgRNAs targeting ribosomal protein genes as their efficacy indicator. Precisely, the log2 fold change was applied to build a binary classification, where ribosomal protein gene-targeting sgRNAs were designated either as weak or as strong. Doench et al. [2014] trained a logistic regression classifier to differentiate the highest activity quintile of sgRNAs from their lowest activity quintile. The authors used sequence features from nine mouse and human genes with cross-validation to ensure the generalisation across genes. Xu et al. [2015] applied a regularized regression technique that linearly combines the penalties of the Lasso and Ridge methods, and Elastic-Net, to predict sgRNA efficiency in

CRISPR/Cas9 knockout experiments. The authors demonstrated that Elastic-Net outperforms existing models on different independent data sets. Fusi et al. [2015] investigated how to achieve the best optimal predicative performance in CRISPR/Cas9 gene editing. The authors relied on two different primary data sets composed of mouse and human genes. They built and trained five traditional machine learning classifiers to predict the knockout efficacy, and observed that the gradient-boosted regression trees yielded the best performance overall.

Rahman and Rahman [2017] introduced the CRISPRpred model aiming at providing accurate in silico predictions of sgRNA on-target activity. CRISPRpred is capable to extract relevant features in order to use them in an SVM-based machine learning framework. The work of Rahman et al. emphasizes the importance of feature engineering in boosting the predictive performance of sgRNA on-target prediction models.

Furthermore, Muhammad Rafid et al. [2020] demonstrated the importance of feature engineering and data pre-processing to ensure effective sgRNA on-target activity prediction. The authors proposed a novel SVM-based machine learning tool, named CRISPRpred(SEQ), which is capable to challenge the effective DeepCRISPR model [Chuai et al., 2018] based on deep learning. The authors demonstrated that thanks to designing better explanatory features, CRISPRpred(SEQ), that used a simpler model architecture, was able to outperform DeepCRISPR in 3 out 4 cell lines.

Wang et al. [2020a] proposed a novel methodology targeting cross-species generalization of on-target activities. The authors developed the GNL-Scorer software computing two cross-species generalization scores, GNL and GNL-Human. GNL-Scorer also combines different data sets, features, and models for sgRNA activity prediction, agnostic to the species. The authors claimed that GNL-Scorer facilitates the current in silico design of sgRNAs. Konstantakos et al. [2022a] introduced a new interpretable gRNA efficiency prediction model and the related web tool, called CRISPRedict, including various regression and classification models for gRNA scoring. This web tool offers accurate efficiency predictions under different experimental conditions (e.g. U6/T7 transcription) and the related visualizations facilitating the explaination of the obtained results.

As explanatory features, the authors considered overall and position-specific nucleotide composition, as well as variables reflecting the structural properties of gRNAs. They conducted a multi-step feature selection strategy to infer a minimal relevant feature subset. Then, they used a binomial and a linear regression models to predict the percentage of successful edits for the U6 and T7 variants, and trained two logistic regression

models by labeling the top 20% and the bottom 20% of gRNAs as efficient and inefficient, respectively. Konstantakos et al. evaluated the performance of CRISPRedict, comparing it to state-of-the-art gRNAs design tools, including some deep learning models, and concluded that despite its simplicity CRISPRedict provides interpretable efficiency predictions with comparable performance. Zarate et al. [2022] developed a new machine learning model, called BoostMEC (Boosting Model for Efficient CRISPR), to predict CRISPR-Cas9 editing efficiency. BoostMEC is based on a gradient boosting technique and LightGBM (Light Gradient-Boosting Machine). The LightGBM hyperparameters were tuned using tenfold cross-validation and Bayesian hyperparameter optimization. The authors compared BoostMec to 10 state-of-the-art on-target prediction models on 13 benchmark data sets. They concluded that BoostMEC, which relies on direct and derived sgRNA features and traditional machine learning, has an advantage over state-of-the-art prediction models based on deep learning because of its ability to produce more interpretable feature insights and predictions.

Finally, we discuss papers addressing the problem of both on- and off-target activity prediction by means of conventional machine learning models. Doench et al. [2016] designed and tested their novel sgRNA design rules to create human and mouse genome-wide libraries and carry out the corresponding positive and negative selection screens. The authors proposed a new metric to predict off-target sites, and designed optimized sgRNA libraries with maximized on-target activity and minimized off-target effects. In order to identify an optimal classifier, they compared the performance of eight conventional machine learning models, including linear regression, L1-regularized linear regression, L2-regularized linear regression, a hybrid SVM plus logistic regression, random forest, gradient-boosted regression trees, L1 logistic regression (a classifier), and SVM (with linear kernel with default L2 regularization). Liu et al. [2020c] proposed an open-source software, called SeqCor, which relies on the application of the random forest algorithm to extract sequence features that influence gRNA knockout efficiency as well as gRNA off-target activity at specific sites. The aim of their work was to facilitate the extraction of the sequence features and to minimize possible bias effects that may be present in a library used in CRISPR/Cas9-based screening.

Although the use of traditional machine learning algorithms, whose main advantages are their relative simplicity and fast training, led to some impressive on- and off-target activity prediction results, recent studies conducted using deep learning methods (see the next section) often demonstrated a superior performance. We are convinced that, in general, deep learning models are better suited for both on- and off-target activity prediction than conventional machine learning models, since modern CRISPR/Cas9 data sets contain hundreds of thousands, and sometimes millions, of samples, and state-of-the-art deep learning

algorithms can be effectively used on such huge volumes of data encompassing complex non-linear patterns. However, for benchmark purpose, the results obtained using state-of-the-art deep learning algorithms should be always compared with those provided by some well-performing traditional machine learning methods such as SVM, random forest, and XGBoost, as well as their ensemble frameworks, which are capable of increasing the accuracy of individual ML methods. Table 1.2 reports the main traditional machine learning classifiers and regressors used for on- and off-target prediction in CRISPR/Cas9.

Table 1.2 – Summary of studies applying traditional machine learning for on/off-target prediction in CRISPR/Cas9.

Study	Target prediction	ML Models	Encoding	Data	Link for data/software	Metric/Results
Wang et al. [2014]	On-target	SVM	One-hot + GC content	A library of 73,000 sgRNAs in human cell lines	`https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3972032`	log2 fold change estimations
Doench et al. [2014]	On-target	SVM, Logistic regression	One-hot + GC counts + position-specific features	1,831 gRNAs targeting three human and six mouse genes	`http://www.broadinstitute.org/rnai/public/analysis-tools/sgrna-design`	AUROC : 0.8
Xu et al. [2015]	On-target	Logistic regression	One-hot + GC counts + position-specific features	Wang [Wang et al., 2014], Koike-Yusa [Koike-Yusa et al., 2014], Shalem [Shalem et al., 2014], Zhou [Zhou et al., 2014], Gilbert [Gilbert et al., 2014], Konermann [Konermann et al., 2015]	`http://crispr.dfci.harvard.edu/SSC`	AUROC : 0.73

Fusi et al. [2015]	On-target	SVM, L1/L2 regression, RF regression, SVM+logistic regression, L1 logistic regression, linear regression, GBRT	One-hot + GC counts + position-specific features	Wang ribosomal [Wang et al., 2014], Koike-Yusa [Koike-Yusa et al., 2014], Doench V1 [Doench et al., 2014]	`http://research.microsoft.com/en-us/projects/azimuth`	Spearman : 0.52, AUROC : 0.75
Doench et al. [2016]	Off-target and on-target	Boosted RT, L1/L2 regression, SVM+logistic regression, RF, linear regression	One-hot + GC counts + position-specific and position-independent features + thermodynamic features	2,549 unique guides to generate Doench V2 dataset	`http://www.broadinstitute.org/rnai/public/analysistools/sgrna-design`, `http://research.microsoft.com/en-us/projects/azimuth`	Spearman : 0.54 (on-target), AUROC : 0.8 (off-target)
Rahman and Rahman [2017]	On-target	CRISPRpred (SVM, RF, linear regression)	One-hot + position-specific and position-independent features + structural/thermodynamic features	Doench V1 [Doench et al., 2014]	`https://github.com/khaled-buet/CRISPRpred`	AUROC : 0.85, AUPRC : 0.56, MCC : 0.4
Abadi et al. [2017]	Off-target	CRISTA (CRISPR Target Assessment using RF regression)	GC content + sgRNA secondary structure features	GUIDE-Seq [Tsai et al., 2015, Kleinstiver et al., 2016], HTGTS [Frock et al., 2015], BLESS [Ran et al., 2015, Slaymaker et al., 2016]	`http://crista.tau.ac.il`	Spearman : 0.81, AUROC : 0.96, AUPRC : 0.96, $R^2 = 0.8$
Peng et al. [2018]	Off-target	Ensemble SVM	One-hot + GC content + position-specific features	CRISPOR [Haeussler et al., 2016]	`https://github.com/penn-hui/OfftargetPredict`, Cas-OFFinder [Bae et al., 2014]	AUROC : 0.99, AUPRC : 0.45

Schoonenberg et al. [2018]	On-target	CRISPRO (GBDT, Ridge, RF, Lasso, SVM)	One-hot + GC content + position-specific features	Doench V2 [Doench et al., 2016], Munoz [Munoz et al., 2016], Donovan [Donovan et al., 2017], Brenan [Brenan et al., 2016]	`http://gitlab.com/bauerlab/crispro`	Spearman : 0.57
Listgarten et al. [2018]	Off-target	Elevation (Boosted regression trees, L1 regression, Naïve Bayes)	One-hot	GUIDE-Seq [Tsai et al., 2015], Doench V2 [Doench et al., 2016], CRISPOR [Haeussler et al., 2016]	`http://research.microsoft.com/en-us/projects/crispr`	AUROC : 0.98
Chen and Tran [2019]	Off-target	Logistic regression, SVM, RF, NN	One-hot + Ref allele + Alt allele + mismatch position + CRISPEY guide features	Unpublished data of Sharon et al. were used to generate CRISPEY (Cas9-Retron precISe Parallel Editing via homologY) data consisted of 23,936 samples (18,717 training set and 4,680 testing set	`https://github.com/elizapandabella/CRISPEY_ML_Public`	Accuracy : 94%
Zhang et al. [2019b]	Off-target	Ensemble learning framework of scoring features using AdaBoost	One-hot + GC counts + position-specific features	GUIDE-Seq [Tsai et al., 2015], CRISPOR [Haeussler et al., 2016]	`https://github.com/Alexzsx/CRISPR`	AUROC : 0.938, AUPRC : 0.299
Lazzarotto et al. [2020]	Off-target	CHANGE-seq (GTB)	One-hot + sequence features	High-throughput sequencing data generated from CHANGE-seq, GUIDE-Seq [Tsai et al., 2015], CIRCLE-seq [Tsai et al., 2017]	`https://github.com/tsailabSJ/changeseq`, Cas-OFFinder [Bae et al., 2014], DeepTools [Ramírez et al., 2014]	AUROC : 0.995, AUPRC : 0.881
Muhammad Rafid et al. [2020]	On-target	CRISPRpred(SEQ) (SVM)	Position-independent + position-specific + n-gapped di-nucleotide features	CRISPOR [Haeussler et al., 2016], DeepHF [Wang et al., 2019a]	`https://github.com/Rafid013/CRISPRpredSEQ`	Spearman : 0.829, AUROC : 0.893

He et al. [2021]	On-target	GuidePro (two-layer ensemble, SVM, RF)	Sequence-specific features	Doench V1 [Doench et al., 2014], Doench V2 [Doench et al., 2016], CRISPOR [Haeussler et al., 2016], Munoz [Munoz et al., 2016], Schoonenberg [Schoonenberg et al., 2018], Aguirre [Aguirre et al., 2016], Evers [Evers et al., 2016], Bertomeu [Bertomeu et al., 2018]	`https://bioinformatics.mdanderson.org/apps/GuidePro`, inDelphi [Shen et al., 2018], Lindel [Chen et al., 2019], FORECasT [Allen et al., 2019]	Spearman : 0.523
Wang et al. [2020a]	On-target	GNL-Scorer Eight models ; (GBRT, DT, linear regression, L2/L1 regression, BRR, RF, NN)	One-hot + GC count + position-independent + position-dependent features + thermodynamic features	10 public datasets : Doench V1 [Doench et al., 2014], Doench V2 [Doench et al., 2016], HCT116 [Hart et al., 2015], Hela [Hart et al., 2015], Zebrafish MM [Moreno-Mateos et al., 2015], Zebrafish VZ [Varshney et al., 2015], Zebrafish GZ [Gagnon et al., 2014], Drosophila [Ren et al., 2014], HEK293T [Chari et al., 2015], Ciona [Gandhi et al., 2016]	`https://github.com/TerminatorJ/GNL_Scorer`	Spearman : 0.502
Dhanjal et al. [2020]	Off-target	L1/L2 logistic regression, RF, xgboost	One-hot + GC content + position-specific features	CIRCLE-seq [Tsai et al., 2017], CRISPcut [Dhanjal et al., 2019]	`http://web.iitd.ac.in/crispcut/off-targets`	Accuracy : 91.49%, AUROC : 0.97
Liu et al. [2020c]	Off-target and on-target	SeqCor (open-source random forest software)	A general-purpose hash function	DeepCRISPR [Chuai et al., 2018]	`https://github.com/wangyi-fudan/SeqCor`	Spearman : 0.4 (off-target), 0.369 (on-target)

Hiranniramol et al. [2020]	On-target	sgDesigner (stacking SVM and XGBoost with logistic regression)	GC content + structural features	Wang [Wang et al., 2014], Koike-Yusa [Koike-Yusa et al., 2014], Doench V1 [Doench et al., 2014], Chari [Chari et al., 2015], Shalem [Shalem et al., 2014]	`https://github.com/wang-lab/sgDesigner`, RNAfold [Hofacker, 2003]	Spearman : 0.75, AUROC : 0.934, Accuracy : 86.3%
Konstantakos et al. [2022a]	On-target	CRISPRedict (linear regression, binomial regression, logistic regression)	Overall and position-specific nucleotide composition + structural properties of sRNAs	Koike-Yusa [Koike-Yusa et al., 2014], DeepSpCas9 [Kim et al., 2019], CRISPOR [Haeussler et al., 2016], Labuhn [Chuai et al., 2018, Labuhn et al., 2018]	`https://github.com/VKonstantakos/CRISPRedict`, `http://www.crispredict.org`	Spearman : 0.380 (U6 data set), 0.355 (T7 data set), nDCG : 0.805 (U6 data set), 0.554 (T7 data set)
Zarate et al. [2022]	On-target	BoostMEC (Boosting Model for Efficient CRISPR)	GC content + position-specific features + thermodynamic features	DeepSpCas9 [Kim et al., 2019], CRISPRon [Xiang et al., 2021]	`https://github.com/oazarate/BoostMEC`	Spearman : 0.78

RF : Random forest, GBRT : Gradient-boosted regression tree, SVM : Support Vector machine, MCC : Matthews Correlation Coefficient, GB : Gradient boosting, NN : neural networks, KNN : k-nearest neighbors, GTB : Gradient Tree Boosting, DT : Decision Tree, BRR : Bayesian Ridge regression, nDCG : normalized discounted cumulative gain.

1.6 A brief review of deep neural networks

Deep learning applications across all research fields have recently gained popularity due to easier access to data, boosted computing power, and recent theoretical progress in supervised learning. Deep neural networks are at the core of deep learning. They are capable of learning complex patterns from the data using multiple layers of interconnected neurons. Nonetheless, their training and optimization are still very challenging problems. This section is divided into two parts. First, we present the main properties of existing deep learning network architectures. Second, we discuss their applications in CRISPR/Cas9.

In this section, we describe succinctly the three main types of deep neural networks used for on- and off-target activity prediction in CRISPR/Cas9. They are Feedforward Neural Networks (FNNs), Convolutional

Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). Figure 3.3 illustrates three typical deep learning network architectures used in CRISPR/Cas9. Finally, we present some popular activation functions of neurons used in deep learning models.

The FNNs is one of the most popular deep neural network architectures [Heaton, 2018]. In a standard FNN architecture, the information always moves forward between different layers of interconnected neurons. The two main categories of FNN are as follows : a single-layer FNN and a multi-layer FNN. In a single-layer FNN, the input layer, i.e. the first layer of neurons receiving the data as input, is directly fully connected to the output layer. The output layer is the last layer outputting the predictions. In the multi-layer FNN, the input layer and the output layer are fully connected to hidden layers. Thus, a multi-layer FNN has at least 3 layers of neurons. Figure 3.3 presents a typical multi-layer FNN architecture used for off-target prediction in CRISPR/Cas9 (for more details, see [Charlier et al., 2021]).

The CNNs introduced by LeCun et al. [1998] rely, as FNNs, on fully connected layers, but also include convolutional layers and pooling layers. A convolutional layer consists of a collection of convolutional filters used to extract spatial features from the input image. Within the convolutional layer, different filters, or kernels, can be applied to process the data and generate feature maps. These feature maps help the neural network to better regress or classify the input data. Following the convolutional layers, hidden fully connected layers are often used to further improve the predictive performance of a CNN. An example of a CNN architecture used for off-target prediction in CRISPR/Cas9 is presented in Figure 3.3 (for more details, see [Charlier et al., 2021]).

The RNNs is another type of neural networks that has found applications for on- and off-target prediction in CRISPR/Cas9. Contrary to the FNNs and CNNs, the information in RNNs does not always move forward ; it can also move backward. The aim of RNNs is to replicate a memory process through a recurrent learning mechanism. The RNNs aggregate information of the past inputs and that of the current input in order to produce the current output. An example of an RNN architecture used for off-target prediction in CRISPR/Cas9 is presented in Figure 3.3 (see also [Charlier et al., 2021]). Two popular types of RNNs are the Long Short-Term Memory (LSTM) models, which are designed to learn order dependence in sequence prediction problems [Hochreiter and Schmidhuber, 1997, Gers et al., 2000] and the Gated Recurrent Unit (GRU) models, which use a similar to LSTMs prediction mechanism, but require less memory and are usually faster than LSTMs as they have no output gate [Chung et al., 2014]. Both the LSTM and GRU model architectures have the

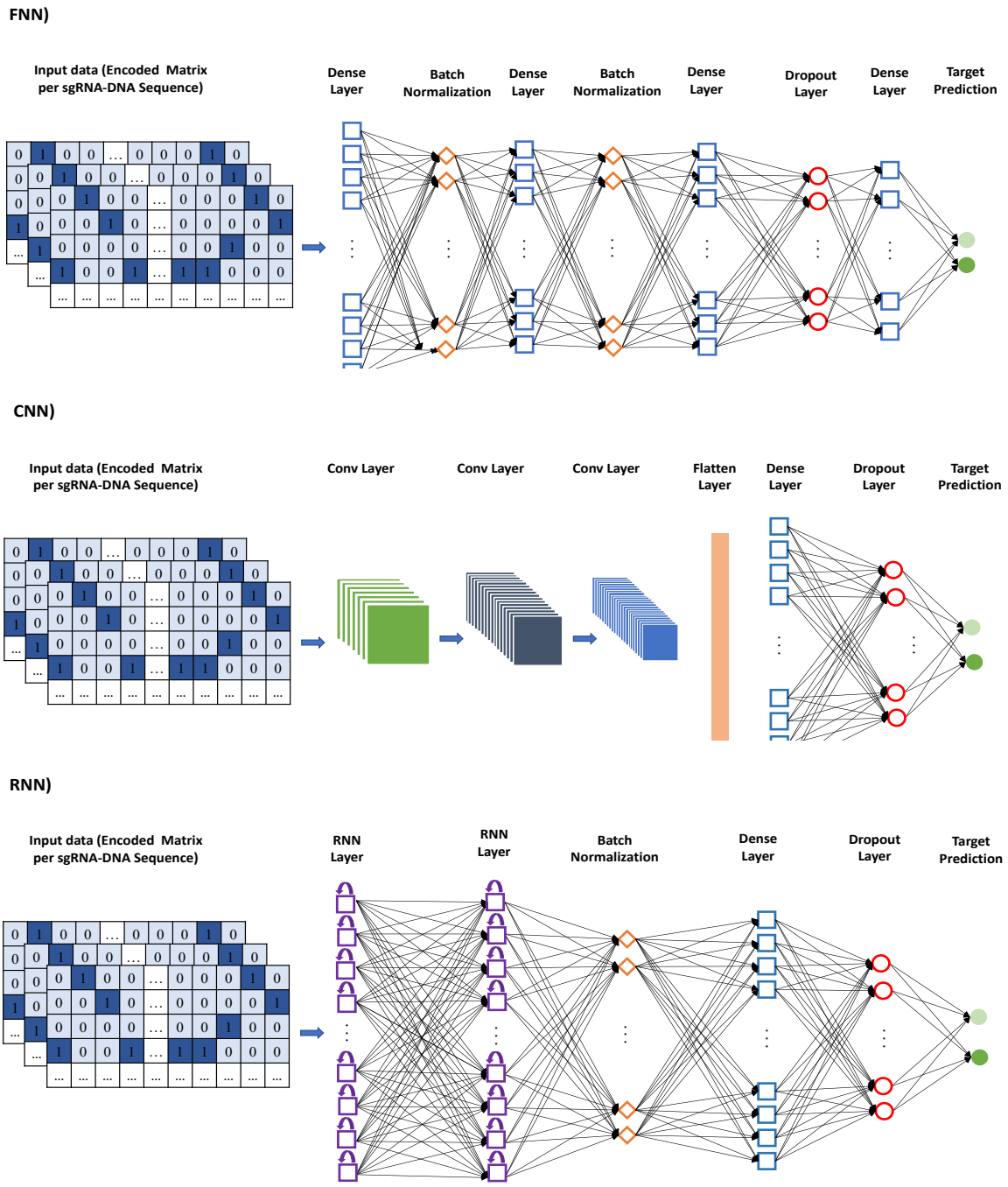

Figure 1.4 – Some standard architectures of Feedforward Neural Networks (FNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) used to on- and off-targets in CRISPR/Cas9. For each network, the encoded matrix containing the sgRNA-DNA sequence pair information is used as input (for more details, see Charlier et al. [2021]).

ability to forget the information by using a forget gate. One of the advantages of the LSTMs is that they are able to overcome the vanishing gradient problem that occurs while training networks with backpropagation and gradient-based learning methods, preventing undesirable weight updates in the RNN. For the input sequence $< x_1, x_2, \ldots, x_T >$, the key mathematical equations of the forward pass of an LSTM unit are as follows [Heaton, 2018] :

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c), \\
h_t &= o_t \circ \sigma_h(c_t),
\end{aligned}
\tag{1.1}
$$

where $\circ$ denotes the Hadamard product, x_t is the unit's input at time t, h_t is the corresponding unit's output, c_t is the hidden unit's memory, and i_t, f_t, and o_t are, respectively, the activation vectors of the input gate, of the forget gate, and of the output gate. The variables W, U, and b are, respectively, the weight matrices and the bias parameter, and σ_g denotes a sigmoid function. Finally, both σ_c and σ_h denote hyperbolic tangent functions.

For all deep learning network architectures presented here, the neurons rely on an activation function that is used to determine whether a given neuron should be activated or not. Thus, the activation function being used determines whether the neuron's contribution to the network is important or not in the prediction process. An activation function allows the model to transform the weighted sum of the neuron's input signals to an output signal. Table 1.3 summarises the most common activation functions used in artificial neural networks with their respective formulas.

Deep neural networks, by their nature, have a large number of parameters to fine-tune during their training. Special attention must be given to the problem of overfitting in the model's training. Overfitting occurs when the model learns patterns only present in the training set and cannot generalize its predictive performance on the test set. Different techniques exist to limit the overfitting while training deep neural networks. The most important of them are early-stopping, network-reduction, regularization, and dropout [Heaton, 2018, Ying, 2019]. We invite the reader to consult the aforementioned literature for more details.

Table 1.3 – Activation functions commonly used in artificial neural networks.

Name of the function	Formula
ReLU	$\sigma(x) = \max(0, x)$
Leaky ReLU	$\sigma(x) = \max(\alpha x, x)$
Randomized leaky ReLU (RReLU)	$\sigma(x) = \max(0, x) + \alpha \times \min(0, x)$
Parametric leaky ReLU (PReLU)	$\sigma(x) = \max(0, x) - \alpha \times \max(0, -x)$
Scaled exponential Linear Units (SeLU)	$\sigma(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha\, e^x - \alpha, & x \leq 0 \end{cases}$
Logistic (Sigmoid)	$\sigma(x) = \dfrac{1}{1 + e^{-x}}$
Hyperbolic Tangent (Tanh)	$\sigma(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$
Softmax	$\sigma(x) = \dfrac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$, where K the number of classes

1.7 Deep learning models and their applications in CRISPR/Cas9

This section includes four thematic subsections. First, we discuss the studies emphasizing the use of novel sequence encoding strategies along with deep learning models. Second, we present works using different feature engineering approaches. Third, we highlight the papers applying class rebalancing techniques prior to carrying out deep learning algorithms. Forth, we describe some recent works relying on the use of attention mechanism. Within each category, the works follow the off-target, on-target, and both on- and off-target prediction order.

1.7.1 Models relying on novel sequence encoding strategies

As regards the off-target prediction, Lin and Wong [2018] proposed a novel sequence encoding scheme based on the use of deep neural networks. They investigated several network architectures, including CNNs and FNNs. In their experiments, the authors used publicly available CRISPOR (18,236 samples) and GUIDE-Seq (430 samples) data sets. The transfer learning strategy was applied to obtain the off-target predictions for a much smaller GUIDE-Seq data set as the model trained on the CRISPOR data set was used to predict the GUIDE-Seq off-targets. Despite the fact that the encoding strategy proposed by Lin and Wong [2018] could lead to some information loss as the 4×23 matrix considered by the authors cannot represent the unique

match-mismatch information for a given sgRNA-DNA sequence pair (it was then corrected by Lin et al. [2020], who considered a loss-free model based on a 7×23 matrix encoding), very encouraging AUCROC results were obtained by these authors (i.e. an area under the curve values up to 0.972 were generated). The main conclusion of this work was that CNN and FNN deep learning networks steadily outperform state-of-the-art off-target scoring prediction methods (i.e. CFD, MIT, CROP-IT, and CCTOP) as well as some traditional machine learning classifiers (i.e. random forest, gradient boosting trees, and logistic regression). Afterwards, Lin et al. [2020] proposed an original one-hot sequence encoding scheme (see Fig. 1.3) and an effective CRISPR-Net model, where a Long-term Recurrent Convolutional Neural Network (LRCN) was playing the role of a feature extractor. The authors noticed that convolutional layers of LRCN were able to discover useful features from sequences directly and independently, preventing possible biases introduced by hand-crafted sequence features. The convolutional kernels within the LRCN were replaced by an inception module and a bidirectional LSTM was used for scoring the off-target activity of each potential sgRNA-target. Charlier et al. [2021] proposed a novel sgRNA-DNA sequence encoding technique, which was applied in a deep learning off-target prediction framework. The loss-free encoding model introduced by the authors relayed on 8×23 matrix representations. Charlier et al. compared the prediction performance of different FNN, CNN, and RNN network architectures (see Fig. 3.3) as well as of several machine learning classifiers (i.e. random forest, naive Bayes, and logistic regression). The predictions were performed on two well-known gene editing data sets, CRISPOR and GUIDE-Seq, which were previously considered by Lin and Wong [2018]. The transfer learning approach was used as well to predict the off-targets on the much smaller GUIDE-Seq data set. The proposed prediction framework led to more accurate off-target prediction results, compared to those obtained by Lin and Wong [2018], yielding an improvement of the AUCROC metric up to 35%.

Regarding the on-target prediction, Xue et al. [2018] introduced DeepCas9, an effective deep-learning framework based on CNNs. The authors proposed a network architecture to automatically learn the sequence determinants, conducting their experiments with 10 CRISPR/Cas9 data sets of different sizes. Xue et al. demonstrated that DeepCas9 is capable of outperforming some traditional machine learning methods such as random forest and logistic regression, in terms of on-target activity prediction. In their timely review, Konstantakos et al. [2022b] evaluated eleven tools for gRNA efficiency prediction, which were applied to analyze six benchmark data sets. The evaluated tools included 10 deep learning models and one conventional machine learning model, called Azimuth 2.0 [Doench et al., 2016] and based on gradient-boosted regression trees. Most of the considered deep learning models represented the target sequence by means of different one-hot encoding strategies, while a few of them captured the epigenetic features as well. Comparison of

gRNA efficiency prediction was carried out using the Spearman correlation. The authors observed that the correlation between the predicted and the true efficiency varied a lot, depending on the data set being analyzed and the model being used. Overall, the DeepHF [Wang et al., 2019a] and DeepSpCas9 [Kim et al., 2019] models were consistently among the top performers along with the Average prediction strategy that consisted of the mean of the DeepCRISPR [Chuai et al., 2018], DeepCas9 [Xue et al., 2018], DeepSpCas9 [Kim et al., 2019], and Azimuth 2.0 [Doench et al., 2016] predictions. Konstantakos et al. pointed out that simpler machine learning tools can sometimes outperform their much more sophisticated deep learning counterparts, and that large data sets that benefit from unbiased experimental measurements play a crucial role in training a generalizable model, such as DeepHF or DeepSpCas9.

Concerning the prediction of both on- and off-targets, we need to mention the work of Chuai et al. [2018], who were among the first to tackle the problem of sgRNA-DNA sequence encoding adapted to the input of deep learning models. The authors implemented a deep learning framework, called DeepCRISPR, predicting simultaneously the sgRNA on-target knockout efficacy and the off-target cleavage. An original one-hot encoding strategy used by the authors consisted of a four-channel-based sgRNA-DNA sequence encoding and epigenetic feature encoding in which each feature was considered as an independent channel. Chuai et al. introduced an unsupervised representation learning strategy to train a Deep Convolutional Denoising Neural Network (DCDNN) auto-encoder to learn the underlying representation of sgRNAs. The unsupervised deep representation learning approach was then used to transfer the encoded data to a hybrid deep neural network. The proposed network included a softmax activation function and an identity function for the classification and regression tasks, respectively. DeepCRISPR provided good prediction results in terms of both AUPRC and AUCROC, compared to the CFD prediction method [Doench et al., 2014].

Most of the aforementioned studies demonstrated impressive prediction results using novel sgRNA-DNA one-hot sequence encoding schemes combined with deep learning models. We think, however, that further prediction improvements will not anymore result from the use of sophisticated data encoding schemes, nor from the models complexity, but rather from an effective use of additional biological or physical features as well as from the application of the feature engineering and class rebalancing techniques.

1.7.2 Models relying on feature engineering

In addition to conventional nucleotide sequence data, some plausible biological and physical features, such as gene melting temperature, molecular weight, or microhomology features, can be also used as input of

CRISPR/Cas9 predictive models. Moreover, some new informative features can be created, or reengineered, from the existing ones. Feature engineering has proven to be an important milestone when developing and optimizing predictive models [Heaton, 2018].

As regards the off-target prediction, Liu et al. [2020b] introduced a deep learning architecture, called Cnn-Crispr, to predict the off-target propensity of sgRNAs at specific DNA fragments. The approach proposed by these researchers relies on the GloVe embedding model [Pennington et al., 2014] to extract the global statistical information from genes. The constructed word vector matrix was then embedded into the considered deep learning model including a bidirectional LSTM and a CNN. The authors demonstrated that the proposed approach outperforms state-of-the-art classification and regression algorithms. Störtz et al. [2021] introduced the piCRISPR deep learning model intended for off-target prediction using physically informed features. The authors designed four different feature encoding schemes to incorporate the following physically informed features : the target-guide encoding, the target-mismatch encoding, the target-mismatch-type encoding, and the target-OR-guide encoding. Moreover, they assessed the feature importance using the model-agnostic SHAP (SHapley Additive exPlanations) technique [Lundberg et al., 2018]. In their experiments conducted with the crisprSQL data set, Stortz et al. demonstrated the importance of both the sequence context and the chromatin accessibility for effective cleavage prediction. Niu et al. [2021] proposed the R-CRISPR deep learning model that encodes sgRNA target sequences into a binary matrix and then uses a CNN model as a feature extractor. Precisely, the authors applied a Rep-VGG inference time body composed of a stack of 3×3 convolutions and ReLUs [Ding et al., 2021] in the convolutional layers to extract relevant features. The CNN output was then passed to a bi-directional recurrent layers using an LSTM to get accurate off-target predictions. Fu et al. [2022] introduced the MOFF off-target predictor based on the MOFF-target score function that is the sum of the multiplication of individual mismatch effect (IME), the combinatorial effect (CE), and the guide-intrinsic mismatch tolerance effect (GMTE), where GMTE is estimated from a dinucleotide CNN regression model. Two different encoding strategies : (1) mononucleotide encoding and (2) dinucleotide encoding were used to vectorize the input sequences. The tests conducted on a high-throughput allele-editing screen of 18 cancer hotspot mutations confirmed that MOFF significantly improves the selectivity and expands the application domain of Cas9-based allele-specific editing.

Concerning the on-target prediction, Shrawgi and Sisodia [2019] introduced DeepSgRNA, a deep learning architecture relying on CNN, to identify and predict RNA guides. The aim of their approach was to eliminate the need in manual feature construction, which improved the scalability of the approach. DeepSgRNA

relies on hierarchical feature generation abilities of CNNs. In their experiments with the GenomeCRISPR data, the authors proved that DeepSgRNA was able to achieve state-of-the-art sgRNA prediction efficiency. Furthermore, we need to mention the pioneering work of Wang et al. [2019a], who compared several deep learning and conventional machine learning models to provide gRNA activity predictions for SpCas9 (eSpCas9(1.1)), Cas9-High Fidelity (SpCas9-HF1), and wild-type SpCas9 (WT-SpCas9) data. Feature engineering was performed using the effective Tree SHAP technique [Lundberg et al., 2018] that combines the SHAP values [Lundberg and Lee, 2017] with the XGBoost algorithm. The authors built two RNN and one CNN deep learning models, and trained them along with a linear regression, a L2-regularized linear regression, an XGBoost, and a multilayer perceptron. In their experiments, Wang et al. demonstrated that the RNN that received as input the sequence data with added biological features was able to outperform the other competing models. Their second best-performing model was the RNN that received as input sequence data only. Furthermore, Wang et al. developed a DeepHF website to ease the access to WT-SpCas9, eSpCas9(1.1), and SpCas9-HF1 indel data. The authors also applied SHAP with XGBoost and RNNs to assess the feature importance of the sequence input. Moreover, they used the Deep SHAP algorithm [Lundberg and Lee, 2017] to estimate the position-dependent gRNA nucleotide contribution to deep learning predictions. Based on the results of Wang et al., we can conclude that additional biological information and plausible reengineered features increase the predictive performance of deep learning models, leaving opportunities for future model enhancement. Elkayam and Orenstein [Elkayam and Orenstein, 2022] proposed the DeepCRISTL on-target prediction model, which can be considered as an improvement of DeepHF [Wang et al., 2019a]. It is based on the use of the BLSTM (Bidirectional Long Short-Term Memory) and transfer learning techniques. To improve the prediction performance of DeepHF, the authors also considered some plausible biological features of the DeepHF data set. Elkayam and Orenstein used random hyperparameter search to carry out hyperparameter optimization of their DeepCRISTL-pre-train model, which allowed them to outperform the DeepHF model proposed by Wang et al. [2019a].

Finally, the CRISPRon and CRISPRoff deep learning models, based on CNN and Gradient boosting regression trees, and the related interactive webserver were developed by Xiang et al. [2021]. For their prediction, the authors used different position-specific sequence, position-independent, and thermodynamic features. They established that the gRNA-DNA binding energy is a major contributor in predicting the on-target activity of gRNAs. Using the CRISPRon model, the user can compute the on-target efficiency of all possible gRNAs with NGG PAM sequences, for genes, genomic regions, or custom sequences. The CRISPRoff model allows the user to compute the specificity of gRNAs with NGG PAM sequences. Moreover, by means of CRISPRoff,

one can compute the relative likelihood of cleavage by CRISPR-Cas9 on off-target sites compared to the likelihood of cleavage at the on-target sites.

Feature engineering has proven to be essential in many research fields involving machine learning predictions. We think that this aspect has not yet been fully explored in CRISPR/Cas9 and that future investigations should contribute to higher predictive performance of both traditional machine learning and deep learning models applied to predict on- and off-target activities.

1.7.3 Models relying on class rebalancing techniques

Training deep learning models with real-world CRISPR-Cas9 data is challenging because of a large natural imbalance existing between positive and negative samples. This leads to an imbalanced data classification problem with a much larger majority class and a much smaller minority class. Thus, the predictive models observe and learn more from samples of the majority class and, as a consequence, can fail to identify accurately samples from the minority class. This can negatively impact their overall predictive performance. In CRISPR/Cas9, the problem of data imbalance mainly applies to the task of off-target prediction.

In this context, we need to mention the work of Zhang et al. [2020c] and their DL-CRISPR deep learning model for off-target activity prediction, with data augmentation as a solution for the class imbalance problem. The authors first gathered data from two source types (i.e. from in vitro and cell-based experiments) to increase the size of the positive class samples (i.e. off-targets), and thus the model's competency. Precisely, Zhang et al. proposed to increase synthetically the number of positive samples by rotating the sgRNA-DNA encoded images by 90 degrees, 180 degrees, and 270 degrees, respectively. Hence, the number of positive samples in the data set was quadrupled. The data was then passed to a four-layer CNN to perform the off-target prediction. The main finding of their work was that data augmentation was a critical step for improving the predictive performance of DL-CRISPR. Class rebalancing and data augmentation are still fast evolving domains. Alleviating data imbalance should boost the predictive performance of off-target prediction models as class imbalance remains one of the intrinsic properties of CRISPR-Cas9 data.

1.7.4 Models relying on attention mechanism

Recent progress in the development of deep learning models using attention mechanism [Vaswani et al., 2017] has triggered interest in many research fields, including CRISPR-Cas9, where it has already provided

some promising off-target specificity and on-target efficiency prediction results.

Recently, Zhang and Jiang [2022] have implemented the CRISPR-IP off-target prediction model that is based on a CNN, a BLSTM, and an attention layer learning the sgRNA-DNA sequence pair features. CRISPR-IP combines the four following types of network layers : (i) the convolutional layer to learn local features, (ii) the recurrent layer to learn the context features of the sequence, (iii) the attention layer to learn global features from the attention score, and (iv) the dense layer to map the features to the sample label space. The authors also used a new type of encoding scheme to overcome the problem of information loss in the sequence encoding. Xiao et al. [2021] designed AttCRISPR, a deep learning framework based on the attention mechanism for predicting on-target activity. The proposed approach relies on the two attention modules, one spatial and one temporal, facilitating the model's interpretability. AttCRISPR relies on an ensemble learning strategy stacking encoding-based and embedding-based methods to improve its predictive performance. Liu et al. [2019] analyzed two transformer-based neural networks, AttnToMismatch_CNN and AttnToCrispr_CNN, using cell-specific information of genes. Both models are similar, except that AttnToCrispr_CNN employs a linear regression at the final layer. AttnToMismatch_CNN and AttnToCrispr_CNN demonstrated competitive performance for both off-target sgRNA specificity prediction and on-target efficiency prediction. Furthermore, Liu et al. introduced a third model, called seqCrispr, which harbours an LSTM component and a CNN component in parallel to provide accurate on-target efficiency predictions. Furthermore, Zhang et al. [2021] proposed two novel interpretable attention-based CNN models, called CRISPR-ONT and CRISPR-OFFT, designed for predicting sgRNA on- and off-target activities, respectively. Their methodology emphasizes the importance of the feature explainability for obtaining accurate on- and off-target predictions. Interpretable attention-based CNNs were used to highlight how RNA-guide Cas9 nucleases could be used to investigate mammalian genomes.

We are confident that future promising methods involving attention mechanism and deep learning should soon emerge in the field. Table 2.4 summarizes the most important recent deep learning models used for on- and off-target target prediction in CRISPR/Cas9.

Table 1.4 – Summary of deep learning models for on- and off-target prediction in CRISPR/Cas9.

Study	Target prediction	ML model(s)	Encoding	Data	Link for data/software	Prediction metric/results

Chuai et al. [2018]	Off- and on-target	DeepCRISPR (DCDNN)	One-hot	0.68 billion sgRNA sequences from 13 human cell lines were considered to generate DeepCRISPR data	`http://www.deep/crispr.net`, `https://github.com/bm2-lab/DeepCRISPR`	Spearman : 0.246, AUROC : 0.804, AUPRC : 0.303
Lin and Wong [2018]	Off-target	CNN and FNN	One-hot	GUIDE-seq [Tsai et al., 2015], CRISPOR [Haeussler et al., 2016]	`https://github.com/MichaelLinn/off_target_prediction`	AUROC : CNN 97.2%, FNN 97%
Xue et al. [2018]	On-target	DeepCas9 (1D CNN)	One-hot	Wang [Wang et al., 2014], Doench V1 [Doench et al., 2014], Doench V2 [Doench et al., 2016], HCT116 [Hart et al., 2015],Z_fish MM [Moreno-Mateos et al., 2015], Z_fish VZ [Varshney et al., 2015], Z_fish GZ [Gagnon et al., 2014], Chari [Chari et al., 2015], Ciona[Gandhi et al., 2016], Farboud [Farboud and Meyer, 2015]	`https://github.com/lje00006/DeepCas9`	Spearman : 0.23–0.61
Liu et al. [2018]	On-target	SeqCrispr (RNN+CNN + transfer learning)	Embedding, sgRNA-DNA binding melting temperature + DNase, CTCF, RRBS, and H3K4me3 peaks + global gene network properties	DeepCRISPR [Chuai et al., 2018], CRISPR-Cpf1 [Kim et al., 2018]	`https://github.com/qiaoliuhub/seqCrisp`	Spearman : 0.77
Wang et al. [2019a]	On-target	DeepHF (RNN)	Embedding + GC content + position-specific and position-independent features + thermodynamic features	DeepHF - The largest gRNA on-target activity dataset for mammalian cells	`http://www.DeepHF.com/`	Spearman : 0.867, 0.862, 0.860

Shrawgi and Sisodia [2019]	On-target	DeepSgRNA (CNN with hierarchical feature generation abilities)	One-hot	GenomeCRISPR [Rauscher et al., 2016]	http://genomecrispr.org	Spearman : 0.82, AUROC : 0.85
Liu et al. [2019]	Off-target	AttnToMismatch_CNN (Transformer + 2D CNN)	Embedding	DeepCRISPR [Chuai et al., 2018]	https://github.com/qiaoliuhub/AttnToCrispr	AUROC : 0.961, AUPRC : 0.071
	Off-target	AttnToCrispr_CNN (Transformer + CNN)	Embedding			Spearman : 0.778, Pearson : 0.781, MSE : 412 ± 27
	On-target	seqCrispr (LSTM + CNN)	One-hot			Spearman : 0.765, Pearson : 0.760, MSE : 442 ± 33
Dimauro et al. [2019]	On-target	CRISPRLearner (deep CNN + data augmentation)	One-hot	Farboud [Farboud and Meyer, 2015], Wang [Wang et al., 2014], Doench V1 [Doench et al., 2014], Doench V2 [Doench et al., 2016], HCT116 [Hart et al., 2015], Z_fish MM [Moreno-Mateos et al., 2015], Z_fish VZ [Varshney et al., 2015], Z_fish GZ [Gagnon et al., 2014], Chari [Chari et al., 2015], Ciona[Gandhi et al., 2016]	https://github.com/pierclgr/crisprlearner	Spearman : 0.23 to -0.69
Wang and Zhang [2019]	On-target	CNN (5 layers + transfer learning)	One-hot	Cas9, eSpCas9, Cas9 (ΔrecA) [Guo et al., 2018]	https://github.com/biomedBit/DeepSgrnaBacteria	Spearman : 0.582, 0.7105, 0.360
Aktas et al. [2019]	Off- and on-target	CNN, MLP, BLSTM	One-hot	DeepCRISPR [Chuai et al., 2018]	https://github.com/bm2-lab/DeepCRISPR	Accuracy : 96.7%
Kim et al. [2019]	On-target	DeepSpCas9 (3 1D-CNN)	One-hot	DeepSpCas9 data	http://deepcrispr.info/DeepSpCas9	Spearman : 0.73

Liu et al. [2020b]	Off-target	CnnCrispr (BLSTM and CNN)	Embedding (GloVe embedding model [Pennington et al., 2014])	DeepCRISPR [Chuai et al., 2018]	https://github.com/LQYoLH/CnnCrispr	AUROC : 0.957, AUPRC : 0.429
Zhang et al. [2020c]	Off-target	DL-CRISPR (Data augmentation)	One-hot	A series of and cell-based assays were collected to generate data using a new data augmentation method	https://github.com/yuuzhang/DL-CRISPR_offtarget_prediction	Accuracy : 98.57%, Sensitivity : 95.13%
Zhang et al. [2020b]	On-target	CNN-SVR	One-hot	DeepCRISPR [Chuai et al., 2018]	https://github.com/Peppags/CNN-SVR	AUROC : 0.94, Spearman : 0.7
Chen et al. [2020a]	Off-target	DNA-BERT and LightGBM	Embedding	DeepCRISPR [Chuai et al., 2018]	https://github.com/bm2-lab/DeepCRISPR	AUROC : 0.993, AUPRC : 0.594, Spearman : 0.276
Zhang et al. [2020a]	On-target	C-RNNCrispr (CNN + RNN)	One-hot	DeepCRISPR [Chuai et al., 2018]	https://github.com/Peppags/C_RNNCrispr	AUROC : 0.976, Spearman : 0.877
Trivedi et al. [2020]	Off-target	Crispr2vec (logistic regression, SVM, DNN)	One-hot	GUIDE-seq [Tsai et al., 2015], CIRCLE-seq [Tsai et al., 2017]	http://www.rgenome.net/cas-offinder	Spearman : 0.60, AUROC : 0.91 on unseen sgRNAs
Lin et al. [2020]	Off-target	CRISPR-Net (LRCN)	One-hot	GUIDE-seq [Tsai et al., 2015], Doench V2 [Doench et al., 2016], CRISPOR [Haeussler et al., 2016], CIRCLE-seq [Tsai et al., 2017], SITE-Seq [Cameron et al., 2017]	https://codeocean.com/capsule/9553651/tree/v1	AUROC : 0.995, AUPRC : 0.317
Zhang et al. [2021]	Off-target	CRISPR-OFFT (1D-CNN, attention)	Embedding	Off-target data sets : Digenome-seq [Kim et al., 2015], GUIDE-seq [Tsai et al., 2015], BLESS [Ran et al., 2015, Slaymaker et al., 2016]	https://github.com/Peppags/CRISPRont-CRISPRofft	AUROC : 0.97, AUPRC : 0.79

	On-target	CRISPR-ONT	Embedding	On-target data sets : DeepHF [Wang et al., 2019a], Sniper-Cas9 [Lee et al., 2018], SpCas9-NG [Nishimasu et al., 2018], xCas9 [Hu et al., 2018]		AUROC : 0.865
Xiao et al. [2021]	On-target	AttCRISPR (embedding-based)	One-hot and embedding	DeepHF [Wang et al., 2019a]	`https://github.com/South-Walker/AttCRISPR`	Spearman : 0.872
Charlier et al. [2021]	Off-target	FNN, CNN, RNN, RF, NB, LR	One-hot	GUIDE-seq [Tsai et al., 2015], CRISPOR [Haeussler et al., 2016]	`https://github.com/dagrate/dl-offtarget`	AUROC : 0.995, AUC PR1 : 0.949, Accuracy : 99.9%
Störtz et al. [2021]	Off-target	piCRISPR (RNN, CNN)	One-hot + physically informed features the target-guide, the target-mismatch, the target-mismatch-type and the target-OR-guide encoding	crisprSQL [Störtz and Minary, 2021]	`https://github.com/florianst/picrispr`	AUROC : 0.983, AUPRC : 0.978, Spearman : 0.1
Xiang et al. [2021]	On- and off-target	CRISPRon and CRISPRoff : Gradient boosting, regression trees, CNN)	One-hot + GC content + position-specific features, position-independent features and thermodynamic features	A pool of 12,000 gRNA oligos targeting 3,834 human protein-coding genes included in CRISPRon database, Kim et al. data set [Kim et al., 2020b]	`https://rth.dk/resources/crispr/crispron`	CRISPRon Spearman : 0.91
Niu et al. [2021]	Off-target	R-CRISPR (bi-directional recurrent network)	One-hot	GUIDE-seq [Tsai et al., 2015], Doench V2 [Doench et al., 2016], CRISPOR [Haeussler et al., 2016], CIRCLE-seq [Tsai et al., 2017], SITE-Seq [Cameron et al., 2017]	`https://codeocean.com/capsule/9553651/tree/v1`	AUROC : 0.991, AUPRC : 0.319

Vinodkumar et al. [2021]	Off-target	GCN-CRISPR (Graph Convolution Network)	One-hot	CRISPOR [Haeussler et al., 2016]	DeepCrispr : https ://doi.org/10.1186/s13059-018-1459-4, CnnCrispr : https ://doi.org/10.1186/s12859-020-3395-z	AUROC : 0.987
Zhang and Jiang [2022]	Off-target	CRISPR-IP (CNN, BLSTM)	One-hot	CIRCLE-Seq [Tsai et al., 2017], SITE-Seq [Cameron et al., 2017]	`https://github.com/BioinfoVirgo/CRISPR-IP`	AUROC : 0.982, AUPRC : 0.751, Accuracy : 0.990
Elkayam and Orenstein [2022]	On-target	DeepCRISTL (BLSTM + transfer learning)	Embedding + GC content + position-specific, position-independent, and thermodynamic features	DeepHF [Wang et al., 2019a], CRISPRon [Xiang et al., 2021]	`https://github.com/OrensteinLab/DeepCRISTL`	Spearman : 0.878
Fu et al. [2022]	Off-target	MOFF (two CNN regression models)	One-hot	GUIDE-seq [Tsai et al., 2015], CHANGE-seq [Lazzarotto et al., 2020], TTISS [Schmid-Burgk et al., 2020]	`https://github.com/MDhewei/MOFF`	Spearman : 0.5

RNN : Recurrent Neural Network, CNN : Convolutional Neural Network, FNN : Feedforward Neural Network, DCDNN : Deep Convolutional Denoising Neural Network, LSTM : Long Short-Term Memory, BLSTM : Bidirectional Long Short-Term Memory, LRCN : Long-term Recurrent Convolutional Network, SVR : Support Vector Regression.

1.8 Conclusions and outlook

Artificial intelligence methods have emerged as state-of-the-art approach in the field of genome editing. We reviewed recent applications of traditional machine learning and deep learning algorithms for prediction of on- and off-target activity in CRISPR/Cas9. We believe that our review paper can serve as a guideline for CRISPR/Cas9 practitioners willing to apply artificial intelligence methods in genome editing.

1.8.1 Main Conclusions

The main conclusions of our study are as follows :

- First, we highlighted the importance of sequence encoding for sgRNA-DNA on- and off-target prediction. Initial models implied straightforward one-hot sequence encoding of the sgRNA-DNA sequence pairs [Lin and Wong, 2018]. Subsequent sequence encoding schemes were introduced [Charlier et al., 2021] demonstrating higher predictive performance. The latest efforts have been focusing on the supplementary information embedding with different channels reflecting insertions, deletions and mismatches [Lin et al., 2020].
- Second, some recent work has demonstrated that the ensemble machine learning methods have generally outperformed non-ensemble methods [Zhang et al., 2019b, Abadi et al., 2017]. For instance, AdaBoost [Zhang et al., 2019b] and Random Forest [Abadi et al., 2017] led to superior predictive performance than a standard logistic regression or an SVM [Fusi et al., 2015].
- Third, recent studies have highlighted the importance of feature selection and feature engineering for accurate activity prediction in CRISPR/Cas9. New methodologies have been introduced to incorporate sequence information such as gene melting temperature, molecular weight, or microhomology features [Wang et al., 2019a]. Some works emphasize the need of automated feature learning and automated feature engineering to boost the performance of deep learning models [Zhang and Jiang, 2022].
- Fourth, we observed that most of publicly available data sets have incomparable numbers of positive and negative samples, thus leading to a class imbalance problem that has a negative impact on the performance of both traditional machine learning and deep learning methods, especially when predicting off-targets. Recent papers propose to use data augmentation to increase the number of samples of the minority class [Zhang et al., 2020c] or to apply some standard re-sampling techniques, such as under-sampling [Liu et al., 2020b], to mitigate the impact of data imbalance [Heaton, 2018].
- Fifth, for sufficiently large data sets, deep neural networks have demonstrated their superior predictive performance in comparison to both scoring methods and conventional machine learning algorithms such as SVM, Random Forest and XGBoost [Lin and Wong, 2018, Wang et al., 2019a, Lin et al., 2020, Charlier et al., 2021]. However, for smaller data sets, simpler machine learning tools were sometimes able to outperform some of their deep learning counterparts [Konstantakos et al., 2022b].
- Sixth, attention-based deep learning models have been extensively used in some recent works in the field [Zhang et al., 2021, Xiao et al., 2021, Vaswani et al., 2017]. The attention mechanisms have

been shown to increase the efficacy of the deep learning process [Chen et al., 2022, Shen et al., 2022, de Santana Correia and Colombini, 2022, Basiri et al., 2021]. The latest deep learning models that rely on recurrent neural networks and attention-based mechanism have demonstrated very promising prediction performances [Liu et al., 2019, Xiao et al., 2021].

1.8.2 Research Gaps and Future Research Directions

Research gaps and future research directions related to the application of artificial intelligence methods in genome editing include :

- Powerful deep learning models have a huge number of parameters that need to be tuned in the training process. Thus, to be effective, these methods require large amounts of input data. Transfer learning started to be used in the field to leverage the problem of deep learning training on data sets with insufficient amount of training samples. For instance, Lin et al. and Charlier et al. [Lin and Wong, 2018, Charlier et al., 2021] have successively employed transfer learning to predict the off-target sequences in small data sets. Further experiments should show how the most appropriate larger data sets used for training could be selected. Moreover, it would be interesting to see whether some pretrained deep learning models could be effectively used for transfer learning in CRISPR/Cas9.
- Deep neural networks usually stack several layers of different types, each of which often containing dozens of neurons. Thus, designing efficient deep learning network architectures and finding optimal sets of optimization hyperparameters are extremely important and challenging tasks [Cho et al., 2020, Wu et al., 2020]. Various hyperparameter tuning techniques which should be extensively tested with CRISPR/Cas9 data include : evolutionary strategies, random grid search, exhaustive grid search, and Bayesian optimization [Heaton, 2018].
- Explainability and interpretability of deep neural networks has been a topic of interest for the past few years [Heaton, 2018, Ribeiro et al., 2016]. Recent methodologies have been introduced to further address the lack of human-level explainability and interpretability [Miller, 2019, Chou et al., 2022, Vilone and Longo, 2021]. Future research in genome editing could fill the gap in understanding the nature of on- and off-target activities, an important milestone for clinical applications.
- As we pointed out, some recent works in the field have focused on the use of features engineering to boost the predictive performance of machine learning models [Wang et al., 2019a, Shrawgi and Sisodia, 2019]. Informative features such as epigenetic features, microhomology properties, or RNA

fold score, can be further exploited to increase accuracy. Convolutional layers of CNN and LRCN deep learning networks are able to discover useful features from sequences directly and independently, avoiding eventual biases introduced by hand-crafted features [Shrawgi and Sisodia, 2019, Lin et al., 2020, Niu et al., 2021]. The use of the SHAP [Lundberg and Lee, 2017] (SHapley Additive exPlanations - this algorithm gives an explanation to the model's behavior, connecting optimal credit allocation with local explanations using the classic Shapley values from game theory), Tree SHAP [Lundberg et al., 2018] (this algorithm calculates SHAP values for tree-based models), and Deep SHAP [Lundberg and Lee, 2017] algorithms (this is a high-speed approximation algorithm for SHAP values) is highly recommended to assess how each feature impacts the selected model.

— Uncertainty quantification is a key technique to improve the trustworthiness of predictions made by a trained network. This technique has become popular for evaluating uncertainty in various research fields [Abdar et al., 2021a,b, 2023, Hoffmann et al., 2021, Mazoure et al., 2022, Sherkatghanad et al., 2025]. There are two types of uncertainty : the aleatoric uncertainty that is an inherent property of the data distribution, and the epistemic uncertainty that refers to the model's uncertainty. This technique could be effectively applied in genome editing to improve the trustworthiness of on- and off-target predictions. Kirillov et al. [2022] have recently designed one of the first methods that incorporates uncertainty into the final prediction. It is designed to provide interpretable evaluation of Cas9–gRNA and Cas12a–gRNA specificity using deep kernel learning, predicting the cleavage efficiency of a gRNA with a corresponding confidence interval.

— Active learning is a semi-supervised technique in which a learning algorithm is used to label unlabeled data. An active learning algorithm uses an initial subset of labeled data for training. The algorithm then predicts the most appropriate labels for unlabeled data. This technique is of particular interest in biology because obtaining labeled data is often costly and time consuming [Gordon et al., 2018, Lee et al., 2019, Nguyen et al., 2020]. Active learning can be employed in genome editing in situations when unlabeled data are abundant, while accurate automatic or manual labeling is impossible.

1.9 Key Points

— We reviewed current knowledge regarding the use of supervised machine learning methods for on- and off-target prediction in CRISPR/Cas9.

— We highlighted the importance of the data pre-processing step including encoding of the sgRNA-DNA sequence pairs without any information loss, embedding supplementary data with different

channels reflecting insertions, deletions and mismatches, and considering some additional sequence information such as gene melting temperature, molecular weight, or microhomology features.

— Most of CRISPR/Cas9 data sets have incomparable numbers of positive and negative samples, thus leading to a class imbalance situation that should be mitigated using either data augmentation or data re-sampling techniques, especially in the case of off-target prediction.

— When training data sets were large enough, deep neural networks have demonstrated their superior predictive performance in comparison to scoring methods and traditional machine learning algorithms. However, for benchmark purpose, the results obtained using state-of-the-art deep learning methods should be compared with those provided by some effective conventional machine learning algorithms, such as SVM, random forest, and XGBoost.

— We emphasized the importance of feature selection for accurate on- and off-target prediction in CRISPR/Cas9. Thus, the automated feature learning and automated feature engineering techniques should be used to boost the performance of deep learning models.

CHAPITRE 2

BAYTTA : UNCERTAINTY-AWARE MEDICAL IMAGE CLASSIFICATION WITH OPTIMIZED TEST-TIME AUGMENTATION USING BAYESIAN MODEL AVERAGING

This chapter is a reproduction of the following article : Zeinab Sherkatghanad, Moloud Abdar, Mohammadreza Bakhtyari, Pawel Plawiak, Vladimir Makarenkov, "BayTTA : Uncertainty-aware medical image classification with optimized test-time augmentation using Bayesian model averaging", Knowledge-Based Systems (2025) : 114123.

2.1 Abstract

2.1.1 Motivation

Test-time augmentation (TTA) is a well-known technique employed during the testing phase of computer vision tasks. It involves aggregating multiple augmented versions of input data. Combining predictions using a simple average formulation is a common and straightforward approach after performing TTA.

2.1.2 Results

This research work introduces a novel framework for optimizing TTA, called BayTTA (Bayesian-based TTA), which is based on Bayesian Model Averaging (BMA). First, we generate a prediction list associated with different variations of the input data created through TTA. Then, we use BMA to combine predictions weighted by the respective posterior probabilities. Such an approach allows one to take into account model uncertainty, and thus to enhance the predictive performance of the related machine learning or deep learning model. We evaluate the performance of BayTTA on various public data, including three medical image datasets comprising skin cancer, breast cancer, and chest X-ray images and two well-known gene editing datasets, CRISPOR and GUIDE-seq.

2.1.3 Conclusions

Our experimental results indicate that BayTTA can be effectively integrated into state-of-the-art deep learning models used in medical image analysis as well as into some popular pre-trained CNN models such as VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and InceptionRes-NetV2, leading to the enhancement in their

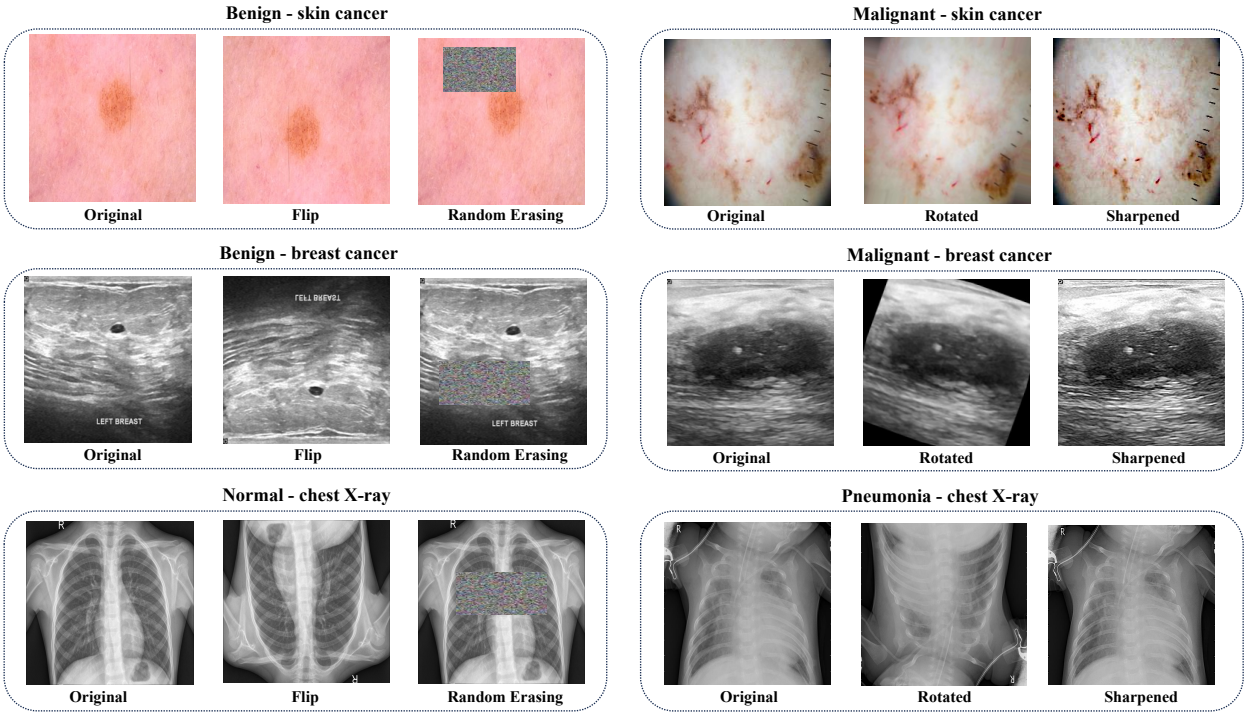

Figure 2.1 – Examples of original benign and malignant skin cancer, breast cancer, and chest X-ray images and their augmented versions considered in our study.

accuracy and robustness performance.

2.1.4 Availability and Implementation

The source code of the proposed BayTTA method is freely available at : https ://github.com/Z-Sherkat/BayTTA.

2.2 Introduction

Deep learning models are highly effective tools for identifying complex patterns in data. However, they are also prone to overfitting and memorizing the training data rather than generalizing real data patterns. Data augmentation is a critical strategy to overcome this issue, which involves synthesizing new training samples by applying to the original data a series of transformations and perturbations, such as random erasing, flipping, rotating, and scaling (see Fig. 3.2). Leveraging data augmentation techniques effectively introduces variability in the training data, leading to a more robust and general model. Machine learning (ML) and deep learning (DL) models equipped with data augmentation can handle the complexities inherent in medical image data, often leading to an enhanced diagnostic accuracy [Tsuneki, 2022, Biswas et al., 2023, Goceri,

2023, Bozkurt, 2023].

Test-Time Augmentation (TTA) is a popular data augmentation technique which can increase the prediction power of ML and DL models. TTA involves applying data augmentation on test input rather than on training data. It provides a more reliable estimate of the target variable by averaging the predictions across augmented inputs to find the output result. While data augmentation is mainly used for improving training data diversity to enhance the model generalization on unseen data, TTA produces multiple versions of each test image to obtain a more robust estimate of the target variable [Jin et al., 2018, Maron et al., 2021].

Recently, Uncertainty Quantification (UQ) methods have been effectively applied in a variety of research domains, contributing to the robustness and reliability of the optimization and decision-making processes. There are two primary categories of uncertainty [Abdar et al., 2021a, Hüllermeier and Waegeman, 2021], referred to as aleatoric and epistemic. While aleatoric uncertainty pertains to the inherent characteristic of the data distribution, epistemic uncertainty relates to the lack of knowledge about the optimal model. Uncertainty quantification is a vital tool for addressing the limitations of data (aleatoric uncertainty) and models (epistemic uncertainties) in various scientific and engineering applications, and thus for improving the trustworthiness of predictions, especially in the testing phase [Abdar et al., 2021b]. Although deep learning models often outperform traditional machine learning methods, overconfident predictions remain their crucial issue, leading to excessive prediction errors [Sherkatghanad et al., 2020]. By combining uncertainty quantification with TTA, we can gain a deeper insight into the reliability and robustness of model predictions [Wang et al., 2019c, Bahat and Shakhnarovich, 2020, Ayhan and Berens, 2022]. This is particularly beneficial for machine learning and deep learning applications in healthcare, autonomous systems, and safety-critical domains, where understanding and managing uncertainty may be critical.

Here, we describe a novel technique, called BayTTA, that employs Bayesian Model Averaging (BMA) logistic regression to optimize TTA. Our new method applies BMA on the predictions obtained through TTA. BMA can effectively handle model uncertainty, combining multiple models generated by TTA and offering precise and robust predictions. It aggregates predictions from all candidate models, weighted by their posterior probabilities, and results in a model-averaged prediction that accounts for model uncertainty.

This article showcases a number of contributions that are as follows :

- We propose the BayTTA method for optimizing TTA using the BMA approach and apply it to medical image classification.
- We estimate the uncertainty associated with predictions obtained through TTA and BMA, offering further insight into the reliability of our method.
- We assess the performance of BayTTA on three public medical image datasets, including skin cancer,

breast cancer, and chest X-ray images available on the Kaggle and Mendeley data repositories as well as on two popular gene editing datasets, CRISPOR and GUIDE-seq, generated using the CRISPR-Cas9 technology [Haeussler et al., 2016]. We demonstrate that our method can be applied successfully with different deep learning network architectures. Our results indicates a superior performance of the proposed method compared to standard averaging.

— We evaluate the performance of BayTTA used in combination with some well-known pre-trained deep learning networks, including VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and Inception-ResNetV2.

— We demonstrate how incorporating BayTTA enhances the predictive power and robustness of state-of-the-art deep learning models used in medical image analysis, compared to standard TTA.

2.3 Related work

2.3.1 Uncertainty quantification

Technical progress : Uncertainty quantification (UQ) is essential to enhance the credibility of predictions during the testing phase. Since the excessive confidence of deep neural networks may lead to prediction errors, it is imperative to address the issue of overconfident predictions in order to improve their reliability and trustworthiness [Hamedani-KarAzmoudehFar et al., 2023]. As deep learning models are now constantly used in critical areas, the ability to quantify and manage uncertainty becomes increasingly vital [Hoffmann et al., 2021, Mazoure et al., 2022, Abdar et al., 2023].

Nowadays, UQ has important applications in image processing, computer vision, medical image analysis, diagnostic modeling, and healthcare decision-making [Abdar et al., 2021a, Lambert et al., 2022, Loftus et al., 2022, Seoni et al., 2023]. There are several well-known methods for measuring uncertainty such as Monte Carlo dropout [Gal and Ghahramani, 2016], Variational Inference [Wang and Van Hoof, 2020, Rudner et al., 2022], Deep Ensembles [D'Angelo and Fortuin, 2021, Rahaman et al., 2021, Abe et al., 2022], and Bayesian Deep Ensembles [He et al., 2020]. Bayesian Model Averaging (BMA) is another effective technique used to take into account prediction uncertainty [Wintle et al., 2003, Monteith et al., 2011, Izmailov et al., 2021, Bartoš et al., 2021]. Development of BMA in the context of model uncertainty has been influenced by the seminal works of [Draper, 1995, 2013] and [Hoeting et al., 1998] as their original methods provide insight into the quantification of both epistemic and aleatoric uncertainties.

UQ application in medical image analysis : Nowadays, UQ methods have become a tool of choice for estimating the uncertainty associated with disease detection, diagnosis, medical image segmentation, and identification of the region of interest (ROI) in medical image analysis. In their recent work, Abdar et al. [2021b] have introduced an efficient hybrid dynamic model of uncertainty quantification, called TWDBDL. The model is based on the Three-Way Decision (TWD) theory and Bayesian Deep Learning (BDL) methods used together to improve the trustworthiness of predictions in skin cancer detection. Edupuganti et al. [2021] used a probabilistic variational autoencoders (VAEs), i.e. a Monte Carlo technique to generate pixel uncertainty maps, and Stein's Unbiased Risk Estimator (URE) to provide accurate uncertainty estimations in knee magnetic resonance imaging. Gour and Jain [2022] introduced the UA-ConvNet, i.e. an uncertainty-aware Convolutional Neural Network (CNN) model, for COVID-19 detection in chest X-ray (CXR) images. The model estimates the uncertainty based on the EfficientNet-B3 Bayesian network supplemented with Monte Carlo dropout.

Mazoure et al. [2022] designed a novel web server, called DUNEScan, for uncertainty estimation in CNN models applied to skin cancer detection. The web server employs binary dropout to compare the average model predictions, providing visualization for uncertainty to diagnose precisely skin cancer cases. Han et al. [2024] proposed a novel dynamic multi-scale convolutional neural network (DM-CNN) that leverages a hierarchical dynamic uncertainty quantification attention (HDUQ-Attention) submodel. HDUQ-Attention includes a tuning block for adjusting the attention weights as well as Monte Carlo dropout for quantifying uncertainty. The experiments conducted on skin disease images (HAM10000), colorectal cancer images (NCT-CRC-HE-100K), and lung disease images (OCT2017 and Chest X-ray) demonstrated that the DM-CNN model accurately quantifies uncertainty, showing a stable performance.

UQ application in gene editing : Uncertainty quantification can be effectively applied in the context of gene editing in order to improve the trustworthiness of on- and off-target predictions in CRISPR-Cas9 experiments [Sherkatghanad et al., 2023].

For example, Zhang et al. [2020c] presented a deep learning model for off-target activity prediction, employing data augmentation to mitigate the class imbalance issue. The authors collected data from two source types, i.e. in vitro and cell-based experiments, to increase the size of the positive class samples (off-targets). They suggested synthetically expanding the number of positive samples by rotating the sgRNA-DNA encoded images by 90, 180, and 270 degrees, respectively, to enhance the model competency.

Moreover, Kirillov et al. [2022] have recently introduced a pioneering method that incorporates uncertainty into off-target predictions. Their approach offers an interpretable evaluation of Cas9-gRNA and Cas12a-gRNA specificity through deep kernel learning, estimating a gRNA's cleavage efficiency with a corresponding confidence interval.

2.3.2 Test-time augmentation

Technical progress : Test-time augmentation (TTA) is a well-known technique that applies data augmentation during the testing phase. TTA has multiple benefits, including improving the model generalization and reliability capacity [Song et al., 2017, Cohen et al., 2019], estimating uncertainty in model predictions [Wang et al., 2019c, Bahat and Shakhnarovich, 2020, Ayhan and Berens, 2022], and boosting accuracy in classification and segmentation tasks [Krizhevsky et al., 2012, Szegedy et al., 2015, He et al., 2016, Shanmugam et al., 2021]. Figure 2.2 presents a detailed flowchart of a typical TTA process.

Recently, some advanced TTA methods using diverse aggregation strategies have been proposed. They include, among others, selective augmentation techniques such as the instance-aware TTA based on a loss predictor [Kim et al., 2020a], the instance-level TTA with entropy weight method [Chun et al., 2022], and the selective-TTA method [Son and Kang, 2023].

Lyzhov et al. [2020] presented a straightforward and efficient method based on a policy search algorithm, known as greedy policy search (GPS), designed to optimize image classification. The method aims to optimize and determine the most effective data augmentation strategy to be applied during the testing phase of a machine learning process, and thus to improve the prediction accuracy and robustness of ML models. Kim et al. [2020a] developed an instance-aware test-time augmentation approach that employs a loss predictor to dynamically select test-time transformations based on the expected losses for individual instances. Furthermore, Chun et al. [2022] introduced an instance-level TTA with Entropy Weight Method (EWM) as an innovative approach to improve the accuracy and robustness of classification models.

The paper of Shanmugam et al. [2021] describes a new method for aggregating model predictions obtained from TTA. Unlike the traditional approach of averaging model predictions, this method focuses on learning different augmentation weights to aggregate predictions obtained from transformations during TTA. The authors noticed that existing aggregation methods, based on the mean or the maximum of predictions obtained from augmented images, may not be optimal because they do not consider the relationship between

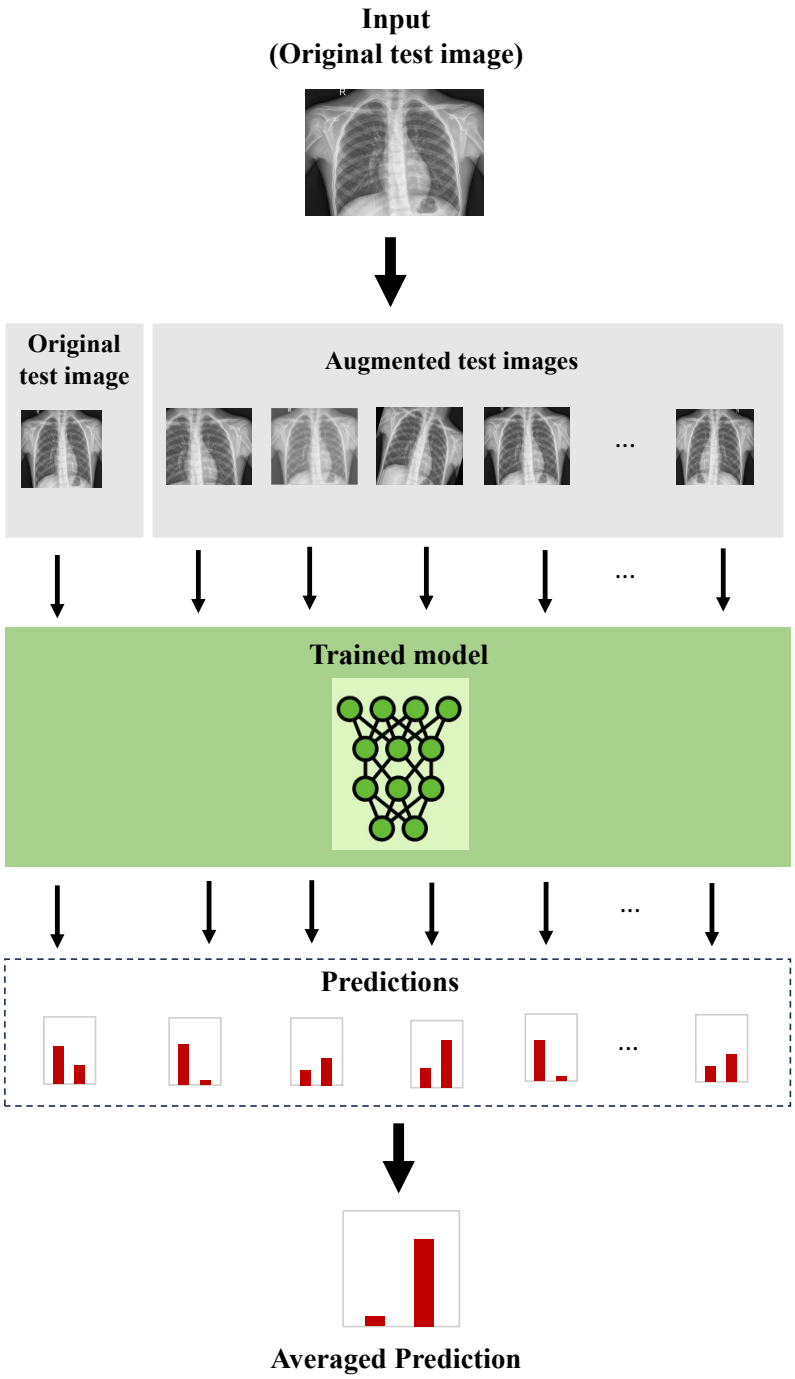

Figure 2.2 – A schematic view of a conventional test-time augmentation (TTA) process.

the original test image and its augmented versions. The paper also offers insights into the point when TTA can be helpful and provides guidance regarding the use of different TTA policies. Moreover, Shanmugam and co-authors characterized the cases where TTA transforms correct predictions into incorrect ones, and vice versa.

TTA application in medical image analysis : TTA has been extensively studied by the medical imaging community due to its capability to contribute to model robustness and improve the trustworthiness and generalization of predictions during the testing phase. Wang et al. [2019c] conducted critical research in the context of deep convolutional neural network (CNN)-based medical image segmentation. Their work focuses on epistemic and aleatoric uncertainty analysis at pixel and structure levels, providing valuable insights into the reliability of segmentation results. Gaillochet et al. [2022] introduced a simple and powerful task-agnostic semi-supervised active learning segmentation approach, called TAAL (TTA for Active Learning). TAAL exploits unlabeled samples during training and sampling phases by using a technique known as cross-augmentation consistency.

2.4 Methodology

In this section, we elaborate on the background, foundation, and comprehensive explanation of the methodological procedures used within the proposed method. Our new method, called BayTTA, aims to optimize TTA using BMA and uncertainty estimation. The first component of our method involves formulating a mathematical model that applies BMA to output predictions generated from multiple transformed versions of the input data (e.g. input images). The second component of the method estimates the uncertainty of the predictions obtained from an augmented set of test images associated with image transformations or noise. Together, these components offer a comprehensive approach to enhance the robustness and reliability of ML or DL models during the testing phase. An overview of the proposed BayTTA method is presented in Figure 2.3.

Traditional TTA involves applying a number of data augmentation techniques, including rotation, cropping, flipping, and brightness adjustments, to the test data before making predictions. This widely used approach leverages ensemble methods, such as averaging or taking the majority vote of predictions made on augmented samples, to reduce the impact of random noise and variation in the test data. TTA usually produces stable and accurate predictions by averaging multiple augmented versions of the test dataset [Wang et al., 2019c,d].

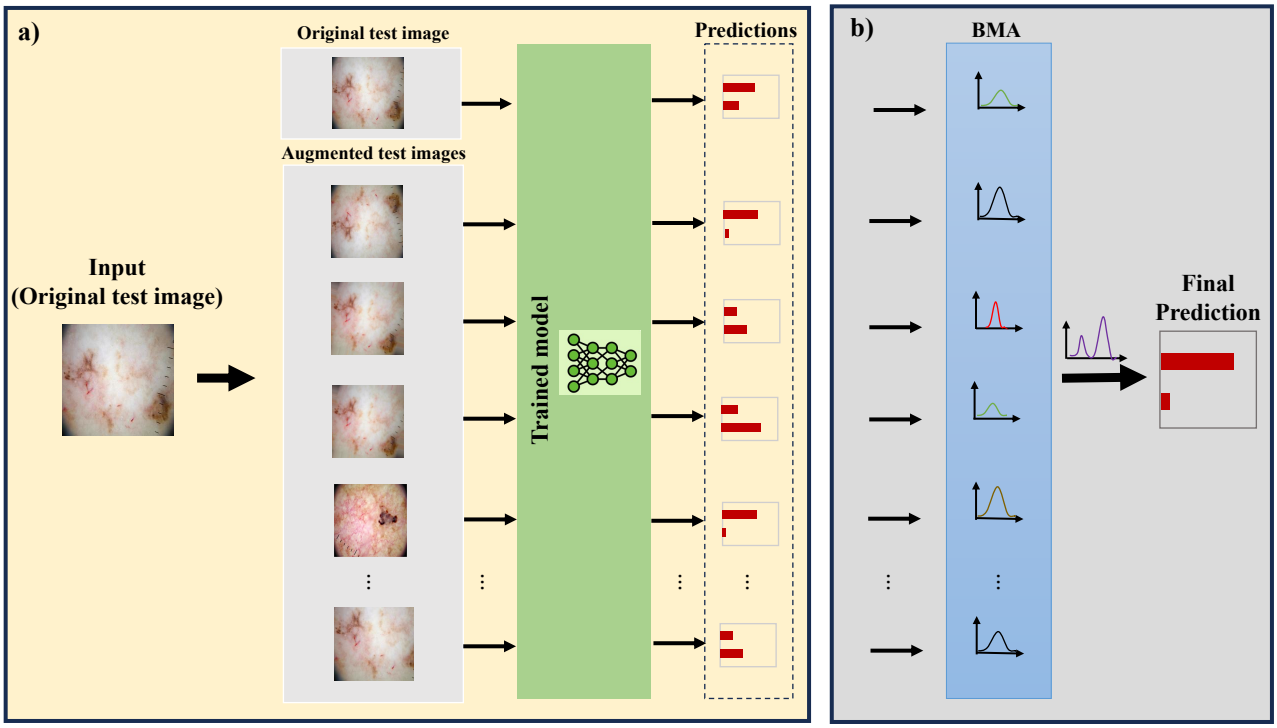

Figure 2.3 – An overview of the proposed BayTTA method. During the testing phase : (a) TTA generates predictions from a set of fixed augmented images, and (b) BMA is then applied to combine and aggregate these predictions by treating each unique combination as a distinct candidate model.

Our study implements BMA to integrate predictions obtained from TTA (i.e. predictions obtained from the same model but using different distorted images), thus deviating from the conventional averaging approach used in TTA to yield more precise and robust outcomes. Let us focus on a single image I_y from the test dataset, where $y \in \{0, 1\}$ denotes its classification label. In a scenario generating k augmented versions of the test image in TTA, the input to the BMA comes from the vector $\mathbf{x} = \{x_1, x_2, ..., x_{k+1}\}$, where each element corresponds to a prediction from the TTA procedure. Here, x_1 denotes the prediction derived from the original test image I_y, and $\{x_2, x_3, ..., x_{k+1}\}$ denotes the predictions from augmented versions of this image.

Rather than treating the TTA outputs as independent predictions to be averaged, we consider them as predictor variables, each with a unique combination, defining a distinct candidate model within the BMA framework. BMA involves generating various models by combining predictor variables and selecting candidate models based on likelihood. To formalize this process, we introduce $\mathcal{P}(\mathbf{x})$, a non-empty power set of $\mathbf{x}$, representing all combinations of predictor variables. Subsequently, utilizing $I \in \mathcal{P}(\mathbf{x})$, we select desired

predictor variables from $\mathbf{x}$ for model M_I, where $\mathbf{x}_I$ represents the input of the respective model M_I. In the context of Bayesian model averaging logistic regression, we further elucidate the mathematical formulations that define our algorithm in the subsequent discussion.

According to the Bayes theorem, the posterior distribution for model M_I is given by :

$$p(M_I \mid \mathbf{x}_I, y) = \frac{p(\mathbf{x}_I, y \mid M_I)p(M_I)}{\mathcal{L}_{total}}, \tag{2.1}$$

where $p(M_I)$ denotes the prior probability of M_I, $\mathcal{L}_{total}$ is the marginal likelihood, and $p(\mathbf{x}_I, y \mid M_I)$ is the likelihood of M_I estimated using the Bayesian Information Criterion (BIC) :

$$\begin{aligned} BIC_I &= p_I \ln(N) - 2\ln(\tilde{L}_I), \\ \mathcal{L}_{M_I} &= p(\mathbf{x}_I, y \mid M_I) = e^{-BIC_I/2}. \end{aligned} \tag{2.2}$$

Here, p_I is the number of parameters in model M_I, N is the number of data points in the input data, and $\tilde{L}_I$ is the maximized value of the likelihood function for model M_I [Draper, 1995]. The maximum likelihood estimate $\tilde{L}_I$ is not computed directly but is obtained implicitly through the model fitting process that involves the maximization of the log-likelihood function using iterative numerical optimization. In our implementation, we use the `statsmodels` library [Seabold and Perktold, 2010], where the logistic regression is carried out using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [Fletcher, 2000]. BFGS is a widely used quasi-Newton optimization method that iteratively approximates the Hessian matrix, enabling efficient convergence to the maximum likelihood estimates.

To integrate predictions obtained from TTA using BMA, we define the posterior probability of each candidate model :

$$p(M_I \mid \mathbf{x}_I, y) = \frac{e^{-BIC_I/2}p(M_I)}{\mathcal{L}_{total}}, \tag{2.3}$$

where M_I represents a model defined by a specific combination of predictor variables derived from the TTA procedure. The marginal likelihood, $\mathcal{L}_{total}$, is computed as the sum over all candidate models, and defined as follows :

$$\mathcal{L}_{\text{total}} = \sum_j e^{-\text{BIC}_j/2}\, p(M_j) = \sum_{M_j \in \mathcal{P}(\mathbf{x})} e^{-\text{BIC}_j/2}\, p(M_j), \tag{2.4}$$

where each M_j denotes a distinct model formed from all possible combinations of predictor variables generated through TTA. The model selection process is guided by the likelihood of each model, as quantified by the BIC, enabling the integration of multiple augmented predictions to improve the robustness and precision of the final model output.

In BMA, the Bayes factor (BF) plays a crucial role in model selection. Thus, recognizing the relationship between posterior model probabilities and Bayes Factors application is crucial. The Bayes factor, for comparing model M_I against model M_J, is defined as the ratio of the posterior odds to the prior odds of the two models [Hoeting et al., 1998]. When models are assigned equal prior probabilities (i.e., $p(M_I) = p(M_J)$), the Bayes factor simplifies to the ratio of their posterior probabilities. Formally, the Bayes factor comparing the models M_I and M_J is given by :

$$BF_{IJ} = \frac{p(M_I \mid \mathbf{x}_I, y)}{p(M_J \mid \mathbf{x}_J, y)}. \tag{2.5}$$

If the value of BF exceeds 1, the observed data strongly favors model M_I over model M_J. In practical terms, this implies that the information provided by BIC can guide us in selecting the best model that is the model with the highest log of marginal likelihood and, consequently, the smallest BIC.

Computing output values for the Bayesian averaging logistic regression model involves determining the probability of any predictor variable, $p(x_i)$, and the expected value of the coefficient associated with this predictor variable, i.e. $E[\beta_i]$:

$$\begin{aligned} p(x_i) &= \sum_{M_I \ such \ that \ x_i \in \mathbf{x}_I} p(M_I \mid \mathbf{x}_I, y), \\ E[\beta_i] &= \sum_{M_I \ such \ that \ x_i \in \mathbf{x}_I} p(M_I \mid \mathbf{x}_I, y) \times \beta_i^I. \end{aligned} \tag{2.6}$$

Here, $i \in \{1, 2, ..., k+1\}$ and β_i^I is the coefficient of x_i in model M_I - it accounts for the impact of predictor variable x_i within that specific model. In our implementation, the model parameters are estimated using the logistic regression function from the `statsmodels` library [Seabold and Perktold, 2010]. Specifically, each candidate model M_I is fitted using the `Logit` function from this library (the fitted model is referred to as `model_regr`). The estimated coefficients correspond to the maximum likelihood estimates obtained during the model fitting. They are subsequently used to compute the posterior-weighted expectations $E[\beta_i]$, as part of the Bayesian model averaging framework. The BayTTA model carries out a series of steps to compute the probabilities : $p(\mathbf{x}) = (p(x_1), p(x_2), ..., p(x_{k+1}))$, and the expected values : $E[\boldsymbol{\beta}] = (E[\beta_1], E[\beta_2], ..., E[\beta_{k+1}])$. Algorithm 1 presents the pseudocode outlining the main steps of our method.

We define the set $I_{current} \subseteq \mathcal{P}(\mathbf{x})$ to identify candidate models that we want to process at each iteration. Initially, we form $I_{current} = \{\{1\}, \{2\}, ..., \{k+1\}\}$, where each $I \in I_{current}$ corresponds to a model M_I with one (i.e. $m = 1$) predictor variable. Subsequently, logistic regression on each model is carried out to calculate the BIC values and the coefficients of the predictor variables. Following this step, we assess the

Algorithme 2.1 Optimizing TTA using BMA (BayTTA)

Entrée: Trained model

Entrée: Original test image X_{test}

Entrée: Set of k transformations

Sortie: Uncertainty estimation for test image X_{test}

$\hat{\mathbf{p}} \leftarrow \mathbf{0}_{k+1}, \hat{E}[\boldsymbol{\beta}] \leftarrow \mathbf{0}_{k+1}$

$\mathcal{L}_{total} \leftarrow 0, \mathcal{L}_{max} \leftarrow 0$

pour $i \leftarrow 1, ..., k+1$ **faire**

  Calculate I_{next} ▷ $\{I_{next} \subseteq \mathcal{P}(\mathbf{x}) \mid \text{length}(S \in I_{next}) = i\}$

  $I_{current} = \emptyset$

  si $i == 1$ **alors**

    $I_{current} = I_{next}, I_{previous} = \emptyset$

  sinon

    $A = \{S \in I_{next} \mid S \text{ contains an element of } I_{previous}\}$

    $I_{current} = I_{current} \cup A, I_{previous} = \emptyset$

  fin si

  pour I **in** $I_{current}$ **faire**

    $\boldsymbol{\beta}' \leftarrow \texttt{Logit}(\mathbf{x}_I, y)\texttt{.fit()}$ ▷ Carry out logistic regression using predictor variables $\mathbf{x}_I$.

    $\mathcal{L}_{M_I} \leftarrow e^{-BIC_I/2}$ ▷ using Eq. 2.2

    si $\mathcal{L}_{M_I} > \mathcal{L}_{max}$ **alors**

      $\mathcal{L}_{total} \mathrel{+}= \mathcal{L}_{M_I}$ ▷ using Eq. 2.4

      $\mathcal{L}_{max} = \mathcal{L}_{M_I}$

      pour j **in** I **faire**

        $\hat{\mathbf{p}}_j \mathrel{+}= \mathcal{L}_{M_I}$ ▷ for Eq. 2.6

        $\hat{E}[\boldsymbol{\beta}_j] \mathrel{+}= \boldsymbol{\beta'}_j \times \mathcal{L}_{M_I}$ ▷ for Eq. 2.6

      fin pour

      $I_{previous} = I_{previous} \cup I$

    fin si

  fin pour

fin pour

Calculate $p(\mathbf{x})$, $E[\boldsymbol{\beta}]$ and y_{BMA} ▷ using Eq. 2.6 and Eq. 2.7

Calculate the uncertainty, σ_{BayTTA} ▷ using Eq. 2.8

model's likelihood using the estimated value of BIC, as outlined in Eq. 2.2. To finalize the model selection process in BMA (see Eq. 2.5), we specify a uniform prior distribution for all candidate models and set an initial threshold $\mathcal{L}_{max}$ to zero. For each model M_I whose likelihood exceeds $\mathcal{L}_{max}$, we perform the following steps : (1) replace the threshold $\mathcal{L}_{max}$ with the likelihood of the model M_I, (2) aggregate its likelihood to calculate $\mathcal{L}_{total}$ (i.e. the denominator in Eq. 2.3), and (3) update the probabilities and the coefficients of the predictor variables of the model M_I using Eqs. 2.6. At the subsequent iteration, we build a set of candidate models with one additional predictor variable, $m = m + 1$, based on models meeting the threshold at the previous iteration. After repeating this procedure $k + 1$ times, and considering all possible combinations of predictor variables for generating models, the BMA prediction can be calculated as follows :

$$y_{BMA} = \frac{1}{1 + exp(-E[\boldsymbol{\beta}]^T \mathbf{x})}. \tag{2.7}$$

In this procedure, the probability of each predictor variable, $p(x_i)$, corresponds to TTA augmentations, and is defined as the sum of the probabilities of all models that incorporate this predictor variable. Furthermore, the expected value for the coefficient of each predictor variable, $E[\beta_i]$, is calculated as a weighted average of the coefficients determined by the posterior probability across all models that include this predictor variable. By implementing such a TTA optimization technique that selects more confident augmentations during the testing phase, we ensure that it improves predictive performance, uncertainty estimations, and the overall robustness of deep learning models, compared to simple averaging.

The uncertainty measure, σ_{BayTTA}, for a given original image is defined as follows :

$$\sigma_{BayTTA} = \sqrt{\frac{1}{k+1} \sum_{i=1}^{k+1} \Big(p(x_i) \times (acc_i - \mu_{BMA}) \Big)^2}, \tag{2.8}$$

where acc_i is the accuracy of a baseline model (e.g., in our study, we used VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and InceptionResNetV2 as baseline - see Section 4) obtained for image i ($i = 1, ..., k + 1$) - it includes the original image, when $i = 1$, and k of its augmented versions ; μ_{BMA} is the logistic regression accuracy obtained using BayTTA for the original image and k of its augmented versions.

Thus, in the context of BMA, the uncertainty of the predictions, σ_{BayTTA}, is quantified using a square root from a weighted squared difference between the accuracy of each involved image i ($i = 1, ..., k + 1$), acc_i, and the BMA accuracy, μ_{BMA}. The BMA accuracy is derived by the logistic regression by means of the BMA framework. The formula for σ_{BayTTA} (see Eq. 2.8) assesses the variation in the prediction accuracy related to the original image and its k augmented versions obtained through TTA. Hence, the defined uncertainty

measure captures the accuracy variation, and thus the uncertainty in the model predictions, in the framework of our BMA approach. Larger values of σ_{BayTTA} indicate higher uncertainty, whereas smaller values reflect greater consistency across predictions.

2.5 Experimental evaluations and discussion

In this section, we present the results of our comprehensive experimental study conducted using different DL models in combination with TTA and BayTTA methods. Additionally, we elucidate the experimental setup and the optimization parameters employed in our study. Lastly, we present and discuss the main findings derived from our experimental investigation.

2.5.1 Data used in evaluation

We conducted our experiments on three publicly available medical image datasets : skin cancer, breast cancer, and chest X-ray data, which are freely available on the Kaggle and Mendeley data repositories (see also Fig. 3.2). Subsequently, we evaluated the efficacy of the proposed BayTTA method using two well-known gene editing datasets, CRISPOR and GUIDE-seq, both generated through the CRISPR-Cas-9 technology [Sherkatghanad et al., 2023]. We adopted an effective one-hot encoding strategy, originally proposed by [Charlier et al., 2021], to represent sgRNA and DNA sequence pairs as binary images from which off-target effects can be predicted by advanced DL models (see Fig. 2.4).

2.5.1.1 Medical image datasets

Skin cancer dataset is the first group of data taken from Kaggle [1], encompassing images categorized into two distinct classes : Benign and Malignant. This dataset comprises 2637 training images, among which 1440 fall under the Benign category and 1197 under the Malignant category. Additionally, it includes 660 test images, consisting of 360 Benign and 300 Malignant samples.

Breast cancer dataset is the second group of data supplied by Kaggle [2], comprising 8116 training images. This dataset has 4074 samples categorized as Benign and 4042 samples categorized as Malignant. Additionally, the dataset includes 900 test samples, 500 of which are classified as Benign and 400 as Malignant.

1. https ://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign
2. https ://www.kaggle.com/datasets/vuppalaadithyasairam/ultrasound-breast-images-for-breast-cancer

Chest X-ray dataset is a large collection of X-ray images extracted from a publicly available medical image database [3] [Kermany et al., 2018]. The dataset contains 5216 training images, among which 1341 are categorized as Normal and 3875 as Pneumonia samples. Additionally, the test dataset includes 624 images, among which 234 are classified as Normal and 390 as Pneumonia samples.

2.5.1.2 Gene editing datasets

CRISPOR database, organized and maintained by Haeussler et al. [2016] aggregates widely-used public datasets aimed at quantifying on-target guide efficiency and detecting off-target cleavage sites [4]. The dataset we selected from this database (see also [Charlier et al., 2021] comprises $18,211$ black and white training images (each with 8×23 pixel dimension), among which 18,112 are categorized as on-targets and 99 as off-targets. Additionally, the dataset includes 7806 testing images of the same dimension, 7763 of which are classified as on-targets and 43 as off-targets.

GUIDE-seq is one of the pioneering off-target data repositories, derived from the outcomes of the GUIDE-seq technique developed by Tsai et al. [2015]. It serves as an accurate framework for genome-wide identification of off-target effects. The sgRNAs used in GUIDE-seq target the following sites : VEGFA site 1, VEGFA site 2, VEGFA site 3, FANCF, HEK293 site 2, HEK293 site 3, and HEK293 site 4, wherein 28 off-targets with a minimum modification frequency of 0.1 were identified (among 403 potential off-targets). This dataset consists of black and white images with 8×23 pixel dimension. It comprises 309 training images, including 291 on-target and 18 off-target samples. Additionally, it comprises 133 testing images, including 121 on-target and 12 off-target samples.

2.5.2 Implementation details and model settings

To facilitate a comprehensive understanding of the proposed BayTTA method, we provide additional insights into the selected experimental configuration.

Experimental Setup. For the training phase, we used a Compute Canada cluster equipped with NVIDIA Tesla P100 and NVIDIA v100 GPUs (referred to as the Cedar cluster). Additionally, we used a software environment with Python, TensorFlow, and PyTorch stack.

3. https ://easy.dans.knaw.nl/ui/datasets/id/easy-dataset :102451
4. http ://crispor.tefor.net

Table 2.1 – Hyperparameter configuration of the pre-trained deep learning models (VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and Inception-ResNetV2) for the Skin Cancer, Breast Cancer, and Chest X-ray medical image datasets considered in our study.

	Skin Cancer				Breast Cancer				Chest X-ray			
Models	Opt	LR	BS	Epoch	Opt	LR	BS	Epoch	Opt	LR	BS	Epoch
VGG-16	Adam	0.0005	256	124	Adamax	0.005	128	157	Adamax	0.0001	128	156
MobileNetV2	Adam	0.001	256	82	SGD	0.001	16	171	SGD	0.00001	256	100
DenseNet201	Adamax	0.0001	32	138	SGD	0.001	16	75	SGD	0.0001	256	145
ResNet152V2	Adamax	0.0005	128	127	SGD	0.001	16	251	SGD	0.00005	256	200
InceptionResNetV2	Adamax	0.0001	16	256	SGD	0.001	16	249	SGD	0.00001	256	140

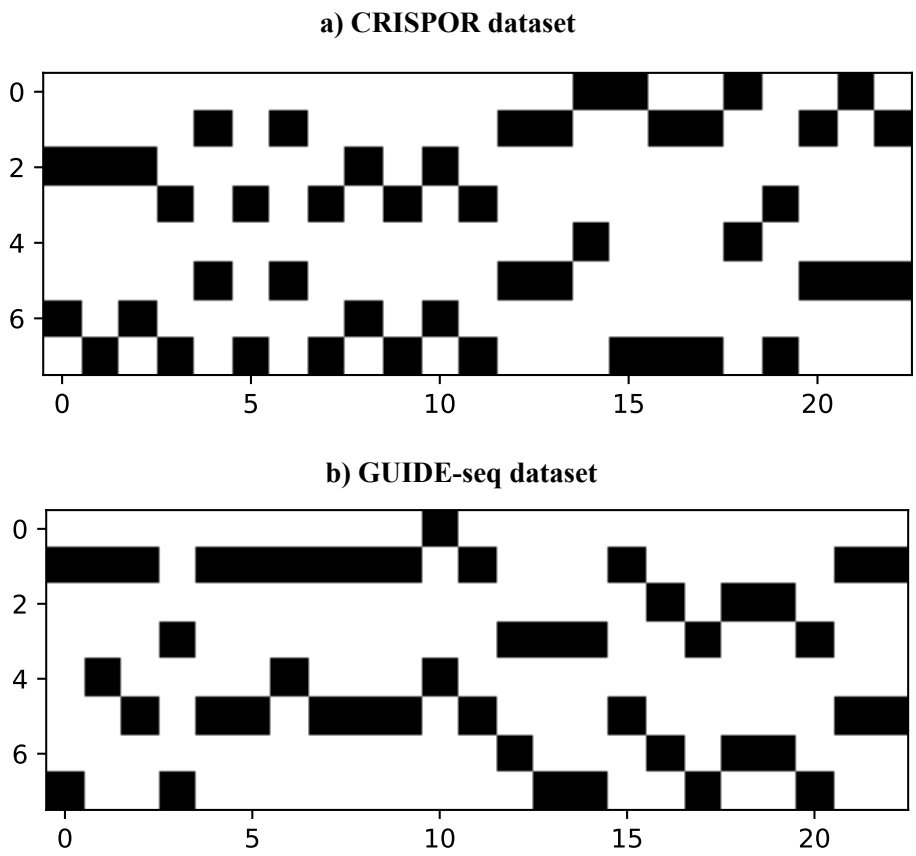

Figure 2.4 – Examples of visualizing CRISPR-Cas9 sgRNA-DNA sequence pairs encoded onto 8×23 matrices, then transformed into black and white images from the (a) CRISPOR and (b) GUIDE-seq gene editing datasets, respectively [Charlier et al., 2021]. These images can be processed by neural networks to predict off-targets generated by CRISPR-Cas9 technology.

Pre-trained CNN models. During the training phase of our first experiment, we employed five well-established convolutional networks (CNNs), namely VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and Inception-ResNetV2, as our feature extraction backbone, initializing weights through pre-training on the ImageNet dataset. The baseline versions of these models were initially used (without applying TTA or BayTTA), followed by their application in combination with TTA and BayTTA. All medical images were resized to the 224×224 pixel size to ensure uniformity in model inputs. The hyperparameters, including the optimizer (Opt), the learning rate (LR), the batch size (BS), and the number of epochs for each of the five pre-trained models and each medical image dataset are summarized in Table 2.1. Additionally, we employed the early stopping technique to handle overfitting [Prechelt, 2002]. During our network training, the data augmentation techniques were implemented using the following parameters : rotation range = 20, width shift range = 0.2, height shift

range = 0.2, shear range = 0.2, zoom range = 0.2, horizontal flip, and vertical flip.

Four pre-trained CNN models, VGG-16, MobileNetV2, DenseNet201, and ResNet152V2, as well as a custom CNN with five layers, comprising two convolutional and three fully connected dense layers, were used to predict off-targets in highly imbalanced gene editing datasets. The models were trained using the RMSprop optimizer, with a learning rate of 0.001 and a batch size of 64. To address potential overfitting issues a set of callback functions was used during training, specifically the ReduceLROnPlateau and EarlyStopping functions.

State-of-the-art DL models intended for medical image analysis. To evaluate the performance of BayTTA, we also compared its performance against TTA and baseline, considering some state-of-the-art DL models recently used in medical image analysis, namely Attention Residual Learning (ARL) [Zhang et al., 2019a], COVID-19 [Ozturk et al., 2020], LoTeNet [Raghavendra Selvan, 2020], IRv2+SA [Datta et al., 2021], and PCXRNet [Feng et al., 2022].

IRv2+SA [Datta et al., 2021] explores the efficacy of soft attention in deep neural architectures. The core objective of this technique is to emphasize the significance of essential features, while mitigating the influence of noise-inducing ones. The ARL model, proposed by Zhang et al. [2019a], is intended for classifying skin lesions in dermoscopy images. The model architecture includes several ARL blocks, a global average pooling layer, and a classification layer. Each ARL block employs a combination of residual learning and an innovative attention-learning mechanism to boost the capacity of discriminative representations. Ozturk et al. [2020] proposed a novel model based on the DarkNet method designed for automated detection of COVID-19 cases using chest X-ray images to deliver precise diagnostic results in the framework of binary and multi-class classifications. PCXRNet [Feng et al., 2022] is an attention-driven convolutional neural network designed for pneumonia diagnosis based on chest X-ray image analysis. PCXRNet incorporates a novel Condensed Attention Module (CDSE) to harness the information within the feature map channels. The LoTeNet (Locally order-less Tensor Network) model [Raghavendra Selvan, 2020] is based on the use of tensor networks, a crucial tool for physicists to analyze complex quantum many-body systems.

2.5.3 Experimental results

2.5.3.1 Results for medical image datasets

We present a thorough experimental assessment of the BayTTA model in two stages.

Table 2.2 – Comparison of the baseline CNN model accuracy (%) ± STD performance against the TTA and BayTTA versions on the skin cancer dataset. The highest accuracy per column is in bold. The asterisk (∗) denotes the highest overall accuracy obtained by the models.

Models	VGG-16	MobileNetV2	DenseNet201	ResNet152V2	InceptionResNetV2
Baseline	84.95 ± 0.40	85.75 ± 1.31	88.28 ± 0.76	83.33 ± 1.75	81.63 ± 1.70
TTA	85.24 ± 0.39	87.39 ± 0.42	88.33 ± 0.58	83.82 ± 0.52	83.01 ± 0.77
BayTTA	**85.50 ± 0.14**	**87.52 ± 0.11**	**89.38 ± 0.17***	**84.04 ± 0.09**	**83.98 ± 0.46**

Table 2.3 – Comparison of the baseline CNN model accuracy (%) ± STD performance against the TTA and BayTTA versions on the breast cancer dataset. The highest accuracy per column is in bold. The asterisk (∗) denotes the highest overall accuracy obtained by the models.

Models	VGG-16	MobileNetV2	DenseNet201	ResNet152V2	InceptionResNetV2
Baseline	88.92 ± 1.70	86.70 ± 0.94	**86.81 ± 1.06**	91.52 ± 1.18	91.25 ± 0.98
TTA	89.64 ± 0.52	87.84 ± 0.81	86.64 ± 0.74	93.64 ± 1.04	**93.13 ± 0.71**
BayTTA	**90.11 ± 0.64**	**88.36 ± 0.34**	86.18 ± 0.59	**93.81 ± 0.99***	92.84 ± 0.69

Table 2.4 – Comparison of the baseline CNN model accuracy (%) ± STD performance against the TTA and BayTTA versions on the chest X-ray dataset. The highest accuracy per column is in bold. The asterisk (∗) denotes the highest overall accuracy obtained by the models.

Models	VGG-16	MobileNetV2	DenseNet201	ResNet152V2	InceptionResNetV2
Baseline	71.02 ± 1.01	61.53 ± 0.73	66.77 ± 1.11	62.45 ± 0.17	63.31 ± 0.57
TTA	71.11 ± 0.78	61.32 ± 0.62	68.20 ± 0.63	62.54 ± 0.19	63.45 ± 0.56
BayTTA	**72.49 ± 0.25***	**62.50 ± 0.27**	**69.98 ± 0.28**	**62.82 ± 0.06**	**64.30 ± 0.21**

First, we evaluate its performance in the case when it was used in combination with five pre-trained CNN models, including VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and InceptionResNetV2, to determine the most suitable backbone model for the skin cancer (Table 2.2), breast cancer (Table 2.3), and chest X-ray (Table 2.4) datasets. In this study, we executed the baseline models three times, computing the mean accuracy and the standard deviation (STD) of accuracy. We applied TTA during the testing phase on the

Table 2.5 – Comparison of state-of-the-art classification models against their TTA and BayTTA counterparts, in terms of accuracy (%) and STD, after their application on the skin cancer, breast cancer, and chest X-ray datasets considered in our study. The highest overall accuracy per dataset is highlighted in bold.

Models	Skin Cancer	Breast Cancer	Chest X-ray
ARL [Zhang et al., 2019a]	86.21 ± 0.42	84.87 ± 0.15	64.58 ± 0.15
COVID-19 [Ozturk et al., 2020]	85.65 ± 0.61	94.92 ± 0.88	84.07 ± 1.41
LoTeNet [Raghavendra Selvan, 2020]	74.89 ± 0.72	72.63 ± 1.24	79.48 ± 0.80
IRv2+SA [Datta et al., 2021]	90.92 ± 0.32	95.84 ± 0.54	$\mathbf{87.76 \pm 1.20}$
PCXRNet [Feng et al., 2022]	79.70 ± 0.21	93.05 ± 0.48	79.15 ± 1.46
COVID-19+TTA	85.70 ± 0.74	94.88 ± 0.34	79.10 ± 1.96
COVID-19+BayTTA	87.17 ± 0.31	95.55 ± 0.31	85.04 ± 1.43
IRv2+SA+TTA	89.74 ± 0.71	94.48 ± 0.67	83.71 ± 1.36
IRv2+SA+BayTTA	$\mathbf{91.07 \pm 0.04}$	$\mathbf{96.66 \pm 0.49}$	$\mathbf{87.76 \pm 1.20}$

baseline models to assess the model accuracy and conducted a comparative analysis with BayTTA, in which the BMA method was used for model averaging. We considered each original image and six of its random augmentations (obtained through rotations) during the testing phase, evaluating the mean accuracy and STD obtained with these random augmentations. The results reported in Tables 2.2 demonstrate that the use of BayTTA allowed us to outperform the baseline and TTA-based models, achieving the highest accuracy and significantly reducing the standard deviation values for the skin cancer dataset. Regarding the breast cancer dataset, BayTTA outperforms the baseline and TTA-based models in accuracy for the three out of five pre-trained CNNs, i.e. VGG-16, MobileNetV2, and ResNet152V2. Nonetheless, the BayTTA-based networks consistently exhibited the lowest STD values, as outlined in Table 2.3. As shown in Table 2.3, the architectural properties of the DenseNet201 and InceptionResNetV2 models, combined with the characteristics of the dataset being analyzed, play a significant role in shaping their performance across different prediction aggregation strategies (Baseline, TTA, and BayTTA). DenseNet201's design, which emphasizes effective feature reuse and stable learning by propagating features directly from earlier layers, ensures consistent performance across aggregation methods, reducing reliance on transformations introduced by TTA or BayTTA. As a result, it demonstrates some small performance gaps, as it already captures critical features from the baseline data. InceptionResNetV2, on the other hand, combines the multi-scale feature extraction of the Inception modules with the gradient-stabilizing skip connections of ResNet, making it inherently adept at

Table 2.6 – Comparison of state-of-the-art classification models against their TTA and BayTTA counterparts, in terms of precision (PR (%)), recall (RE (%)), and F1-score (FS (%)), after their application on the skin cancer dataset considered in our study. The highest overall accuracy per dataset is highlighted in bold.

Models	Precision (PR %)	Recall (RE %)	F1-score (FS %)
ARL [Zhang et al., 2019a]	86.66	85.80	86.09
LoTeNet [Raghavendra Selvan, 2020]	75.38	86.50	66.80
IRv2+SA [Datta et al., 2021]	**89.60**	90.55	90.04
PCXRNet [Feng et al., 2022]	79.98	**99.51**	88.69
COVID-19+TTA	83.81	84.61	84.17
COVID-19+BayTTA	85.80	86.01	85.90
IRv2+SA+TTA	87.22	84.63	90.37
IRv2+SA+BayTTA	88.70	92.11	**90.42**

Table 2.7 – Comparison of state-of-the-art classification models against their TTA and BayTTA counterparts, in terms of precision (PR (%)), recall (RE (%)), and F1-score (FS (%)), after their application on the breast cancer dataset considered in our study. The highest overall accuracy per dataset is highlighted in bold.

Models	Precision (PR %)	Recall (RE %)	F1-score (FS %)
ARL [Zhang et al., 2019a]	89.60	83.15	84.33
LoTeNet [Raghavendra Selvan, 2020]	75.22	62.25	68.13
IRv2+SA [Datta et al., 2021]	**97.68**	95.16	95.52
PCXRNet [Feng et al., 2022]	93.03	73.90	82.37
COVID-19+TTA	94.55	94.05	94.26
COVID-19+BayTTA	94.41	**95.75**	95.05
IRv2+SA+TTA	98.59	91.58	95.19
IRv2+SA+BayTTA	95.99	**96.58**	**96.25**

Table 2.8 – Comparison of state-of-the-art classification models against their TTA and BayTTA counterparts, in terms of precision (PR (%)), recall (RE (%)), and F1-score (FS (%)), after their application on the chest X-ray dataset considered in our study. The highest overall accuracy per dataset is highlighted in bold.

Models	Precision (PR %)	Recall (RE %)	F1-score (FS %)
ARL [Zhang et al., 2019a]	81.46	51.46	41.57
LoTeNet [Raghavendra Selvan, 2020]	77.53	96.41	85.94
IRv2+SA [Datta et al., 2021]	84.24	98.97	**91.03**
PCXRNet [Feng et al., 2022]	75.22	**99.37**	85.63
COVID-19+TTA	**99.17**	75.67	85.61
COVID-19+BayTTA	81.31	98.80	89.53
IRv2+SA+TTA	79.92	**99.57**	88.66
IRv2+SA+BayTTA	84.24	98.97	**91.03**

handling complex transformations, such as rotations. Consequently, TTA outperforms BayTTA and Baseline with InceptionResNetV2, particularly for the breast cancer dataset, which has a large number of high-quality, well-balanced training samples. This allows the model to learn generalizable and robust features. InceptionResNetV2 has a large capacity to capture complex patterns and is inherently strong at handling various transformations, making it well-suited for this task. For well-balanced datasets like breast cancer, where the classification boundaries are clear and the model has ample training data, the need for reducing uncertainty through techniques like BayTTA is less pronounced. In the case of the imbalanced chest X-ray dataset (see Table 2.4), the combination of TTA and BMA (i.e. BayTTA) allowed us to improve the baseline and TTA results in all cases in terms of both accuracy and STD.

Table 2.9 – Comparison of the baseline CNN model accuracy (%) $\pm$ STD performance against their TTA and BayTTA versions on the CRISPOR dataset. The highest accuracy per column is in bold. The asterisk ($*$) denotes the highest overall accuracy obtained by the models.

Models	VGG-16	MobileNetV2	DenseNet201	ResNet152V2	CNN-5 layers
Baseline	**99.41 $\pm$ 0.005**	**99.55 $\pm$0.04**	**99.53 $\pm$ 0.05**	**99.62 $\pm$ 0.08**	99.82 $\pm$ 0.057
TTA	99.37 $\pm$ 0.018	99.46 $\pm$ 0.05	99.46 $\pm$ 0.03	99.53 $\pm$ 0.04	99.77 $\pm$ 0.016
BayTTA	**99.41 $\pm$ 0.011**	**99.55 $\pm$ 0.04**	**99.53 $\pm$ 0.03**	**99.62 $\pm$ 0.02**	**99.84 $\pm$ 0.008***

Table 2.10 – Comparison of the baseline CNN model accuracy (%) $\pm$ STD performance against their TTA and BayTTA versions on the GUIDE-seq dataset. The highest accuracy per column is in bold. The asterisk ($*$) denotes the highest overall accuracy obtained by the models.

Models	VGG-16	MobileNetV2	DenseNet201	ResNet152V2	CNN-5 layers
Baseline	**93.51 $\pm$ 0.98**	90.47 $\pm$ 0.70	90.97 $\pm$ 0.60	87.95 $\pm$ 0.02	94.22 $\pm$ 0.92
TTA	91.61 $\pm$ 1.18	90.54 $\pm$ 0.55	90.83 $\pm$ 0.50	90.65 $\pm$ 0.79	94.45 $\pm$ 0.26
BayTTA	**93.51 $\pm$ 0.67**	**90.97$\pm$ 0.15**	**91.72 $\pm$ 0.30**	**90.98 $\pm$ 0.16**	**94.73 $\pm$ 0.11***

Second, we conducted a comparative study of the TTA-based, BayTTA-based, and baseline state-of-the-art classification models which have been recently used in the literature to process medical image data (see Tables 2.5, 2.6, 2.7 and 2.8). To ensure a thorough performance assessment, we reported the experimental results for the four following key metrics : accuracy, precision, recall, and F1-score. Our analysis revealed that among state-of-the-art models compared, COVID-19 and IRv2+SA provided the highest overall accuracy, surpassing the results of ARL, LoTeNet, and PCXRNet models. Thus, we used the TTA and BayTTA frameworks in combination with these two top-performing models. In this experiment, we considered original images as well as ten random augmentation samples generated for each of them using rotations during the test phase. Our results, reported in Table 2.5, indicate that the use of TTA leads to a decrease in the accuracy of IRv2+SA, but the performance of TTA is notably worse for the imbalanced chest X-ray dataset. Nonetheless, BayTTA improves the models accuracy even in cases of suboptimal performance of TTA, demonstrating its high effectiveness. The model performances in terms of precision, recall, and F1-score (see Table 2.6, 2.7 and 2.8) also demonstrate a much higher robustness of the proposed BayTTA computational framework, compared to the TTA-based and baseline models.

The primary aim of considering BMA in our model is the improvement of the TTA aggregation method and the ability to perform uncertainty estimation. While the additional computational load introduced by BMA is acknowledged, it is important to note that this method's key contribution lies in its ability to quantify the uncertainty of the model predictions. This feature is especially valuable in domains where high confidence in predictions is critical, such as medical image analysis. We emphasize that the computational cost of BMA is a trade-off between the method's running time and significant improvements in predictive performance, including uncertainty estimation capacity provided by our approach. Standard TTA aggregation methods, such as simple averaging or majority voting, are computationally lighter but lack the capability of providing

Table 2.11 – Comparison of state-of-the-art classification models against their TTA and BayTTA counterparts, in terms of accuracy (%) and STD on the CRISPOR and GUIDE-seq gene editing datasets. The highest accuracy per column is highlighted in bold.

Models	CRISPOR	GUIDE-seq
ARL [Zhang et al., 2019a]	94.20 ± 1.19	91.04 ± 0.31
COVID-19 [Ozturk et al., 2020]	99.50 ± 0.20	93.87 ± 0.96
LoTeNet [Raghavendra Selvan, 2020]	98.85 ± 0.43	93.38 ± 0.99
IRv2+SA [Datta et al., 2021]	99.38 ± 0.24	91.22 ± 1.03
PCXRNet [Feng et al., 2022]	95.69 ± 2.51	91.42 ± 0.97
COVID-19+TTA	99.39 ± 0.18	95.07 ± 0.90
COVID-19+BayTTA	$\mathbf{99.66 \pm 0.02}$	$\mathbf{96.56 \pm 0.41}$
IRv2+SA+TTA	95.24 ± 0.57	93.75 ± 0.71
IRv2+SA+BayTTA	99.43 ± 0.38	94.29 ± 0.43

Table 2.12 – Comparison of state-of-the-art classification models against their TTA and BayTTA counterparts, in terms of precision (PR (%)), recall (RE (%)), and F1-score (FS (%)) on the CRISPOR and GUIDE-seq gene editing datasets. The highest accuracy per column is highlighted in bold.

	CRISPOR			GUIDE-seq		
Models	PR	RE	FS	PR	RE	FS
ARL [Zhang et al., 2019a]	88.33	93.66	90.72	33.91	56.81	42.38
COVID-19 [Ozturk et al., 2020]	98.03	98.07	98.04	20.73	33.30	37.93
LoTeNet [Raghavendra Selvan, 2020]	89.74	92.10	94.59	33.30	12.12	17.74
IRv2+SA [Datta et al., 2021]	**99.73**	95.20	96.71	37.86	41.91	37.54
PCXRNet [Feng et al., 2022]	74.13	98.01	84.43	32.90	69.67	43.47
COVID-19+TTA	96.12	97.41	97.55	60.68	60.69	58.78
COVID-19+BayTTA	99.36	**98.08**	**98.70**	**80.79**	**70.03**	**72.64**
IRv2+SA+TTA	88.99	95.91	91.92	30.76	7.76	10.18
IRv2+SA+BayTTA	**99.73**	95.61	97.49	50.01	8.61	14.54

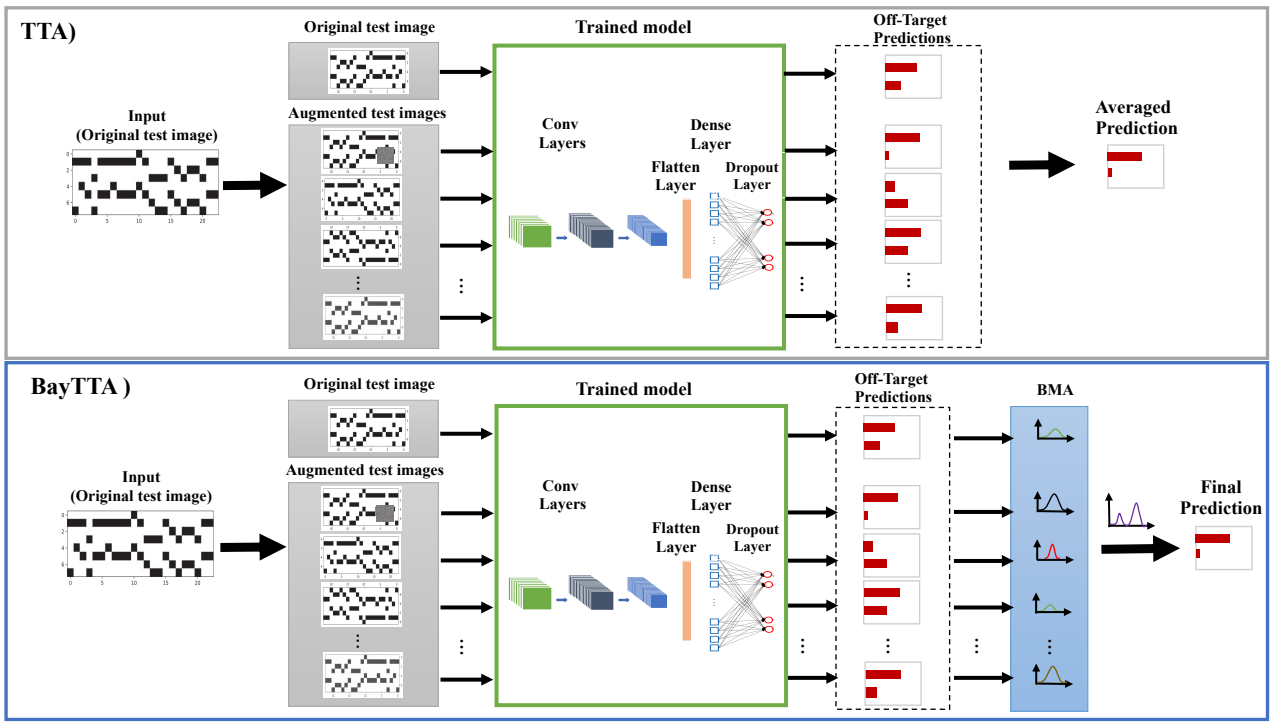

Figure 2.5 – An overview of the proposed BayTTA method for gene editing datasets. During the testing phase : (a) TTA generates predictions from a set of fixed augmented images, and (b) BMA is then applied to combine and aggregate these predictions by treating each unique combination as a distinct candidate model.

insights into model uncertainty. Our method, therefore, not only improves accuracy but also enhances the model's uncertainty awareness, a crucial aspect that is often overlooked by existing frameworks. It leads to more reliable and informed predictions, making our method a valuable tool for high-stakes applications, where understanding uncertainty is as important as achieving accuracy.

2.5.3.2 Results for gene editing datasets

Similarly to medical image data, we first evaluated the performance of the proposed BayTTA method on gene editing datasets using it in combination with four pre-trained CNN models VGG-16, MobileNetV2, DenseNet201, ResNet152V2 as well as a custom CNN model with 5 layers (see Fig. 2.5). A comparison against the TTA-based and baseline models was carried out. The results obtained are reported in Tables 2.9 and 2.10.

We executed each baseline model three times, computing the mean accuracy and standard deviation (STD) of accuracy. During the testing phase, we used original images along with six random augmented samples

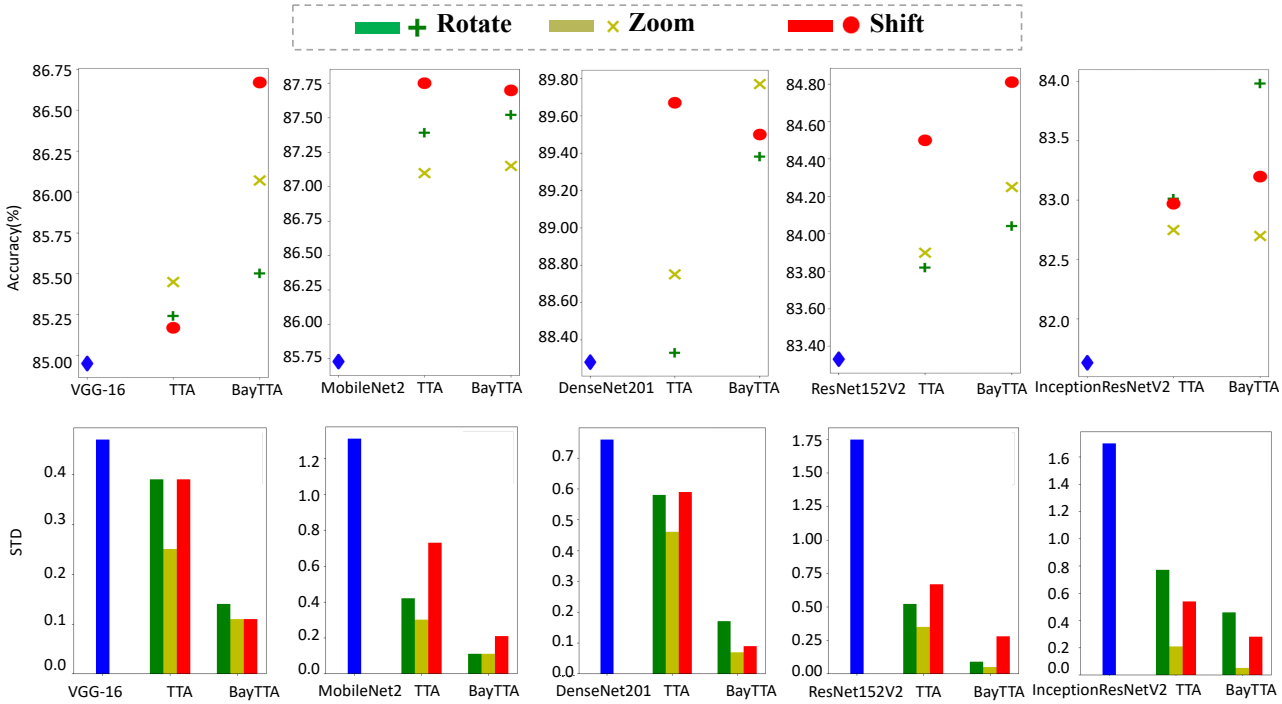

Figure 2.6 – Comparison of the TTA and BayTTA method performance on the skin cancer dataset in terms of accuracy and standard deviation, while considering pre-trained baseline models with rotate, zoom, and shift augmentations.

(generated for each original image through rotations and random erasing). By observing the results obtained for both CRISPOR and GUIDE-seq gene editing datasets, we can conclude that BayTTA demonstrated an enhanced accuracy performance, while consistently yielding lower standard deviation values. In addition to the comparison with pre-trained baseline models, we conducted a comparative analysis between BayTTA and TTA used in combination with state-of-the-art DL classification models whose input was adapted to gene editing datasets. The results obtained in this experiment are presented in Tables 2.11 and 2.12. To conduct our assessment, we used original images with ten randomly created samples (generated for each original image during the testing phase using rotations and random erasing). The results reported in Table 2.11 suggest that the proposed BayTTA computational framework used in combination with the COVID-19 model provided the highest accuracy and the lowest STD values for both CRISPOR and GUIDE-seq datasets. Similar model performances can be observed in Table 2.12, where the obtained precision, recall, and F1-score metric values are reported - the COVID-19+BayTTA model provided the best results overall.

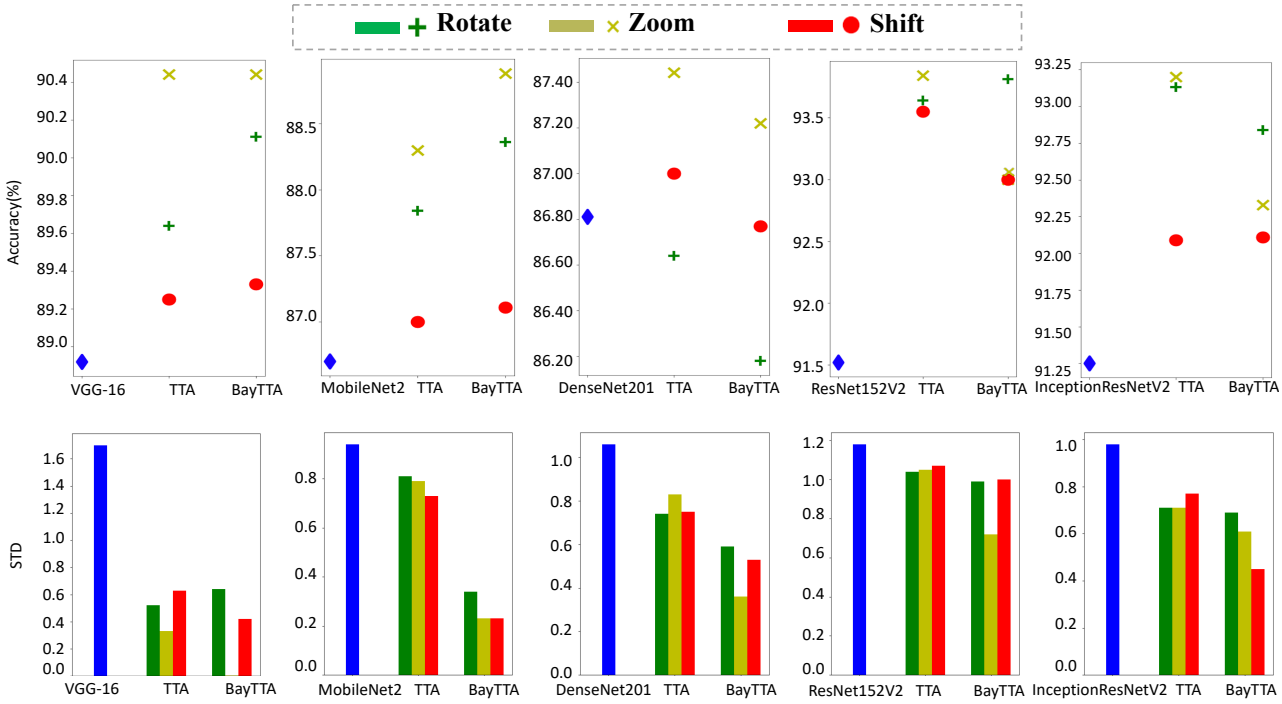

Figure 2.7 – Comparison of the TTA and BayTTA method performance on the breast cancer dataset in terms of accuracy and standard deviation, while considering the pre-trained baseline models with rotate, zoom, and shift augmentations.

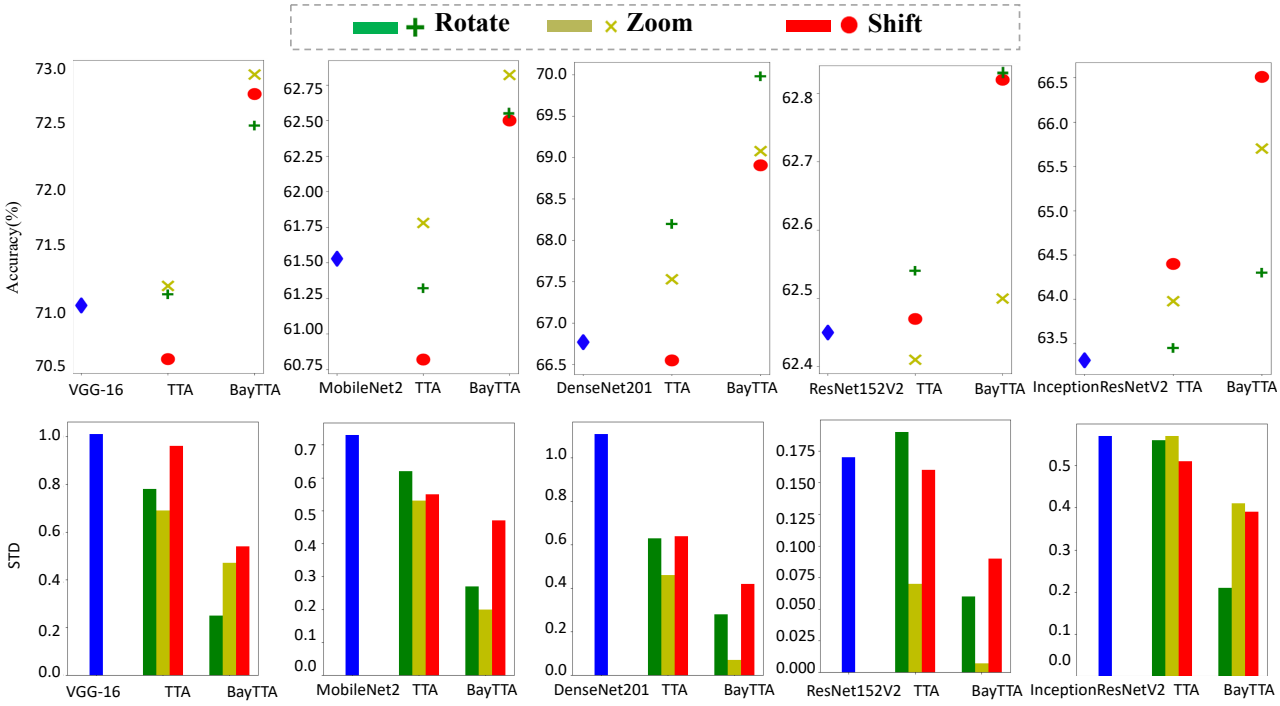

Figure 2.8 – Comparison of the TTA and BayTTA method performance on the chest X-ray dataset in terms of accuracy and standard deviation, while considering the pre-trained baseline models with rotate, zoom, and shift augmentations.

2.5.4 Assessing the impact of different data augmentations and of increasing the number of samples

Furthermore, we conducted a number of supplementary experiments to determine the key factors that may influence the performance of the TTA and BayTTA methods considered in this study. Our analysis was performed on the above-described skin cancer, breast cancer, and chest X-ray datasets to study the influence of sample sizes and different data augmentation techniques and sample sizes.

2.5.5 Evaluation of different data augmentations

To analyze the impact of various data augmentation types and compare the results with traditional TTA, we tested the performance of the proposed BayTTA method on the three following augmentation types : rotation, zoom, and shift. These augmentations were incorporated into the data augmentation process of TTA and BayTTA at the testing phase. During testing, we used six augmented samples in addition to the original image in our experiments including the VGG-16, MobileNetV2, DenseNet201, ResNet152V2, and InceptionResNetV2 pre-trained CNN models.

Our results are graphically represented in Figures 2.6, 2.7, and 2.8, corresponding to the skin cancer, breast cancer, and chest X-ray datasets, respectively. These graphs demonstrate how the accuracy and STD of the model alter as we vary data augmentation types, highlighting which augmentations offer higher accuracy and boost the robustness of CNN models.

We can conclude that BayTTA generally enhances classification performance by increasing the models accuracy and reducing STD after integrating various data augmentations. This outcome was expected, as our proposed method combines predictions from each candidate model based on a constraint on model likelihood obtained from logistic regression. This combination results in model-averaged predictions that account for both the model's uncertainty and accuracy.

One of our important observations is that the choice of transformation plays a substantial role in the model performance, particularly under different prediction aggregation strategies. As shown in Figures 2.7, 2.8, the top-performing transformation set for the standard TTA approach differs from that used by BayTTA. To optimize the selection of transformations for a given aggregation strategy, further investigations are necessary to systematically assess the impact of different transformations on the model performance and uncertainty estimation results. This could involve testing a broader range of transformations, analyzing their

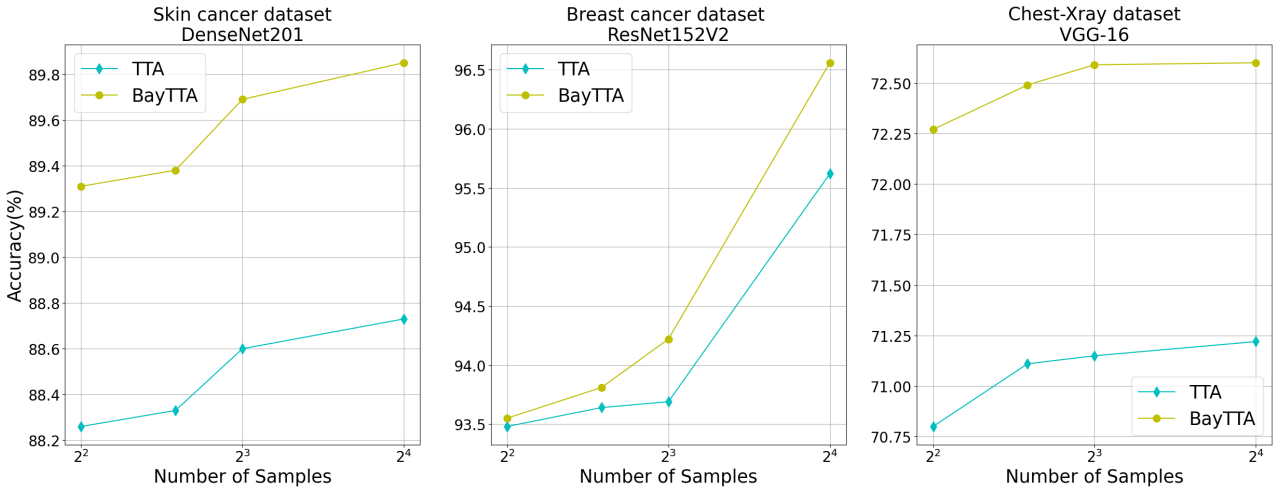

Figure 2.9 – Comparison of the TTA and BayTTA method performance in terms of accuracy with different numbers of samples, while considering the best baseline model from Tables 2.2, 2.3, and 2.4 for each medical image dataset.

influence on model uncertainty, and identifying the most effective combinations for different aggregation methods. We should acknowledge that this is an open problem which we plan to explore more thoroughly in our future work.

2.5.6 Evaluation of increasing the number of samples

We analyzed the impact of increasing the number of augmented samples on the models accuracy during the testing phase. As indicated in Tables 2.2, 2.3 and 2.4, the combination of BMA and TTA contributes to a higher accuracy of DenseNet201 on the skin cancer dataset, of ResNet152V2 on the breast cancer dataset, and of VGG-16 on the chest X-ray dataset, compared to baseline models and TTA. Thus, we examined the effect of increasing the number of augmented samples in these three specific cases.

Figure 2.9 demonstrates that the accuracy of both TTA and BayTTA increases as the number of samples in the test data with diverse transformations grows. Hence, training the model to recognize patterns in their diverse forms generally enhances its ability to make accurate predictions on unseen examples. Obviously, the increase in the amount of augmented data results in a higher accuracy at the expense of a higher computational cost.

2.6 Conclusion

In this study, we investigated how the combination of Test-Time Augmentation (TTA) and Bayesian Model Averaging (BMA) techniques can improve the accuracy of pre-trained and state-of-the-art deep learning models applied in the field of medical image classification. In particular, we focused on the two following modeling aspects : TTA optimization and uncertainty evaluation based on posterior probabilities. Our empirical observations indicate that using BMA as a method for combining model predictions is highly effective for enhancing the performance of traditional TTA. BMA allows one to assign weights to different models based on their posterior probabilities. Therefore, the proposed BayTTA technique can be viewed as a novel and effective methodology that harnesses the combined strengths of TTA and BMA. One of the key strengths of our approach is its capacity to quantify model uncertainty, which means that we not only obtain more accurate predictions, but also gain insight into the level of confidence the model has in each of its predictions. This insight is crucial, especially in applications where incorrect predictions can lead to critical consequences, such as medical image and gene editing analyses.

Although our method is validated within the medical imaging domain, it is not inherently limited to this field. The proposed approach can be extended to general image classification tasks, particularly in scenarios where uncertainty quantification is a crucial factor. We believe that this approach could enhance the performance of classifiers in various domains by providing a more robust and uncertainty-aware aggregation method. However, it is important to note that the effectiveness of the method may depend on several factors, including the size and variability of a given dataset, as well as the distribution of samples across different classes. These considerations may influence the method's generalizability and performance in tasks beyond medical image analysis.

CHAPITRE 3

SIMILARITY-BASED TRANSFER LEARNING WITH DEEP LEARNING NETWORKS FOR ACCURATE CRISPR-CAS9 OFF-TARGET PREDICTION

3.1 Abstract

This chapter is a reproduction of the following article : Jeremy Charlier *, Zeinab Sherkatghanad *, Vladimir Makarenkov, "Similarity-based transfer learning with deep learning networks for accurate CRISPR-Cas9 off-target prediction", PLOS Computational Biology 21.10 (2025) : e1013606.

* These authors contributed equally to this work.

3.1.1 Motivation

Transfer learning has emerged as a powerful tool for enhancing predictive accuracy in complex tasks, particularly in scenarios where data is limited or imbalanced. This study explores the use of similarity-based pre-evaluation as a methodology to identify optimal source datasets for transfer learning, addressing the dual challenge of efficient source-target dataset pairing and off-target prediction in CRISPR-Cas9, while existing transfer learning applications in the field of gene editing often lack a principled method for source dataset selection.

3.1.2 Results

We use cosine, Euclidean, and Manhattan distances to evaluate the similarity between the source and target datasets used in our transfer learning experiments. Four deep learning network architectures, i.e. Multilayer Perceptron (MLP), Convolutional Neural Networks (CNNs), Feedforward Neural Networks (FNNs), and Recurrent Neural Networks (RNNs), and two traditional machine learning models, i.e. Logistic Regression (LR) and Random Forest (RF), were tested and compared in our simulations. The results suggest that similarity scores are reliable indicators for pre-selecting source datasets in CRISPR-Cas9 transfer learning experiments, with cosine distance proving to be a more effective dataset comparison metric than either Euclidean or Manhattan distances. An RNN-GRU, a 5-layer FNN, and two MLP variants provided the best overall prediction results in our simulations. By integrating similarity-based source pre-selection with machine learning outcomes, we propose a dual-layered framework that not only streamlines the transfer learning process but

also significantly improves off-target prediction accuracy [Charlier et al., 2025].

3.1.3 Conclusions

CRISPR-Cas9 is a popular gene-editing technology that allows researchers to modify an organism's genomic DNA at precise locations. Significant research efforts have been focusing on improving its precision and effectiveness, with particular emphasis on minimizing off-target effects. At the same time, transfer learning techniques are becoming increasingly important for addressing deep learning challenges in computational biology, especially in the field of CRISPR-Cas9, where plausible training data availability can be limited. This study investigates the effectiveness of integrating similarity-based analysis with transfer learning for improving CRISPR-Cas9 off-target prediction. Our key contribution consists in an experimental evaluation of three distance metrics, i.e. cosine, Euclidean, and Manhattan distances, along with several traditional machine learning and deep learning models, in the context of knowledge transfer by transfer learning applied to gene editing data. For each considered target dataset our transfer learning framework determines the most suitable source dataset to be used in the model pre-training. The proposed computational framework offers a reliable and systematic method for selecting suitable source data, streamlining the transfer learning process, and improving prediction accuracy.

3.1.4 Availability and Implementation

The code and data used in this study are freely available at : `https://github.com/dagrate/transferlearning_offtargets`.

3.2 Introduction

CRISPR-Cas9 (Clustered Regularly Interspaced Short Palindromic Repeats and the associated protein 9) has become a leading technology for precise and efficient genome (or gene) editing, allowing genetic material to be added, removed, or altered at particular locations of a given genome. Its simplicity, high precision, and versatility across various applications have made it a dominant tool in the field [Jinek et al., 2012, Cho et al., 2013, Cong et al., 2013, Gupta et al., 2014]. The CRISPR-Cas9 genetic engineering system reflects the immune defense mechanism of certain bacteria. Bacteria identify the invading viral DNA and cut out a segment of the virus DNA, known as a protospacer, to insert it into the front of the CRISPR array. Bacteria are armed by the protein Cas9 to produce RNA segments from CRISPR arrays to cut the DNA of the phage virus, and

thus defend themselves from the phage infection return [Barrangou et al., 2007]. In CRISPR-Cas9, single-guide RNA (sgRNA) consists of a crRNA and tracrRNA duplex that guides Cas9 to its Protospacer-Adjacent Motif (PAM) target at the end of the DNA sequence [Shah et al., 2013]. The PAM sequence that follows the protospacer sequence in a viral genome helps Cas9 to distinguish between itself and the enemy. The CRISPR-Cas9 gene editing system covers many areas of human health and welfare [Hsu et al., 2014, Sander and Joung, 2014]. The technology has demonstrated important clinical potential for drug development to treat various human diseases, including cancer [Kang et al., 2017, Liang et al., 2015, Ma et al., 2017], for preventing genetic disorders in plant genetic engineering [Liu et al., 2017, Tang et al., 2016, Raitskin and Patron, 2016], for providing animal disease treatment [Wang et al., 2013, Zarei et al., 2019], as well as for assisting bio-fuel production [Lakhawat et al., 2022, Lee et al., 2021].

A significant challenge in the CRISPR-Cas9 gene editing process is the off-target effect, where the sgRNA targets DNA fragments other than the original DNA fragment aimed, resulting in unwanted cuttings of the DNA sequence [Zhang et al., 2015, Chen et al., 2017]. To ensure safe, reliable, and efficient application of the CRISPR-Cas9 technology, it is essential to develop an accurate method to maximize the on-target efficiency and minimize the number of potential off-targets. There are common scoring methods for off-target prediction, such as CFD score [Doench et al., 2014], MIT score [Haeussler et al., 2016], CHOPCHOP [Montague et al., 2014] and CCTop score [Stemmer et al., 2015] that are based on specific scoring function highlighting mismatch locations. The main disadvantage of the traditional scoring methods is their incapability to improve predictive performance when the number of samples increases as well as their inability to discover relationships between mismatched and matched sites [Lin and Wong, 2018]. Nowadays, effective and feasible solutions to address these issues are provided by data-driven algorithms [Sherkatghanad et al., 2023]. The modern data-driven models that rely on deep learning (DL) show promising results with the growing number of CRISPR-Cas9 data ; they typically outperform existing scoring methods in terms of off-target prediction [Lin and Wong, 2018].

However, deep learning models employ thousands of parameters, requiring a substantial number of samples in CRISPR-Cas9 datasets. To this end, Transfer Learning (TL) has emerged as an effective approach to overcome the problem of insufficient number of samples [Weiss et al., 2016, Lin and Wong, 2018, Charlier et al., 2021]. TL is used to learn properties of large source datasets in order to transfer them to smaller target datasets. TL is employed to improve the prediction accuracy and to avoid data overfitting on small datasets by leveraging the knowledge learned from larger but different datasets.

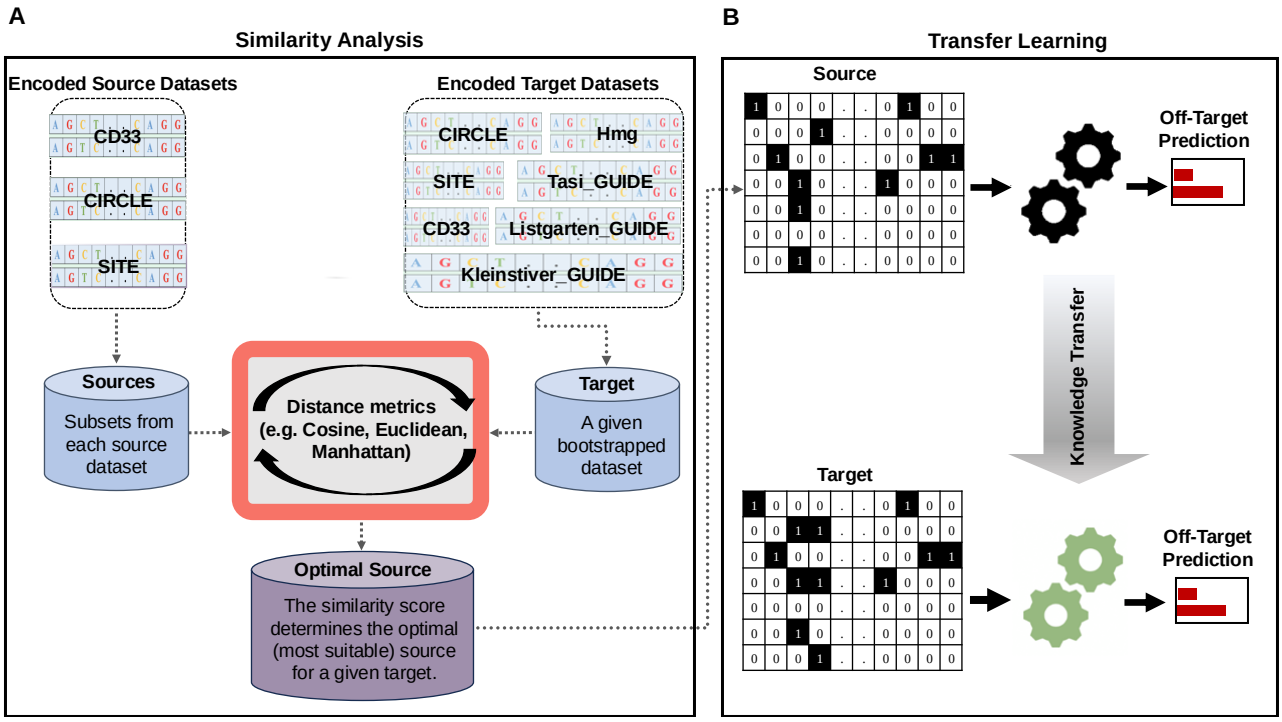

Figure 3.1 – An overview of the proposed framework leveraging data similarity analysis with genome editing transfer learning. (A) Three distance measures : cosine, Euclidean, and Manhattan distances are used to identify the most suitable source dataset, among three benchmark candidate datasets CD33, CIRCLE, and SITE (complete large dataset), for a given target dataset (smaller bootstrapped dataset) ; (B) The framework subsequently transfers the learned model knowledge from the selected optimal source dataset to the target dataset, enhancing the predictive accuracy.

Although some CRISPR-Cas9 benchmark datasets for on- and off-target prediction are currently available [Haeussler et al., 2016], the number of samples they contain is often insufficient to achieve accurate deep learning predictions. In this case, TL can be viewed as a viable alternative approach to the use of traditional machine learning (ML) or more sophisticated DL models which are both prone to overfitting when the data availability is limited. Recently, Lin and Wong [2018] and Charlier et al. [2021] used TL to predict off-targets in small CRISPR-Cas9 datasets. Precisely, they trained the model on a large CRISPOR dataset (18,236 samples) to predict off-targets in a much smaller GUIDE-Seq dataset (430 samples). Elkayam et al. [Elkayam and Orenstein, 2022] introduced the DeepCRISTL model, pretraining it on high-throughput source datasets, including more than 150 000 gRNAs. Then, using TL, they successfully applied DeepCRISTL on target data consisting of much smaller functional or endogenous datasets. Zhang et al. [2020a] proposed the C-RNNCrispr model to

predict sgRNA activity using convolutional and recurrent neural networks (CNNs and RNNs, respectively). After pretraining their model on benchmark data, the authors applied TL by using small-size datasets to fine-tune C-RNNCrispr. Zhang et al. [2021] developed two attention-based CNN models, called CRISPR-ONT and CRISPR-OFFT, for on- and off-target prediction, respectively. They employed TL for small-size cell-line sgRNA specificity prediction. Zhang et al. [2020b] applied TL by using their pre-trained Hybrid CNN-SVR model that was fine-tuned to provide predictions for small sample cell-line datasets. Yaish and Orenstein [2024] also proposed a novel DL network leveraging TL for off-target activity prediction. The authors introduced some innovative metrics and visualization techniques to enhance the understanding of the bulges impact on genome editing. Elkayam et al. [2024] developed the DeepCRISTL model to predict the editing efficiency in a specific cellular context. The authors proposed and compared four TL approaches that are as follows : (a) the full approach that fine-tunes all model weights ; (b) the last-layer approach that fine-tunes only the weights of the last hidden layer and of the output layers ; (c) the no-embedding/convolution approach that fine-tunes all model weights besides those of the embedding and the convolutional layers ; (d) the gradual-learning approach that first fine-tunes the weights of the last hidden and the output layers, and then all other model weights with a smaller learning rate.

We need to highlight that existing transfer learning applications in CRISPR-Cas9 (e.g. DeepCRISTL [Elkayam et al., 2024], C-RNNCrispr [Zhang et al., 2021], CRISPR-ONT and CRISPR-OFFT [Zhang et al., 2021]) often lack a principled method for effective source dataset selection. Thus, our key contribution is in optimizing the transfer learning process through intelligent source selection, and not in inventing new deep learning architectures designed for transfer learning experiments.

Our key contributions are outlined below :

— First, we propose a robust dual-layer framework that integrates similarity-based pre-evaluation with transfer learning for off-target predictions in CRISPR-Cas9 (see Algorithm 1). In contrast to previous studies applying transfer learning to CRISPR-Cas9 datasets, our approach first compares the sgRNA-DNA sequence patterns of the source and target datasets using cosine, Euclidean, or Manhattan distance to identify the optimal source-target pair. The model knowledge is then transferred from a source dataset with a similar sgRNA-DNA sequence pattern to the target dataset.

— Second, we compare the suitability of cosine, Euclidean, and Manhattan distances for transfer learning experiments in CRISPR-Cas9, based on the performance of data-driven DL and ML models.

— Third, we identify the best-performing DL and ML models using reliable performance evaluation metrics to effectively predict off-targets in CRISPR-Cas9.
— Fourth, we demonstrate the effectiveness of our proposed framework by applying it to the analysis of seven popular benchmark datasets : CD33, CIRCLE, SITE, Tasi_GUIDE, Listgarten_GUIDE, Kleinstiver_GUIDE, and Listgarten_Elevation_Hmg.

We need to highlight that existing transfer learning applications in CRISPR-Cas9 (e.g. DeepCRISTL [Elkayam et al., 2024], C-RNNCrispr [Zhang et al., 2021], CRISPR-ONT and CRISPR-OFFT [Zhang et al., 2021]) often lack a principled method for effective source dataset selection. Thus, our key contribution is in optimizing the transfer learning process through intelligent source selection, and not in inventing new deep learning architectures designed for transfer learning.

3.3 Materials and Methods

3.3.1 Datasets

We conducted our off-target prediction experiments on seven well-known public CRISPR-Cas9 datasets :

— **CD33 dataset** was constructed and made available by Doench et al. [2016]. It consists of gRNA-target pairs with only mismatches, comprising 4,853 gRNAs targeting the human coding sequence of CD33. This is one of the rare well-balanced datasets in which the class imbalance ratio is 0.81 (i.e. close to 1).

— **CIRCLE dataset** contains gRNA-target pairs with both mismatches and indels from 10 different guide-RNAs. In our study, we voluntarily modified the encoding process to aggregate mismatches and indels into the same group. This was a deliberate choice on our end in order not to bias the results of our experiments. The dataset contains 7,371 active off-targets, which were validated using the Circularization for In vitro Reporting of CLeavage Effects by sequencing (CIRCLE-seq) technique [Tsai et al., 2017]. Additionally, Lin et al. [2020] used Cas-Offinder [Bae et al., 2014] to identify 577,578 inactive off-target genomic sites in this dataset, including mismatches and indels.
— **SITE dataset** contains 217,733 sgRNA-DNA sequence pairs with 9 guide sequences ; 3,767 of them correspond to active off-targets. The dataset is validated by the SITE-Seq [Cameron et al., 2017, May et al., 2017] biochemical method which employs Cas9 programmed with sgRNAs to recognize cut sites within genomic DNA.
— **Tasi_GUIDE dataset** has been provided by Tsai et al. [2015] based on the cellular method, called GUIDE-seq. This dataset includes a total of 294,534 target sites, with 354 off-target sites containing

Table 3.1 – Seven CRISPR-Cas9 benchmark off-target datasets used in our study. Six of them include gRNA-target pairs with mismatches only, and one of them (CIRCLE, denoted with an asterisk) includes gRNA-target pairs with both mismatches and indels. Minority class samples correspond to active off-target sites (or active off-targets) and Majority class samples correspond to inactive off-target sites.

Dataset	CRISPR-Cas9 technique	gRNAs	Minority class samples	Majority class samples	Class imbalance ratio
CD33	Protein Knockout Detection	65	2,273	2,580	0.8810
CIRCLE*	CIRCLE-Seq	10	7,371	577,578	0.0128
SITE	SITE-Seq	9	3,767	213,966	0.0176
Tasi_GUIDE	GUIDE-Seq	9	354	294,180	0.0012
Listgarten_GUIDE	GUIDE-Seq	22	56	383,463	0.0001
Kleinstiver_GUIDE	GUIDE-Seq	5	54	95,775	0.0005
Listgarten_Elevation_Hmg	PCR, Digenome-Seq and HTGTS	19	52	10,077	0.0052

mismatches.

— **Listgarten_GUIDE** is the fifth dataset used in our experiments, comprising 56 minority class samples and 383,463 majority class samples, validated with the GUIDE-seq technology [Listgarten et al., 2018].

— **Kleinstiver_GUIDE** dataset consists of 54 positive off-target sites and 95,775 inactive off-target sites, validated by the GUIDE-seq technology [Kleinstiver et al., 2015].

— **Listgarten_Elevation_Hmg** dataset, referred to as Hmg, comprises 52 active off-targets among 10,129 potential off-target sites from 19 gRNAs, which was organized and made publicly available by Haeussler et al. [2016].

The dataset's name, the CRISPR-Cas9 technique used, the number of gRNAs, the number of samples in both the minority and majority classes, and the class imbalance ratio for each of these datasets are summarized in Table 3.1. The datasets are publicly available in our GitHub repository at : `https://github.com/dagrate/transferlearning_offtargets`.

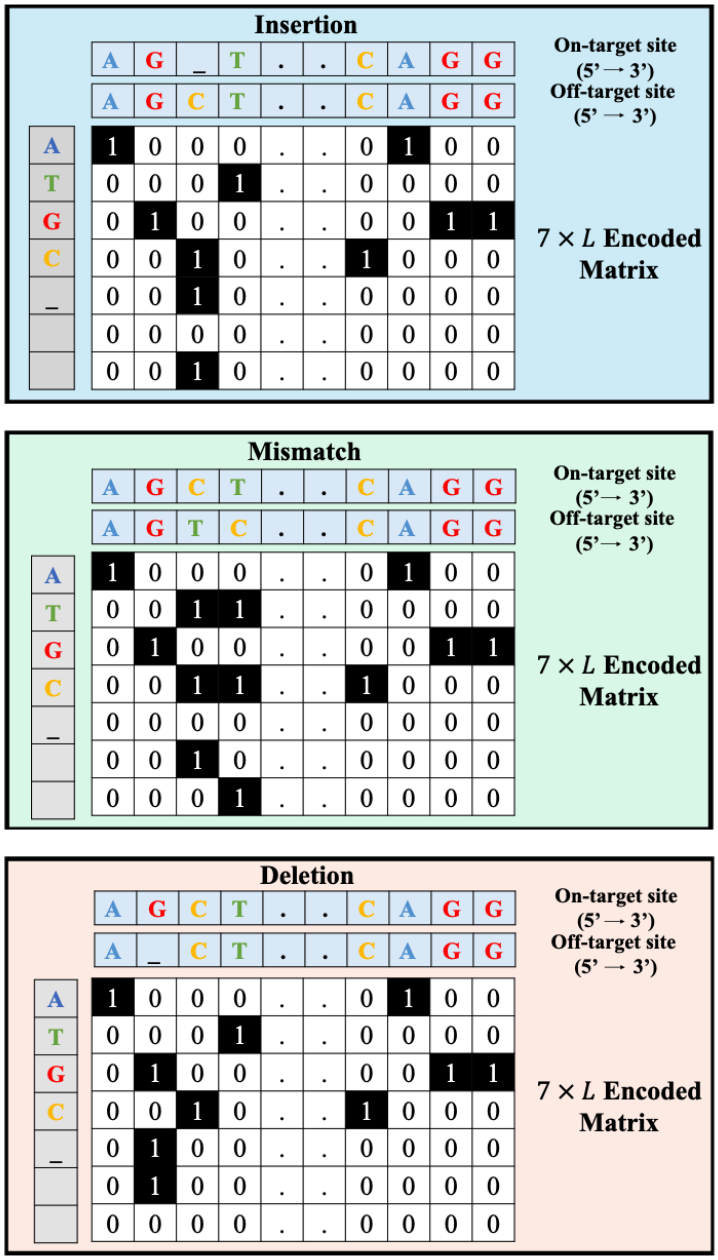

Figure 3.2 – A schematic view of the encoding of an sgRNA-DNA sequence pair, as employed in the study of Lin et al. [2020]. A seven-bit encoding example is illustrated, where the _ symbol indicates the position of DNA or RNA bulges. Each sgRNA-DNA sequence pair is encoded as a fixed-length matrix with seven rows, comprising a five-bit character channel (A, G, C, T, _) and a two-bit direction channel. The five-bit channel encodes the nucleotides at the on- and off-target sites, while the direction channel identifies the locations of mismatches and indels. L denotes the sequence length (L=23 in our study).

3.3.2 Data Encoding

To encode sgRNA-DNA sequences, we adopted the encoded scheme introduced by Lin et al. [2020] that integrates mismatches, insertions, deletions, and matches to preserve the mutual information between on-target and off-target sites. This scheme represents each sgRNA-DNA sequence pair using seven-bit one-hot encoding : a five-bit channel (A, C, G, T, _) and a two-bit direction channel used to indicate the insertion/indel or mismatch directions. Consequently, a 7×23 matrix (where 23 represents the sequence length, including the 3-bp PAM adjacent to the 20 bases) allows for considering three types of base mismatches, missing bases (RNA bulge or insertion), and extra bases (DNA bulge or deletion) in off-target sites. An overview of this encoding technique, with examples including an insertion (RNA bulge), a mismatch, and a deletion (DNA bulge) is presented in Fig. 3.2.

3.3.3 Data Splitting Procedure for Model Training

We specifically selected three datasets, CD33, CIRCLE, and SITE, as potential source datasets due to a large number of positive samples in their minority class (i.e. active off-targets) and the lowest class imbalance among the seven datasets considered, as indicated in Table 3.1. This selection enhances the robustness of our analyses during the training process.

We used a standard train-test split from the scikit-learn [Pedregosa et al., 2011] implementation with shuffling, a ratio of 0.3, and equal stratification of the classes. The stratification was employed to ensure that the class distribution within the training and testing datasets accurately reflects the original class proportions before the train-test split. By maintaining relative class ratios, stratification mitigates biases and enhances the reliability of model evaluation to address the issue of data imbalance [Breiman, 2001].

3.3.4 Model Description

The two following Python libraries were used for model implementation :

— **Scikit-Learn ML and DL models :** Four classification models were implemented using this library : One Hidden Layer Perceptron (MLP1), Two Hidden Layer Perceptron (MLP2), Random Forest (RF) classifier, and Logistic Regression (LR) classifier. These models are well-established ML and DL techniques widely used in practical applications.

— **DL networks with TensorFlow :** We provide details on eight deep neural network models implemented using the Python package TensorFlow. They include a three-layer feedforward neural network (FNN3), a five-layer FNN (FNN5), and a ten-layer FNN (FNN10) ; a three-layer convolutional neural network (CNN3), a five-layer CNN (CNN5), and a ten-layer CNN (CNN10) ; a three-layer Long Short-Term Memory (LSTM) RNN model and a three-layer Gated Recurrent Unit (GRU) RNN model. These network architectures offer flexibility for complex data representations. Figure 3.3 outlines the main features of the FNN, CNN, and RNN networks used in our study. Minor variations exist between different CNN and FNN architectures used in our experiments, based on the number of layers included. However, the architecture presented is consistent across all RNNs evaluated in our study. From top to bottom : (a) Representation of a standard architecture of a CNN used for off-target predictions. Different convolutional layers capture the sgRNA-DNA information of the encoded matrix. A maxpooling is used to downsample the output of the convolutional layers. Fully connected layers are used to perform off-target predictions, (b) Representation of a standard architecture of a FNN used for off-target predictions. Fully dense connected layers are used to perform off-target predictions. The dense connected layers are separated by batch normalization layers or dropout layers to increase the performance of the neural network, (c) Representation of a standard architecture of a RNN used for off-target predictions. The first layer is a Recurrent Neural Network layer, either an LSTM or a GRU in our simulations. Then, three fully connected layers process the sgRNA-DNA information before reaching the last layer performing off-target predictions. A batch normalization is carried out between the second and third dense layers, and a dropout layer is added between the third and the last layers to increase the prediction performance.

3.4 Scikit-Learn models

We hereby present briefly the four Scikit-Learn models we developed : a One-Layer Perceptron, a Two-Layer Perceptron, a Random Forest classifier, and a Logistic Regression classifier. The Scikit-Learn models are used as benchmark in our experiments to evaluate the performance gain (if any) between machine learning models and deep learning models.

One Hidden Layer Perceptron (MLP1) : A one hidden layer perceptron is a three-layer perceptron. It is a type of feedforward artificial neural network with three distinct layers : an input layer, a hidden layer, and an output layer. The input layer receives the 7×L matrices as input data, the hidden layer processes intermediate

representations, and the output layer produces the off-target predictions. The activation function applied to the neurons in the hidden and output layers is a Rectified Linear Unit (ReLU) activation function [Kruse et al., 2022].

Two Hidden Layer Perceptron (MLP2) : A two hidden layer perceptron extends the architecture of a three-layer perceptron by introducing an additional hidden layer. It is thus often referred as a four-layer perceptron. The four-layer perceptron allows for more complex representations of data. Each hidden layer processes intermediate features, enabling the network to capture intricate patterns and non-linear relationships. The activation function applied to the neurons in the hidden and output layers is a Rectified Linear Unit (ReLU) activation function [Kruse et al., 2022].

Random Forest (RF) classifier : An RF classifier is an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting. Each decision tree in the forest is trained on a random subset of the data, and the final prediction is obtained by aggregating the individual tree predictions. [Breiman, 2001].

Logistic Regression (LR) Classifier : An LR classifier models the probability of an instance belonging to a particular class. The logistic function (sigmoid) maps linear combinations of input features to probabilities within the range of values [0, 1]. In this study, we used the standard L2 regularization technique [Friedman et al., 2001].

3.5 Deep Neural Networks with TensorFlow

Here, we describe three types of deep neural networks used in our study : (i) FNNs (Feedforward Neural Networks), (ii) CNNs (Convolutional Neural Networks), and (iii) RNNs (Recurrent Neural Networks). As standard ML models, these networks can be employed for transfer learning in the context of off-target predictions in genome editing.

Feed-forward Neural Networks (FNNs) : We implemented 3 different FNN models for our experiments : a three-layer FNN (FNN3), a five-layer FNN (FNN5), and a ten-layer FNN (FNN10). We used 7×L matrices containing sgRNA-DNA information as input. The networks begin with a first dense layer that collects information from the sgRNA-DNA matrices. Before hyperparameter tuning, the Feedforward Neural Networks

(FNNs) are constructed such that the dense layers are distinct from the dropout and batch normalization layers. Specifically, the three-layer FNN consists of three dense layers, each accompanied by a dropout layer and a batch normalization layer. Similarly, the five-layer and ten-layer FNNs consist of five dense layers and ten dense layers, respectively, followed by a dropout layer and a batch normalization layer. By default, all dense layers use a uniform kernel initializer and a Rectified Linear Unit (ReLU) activation function, except for the final layer, which employs a softmax activation.

Convolutional Neural Networks (CNNs) : We implemented three different Convolutional Neural Network (CNN) models : a three-layer CNN (CNN3), a five-layer CNN (CNN5), and a ten-layer CNN (CNN10). Similar to FNNs, we used as input 7×L matrices containing sgRNA-DNA information. All networks begin with a 2-dimensional convolutional layer. The differences between CNN3, CNN5, and CNN10 emerge at the second layer. Specifically :

- CNN3 : After the initial convolutional layer, CNN3 includes a 2-dimensional max pooling layer, followed by a batch normalization layer, and a dropout layer. Next, a flatten layer is employed, leading to a dense layer, another batch normalization layer, and a dropout layer. Finally, the network reaches the last layer, consisting of a single neuron.
- CNN5 : In addition to the initial convolutional layer, CNN5 introduces another 2-dimensional convolutional layer as the second layer. Following this, a max pooling layer, batch normalization, and dropout layer are applied. The architecture then features two blocks of dense layers, each accompanied by batch normalization and dropout layers. The final dense layer contains a single neuron.
- CNN10 : Up to the flatten layer, CNN10 shares the same architecture as CNN5. However, beyond the flatten layer, CNN10 incorporates seven blocks of dense layers, each accompanied by batch normalization and dropout layers.

By default, all dense layers employ a uniform kernel initializer and a Rectified Linear Unit (ReLU) activation function.

Recurrent Neural Networks (RNNs) : We implemented two different Recurrent Neural Network (RNN) models : a three-layer Long Short-Term Memory (LSTM) model and a three-layer Gated Recurrent Unit (GRU) model. For the LSTM model, the initial block consists of an LSTM layer, followed by a batch normalization layer and a dropout layer. Similarly, for the GRU model, the first block includes a GRU layer, followed by batch normalization and dropout layers. Beyond the first block, both the LSTM and GRU models share an identical architecture. A dense layer is employed, followed once again by a batch normalization layer and a dropout

layer, before reaching the final layer comprising a single neuron. All layers use by default a ReLU activation function.

Table 3.2 – Hyperparameters for machine learning models for CD33 dataset. If a parameter is not mentioned specifically, we used the parameter by default of the model implementation in the $scikit\_learn$ library.

Model	Parameters
MLP1	learning_rate : invscaling, hidden_layer_sizes : 227, early_stopping : False, activation : tanh
MLP2	learning_rate : invscaling, hidden_layer_sizes : (216, 25)), early_stopping : False, activation : tanh
RF	n_estimators : 483, criterion : entropy
LR	solver : newton-cg, penalty : none

Table 3.3 – Hyperparameters for machine learning models for CIRCLE dataset. If a parameter is not mentioned specifically, we used the parameter by default of the model implementation in the scikit_learn library.

Model	Parameters
MLP1	learning_rate : constant, hidden_layer_sizes : 72, early_stopping : False, activation : logistic
MLP2	learning_rate : constant, hidden_layer_sizes : (216, 75)), early_stopping : False, activation : logistic
RF	n_estimators : 183, criterion : entropy
LR	solver : sag, penalty : none

3.5.1 Model Hypertuning

We present hereinafter the methodology used for hypertuning the traditional ML classifiers and DL network models considered in our study.

3.5.1.1 Classifiers Hypertuning

To determine the optimal parameters for each trained classifier, we employed random search with a 3-fold cross-validation on the training set. The 3-fold cross-validation ensures that there is no data leakage when

Table 3.4 – Hyperparameters for machine learning models for SITE dataset. If a parameter is not mentioned specifically, we used the parameter by default of the model implementation in the scikit_learn library.

Model	Parameters
MLP1	learning_rate : invscaling, hidden_layer_sizes : 94, early_stopping : False, activation : logistic
MLP2	learning_rate : constant, hidden_layer_sizes : (250, 25)), early_stopping : False, activation : logistic
RF	n_estimators : 784, criterion : entropy
LR	solver : sag, penalty : l2, C : 0.233572

assessing performance on the test set in our experiments. We utilized the *RandomizedSearchCV* function from the Scikit-Learn library, which implements a random search. Unlike exhaustive grid search, random search explores a subset of hyperparameter values using a fixed number of samples. These values can be specified as lists or sampled from distributions. By doing so, random search efficiently explores a wider range of hyperparameters while minimizing the running time [Bengio et al., 2017]. The hyperparameters used with the CD33, CIRCLE, and SITE datasets are detailed in Tables 3.2, 3.3 and 3.4.

3.5.1.2 Deep Neural Networks Hypertuning

Finding the optimal set of hyperparameters for DL models can require significant computational time and resources [Bengio et al., 2017]. Thus, we decided to use a random search method optimized for DL models offering a good compromise between the computational resources being employed and the optimality of the model's parameters [O'Malley et al., 2019]. The Keras Tuner library offers a simple and efficient framework to fine-tune the deep learning models used in our experiments : FNN3, FNN5, FNN10, CNN3, CNN5, CNN10, LSTM, and GRU. With DL models, we applied a 3-fold cross-validation on the training set applying a similar methodology as that used with traditional scikit-learn classifiers. The number of maximum trials was set to 30. This parameter represents the maximum total number of trials during a hyperparameter search. The hyperparameters for the CD33, CIRCLE, and SITE datasets are detailed in Tables 3.5, 3.6 and 3.7.

Table 3.5 – Hyperparameters for deep neural networks for CD33 dataset. If a parameter is not mentioned specifically, we used the parameter by default of the model implementation in the TensorFlow library.

Model	Parameters
FFN3	unit_layer_1 : 64, unit_layer_2 : 200, unit_layer_3 : 5, unit_dropout_1 : 0.3, is_batch_normalization_1 : True, unit_batch : 64
FFN5	unit_layer_1 : 200, unit_layer_2 : 75, unit_layer_3 : 256, unit_layer_4 : 8, unit_layer_5 : 128, unit_dropout_1 : 0.1, unit_dropout_2 : 0.1, is_batch_normalization_1 : True, is_batch_normalization_2 : True, unit_batch : 64
FFN10	unit_layer_1 : 200, unit_layer_2 : 100, unit_layer_3 : 32, unit_layer_4 : 256, unit_layer_5 : 32, unit_layer_6 : 75, unit_layer_7 : 128, unit_layer_8 : 256, unit_layer_9 : 200, unit_layer_10 : 128, unit_dropout_1 : 0.1, unit_dropout_2 : 0.2, unit_dropout_3 : 0.1, unit_dropout_4 : 0.05, unit_batch : 64, is_batch_normalization_[1, 2, 3, 4, 5, 6] : True
CNN3	unit_layer_1 : 100, unit_layer_2 : 256, activation_layer_1 : relu, activation_layer_2 : relu, activation_layer_3 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.1, is_batch_normalization_1 : True, is_batch_normalization_2 : True, unit_batch : 256
CNN5	unit_layer_1 : 100, unit_layer_2 : 200, unit_layer_3 : 64, unit_layer_4 : 75, activation_layer_1 : relu, activation_layer_2 : tanh, activation_layer_3 : relu, activation_layer_4 : relu, activation_layer_5 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0, unit_batch : 32, is_batch_normalization_[1, 2, 3] : True
CNN10	unit_layer_1 : 256, unit_layer_2 : 128, unit_layer_3 : 100, unit_layer_4 : 256, unit_layer_5 : 64, unit_layer_6 : 32, unit_layer_7 : 8, unit_layer_8 : 64, unit_layer_9 : 75, activation_layer_1 : relu, activation_layer_2 : relu, activation_layer_3 : tanh, activation_layer_4 : tanh, activation_layer_5 : relu, activation_layer_6 : relu, activation_layer_7 : relu, activation_layer_8 : tanh, activation_layer_9 : tanh, activation_layer_10 : sigmoid, unit_dropout_1 : 0.15, unit_dropout_2 : 0.1, unit_dropout_3 : 0.05, unit_dropout_4 : 0.15, unit_dropout_5 : 0.15, unit_dropout_6 : 0.05, is_batch_normalization_[1, 2, 3, 4, 5, 6, 7] : True, unit_batch : 64
LSTM	unit_layer_1 : 200, unit_layer_2 : 256, activation_layer_[1, 2] : relu, activation_layer_3 : sigmoid, unit_dropout_1 : 0.15, unit_dropout_2 : 0, unit_batch : 256, is_batch_normalization_[1, 2] : True
GRU	unit_layer_1 : 256, unit_layer_2 : 64, activation_layer_1 : tanh, activation_layer_2 : tanh, activation_layer_3 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.1, unit_batch : 32, is_batch_normalization_1 : True, is_batch_normalization_2 : True,

Table 3.6 – Hyperparameters for deep neural networks for CIRCLE dataset. If a parameter is not mentioned specifically, we used the parameter by default of the model implementation in the TensorFlow library.

Model	Parameters
FFN3	unit_layer_1 : 200, unit_layer_2 : 8, unit_layer_3 : 2, unit_dropout_1 : 0.3, is_batch_normalization_1 : True, unit_batch : 32
FFN5	unit_layer_1 : 128, unit_layer_2 : 128, unit_layer_3 : 32, unit_layer_4 : 75, unit_layer_5 : 200, unit_dropout_1 : 0.3, unit_dropout_2 : 0.15, is_batch_normalization_[1, 2] : True, unit_batch : 128
FFN10	unit_layer_1 : 200, unit_layer_2 : 200, unit_layer_3 : 8, unit_layer_4 : 64, unit_layer_5 : 200, unit_layer_6 : 75, unit_layer_7 : 200, unit_layer_8 : 32, unit_batch : 256, unit_layer_9 : 32, unit_layer_10 : 200, unit_dropout_1 : 0.3, unit_dropout_2 : 0.2, unit_dropout_3 : 0.1, unit_dropout_4 : 0.3, is_batch_normalization_[1, 2, 3, 4, 5, 6] : True
CNN3	unit_layer_1 : 200, unit_layer_2 : 75, activation_layer_1 : relu, activation_layer_2 : tanh, activation_layer_3 : sigmoid, unit_dropout_1 : 0.15, unit_dropout_2 : 0.1, is_batch_normalization_1 : True, is_batch_normalization_2 : True, unit_batch : 64
CNN5	unit_layer_1 : 256, unit_layer_2 : 256, unit_layer_3 : 32, unit_layer_4 : 75, activation_layer_1 : relu, activation_layer_2 : tanh, activation_layer_3 : relu, activation_layer_4 : tanh, activation_layer_5 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.15, unit_batch : 32, is_batch_normalization_[1, 2, 3] : True,
CNN10	unit_layer_1 : 128, unit_layer_2 : 256, unit_layer_3 : 128, unit_layer_4 : 128, unit_layer_5 : 32, unit_layer_6 : 75, unit_layer_7 : 75, unit_layer_8 : 64, unit_layer_9 : 32, activation_layer_1 : relu, activation_layer_2 : tanh, activation_layer_3 : tanh, activation_layer_4 : relu, activation_layer_5 : relu, activation_layer_6 : tanh, activation_layer_7 : tanh, activation_layer_8 : relu, activation_layer_9 : relu, activation_layer_10 : sigmoid, unit_dropout_1 : 0.05, unit_dropout_2 : 0.1, unit_dropout_3 : 0, unit_dropout_4 : 0, unit_dropout_5 : 0.1, unit_dropout_6 : 0.05, is_batch_normalization_[1, 2, 3, 4, 5, 6] : True, unit_batch : 512
LSTM	unit_layer_1 : 64, unit_layer_2 : 100, activation_layer_1 : tanh, activation_layer_2 : tanh, activation_layer_3 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.25, unit_batch : 32 is_batch_normalization_1 : True, is_batch_normalization_2 : True,
GRU	unit_layer_1 : 32, unit_layer_2 : 128, activation_layer_1 : tanh, activation_layer_2 : tanh, activation_layer_3 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.05, is_batch_normalization_1 : True, is_batch_normalization_2 : True, unit_batch : 64

Table 3.7 – Hyperparameters for deep neural networks for SITE dataset. If a parameter is not mentioned specifically, we used the parameter by default of the model implementation in the TensorFlow library.

Model	Parameters
FFN3	unit_layer_1 : 128, unit_layer_2 : 75, unit_layer_3 : 16, unit_dropout_1 : 0.3, is_batch_normalization_1 : True, unit_batch : 512
FFN5	unit_layer_1 : 8, unit_layer_2 : 32, unit_layer_3 : 128, unit_layer_4 : 200, unit_layer_5 : 32, unit_dropout_1 : 0.05, unit_dropout_2 : 0.2, is_batch_normalization_[1, 2] : True, unit_batch : 512
FFN10	unit_layer_1 : 256, unit_layer_2 : 100, unit_layer_3 : 200, unit_layer_4 : 8, unit_layer_5 : 100, unit_layer_6 : 75, unit_layer_7 : 128, unit_layer_8 : 75, unit_layer_9 : 100, unit_layer_10 : 128, unit_dropout_1 : 0.2, unit_dropout_2 : 0.05, unit_dropout_3 : 0.1, unit_dropout_4 : 0.2, unit_batch : 128, is_batch_normalization_[1, 2, 3, 4, 5, 6] : True,
CNN3	unit_layer_1 : 100, unit_layer_2 : 64, activation_layer_1 : tanh, activation_layer_2 : relu, activation_layer_3 : sigmoid, unit_dropout_1 : 0.15, unit_dropout_2 : 0, unit_batch : 64, is_batch_normalization_1 : True, is_batch_normalization_2 : True,
CNN5	unit_layer_1 : 100, unit_layer_2 : 100, unit_layer_3 : 32, unit_layer_4 : 16, activation_layer_1 : relu, activation_layer_2 : tanh, activation_layer_3 : relu, activation_layer_4 : tanh, activation_layer_5 : sigmoid, unit_dropout_1 : 0.15, unit_dropout_2 : 0.05, is_batch_normalization_[1, 2, 3] : True, unit_batch : 32
CNN10	unit_layer_1 : 200, unit_layer_2 : 128, unit_layer_3 : 128, unit_layer_4 : 100, unit_layer_5 : 75, unit_layer_6 : 8, unit_layer_7 : 64, unit_layer_8 : 64, unit_layer_9 : 8, activation_layer_1 : relu, activation_layer_2 : relu, activation_layer_3 : tanh, activation_layer_4 : tanh, activation_layer_5 : relu, activation_layer_6 : tanh, activation_layer_7 : relu, activation_layer_8 : tanh, activation_layer_9 : tanh, activation_layer_10 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.05, unit_dropout_3 : 0, unit_dropout_4 : 0.05, unit_dropout_5 : 0, unit_dropout_6 : 0.15, is_batch_normalization_[1, 2, 3, 4, 5, 6, 7] : True, unit_batch : 128
LSTM	unit_layer_1 : 256, unit_layer_2 : 256, activation_layer_1 : relu, activation_layer_2 : relu, activation_layer_3 : sigmoid, unit_dropout_1 : 0.1, unit_dropout_2 : 0.25, unit_batch : 64, is_batch_normalization_1 : True, is_batch_normalization_2 : True,
GRU	unit_layer_1 : 32, unit_layer_2 : 200, activation_layer_1 : tanh, activation_layer_2 : relu, activation_layer_3 : sigmoid, unit_dropout_1 : 0.05, unit_dropout_2 : 0.15, unit_batch : 128, is_batch_normalization_1 : True, is_batch_normalization_2 : True,

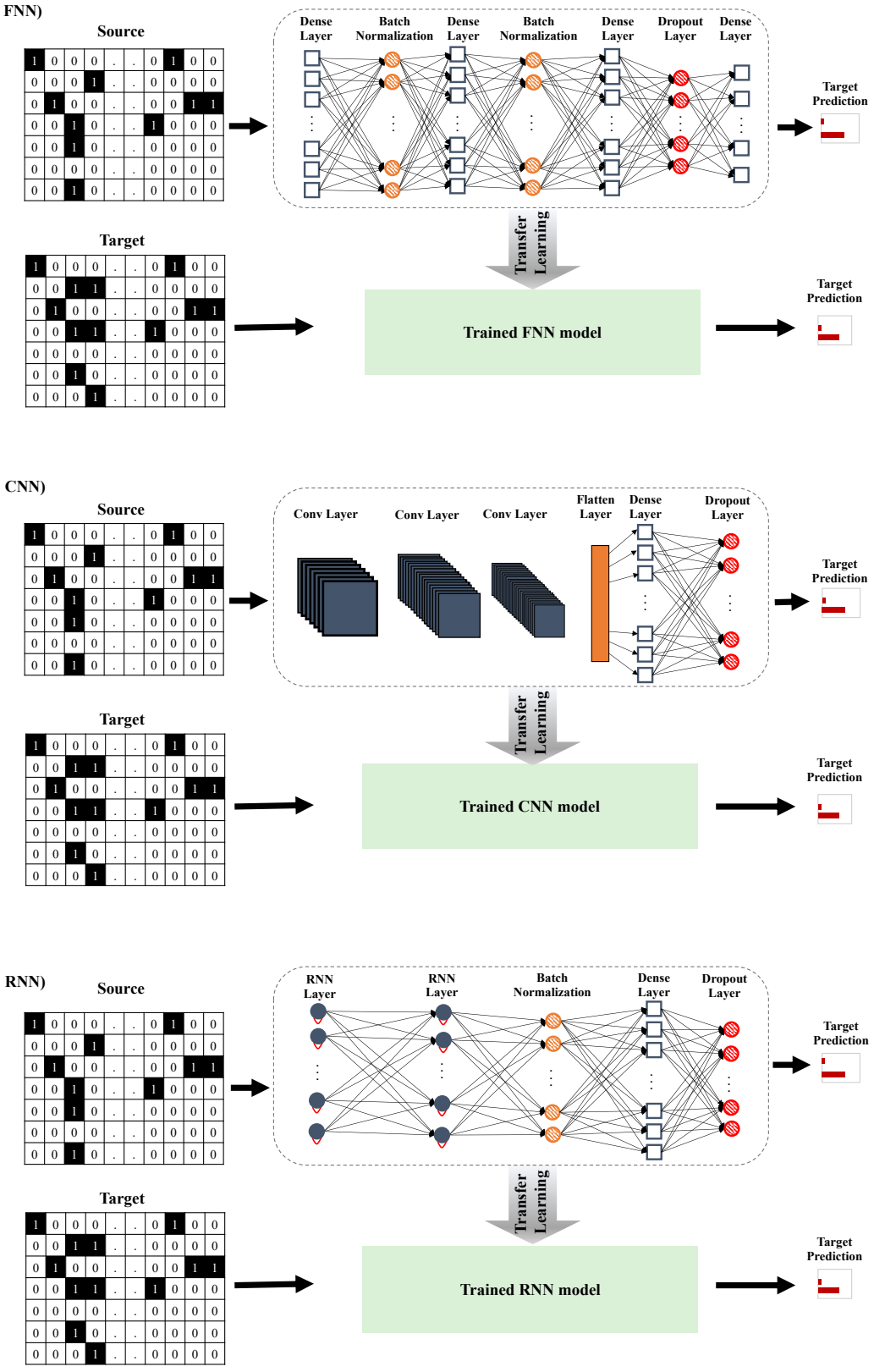

Figure 3.3 – Representation of transfer learning for FNNs, CNNs, and RNNs.

3.5.2 Neural Network Overfit Monitoring

Both traditional ML classifiers and DL networks require overfit monitoring during their training [Bengio et al., 2017]. In our experiments, we employed two essential callbacks to mitigate overfitting in our deep learning models. First, we used the Reduce Learning Rate on Plateau callback dynamic approach that automatically adjusts the learning rate during training based on the model's performance. If the validation loss reaches a plateau (i.e., stops improving), the learning rate is reduced, allowing the model to converge more effectively. Second, we used the Early Stopping callbacks with a patience of 8 epochs and a minimum $delta$ parameter of 0.02 on the validation loss. This callback function monitors the model's performance during training. If the validation loss fails to improve significantly (i.e. it is less than the specified $delta$) for a certain number of consecutive epochs (determined by the patience value), the training is halted early to prevent overfitting. These combined strategies help ensure that our DL models generalize well to unseen data [Abadi et al., 2015].

3.5.3 Transfer Learning Based On Distance Evaluation

In this section, we delve into the background and detailed explanation of the proposed approach for similarity-based transfer learning off-target prediction in CRISPR-Cas9. A notable concern in transfer learning is the risk of negative transfer [Wang et al., 2019e], which arises when the source dataset is inappropriately selected. In this case, a model pre-trained on a larger but dissimilar dataset may perform worse than a model trained from scratch with randomly initialized weights. This issue emphasizes the importance of quantifying the similarity between source and target datasets to ensure the success of transfer learning. To address this challenge for off-target CRISPR-Cas9 data, we evaluate the similarity between the two involved datasets (i.e. a potential source and the given target datasets) using three different metrics : cosine similarity (here, we use its distance form), and Euclidean and Manhattan distances :

$$d_{cosine}(\mathbf{a}, \mathbf{b}) = 1 - \frac{\sum_{i=1}^{K} a_i b_i}{\sqrt{\sum_{i=1}^{K} a_i^2}\sqrt{\sum_{i=1}^{K} b_i^2}}, \tag{3.1}$$

$$d_{Euclidean}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{K} (a_i - b_i)^2}, \tag{3.2}$$

$$d_{Manhattan}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{K} |a_i - b_i|. \tag{3.3}$$

Here, $\mathbf{a}$ and $\mathbf{b}$ are vectors of length $K = 7L$ (i.e. L is the sequence length, which is equal to 23 in our study) representing the encoded sgRNA-DNA sequence pairs in the source and target datasets, respectively. Each encoded matrix of size $7 \times L$ (see Fig. 3.2) is flattened into a vector of length $7L$ in order to calculate the distance between sgRNA-DNA sequence pairs from the source and target datasets. It is worth noting that the number of rows, i.e. 7, in the matrix corresponds to the number of bits used to encode a given sgRNA-DNA sequence pair, including a five-bit character channel (i.e. A, G, C, T, and _) and a two-bit direction channel (to indicate if they appear in the sgRNA or DNA sequence). The five-bit channel encodes the presence-absence of the four nucleotides and insertions/deletions, whereas the direction channel identifies the location of the mismatched nucleotides or insertions/deletions, if any (see Fig. 3.2 for three examples of such a sequence encoding). We should highlight that 7 is the minimum number of rows one can use to encode the presence-absence of the A, G, C, T, and _ characters in two sequences, including their location information, without any information loss.

It is important to note that cosine, Euclidean, and Manhattan distances compared in our study have different strengths and disadvantages. For example, cosine distance is sensitive to direction in high-dimensional sparse spaces. It is used when the magnitude of the vectors is not important. This is the case of binary data such as our sgRNA-DNA sequence pairs encoded as $7L$ binary vectors. With cosine distance the difference in values is not fully taken into account, but this is not a disadvantage in our settings since this difference can be either 0 or 1 with our sequence encoding. Theoretically, cosine distance should work better in cases when the encoded sgRNA-DNA sequence pairs have more matching nucleotides, than in cases with frequent insertions, deletions, and mismatches, since better matching sequences would lead to more sparse binary spaces. Euclidean distance is certainly the most natural distance choice as it is directly computed from the cartesian coordinates of the points using the Pythagorean theorem. Euclidean distance is sensitive to magnitude but, as specified above, this is not of importance in our binary settings. This distance usually works well with low-dimensional data. Manhattan distance is known for its robustness to outliers. It is less intuitive than Euclidean distance but works well with discrete and binary components as it considers the veritable path that can be taken within values of those components. Thus, in our settings, we could expect that Manhattan distance treats equally well binary encoded matching nucleotides, insertions, deletions, and mismatches.

Clearly, the most intuitive way of computing the distance between the source and the target datasets is the following :

— A. For each element in the source dataset, calculate the minimum distance between it and all elements

in the target dataset.

— B. For each element in the target dataset, calculate the minimum distance between it and all elements in the source dataset.

— C. Take an average of all these minimum distances. This average can play the role of the distance between the source and the target data. This distance could then be used to determine the most appropriate source dataset for a given target dataset to carry out transfer learning.

Such an exhaustive approach would work in practice when both the source and target datasets are small ($< 50,000$ elements each). However, real-world CRISPR-Cas 9 datasets often contain hundreds of thousands elements (see Table 3.1) each of which must be encoded in a numerical vector format beforehand in order to perform machine learning experiments. For example, the execution of the above-mentioned exhaustive approach applied to the CIRCLE (used as source data) and SITE (used as target data) datasets would require several weeks of intensive computation on a modern PC computer. Moreover, in many practical situations the target datasets is so small that deep learning experiments, which usually necessitate a huge amount of data, cannot be performed on it (e.g. see [Lin and Wong, 2018, Charlier et al., 2021] for examples of such off-target datasets used in CRISPR-Cas9). Finally, to perform our Monte Carlo simulations to determine the most appropriate distance measure as well as the most suitable ML and DL models in the context of CRISPR-Cas9 off-target transfer learning, we need several hundred real-world datasets of realistic size.

Thus, we decided to perform our Monte Carlo simulations with bootstrap replicates of the considered benchmark target datasets. A bootstrap replicate of a given target dataset per similarity experiment was generated and the average simulation results were then reported. The size of each bootstrapped target dataset was 250, while the number of iterations (i.e. comparisons of each target element with the source elements used to assess the distance between the source and target datasets) was set to 5,000. As we determined experimentally, with this number of iterations (denoted by n_{itr} in Algorithm 1 below), the average distance between the source and target datasets found by the random search converges towards the distance provided by an exhaustive search algorithm. In the large majority of cases, this number of iterations was sufficient to achieve two-digit precision after the decimal point during the distance calculation. This allowed us to obtain reliable results without having to run the computations for several days. Regarding the size of 250 of the bootstrapped targets, it was selected to see how the proposed methodology would work with some high-quality CRISPR-Cas9 datasets of realistically small size. For example, in a recent Nature Communication paper, Ham et al. [Ham et al., 2023] used the TevSpCas9 dataset with 279 samples as well as the SpCas9 dataset with 303 samples to conduct their transfer learning experiments with a novel machine

learning architecture (crisprHAL) meant to improve sgRNA activity prediction. Furthermore, we made sure that the sample ratios between the majority and minority classes in each bootstrapped target datasets were equivalent to those in the complete target dataset.

Algorithm 1 outlines the key steps of our similarity-based transfer learning approach. The algorithm takes as input N potential source datasets (representing labeled data), denoted as $\mathcal{D}_S := \{D_1, ..., D_N\}$, a given bootstrapped target dataset (representing unlabeled data), denoted as $D_{\mathcal{T}}$, and a distance measure d (cosine, Euclidean, or Manhattan distance, in our case).

The set of the encoded vector representations of the source dataset D_i ($i = 1, ..., N$) is denoted as $\{\mathbf{X^i}\}$, and of the bootstrapped target dataset $D_{\mathcal{T}_b}$ as $\{\mathbf{X^t}\}$, each of them of dimension $7L$.

During the first phase of Algorithm 1, refereed to as Similarity Analysis phase, we systematically evaluate the distance between each potential source dataset $\mathcal{D}_i \in D_N$ and the bootstrapped target dataset $D_{\mathcal{T}_b}$ to determine the most suitable source dataset for a given target. Cosine, Euclidean, or Manhattan distance between the following vectors is then computed : (1) $\mathbf{a}_m$ - a vector of length $7L$ representing the m^{th} encoded sgRNA-DNA sequence pair in the bootstrapped target dataset $D_{\mathcal{T}_b}$, where $m \in \{1, \ldots, |D_{\mathcal{T}_b}|\}$ and (2) $\mathbf{b}_n$ - a randomly selected vector of length $7L$ representing the n^{th} encoded sgRNA-DNA sequence pair in the i^{th} source dataset $\mathcal{D}_i$, where $n \in \{1, \ldots, |\mathcal{D}_i|\}$. For each dataset $\mathcal{D}_i \in \mathcal{D}_{\mathcal{S}}$, we iterate over every element in $D_{\mathcal{T}_b}$ computing the current distance value, $d_{current}$, between each element in $D_{\mathcal{T}_b}$ (i.e. $\mathbf{a}_m$ vectors) and a random subset of elements in $\mathcal{D}_i$ (i.e. $\mathbf{b}_n$ vectors). The number of elements in this random subset equals n_{itr}. Thus, a unique subset of the current source dataset $\mathcal{D}_i$ is generated through random sampling with replacement. If the computed current distance $d_{current}$ is smaller than the previously stored minimum distance $dist_{min}$, we update $dist_{min}$ to $d_{current}$. This process is repeated over n_{itr} iterations, ensuring that $dist_{min}$ consistently represents a close match between the target element and the source dataset $\mathcal{D}_i$. We then construct the vector $\mathbf{d}$ of dimension $|D_{\mathcal{T}_b}|$ that includes the minimum distance values for all samples in $D_{\mathcal{T}_b}$. The mean of this vector is used to determine the optimal source candidate dataset for transfer learning (i.e. the set that exhibits the highest similarity with a given target).

The second phase of Algorithm 1, referred to as Transfer Learning phase, implements the transfer learning process using the optimal source dataset $D_{S_{Opt}}$ (selected at Phase 1).

Algorithme 3.1 Similarity-Based Transfer Learning for CRISPR-Cas9 Off-Target Prediction

Entrée: Set D_S of N potential source datasets $\{D_1, ..., D_N\}$ (labeled off-target data)

Entrée: Bootstrapped target dataset $D_{\mathcal{T}_b}$ (unlabeled off-target data)

Entrée: Distance measure d (i.e. cosine, Euclidean, or Manhattan distance)

Sortie: Off-target predictions by similarity-based transfer learning

$\mathcal{D}_{\mathcal{S}} := \{\mathcal{D}_1, ..., \mathcal{D}_N\}, \mathcal{D}_i =$ Encode$(D_i) = \{\mathbf{X^i}\}, i = 1, ...N$ and $D_{\mathcal{T}_b}$ = Bootstrap(Encode($D_{\mathcal{T}}$))

Phase 1 - Similarity Analysis - Selecting Optimal Source Dataset

$Dist_{Opt} \leftarrow \infty$ ▷ Initialize optimal (minimum) distance between source and target

$D_{S_{Opt}} = \emptyset$ ▷ Initialize optimal source dataset for transfer learning

pour $\mathcal{D}_i \in D_N$ **faire**

  pour $m \leftarrow 1, ..., |D_{\mathcal{T}_b}|$ **faire**

    $dist_{min} \leftarrow \infty$ ▷ Initialize the minimum distance

    $\mathbf{a}_m = \mathbf{X^t}_{m,:}$ ▷ Extract the m^{th} sample of the target dataset

    pour $iteration \leftarrow 1, ..., n_{itr}$ **faire**

      $n =$ randint$[1, |\mathcal{D}_i|)$ ▷ Randomly sample an index from the source dataset

      $\mathbf{b}_n = \mathbf{X^i}_{n,:}$ ▷ Extract the n^{th} sample of the i^{th} source dataset

      $d_{current} \leftarrow d(\mathbf{a}_m, \mathbf{b}_n)$ ▷ Using Eqs. (1)-(3)

      si $d_{current} < dist_{min}$ **alors**

        $dist_{min} \leftarrow d_{current}$

      fin si

    fin pour

    $\mathbf{d}_m \leftarrow dist_{min}$ ▷ Store the minimum distance for the m^{th} sample in vector $\mathbf{d}$

  fin pour

  si $\overline{\mathbf{d}} < Dist_{Opt}$ **alors** ▷ $\overline{\mathbf{d}}$ is the mean of the minimum distance vector $\mathbf{d}$

    $Dist_{Opt} \leftarrow \overline{\mathbf{d}}$ ▷ Update the optimal distance between source and target

    $D_{S_{Opt}} \leftarrow \mathcal{D}_i$ ▷ Update the optimal source dataset

  fin si

fin pour

Phase 2 - Transfer Learning

$\mathcal{M}_{\mathcal{S}} \leftarrow$ Train model using the selected source dataset $D_{S_{Opt}}$, $w_{\mathcal{S}} \leftarrow$ Save the trained model weights

$\mathcal{M}_{\mathcal{T}_S} \leftarrow$ Apply transfer learning using target data by loading weights $w_{\mathcal{S}}$

Perform off-target predictions using $\mathcal{M}_{\mathcal{T}_S}$

Table 3.8 – Minority and majority class distribution, and class imbalance ratio for bootstrapped target datasets, with sample size of 250, used in our experiments.

Dataset	Minority Class Samples	Majority Class Samples	Class Imbalance Ratio
CD33_BS	117	133	0.879
CIRCLE_BS	3	247	0.012
SITE_BS	4	246	0.016
Tasi_GUIDE_BS	2	248	0.008
Listgarten_GUIDE_BS	2	248	0.008
Kleinstiver_GUIDE_BS	2	248	0.008
Hmg_BS	3	247	0.012

3.6 Results and Discussion

In this section, we present the results of our simulation study that addresses two core objectives : (i) evaluating the effectiveness of transfer learning in improving off-target predictions in CRISPR-Cas9, and (ii) developing a methodology for pre-assessing the success of transfer learning predictions through a similarity-based analysis of the source and target data. The flowchart of our approach is illustrated in Fig 3.1.

3.6.1 Similarity Analysis

Similarity analysis evaluates the closeness of a given target dataset to a potential source dataset. This analysis is crucial for determining the appropriateness of employing transfer learning for the data at hand. In our study, cosine, Euclidean, and Manhattan distances were used to quantify the degree of similarity between datasets. Each of these metrics has its own strengths and weaknesses. Thus, cosine distance is preferable for high-dimensional and text data, Euclidean distance provides an intuitive measure of similarity for normalized data, whereas Manhattan distance is beneficial for datasets encompassing outliers or non-linear relationships [Shirkhorshidi et al., 2015].

To conduct our simulation study, we selected the CD33, CIRCLE, and SITE datasets as candidate sources datasets, as they offer a sufficient number of minority class samples (i.e. off-targets), thereby increasing the robustness of our approach. In our simulations, the size of the subset of the source dataset compared to the given target dataset was set to 5,000 (i.e. $n_{itr} = 5,000$ in Algorithm 1), whereas the size of the bootstrapped

target datasets was set to 250 (the class imbalance ratio in the bootstrapped datasets was equivalent to that of the corresponding complete dataset ; see Table 3.8). We observed that the distance estimations usually converged when the number of iterations, i.e. n_{itr}, was between 4000 and 5000.

Table 3.9 reports the average estimated similarities between the three source datasets (CD33, CIRCLE, and SITE) and the seven bootstrapped target datasets (CD33_BS, CIRCLE_BS, SITE_BS, Tasi_GUIDE_BS, Listgarten_GUIDE_BS, Kleinstiver_GUIDE_BS, and Hmg_BS) calculated using cosine, Euclidean, and Manhattan metrics. Each similarity estimate appearing in Table 3.9 was computed as $1-NormalizedAverageDistance$ between the selected source and bootstrapped target dataset using cosine, Euclidean, or Manhattan distance. The exact procedure used in our experiments to compute the similarity values is as follows :

1. A bootstrapped target dataset of size 250 was generated for each of the 7 (complete) benchmark off-target datasets considered in our work. The class imbalance ratio of the corresponding complete benchmark dataset was preserved in bootstrapped data (e.g. see Table 3.8) ;
2. A 7×7 distance matrix, **Dist**, containing pairwise distances between 7 complete and 7 bootstrapped datasets was computed using Algorithm 1 ;
3. Steps 1 and 2 above were repeated 5 times to create 5 replicates of the distance matrix **Dist** ;
4. The average 7×7 distance matrix **Dist_av** was computed using these replicates ;
5. The average similarity matrix **S** was computed from this average distance matrix using the Min-Max normalization : $s(i,j) = 1 - \dfrac{Dist\_av(i,j) - min(Dist\_av)}{max(Dist\_av) - min(Dist\_av)}$,

   where $1 \leq i, j \leq 7$, and $max(Dist\_av)$ and $min(Dist\_av)$ are, respectively, the minimum and maximum values of the distance matrix **Dist_av**. Obviously, higher similarity values are associated with lower distances.

In addition, Figure 3.4 presents the corresponding bar plot diagrams for each of the three considered distance measures.

The results presented in Table 3.9 and Figure 3.4 demonstrate that the cosine metric provides the highest overall similarity values between source and target datasets, compared to Manhattan and Euclidean distances, whereas Euclidean distance corresponds to the lowest similarities. However, Manhattan and Euclidean metrics provide the largest differences between the similarities corresponding to the recommended and non-recommended source datasets. This means that Euclidean and Manhattan distances, being sensitive to absolute differences, better highlight stark dissimilarity between datasets. However, our findings suggest

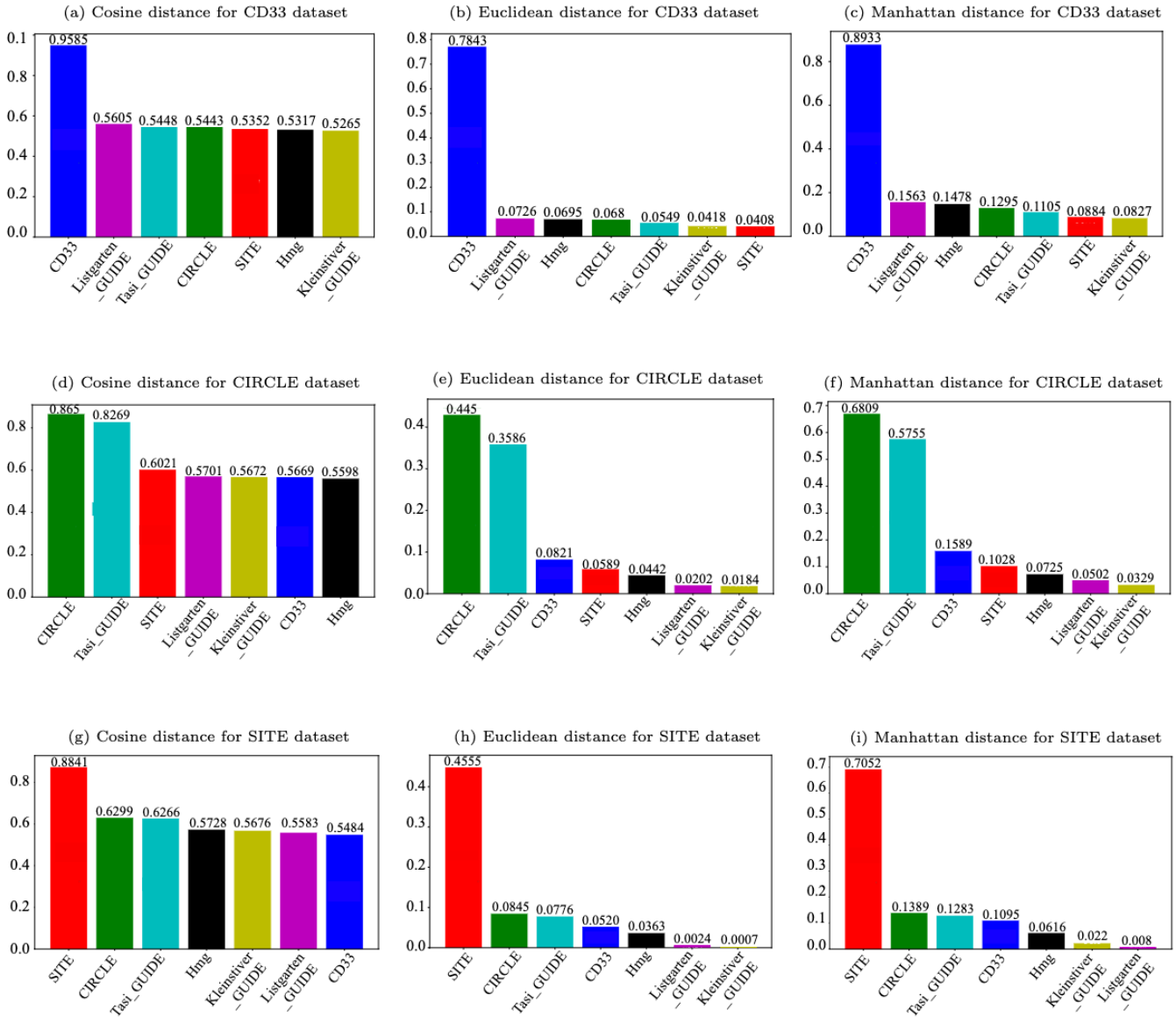

Figure 3.4 – Bar plot representation of the average estimated similarities. Similarities between the three source datasets (CD33, CIRCLE, and SITE) and the seven bootstrapped target datasets (CD33_BS, CIRCLE_BS, SITE_BS, Tasi_GUIDE_BS, Listgarten_GUIDE_BS, Kleinstiver_GUIDE_BS, and Hmg_BS) were assessed based on cosine, Euclidean, and Manhattan metrics.

Table 3.9 – Average Estimated Similarities (1 - Normalized Average Distances) between the three source datasets (CD33, CIRCLE, and SITE) and the seven bootstrapped target datasets (CD33_BS, CIRCLE_BS, SITE_BS, Tasi_GUIDE_BS, Listgarten_GUIDE_BS, Kleinstiver_GUIDE_BS, and Hmg_BS) calculated using cosine, Euclidean, and Manhattan distances. Each similarity value is computed by subtracting from 1 the corresponding normalized average distance estimate. Similarity values corresponding to the most suitable source-target dataset pairs are highlighted in bold.

Target data	Metric	CD33	CIRCLE	SITE
CD33_BS	Cosine	**0.9585**	0.5669	0.5484
	Euclidean	**0.7843**	0.0821	0.0520
	Manhattan	**0.8933**	0.1589	0.1095
CIRCLE_BS	Cosine	0.5443	**0.8650**	0.6299
	Euclidean	0.0680	**0.4450**	0.0845
	Manhattan	0.1295	**0.6809**	0.1389
SITE_BS	Cosine	0.5352	0.6021	**0.8841**
	Euclidean	0.0408	0.0589	**0.4550**
	Manhattan	0.0884	0.10280	**0.7052**
Tasi_ GUIDE_BS	Cosine	0.5448	**0.8269**	0.6266
	Euclidean	0.0549	**0.3586**	0.0776
	Manhattan	0.1105	**0.5755**	0.1283
Listgarten_ GUIDE_BS	Cosine	0.5605	**0.5701**	0.5583
	Euclidean	**0.0726**	0.0202	0.0024
	Manhattan	**0.1563**	0.0502	0.0080
Kleinstiver_ GUIDE_BS	Cosine	0.5265	**0.5672**	**0.5676**
	Euclidean	**0.0418**	0.0184	0.0007
	Manhattan	**0.0827**	0.0329	0.0220
Hmg_BS	Cosine	0.5317	0.5598	**0.5728**
	Euclidean	**0.0695**	0.0442	0.0363
	Manhattan	**0.1478**	0.0725	0.0616

that overall magnitude is less important for transferability than feature direction, which is captured by cosine distance. Such a result should be related to the binary nature of the encoded sgRNA-DNA sequence

pairs since in our settings even the vectors with different locations of matches, mismatches, and indels are sparse enough to have at least 50% of matching 0 values, thus leading to the lowest cosine similarity values that are slightly higher than 0.5. In the case of Euclidean and Manhattan distances, the lowest normalized distance values can be close to 1, leading to the corresponding similarity values that are slightly higher than 0. Obviously, the maximum similarity value is limited by 1 (in case of a perfect sequence match) for all three metrics considered.

As expected, the most appropriate source datasets for the bootstrapped target datasets CD33_BS, CIRCLE_BS, and SITE_BS were their complete source counterparts CD33, CIRCLE, and SITE, respectively. The corresponding cosine similarity values for these source-target pairs were 0.9585, 0.8650, and 0.8841, respectively.

Interestingly, for the other target datasets, i.e. Tasi_GUIDE_BS, Listgarten_GUIDE_BS, Kleinstiver_GUIDE_BS, and Hmg_BS, the choice of the most suitable source dataset depends on the selected distance/similarity measure. For example, for Tasi_GUIDE_BS, the CIRCLE dataset stands out as the most suitable source, achieving the highest similarity across all three distance metrics (cosine, Euclidean, and Manhattan). However, for Listgarten_GUIDE_BS, the CIRCLE dataset is the most suitable source according to cosine similarity, but both Manhattan and Euclidean metrics indicate CD33 as the most suitable source dataset for it. In the case of Kleinstiver_GUIDE_BS, the CIRCLE and SITE datasets provide a slightly better performance compared to CD33 according to cosine similarity. However, according to both Euclidean and Manhattan metrics, CD33 shows the highest similarity with this target dataset. For Hmg_BS, the obtained results reveal that the SITE dataset demonstrates the highest similarity with it using cosine similarity, whereas the CD33 dataset is designated as the most suitable source for it according to Euclidean and Manhattan metrics. Clearly, the choice of the most appropriate source dataset depends on the specific distance/similarity measure being employed as the results provided by Euclidean and Manhattan metrics are usually well aligned, but don't always correspond to those yielded by the cosine metric.

3.6.2 Evaluation Metrics

The performance of the proposed model was assessed using several standard evaluation metrics, as detailed below :

— **AUC_ROC (Area Under the Receiver Operating Characteristic Curve) :** This metric evaluates the model's ability to distinguish between positive and negative classes. It represents the probability that

the classifier will assign a higher score to a randomly chosen positive instance than to a randomly chosen negative instance.

— **Precision** is defined as the proportion of true positive predictions among all predicted positives :

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{3.4}$$

where TP denotes the number of true positives and FP the number of false positives.

— **Recall**, which is also known as sensitivity or true positive rate, assesses the proportion of correctly identified positive samples :

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{3.5}$$

where FN is the number of false negatives.

— **F1-score** is the harmonic mean of precision and recall, offering a balance between them :

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.6}$$

— **Brier score** is defined as the mean squared difference between the predicted probability and its binary outcome :

$$\text{Brier score} = \frac{1}{N} \sum_{i=1}^{N} (p_i - o_i)^2, \tag{3.7}$$

where p_i is the predicted probability of sample i, $o_i \in \{0, 1\}$ is the observed outcome of i, and N is the total number of samples. Lower values of the Brier score correspond to better calibrated probabilistic predictions.

— **Accuracy** is the proportion of correctly classified samples among their total number :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.8}$$

where TN is the number of true negatives.

3.6.3 Assessing the Impact of Similarity Analysis in Transfer Learning

In this section, we evaluate the impact of similarity analysis in transfer learning with CRISPR-Cas9 off-target data. Thus, we assess the reliability of the similarity scores reported in Table 3.9 for ML and DL-based off-target predictions. This evaluation is structured around three distinct scenarios, where machine learning models are trained on one of the three source datasets (CD33, CIRCLE, and SITE) and applied, via transfer learning, to different variants of seven bootstrapped datasets constructed as outlined in Table 3.8. For each scenario, we report the average results for 10 DL models : FNN models with 3, 5, and 10 layers ; CNN models

with 3, 5, and 10 layers; LSTM and GRU models with 4 layers; MLP models with 1 and 2 layers; as well as for traditional RF and LR classifiers (see Supplementary Information for further details on the models considered).

In the first scenario, the CD33 dataset served as the source dataset for transfer learning. The Receiver Operating Characteristic (ROC) curves (see Fig 3.5A) and Precision-Recall (PR) curves (see Fig 3.9A) are presented for various models trained on the CD33 dataset and tested on its bootstrapped counterpart, CD33_BS. To help evaluate the models performance, the AUC-ROC values are displayed in descending order within these figures. Furthermore, Fig 3.6 and Fig S2 present the ROC and PR curves, respectively, for all considered ML and DL models using the CD33 dataset as source and six other bootstrapped datasets as targets. Additionally, Table 3.10 reports the values of the six selected evaluation metrics, including AUC ROC, Precision, Recall, F1-score, Brier score, and Accuracy, obtained using the CD33 dataset as source for all bootstrapped targets. Target datasets exhibiting the highest similarity to the CD33 dataset are marked with an asterisk (based on the similarity scores reported in Table 3.9). When the CD33_BS dataset served as the transfer learning target, GRU and MLP1 achieved a superior AUC-ROC performance compared to other models, with MLP1 providing the highest AUC-ROC score of 0.9863, closely followed by GRU at 0.9839. Both GRU and MLP1 consistently outperformed the other competing models across all evaluation metrics. Among the bootstrapped datasets, Listgarten_GUIDE_BS, Kleinstiver_GUIDE_BS, and Hmg_BS showed the highest similarity with CD33 according to Euclidean and Manhattan metrics (see Table 3.9). When Listgarten_GUIDE_BS was used as target, MLP1 achieved the highest AUC ROC (0.9629) and Precision (0.6828) results. When Kleinstiver_GUIDE_BS was used as target, MLP1 and MLP2 outperformed all other models across all metrics. Moreover, MLP2 consistently provided the best results across all metrics for the Hmg_BS target dataset, with the highest AUC-ROC, precision, and F1-score values, and the lowest Brier score.

In the second scenario, the CIRCLE dataset was used as the source dataset in our transfer learning experiments. The corresponding ROC curves (see Fig 3.5B) and PR curves (see Fig 3.9B) are presented for the considered ML and DL models trained on the entire CIRCLE dataset and evaluated on the CIRCLE_BS dataset. Additionally, Fig. 3.7 and Fig. 3.11 show the ROC and PR curves for all considered models obtained using the CIRCLE dataset as source and the six other bootstrapped datasets as targets. The detailed quantitative results are reported in Table 3.11. When the CIRCLE_BS dataset was used as target, MLP2 performed exceptionally well with an AUC ROC score of 0.9959, a precision of 0.9564, a recall of 0.9231, an F1-score of 0.9600, a Brier score of 0.0021, and an accuracy of 0.9980. When the Tasi_GUIDE_BS dataset was used as target, MLP1 provided the best

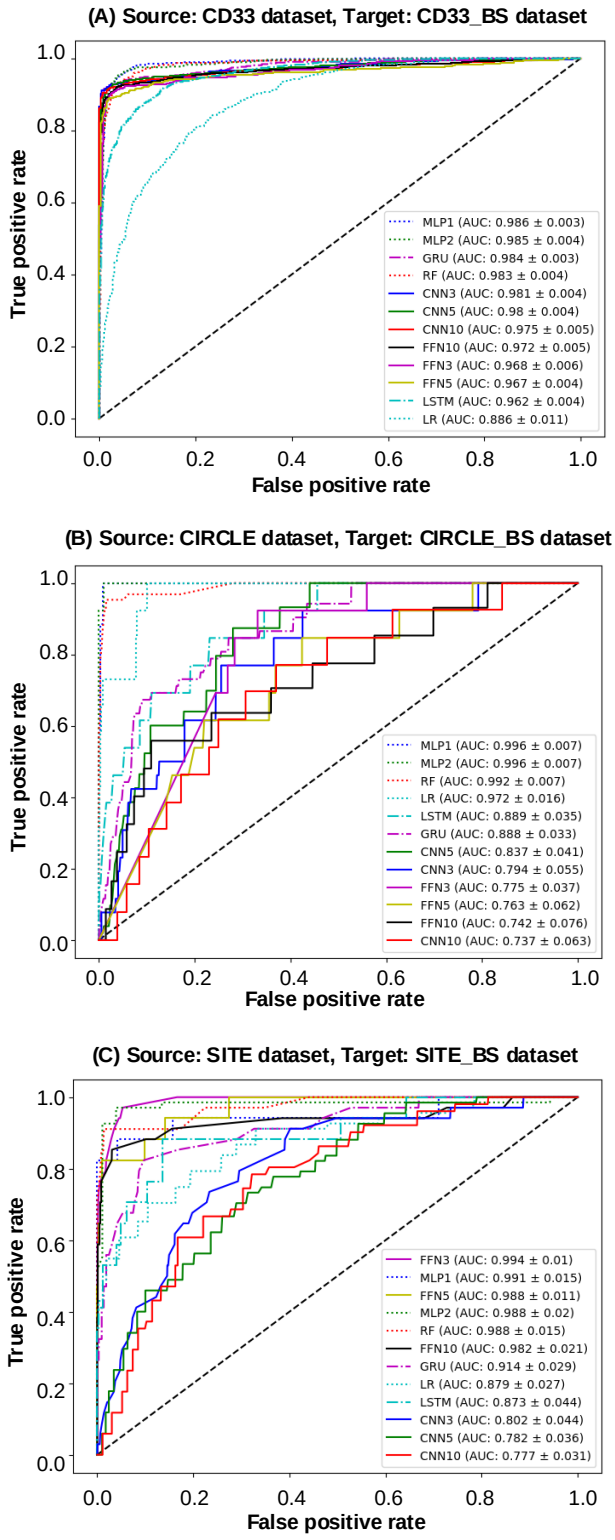

Figure 3.5 – ROC curves for model evaluation. ROC curves for models trained on : (A) CD33 dataset, (B) CIRCLE dataset, and (C) SITE dataset, used as sources, and evaluated on their respective bootstrapped targets. The AUC ROC values for each model are displayed in descending order within each figure.

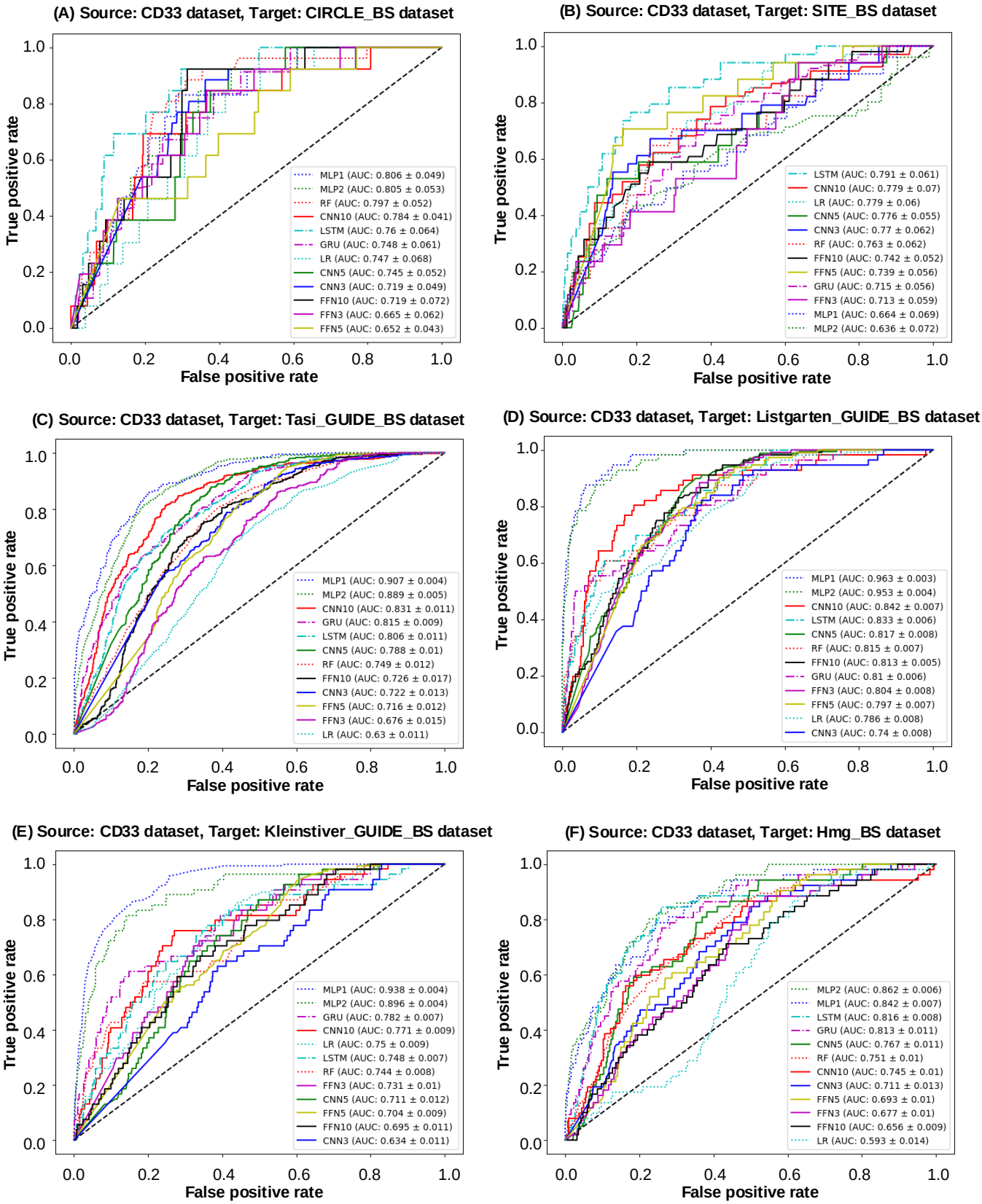

Figure 3.6 – ROC curves for model evaluation. ROC curves for models trained on the CD33 dataset, used as source, and six bootstrapped datasets

Table 3.10 – Performance metrics for classification models obtained using the CD33 dataset.

Target	Metric	FNN3	FNN5	FNN10	CNN3	CNN5	CNN10	LSTM	GRU	**MLP1**	**MLP2**	RF	LR
CD33_BS*	AUC ROC	0.9680	0.9671	0.9720	0.9814	0.9799	0.9754	0.9618	**0.9839**	**0.9863**	0.9851	0.9829	0.8860
	Precision	0.9740	0.9745	0.9785	0.9859	0.9842	0.9823	0.9608	**0.9867**	**0.9832**	0.9803	0.9822	0.8621
	Recall	0.8782	0.8889	0.9145	0.8931	0.8814	0.9060	0.9299	**0.9402**	**0.9583**	0.9487	0.9701	0.8120
	F1 Score	0.9278	0.9265	0.9369	0.9425	0.9348	0.9464	0.8838	**0.9392**	**0.9522**	0.9541	0.9390	0.7925
	Brier Score	0.0570	0.0578	0.0493	0.0481	0.0499	0.0440	0.0874	**0.0464**	**0.0361**	0.0371	0.0550	0.1489
	Accuracy	0.9360	0.9340	0.9423	0.9490	0.9425	0.9520	0.8856	**0.9430**	**0.9550**	0.9573	0.9410	0.8010
CIRCLE_BS	AUC ROC	0.6650	0.6518	0.7189	0.7193	0.7447	0.7840	0.7599	0.7477	0.8056	0.8051	0.7974	0.7467
	Precision	0.0576	0.0445	0.0671	0.0533	0.0667	0.1225	0.1052	0.1040	0.1342	0.1165	0.0951	0.0799
	Recall	0.5384	0.4615	0.5385	0.9231	0.8462	0.8462	0.8846	0.4403	0.0000	0.0170	0.3846	0.5637
	F1 Score	0.1356	0.0752	0.1057	0.0721	0.0965	0.0799	0.1365	0.12345	0.0000	0.0282	0.1266	0.0947
	Brier Score	0.1703	0.2736	0.2115	0.5926	0.3815	0.4633	0.2392	0.1378	0.0275	0.0282	0.1069	0.1754
	Accuracy	0.8185	0.7050	0.7630	0.3820	0.5880	0.4930	0.7070	0.8390	0.9722	0.9705	0.8574	0.7231
SITE_BS	AUC ROC	0.6759	0.7158	0.7256	0.7221	0.7880	0.8313	0.8062	0.8151	0.9067	0.8887	0.7487	0.6295
	Precision	0.4387	0.4925	0.4969	0.5187	0.5899	0.6840	0.6508	0.6683	0.8363	0.7926	0.5625	0.4138
	Recall	0.5156	0.6884	0.6884	0.9632	0.9292	0.9575	0.7652	0.6686	0.1218	0.1132	0.5542	0.7651
	F1 Score	0.5098	0.5810	0.6152	0.6066	0.6735	0.6377	0.6526	0.6413	0.2166	0.2022	0.5602	0.5670
	Brier Score	0.3338	0.3282	0.2767	0.4232	0.2974	0.3464	0.2415	0.2190	0.3038	0.3083	0.2081	0.2543
	Accuracy	0.6500	0.6495	0.6960	0.5590	0.6820	0.6160	0.7120	0.7360	0.6890	0.6843	0.6920	0.5870
Tasi_ GUIDE_BS	AUC ROC	0.7131	0.7387	0.7419	0.7705	0.7762	0.7793	0.7905	0.7152	0.6638	0.6360	0.7632	0.7787
	Precision	0.1156	0.0922	0.1944	0.1127	0.1551	0.1667	0.2499	0.1122	0.1443	0.1227	0.1739	0.1693
	Recall	0.2941	0.7647	0.5686	0.7005	0.5882	0.7487	0.7941	0.3835	0.0000	0.0000	0.3529	0.7647
	F1 Score	0.1136	0.1425	0.1400	0.0987	0.1299	0.1163	0.1730	0.1444	0.0000	0.0000	0.1277	0.1197
	Brier Score	0.1461	0.2801	0.2045	0.4021	0.2470	0.3335	0.2125	0.1349	0.0336	0.0335	0.1459	0.2321
	Accuracy	0.8440	0.6870	0.7633	0.5700	0.7320	0.6162	0.7425	0.8457	0.9663	0.9663	0.8360	0.6175
Listgarten_ GUIDE_BS*	AUC ROC	0.8037	0.7974	0.8134	0.7396	0.8169	0.8424	0.8327	0.8098	**0.9629**	0.9530	0.8148	0.7860
	Precision	0.1416	0.1443	0.1741	0.1119	0.1616	0.2399	0.2560	0.3021	**0.6828**	0.6027	0.2962	0.2431
	Recall	0.5893	0.7679	0.7143	0.9286	0.9107	0.9107	0.6964	0.5714	**0.1429**	0.1071	0.6071	0.7857
	F1 Score	0.2662	0.2266	0.2459	0.1578	0.2214	0.1882	0.2400	0.2832	**0.2500**	0.1935	0.2798	0.1832
	Brier Score	0.1739	0.2736	0.2199	0.5259	0.3346	0.3846	0.2011	0.1386	**0.0450**	0.0471	0.1501	0.2311
	Accuracy	0.8175	0.7060	0.7540	0.4450	0.6410	0.5600	0.7530	0.8380	**0.9520**	0.9500	0.8250	0.6075
Kleinstiver_ GUIDE_BS*	AUC ROC	0.7305	0.7036	0.6947	0.6343	0.7106	0.7713	0.7485	0.7821	**0.9379**	0.8956	0.7442	0.7502
	Precision	0.1083	0.0967	0.1101	0.0768	0.0917	0.1636	0.1508	0.2459	**0.5460**	0.4574	0.1871	0.1323
	Recall	0.7222	0.7037	0.6296	0.9074	0.8519	0.8889	0.7778	0.6667	**0.1111**	0.1481	0.6111	0.9259
	F1 Score	0.1862	0.1535	0.1704	0.1266	0.1620	0.1387	0.1888	0.1875	**0.1946**	0.2500	0.1724	0.1362
	Brier Score	0.3251	0.3998	0.3000	0.6531	0.4484	0.5496	0.3101	0.2738	**0.0479**	0.0460	0.1848	0.3372
	Accuracy	0.6590	0.5807	0.6690	0.3241	0.5241	0.4041	0.6390	0.6880	**0.9503**	0.9521	0.6830	0.3660
Hmg_BS*	AUC ROC	0.6770	0.6935	0.6559	0.7106	0.7670	0.7454	0.8157	0.8130	0.8418	**0.8620**	0.7511	0.5934
	Precision	0.0837	0.0885	0.0833	0.1009	0.1205	0.1455	0.1796	0.2009	0.3617	**0.3832**	0.1216	0.0705
	Recall	0.3419	0.5962	0.3846	0.7885	0.6538	0.7885	0.8503	0.5748	0.1923	**0.2885**	0.4423	0.6154
	F1 Score	0.1461	0.1722	0.1541	0.1488	0.1771	0.1643	0.2536	0.2706	0.2778	**0.3721**	0.2125	0.1210
	Brier Score	0.1926	0.2770	0.1960	0.4351	0.2864	0.3756	0.2021	0.1397	0.0476	**0.0469**	0.1752	0.2649
	Accuracy	0.7927	0.7020	0.7801	0.5310	0.6833	0.5830	0.7405	0.8395	0.9480	**0.9493**	0.8295	0.5350

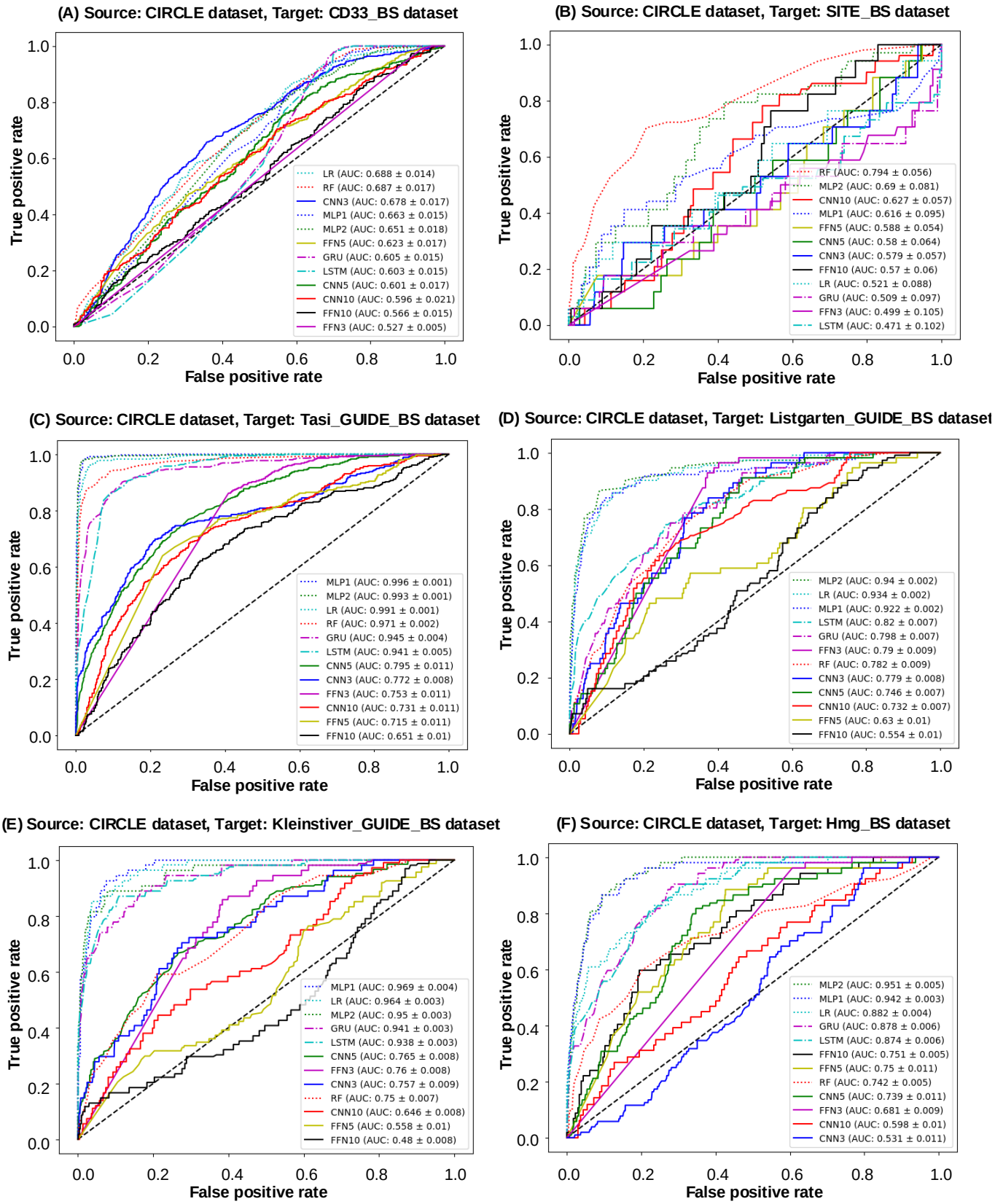

Figure 3.7 – ROC curves for model evaluation. ROC curves for models trained on the CIRCLE dataset, used as source, and six bootstrapped datasets.

Table 3.11 – Performance metrics for classification models obtained using the CIRCLE dataset.

Target	Metric	FNN3	FNN5	FNN10	CNN3	CNN5	CNN10	LSTM	GRU	**MLP1**	**MLP2**	RF	LR
CD33_BS	AUC ROC	0.5269	0.6227	0.5660	0.6781	0.6014	0.5955	0.6026	0.6053	0.6632	0.6515	0.6872	0.6884
	Precision	0.4818	0.5690	0.5164	0.6055	0.5245	0.5399	0.5159	0.5113	0.5729	0.5673	0.6190	0.6348
	Recall	0.9952	0.9797	1.0000	0.9957	0.9893	0.0085	1.0000	1.0000	1.0000	0.9786	0.0000	0.9979
	F1 Score	0.6477	0.6501	0.6407	0.6430	0.6391	0.0167	0.6376	0.6376	0.6624	0.6735	0.0000	0.6518
	Brier Score	0.3197	0.4865	0.4543	0.5132	0.5118	0.4009	0.5320	0.5320	0.4676	0.4266	0.4287	0.4843
	Accuracy	0.4932	0.5065	0.4750	0.4825	0.4770	0.5290	0.4680	0.4680	0.5230	0.5560	0.5320	0.5010
CIRCLE_BS*	AUC ROC	0.7754	0.7625	0.7419	0.7939	0.8369	0.7368	0.8888	0.8880	0.9959	**0.9956**	0.9924	0.9716
	Precision	0.0604	0.0692	0.1300	0.1950	0.1332	0.0762	0.2730	0.2354	0.9385	**0.9564**	0.9307	0.6556
	Recall	0.6923	0.8462	0.8523	0.8846	1.0000	0.1488	1.0000	1.0000	0.8846	**0.9231**	0.7551	0.4231
	F1 Score	0.1205	0.0742	0.0656	0.0838	0.0778	0.0868	0.0632	0.0633	0.9028	**0.9600**	0.8503	0.5703
	Brier Score	0.1723	0.5441	0.5808	0.4984	0.5994	0.0819	0.7650	0.7660	0.0041	**0.0021**	0.0058	0.0125
	Accuracy	0.7368	0.4510	0.3760	0.4965	0.3898	0.9198	0.2290	0.2298	0.9950	**0.9980**	0.9934	0.9835
SITE_BS	AUC ROC	0.4990	0.5876	0.5702	0.5786	0.5796	0.6268	0.4711	0.5090	0.6160	0.6902	0.7940	0.5208
	Precision	0.0403	0.0463	0.0462	0.0536	0.0570	0.0530	0.0631	0.0548	0.1544	0.1785	0.2569	0.1148
	Recall	0.2941	0.7059	1.0000	1.0000	0.7647	0.0392	0.1934	0.1765	0.0588	0.1176	0.0701	0.1765
	F1 Score	0.0505	0.0645	0.0720	0.0666	0.0633	0.0485	0.0613	0.0591	0.0612	0.1345	0.1173	0.1319
	Brier Score	0.2543	0.6838	0.7566	0.9450	0.7421	0.0595	0.2003	0.1967	0.0549	0.0470	0.0301	0.0675
	Accuracy	0.6245	0.3040	0.1230	0.0470	0.2300	0.9480	0.8010	0.8090	0.9385	0.9505	0.9663	0.9210
Tasi_ GUIDE_BS*	AUC ROC	0.7528	0.7151	0.6508	0.7718	0.7952	0.7312	0.9412	0.9449	**0.9964**	0.9930	0.9713	0.9909
	Precision	0.5341	0.5367	0.4677	0.7023	0.6763	0.5653	0.8822	0.9138	**0.9933**	0.9861	0.9522	0.9861
	Recall	0.8784	0.9011	0.8842	0.9294	1.0000	0.4492	1.0000	1.0000	**0.9746**	0.9435	0.7680	0.8588
	F1 Score	0.6568	0.5411	0.5283	0.5595	0.5476	0.5282	0.5229	0.5229	**0.9705**	0.9612	0.8599	0.9129
	Brier Score	0.2213	0.5245	0.4841	0.5067	0.5733	0.2172	0.6457	0.6454	**0.0182**	0.0216	0.0840	0.0458
	Accuracy	0.6750	0.4590	0.4410	0.4820	0.4150	0.7160	0.3540	0.3540	**0.9790**	0.9730	0.9115	0.9420
Listgarten_ GUIDE_BS*	AUC ROC	0.7896	0.6300	0.5541	0.7792	0.7461	0.7317	0.8204	0.7975	0.9224	**0.9395**	0.7819	0.9340
	Precision	0.1261	0.0897	0.0986	0.1491	0.1573	0.1177	0.3388	0.2091	0.6376	**0.6654**	0.1773	0.6347
	Recall	0.9464	0.9643	1.0000	1.0000	1.0000	0.0000	1.0000	1.0000	0.5357	**0.4286**	0.0000	0.6786
	F1 Score	0.2179	0.1234	0.1067	0.1167	0.1163	0.0000	0.1061	0.1061	0.5646	**0.5524**	0.0000	0.5803
	Brier Score	0.2608	0.7585	0.8058	0.8393	0.8341	0.0681	0.9438	0.9437	0.0389	**0.0381**	0.0514	0.0422
	Accuracy	0.6195	0.2330	0.0625	0.1520	0.1490	0.9260	0.0560	0.0560	0.9537	**0.9610**	0.9440	0.9450
Kleinstiver_ GUIDE_BS*	AUC ROC	0.7602	0.5580	0.4803	0.7566	0.7649	0.6457	0.9377	0.9412	**0.9690**	**0.9505**	0.7497	0.9643
	Precision	0.1126	0.0708	0.0899	0.2345	0.2391	0.0980	0.6364	0.6774	**0.6964**	**0.7314**	0.1759	0.6776
	Recall	0.7222	0.8696	1.0000	1.0000	0.9815	0.0000	1.0000	1.0000	**0.5741**	**0.6852**	0.0000	0.5185
	F1 Score	0.1973	0.1137	0.1086	0.1070	0.1168	0.0000	0.1025	0.1025	**0.6596**	**0.7048**	0.0000	0.6222
	Brier Score	0.2136	0.7190	0.7624	0.8903	0.7784	0.0590	0.9458	0.9455	**0.0276**	**0.0294**	0.0496	0.0244
	Accuracy	0.6827	0.2727	0.1130	0.0990	0.1980	0.9420	0.0540	0.0540	**0.9680**	**0.9690**	0.9462	0.9660
Hmg_BS	AUC ROC	0.6815	0.7501	0.7508	0.5313	0.7387	0.5976	0.8736	0.8782	0.9419	0.9508	0.7417	0.8817
	Precision	0.0797	0.1109	0.1485	0.0529	0.1140	0.0722	0.3842	0.3497	0.6047	0.4935	0.2402	0.4825
	Recall	0.9807	1.0000	1.0000	1.0000	0.9808	0.0000	1.0000	1.0000	0.9615	0.8654	0.0000	0.9615
	F1 Score	0.1417	0.1034	0.0990	0.1034	0.1079	0.0000	0.0989	0.0989	0.2813	0.4823	0.0000	0.1741
	Brier Score	0.3939	0.8954	0.8136	0.8969	0.8278	0.0603	0.9479	0.9476	0.2098	0.0883	0.0475	0.3634
	Accuracy	0.3823	0.0980	0.0530	0.0985	0.1570	0.9435	0.0520	0.0520	0.7445	0.9027	0.9480	0.5253

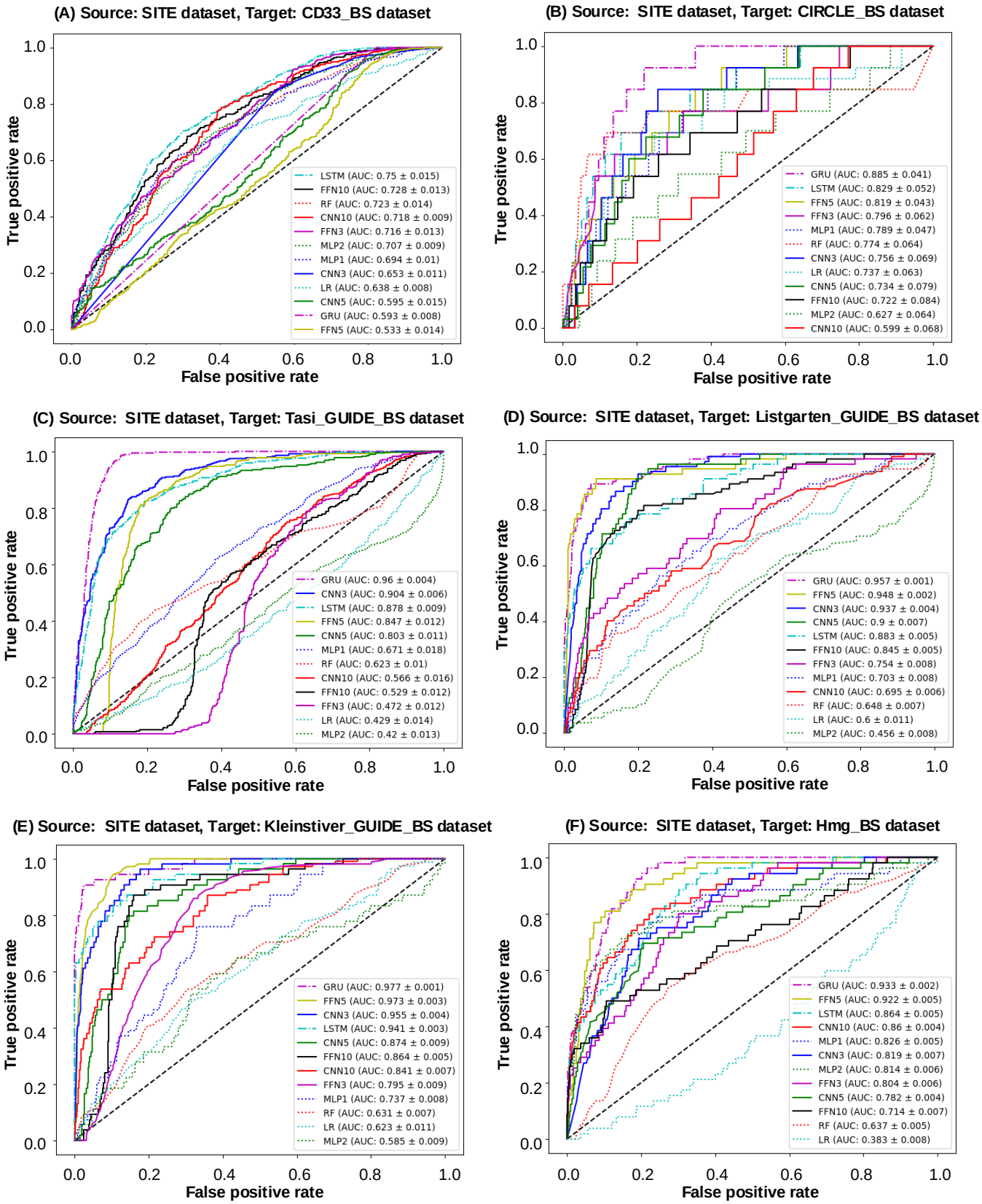

Figure 3.8 – ROC curves for model evaluation. ROC curves for models trained on the SITE dataset, used as source, and six bootstrapped datasets.

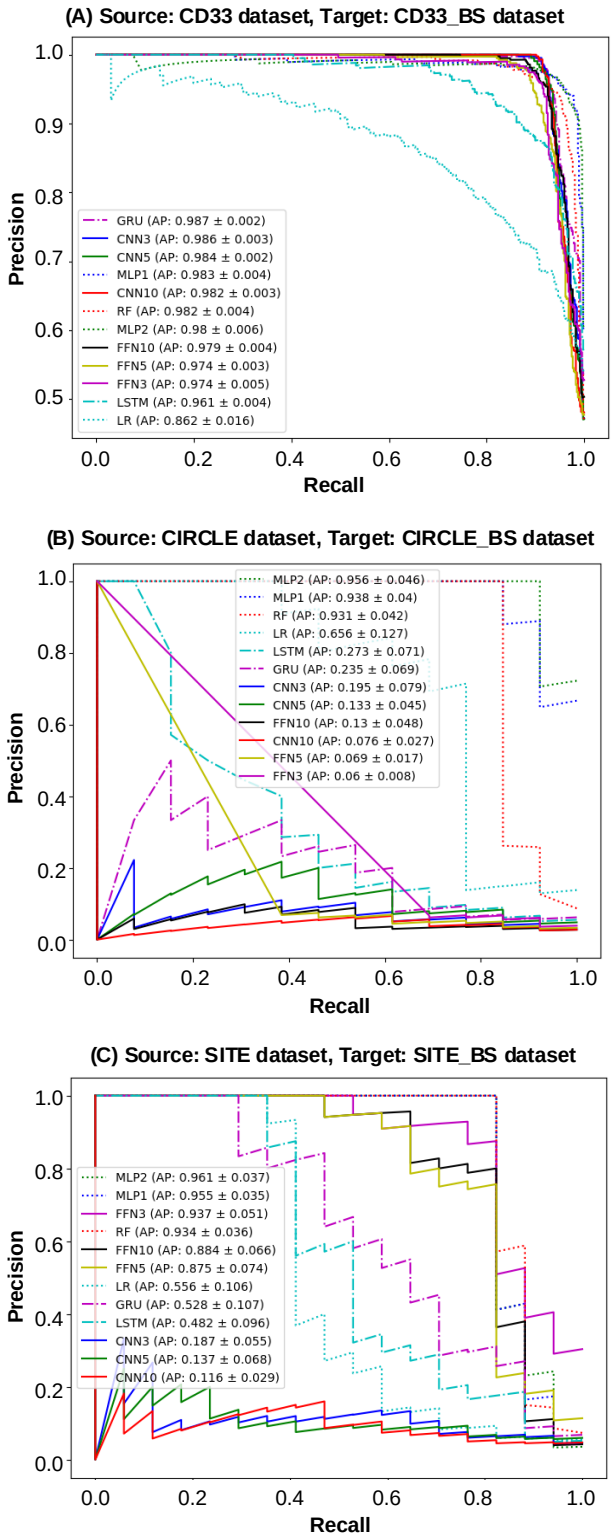

Figure 3.9 – Precision-Recall curves for model evaluation. Precision-Recall curves for models trained on : (A) CD33 dataset, (B) CIRCLE dataset, and (C) SITE dataset used as source and evaluated on their bootstrapped target counterparts.

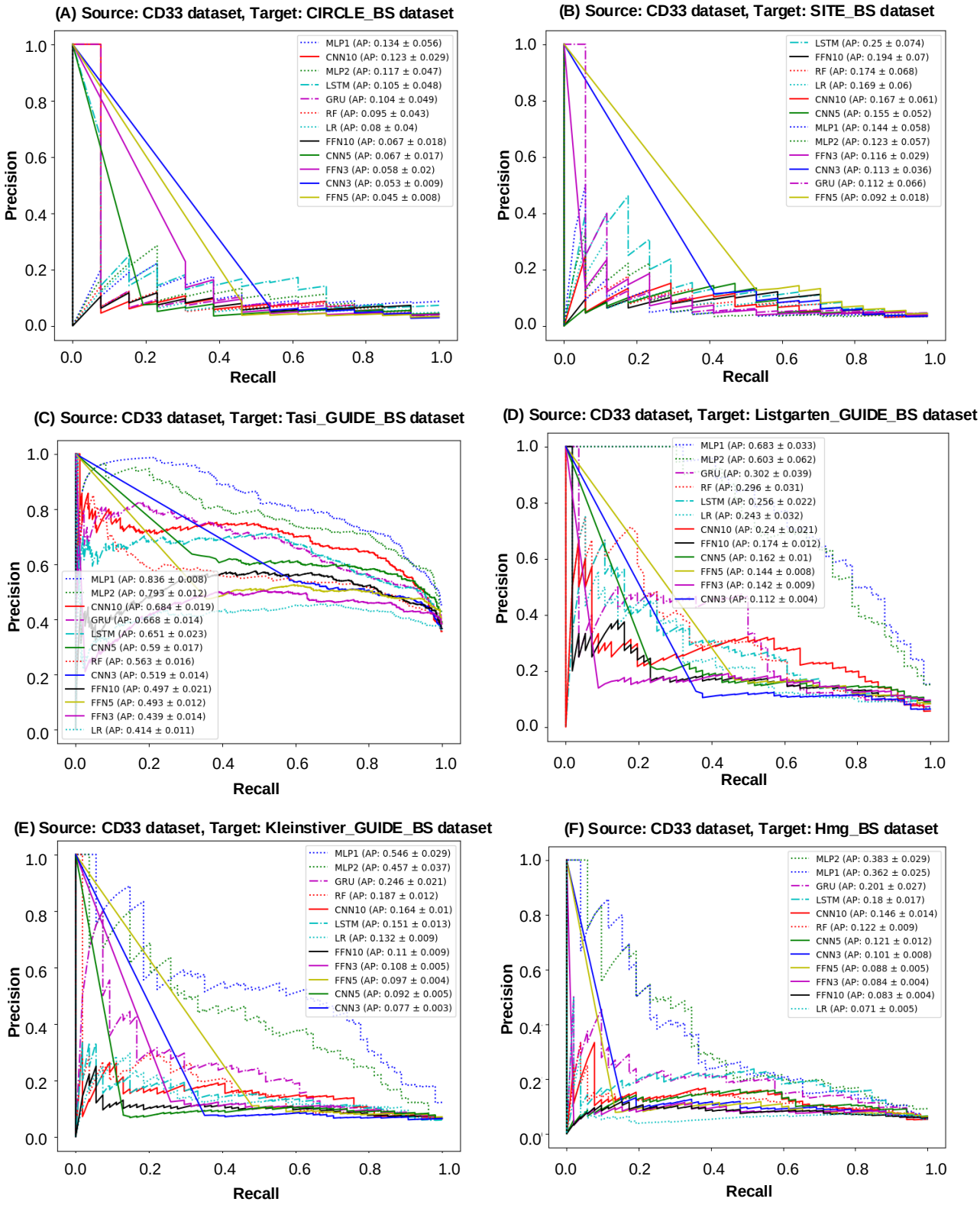

Figure 3.10 – Precision-Recall curves for model evaluation - CD33 dataset. Precision-Recall curves for the CD33 dataset, used as source, and six bootstrapped datasets.

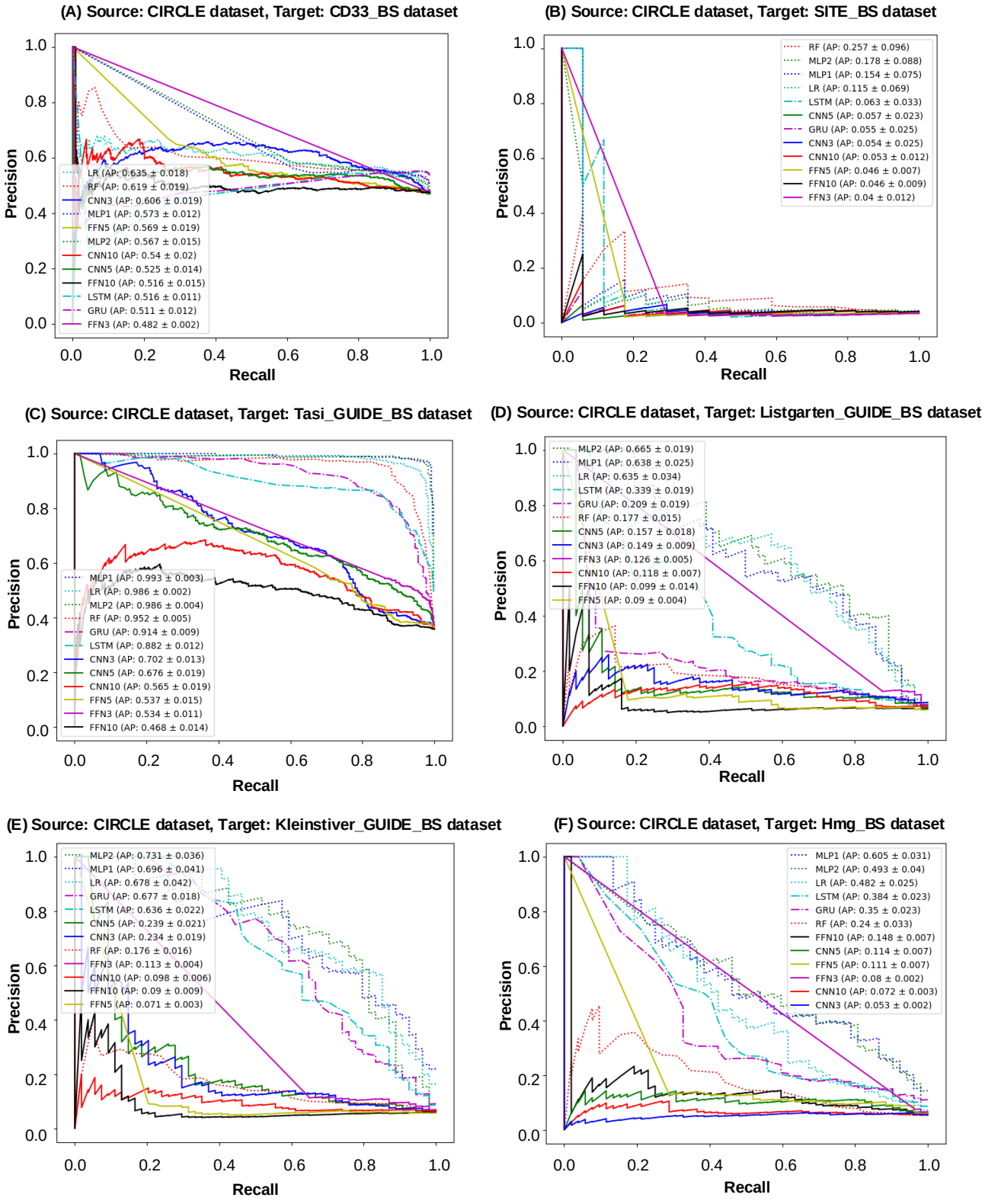

Figure 3.11 – Precision-Recall curves for model evaluation - CIRCLE dataset. Precision-Recall curves for the CIRCLE dataset, used as source, and six bootstrapped datasets.

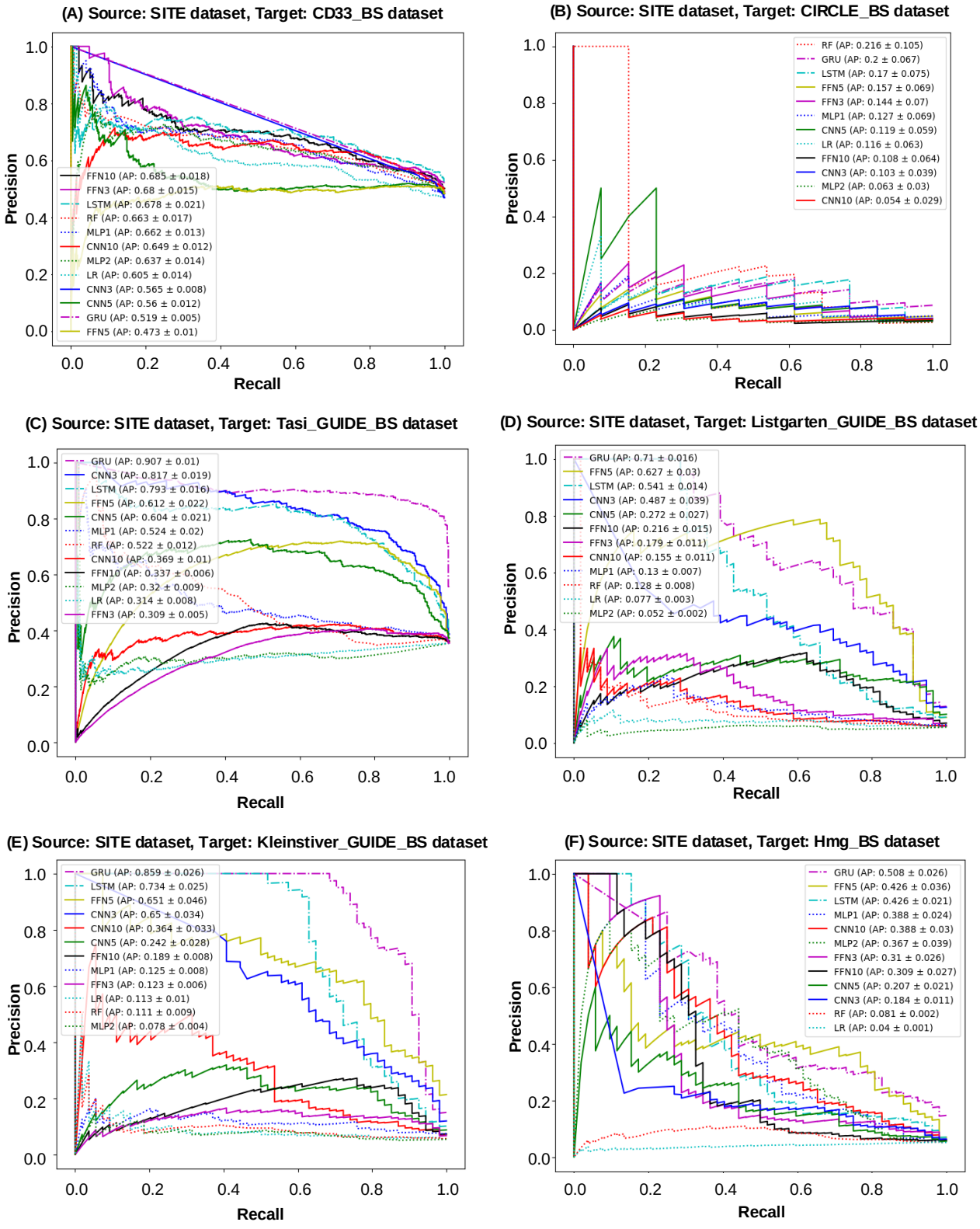

Figure 3.12 – Precision-Recall curves for model evaluation - SITE dataset. Precision-Recall curves for the SITE dataset, used as source, and six bootstrapped datasets.

Table 3.12 – Performance metrics for each considered classification model obtained using the SITE dataset.

Target	Metric	FNN3	FNN5	FNN10	CNN3	CNN5	CNN10	LSTM	GRU	**MLP1**	**MLP2**	RF	LR
CD33_BS	AUC ROC	0.7158	0.5332	0.7281	0.6531	0.5949	0.7178	0.7497	0.5926	0.6945	0.7074	0.7226	0.6376
	Precision	0.6802	0.4725	0.6852	0.5647	0.5598	0.6488	0.6777	0.5192	0.6619	0.6372	0.6631	0.6051
	Recall	0.5021	0.6282	0.7115	1.0000	1.0000	0.9808	1.0000	1.0000	0.4615	0.8568	0.0000	0.0000
	F1 Score	0.5725	0.5460	0.6755	0.6376	0.6376	0.6770	0.6376	0.6376	0.5461	0.6820	0.0000	0.0000
	Brier Score	0.2577	0.2694	0.2216	0.5318	0.5137	0.3577	0.5265	0.5320	0.3129	0.3294	0.4422	0.4665
	Accuracy	0.6490	0.5110	0.6800	0.4680	0.4680	0.5620	0.4680	0.4680	0.6410	0.6260	0.5320	0.5320
CIRCLE_BS	AUC ROC	0.7962	0.8191	0.7221	0.7563	0.7340	0.5987	0.8287	0.8854	0.7894	0.6273	0.7739	0.7371
	Precision	0.1440	0.1565	0.1077	0.1034	0.1188	0.0536	0.1704	0.2002	0.1270	0.0634	0.2160	0.1158
	Recall	0.0385	0.0220	0.0684	1.0000	0.8462	0.2462	0.9872	1.0000	0.0849	0.0559	0.0000	0.0000
	F1 Score	0.0571	0.0323	0.0757	0.0586	0.0854	0.0736	0.0627	0.0817	0.1008	0.0429	0.0000	0.0000
	Brier Score	0.0282	0.0280	0.0375	0.7098	0.3803	0.1235	0.5620	0.5001	0.0345	0.0577	0.0238	0.0255
	Accuracy	0.9670	0.9681	0.9570	0.1640	0.5292	0.8393	0.2330	0.4152	0.9627	0.9360	0.9740	0.9740
SITE_BS*	AUC ROC	**0.9939**	0.9883	0.9820	0.8019	0.7815	0.7767	0.8729	0.9142	0.9908	**0.9881**	0.9875	0.8795
	Precision	**0.9373**	0.8747	0.8838	0.1866	0.1367	0.1163	0.4825	0.5284	0.9547	**0.9613**	0.9338	0.5560
	Recall	**0.6176**	0.4706	0.4706	0.9706	0.9403	0.6863	1.0000	0.9118	0.8235	**0.9113**	0.8209	0.3262
	F1 Score	**0.7565**	0.6275	0.6275	0.0831	0.0882	0.1514	0.0799	0.1426	0.9032	**0.9450**	0.9016	0.4787
	Brier Score	**0.0118**	0.0144	0.0154	0.6250	0.5370	0.2173	0.5822	0.3077	0.0061	**0.0035**	0.0073	0.0208
	Accuracy	**0.9865**	0.9810	0.9810	0.2710	0.3432	0.7397	0.2170	0.6270	0.9940	**0.9965**	0.9940	0.9768
Tasi_ GUIDE_BS	AUC ROC	0.4724	0.8475	0.5287	0.9041	0.8027	0.5658	0.8784	0.9597	0.6712	0.4202	0.6226	0.4288
	Precision	0.3086	0.6122	0.3374	0.8173	0.6037	0.3693	0.7931	0.9072	0.5243	0.3196	0.5215	0.3138
	Recall	0.0932	0.1723	0.1384	1.0000	0.9520	0.2740	0.9972	1.0000	0.1461	0.1864	0.0565	0.0000
	F1 Score	0.1048	0.2552	0.1620	0.5527	0.6172	0.3201	0.5700	0.6093	0.2381	0.2333	0.1067	0.0000
	Brier Score	0.5417	0.2923	0.4474	0.4799	0.3151	0.3362	0.4143	0.3880	0.3131	0.4184	0.3067	0.3477
	Accuracy	0.4360	0.6440	0.4930	0.4270	0.5820	0.5880	0.4675	0.5460	0.6693	0.5660	0.6650	0.6460
Listgarten_ GUIDE_BS	AUC ROC	0.7540	0.9485	0.8447	0.9366	0.9003	0.6951	0.8826	0.9570	0.7035	0.4565	0.6475	0.6002
	Precision	0.1793	0.6272	0.2161	0.4875	0.2725	0.1548	0.5412	0.7098	0.1302	0.0521	0.1279	0.0775
	Recall	0.0893	0.4286	0.1429	1.0000	1.0000	0.3214	1.0000	1.0000	0.1071	0.0893	0.0000	0.0000
	F1 Score	0.1389	0.5393	0.1517	0.1208	0.1528	0.2130	0.1109	0.1615	0.1270	0.0400	0.0000	0.0000
	Brier Score	0.0592	0.0303	0.0684	0.7149	0.4775	0.0955	0.7056	0.4989	0.0740	0.2192	0.0524	0.0566
	Accuracy	0.9380	0.9590	0.9103	0.1845	0.3790	0.8670	0.1020	0.4185	0.9175	0.7600	0.9440	0.9420
Kleinstiver_ GUIDE_BS*	AUC ROC	0.7946	**0.9726**	0.8640	0.9547	0.8736	0.8406	0.9409	0.9769	0.7366	0.5852	0.6313	0.6227
	Precision	0.1231	**0.6514**	0.1887	0.6502	0.2419	0.3640	0.7339	0.8588	0.1246	0.0780	0.1111	0.1128
	Recall	0.0556	**0.5001**	0.2407	1.0000	0.9815	0.6111	1.0000	1.0000	0.0741	0.1667	0.0000	0.0000
	F1 Score	0.0619	**0.5902**	0.1745	0.1155	0.1608	0.3002	0.1425	0.2097	0.0879	0.1395	0.0000	0.0000
	Brier Score	0.0831	**0.0286**	0.0860	0.7231	0.4254	0.0994	0.4717	0.3086	0.0709	0.1052	0.0507	0.0529
	Accuracy	0.9086	**0.9625**	0.8770	0.1730	0.4470	0.8460	0.3500	0.5930	0.9170	0.8890	0.9460	0.9457
Hmg_BS*	AUC ROC	0.8035	**0.9225**	0.7144	0.8187	0.7817	0.8598	0.8642	0.9326	0.8259	0.8140	0.6368	0.3828
	Precision	0.3102	**0.4260**	0.3091	0.1840	0.2073	0.3882	0.4257	0.5081	0.3882	0.3666	0.0810	0.0396
	Recall	0.1123	**0.7692**	0.3058	1.0000	0.9805	0.4423	1.0000	1.0000	0.2692	0.4423	0.0000	0.0000
	F1 Score	0.1941	**0.5011**	0.3795	0.1042	0.1145	0.3566	0.1144	0.1244	0.3590	0.4600	0.0000	0.0000
	Brier Score	0.0437	**0.0599**	0.0481	0.8443	0.6543	0.0587	0.6440	0.6359	0.0432	0.0502	0.0491	0.0520
	Accuracy	0.9518	**0.9204**	0.9485	0.1060	0.2217	0.9170	0.1950	0.2680	0.9500	0.9460	0.9480	0.9480

overal performance across all metrics. Similarly, for the Listgarten_GUIDE_BS target dataset, MLP2 showcased a robust performance, maintained consistency across all evaluation metrics. When the Kleinstiver_BS dataset was used as target, the best overall results were achieved once again using the MLP1 and MLP2 models.

In the third scenario, the SITE dataset served as the source dataset in our transfer learning experiments. The obtained ROC curves (see Fig 3.5C) and PR curves (see Fig 3.9) are presented for all considered ML and DL models trained on the SITE dataset and evaluated on its bootstrapped counterpart, SITE_BS. Additionally, Fig. 3.8 compares the ROC curves for all considered models, using the complete SITE dataset as source and the bootstrapped variants of the remaining datasets as targets (for PR curves see Fig 3.12). Further quantitative results are provided in Table 3.12. For the SITE_BS target dataset, FNN3 and MLP2 emerged as the best-performing models across all metrics. When the Kleinstiver_BS and Hmg_BS datasets were used as targets, the FNN5 model demonstrated notable results across diverse evaluation metrics.

Based on the evaluation results summarized in Tables 3.10, 3.11, and 3.12 across the three scenarios, the CD33, CIRCLE, and SITE datasets were found to be the most suitable sources for their respective bootstrapped counterparts : CD33_BS, CIRCLE_BS, and SITE_BS. This result was rather expected given the highest similarity scores between the complete datasets and their bootsrapped counterparts observed for all the three similarity measures considered (see Table 3.9). Furthermore, among the three source datasets (CD33, SITE, and CIRCLE), CIRCLE was identified as the optimal source for the Tasi_GUIDE_BS target dataset across all metrics, when using the MLP1 model (this corresponds to the highest similarity scores between Tasi_GUIDE_BS and CIRCLE provided by cosine, Euclidean, and Manhattan metrics ; see Table 3.9), and for the Listgarten_GUIDE_BS target dataset, also across all metrics, when using the MLP2 model (this corresponds to the highest similarity score between Listgarten_GUIDE_BS and CIRCLE provided by cosine similarity ; see Table 3.9). When the Kleinstiver_GUIDE_BS dataset was used as target, the CIRCLE (with the MLP1 and MLP2 models) and SITE (with the FNN5 model) datasets emerged as the optimal sources (see Tables 3.11 and 3.12). This corresponds to the highest similarity score between Kleinstiver_GUIDE_BS and both CIRCLE and SITE provided by cosine similarity (see Table 3.9). When the Hmg_BS dataset was used as target, the SITE dataset (using the FNN5 model) was identified as the optimal source (see Table 3.12) ; once again, this reflects the highest similarity score between Hmg_BS and SITE provided by cosine similarity (see Table 3.9).

These results validate two critical points of our study : First, similarity score results are reliable and trustworthy indices for determining the most appropriate source dataset for a given target dataset prior to performing

transfer learning experiments in CRISPR-Cas9. They reinforce the effectiveness of our methodology as a robust pre-selection tool for transfer learning, providing a systematic approach for identifying efficiently suitable source data. Second, cosine distance (or cosine similarity) emerges as the most dependable metric, among the three metrics considered, for selecting the most appropriate source dataset for transfer learning.

Moreover, our results clearly demonstrate that similarity-based source data pre-selection is necessary to mitigate negative knowledge transfers. If a source dataset is chosen solely by size, availability, or even class imbalance ratio, but without similarity assessment, this could eventually lead to a suboptimal or negative transfer. For example, the CIRCLE and SITE datasets considered in our study have comparable sizes and almost identical class imbalance ratios (0.0128 and 0.0176, respectively - see Table 3.1), but a knowledge transfer from CIRCLE to SITE as well as that from SITE to CIRCLE are clearly negative with the highest F1-score values of 0.1345 (see Table 3.11) and 0.1008 (see Table 3.12), respectively, over all competing ML and DL models. Our similarity-based analysis suggests that such transfers should be avoided (see Table 3.9).

It is worth noting that in some cases a potential source dataset with a slightly lower cosine similarity with the target might still yield competitive or even superior performance for specific models - MLP-based models in our case. This could be due to such factors as a richer representation of specific rare patterns important for the target task or some MLP inductive biases aligning better with the source data distribution. For example, the recommended transfers with lower cosine similarity scores of 0.5701 from CIRCLE to Listgarten_GUIDE_BS and of 0.5672 from CIRCLE to Kleinstiver_GUIDE_BS led to competitive knowledge transfers with the corresponding best F1-score score values of 0.5646 and 0.7048, obtained, respectively, with MLP1 and MLP2 (see Table 3.11). It was not so for Euclidean and Manhattan similarities whose highest values in this case led to transfers from CD33 (instead of CIRCLE) with much lower best F1-score score values of 0.2500 for Listgarten_GUIDE_BS and of 0.1946 for Kleinstiver_GUIDE_BS, both obtained with MLP1 (see Table 3.10).

3.7 Conclusion

This study explores the effectiveness and applicability of transfer learning in improving CRISPR-Cas9 off-target predictions by adapting a similarity-based approach. We consider three popular distance measures - cosine, Euclidean, and Manhattan distances to assess similarity between a given target dataset and an ensemble of potential source datasets. A candidate source dataset having the highest similarity with the given target can then be recommended for transfer learning experiments.

Establishing the most appropriate source dataset for a given target dataset in the transfer learning prospective is a relevant theoretical problem in itself. We show how it can be effectively solved in practice in the context of CRISPR-Cas9 off-target prediction. The main novelty of our study consists in the proposed similarity-based pre-evaluation rather than in an innovative transfer learning algorithm or an effective deep learning network architecture.

Our experiments were conducted using seven real-world CRISPR-Cas9 off-target datasets : CD33, CIRCLE, SITE, Tasi_GUIDE, Listgarten_GUIDE, Kleinstiver_GUIDE, and Hmg. The performance of various deep learning network architectures, i.e. CNNs, FNNs, LSTM-RNNs, GRU-RNNs, and MLPs, alongside two traditional machine learning models, i.e. RF and LR, was evaluated in a comprehensive simulation study. Six evaluation metrics, including AUC ROC, Precision, Recall, F1-score, Brier score, and Accuracy were considered. AUC ROC, F1-score, and Brier score are well adapted for assessing the model performances in our case since real-world CRISPR-Cas9 data are often highly imbalanced.

Our results indicate that cosine distance stands out as the most reliable and consistent measure for assessing similarity between two CRISPR-Cas9 datasets in terms of off-target transfer learning experiments. High similarity values provided by cosine similarity usually correspond to the top results achieved by the considered evaluation metrics. This was not always the case of Euclidean and Manhattan distances whose results were highly correlated as we worked with binary data representations. Overall, MLP variants 1 and 2, 3- and 5-layer FNNs, and an RNN-GRU turned to be the best-performing models in our transfer learning scenarios. While these models tend to offer a superior performance in most cases, the choice between machine learning and deep learning models should depend on the characteristics of the given target and source datasets, taking into account the dataset sizes and an eventual class imbalance.

The fact that in many instances two simple MLP models outperformed much more sophisticated RNN-GRU and, especially, RNN-LSTM neural network architectures is not very surprising since MLPs usually cope well with tabular data, such as our CRISPR-Cas9 one-hot encoded sequence datasets, allowing for capturing complex, non-linear patterns, whereas RNN-based models excel at capturing time-series patterns and long-term dependencies in complex scenarios, being particularly useful in natural language processing and speech recognition.

Our findings highlight the critical role of similarity-based insights in optimizing transfer learning workflows.

Broader impacts of the proposed dual-layered framework are the following : (1) The new framework streamlines the transfer learning process by reducing the number of potential source datasets and recommended ML and DL models, and thus the number of trial-and-error attempts, which are convenient for a selected target dataset, and (2) it enables faster development of transfer learning models for CRISPR-Cas9 off-target prediction, which can now be successfully tested on mutually compatible sets (i.e. those with high cosine similarity scores) of source and target data.

In the future, we plan to compare our approach with transformer-based models optimized for tabular data [Gezici and Sefer, 2024] as well as with different data augmentation techniques allowing for better leverage of limited datasets [Mumuni and Mumuni, 2022]. Moreover, it would be interesting to extend the proposed similarity framework beyond sequence similarity by incorporating into it available biological and experimental information, such as species, cell type, enzyme type, experimental conditions/technology being used, and data size. Specifically, each of these factors could be added to the discussed $7L$ input vectors as an extra component, normalized to [0,1] range for numerical data and one-hot encoded for categorical data. It would also be interesting to integrate the proposed similarity-based selection with deep learning architectures for domain adaptation that explicitly address distribution shifts between source and target ([Sun et al., 2015, Rozantsev et al., 2018]). In this case, the source labeled sample weights could ideally be calculated leveraging both distribution and similarity patterns of the source and target samples.

CONCLUSION

This thesis has explored the integration of traditional machine learning and modern deep learning methods for the prediction of CRISPR/Cas9 activity, with particular emphasis on addressing the challenges of on-target efficiency and off-target specificity. We began by conducting a systematic review of existing computational approaches, highlighting the critical roles of data preprocessing, feature encoding, class imbalance handling, and model architecture in determining predictive success. While conventional scoring tools [Ran et al., 2013, Stemmer et al., 2015, Hsu et al., 2013, Heigwer et al., 2014, Montague et al., 2014] have historically been widely used, recent advances in data-driven ML and DL models have demonstrated superior predictive power, particularly as the availability of large-scale CRISPR datasets continues to expand. These models are increasingly central to clinical research efforts, where ensuring both accuracy and reliability is of paramount importance.

A key contribution of this thesis lies in the development of a Bayesian Test-Time Augmentation (BayTTA) framework. By integrating Bayesian Model Averaging with TTA, BayTTA improved predictive accuracy while simultaneously quantifying model uncertainty, thereby enhancing both robustness and interpretability. This methodological advance is particularly valuable in high-stakes fields such as genome editing and medical diagnostics, where decision-making depends not only on accuracy but also on an understanding of the confidence associated with predictions.

This thesis has also investigated transfer learning as a strategy for addressing data scarcity in CRISPR/Cas9 applications. In particular, we proposed a similarity-based framework that evaluates cosine, Euclidean, and Manhattan distance measures to guide the selection of appropriate source datasets for pretraining. Through extensive experimentation, we demonstrated that dataset similarity plays a decisive role in determining the success of TL, as mismatches between source and target domains can significantly degrade predictive performance. By incorporating similarity analysis, our framework offers a systematic and reliable method for dataset pairing, thereby streamlining the TL process and improving generalizability across heterogeneous CRISPR datasets.

Taken together, the findings of this thesis advance the application of AI-driven computational models in genome editing by : (i) improving accuracy in on- and off-target prediction, (ii) introducing uncertainty-aware frameworks for more trustworthy decision-making, and (iii) establishing similarity-based strategies for

optimizing transfer learning in data-limited contexts. These contributions not only enhance the scientific understanding of CRISPR/Cas9 predictive modeling but also provide practical tools that can support safer and more effective applications in both experimental and clinical settings. Future work may build upon these foundations by exploring multimodal learning approaches, integrating structural and epigenetic features, and expanding the transfer learning paradigm to broader domains within computational biology.

In **Chapter I**, we conducted an extensive review of current applications of machine learning and deep learning algorithms for CRISPR/Cas9 activity prediction. Our analysis highlighted several important observations : Firstly, the encoding of sgRNA-DNA sequences plays a pivotal role in model performance. Early models adopted simple one-hot encoding [Lin and Wong, 2018], but recent advancements have introduced more sophisticated encoding schemes that incorporate supplementary information channels reflecting insertions, deletions, and mismatches [Charlier et al., 2021, Lin et al., 2020], significantly improving predictive accuracy. Secondly, ensemble learning techniques such as AdaBoost [Zhang et al., 2019b] and Random Forests [Abadi et al., 2017] have demonstrated superior performance compared to non-ensemble methods like logistic regression and support vector machines (SVMs) [Fusi et al., 2015], owing to their ability to aggregate diverse models and reduce variance. Thirdly, the importance of feature selection and engineering was emphasized, with recent methodologies incorporating sequence-derived features such as gene melting temperature, molecular weight, and microhomology properties [Wang et al., 2019a]. Automated feature learning strategies have been explored to further enhance model generalization [Zhang and Jiang, 2022].

Additionally, we discussed the persistent challenge of class imbalance in publicly available CRISPR datasets, which often contain disproportionate numbers of positive and negative samples. Techniques such as data augmentation [Zhang et al., 2020c] and under-sampling [Liu et al., 2020b] have been proposed to mitigate this issue, ensuring more balanced and representative training data [Heaton, 2018].

Moreover, while deep neural networks have consistently outperformed traditional machine learning methods on large datasets [Lin and Wong, 2018, Wang et al., 2019a, Lin et al., 2020, Charlier et al., 2021], we observed that for smaller datasets, simpler models such as SVMs or Random Forests may yield better predictive results [Konstantakos et al., 2022b]. Attention-based architectures have recently gained traction, with models incorporating attention mechanisms showing substantial improvements in model interpretability and performance [Zhang et al., 2021, Xiao et al., 2021, Vaswani et al., 2017, Chen et al., 2022, Shen et al., 2022, de Santana Correia and Colombini, 2022, Basiri et al., 2021, Liu et al., 2019].

In addition to summarizing current progress, this chapter has identified several research gaps and future directions. While many studies have made substantial contributions to the field, important challenges remain. A critical issue is the explainability and interpretability of deep learning models. As ML and DL methods are increasingly deployed in clinical and other high-stakes applications, it is imperative to ensure that model decisions can be understood at a human-interpretable level [Miller, 2019, Chou et al., 2022, Vilone and Longo, 2021].

Despite the advances achieved thus far, future research should prioritize :

1. Developing explainable AI frameworks to enhance model transparency in clinical contexts ;
2. Improving transfer learning effectiveness through advanced similarity assessment techniques ;

This thesis outlines a roadmap for tackling these challenges in the context of genome editing. The methodologies and insights presented are intended to guide researchers and practitioners in building robust, accurate, and interpretable AI models for CRISPR/Cas9 activity prediction.

Chapter II introduced a novel methodology combining Test-Time Augmentation (TTA) and Bayesian Model Averaging (BMA), termed BayTTA, to enhance the robustness and uncertainty-awareness of deep learning models. Our empirical results demonstrated that BayTTA outperforms traditional TTA by incorporating posterior probabilities as weighting factors during model aggregation. This allows for a more principled combination of model predictions, significantly improving predictive performance, particularly in high-stakes domains such as medical imaging and genome editing where predictive confidence is essential.

One of the key strengths of the BayTTA approach lies in its capacity to quantify model uncertainty. This capability not only enhances the accuracy of model outputs but also provides interpretable confidence estimates, enabling practitioners to assess the reliability of each prediction. Although validated in the context of medical image classification, the proposed methodology is generalizable and can be extended to other AI applications requiring uncertainty-aware decision-making. However, we also noted that the effectiveness of BayTTA depends on factors such as dataset size, sample variability, and class distribution, which may affect its scalability in other domains.

In **Chapter III**, we addressed a significant gap in transfer learning for CRISPR/Cas9 off-target prediction. Establishing the most appropriate source dataset for a given target dataset is a crucial yet often overlooked

aspect of transfer learning. We proposed a similarity-based framework that evaluates cosine, Euclidean, and Manhattan distance metrics to systematically pre-select the optimal source dataset. This pre-evaluation approach allows practitioners to ensure better alignment between source and target domains before performing transfer learning experiments. Our experiments, conducted on seven real-world CRISPR off-target datasets (CD33, CIRCLE, SITE, Tasi_GUIDE, Listgarten_GUIDE, Kleinstiver_GUIDE, and Hmg), revealed that cosine similarity provides the most reliable measure for dataset alignment. Furthermore, we observed that simple architectures such as Multi-Layer Perceptrons (MLPs) often outperform more complex networks like RNN-LSTM when applied to tabular, one-hot encoded CRISPR data. This observation underscores the importance of model selection strategies that are tailored to dataset characteristics, particularly size, representation format, and class balance.

Future investigations at the intersection of AI and CRISPR/Cas9 genome editing may progress along three complementary research pathways :

1. **Enhancing Transfer Learning and Data Utilization :** Developing methods to better select suitable source datasets for transfer learning, integrating biological and experimental context (e.g., species, cell type, enzyme type), and exploring domain adaptation techniques to handle distribution shifts ([Sun et al., 2015, Rozantsev et al., 2018]). Additionally, comparing current approaches with transformer-based models optimized for tabular data [Gezici and Sefer, 2024] and advanced data augmentation strategies may further improve predictive performance on limited datasets.
2. **Model Optimization and Interpretability :** Improving network architectures and hyperparameter tuning (e.g., Bayesian optimization, evolutionary strategies) is essential for building more efficient and robust models. At the same time, explainability techniques such as SHAP, TreeSHAP, and counterfactual explanations should be employed to better interpret predictions and gain insights into on- and off-target activities, which is critical for clinical translation.
3. **Feature Engineering and Active Learning :** Incorporating additional informative biological features (e.g., epigenetic properties, RNA folding scores, microhomology signals) alongside deep learning–based automatic feature discovery may increase predictive accuracy. Furthermore, applying active learning strategies to leverage large pools of unlabeled genomic data can help reduce labeling costs while improving model training efficiency.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow : Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Shiran Abadi, Winston X. Yan, David Amar, and Itay Mayrose. A machine learning approach for predicting crispr-cas9 cleavage efficiencies and patterns underlying its mechanism of action. *PLoS Computational Biology*, 13(10) :e1005807, 2017.

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning : Techniques, applications and challenges. *Information Fusion*, 76 : 243–297, 2021a.

Moloud Abdar, Maryam Samami, Sajjad Dehghani Mahmoodabad, Thang Doan, Bogdan Mazoure, Reza Hashemifesharaki, Li Liu, Abbas Khosravi, U Rajendra Acharya, Vladimir Makarenkov, et al. Uncertainty quantification in skin cancer classification using three-way decision-based bayesian deep learning. *Computers in Biology and Medicine*, 135 :104418, 2021b.

Moloud Abdar, Soorena Salari, Sina Qahremani, Hak-Keung Lam, Fakhri Karray, Sadiq Hussain, Abbas Khosravi, U Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. Uncertaintyfusenet : robust uncertainty-aware hierarchical feature fusion model with ensemble monte carlo dropout for covid-19 detection. *Information Fusion*, 90 :364–381, 2023.

Taiga Abe, Estefany Kelly Buchanan, Geoff Pleiss, Richard Zemel, and John P. Cunningham. Deep ensembles work, but are they necessary ? *Advances in Neural Information Processing Systems*, 35 : 33646–33660, 2022.

Andrew J Aguirre, Robin M Meyers, Barbara A Weir, Francisca Vazquez, Cheng-Zhong Zhang, Uri Ben-David, April Cook, Gavin Ha, William F Harrington, Mihir B Doshi, et al. Genomic copy number dictates a gene-independent cell response to crispr/cas9 targetinggenomic copy number affects crispr/cas9 screens. *Cancer Discovery*, 6(8) :914–929, 2016.

Özlem Aktas, Elif Dogan, and Tolga Ensari. Crispr/cas9 target prediction with deep learning. In *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pages 1–5. IEEE, 2019.

Felicity Allen, Luca Crepaldi, Clara Alsinet, Alexander J. Strong, Vitalii Kleshchevnikov, Pietro De Angeli, Petra Páleníková, Anton Khodak, Vladimir Kiselev, Michael Kosicki, et al. Predicting the mutations generated by repair of cas9-induced double-strand breaks. *Nature Biotechnology*, 37(1) :64–72, 2019.

Talal Almutiri, Faisal Saeed, and Manar Alassaf. A survey of machine learning and deep learning applications in genome editing. In *Advances on smart and soft computing*, pages 145–162. Springer, 2022.

Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. In *Medical Imaging with Deep Learning*, 2022.

Sangsu Bae, Jeongbin Park, and Jin-Soo Kim. Cas-offinder : a fast and versatile algorithm that searches for potential off-target sites of cas9 rna-guided endonucleases. *Bioinformatics*, 30(10) :1473–1475, 2014.

Yuval Bahat and Gregory Shakhnarovich. Classification confidence estimation with test-time data-augmentation. *arXiv preprint arXiv :2006.16705*, 2020.

Rasmus O. Bak, Natalia Gomez-Ospina, and Matthew H. Porteus. Gene editing on center stage. *Trends in Genetics*, 34(8) :600–611, 2018.

Rodolphe Barrangou and Jennifer A. Doudna. Applications of crispr technologies in research and beyond. *Nature Biotechnology*, 34(9) :933–941, 2016.

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

František Bartoš, Quentin F Gronau, Bram Timmers, Willem M Otte, Alexander Ly, and Eric-Jan Wagenmakers. Bayesian model-averaged meta-analysis in medicine. *Statistics in Medicine*, 40(30) : 6743–6761, 2021.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U Rajendra Acharya. Abcdm : An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115 :279–294, 2021.

Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press, 2017.

Thierry Bertomeu, Jasmin Coulombe-Huntington, Andrew Chatr-Aryamontri, Karine G Bourdages, Etienne Coyaud, Brian Raught, Yu Xia, and Mike Tyers. A high-resolution genome-wide crispr/cas9 viability screen reveals structural features and contextual diversity of the human cell-essential proteome. *Molecular and Cellular Biology*, 38(1) :e00302–17, 2018.

Christopher M. Bishop et al. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Ankur Biswas, Paritosh Bhattacharya, Santi P Maity, and Rita Banik. Data augmentation for improved brain tumor segmentation. *IETE Journal of Research*, 69(5) :2772–2782, 2023.

Adam J. Bogdanove, Andrew Bohm, Jeffrey C. Miller, Richard D. Morgan, and Barry L. Stoddard. Engineering altered protein–dna recognition specificity. *Nucleic Acids Research*, 46(10) :4845–4871, 2018.

Ferhat Bozkurt. Skin lesion classification on dermatoscopic images using effective data augmentation and pre-trained deep learning approach. *Multimedia Tools and Applications*, 82(12) :18985–19003, 2023.

Leo Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

Lisa Brenan, Aleksandr Andreev, Ofir Cohen, Sasha Pantel, Atanas Kamburov, Davide Cacchiarelli, Nicole S Persky, Cong Zhu, Mukta Bagul, Eva M Goetz, et al. Phenotypic characterization of a comprehensive set of mapk1/erk2 missense mutants. *Cell Reports*, 17(4) :1171–1183, 2016.

Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.

Peter Cameron, Chris K. Fuller, Paul D. Donohoue, Brittnee N. Jones, Matthew S. Thompson, Matthew M. Carter, Scott Gradia, Bastien Vidal, Elizabeth Garner, Euan M. Slorach, et al. Mapping the genomic landscape of crispr–cas9 cleavage. *Nature Methods*, 14(6) :600–606, 2017.

Dana Carroll. Genome engineering with zinc-finger nucleases. *Genetics*, 188(4) :773–782, 2011.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1) :41–75, 1997.

Nannan Chang, Changhong Sun, Lu Gao, Dan Zhu, Xiufei Xu, Xiaojun Zhu, Jing-Wei Xiong, and Jianzhong Jeff Xi. Genome editing with rna-guided cas9 nuclease in zebrafish embryos. *Cell Research*, 23(4) : 465–472, 2013.

Raj Chari, Prashant Mali, Mark Moosburner, and George M. Church. Unraveling crispr-cas9 genome engineering parameters via a library-on-library approach. *Nature Methods*, 12(9) :823–826, 2015.

Jeremy Charlier, Robert Nadon, and Vladimir Makarenkov. Accurate deep learning off-target prediction with novel sgrna-dna sequence encoding in crispr-cas9 gene editing. *Bioinformatics*, 37(16) :2299–2307, 2021.

Jeremy Charlier, Zeinab Sherkatghanad, and Vladimir Makarenkov. Similarity-based transfer learning with deep learning networks for accurate crispr-cas9 off-target prediction. *PLOS Computational Biology*, 21(10) :e1013606, 2025.

Dong Chen, Wenjie Shu, and Shaoliang Peng. Predicting crispr-cas9 off-target with self-supervised neural networks. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 245–250. IEEE, 2020a.

Janice S. Chen, Yavuz S. Dagdas, Benjamin P. Kleinstiver, Moira M. Welch, Alexander A. Sousa, Lucas B. Harrington, Samuel H. Sternberg, J. Keith Joung, Ahmet Yildiz, and Jennifer A. Doudna. Enhanced proofreading governs crispr–cas9 targeting accuracy. *Nature*, 550(7676) :407–410, 2017.

Shengmiao Chen, Yufeng Yao, Yanchun Zhang, and Gaofeng Fan. Crispr system : discovery, development and off-target detection. *Cellular Signalling*, 70 :109577, 2020b.

Shi-an Anderson Chen and Elizabeth Tran. Optimizing precision genome editing through machine learning. *Forest (C= 0.01, l2)*, 85(15.78) :1–39, 2019.

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

Weitao Chen, Shubing Ouyang, Wei Tong, Xianju Li, Xiongwei Zheng, and Lizhe Wang. Gcsanet : A global context spatial attention deep learning network for remote sensing scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15 :1150–1162, 2022.

Hyunghun Cho, Yongjin Kim, Eunjung Lee, Daeyoung Choi, Yongjae Lee, and Wonjong Rhee. Basic enhancement strategies when using bayesian optimization for hyperparameter tuning of deep neural networks. *IEEE Access*, 8 :52588–52608, 2020.

Seung Woo Cho, Sojung Kim, Jong Min Kim, and Jin-Soo Kim. Targeted genome engineering in human cells with the cas9 rna-guided endonuclease. *Nature Biotechnology*, 31(3) :230–232, 2013.

Francois Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015. GitHub repository.

Yu-Liang Chou, Catarina Moreira, Peter Bruza, Chun Ouyang, and Joaquim Jorge. Counterfactuals and causability in explainable artificial intelligence : Theory, algorithms, and applications. *Information Fusion*, 81 :59–83, 2022.

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

Sewhan Chun, Jae Young Lee, and Junmo Kim. Cyclic test time augmentation with entropy weight method. In *Uncertainty in Artificial Intelligence*, pages 433–442. PMLR, 2022.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv :1412.3555*, 2014.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

Le Cong, F Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D. Hsu, Xuebing Wu, Wenyan Jiang, Luciano A. Marraffini, et al. Multiplex genome engineering using crispr/cas systems. *Science*, 339(6121) :819–823, 2013.

Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are bayesian. *Advances in Neural Information Processing Systems*, 34 :3451–3465, 2021.

Soumyya Kanti Datta, Mohammad Abuzar Shaikh, Sargur N Srihari, and Mingchen Gao. Soft attention improves skin cancer classification performance. In *Interpretability of Machine Intelligence in Medical Image Computing, and Topological Data Analysis and Its Applications for Medical Data : 4th International Workshop, iMIMIC 2021, and 1st International Workshop, TDA4MedicalData 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 4*, pages 13–23. Springer, 2021.

Alana de Santana Correia and Esther Luna Colombini. Attention, please ! a survey of neural attention models in deep learning. *Artificial Intelligence Review*, pages 1–88, 2022.

Jaspreet Kaur Dhanjal, Navaneethan Radhakrishnan, and Durai Sundar. Crispcut : a novel tool for designing optimal sgrnas for crispr/cas9 based experiments in human cells. *Genomics*, 111(4) :560–566, 2019.

Jaspreet Kaur Dhanjal, Samvit Dammalapati, Shreya Pal, and Durai Sundar. Evaluation of off-targets predicted by sgrna design tools. *Genomics*, 112(5) :3609–3614, 2020.

Giovanni Dimauro, Pierpasquale Colagrande, Roberto Carlucci, Mario Ventura, Vitoantonio Bevilacqua, and Danilo Caivano. Crisprlearner : A deep learning-based system to predict crispr/cas9 sgrna on-target cleavage efficiency. *Electronics*, 8(12) :1478, 2019.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg : Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021.

John G. Doench, Ella Hartenian, Daniel B. Graham, Zuzana Tothova, Mudra Hegde, Ian Smith, Meagan Sullender, Benjamin L. Ebert, Ramnik J. Xavier, and David E. Root. Rational design of highly active sgrnas for crispr-cas9–mediated gene inactivation. *Nature Biotechnology*, 32(12) :1262–1267, 2014.

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

Katherine F Donovan, Mudra Hegde, Meagan Sullender, Emma W Vaimberg, Cory M Johannessen, David E Root, and John G Doench. Creation of novel protein variants with crispr/cas9-mediated mutagenesis : turning a screening by-product into a discovery tool. *PLoS One*, 12(1) :e0170445, 2017.

David Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society Series B : Statistical Methodology*, 57(1) :45–70, 1995.

David Draper. Bayesian model specification : heuristics and examples. *Bayesian theory and applications*, pages 409–431, 2013.

V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly. Uncertainty quantification in deep mri reconstruction. *IEEE Transactions on Medical Imaging*, 40 :239–250, 2021.

Amy R Eggers, Kai Chen, Katarzyna M Soczek, Owen T Tuck, Erin E Doherty, Bryant Xu, Marena I Trinidad, Brittney W Thornton, Peter H Yoon, and Jennifer A Doudna. Rapid dna unwinding accelerates genome editing by engineered crispr-cas9. *Cell*, 187(13) :3249–3261, 2024.

Shai Elkayam and Yaron Orenstein. Deepcristl : deep transfer learning to predict crispr/cas9 functional and endogenous on-target editing efficiency. *Bioinformatics*, 38(Supplement_1) :i161–i168, 2022.

Shai Elkayam, Ido Tziony, and Yaron Orenstein. Deepcristl : deep transfer learning to predict crispr/cas9 on-target editing efficiency in specific cellular contexts. *Bioinformatics*, 40(8) :btae481, 2024.

Kevin M. Esvelt and Harris H. Wang. Genome-scale engineering for systems and synthetic biology. *Molecular Systems Biology*, 9(1) :641, 2013.

Bastiaan Evers, Katarzyna Jastrzebski, Jeroen PM Heijmans, Wipawadee Grernrum, Roderick L Beijersbergen, and Rene Bernards. Crispr knockout screening outperforms shrna and crispri in identifying essential genes. *Nature Biotechnology*, 34(6) :631–633, 2016.

Behnom Farboud and Barbara J. Meyer. Dramatic enhancement of genome editing by crispr/cas9 through improved guide rna design. *Genetics*, 199(4) :959–971, 2015.

Yibo Feng, Xu Yang, Dawei Qiu, Huan Zhang, Dejian Wei, and Jing Liu. Pcxrnet : Pneumonia diagnosis from chest x-ray images using condense attention block and multiconvolution attention block. *IEEE Journal of Biomedical and Health Informatics*, 26(4) :1484–1495, 2022.

Roger Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2000.

Yoav Freund, Robert E. Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156. Citeseer, 1996.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA :, 2001.

Richard L. Frock, Jiazhi Hu, Robin M. Meyers, Yu-Jui Ho, Erina Kii, and Frederick W. Alt. Genome-wide detection of dna double-stranded breaks induced by engineered nucleases. *Nature Biotechnology*, 33(2) :179–186, 2015.

Rongjie Fu, Wei He, Jinzhuang Dou, Oscar D. Villarreal, Ella Bedford, Helen Wang, Connie Hou, Liang Zhang, Yalong Wang, Dacheng Ma, et al. Systematic decomposition of sequence determinants governing crispr/cas9 specificity. *Nature Communications*, 13(1) :474, 2022.

Nicolo Fusi, Ian Smith, John Doench, and Jennifer Listgarten. In silico predictive modeling of crispr/cas9 guide efficiency. *BioRxiv*, page 021568, 2015.

James A. Gagnon, Eivind Valen, Summer B. Thyme, Peng Huang, Laila Ahkmetova, Andrea Pauli, Tessa G. Montague, Steven Zimmerman, Constance Richter, and Alexander F. Schier. Efficient mutagenesis by cas9 protein-mediated oligonucleotide insertion and large-scale assessment of single-guide rnas. *PLOS ONE*, 9(5) :e98186, 2014.

Mélanie Gaillochet, Christian Desrosiers, and Hervé Lombaert. Taal : Test-time augmentation for active learning in medical image segmentation. In *MICCAI Workshop on Data Augmentation, Labelling, and Imperfections*, pages 43–53. Springer, 2022.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation : Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059. PMLR, 2016.

Shashank Gandhi, Lionel Christiaen, and Alberto Stolfi. Rational design and whole-genome predictions of single guide rnas for efficient crispr/cas9-mediated genome editing in ciona. 2016.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget : Continual prediction with lstm. *Neural Computation*, 12(10) :2451–2471, 2000.

Abdul Haluk Batur Gezici and Emre Sefer. Deep transformer-based asset price and direction prediction. *IEEE Access*, 12 :24164–24178, 2024.

Luke A Gilbert, Max A Horlbeck, Britt Adamson, Jacqueline E Villalta, Yuwen Chen, Evan H Whitehead, Carla Guimaraes, Barbara Panning, Hidde L Ploegh, Michael C Bassik, et al. Genome-scale crispr-mediated control of gene repression and activation. *Cell*, 159(3) :647–661, 2014.

Evgin Goceri. Medical image data augmentation : Techniques, comparisons and interpretations. *Artificial Intelligence Review*, pages 1–45, 2023.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E Turner. Meta-learning probabilistic inference for prediction. *arXiv preprint arXiv :1805.09921*, 2018.

Mahesh Gour and Sweta Jain. Uncertainty-aware convolutional neural network for covid-19 x-ray images classification. *Computers in Biology and Medicine*, 140 :105047, 2022.

Jiahui Guo, Tianmin Wang, Changge Guan, Bing Liu, Cheng Luo, Zhen Xie, Chong Zhang, and Xin-Hui Xing. Improved sgrna design in bacteria via genome-wide activity profiling. *Nucleic Acids Research*, 46 (14) :7052–7069, 2018.

Rajat M Gupta, Kiran Musunuru, et al. Expanding the genetic editing tool kit : Zfns, talens, and crispr-cas9. *The Journal of clinical investigation*, 124(10) :4154–4161, 2014.

Maximilian Haeussler, Kai Schönig, Hélène Eckert, Alexis Eschstruth, Joffrey Mianné, Jean-Baptiste Renaud, Sylvie Schneider-Maunoury, Alena Shkumatava, Lydia Teboul, Jim Kent, et al. Evaluation of off-target and on-target scoring algorithms and integration into the guide rna selection tool crispor. *Genome Biology*, 17(1) :1–12, 2016.

Dalton T Ham, Tyler S Browne, Pooja N Banglorewala, Tyler L Wilson, Richard K Michael, Gregory B Gloor, and David R Edgell. A generalizable cas9/sgrna prediction model using machine transfer learning with small high-quality datasets. *Nature communications*, 14(1) :5514, 2023.

Fatemeh Hamedani-KarAzmoudehFar, Reza Tavakkoli-Moghaddam, Amir Reza Tajally, and Seyed Sina Aria. Breast cancer classification by a new approach to assessing deep neural network-based uncertainty quantification methods. *Biomedical Signal Processing and Control*, 79 :104057, 2023.

Q. Han, X. Qian, H. Xu, K. Wu, L. Meng, Z. Qiu, T. Weng, B. Zhou, and X. Gao. Dm-cnn : Dynamic multi-scale convolutional neural network with uncertainty quantification for medical image classification. *Computers in Biology and Medicine*, 168 :107758, 2024.

Traver Hart, Megha Chandrashekhar, Michael Aregger, Zachary Steinhart, Kevin R. Brown, Graham MacLeod, Monika Mis, Michal Zimmermann, Amelie Fradet-Turcotte, Song Sun, et al. High-resolution crispr screens reveal fitness genes and genotype-specific cancer liabilities. *Cell*, 163(6) :1515–1526, 2015.

Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in neural information processing systems*, 33 :1010–1022, 2020.

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

Wei He, Helen Wang, Yanjun Wei, Zhiyun Jiang, Yitao Tang, Yiwen Chen, and Han Xu. Guidepro : a multi-source ensemble predictor for prioritizing sgrnas in crispr/cas9 protein knockouts. *Bioinformatics*, 37(1) :134–136, 2021.

Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville : Deep learning, 2018.

Florian Heigwer, Grainne Kerr, and Michael Boutros. E-crisp : fast crispr target site identification. *Nature Methods*, 11(2) :122–123, 2014.

Kasidet Hiranniramol, Yuhao Chen, Weijun Liu, and Xiaowei Wang. Generalizable sgrna design for improved crispr/cas9 editing efficiency. *Bioinformatics*, 36(9) :2684–2689, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8) :1735–1780, 1997.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging. In *Proceedings of the AAAI workshop on integrating multiple learned models*, volume 335, pages 77–83. Citeseer, 1998.

Ivo L Hofacker. Vienna rna secondary structure server. *Nucleic Acids Research*, 31(13) :3429–3431, 2003.

Lara Hoffmann, Ines Fortmeier, and Clemens Elster. Uncertainty quantification by ensemble learning for computational optical form measurements. *Machine Learning : Science and Technology*, 2(3) : 035030, 2021.

Patrick D. Hsu, David A. Scott, Joshua A. Weinstein, F. Ran, Silvana Konermann, Vineeta Agarwala, Yinqing Li, Eli J. Fine, Xuebing Wu, Ophir Shalem, et al. Dna targeting specificity of rna-guided cas9 nucleases. *Nature Biotechnology*, 31(9) :827–832, 2013.

Patrick D. Hsu, Eric S. Lander, and Feng Zhang. Development and applications of crispr-cas9 for genome engineering. *Cell*, 157(6) :1262–1278, 2014.

Johnny H Hu, Shannon M Miller, Maarten H Geurts, Weixin Tang, Liwei Chen, Ning Sun, Christina M Zeina, Xue Gao, Holly A Rees, Zhi Lin, et al. Evolved cas9 variants with broad pam compatibility and high dna specificity. *Nature*, 556(7699) :57–63, 2018.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning : An introduction to concepts and methods. *Machine Learning*, 110 :457–506, 2021.

Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew G Wilson. Dangers of bayesian model averaging under covariate shift. *Advances in Neural Information Processing Systems*, 34 :3309–3322, 2021.

Hongsheng Jin, Zongyao Li, Ruofeng Tong, and Lanfen Lin. A deep 3d residual cnn for false-positive reduction in pulmonary nodule detection. *Medical Physics*, 45(5) :2097–2107, 2018.

Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A. Doudna, and Emmanuelle Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *Science*, 337(6096) :816–821, 2012.

Xiang Jin Kang, Chiong Isabella Noelle Caparas, Boon Seng Soh, and Yong Fan. Addressing challenges in the clinical applications associated with crispr/cas9 technology and ethical questions to prevent its misuse. *Protein & Cell*, 8(11) :791–795, 2017.

Daniel Kermany, Kang Zhang, and Michael Goldbaum. Large dataset of labeled optical coherence tomography (oct) and chest x-ray images. *Mendeley Data*, 3(10.17632), 2018.

Daesik Kim, Sangsu Bae, Jeongbin Park, Eunji Kim, Seokjoong Kim, Hye Ryeong Yu, Jinha Hwang, Jong-Il Kim, and Jin-Soo Kim. Digenome-seq : genome-wide profiling of crispr-cas9 off-target effects in human cells. *Nature Methods*, 12(3) :237–243, 2015.

Daesik Kim, Sojung Kim, Sunghyun Kim, Jeongbin Park, and Jin-Soo Kim. Genome-wide target specificities of crispr-cas9 nucleases revealed by multiplex digenome-seq. *Genome Research*, 26(3) :406–415, 2016.

Hui Kwon Kim, Seonwoo Min, Myungjae Song, Soobin Jung, Jae Woo Choi, Younggwang Kim, Sangeun Lee, Sungroh Yoon, and Hyongbum Henry Kim. Deep learning improves prediction of crispr–cpf1 guide rna activity. *Nature Biotechnology*, 36(3) :239–241, 2018.

Hui Kwon Kim, Younggwang Kim, Sungtae Lee, Seonwoo Min, Jung Yoon Bae, Jae Woo Choi, Jinman Park, Dongmin Jung, Sungroh Yoon, and Hyongbum Henry Kim. Spcas9 activity prediction by deepspcas9, a deep learning–based model with high generalization performance. *Science Advances*, 5(11) : eaax9249, 2019.

Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. *Advances in Neural Information Processing Systems*, 33 :4163–4174, 2020a.

Nahye Kim, Hui Kwon Kim, Sungtae Lee, Jung Hwa Seo, Jae Woo Choi, Jinman Park, Seonwoo Min, Sungroh Yoon, Sung-Rae Cho, and Hyongbum Henry Kim. Prediction of the sequence-specific cleavage activity of cas9 variants. *Nature Biotechnology*, 38(11) :1328–1336, 2020b.

Bogdan Kirillov, Ekaterina Savitskaya, Maxim Panov, Aleksey Y Ogurtsov, Svetlana A Shabalina, Eugene V Koonin, and Konstantin V Severinov. Uncertainty-aware and interpretable evaluation of cas9–grna and cas12a–grna specificity for fully matched and partially mismatched targets with deep kernel learning. *Nucleic Acids Research*, 50(2) :e11–e11, 2022.

Benjamin P Kleinstiver, Michelle S Prew, Shengdar Q Tsai, Ved V Topkar, Nhu T Nguyen, Zongli Zheng, Andrew PW Gonzales, Zhuyun Li, Randall T Peterson, Jing-Ruey Joanna Yeh, et al. Engineered crispr-cas9 nucleases with altered pam specificities. *Nature*, 523(7561) :481–485, 2015.

Benjamin P. Kleinstiver, Vikram Pattanayak, Michelle S. Prew, Shengdar Q. Tsai, Nhu T. Nguyen, Zongli Zheng, and J. Keith Joung. High-fidelity crispr–cas9 nucleases with no detectable genome-wide off-target effects. *Nature*, 529(7587) :490–495, 2016.

Hiroko Koike-Yusa, Yilong Li, E.-Pien Tan, Martín Del Castillo Velasco-Herrera, Kosuke Yusa, et al. Genome-wide recessive genetic screening in mammalian cells with a lentiviral crispr-guide rna library. *Nature Biotechnology*, 32(3) :267–273, 2014.

Silvana Konermann, Mark D Brigham, Alexandro E Trevino, Julia Joung, Omar O Abudayyeh, Clea Barcena, Patrick D Hsu, Naomi Habib, Jonathan S Gootenberg, Hiroshi Nishimasu, et al. Genome-scale transcriptional activation by an engineered crispr-cas9 complex. *Nature*, 517(7536) :583–588, 2015.

Vasileios Konstantakos, Anastasios Nentidis, Anastasia Krithara, and Georgios Paliouras. Crispredict : a crispr-cas9 web tool for interpretable efficiency predictions. *Nucleic Acids Research*, 50(W1) : W191–W198, 06 2022a. ISSN 0305-1048.

Vasileios Konstantakos, Anastasios Nentidis, Anastasia Krithara, and Georgios Paliouras. Crispr–cas9 grna efficiency prediction : an overview of predictive tools and the role of deep learning. *Nucleic Acids Research*, 50(7) :3616–3637, 2022b.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Rudolf Kruse, Sanaz Mostaghim, Christian Borgelt, Christian Braune, and Matthias Steinbrecher. Multi-layer perceptrons. In *Computational intelligence : a methodological introduction*, pages 53–124. Springer, 2022.

Maurice Labuhn, Felix F Adams, Michelle Ng, Sabine Knoess, Axel Schambach, Emmanuelle M Charpentier, Adrian Schwarzer, Juan L Mateo, Jan-Henning Klusmann, and Dirk Heckl. Refined sgrna efficacy prediction improves large-and small-scale crispr–cas9 applications. *Nucleic Acids Research*, 46(3) : 1375–1385, 2018.

Sudarshan S Lakhawat, Naveen Malik, Vikram Kumar, Sunil Kumar, and Pushpender Kumar Sharma. Implications of crispr-cas9 in developing next generation biofuel : A mini-review. *Current Protein and Peptide Science*, 23(9) :574–584, 2022.

Benjamin Lambert, Florence Forbes, Alan Tucholka, Senan Doyle, Harmonie Dehaene, and Michel Dojat. Trustworthy clinical ai solutions : a unified review of uncertainty quantification in deep learning models for medical image analysis. *arXiv preprint arXiv :2210.03736*, 2022.

Cicera R. Lazzarotto, Nikolay L. Malinin, Yichao Li, Ruochi Zhang, Yang Yang, GaHyun Lee, Eleanor Cowley, Yanghua He, Xin Lan, Kasey Jividen, et al. Change-seq reveals genetic and epigenetic effects on crispr–cas9 genome-wide activity. *Nature Biotechnology*, 38(11) :1317–1327, 2020.

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

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance : Bayesian meta-learning for imbalanced and out-of-distribution tasks. *arXiv preprint arXiv :1905.12917*, 2019.

Jae Hoon Lee, Hyo Jun Won, Phuong Hoang Nguyen Tran, Sun-mi Lee, Ho-Youn Kim, and Je Hyeong Jung. Improving lignocellulosic biofuel production by crispr/cas9-mediated lignin modification in barley. *GCB Bioenergy*, 13(4) :742–752, 2021.

Jungjoon K Lee, Euihwan Jeong, Joonsun Lee, Minhee Jung, Eunji Shin, Young-hoon Kim, Kangin Lee, Inyoung Jung, Daesik Kim, Seokjoong Kim, et al. Directed evolution of crispr-cas9 to increase its specificity. *Nature Communications*, 9(1) :1–10, 2018.

Puping Liang, Yanwen Xu, Xiya Zhang, Chenhui Ding, Rui Huang, Zhen Zhang, Jie Lv, Xiaowei Xie, Yuxi Chen, Yujing Li, et al. Crispr/cas9-mediated gene editing in human tripronuclear zygotes. *Protein & Cell*, 6 (5) :363–372, 2015.

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

Jiecong Lin, Zhaolei Zhang, Shixiong Zhang, Junyi Chen, and Ka-Chun Wong. Crispr-net : A recurrent convolutional network quantifies crispr off-target activities with mismatches and indels. *Advanced Science*, 7(13) :1903562, 2020.

Jennifer Listgarten, Michael Weinstein, Benjamin P Kleinstiver, Alexander A Sousa, J Keith Joung, Jake Crawford, Kevin Gao, Luong Hoang, Melih Elibol, John G Doench, et al. Prediction of off-target activities for the end-to-end design of crispr guide rnas. *Nature Biomedical Engineering*, 2(1) :38–47, 2018.

Guanqing Liu, Yong Zhang, and Tao Zhang. Computational approaches for effective crispr guide rna design and evaluation. *Computational and Structural Biotechnology Journal*, 18 :35–44, 2020a.

Hao Liu, Yuduan Ding, Yanqing Zhou, Wenqi Jin, Kabin Xie, and Ling-Ling Chen. Crispr-p 2.0 : an improved crispr-cas9 tool for genome editing in plants. *Molecular Plant*, 10(3) :530–532, 2017.

Qiao Liu, Di He, and Lei Xie. Identifying context-specific network features for crispr-cas9 targeting efficiency using accurate and interpretable deep neural network. *bioRxiv*, page 505602, 2018.

Qiao Liu, Di He, and Lei Xie. 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, 2019.

Qiaoyue Liu, Xiang Cheng, Gan Liu, Bohao Li, and Xiuqin Liu. Deep learning improves the ability of sgrna off-target propensity prediction. *BMC Bioinformatics*, 21(1) :1–15, 2020b.

Xiaojian Liu, Yuanyuan Yang, Yan Qiu, Qiurong Ding, Yi Wang, et al. Seqcor : correct the effect of guide rna sequences in clustered regularly interspaced short palindromic repeats/cas9 screening by machine learning algorithm. *Journal of Genetics and Genomics*, 47(11) :672–680, 2020c.

Tyler J Loftus, Benjamin Shickel, Matthew M Ruppert, Jeremy A Balch, Tezcan Ozrazgat-Baslanti, Patrick J Tighe, Philip A Efron, William R Hogan, Parisa Rashidi, Gilbert R Upchurch Jr, et al. Uncertainty-aware deep learning in healthcare : a scoping review. *PLOS digital health*, 1(8) :e0000085, 2022.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv :1802.03888*, 2018.

Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search : A simple baseline for learnable test-time augmentation. In *Conference on Uncertainty in Artificial Intelligence*, pages 1308–1317. PMLR, 2020.

Hong Ma, Nuria Marti-Gutierrez, Sang-Wook Park, Jun Wu, Yeonmi Lee, Keiichiro Suzuki, Amy Koski, Dongmei Ji, Tomonari Hayama, Riffat Ahmed, et al. Correction of a pathogenic gene mutation in human embryos. *Nature*, 548(7668) :413–419, 2017.

Prashant Mali, John Aach, P. Benjamin Stranges, Kevin M. Esvelt, Mark Moosburner, Sriram Kosuri, Luhan Yang, and George M. Church. Cas9 transcriptional activators for target specificity screening and paired nickases for cooperative genome engineering. *Nature Biotechnology*, 31(9) :833–838, 2013.

Hakim Manghwar, Keith Lindsey, Xianlong Zhang, and Shuangxia Jin. Crispr/cas system : recent advances and future prospects for genome editing. *Trends in Plant Science*, 24(12) :1102–1125, 2019.

Roman C. Maron, Sarah Haggenmüller, Christof von Kalle, Jochen S. Utikal, Friedegund Meier, Frank F. Gellrich, Axel Hauschild, Lars E. French, Max Schlaak, Kamran Ghoreschi, et al. Robustness of convolutional neural networks in recognition of pigmented skin lesions. *European Journal of Cancer*, 145 :81–91, 2021.

Andrew P May, Peter Cameron, Alexander H Settle, Chris K Fuller, Matthew S Thompson, A Mark Cigan, and Joshua K Young. Site-seq : A genome-wide method to measure cas9 cleavage. 2017.

Bogdan Mazoure, Alexander Mazoure, Jocelyn Bédard, and Vladimir Makarenkov. Dunescan : a web server for uncertainty estimation in skin cancer detection with deep neural networks. *Scientific Reports*, 12 (1) :1–10, 2022.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artificial Intelligence*, 267 : 1–38, 2019.

Tessa G. Montague, José M. Cruz, James A. Gagnon, George M. Church, and Eivind Valen. Chopchop : a crispr/cas9 and talen web tool for genome editing. *Nucleic Acids Research*, 42(W1) :W401–W407, 2014.

Kristine Monteith, James L Carroll, Kevin Seppi, and Tony Martinez. Turning bayesian model averaging into bayesian model combination. In *The 2011 international joint conference on neural networks*, pages 2657–2663. IEEE, 2011.

Miguel A. Moreno-Mateos, Charles E. Vejnar, Jean-Denis Beaudoin, Juan P. Fernandez, Emily K. Mis, Mustafa K. Khokha, and Antonio J. Giraldez. Crisprscan : designing highly efficient sgrnas for crispr-cas9 targeting in vivo. *Nature Methods*, 12(10) :982–988, 2015.

Ali Haisam Muhammad Rafid, Md Toufikuzzaman, Mohammad Saifur Rahman, and M Sohel Rahman. Crisprpred (seq) : a sequence-based method for sgrna on target activity prediction using traditional machine learning. *BMC Bioinformatics*, 21(1) :1–13, 2020.

Alhassan Mumuni and Fuseini Mumuni. Data augmentation : A comprehensive survey of modern approaches. *Array*, 16 :100258, 2022.

Diana M. Munoz, Pamela J. Cassiani, Li Li, Eric Billy, Joshua M. Korn, Michael D. Jones, Javad Golji, David A. Ruddy, Kristine Yu, Gregory McAllister, et al. Crispr screens provide a comprehensive assessment of cancer vulnerabilities but generate false-positive hits for highly amplified genomic regionscrispr screens for the discovery of cancer vulnerabilities. *Cancer Discovery*, 6(8) :900–913, 2016.

Muhammad Naeem, Saman Majeed, Mubasher Zahir Hoque, and Irshad Ahmad. Latest developed strategies to minimize the off-target effects in crispr-cas-mediated genome editing. *Cells*, 9(7) :1608, 2020.

Anthony Newman, Lora Starrs, and Gaetan Burgio. Cas9 cuts and consequences ; detecting, predicting, and mitigating crispr/cas9 on-and off-target damage : techniques for detecting, predicting, and mitigating the on-and off-target effects of cas9 editing. *BioEssays*, 42(9) :2000047, 2020.

Cuong Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Uncertainty in model-agnostic meta-learning using variational inference. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3090–3100, 2020.

Hiroshi Nishimasu, Xi Shi, Soh Ishiguro, Linyi Gao, Seiichi Hirano, Sae Okazaki, Taichi Noda, Omar O Abudayyeh, Jonathan S Gootenberg, Hideto Mori, et al. Engineered crispr-cas9 nuclease with expanded targeting space. *Science*, 361(6408) :1259–1262, 2018.

Rui Niu, Jiajie Peng, Zhipeng Zhang, and Xuequn Shang. R-crispr : A deep learning network to predict off-target activities with mismatch, insertion and deletion in crispr-cas9 system. *Genes*, 12(12) :1878, 2021.

Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. `https://github.com/keras-team/keras-tuner`, 2019.

Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in biology and medicine*, 121 :103792, 2020.

Aidan R O'Brien, Gaetan Burgio, and Denis C Bauer. Domain-specific introduction to machine learning terminology, pitfalls and opportunities in crispr-based gene editing. *Briefings in Bioinformatics*, 22 (1) :308–314, 2021.

Martin Pacesa, Oana Pelea, and Martin Jinek. Past, present, and future of crispr genome editing technologies. *Cell*, 187(5) :1076–1100, 2024.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 : 2825–2830, 2011.

Hui Peng, Yi Zheng, Zhixun Zhao, Tao Liu, and Jinyan Li. Recognition of crispr/cas9 off-target sites through ensemble learning of uneven mismatch distributions. *Bioinformatics*, 34(17) :i757–i765, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Lutz Prechelt. Early stopping-but when ? In *Neural Networks : Tricks of the trade*, pages 55–69. Springer, 2002.

Holger Puchta and Friedrich Fauser. Gene targeting in plants : 25 years later. *International Journal of Developmental Biology*, 57(6-8) :629–637, 2013.

J. Ross Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3) :221–234, 1987.

Erik B Dam Raghavendra Selvan. Tensor networks for medical image classification. In *International Conference on Medical Imaging with Deep Learning – Full Paper Track*, July 2020. URL `https://openreview.net/forum?id=jjk6bxk07G`.

Rahul Rahaman et al. Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, 34 :20063–20075, 2021.

Md Khaledur Rahman and M Sohel Rahman. Crisprpred : a flexible and efficient tool for sgrnas on-target activity prediction in crispr/cas9 systems. *PLoS One*, 12(8) :e0181943, 2017.

Oleg Raitskin and Nicola J. Patron. Multi-gene engineering in plants with rna-guided cas9 nuclease. *Current Opinion in Biotechnology*, 37 :69–75, 2016.

Fidel Ramírez, Friederike Dündar, Sarah Diehl, Björn A Grüning, and Thomas Manke. deeptools : a flexible platform for exploring deep-sequencing data. *Nucleic Acids Research*, 42(W1) :W187–W191, 2014.

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

F. A. C. L. Ran, Le Cong, Winston X. Yan, David A. Scott, Jonathan S. Gootenberg, Andrea J. Kriz, Bernd Zetsche, Ophir Shalem, Xuebing Wu, Kira S. Makarova, et al. In vivo genome editing using staphylococcus aureus cas9. *Nature*, 520(7546) :186–191, 2015.

Benedikt Rauscher, Florian Heigwer, Marco Breinig, Jan Winter, and Michael Boutros. Genomecrispr—a database for high-throughput crispr/cas9 screens. *Nucleic Acids Research*, page gkw997, 2016.

Xingjie Ren, Zhihao Yang, Jiang Xu, Jin Sun, Decai Mao, Yanhui Hu, Su-Juan Yang, Huan-Huan Qiao, Xia Wang, Qun Hu, et al. Enhanced specificity and efficiency of the crispr/cas9 system with optimized sgrna parameters in drosophila. *Cell Reports*, 9(3) :1151–1162, 2014.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4) :801–814, 2018.

Tim GJ Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. Tractable function-space variational inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 35 :22686–22698, 2022.

Jeffry D Sander and J Keith Joung. Crispr-cas systems for editing, regulating and targeting genomes. *Nature biotechnology*, 32(4) :347–355, 2014.

Jonathan L Schmid-Burgk, Linyi Gao, David Li, Zachary Gardner, Jonathan Strecker, Blake Lash, and Feng Zhang. Highly parallel profiling of cas9 variant specificity. *Molecular Cell*, 78(4) :794–800, 2020.

Vivien AC Schoonenberg, Mitchel A Cole, Qiuming Yao, Claudio Macias-Treviño, Falak Sher, Patrick G Schupp, Matthew C Canver, Takahiro Maeda, Luca Pinello, and Daniel E Bauer. Crispro : identification of functional protein coding sequences based on genome editing dense mutagenesis. *Genome Biology*, 19(1) :1–19, 2018.

Skipper Seabold and Josef Perktold. Statsmodels : econometric and statistical modeling with python. *SciPy*, 7(1) :92–96, 2010.

Silvia Seoni, Vicnesh Jahmunah, Massimo Salvi, Prabal Datta Barua, Filippo Molinari, and U Rajendra Acharya. Application of uncertainty quantification to artificial intelligence in healthcare : A review of last decade (2013–2023). *Computers in Biology and Medicine*, page 107441, 2023.

Shiraz A. Shah, Susanne Erdmann, Francisco J. M. Mojica, and Roger A. Garrett. Protospacer recognition motifs : mixed identities and functional diversity. *RNA Biology*, 10(5) :891–899, 2013.

Ophir Shalem, Neville E. Sanjana, Ella Hartenian, Xi Shi, David A. Scott, Tarjei S. Mikkelsen, Dirk Heckl, Benjamin L. Ebert, David E. Root, John G. Doench, et al. Genome-scale crispr-cas9 knockout screening in human cells. *Science*, 343(6166) :84–87, 2014.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1214–1223, 2021.

Liyue Shen, Jimmy Zheng, Edward H Lee, Katie Shpanskaya, Emily S McKenna, Mahesh G Atluri, Dinko Plasto, Courtney Mitchell, Lillian M Lai, Carolina V Guimaraes, et al. Attention-guided deep learning for gestational age prediction using fetal brain mri. *Scientific Reports*, 12(1) :1–10, 2022.

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

Zeinab Sherkatghanad, Mohammadsadegh Akhondzadeh, Soorena Salari, Mariam Zomorodi-Moghadam, Moloud Abdar, U Rajendra Acharya, Reza Khosrowabadi, and Vahid Salari. Automated detection of autism spectrum disorder using a convolutional neural network. *Frontiers in neuroscience*, 13 :1325, 2020.

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

Zeinab Sherkatghanad, Moloud Abdar, Mohammadreza Bakhtyari, Paweł Pławiak, and Vladimir Makarenkov. Baytta : Uncertainty-aware medical image classification with optimized test-time augmentation using bayesian model averaging. *Knowledge-Based Systems*, page 114123, 2025.

Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, and Teh Ying Wah. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, 10(12) :e0144059, 2015.

Hari Shrawgi and Dilip Singh Sisodia. Convolution neural network model for predicting single guide rna efficiency in crispr/cas9 system. *Chemometrics and Intelligent Laboratory Systems*, 189 :149–154, 2019.

Ritambhara Singh, Cem Kuscu, Aaron Quinlan, Yanjun Qi, and Mazhar Adli. Cas9-chromatin binding information enables more accurate crispr off-target prediction. *Nucleic Acids Research*, 43(18) : e118–e118, 2015.

Ian M. Slaymaker, Linyi Gao, Bernd Zetsche, David A. Scott, Winston X. Yan, and Feng Zhang. Rationally engineered cas9 nucleases with improved specificity. *Science*, 351(6268) :84–88, 2016.

Jongwook Son and Seokho Kang. Efficient improvement of classification accuracy via selective test-time augmentation. *Information Sciences*, 642 :119148, 2023.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend : Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv :1710.10766*, 2017.

Manuel Stemmer, Thomas Thumberger, Maria del Sol Keyer, Joachim Wittbrodt, and Juan L. Mateo. Cctop : an intuitive, flexible and reliable crispr/cas9 target prediction tool. *PLOS ONE*, 10(4) :e0124633, 2015.

Florian Störtz and Peter Minary. crisprsql : a novel database platform for crispr/cas off-target cleavage assays. *Nucleic Acids Research*, 49(D1) :D855–D861, 2021.

Florian Störtz, Jeffrey Mak, and Peter Minary. picrispr : Physically informed features improve deep learning models for crispr/cas9 off-target cleavage prediction. *bioRxiv*, 2021.

Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24 :84–92, 2015. ISSN 1566-2535.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Xu Tang, Xuelian Zheng, Yiping Qi, Dengwei Zhang, Yan Cheng, Aiting Tang, Daniel F. Voytas, and Yong Zhang. A single transcript crispr-cas9 system for efficient genome editing in plants. *Molecular Plant*, 9(7) : 1088–1091, 2016.

Tara Basu Trivedi, Ron Boger, Govinda M Kamath, Georgios Evangelopoulos, Jamie Cate, Jennifer Doudna, and Jack Hidary. Crispr2vec : Machine learning model predicts off-target cuts of crispr systems. *bioRxiv*, 2020.

Shengdar Q. Tsai, Zongli Zheng, Nhu T. Nguyen, Matthew Liebers, Ved V. Topkar, Vishal Thapar, Nicolas Wyvekens, Cyd Khayter, A. John Iafrate, Long P. Le, et al. Guide-seq enables genome-wide profiling of off-target cleavage by crispr-cas nucleases. *Nature Biotechnology*, 33(2) :187–197, 2015.

Shengdar Q. Tsai, Nhu T. Nguyen, Jose Malagon-Lopez, Ved V. Topkar, Martin J. Aryee, and J. Keith Joung. Circle-seq : a highly sensitive in vitro screen for genome-wide crispr–cas9 nuclease off-targets. *Nature Methods*, 14(6) :607–614, 2017.

Masayuki Tsuneki. Deep learning models in medical image analysis. *Journal of Oral Biosciences*, 64(3) : 312–320, 2022.

Gaurav K. Varshney, Wuhong Pei, Matthew C. LaFave, Jennifer Idol, Lisha Xu, Viviana Gallardo, Blake Carrington, Kevin Bishop, MaryPat Jones, Mingyu Li, et al. High-throughput gene targeting and phenotyping in zebrafish using crispr/cas9. *Genome Research*, 25(7) :1030–1042, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Lukas Villiger, Julia Joung, Luke Koblan, Jonathan Weissman, Omar O Abudayyeh, and Jonathan S Gootenberg. Crispr technologies for genome, epigenome and transcriptome editing. *Nature Reviews Molecular Cell Biology*, 25(6) :464–487, 2024.

Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76 :89–106, 2021.

Prasoon Kumar Vinodkumar, Cagri Ozcinar, and Gholamreza Anbarjafari. Prediction of sgrna off-target activity in crispr/cas9 gene editing using graph convolution network. *Entropy*, 23(5) :608, 2021.

Daqi Wang, Chengdong Zhang, Bei Wang, Bin Li, Qiang Wang, Dong Liu, Hongyan Wang, Yan Zhou, Leming Shi, Feng Lan, et al. Optimized crispr guide rna design for two high-fidelity cas9 variants by deep learning. *Nature Communications*, 10(1) :1–14, 2019a.

Dong Wang, Jingbin Huang, Xinxia Wang, Yuan Yu, He Zhang, Yan Chen, Junjie Liu, Zhiguo Sun, Hao Zou, Duxin Sun, et al. The eradication of breast cancer cells and stem cells by 8-hydroxyquinoline-loaded hyaluronan modified mesoporous silica nanoparticle-supported lipid bilayers containing docetaxel. *Biomaterials*, 34(31) :7662–7673, 2013.

Fei Wang, Lianrong Wang, Xuan Zou, Suling Duan, Zhiqiang Li, Zixin Deng, Jie Luo, Sang Yup Lee, and Shi Chen. Advances in crispr-cas systems for rna targeting, tracking and editing. *Biotechnology Advances*, 37(5) :708–729, 2019b.

Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338 :34–45, 2019c.

Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In *Brainlesion : Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries : 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, pages 61–72. Springer, 2019d.

Jun Wang, Xi Xiang, Lars Bolund, Xiuqing Zhang, Lixin Cheng, and Yonglun Luo. 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, 2020a.

Jun Wang, Xiuqing Zhang, Lixin Cheng, and Yonglun Luo. An overview and metanalysis of machine and deep learning-based crispr grna design tools. *RNA Biology*, 17(1) :13–22, 2020b.

Lei Wang and Juhua Zhang. Prediction of sgrna on-target activity in bacteria by deep learning. *BMC Bioinformatics*, 20(1) :1–14, 2019.

Qi Wang and Herke Van Hoof. Doubly stochastic variational inference for neural processes with hierarchical latent variables. In *International Conference on Machine Learning*, pages 10018–10028. PMLR, 2020.

Tim Wang, Jenny J. Wei, David M. Sabatini, and Eric S. Lander. Genetic screens in human cells using the crispr-cas9 system. *Science*, 343(6166) :80–84, 2014.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11293–11302, 2019e.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3 (1) :1–40, 2016.

Laurence O. W. Wilson, Aidan R. O'Brien, and Denis C. Bauer. The current state and future of crispr-cas9 grna design tools. *Frontiers in Pharmacology*, 9 :749, 2018.

Brendan A Wintle, Michel A McCarthy, Chris T Volinsky, and Rodney P Kavanagh. The use of bayesian model averaging to better represent uncertainty in ecological models. *Conservation biology*, 17(6) : 1579–1590, 2003.

Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pages 788–798. PMLR, 2020.

Xi Xiang, Giulia I. Corsi, Christian Anthon, Kunli Qu, Xiaoguang Pan, Xue Liang, Peng Han, Zhanying Dong, Lijun Liu, Jiayan Zhong, et al. Enhancing crispr-cas9 grna efficiency prediction by data integration and deep learning. *Nature Communications*, 12(1) :1–9, 2021.

Li-Ming Xiao, Yun-Qi Wan, and Zhen-Ran Jiang. Attcrispr : a spacetime interpretable model for prediction of sgrna on-target activity. *BMC Bioinformatics*, 22(1) :1–17, 2021.

Han Xu, Tengfei Xiao, Chen-Hao Chen, Wei Li, Clifford A. Meyer, Qiu Wu, Di Wu, Le Cong, Feng Zhang, Jun S. Liu, et al. Sequence determinants of improved crispr sgrna design. *Genome Research*, 25(8) : 1147–1157, 2015.

Li Xue, Bin Tang, Wei Chen, and Jiesi Luo. Prediction of crispr sgrna activity using a deep convolutional neural network. *Journal of Chemical Information and Modeling*, 59(1) :615–624, 2018.

Ofir Yaish and Yaron Orenstein. Generating, modeling and evaluating a large-scale set of crispr/cas9 off-target sites with bulges. *Nucleic Acids Research*, page gkae428, 2024.

Ofir Yaish, Maor Asif, and Yaron Orenstein. A systematic evaluation of data processing and problem formulation of crispr off-target site prediction. *Briefings in Bioinformatics*, 23(5) :bbac157, 2022.

Jifang Yan, Dongyu Xue, Guohui Chuai, Yuli Gao, Gongchen Zhang, and Qi Liu. Benchmarking and integrating genome-wide crispr off-target detection and prediction. *Nucleic Acids Research*, 48(20) : 11370–11379, 2020.

Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics : Conference Series*, volume 1168, page 022022. IOP Publishing, 2019.

Oscar A. Zarate, Yiben Yang, Xiaozhong Wang, and Ji-Ping Wang. Boostmec : predicting crispr-cas9 cleavage efficiency through boosting models. *BMC Bioinformatics*, 23(1) :1–14, 2022.

Ali Zarei, Vahid Razban, Seyed Ebrahim Hosseini, and Seyed Mohammad Bagher Tabei. Creating cell and animal models of human disease by genome editing using crispr/cas9. *The Journal of Gene Medicine*, 21(4) :e3082, 2019.

Guishan Zhang, Zhiming Dai, and Xianhua Dai. C-rnncrispr : Prediction of crispr/cas9 sgrna activity using convolutional and recurrent neural networks. *Computational and Structural Biotechnology Journal*, 18 :344–354, 2020a.

Guishan Zhang, Zhiming Dai, and Xianhua Dai. A novel hybrid cnn-svr for crispr/cas9 guide rna activity prediction. *Frontiers in Genetics*, 10 :1303, 2020b.

Guishan Zhang, Tian Zeng, Zhiming Dai, and Xianhua Dai. Prediction of crispr/cas9 single guide rna cleavage efficiency and specificity by attention-based convolutional neural networks. *Computational and Structural Biotechnology Journal*, 19 :1445–1457, 2021.

Jianpeng Zhang, Yutong Xie, Yong Xia, and Chunhua Shen. Attention residual learning for skin lesion classification. *IEEE transactions on medical imaging*, 38(9) :2092–2103, 2019a.

Shixiong Zhang, Xiangtao Li, Qiuzhen Lin, and Ka-Chun Wong. Synergizing crispr/cas9 off-target predictions for ensemble insights and practical applications. *Bioinformatics*, 35(7) :1108–1115, 2019b.

Xiao-Hui Zhang, Louis Y. Tee, Xiao-Gang Wang, Qun-Shan Huang, and Shi-Hua Yang. Off-target effects in crispr/cas9-mediated genome engineering. *Molecular Therapy - Nucleic Acids*, 4 :e264, 2015.

Yu Zhang, Yahui Long, Rui Yin, and Chee Keong Kwoh. Dl-crispr : a deep learning method for off-target activity prediction in crispr/cas9 with data augmentation. *IEEE Access*, 8 :76610–76617, 2020c.

Zhong-Rui Zhang and Zhen-Ran Jiang. Effective use of sequence information to predict crispr-cas9 off-target. *Computational and Structural Biotechnology Journal*, 20 :650–661, 2022.

Yuexin Zhou, Shiyou Zhu, Changzu Cai, Pengfei Yuan, Chunmei Li, Yanyi Huang, and Wensheng Wei. High-throughput screening of a crispr/cas9 library for functional genomics in human cells. *Nature*, 509(7501) :487–491, 2014.