UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Technology

Prasoon Kumar Vinodkumar

Predicting Off-target Effects in CRISPR-Cas9 System using Graph Convolutional Network

Master's Thesis (30 EAP)

Bioengineering Curriculum

Supervisor:

Prof. Gholamreza Anbarjafari

Tartu 2021

Abstract

Predicting off-target effects in CRISPR-Cas9 system using Graph Convolutional Network

CRISPR-Cas9 is a powerful genome editing technology that has been widely applied in target gene repair and gene expression regulation. One of the main challenges for the CRISPR-Cas9 system is the occurrence of unexpected cleavage at some sites (off-targets) and predicting them is necessary due to its relevance in gene editing research. Very few deep learning models have been developed so far that predict the off-target propensity of single guide RNA (sgRNA) at specific DNA fragments by using artificial feature extract operations and machine learning techniques. Unfortunately, they implement a convoluted process that is difficult to understand and implement by researchers. This thesis focuses on developing a novel graph-based approach to predict off-target efficacy of sgRNA in CRISPR-Cas9 system that is easy to understand and replicate by researchers. This is achieved by creating a graph with sequences as nodes and by performing link prediction using Graph Convolutional Network (GCN) to predict the presence of links between sgRNA and off-target inducing target DNA sequences. Features for the sequences are extracted from within the sequences.

CERCS: B110 Bioinformatics, P175 Systems Theory, P176 Artificial Intelligence

Keywords: CRISPR-Cas9, Off-target, Synthetic Biology, Deep Learning, Graph Neural Network, Link Prediction

Kokkuvõte

Sihtmärkide ennustamine CRISPR / Cas9 süsteemis Graph Convolutional Network abil

CRISPR-Cas9 on võimas genoomi redigeerimise tehnoloogia, mida on laialdaselt kasutatud sihtgeenide parandamisel ja geeniekspressiooni reguleerimisel. CRISPR-Cas9 süsteemi üks peamisi väljakutseid on ootamatu lõhustumise ilmnemine mõnes kohas (väljaspool sihtmärke) ja nende ennustamine on vajalik, kuna see on asjakohane geenide redigeerimise uuringutes. Siiani on välja töötatud väga vähe süvaõppemudeleid, mis prognoosivad ühe suunava RNA (sgRNA) sihtmärgivälist kalduvust konkreetsete DNA fragmentide suhtes, kasutades tehisfunktsioonide väljavõtte toiminguid ja masinõppe tehnikaid. Kahjuks need rakendavad keerulist protsessi, mida on raske mõista ja rakendada teadlaste poolt. See lõputöö keskendub uudse graafikul põhineva lähenemisviisi väljatöötamisele, et prognoosida sgRNA sihtmärgivälist efektiivsust süsteemis CRISPR-Cas9, mida teadlased hõlpsasti mõistavad ja paljundavad. See saavutatakse, luues graaf, milles on järjestused sõlmpunktidena, ja viies lingi ennustamine läbi graafiku konvolutsioonivõrgu (GCN), et ennustada seoste olemasolu sgRNA ja sihtmärkidest väljaspool indutseerivate sihtmärk-DNA järjestuste vahel. Järjestuste omadused eraldatakse järjestuste seest.

CERCS: B110 Bioinformaatika, P175 Süsteemiteooria, P176 Tehisintellekt

Märksõnad: CRISPR-Cas9, Sihtväline, Sünteetiline bioloogia, Sügav õppimine, Graafiline närvivõrk, Lingi ennustamine

Contents

List of Figures

Abbreviations, constants, definitions

API - Application Programming Interface

AUC - Area Under Curve

Cas9 - CRISPR-associated Protein 9

CRISPR - Clustered Regularly Interspaced Short Palindromic Repeats

DL - Deep Learning

DNA - Deoxyribonucleic Acid

GCN - Graph Convolutional Network

GNN - Graph Neural Network

ML - machine learning

NOT - target sequences that did not induce off-targets

OT - off-target inducing target sequences

PAM - Protospacer Adjacent Motif

PRC - Precision-Recall Curve

ROC - Receiver Operating Characteristic

SGD - Stochastic Gradient Descent

sgRNA - single guide RNA

Acknowledgement

The completion of this study could not have been possible without the expertise of my thesis supervisor, Prof. GHOLAMREZA ANBARJAFARI. It is a genuine pleasure to express my sincere gratitude for his timely advice, intellectual comments, guidance, encouragement and enthusiasm, which helped me in shaping and reshaping this valuable piece of work.

1 Introduction

Genome engineering is the ability to engineer biological systems that allows the modification of genome and transcription products on target sites. Proteins including zinc fingers [1], [2] and transcription activator-like effectors (TALEs) [3], [4] are the first gene editing molecules but demand strenuous engineering of multiple proteins for diverse genomic sequences and might introduce superfluous genetic elements including drug resistant markers, bacterial origins of replication and multiple cloning sites [5], [6].

The RNA-guided Cas9 nuclease from Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) adaptive immune system, based on bacterial adaptive immune system, is a powerful genome editing technology that has been widely used in target gene repair and gene expression regulation. Cas9, a nuclease from the type II system, found in *Streptococcus pyogenes*, is most widely used [7]–[9].

CRISPR-Cas9 system can be conditioned to precise sites by providing three important components in gene editing process:

1. sgRNA (sequence of 20 nucleotides in length) needs to be complementary with its targeting genome sequence,
2. PAM (3 nucleotides motif on the target sequence and a prerequisite for Cas9 protein cleavage) needs to be located around the target site, and,
3. Cas9 protein cleaves the target DNA at the site, three bases upstream of PAM, under the guidance of sgRNA sequence.

The most common PAM type is NGG, where N represents any base of Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) [10], [11]. CRISPR-Cas9 genome editing has been implemented in plants [12], bacteria [13] and mammals [14].

1.1 Motivation

One of the main challenges for CRISPR-Cas9 system is the occurrence of unexpected cleavage at some sites, termed as ”off-targets” [15], [16]. During CRISPR-Cas9 gene editing, sgRNA can influence other regions resulting in unintended cleavage of DNA sequence. Mismatches between 20 nucleotides of sgRNA in the PAM-distal end and target sequences can generate off-target effects due to the sequence homology of the target loci. Off-target mutations could lead to major problems when applying CRISPR-Cas9 gene editing, as they could cause double-stranded breaks (DSB) in DNA by resulting in loss of gene functions as shown in figure 1.1.

The focus of study is to reduce these off-target mutations by accurately predicting them in CRISPR-Cas9 gene editing. With the rapid expansion of off-target data, the existing methods are difficult to implement in extracting features and cannot satisfy enough accuracy in predicting off-target mutations at the gene editing level [17]. Very few deep learning (DL) models like DeepCRISPR [18], CNN_Std [19], AttnToMismatch_CNN [20] and CnnCrispr [21] have used DL algorithms to predict off-target efficacy of sgRNA but implemented a complex process of feature pre-processing that is difficult to understand and implement by researchers.

1.2 Goal

The purpose of this thesis is to introduce a novel graph-based approach in predicting the off-target mutations in CRISPR-Cas9 system. The main objectives of this research are:

1. To develop a graph-based approach for off-target prediction in CRISPR-Cas9 system that is easy to understand and implement by researchers,
2. To use a powerful neural network model for performing representation learning of network graphs created from off-target dataset,
3. To provide features for sequences (nodes) in the graph by extracting sequence-based features,
4. To handle the imbalance issue in the off-target dataset, and,
5. To make use of python library for machine learning that enables researchers to easily identify patterns and implement graph machine learning.

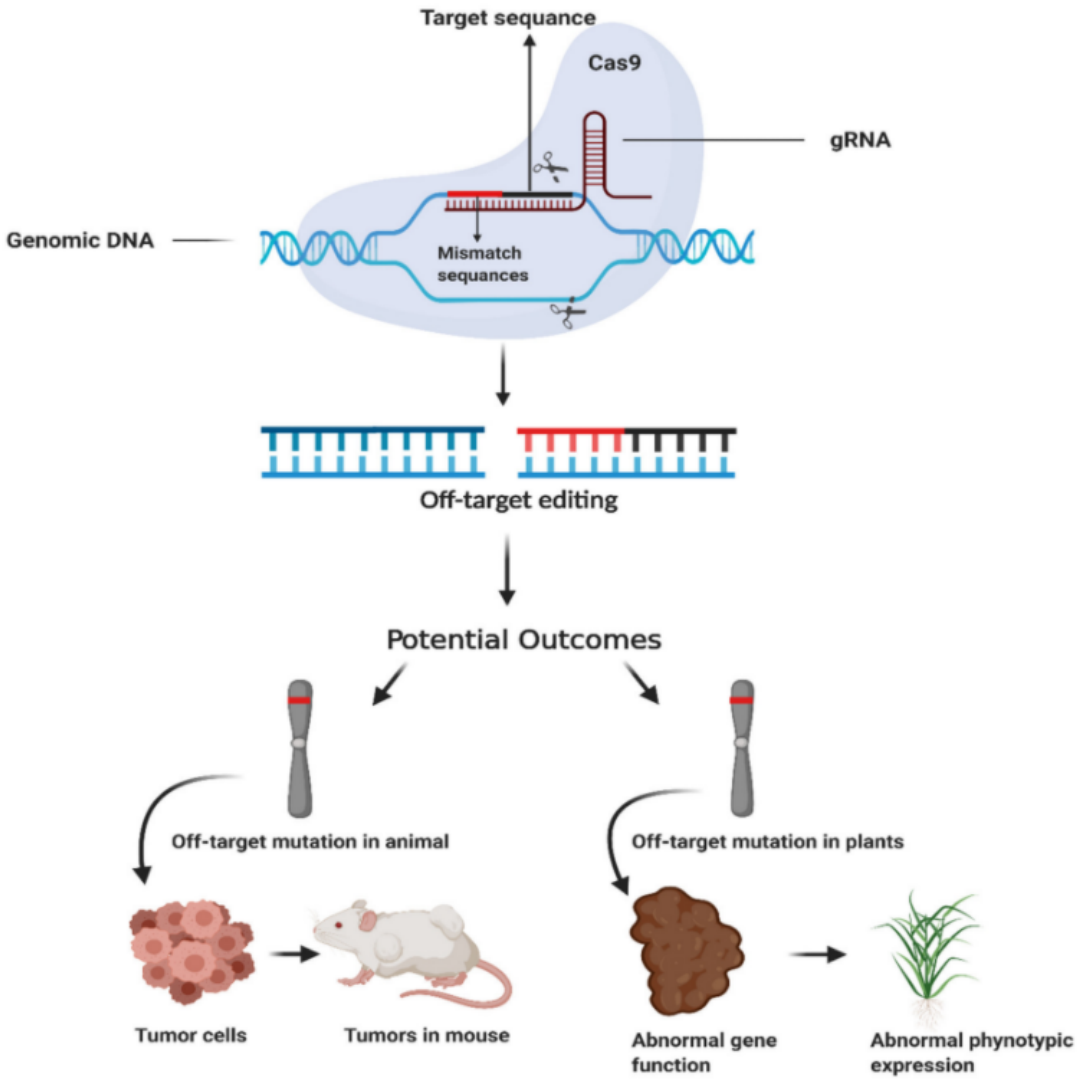


Figure 1.1: Off-target mutations in CRISPR-Cas9 gene editing [22]

1.3 Contributions

To implement an easy and efficient method in predicting off-target mutations in CRISPR-Cas9 gene editing, this study proposes a graph-based approach that is easy to understand and implement by researchers. The main contribution of this work can be summarized as follows:

- This study makes use of the off-target dataset from CnnCrispr [21], created by the authors of DeepCRISPR [18].
- Many in-built functions provided by StellarGraph [23] API are used to create network graphs from off-target dataset and to perform link prediction.
- A data science approach of using DL algorithm is taken to perform link prediction on the created network graph

- In contrast to previous off-target predicting DL models that used Convolutional Neural Network (CNN), this study makes use of Graph Convolutional Network (GCN) [24] model to perform unsupervised learning on the off-target network graph.

1.4 Outline

In this chapter, a brief introduction about CRISPR-Cas9 gene editing and off-target mutations in CRISPR-Cas9 system has been discussed. In the next Chapter, the discussion is about different methods developed by researchers to predict off-target mutations and about graph analysis, that gives an overview of network graphs, Graph Convolutional Network (GCN) and link prediction method. Chapter 3 provides a detailed review of architectures and performances of the previously developed DL models to predict off-target efficacy of sgRNA. Chapter 4 provides details about creating network graphs from off-target dataset, extracting sequence-based features, handling data imbalance using cluster data sampling and performing link prediction using GCN model. Chapter 5 contains the results and discussion of this research. Finally in chapter 6, a concluding statement to this work is provided.

2 Background

In this chapter, a detailed analysis of different off-target predicting methods developed by researchers will be discussed in detail in section 2.1. Section 2.2 will briefly explain about graph analysis by introducing graph neural network and link prediction.

2.1 Off-target prediction in CRISPR-Cas9 system

Predicting off-target mutations is necessary due to its relevance in gene editing research. Many prediction methods have been developed to predict the off-target propensity of single guide RNA (sgRNA) at specific DNA fragments using artificial feature extraction operations and machine learning (ML) techniques [11], [18]–[21], [25], [26]. Algorithm-based computational approaches and in vitro/vivo biochemical assays are two different methods methods developed by researchers that can be categorized into biased and unbiased detection methods. Figure 2.1 gives an overview of different off-target predicting methods implemented by researchers.

2.1.1 Unbiased off-target detection methods

Off-target effects on the whole genomic level can be detected by unbiased detection methods, which are categorized into in vitro genome-wide assays and in vivo genome-wide assays.

2.1.1.1 In vitro genome-wide assays

Digested genome sequencing (Digenome-seq) [27] is a widely used method to detect genome-wide off-target sites at which insertions or deletions were induced with frequencies below 0.1%. It is a robust, unbiased, cost-effective and sensitive method. Digenome-seq requires high read depth and cannot be applied for screening more sgRNA.

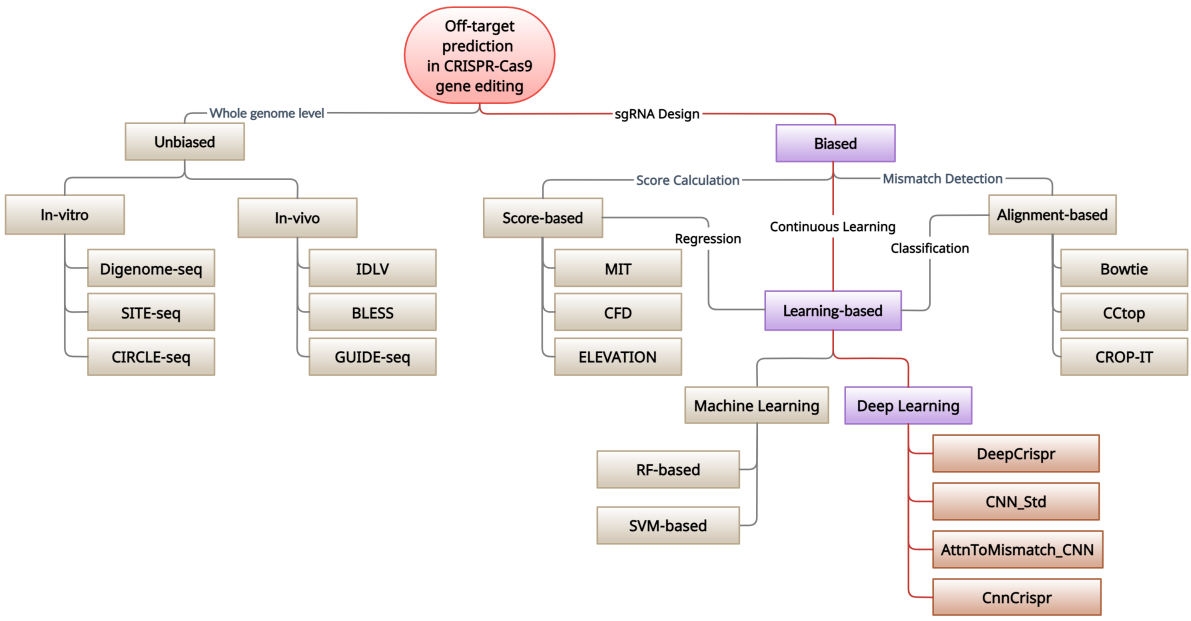


Figure 2.1: Overview of different methods implemented in predicting off-target mutations in CRISPR-Cas9 gene editing

Selective enrichment and identification of tagged genomic DNA ends by Sequencing (SITE-seq) [28] is a technique in which all the Cas9 cleavage sites are mapped and requires minimum next-generation sequencing (NGS) read depth on illumina platform compared to Digenome-seq.

Circularization for in vitro reporting of cleavage effects by sequencing (CIRCLE-seq) [29] can detect off-target sites that are diminished due to Single Nucleotide polymorphism (SNP) and does not need reference genome. For CIRCLE-seq, enormous amount of genomic DNA is required in circularization step restricting its application in off-target detection. Inadequacy of proteins associated with higher-order chromatin structure causes detecting cleavage sites more than detected in live cells using in-vivo methods.

Many off-targets identified by in vitro detection methods could not be refurbished in vivo.

2.1.1.2 In vivo genome-wide assays

Cell-based genome-wide assays are developed to identify real off-targets. Integrase Defective Lentiviral Vectors (IDLV) [30] is the first tool developed to detect genome-wide off-targets in zinc finger nucleases (ZFN) by getting shipped along with the gene-editing system and by integrating with the DSB. Linear double stranded IDLV genomes present in the nucleus of

transduced cells get ligated into DSB by non-homologous end joining (NHEJ), tagging the undetectable DSB. Dissemination of IDLV can be identified by carrying out PCR which results in identifying the off-target effects. IDLV has an off-target identifying efficiency of 0.1% and cannot evaluate various off-target sites [31].

Chromatin immunoprecipitation and high throughput sequencing (Chip-seq) [32] allows detection of off-target sites by binding DNA-binding proteins and histone modifications at base-pair resolution. Chip-seq uses endonuclease dead Cas9 (dCas9) and sgRNA complex instead of Cas9-gRNA complex which modifies the specificity in identifying off-targets [33], [34].

Breaks labelling, enrichments on streptavidin and next-generation sequencing (BLESS) [35] is adaptable, precise and significant in identifying DSB when compared with the other techniques and can also be used to detect endogenous and exogenous DSB. BLESS uses biotinylated DNA to label genomic DNA which allows high-specificity enrichment of samples on streptavidin beads, mapping sequence-based DSB to nucleotide resolution. BLESS depends upon a reference genome and identifies off-targets only during the "labelling" stage [36].

In Genome-wide unbiased identification of DSB enabled by sequencing (GUIDE-seq) [36], the double stranded oligodeoxynucleotides (dsODN) are integrated in DSB, which is later pursued by NHEJ DNA repair pathway. Off-target effects can be identified by amplifying the integrated dsODN. GUIDE-seq does not prevent the cytotoxic effects produced during the transmission of exogenous tag dsODN [37] and employs multiple PCRs to amplify regions of interest.

Linear amplification-mediated high-throughput genome-wide sequencing (LAM-HTGTS) [38] detects genome-wide "prey" DSB to a fixed "bait" DSB by identifying chromosomal translocation in cultured mammalian cells. This detects genome-wide recurrent DSB made by off-target activity of engineered nucleases. LAM-HTGTS is limited to identifying DSB that translocate as translocation events due to DSB are very rare when compared to DSB that arise due to local deletions and insertions.

2.1.2 Biased off-target detection methods

Methods include careful designing of sgRNA for definite targeting of CRISPR-Cas9 system by predicting off-target effects and by minimizing off-target effects using in silico methods by generating ample data on off-target sites using CRISPR systems under different strategies [39]–[41]. In silico methods include algorithmic-based models that are categorized into scoring-based models and alignment-based models.

2.1.2.1 Alignment-based Methods

Conventional algorithmic models, in which off-target effects are detected using sequence homology by aligning sgRNA to the reference genome [22]. Bowtie [42] can detect upto one mismatch by aligning DNA sequence reads to larger genomes.

CRISPR-Cas9 target online predictor (CCTop) [43] uses Bowtie [42] and can predict up to five mismatches. CCTop calculates an aggregate score using a hypothetical formula for target sites by focusing more on the position and mismatch counts.

CRISPR-Cas9 Off-target Prediction and Identification Tool (CROP-IT) [44] aligns the target sequence with the reference genome, divides the sequence into three segments and grades the sequence with different weights. CROP-IT utilizes whole-genome chromatin state information and predicts both Cas9 binding as well as cleavage sites.

Cas-OFFinder [45] searches for potential off-target sites and allows variations in PAM sequences recognized by Cas9. It is not limited by the number of mismatches.

CasOT [46], an on-target predicting tool, can detect up to 6 mismatches from 12 nucleotides closer to the PAM region and can be identify the region of off-targets as intron or exon.

Most of these models can detect upto only one mismatch and are prone to experimental variation due to the cutting efficiency variation of sgRNA.

2.1.2.2 Score-based Methods

Algorithmically designed models that calculate scores and ranks sgRNA based on the detection of off-target effects are developed. MIT (Hsu-Zhang) score [11] is developed to identify potential off-target sites for every sgRNA by calculating a weight matrix depending on mismatch positions in target 20-base pair (bp) and by considering mean distance between two bases of mismatches and number of mismatches.

Cutting frequency determination (CFD) score [25] uses GUIDE-seq [36] and is integrated with sgRNA designing tools like GuideScan [47] and CRISPOR [48]. CFD is considered as a ubiquitous score for off-target evaluation by multiplying the frequency of bases in each position of the sgRNA spacer sequence.

CRISPR target Assessment (CRISTA) [49] is an algorithmic model that uses machine learning (ML) to identify the propensity of genomic sites to be cleaved by a given sgRNA and shows that occurrences of bulges should be included in the prediction process.

Researchers from Microsoft and Broad Institution developed Elevation [26], a two-layer regression model that predicts the individual off-target score for sgRNA as well as aggregate score for sgRNAs. Elevation considers the sequences as well as the epigenetic information of the DNA and integrates it into a website for further research. Elevation method demonstrated to be an efficient tool in identifying all the potential off-target sites and identifies off-target sites for human exome (GRCh38) but does not work with different organisms [22].

These Models only calculate scores and are vulnerable to experimental variation due to the variation in cutting efficiency of sgRNA.

2.1.3 Learning-based models for off-target prediction

Continuous learning-based prediction models are required for effective prediction of off-targets due to variation in cutting efficiency of sgRNA. ML, core of artificial intelligence (AI) has been gradually applied to sgRNA off-target activity prediction [50], off-target site prediction [26] and sgRNA design optimization [51]. For example, CFD [25] uses the support vector machine (SVM) classifier to select the subset of features from 586 features. CRISPRpred [52] uses SVM

to predict on-target activity. WU-CRISPR [53] uses SVM model to select functionally active sgRNAs for all known genes in the human and mouse genomes. To predict sgRNA activity for prokaryotes, gradient-boosting regression (GBR) tree has been used [54]. ML models like CRISPRater [55] and CRISPR-Scan [56] are trained by a simple linear regression model.

Many ML models based on sequence features were used to calculate scores of off-target effects. Based on these scores, the performance of sgRNA are ranked. sgRNA with high cleavage efficiency and low off-target propensity are selected for performing CRISPR-Cas9 gene editing. Traditional ML algorithms have been gradually applied to predict off-target sites in score-based and alignment-based models. However, these algorithms cannot take raw features from large, annotated datasets and use them to identify hidden patterns buried inside these datasets. Most of these models achieved an average performance for sgRNA design as the feature extraction process is labour-intensive. Also, these models performed well only on a training dataset but not on testing dataset [17].

Compared to traditional ML models, deep learning (DL) models offer valuable analysis of experimental data, learn complex patterns and extract important features from large datasets which produce scientific support for bioinformatics. DL is the current hotspot in the field of ML that performs training and test tasks using neural network architectures [57]. DL models are packaged into frameworks like TensorFlow [58], Pytorch [59] and Keras [60] which enable researchers to focus on building and training neural networks without any difficulty [61].

Sequence-based DL models have achieved many performance breakthroughs in genomics research when compared to many classical ML models [57]. For example, AutoBioSeqpy [62] used CNN and bi-directional long short-term memory (biLSTM), a sub-class of Recurrent Neural network (RNN), to classify biological sequences of proteins, DNA and RNA. Another example could be DeepCRISPRpf1 [63], which incorporated chromatin accessibility in a DL framework based on CNN to predict the activity of AsCpf1 sgRNA.

In CRISPR-Cas9 system, many sequence-based DL models have predicted on-target efficacy of sgRNA. CRISPRLearner [64] used CNN model to learn sequence determinants and predicted on-target cleavage efficiency of sgRNA. DeepCRISPRas9 [65], a DL framework based on one convolution layer, predicted sgRNA activity in human cells and was used for de-

signing genome-scale CRISPRi and CRISPRa libraries targeting human and mouse genomes. CNN_5layers [66] predicted on-target activity of sgRNA for wild type and mutant Cas9 in prokaryotes using CNN. DeepHF [67], constructed using biLSTM, measured sgRNA activity for three SpCas9 variants and obtained indel rates for 20,000 genes.

Very few DL models like DeepCRISPR [18], CNN_Std [19], AttnToMismatch_CNN [20] and CnnCrispr [21] have been developed by researchers to measure the off-target propensity of sgRNA in CRISPR-Cas9 system due to the limited availability of off-target data. All these models have used CNN to predict sgRNA off-target activity by implementing automatic recognition of sequence features. However, these models implemented a complex process of feature extraction and off-target prediction, that is difficult to be understood and replicated by researchers.

2.2 Graph Analysis

2.2.1 Network Graph

Graphs are set of non-euclidean data structures that can model a set of objects (as nodes) and their relationships (as edges). A graph (G) is a pair (N, E) specified by the set of nodes (N) and set of edges (E). Graphs are used to denote large number of systems and many underlying relationships in fields of proteomics [68], image analysis [69], scene description [70], [71], software engineering [72], [73] and natural language processing [74]. For example, graphs are used to estimate probability of chemical compound causing some diseases by modeling the chemical compound as a graph, the atoms in the chemical compounds as nodes and the chemical bonds linking the atoms as edges [75]. Graphs are also used to capture interactions between bio-molecules like RNA, DNA and proteins [76] and to represent transcription networks of an organism.

2.2.2 Graph Convolutional Network (GCN)

Traditional ML models, like SVM, work with graphs using a pre-processing phase which converts graph information to simpler representation, like vector of reals. This could result in loss of important information and could lead to unpredictable performance of the model [75]. Standard neural networks, like CNN and RNN, cannot handle graph inputs efficiently as they stack features of the nodes in specific order which is very redundant when computing. Graph cannot

be traversed by all the possible orders of the input by standard neural networks as they consider dependency information of edges between two nodes as a feature of nodes.

Graphs need to be traversed by all the possible orders of the input. Neural networks that operate on graphs were first introduced in 2005 [77]. Graph Neural Networks (GNN) are deep learning methods that can be used to collectively aggregate information from graph structure based on CNN and graph embeddings. GNN models are considered to be more efficient graph analysis when compared to standard neural networks. They can produce output that is invariant to the input order of the nodes, propagate the dependency information by updating the hidden states of nodes as a weighted sum of states of their neighborhoods and can generate graphs from non-structural data [78], [79]. GNN models can work with node-focused applications without any pre-processing steps. GNN models have been applied in biological systems for calculating molecular fingerprints in computer-aided design [80], protein interface prediction and protein-protein interaction network [81], breast cancer sub-type classification [82] and poly-pharmacy side effects prediction [83].

There are several variants of GNN models that learn representations with high quality by making modifications in the propagation step. This step obtains the hidden states of nodes in a graph. Convolution operation in propagation step is a spectral approach which is defined by calculating spectral decomposition of graph laplacian in the fourier domain [78].

Graph Convolutional Network (GCN) [24], a convolution-like propagation rule on graphs, was introduced by using a single weight matrix per layer that dealt with varying node degrees through an appropriate normalization of the adjacency matrix. GCN alleviates over-fitting problem with very wide node degree distributions by limiting the layer-wise convolution operation and uses original graph structure to denote relations between nodes as shown in figure 2.2.

GCN is a powerful neural network architecture for deep learning on graph G,

$$G = (V, E) \tag{2.1}$$

where, V is the vertices/nodes and E is the edges/links in a graph.

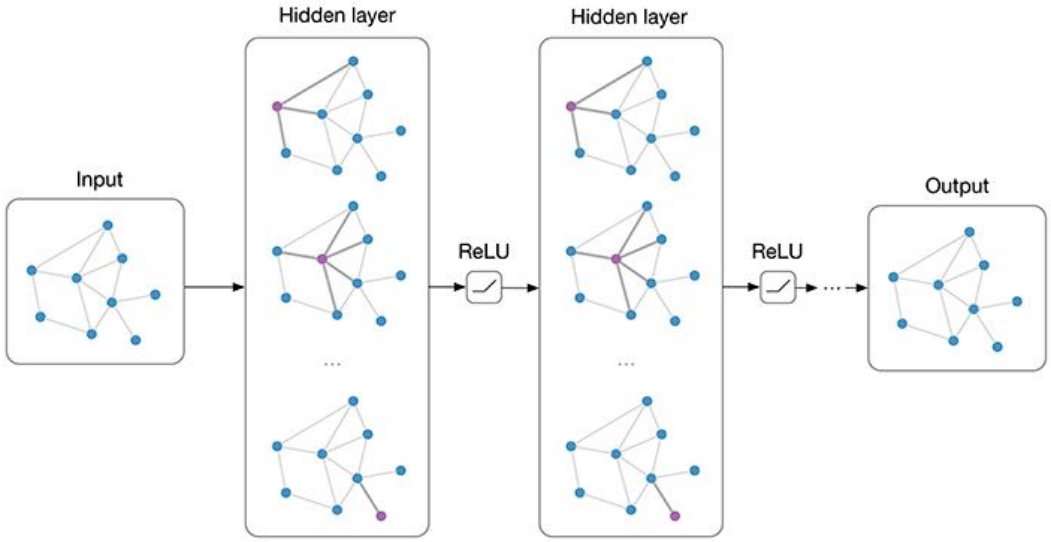


Figure 2.2: Network architecture of Graph Convolutional Network (GCN) model [24]

GCN takes the matrix representation of input feature matrix *(X)* and adjacency matrix *(A)* as inputs. Input feature matrix *(X)* is described as,

$$X = N * F^0 \tag{2.2}$$

where, *N* is the number of nodes and F^0 is the number of input features for each node.

Adjacency Matrix *(A)* is the matrix representation of the graph described as,

$$A = N * N \tag{2.3}$$

where, *N* is the number of nodes in the graph.

A hidden layer in the GCN can be described as,

$$H^i = f(H^{i-1}, A) \tag{2.4}$$

where, H^0 is the input feature matrix and f is the propagation. Each hidden layer *H*, represents the $N \times F$ of the feature matrix, with each row being a feature representation of a node. Using the propagation rule f at every layer, these features are aggregated to form the next layer's features to make them increasingly abstract at each consecutive layer.

The adjacency matrix *(A)* is then transformed by adding it with an identity matrix *(I)* to add a self-loop to each node, as the aggregated representation of a node does not include its own features unless the nodes have a self-loop.

$$\hat{A} = A + I \tag{2.5}$$

Node degree (number of edges connected to the nodes) of the transformed adjacency matrix *($\hat{A}$)* is calculated. The transformed adjacency matrix *($\hat{A}$)* and feature matrix *(X)* are normalized by the computed node degree to avoid any issues for stochastic gradient algorithms that are sensitive to the change of scale of feature vectors when performing the matrix multiplication of the adjacency matrix and the feature matrix. Thus, the propagation rule would look like this,

$$f(X, A) = z(D^{-\frac{1}{2}} \times \hat{A} \times D^{-\frac{1}{2}} \times X) \tag{2.6}$$

z are non-linear function (ReLu function). Multiplying the normalized feature matrix *(X)* with transformed adjacency matrix *($\hat{A}$)* normalized with diagonal node degree *($D^{-\frac{1}{2}}$)* will take the average of neighbouring node features.

2.2.3 Link Prediction

Graph analysis using ML can be divided into two main classes, called nodes classification and link prediction. Link prediction predicts an attribute of links or edges in a graph by predicting whether a link or edge that is not in a graph should exist using binary classification. Features of the nodes are important in identifying the link between the pair of sequences. The train and test sets of links and the corresponding graphs (without these links) are prepared to train a link prediction model. StellarGraph API [23] provides many in-built classes such as “EdgeSplitter” and “FullBatchLinkGenerator” which can be used to work on nodes and links for link prediction.

2.2.3.1 EdgeSplitter Function

“EdgeSplitter” class, provided by StellarGraph, is used to randomly sample the edges by keeping all the sequences in the train and test set, instead of taking a subset of sequences [23].

$$edge\_splitter\_test = EdgeSplitter(G) \tag{2.7}$$

This will return a train graph (that shows whether a link should exist between two sequences)

for training the model and a test graph for evaluating the performance of the model. Both the train graphs and test graphs will have the same number of nodes. The number of links between the nodes will differ as some of the links will be sampled for training and testing the classifier.

2.2.3.2 *FullBatchLinkGenerator* Function

"FullBatchLinkGenerator" class, provided by StellarGraph, is used to create link generators for the train and test link examples to the model. The "flow" method supplies the links as a list of nodes. The link generators will feed the list of nodes obtained from "flow" method and feed it to the Keras model, along with the corresponding binary labels which indicate the nodes true or false links in the form of features array and sparse adjacency matrix.

$$train\_gen = FullBatchLinkGenerator(G\_train, method = "gcn") \tag{2.8}$$

$$train\_flow = train\_gen.flow(edge\_ids\_train, edge\_labels\_train) \tag{2.9}$$

Final link classification layer takes a pair of node embeddings produced by the GCN model as input and produces corresponding link embeddings by applying a binary operator and passes it through a dense layer. The input and output tensors of the GCN model for link prediction are exposed using the $GCN.in\_out\_tensors$ method provided by StellarGraph [23].

$$x\_inp, x\_out = gcn.in\_out\_tensors() \tag{2.10}$$

The *x_out* value is a TensorFlow tensor that holds a 16-dimensional vector for the nodes requested when training or predicting. Predictions are reshaped from (X,1) to (X). The prediction layers are stacked into a keras model and the loss is specified [23].

2.3 Summary

In this chapter, biased, unbiased and learning-based methods to predict off-target mutations in CRISPR-Cas9 system were briefly discussed. This chapter has also introduced and provided more information about graph convolutional network (GCN) [24] and link prediction. The next chapter will provide a detailed analysis on different DL models used for off-target prediction.

3 Related Works

In this chapter, the network architecture, feature extraction and performance of sequence-based DL models developed to predict off-target efficacy of sgRNA will be discussed and reviewed in detail.

3.1 DeepCRISPR

3.1.1 Network Architecture

DeepCRISPR [18] applied the rules of auto-encoders to predict off-target propensity and target cleavage site of sgRNA by extracting epigenetic and sequence features of DNA. The architecture of DeepCRISPR model consists of:

- Two pre-trained deep convolutional denoising neural network (DCDNN)-based encoders (used as parent network) to train unlabelled sequences in unsupervised manner to learn an efficient feature representation of the unlabelled data using encoding and decoding which will be fitted for building the model,
- One merged layer, and,
- CNN layers to predict efficacy of sgRNA.

The training process learned the weights of CNN network and tuned the weights of parent network, creating two different "baby networks" and their weights are used for predicting off-target efficacy of sgRNA as shown in figure 3.1.

3.1.2 Feature Extraction

20-bp of unlabelled sgRNA sequences with NGG PAM extracted from coding and non-coding regions with different epigenetic information curated from 13 human cell types are given as

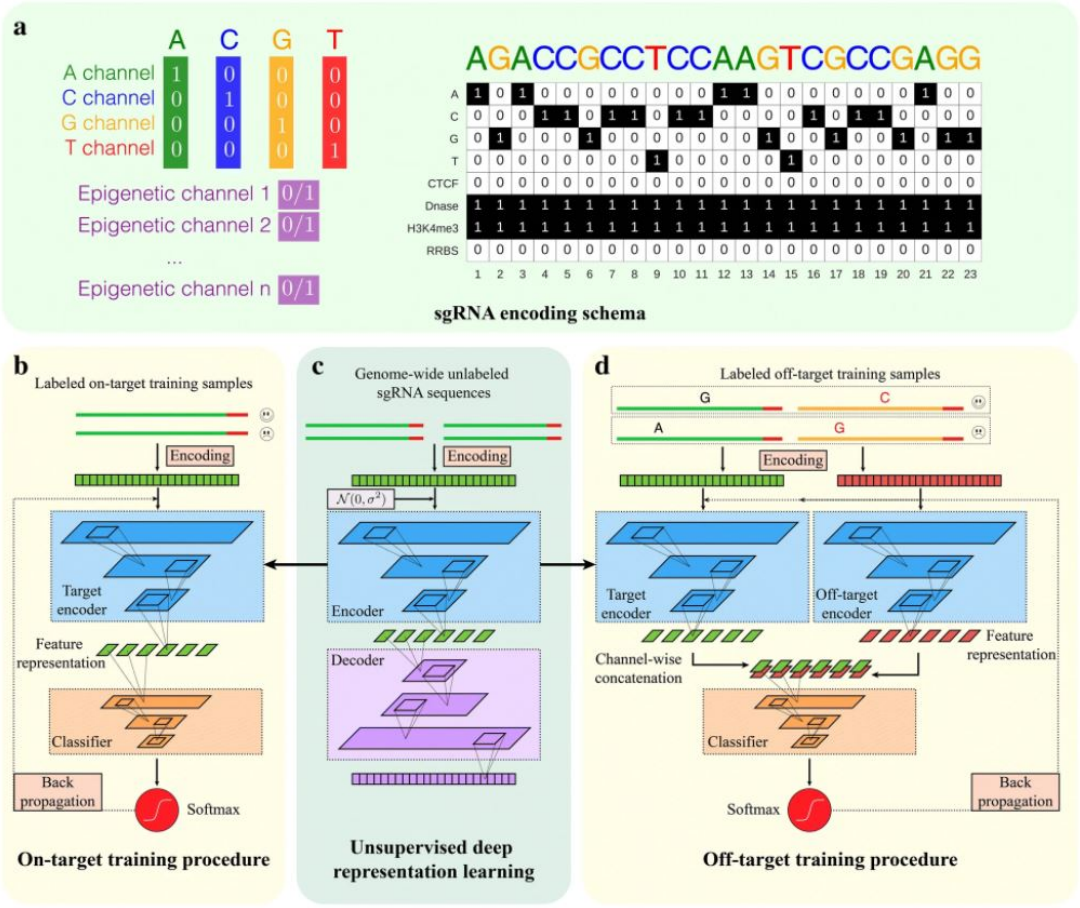


Figure 3.1: Network architecture of DeepCRISPR model [18]

input for the model. sgRNA sequences and their possible off-target sequences, treated as sequence pairs, are encoded using two-part encoding. These sequence-pairs are fit into parent network for feature extraction and the outputs of this network are combined and provided as input for CNN.

3.1.3 Dataset

Dataset used for this research consists of human sgRNA whole genome off-target data detected by Digenome-seq [27], IDLV [30], BLESS [35], GUIDE-seq [36] and HTGTS [38]. The dataset consists of 30 sgRNA from two different cell types (HEK293 cell line with 18 sgRNA and K562T cell line with 12 sgRNA) with maximum of six nucleotide mismatches using Bowtie. As the off-target dataset is heavily imbalanced, bootstrapping sampling is done from minor samples to obtain same number of samples as major samples, alleviating the data imbalance. Classification and regression models are used off-target prediction. Classification was done to classify the off-targets (labelled as '1') from other sequences (labelled as '0') and regression to

detect targeting efficacy measured with indel frequency detected by different assays.

3.1.4 Results

The results of DeepCRISPR are compared with the results of MIT [11], CCTop [43], CROP-IT [44] and CFD [25] on this dataset. DeepCRISPR outperformed all the other models with improved performance to reduce false positives in predicting off-targets. It is also concluded that using classification model for off-target prediction is more preferred as regression model which is used for predicting binding affinities require more data for training.

3.1.5 Review

DeepCRISPR extracts epigenetic features of DNA to predict off-target propensity and target cleavage site of sgRNA. Epigenetic features are highly volatile and have hypothetical dependency on cell state and cell type, which limits its application to selective cell types and cross-species prediction [17]. It is unclear if the epigenetic features will have any specific impact on the model prediction results. DeepCRISPR uses the largest dataset available to train the model but the article of DeepCRISPR did not provide a detailed information about test data and test results [21]. The number of negative samples is much larger than the number of positive samples in the off-target dataset. The authors of DeepCRISPR performed multiple experiments but they did not remove common data between training and testing datasets in their first experiment. For other experiments, some of the labelled and unlabelled data were observed to be similar during pre-training of unlabelled data [84]. On comparing the training and test loss curves, over-fitting and under-fitting issues were observed which lead to poor performance of the model [66].

3.2 CNN_Std

3.2.1 Network Architecture

Deep standard CNN (CNN_Std) [19], uses deep CNN and deep feedforward neural network (FNN) to predict off-target mutations by constructing a two-dimensional matrix using sequence-based features. The architecture of CNN_Std consists of:

- Convolutional layer to extract matching information of sgRNA-DNA sequence pairs,

- Batch normalization (BN) layer with ReLu as activation function to reduce internal covariate shift and allow higher learning rates,
- Global max-pooling layer to verify the mismatches modelled by BN layer,
- Two fully connected dense layers with a dropout layer used on the last dense layer to randomly mask portions of the output to avoid over-fitting, and,
- A final output layer consisting two neurons connected to previous layers.

FNN model contains the architecture of input layer, several hidden layers and output layer with softmax as activation function to convert each neuron output into probability. For both FNN and CNN models, best performance under 5-fold stratified cross-validation, adam algorithms (to optimize cross-entropy loss function) and mini-batch gradient descent (to reduce gradient variance) are adapted as shown in figure 3.2.

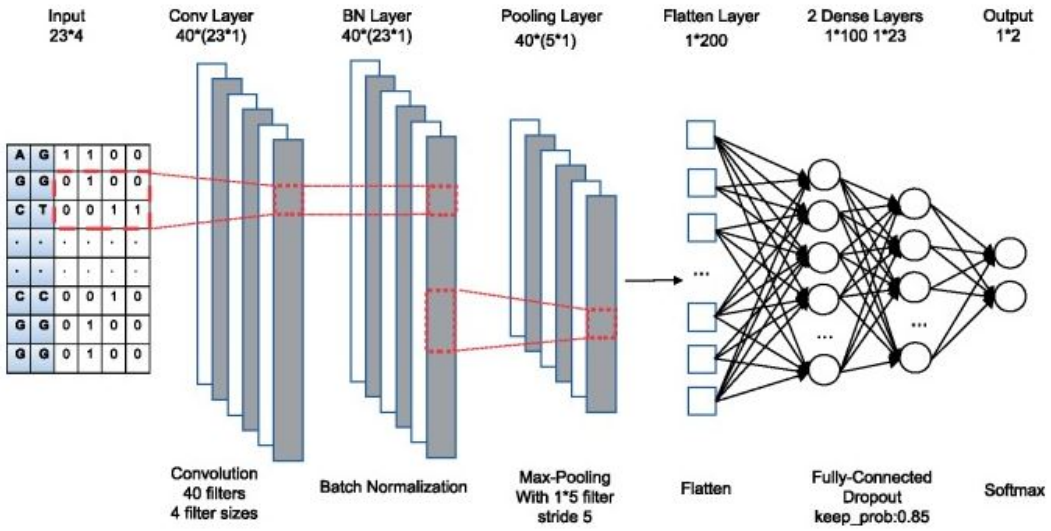


Figure 3.2: Network architecture of CNN_Std model [19]

3.2.2 Feature Extraction

CNN from computer vision is adapted by processing the sgRNA-DNA sequence pair with the length of 23-bp (3-bp PAM adjacent to the 20 bases) into a 4 x 23 matrix, using "XOR" coding design, instead of a 2-dimensional image with colour channels. Each base, (Adenine, Cytosine, Guanine and Thymine), in sgRNA and target DNA sequences are encoded as one of the four one-hot vectors [1,0,0,0], [0,1,0,0], [0,0,1,0] and [0,0,0,1]. The mutated information in the sgRNA-DNA sequence pair is encoded by deriving a 4-length vector. This vector is dervied by

encoding mismatched bases with OR operator. This encoded code matrix of the sgRNA-DNA sequence pair is used for the CNN-based models and the vectorized form of this matrix is used for the traditional ML models and deep FNN.

3.2.3 Dataset

CRISPOR [48] off-target dataset is used for training, testing and validation. This dataset contains 26034 presumed off-targets including 143 validated off-targets, having a mismatch count of upto four with one of the PAM like NAG/NGA/NGG. For additional evaluation, GUIDE-seq [36] off-target dataset containing 28 off-targets among 403 potential off-target sites is used, which is excluded from CRISPOR dataset during training.

3.2.4 Results

On the CRISPOR [48] dataset, FNN_3layer and CNN_Std achieved the best performance under stratified 5-fold cross-validation and demonstrated progress over traditional ML models like GBR, random forest (RF) and logistic regression (LR). Comparing the performance of these models with other off-target predicting tools like, MIT [11], CCTop [43], CROP-IT [44] and CFD [25], both these models outperformed all the other tools. On GUIDE-seq [36] dataset, CNN_Std achieved the highest true positive rate demonstrating the best generalization performance among other prediction models.

3.2.5 Review

CNN_Std achieved high accuracy by constructing 2-dimensional input matrix of sequence-based features using "XOR" coding design. CNN_Std had a poor performance due to over-fitting and under-fitting issues. DeepCRISPRas9 [65] and CNN_Std have similar network architecture, of using only one multi-scale convolution layer, but the input size of DeepCas9 is different (30 nucleotides). CNN can abstract features by convolution. CNN_Std attempted to downsize the fully connected layer by utilizing a maximum pooling layer with window size 5x1 and stride 5. But it is not possible to perform down-sampling for 23x1 feature maps [66].

3.3 AttnToMismatch_CNN

3.3.1 Network Architecture

Attention-based transformer, a DL neural network architecture is used by AttnToMismatch_CNN [20] for off-target specificity prediction of CRISPR-Cas9 system. As shown in figure 3.3, the architecture of AttnToMismatch_CNN consists of:

- Embedding layers to encode each position of the sgRNA and DNA sequence pair into a vector representation and encode into a matrix,
- A transformer layer with encoder and decoder parts to produce output with dimension same as the input,
- CNN layer with two Conv2d and two Maxpooling layers interleaved, and,
- Two fully connected dense layers with a dropout layer used on the last dense layer to randomly mask portions of the output to avoid over-fitting, and,
- A fully connected layer with softmax function to predict probability of sgRNA as positive or negative samples.

5-fold cross validation and leave-3-sgRNAs-out scenario are done to evaluate the model.

3.3.2 Feature Extraction

Base-pairs from each position of the aligned sgRNA and DNA sequences are extracted forming 16 different types. Depending on the length of the input sequence from the dataset, 20 bp are extracted. Raw feature importance is the average loss score obtained by calculating eventual losses and mean square losses for regression by perturbing each input feature across all the samples. Raw feature importance is then normalized by summing all the feature importance values and provided as weights for the model.

3.3.3 Dataset

Off-target dataset was created by collecting 656 off-target sites used in DeepCRISPR [18] model as positive samples and around 165000 sgRNA-DNA mismatch pairs from Cas-OFFinder [45] as negative samples.

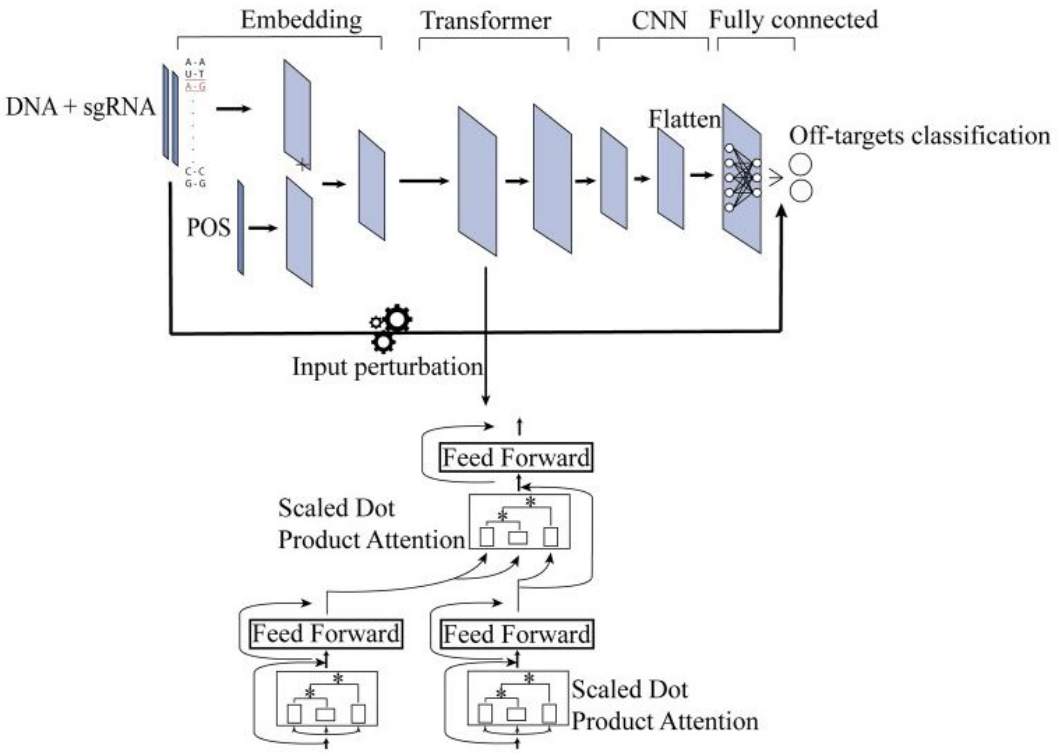


Figure 3.3: Network architecture of AttnToMismatch_CNN model [20]

3.3.4 Results

On comparing the performance of AttToMismatch_CNN model with other models like RF and GBT, AttToMismatch_CNN outperformed other models by a margin of 10% and around 20% margin in the 5-fold cross validation and leave3-sgRNAs-out scenarios. This model improves true positive rate and reduces the false positive rate with the application of embedding and transformer layer in encoding extracted sequence features into vectors.

3.3.5 Review

The process of encoding the sequence features into vectors is inspired by the word embedding technique in the natural language processing (NLP). The off-target dataset used in this study is highly imbalanced and the authors have mentioned that they have over-sampled the positive samples in every mini-batch making it equal to negative samples. But the authors of AttToMismatch_CNN did not give detailed information of how they over-sampled the positive samples. The negative samples of the off-target dataset constructed using the Cas-OFFinder [45] is very similar to the positive samples used from the DeepCRISPR [18] model. Input perturbation component used for identifying the feature importance did not show any difference for the features

other than the first and second positions of 5' end of the sgRNA. For extracting the features from the sgRNA-DNA sequence pair, the authors used 20 base-pairs from the sequence-pairs leaving the PAM region, which is very crucial for predicting off-targets in CRISPR-Cas9 system.

3.4 CnnCrispr

3.4.1 Network Architecture

CnnCrispr [21] constructed Glove vector (GloVe) embedding model by extracting sequence information of sgRNA and corresponding DNA sequence in the form of a co-occurrence matrix and predicted the off-target propensity of sgRNA using CNN and biLSTM. As shown in figure 3.4, the architecture of this model consists of:

- An embedding layer accepts 2-dimensional vector matrix of GloVe model created from a co-occurrence matrix,
- biLSTM network with 5 convolution layers and 2 full connection layers to extract context features from input,
- Batch Normalization and Dropout layers to prevent model over-fitting, and,
- Output layer with softmax and sigmoid functions as activation functions to obtain results of classification and regression model.

Adam algorithm is used to optimize loss function and initial learning rate is set to 0.01 for training the model.

3.4.2 Feature Extraction

Similar to the AttToMismatch_CNN [20] model, the features are extracted by aligning the sgRNA-DNA sequences forming 16 different types of base-pairs set with a unique index value. The sgRNA-DNA sequence-pair are encoded for GloVe embedding. A pre-processed co-occurrence matrix is created from the sequence-pairs and trained using the GloVe model to learn word vectors and produce embedded word vector representation of the base-pairs.

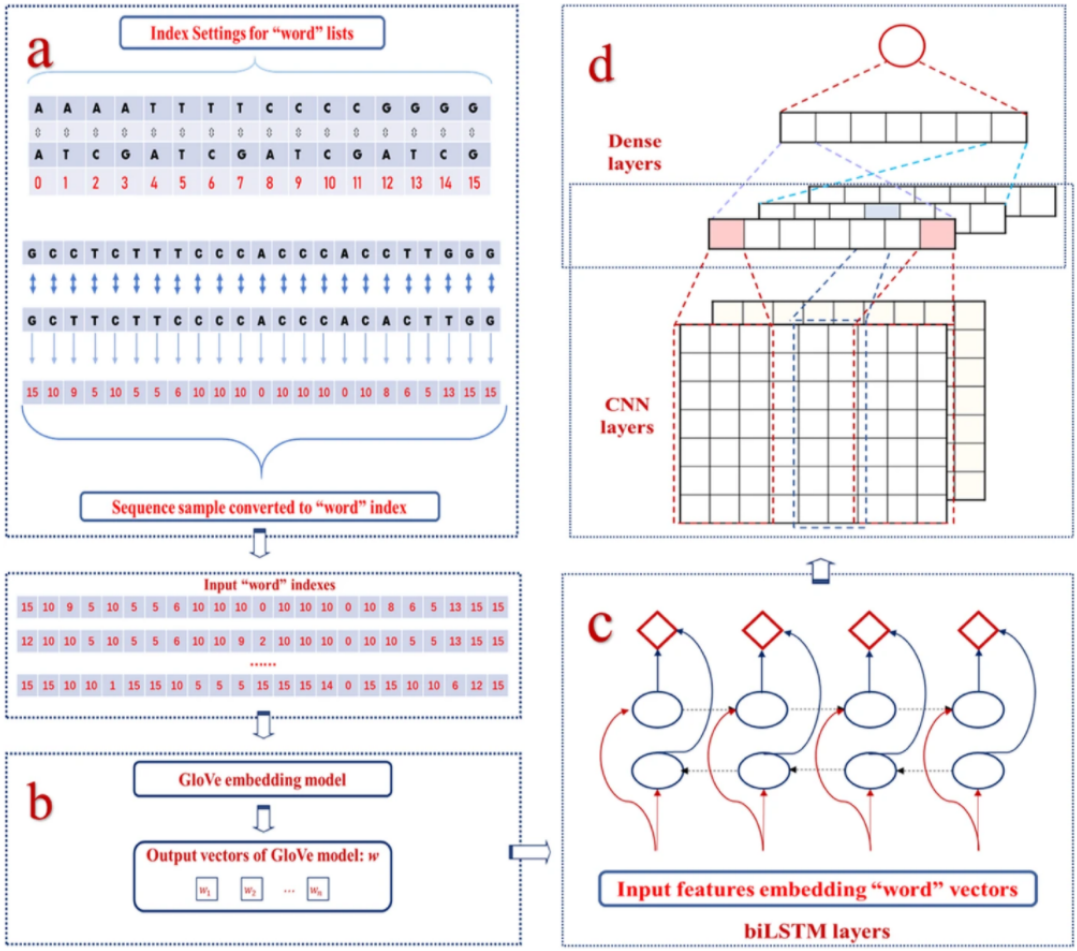


Figure 3.4: Network architecture of CnnCrispr model [21]

3.4.3 Dataset

CnnCrispr was trained on the off-target dataset of DeepCRISPR [18] model (HEK293 cell line with 18 sgRNA and K562T cell line with 12 sgRNA) with 80% of the samples for training and 20% for testing processes.

3.4.4 Results

Performance of the model decreased when the Dropout and BatchNormalization layers were removed, as these layers improved the performance and prevented over-fitting. A comparative study is done on the DeepCRISPR [18] off-target dataset by comparing the performance of CnnCrispr with other models like CFD [25], MIT [11] and CNN_Std [19]. CnnCrispr outperformed all the models by achieving good results. Similar to the AttToMismatch_CNN model [20], leave-1-sgRNA-out and 29-fold cross-validation scenarios were performed to validate the model's performance. CnnCrispr outperformed all the other models in both the scenarios.

3.4.5 Review

CnnCrispr implemented the concept of word embedding technique to encode the sequence features into the vector model as performed in the AttToMismatch_CNN [20] model. The authors have avoided the unknown influence of artificial feature construction on the prediction results by using the GloVe vector model, which created a co-occurrence matrix for the base-pairs. This extracted the sequence information of the sgRNA and corresponding DNA sequences, providing a detailed analysis of the position of nucleotides in the sgRNA-DNA sequence pairs. Use of the GloVe embedding model to extract the sequence information is a novel and innovative approach. However, the application of CnnCrispr model by researchers for off-target prediction is limited due to its complexity in the feature extraction process.

3.5 Summary

In this chapter, a detailed review of different DL models that were developed to predict the off-target efficacy of the sgRNA in CRISPR-Cas9 gene editing was provided. In the next chapter, the implementation of the graph-based approach to predict the off-target effects in CRISPR-Cas9 gene editing will be briefly discussed.

4 Graph Theory to Predict Off-targets

This chapter focuses on the implementation of a graph-based approach to predict the off-targets in CRISPR-Cas9 system. Section 4.1 provides the details of the off-target dataset used in this study. Section 4.2 explains about creating the network graphs from the off-target dataset. Two different case studies of feature extraction methods have been discussed in section 4.3. Section 4.4 explains about performing the cluster data sampling to handle the imbalance issue in the dataset. Section 4.5 explains about how to perform link prediction using StellarGraph to predict off-targets.

4.1 Off-target Dataset

Data used in this study is created by the authors of DeepCRISPR [18] model and has been used for off-target prediction by the AttnToMismatch_CNN [20] and CnnCrispr [21] models. Data is obtained by curating the human sgRNA whole-genome off-target profile data detected by Digenome-seq [27], IDLV [30], BLESS [35], GUIDE-seq [36] and LAM-HTGTS [38]. This dataset includes 29 unique sgRNAs by concatenating data from two different cell types: HEK293 cell line and its derivatives (18 sgRNAs) and K562T (12 sgRNAs), accounting for a maximum of six nucleotide mismatches. This dataset is included in the attachments provided in the CnnCrispr [21] article. Information about this dataset can be found in ”Supplementary Materials” section under ”Appendices”, with the file name as “off-target data”. This dataset contains the labels of off-target producing sites as ”1” and the labels of other sites as ”0” as shown in the figure 4.1.

The obtained dataset is validated against null values. The length of sgRNA and target sequences is validated to be of same length (23 nucleotides in each sequence). A case-sensitive validation is also performed to verify if the sgRNA and DNA sequences do not contain any char-

sg12	GCGCCACCGGTTGATGTGATGGG	GGGCCATGGGTTGATGTGATGAG	1
sg5	GAGTCCGAGCAGAAGAAGAAGGG	GAGAGCAAGCAGAAGAAGAAAAG	1
sg17	GACATCGATGTCCTCCCCATTGG	GACATCGATAGCCTCCCCACTGG	1
sg8	GGGTGGGGGGAGTTTGCTCCTGG	AAATGGGGGGAGTTTGCCCCCCG	0
sg8	GGGTGGGGGGAGTTTGCTCCTGG	AAGTAAGGGAAGTTTGCTCCTGG	0
sg8	GGGTGGGGGGAGTTTGCTCCTGG	GGGTGGGTGGAGTTTGCTACTGG	0
sg8	GGGTGGGGGGAGTTTGCTCCTGG	GAGTGGGTGGAGTTTGCTACAGG	0
sg8	GGGTGGGGGGAGTTTGCTCCTGG	GGTGGGGGTGGGTTTGCTCCTGG	0

Figure 4.1: Off-target dataset used in this study

acters other than upper-cased A, C, G and T, referring to the nucleotides, Adenine, Cytosine, Guanine and Thymine respectively.

4.2 Graph Creation

After validating the dataset, a network graph is created using StellarGraph API [23]. “Nodes” and “Edges” are required to generate network graph from the off-target dataset. All the unique sequences in the dataset including sgRNA and target DNA sequences are made as nodes in the graph. The relationship between the sgRNA and DNA sequences are made as edges for the graph. An edge will have a start node and a destination node or target node. All the sgRNA sequences will be set as start nodes and all the target DNA sequences that are identified as potential off-target sites will be set as target nodes for the edges in this graph. The graph contains 29 clusters based on the 29 unique sgRNA sequences forming links with their corresponding potential off-target sites.

OT and NOT are the target DNA sequences that are differentiated based on the labels in the dataset corresponding to the result of off-targets set by the authors of CnnCrispr. All the target sequences that are identified as potential off-target sites (with label as ’1’ from the dataset) are named as “OT” and target sequences that are not identified as off-targets (with label as ’0’) are named as “NOT” as shown in figure 4.2. The naming of the sequences is done:

1. To create the graph with only positive links (links between sgRNA and its corresponding potential off-target (OT) sequences).

2. To create balanced clusters containing equal number of OT and NOT sequences for every sgRNA cluster using cluster data sampling

The sequence names, OT and NOT, are discarded during link prediction and not saved as labels for nodes. This is done to make sure that the GCN model can accurately predict the presence and absence of links between the sequences by using only the features extracted from these sequences and not based on these sequence names. All the unique sequences in the graphs are numerically encoded forming unique sequence id for each of these sequences. The sequence id for these sequences are generated by alphabetically sorting all the sequences (including the sgRNA, OT and NOT sequences) in a pandas dataframe and then numerically encoding the sequences using label encoding.

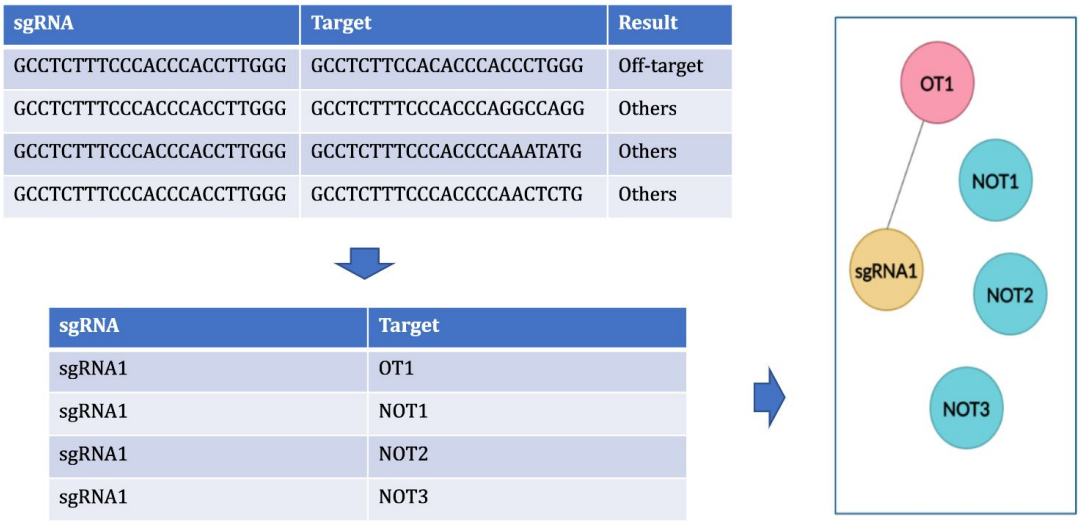


Figure 4.2: Creating network graph from off-target dataset

Network graph containing 29 clusters representing 29 sgRNA sequences with potential off-targets (OT) and a subgraph of 1 sgRNA cluster with its corresponding potential off-targets (OT) is created and drawn using NetworkX [85]. This graph and subgraph can be seen in figure 4.3. The labels of the nodes are the unique sequence ids generated for the sgRNA and target sequences. Node with the label "804" in the center is the sequence id for a sgRNA sequence and its neighbor nodes are its corresponding off-target sequences.

4.3 Feature Extraction

As all the sequences are numerically encoded, features for these sequences need to be provided, which will enable the GCN model to embed these sequences in a graph. Performance of the model is tested by giving two different types of features extracted from within the sequences -

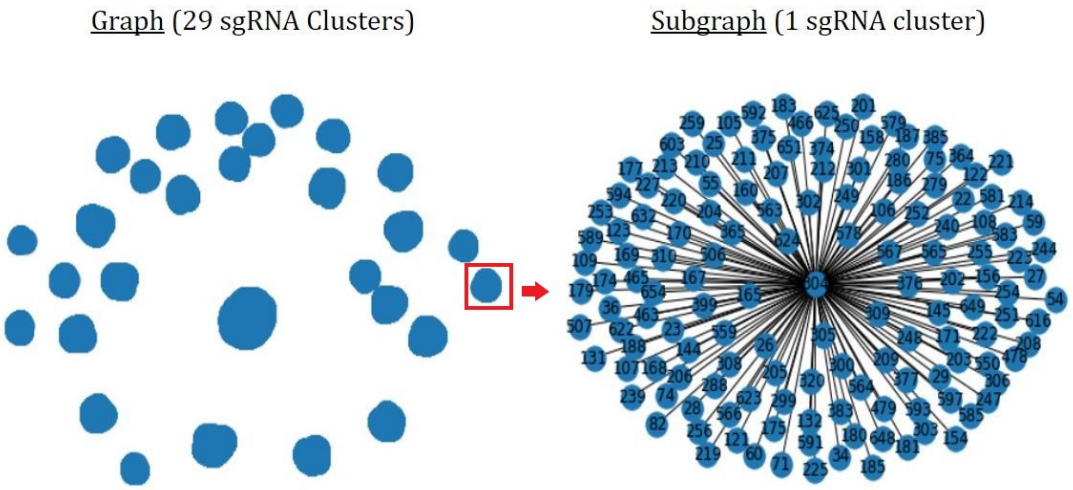


Figure 4.3: Network graph and sub-graph of off-target dataset

position and occurrences of nucleotides in the sequence to identify the sequences in the graph.

4.3.1 Case study 1: Nucleotide Occurrence

The occurrences of nucleotides in a sequence is extracted. Occurrences of nucleotides can be determined by providing different size of *k*-mers. The *k*-mers are the substrings of length *k* contained within a biological sequence. The choice of *k*-mers have different effects on sequence assembly. Features are extracted from within the sequences by providing different *k*-mer sizes. *k* values of 1, 2 and 3 are provided to get the occurrences. For example, as shown in figure 4.4, the features will be A, C, G and T for *k*-mer size of 1. Number of features for the sequences in this case depend on the *k* values of 1, 2 and 3 as 4, 16 and 64 respectively.

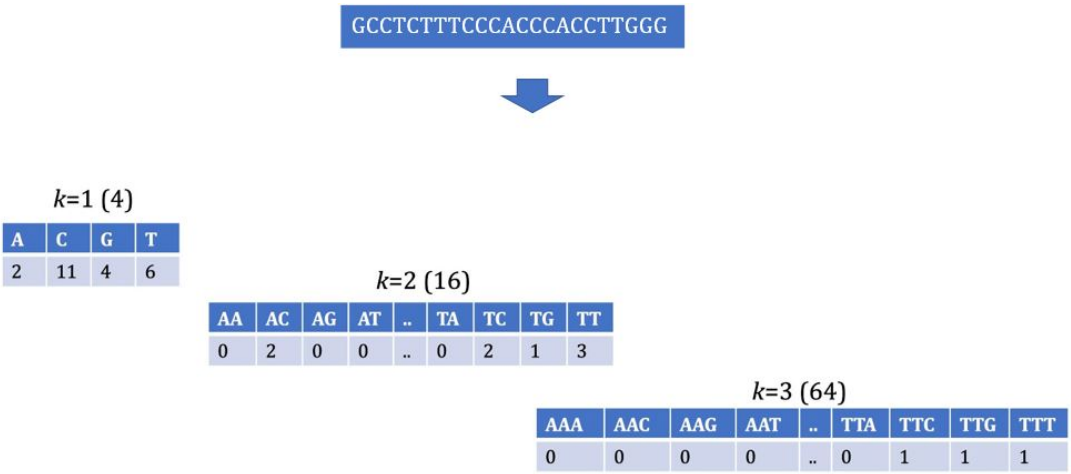


Figure 4.4: Extracting features for nodes using occurrences of nucleotides in sequences with different *k*-mer sizes of 1, 2 and 3.)

4.3.2 Case study 2: Nucleotide Position

Features for the sequences are generated by considering the position of the nucleotides in the sequences. For every sequence, 92 different features are extracted depending on the possibility of 4 nucleotides occurring at 23 positions in the sequence. Based on the presence and absence of nucleotides in the position, the values are entered as 1 and 0, respectively as shown in Figure 4.5.

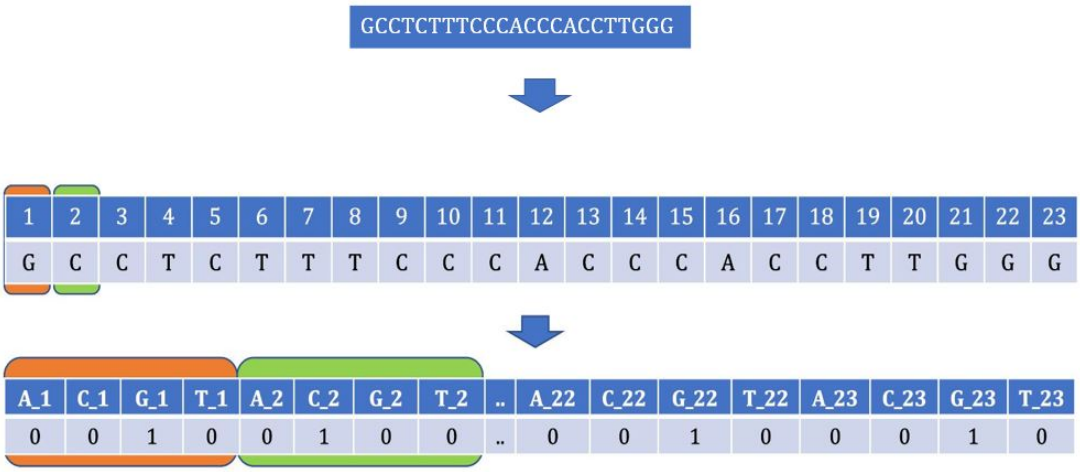
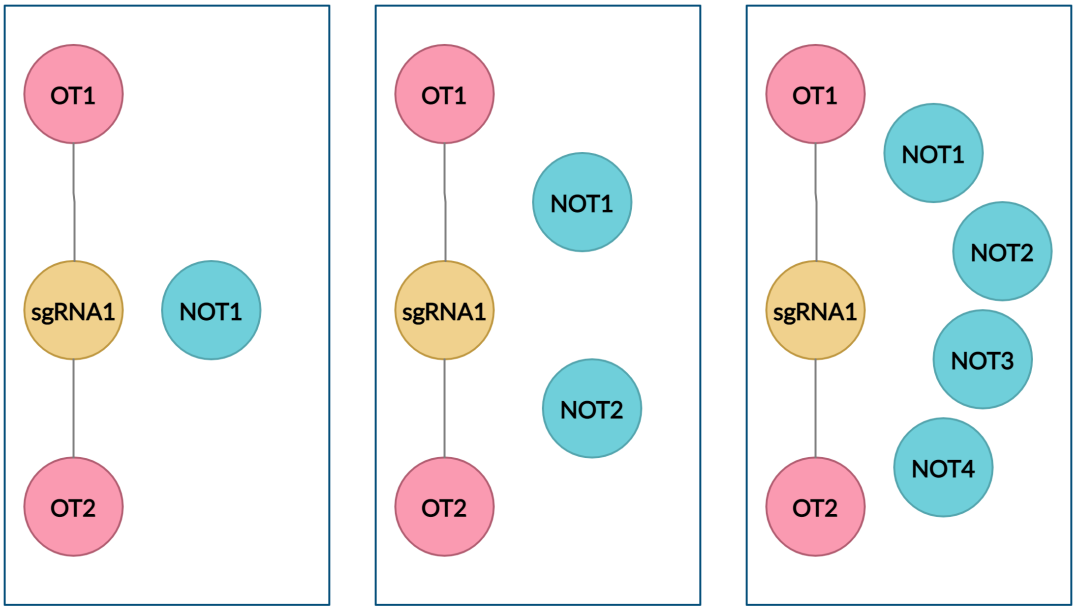


Figure 4.5: Extracting features for nodes using position of nucleotides in the sequences

4.4 Cluster Data Sampling

As the dataset is highly imbalanced, the GCN model needs to be trained and tested on a balanced graph, where the OT and NOT samples are balanced in every sgRNA cluster in the graph as shown in the figure 4.6 (b). Unlike, the leave-sgRNA-out scenario, that was tried in previous models to remove the imbalance between samples, cluster data sampling is done to balance the OT and NOT sequences in each sgRNA cluster. This is achieved by randomly sampling NOT sequences with equal amount of OT sequences present in a sgRNA cluster. By this, it can be verified that all the sgRNA clusters will have equal amount of OT and NOT sequences.

The cluster is also sampled in imbalanced scenarios as shown in Figure 4.6 (a) and (c), where the NOT sequences are randomly sampled depending on the amount of OT sequences present in a sgRNA cluster. For imbalanced towards NOT scenario (imbalanced_NOT clusters), as shown in figure 4.6 (c), NOT sequences are randomly sampled with twice the amount of OT sequences present in every sgRNA cluster. Similarly, for imbalanced towards OT scenario (imbalanced_OT



(a) (b) (c)

Figure 4.6: Cluster data sampling of OT and NOT sequences in Imbalanced_OT (a), balanced (b) and imbalanced_NOT (c) clusters

clusters) as shown in figure 4.6 (a), NOT sequences are randomly sampled with almost half the amount of OT sequences present in every sgRNA cluster. In all the three scenarios, the count of sgRNA and OT sequences remain unchanged and only the NOT sequences are randomly sampled depending on the amount of OT sequences present in every sgRNA cluster. The total amount of sgRNA, OT and NOT sequences used for this research in all three scenarios can be found in ”Table I” added under ”Tables” section in Appendices.

4.5 Off-target Prediction

4.5.1 Off-target Graph

A network graph from the off-target dataset is created as mentioned in section 4.2, by forming the nodes and edges. For nodes, a pandas dataframe is created. All the sequences (sgRNA, OT and NOT) are taken as nodes and encoded with sequence ids. Sequence-based features are extracted from within the sequences as mentioned in section 4.3 and set as column names and column values for the nodes dataframe. For edges, the sequence id of the sgRNA and its corresponding potential off-target sequences are taken in another pandas dataframe. Using StellarGraph API [23], a graph is created by giving these two dataframes as nodes and edges.

4.5.2 ”Edgesplitter” Function

Once the graph is created, link prediction is performed to predict whether a link or edge in a graph should exist. This is done by performing binary classification using the in-built functions provided by the StellarGraph [23]. “EdgeSplitter” function is used to apply a binary operator to classify the relationship between sgRNA and target sequences as positive and negative links, ”1” and ”0”, respectively. ”1” denotes that the link exists between the sgRNA and target sequences. ”0” denotes the absence of link between the sgRNA and target sequences.

“EdgeSplitter” function will carefully split the input graph (with 626 positive edges) into train graph (with 501 positive edges) and a test graph (with 125 positive edges) with a 80:20 train-test split. The train graph (with 501 positive edges) is further split into node embedding set(with 401 positive edges) and temporary test set (with 100 positive edges) using a second 80:20 train-test split. This temporary test set (with 100 positive edges) is again split into link embedding set (with 80 positive edges) and an independent test (with 20 positive edges) using a third 80:20 train-test split. Figure 4.7 shows an overview of how the “EdgeSplitter” function splits the input graph into multiple units.

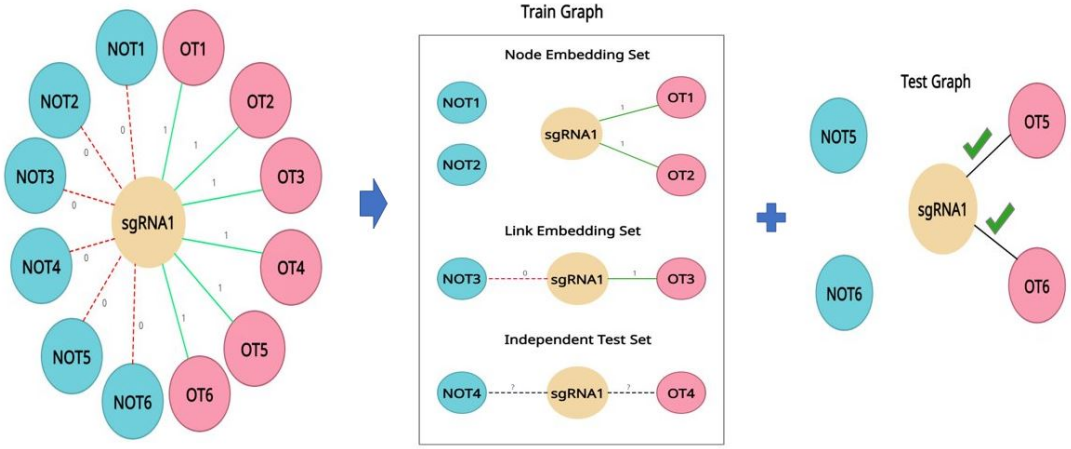


Figure 4.7: Link Prediction to predict off-target efficacy of sgRNA

Every individual set created from the input graph (with 626 positive edges) will have a unique functionality:

- Node embedding set (with 401 positive edges) is used to compute node embeddings. GCN model will learn about all the sequences (nodes) in the graph. This set will have only positive edges.

- Link Embedding set (with 80 positive edges and 80 negative edges) is used to compute link embeddings on positive and negative edges that were not used for computing sequence (node) embeddings. GCN model will use the information gained from node embedding and understand why a positive or negative link exist between the sequences (nodes).

- Independent test set (with 20 positive edges and 20 negative edges) is used to verify the performance of the GCN model in classifying the positive and negative links between the sequences (nodes) in the graph. This set contains positive and negative edges that are not used in the node embedding and link embedding processes.

- A final test graph (with 125 positive edges and 125 negative edges) is used to evaluate the performance of the GCN model. AUC values under ROC curves (auROC) and AUC values under PRC curves (auPRC) values are calculated to measure the performance of the model. This graph contains positive and negative edges that are not used in the node embedding, link embedding and in the independent test set.

The split ratio used in this study is the state-of-the-art values provided by the StellarGraph [23]. ”Table II” added under ”Tables” section in Appendices provides the split count of positive and negative edges for each set from the input graph used for this research.

4.5.3 GCN Model

GCN [24] model is created using StellarGraph [23]. GCN layers are stacked with the graph convolution and dropout layers. Two GCN layers with 16 units each are used with the rate of dropout for the input of each layer set to 30%. The output of each GCN layer is activated using ReLu activation. Adam algorithm is used to optimize the loss function with learning rate set to 0.01 for training the model. The output of the model will be binary classification of 1 and 0 (1 denoting the presence and 0 indicating the absence of links between sgRNA and target sequences). ”FullBatchLinkGenerator” function provided by StellarGraph library will provide the feature array and normalized adjacency matrix to the GCN model. The performance of the model is evaluated by learning node and link embeddings.

4.5.4 Node Embedding

For node embedding, StellarGraph provides an option to compute based on random walks based node embedding. A biased random walk is generated from the off-target graph with fixed random walk parameters of p (”1/p” probability of returning to source node) and q (”1/q” probability of moving to a node away from source node) set to 1. The model learns about the sequences (nodes) co-occurring in short random walks represented closely in the embedding space.

4.5.5 Link Embedding

Link embeddings are calculated by applying a binary operator on the sgRNA and OT sequences of each link. StellarGraph provides the option to evaluate the performance of the model using different binary operators - Hadamard, average, L1 and L2. Sequence representations are obtained from node embedding and a binary classifier is used to predict if a link should exist between any two sequences in a graph.

4.5.6 Performance Evaluation

With node and link embeddings, a logistic regression classifier with 10-fold cross validation (CV) is trained on the embeddings of positive and negative edges to predict a binary value indicating if a link between the sequences should exist or not. The model is trained end-to-end using binary cross entropy between link probabilities and true link labels for 10 epoch values and evaluated using the test set. Finally, the model is applied to the independent test graph to predict only the positive links in the graph. Area under the receiver operating characteristic (auROC) and area under the precision-recall curve (auPRC) metrics are calculated to measure the performance of the model.

4.6 Summary

This chapter provided a detailed information about creating network graphs from off-target dataset, extracting sequence-based features, sampling sequences on balanced and imbalanced clusters using cluster data sampling and implementing link prediction to predict links between sgRNA and off-target producing target sequences. Next chapter will discuss the performance of this model.

5 Results and Discussion

In this chapter, performance of the GCN model in predicting the off-target efficacy of sgRNA is validated. A brief discussion about the performance of the model is provided in the end of this chapter.

5.1 Computing auROC and auPRC values

StellarGraph is not able to perform link prediction on the entire dataset as it is highly imbalanced. Hence, the performance of GCN model is validated on balanced and imbalanced clusters created using cluster data sampling. The performance of the model is validated by calculating the area under the receiver operating characteristic (auROC) and area under the precision-recall curve (auPRC) metrics. As the NOT sequences are randomly sampled, the auROC and auPRC values tend to change for every run. Hence, the experiment is run multiple times and the mean of auROC and auPRC values are calculated.

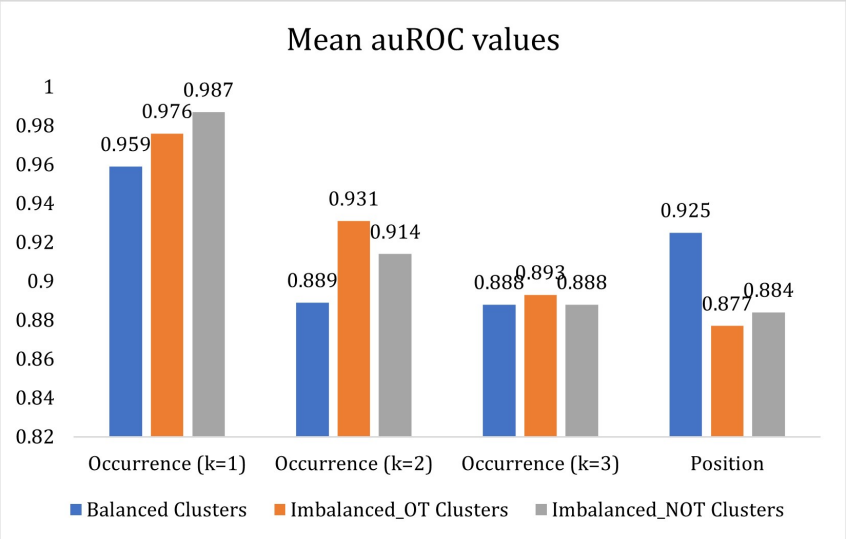


Figure 5.1: Mean auROC values computed for link prediction analysis

auROC and auPRC are performance metrics that can be used to evaluate the performance of the GCN model in predicting the positive edges. The higher the values are, the better the performance will be in predicting the positive links in the graph. An auROC value 0.5 and below corresponds to the worst performing model and a value of 1.0 corresponds to the best performing model.

As shown in figure 5.1, the model performs well when the nucleotide occurrences extracted from sequences are given as features with k value as 1. GCN is able to achieve auROC value of 0.959 when the dataset is balanced. Under the imbalanced datasets, the model has an auROC value of 0.976 and 0.987, when the dataset is imbalanced towards OT and NOT sequences, respectively. The auPRC values of balanced, imbalanced_OT and imbalanced_NOT clusters were found to be 0.961, 0.988 and 0.976, respectively, as shown in figure 5.2. GCN performs very well in imbalanced towards NOT scenario (imbalanced_NOT clusters), where the number of NOT sequences are more than OT sequences.

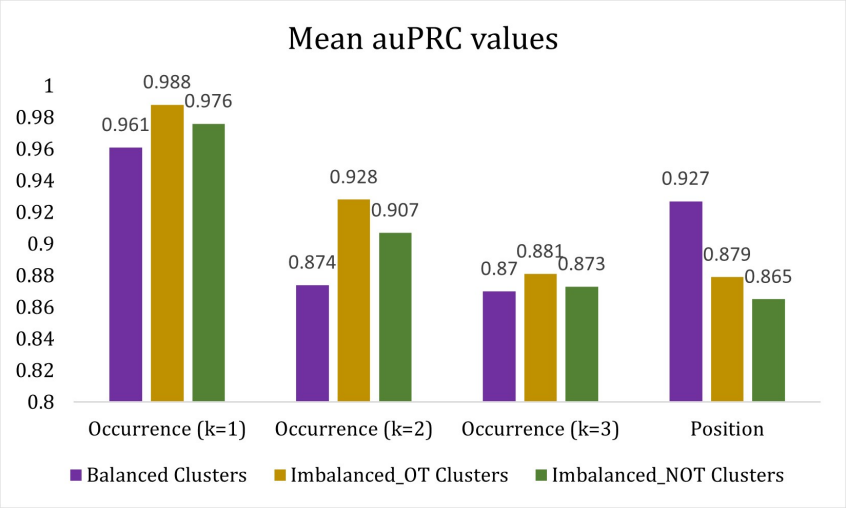


Figure 5.2: Mean auPRC values computed for link prediction analysis

It could be observed from both the figures 5.1 and 5.2, that the performance of the model decreases when the k-mer size is increased. For k-mer sizes of 2, the auROC values and auPRC values in balanced clusters were found to be 0.889 and 0.874, respectively. When the clusters are imbalanced towards OT sequences, the model scores 0.931 and 0.928 of auROC and auPRC values. When the clusters are imbalanced towards NOT sequences, the model scored auROC and auPRC values of 0.914 and 0.907, respectively. In this scenario, the model performed well when there are more OT sequences than NOT sequences in every sgRNA clusters.

Similarly, for *k*-mer size of 3, the performance of the model further decreased. Under balanced conditions, the auROC and auPRC values were found to be 0.888 and 0.870, respectively. When the clusters are imbalanced, the auROC values were found to be 0.893 and 0.888 and the auPRC values were found to be 0.881 and 0.873 for imbalanced towards OT and NOT sequences, respectively. No significant difference was observed with respect to the cluster data samples.

When providing the position of nucleotides in the sequences as features, the model performed very well. In balanced clusters, the model scored auROC and auPRC values of 0.925 and 0.927, respectively. When the cluster is imbalanced towards OT sequences, the model scored 0.877 and 0.879 of auROC and auPRC values respectively. When the model is imbalanced towards NOT sequences, the model scored 0.884 and 0.865 of auROC and auPRC values respectively.

"Table III" added under "Tables" section in Appendices contains the auROC and auPRC values calculated for all the feature types. "Figures I-IV" added under "Figures" section in Appendices show the binary accuracy and loss curves plotted for link prediction using occurrences (with *k*-mer sizes of 1, 2 and 3) and position of nucleotides in sequences as features for balanced, imbalanced_NOT and imbalanced_OT clusters.

5.2 Comparison of auROC values with other models

The auROC values obtained by GCN model for balanced and imbalanced clusters are compared with the results of other off-target predicting models performed by the authors of CnnCrispr [21]. These models include models like MIT [11], DeepCRISPR [18], CNN_Std [19], CnnCrispr [21] and CFD [25] as shown in figure 5.3.

This comparison is significant to understand how well GCN has performed in predicting off-target mutations. The authors of CnnCrispr [21] used 29-fold cross validation method of leave-1-sgRNA-out scenario to handle the data imbalance issue. All these models have been trained and tested on the same off-target dataset.

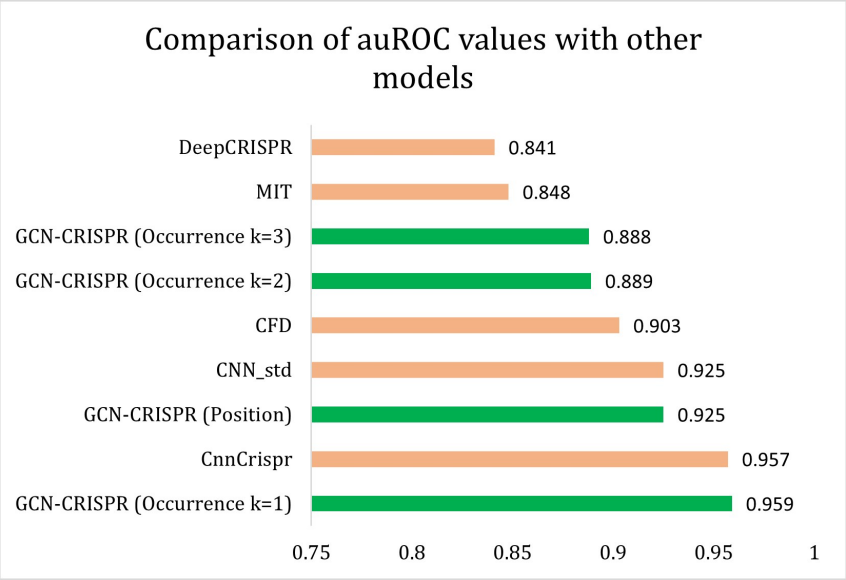


Figure 5.3: Comparison of auROC values with other predictive models

From the figure 5.3, it can be observed that the GCN model managed to perform well when compared to other model. With nucleotide position as feature, GCN was able to perform better than scoring-based models such as CFD and MIT and managed to perform better than DeepCRISPR. This is astounding as the feature extraction with this feature type is very straightforward and does not require manual effort. The performance of GCN is almost equal to the performance of CNN_Std model that was trained and tested with the leave-1-sgRNA-out scenario. With nucleotide occurrence of k-mer sizes of 1 as feature, GCN outperformed all the other models. Table IV added in Appendices contains the auROC values of all the models.

5.3 Discussion

To predict the efficacy of sgRNA, link prediction method is used, in which the existence of links between sgRNA and target DNA sequences is predicted. GCN model is used to validate the off-target efficacy of sgRNA in both feature types. Performance of GCN is extremely good, when nucleotide occurrences with k-mer size of 1 are given as features for the sequences in the network graph. The complexity of sequence-based feature extraction when the nucleotide position is given as feature is very less. it is observed that GCN performs well on train and test graphs. By increasing the k-mer sizes, it can be observed from auROC and auPRC metrics that the performance of the model decreases and when providing the position of nucleotides as feature, the model performs comparatively better.

StellarGraph API provided many in-built functions which was used to perform off-target prediction using link prediction method by automating the process of splitting the edges into different train and test graphs and to perform 10-fold cross validation using logistic regression to compute the auROC and auPRC values. However, StellarGraph did not allow usage of whole dataset, as the off-target dataset is highly imbalanced, negative samples (NOT sequences) were 115 times more than positive samples (OT sequences). Hence, cluster data sampling is done to create balanced and imbalanced clusters. In the imbalanced clusters, for imbalance_NOT clusters, the NOT sequences were sampled with twice the count of OT sequences and for imbalance_OT clusters, NOT sequences were sampled with half the count of OT sequences, thus creating controlled imbalanced clusters. NetworkX is used to create network graph and to visualize the clusters in the off-target graph.

The key takeaways from this research work are:

- A graph-based approach to predict off-target mutations in CRISPR-Cas9 gene editing using link prediction, which is easy to implement and replicate by researchers, is possible.
- Graph Convolutional Network (GCN) [24] can be used to predict off-target efficacy of sgRNA by predicting links between sgRNA and off-target inducing target DNA sequences as it has achieved an auROC value of above 0.85 proving that the performance is excellent.
- Sequence-based features, like position and occurrences of nucleotides in a sequence, can improve the performance of the GCN model in analysing the graphs with high accuracy.
- The imbalance in off-target dataset can be handled using cluster data sampling, where the sequences can be randomly sampled and balanced for every sgRNA cluster.
- StellarGraph [23] provides a user friendly API to create network graphs and perform graph validation using GCN model. This can be used by researchers to automate the creation and normalization of adjacency and feature matrices.

5.4 Summary

This chapter discussed the performance of GCN model in predicting off-target efficacy of sgRNA in CRISPR-Cas9 gene editing. A concluding statement will be provided in the next chapter.

6 Conclusion

In this approach, a novel graph-based approach to predict the off-target efficacy of sgRNA is introduced. Almost all the DL models that predicted off-target effects of sgRNA, have used convolutional neural network (CNN) in their architecture. This is the first time, a graph neural network has been implemented for off-target prediction. The existence of links between sgRNA and its potential off-target sequences can be predicted by using link prediction method. Link prediction is performed using StellarGraph which made the computation process much easier. Graph convolutional network (GCN) model is able to achieve high AUC values under ROC curves (auROC) and AUC values under PRC curves (auPRC) values when predicting off-targets. Unlike the previous deep learning models that were created to predict off-target effects, this approach is easy to understand and replicate for off-target prediction research.

References

[1] F. D. Urnov, E. J. Rebar, M. C. Holmes, H. S. Zhang, and P. D. Gregory, "Genome editing with engineered zinc finger nucleases", *Nature Reviews Genetics*, vol. 11, no. 9, pp. 636–646, 2010.

[2] Y.-G. Kim, J. Cha, and S. Chandrasegaran, "Hybrid restriction enzymes: Zinc finger fusions to fok i cleavage domain", *Proceedings of the National Academy of Sciences*, vol. 93, no. 3, pp. 1156–1160, 1996.

[3] J. Boch, H. Scholze, S. Schornack, A. Landgraf, S. Hahn, S. Kay, T. Lahaye, A. Nickstadt, and U. Bonas, "Breaking the code of dna binding specificity of tal-type iii effectors", *Science*, vol. 326, no. 5959, pp. 1509–1512, 2009.

[4] M. Christian, T. Cermak, E. L. Doyle, C. Schmidt, F. Zhang, A. Hummel, A. J. Bogdanove, and D. F. Voytas, "Targeting dna double-strand breaks with tal effector nucleases", *Genetics*, vol. 186, no. 2, pp. 757–761, 2010.

[5] R. Liu, M. C. Bassalo, R. I. Zeitoun, and R. T. Gill, "Genome scale engineering techniques for metabolic engineering", *Metabolic engineering*, vol. 32, pp. 143–154, 2015.

[6] X. Xu and L. S. Qi, "A crispr–dcas toolbox for genetic engineering and synthetic biology", *Journal of molecular biology*, vol. 431, no. 1, pp. 34–47, 2019.

[7] D. Bhaya, M. Davison, and R. Barrangou, "Crispr-cas systems in bacteria and archaea: Versatile small rnas for adaptive defense and regulation", *Annual review of genetics*, vol. 45, pp. 273–297, 2011.

[8] M. P. Terns and R. M. Terns, "Crispr-based adaptive immune systems", *Current opinion in microbiology*, vol. 14, no. 3, pp. 321–327, 2011.

[9] B. Wiedenheft, S. H. Sternberg, and J. A. Doudna, "Rna-guided genetic silencing systems in bacteria and archaea", *Nature*, vol. 482, no. 7385, pp. 331–338, 2012.

[10] Y. Zhang, X. Ge, F. Yang, L. Zhang, J. Zheng, X. Tan, Z.-B. Jin, J. Qu, and F. Gu, "Comparison of non-canonical pams for crispr/cas9-mediated dna cleavage in human cells", *Scientific reports*, vol. 4, no. 1, pp. 1–5, 2014.

[11] P. D. Hsu, D. A. Scott, J. A. Weinstein, F. A. Ran, S. Konermann, V. Agarwala, Y. Li, E. J. Fine, X. Wu, O. Shalem, *et al.*, "Dna targeting specificity of rna-guided cas9 nucleases", *Nature biotechnology*, vol. 31, no. 9, pp. 827–832, 2013.

[12] L. G. Lowder, D. Zhang, N. J. Baltes, J. W. Paul, X. Tang, X. Zheng, D. F. Voytas, T.-F. Hsieh, Y. Zhang, and Y. Qi, "A crispr/cas9 toolbox for multiplexed plant genome editing and transcriptional regulation", *Plant physiology*, vol. 169, no. 2, pp. 971–985, 2015.

[13] W. Jiang, D. Bikard, D. Cox, F. Zhang, and L. A. Marraffini, "Rna-guided editing of bacterial genomes using crispr-cas systems", *Nature biotechnology*, vol. 31, no. 3, pp. 233–239, 2013.

[14] S. W. Cho, S. Kim, J. M. Kim, and J.-S. Kim, "Targeted genome engineering in human cells with the cas9 rna-guided endonuclease", *Nature biotechnology*, vol. 31, no. 3, pp. 230–232, 2013.

[15] Y. Fu, J. A. Foden, C. Khayter, M. L. Maeder, D. Reyon, J. K. Joung, and J. D. Sander, "High-frequency off-target mutagenesis induced by crispr-cas nucleases in human cells", *Nature biotechnology*, vol. 31, no. 9, pp. 822–826, 2013.

[16] V. Pattanayak, S. Lin, J. P. Guilinger, E. Ma, J. A. Doudna, and D. R. Liu, "High-throughput profiling of off-target dna cleavage reveals rna-programmed cas9 nuclease specificity", *Nature biotechnology*, vol. 31, no. 9, pp. 839–843, 2013.

[17] J. Wang, X. Zhang, L. Cheng, and Y. Luo, "An overview and metanalysis of machine and deep learning-based crispr grna design tools", *RNA biology*, vol. 17, no. 1, pp. 13–22, 2020.

[18] G. Chuai, H. Ma, J. Yan, M. Chen, N. Hong, D. Xue, C. Zhou, C. Zhu, K. Chen, B. Duan, *et al.*, "Deepcrispr: Optimized crispr guide rna design by deep learning", *Genome biology*, vol. 19, no. 1, pp. 1–18, 2018.

[19] J. Lin and K.-C. Wong, "Off-target predictions in crispr-cas9 gene editing using deep learning", *Bioinformatics*, vol. 34, no. 17, pp. i656–i663, 2018.

[20] Q. Liu, D. He, and L. 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*, vol. 15, no. 10, e1007480, 2019.

[21] Q. Liu, X. Cheng, G. Liu, B. Li, and X. Liu, "Deep learning improves the ability of sgrna off-target propensity prediction", *BMC bioinformatics*, vol. 21, no. 1, pp. 1–15, 2020.

[22] M. Naeem, S. Majeed, M. Z. Hoque, and I. Ahmad, "Latest developed strategies to minimize the off-target effects in crispr-cas-mediated genome editing", *Cells*, vol. 9, no. 7, p. 1608, 2020.

[23] C. Data61, "Stellargraph machine learning library", *GitHub Repository*, 2018.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", *arXiv preprint arXiv:1609.02907*, 2016.

[25] J. G. Doench, N. Fusi, M. Sullender, M. Hegde, E. W. Vaimberg, K. F. Donovan, I. Smith, Z. Tothova, C. Wilen, R. Orchard, *et al.*, "Optimized sgrna design to maximize activity and minimize off-target effects of crispr-cas9", *Nature biotechnology*, vol. 34, no. 2, pp. 184–191, 2016.

[26] J. Listgarten, M. Weinstein, B. P. Kleinstiver, A. A. Sousa, J. K. Joung, J. Crawford, K. Gao, L. Hoang, M. Elibol, J. G. Doench, *et al.*, "Prediction of off-target activities for the end-to-end design of crispr guide rnas", *Nature biomedical engineering*, vol. 2, no. 1, pp. 38–47, 2018.

[27] D. Kim, S. Bae, J. Park, E. Kim, S. Kim, H. R. Yu, J. Hwang, J.-I. Kim, and J.-S. Kim, "Digenome-seq: Genome-wide profiling of crispr-cas9 off-target effects in human cells", *Nature methods*, vol. 12, no. 3, pp. 237–243, 2015.

[28] A. P. May, P. Cameron, A. H. Settle, C. K. Fuller, M. S. Thompson, A. M. Cigan, and J. K. Young, "Site-seq: A genome-wide method to measure cas9 cleavage", 2017.

[29] S. Q. Tsai, N. T. Nguyen, J. Malagon-Lopez, V. V. Topkar, M. J. Aryee, and J. K. Joung, "Circle-seq: A highly sensitive in vitro screen for genome-wide crispr–cas9 nuclease off-targets", *Nature methods*, vol. 14, no. 6, p. 607, 2017.

[30] R. Gabriel, A. Lombardo, A. Arens, J. C. Miller, P. Genovese, C. Kaeppel, A. Nowrouzi, C. C. Bartholomae, J. Wang, G. Friedman, *et al.*, "An unbiased genome-wide analysis of zinc-finger nuclease specificity", *Nature biotechnology*, vol. 29, no. 9, pp. 816–823, 2011.

[31] X. Wang, Y. Wang, X. Wu, J. Wang, Y. Wang, Z. Qiu, T. Chang, H. Huang, R.-J. Lin, and J.-K. Yee, "Unbiased detection of off-target cleavage by crispr-cas9 and talens using integrase-defective lentiviral vectors", *Nature biotechnology*, vol. 33, no. 2, pp. 175–178, 2015.

[32] P. J. Park, "Chip–seq: Advantages and challenges of a maturing technology", *Nature reviews genetics*, vol. 10, no. 10, pp. 669–680, 2009.

[33] L. Teytelman, D. M. Thurtle, J. Rine, and A. van Oudenaarden, "Highly expressed loci are vulnerable to misleading chip localization of multiple unrelated proteins", *Proceedings of the National Academy of Sciences*, vol. 110, no. 46, pp. 18 602–18 607, 2013.

[34] X. Wu, D. A. Scott, A. J. Kriz, A. C. Chiu, P. D. Hsu, D. B. Dadon, A. W. Cheng, A. E. Trevino, S. Konermann, S. Chen, *et al.*, "Genome-wide binding of the crispr endonuclease cas9 in mammalian cells", *Nature biotechnology*, vol. 32, no. 7, pp. 670–676, 2014.

[35] N. Crosetto, A. Mitra, M. J. Silva, M. Bienko, N. Dojer, Q. Wang, E. Karaca, R. Chiarle, M. Skrzypczak, K. Ginalski, *et al.*, "Nucleotide-resolution dna double-strand break mapping by next-generation sequencing", *Nature methods*, vol. 10, no. 4, pp. 361–365, 2013.

[36] S. Q. Tsai, Z. Zheng, N. T. Nguyen, M. Liebers, V. V. Topkar, V. Thapar, N. Wyvekens, C. Khayter, A. J. Iafrate, L. P. Le, *et al.*, "Guide-seq enables genome-wide profiling of off-target cleavage by crispr-cas nucleases", *Nature biotechnology*, vol. 33, no. 2, pp. 187–197, 2015.

[37] M. J. Osborn, B. R. Webber, F. Knipping, C.-l. Lonetree, N. Tennis, A. P. DeFeo, A. N. McElroy, C. G. Starker, C. Lee, S. Merkel, *et al.*, "Evaluation of tcr gene editing achieved by talens, crispr/cas9, and megatal nucleases", *Molecular Therapy*, vol. 24, no. 3, pp. 570–581, 2016.

[38] J. Hu, R. M. Meyers, J. Dong, R. A. Panchakshari, F. W. Alt, and R. L. Frock, "Detecting dna double-stranded breaks in mammalian genomes by linear amplification–mediated high-throughput genome-wide translocation sequencing", *Nature protocols*, vol. 11, no. 5, p. 853, 2016.

[39] L. O. Wilson, A. R. O'Brien, and D. C. Bauer, "The current state and future of crispr-cas9 grna design tools", *Frontiers in pharmacology*, vol. 9, p. 749, 2018.

[40] L. Cong, F. A. Ran, D. Cox, S. Lin, R. Barretto, N. Habib, P. D. Hsu, X. Wu, W. Jiang, L. A. Marraffini, *et al.*, "Multiplex genome engineering using crispr/cas systems", *Science*, vol. 339, no. 6121, pp. 819–823, 2013.

[41] S. W. Cho, S. Kim, Y. Kim, J. Kweon, H. S. Kim, S. Bae, and J.-S. Kim, "Analysis of off-target effects of crispr/cas-derived rna-guided endonucleases and nickases", *Genome research*, vol. 24, no. 1, pp. 132–141, 2014.

[42] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome", *Genome biology*, vol. 10, no. 3, pp. 1–10, 2009.

[43] M. Stemmer, T. Thumberger, M. del Sol Keyer, J. Wittbrodt, and J. L. Mateo, "Cctop: An intuitive, flexible and reliable crispr/cas9 target prediction tool", *PloS one*, vol. 10, no. 4, e0124633, 2015.

[44] R. Singh, C. Kuscu, A. Quinlan, Y. Qi, and M. Adli, "Cas9-chromatin binding information enables more accurate crispr off-target prediction", *Nucleic acids research*, vol. 43, no. 18, e118–e118, 2015.

[45] S. Bae, J. Park, and J.-S. Kim, "Cas-offinder: A fast and versatile algorithm that searches for potential off-target sites of cas9 rna-guided endonucleases", *Bioinformatics*, vol. 30, no. 10, pp. 1473–1475, 2014.

[46] A. Xiao, Z. Cheng, L. Kong, Z. Zhu, S. Lin, G. Gao, and B. Zhang, "Casot: A genome-wide cas9/grna off-target searching tool", *Bioinformatics*, vol. 30, no. 8, pp. 1180–1182, 2014.

[47] A. R. Perez, Y. Pritykin, J. A. Vidigal, S. Chhangawala, L. Zamparo, C. S. Leslie, and A. Ventura, "Guidescan software for improved single and paired crispr guide rna design", *Nature biotechnology*, vol. 35, no. 4, pp. 347–349, 2017.

[48] M. Haeussler, K. Schönig, H. Eckert, A. Eschstruth, J. Mianné, J.-B. Renaud, S. Schneider-Maunoury, A. Shkumatava, L. Teboul, J. Kent, *et al.*, "Evaluation of off-target and on-target scoring algorithms and integration into the guide rna selection tool crispor", *Genome biology*, vol. 17, no. 1, pp. 1–12, 2016.

[49] S. Abadi, W. X. Yan, D. Amar, and I. Mayrose, "A machine learning approach for predicting crispr-cas9 cleavage efficiencies and patterns underlying its mechanism of action", *PLoS computational biology*, vol. 13, no. 10, e1005807, 2017.

[50] C. Kuscu, S. Arslan, R. Singh, J. Thorpe, and M. Adli, "Genome-wide analysis reveals characteristics of off-target sites bound by the cas9 endonuclease", *Nature biotechnology*, vol. 32, no. 7, pp. 677–683, 2014.

[51] Y. Lin, T. J. Cradick, M. T. Brown, H. Deshmukh, P. Ranjan, N. Sarode, B. M. Wile, P. M. Vertino, F. J. Stewart, and G. Bao, "Crispr/cas9 systems have off-target activity with insertions or deletions between target dna and guide rna sequences", *Nucleic acids research*, vol. 42, no. 11, pp. 7473–7485, 2014.

[52] M. K. Rahman and M. S. Rahman, "Crisprpred: A flexible and efficient tool for sgrnas on-target activity prediction in crispr/cas9 systems", *PloS one*, vol. 12, no. 8, e0181943, 2017.

[53] N. Wong, W. Liu, and X. Wang, "Wu-crispr: Characteristics of functional guide rnas for the crispr/cas9 system", *Genome biology*, vol. 16, no. 1, pp. 1–8, 2015.

[54] J. Guo, T. Wang, C. Guan, B. Liu, C. Luo, Z. Xie, C. Zhang, and X.-H. Xing, "Improved sgrna design in bacteria via genome-wide activity profiling", *Nucleic acids research*, vol. 46, no. 14, pp. 7052–7069, 2018.

[55] M. Labuhn, F. F. Adams, M. Ng, S. Knoess, A. Schambach, E. M. Charpentier, A. Schwarzer, J. L. Mateo, J.-H. Klusmann, and D. Heckl, "Refined sgrna efficacy prediction improves large-and small-scale crispr–cas9 applications", *Nucleic acids research*, vol. 46, no. 3, pp. 1375–1385, 2018.

[56] M. A. Moreno-Mateos, C. E. Vejnar, J.-D. Beaudoin, J. P. Fernandez, E. K. Mis, M. K. Khokha, and A. J. Giraldez, "Crisprscan: Designing highly efficient sgrnas for crispr-cas9 targeting in vivo", *Nature methods*, vol. 12, no. 10, pp. 982–988, 2015.

[57] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani, and A. Telenti, "A primer on deep learning in genomics", *Nature genetics*, vol. 51, no. 1, pp. 12–18, 2019.

[58] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems", *arXiv preprint arXiv:1603.04467*, 2016.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library", *arXiv preprint arXiv:1912.01703*, 2019.

[60] F. Chollet *et al.*, "Keras: Deep learning library for theano and tensorflow", *URL: https://keras.io/k*, vol. 7, no. 8, T1, 2015.

[61] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, "Deep learning: New computational modelling techniques for genomics", *Nature Reviews Genetics*, vol. 20, no. 7, pp. 389–403, 2019.

[62] R. Jing, Y. Li, L. Xue, F. Liu, M. Li, and J. Luo, "Autobioseqpy: A deep learning tool for the classification of biological sequences", *Journal of Chemical Information and Modeling*, vol. 60, no. 8, pp. 3755–3764, 2020.

[63] H. K. Kim, S. Min, M. Song, S. Jung, J. W. Choi, Y. Kim, S. Lee, S. Yoon, and H. H. Kim, "Deep learning improves prediction of crispr–cpf1 guide rna activity", *Nature biotechnology*, vol. 36, no. 3, p. 239, 2018.

[64] G. Dimauro, P. Colagrande, R. Carlucci, M. Ventura, V. Bevilacqua, and D. Caivano, "Crisprlearner: A deep learning-based system to predict crispr/cas9 sgrna on-target cleavage efficiency", *Electronics*, vol. 8, no. 12, p. 1478, 2019.

[65] L. Xue, B. Tang, W. Chen, and J. Luo, "Prediction of crispr sgrna activity using a deep convolutional neural network", *Journal of chemical information and modeling*, vol. 59, no. 1, pp. 615–624, 2018.

[66] L. Wang and J. Zhang, "Prediction of sgrna on-target activity in bacteria by deep learning", *BMC bioinformatics*, vol. 20, no. 1, pp. 1–14, 2019.

[67] D. Wang, C. Zhang, B. Wang, B. Li, Q. Wang, D. Liu, H. Wang, Y. Zhou, L. Shi, F. Lan, *et al.*, "Optimized crispr guide rna design for two high-fidelity cas9 variants by deep learning", *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.

[68] P. Baldi and G. Pollastri, "The principled design of large-scale recursive neural network architectures–dag-rnns and the protein structure prediction problem", *The Journal of Machine Learning Research*, vol. 4, pp. 575–602, 2003.

[69] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Sheng, G. Soda, and A. Sperduti, "Logo recognition by recursive neural networks", in *International Workshop on Graphics Recognition*, Springer, 1997, pp. 104–117.

[70] E. Krahmer, S. v. Erk, and A. Verleg, "Graph-based generation of referring expressions", *Computational Linguistics*, vol. 29, no. 1, pp. 53–72, 2003.

[71] A. E. W. Mason and E. H. Blake, "A graphical representation of the state spaces of hierarchical level-of-detail scene descriptions", *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 1, pp. 70–75, 2001.

[72] L. Baresi and R. Heckel, "Tutorial introduction to graph transformation: A software engineering perspective", in *International Conference on Graph Transformation*, Springer, 2002, pp. 402–429.

[73] C. Collberg, S. Kobourov, J. Nagra, J. Pitts, and K. Wampler, "A system for graph-based visualization of the evolution of software", in *Proceedings of the 2003 ACM symposium on Software visualization*, 2003, 77–ff.

[74] A. Bua, M. Gori, and F. Santini, "Recursive neural networks applied to discourse representation theory", in *International Conference on Artificial Neural Networks*, Springer, 2002, pp. 290–295.

[75] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model", *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[76] M. Koutrouli, E. Karatzas, D. Paez-Espino, and G. A. Pavlopoulos, "A guide to conquer the biological network era using graph theory", *Frontiers in bioengineering and biotechnology*, vol. 8, p. 34, 2020.

[77] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains", in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, IEEE, vol. 2, 2005, pp. 729–734.

[78] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications", *arXiv preprint arXiv:1812.08434*, 2018.

[79] T. Kawamoto, M. Tsubaki, and T. Obuchi, "Mean-field theory of graph neural networks in graph partitioning", *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2019, no. 12, p. 124 007, 2019.

[80] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints", *arXiv preprint arXiv:1509.09292*, 2015.

[81] A. M. Fout, “Protein interface prediction using graph convolutional networks”, Ph.D. dissertation, Colorado State University, 2017.

[82] S. Rhee, S. Seo, and S. Kim, “Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification”, *arXiv preprint arXiv:1711.05859*, 2017.

[83] M. Zitnik, M. Agrawal, and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks”, *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[84] A. H. Muhammad Rafid, M. Toufikuzzaman, M. S. Rahman, and M. S. Rahman, “Crisprpred (seq): A sequence-based method for sgrna on target activity prediction using traditional machine learning”, *BMC bioinformatics*, vol. 21, pp. 1–13, 2020.

[85] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx”, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

Appendices

Appendix 1: Tables

Table I: Sequences in balanced and imbalanced clusters created using cluster data sampling

Clusters	sgRNA	OT	NOT
Balanced Clusters (OT = NOT)	29	626	626
Imbalanced_OT Clusters (OT > NOT)	29	626	304
Imbalanced_NOT Clusters (OT < NOT)	29	626	1252

Table II: Positive and negative edges created using ”Edgesplitter” function provided by the StellarGraph

Data	Positive Edges	Negative Edges	Total Edges
Input Graph	626	0	626
Node Embedding Set	401	0	401
Link Embedding Set	80	80	160
Independent Test Set	20	20	40
Test Graph	125	125	250

Table III: Mean auROC and auPRC values computed for different feature types for link prediction

Feature Types	Balanced Clusters		Imbalanced_OT Clusters		Imbalanced_NOT Clusters	
	Mean auROC	Mean auPRC	Mean auROC	Mean auPRC	Mean auROC	Mean auPRC
Nucleotide Occurrence (k=1)	0.959	0.961	0.976	0.988	0.987	0.976
Nucleotide Occurrence (k=2)	0.889	0.874	0.931	0.928	0.914	0.907
Nucleotide Occurrence (k=3)	0.888	0.870	0.893	0.881	0.888	0.873
Nucleotide Position	0.925	0.927	0.877	0.879	0.884	0.865

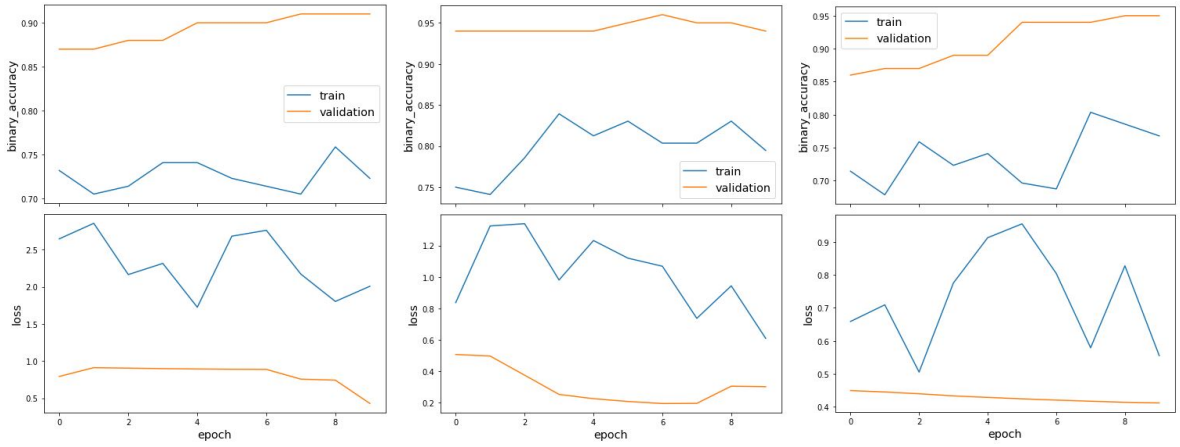
Table IV: Comparison of mean auROC values of GCN model with other off-target predicting models

Model	auROC
GCN-CRISPR (Occurrence k=1)	0.959
CnnCrispr	0.957
GCN-CRISPR (Position)	0.925
CNN_std	0.925
CFD	0.903
GCN-CRISPR (Occurrence k=2)	0,889
GCN-CRISPR (Occurrence k=3)	0,888
MIT	0.848
DeepCrispr	0.841

Appendix 2: Figures

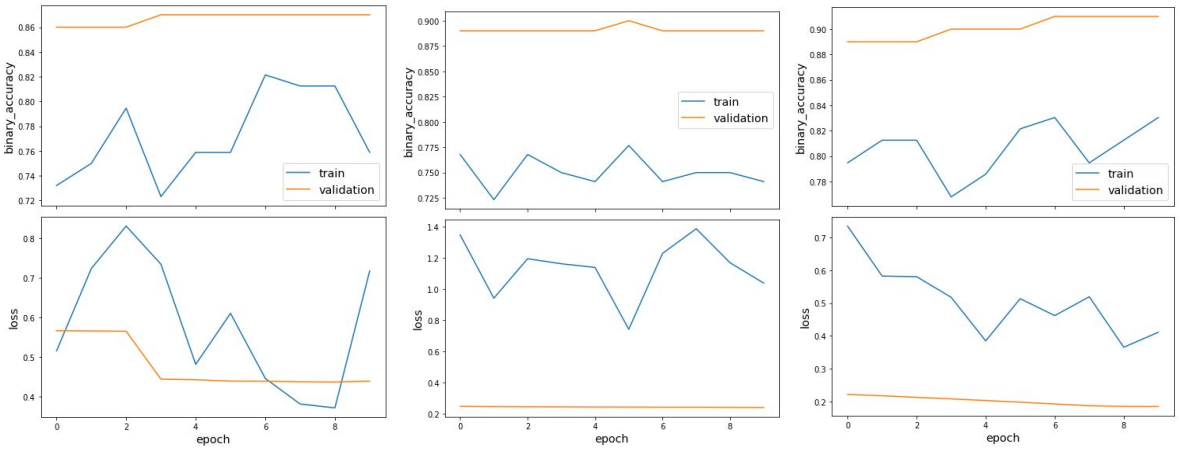
Binary accuracy and loss curves plotted for link prediction using occurrences (with *k*-mer sizes of 1, 2 and 3) and position of nucleotides in sequences as features

Figure I: Nucleotide occurrence with *k*-mer size of 1 for balanced (a), imbalanced_NOT (b) and imbalanced_OT (c) clusters



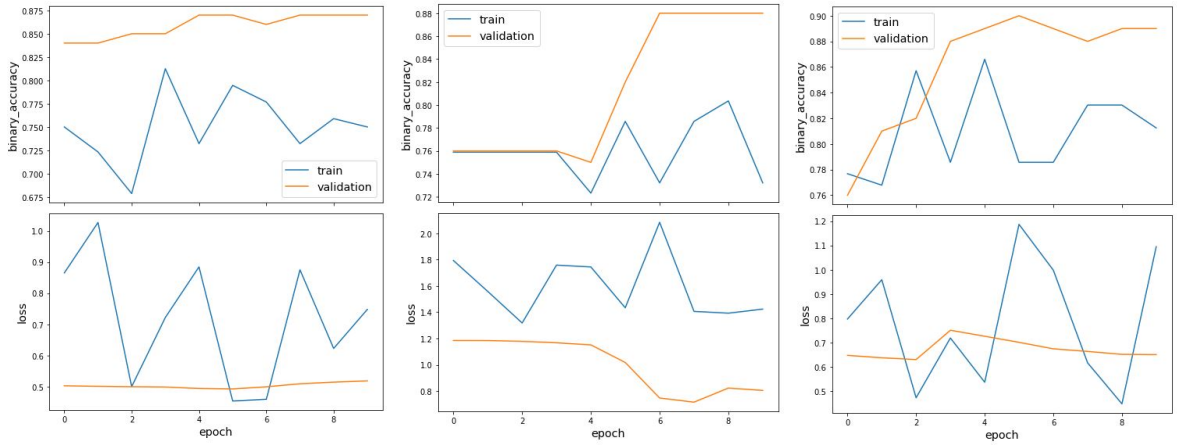
(a) (b) (c)

Figure II: Nucleotide occurrence with *k*-mer size of 2 for balanced (d), imbalanced_NOT (e) and imbalanced_OT (f) clusters



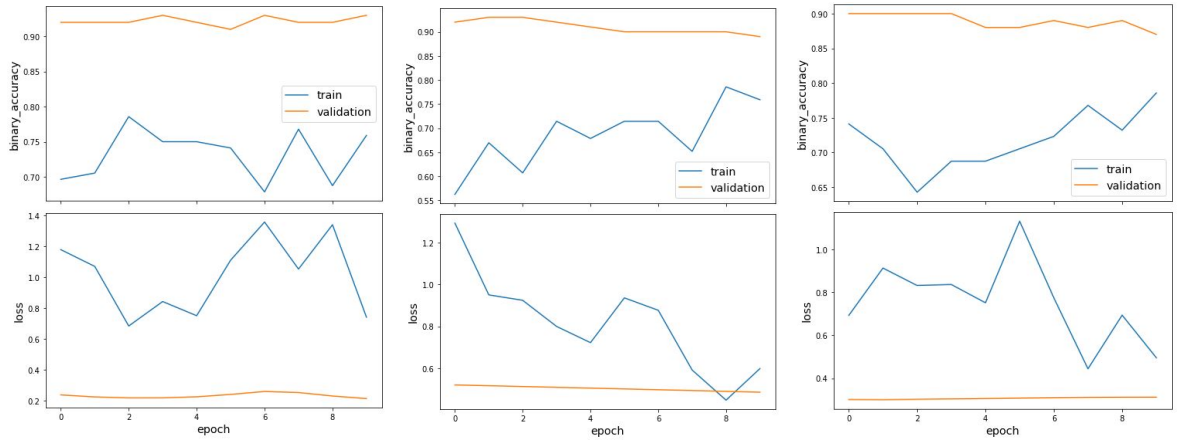
(d) (e) (f)

Figure III: Nucleotide occurrence with *k*-mer size of 3 for balanced (g), imbalanced_NOT (h) and imbalanced_OT (i) clusters



(g) (h) (i)

Figure IV: Nucleotide position for balanced (j), imbalanced_NOT (k) and imbalanced_OT (l) clusters



(j) (k) (l)

Appendix 3: Supplementary Materials

Material I: Source Code

The codes are now available in the following GitHub repository:

https://github.com/Prasoonk02/GCN-CRISPR

Please send a mail to prasoonk02@gmail.com for access.

Material II: Dataset

Data used in this study is included in the published articles "DeepCRISPR" and "CnnCrispr".

The corresponding supplementary information files can be found below:

DeepCRISPR - https://doi.org/10.1186/s13059-018-1459-4

CnnCrispr - https://doi.org/10.1186/s12859-020-3395-z

The data can be downloaded from CnnCrispr [21] and the file name is "*off-target data*".

Licence

Non-Exclusive licence to reproduce thesis and make thesis public

I, **Prasoon Kumar Vinodkumar**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of validity of the copyright, **Predicting Off-target Effects in CRISPR-Cas9 System using Graph Convolutional Network**, supervised by **Prof. Gholamreza Anbarjafari**.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Prasoon Kumar Vinodkumar
21/05/2021