View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by SJSU ScholarWorks

San Jose State University
SJSU ScholarWorks

Master's Projects | Master's Theses and Graduate Research

Spring 5-24-2019

Designing single guide RNAs for CRISPR/Cas9

Neha Atul Bhagwat
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the Artificial Intelligence and Robotics Commons, and the Other Computer Sciences Commons

Recommended Citation
Bhagwat, Neha Atul, "Designing single guide RNAs for CRISPR/Cas9" (2019). *Master's Projects*. 730.
DOI: https://doi.org/10.31979/etd.6zqq-pdc4
https://scholarworks.sjsu.edu/etd_projects/730

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Designing single guide RNAs for CRISPR/Cas9

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Neha Atul Bhagwat

May 2019

© 2019

Neha Atul Bhagwat

ALL RIGHTS RESERVED

The Designated Project Committee Approves the Project Titled

Designing single guide RNAs for CRISPR/Cas9

by

Neha Atul Bhagwat

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2019

Dr. Sami Khuri	Department of Computer Science
Dr. Philip Heller	Department of Computer Science
Dr. Wendy Lee	Department of Computer Science

ABSTRACT

Designing single guide RNAs for CRISPR/Cas9

by Neha Atul Bhagwat

Researchers have been working towards development of tools to facilitate regular use genome engineering techniques. In recent years, the focus of these efforts has been the Clustered Regularly Interspaced Short Palindromic Repeats(CRISPR)/CRISPR associated(Cas) systems. These systems, while found naturally in bacteria and archaea as an immunity mechanism, can be used for genome engineering in eukaryotes.

There are three major computational challenges associated with the use of CRISPR/Cas9 in genome engineering for mammals - identification of CRISPR arrays, single guide RNA design and minimizing off-target effects. This project attempts to solve the problem of single guide RNA design using a novel approach.

Researchers have been trying to solve the problem by using different machine learning classification algorithms. The algorithms have been trained to use the sequential and structural properties of single guide RNAs (sgRNAs). This project explores the use of a neural network based approach to solve the sgRNA design problem. A form of the Recurrent Neural Network (RNN) called the Long Short Term Memory (LSTM) model can be used as feature-less classification model to differentiate between functional and non-functional single guide RNAs.

The project covers different experiments conducted using Support Vector Machine and Random Forest classifiers using sequential and structural features to identify the most potent sgRNAs in a given set of input sgRNAs. It also summarizes the implementation of the LSTM model and its results, along with the cross-validation results for each of these models. Through these results, it has been observed that LSTMs perform better than existing models such as Random Forest Classifiers and

Support Vector Machines and give results comparable to existing tools.

ACKNOWLEDGMENTS

I want to express my sincere gratitude to my advisor Dr. Sami Khuri for his support and mentorship. Without his additional classes to help me understand the basics of CRISPR, regular guidance and valuable inputs, this project would not have been possible.

I would also like to extend my gratitude to my committee members - Dr. Philip Heller and Dr. Wendy Lee for their time and effort.

In addition, a special thanks to Dr. Natalia Khuri for her timely advice and inputs that changed the path of this project and directed it towards better results.

TABLE OF CONTENTS

CHAPTER

1 Introduction 1

1.1 Background 2

2 The CRISPR-Cas9 System 3

2.1 The Natural CRISPR-Cas9 System 3

2.2 The engineered single guide RNA chimera 5

3 Importance of designing single guide RNAs 7

4 Existing Datasets 9

5 Features of single guide RNAs 12

5.1 Sequential Features 12

5.2 Structural Features 15

6 Classification models for designing single guide RNAs 17

6.1 Classification models in sgRNA design 17

7 Methods 20

7.1 Dataset Collection 20

7.2 Feature Extraction 21

7.2.1 Sequential Features 21

7.2.2 Structural Features 23

7.3 Feature and Sequence Encoding 24

7.3.1 One Hot Encoding 24

7.3.2 Label Encoding 25
7.3.3 Sequence Encoding 26
7.4 Machine Learning Classification Models 28
7.5 Long Short Term Memory Classification Model 29
8 Results 31
9 Conclusion 33
LIST OF REFERENCES 35

LIST OF TABLES

1	One hot encoding for categorical variables	25
2	Position 1 nucleotide features in five sgRNAs from the Brunello Dataset	25
3	One hot encoded position 1 nucleotide features in five sgRNAs from the Brunello Dataset	26
4	Labels for order 1 position-----wise nucleotide feature	26
5	Position 1 nucleotide features in five sgRNAs from the Brunello Dataset	27
6	Labels for order 1 position-----wise nucleotide feature	28
7	Encoded sequences for sgRNA sequences from the Brunello dataset	28
8	Comparison of results of SVM and RF classifiers	32
9	Comparison of results of LSTM classifiers with different configurations	32

LIST OF FIGURES

1	The Natural CRISPR-Cas system [1]	4
2	Difference between the natural CRISPR—Cas9 system and sgRNA chimera [2]	6
3	Distribution of cross-----validation accuracies of different models .	32

CHAPTER 1

Introduction

Once the DNA double helix was discovered, scientists worked towards developing techniques of modifying, building and breaking DNA[3]. Following the development of these techniques, the focus moved to techniques aimed at making site-specific modifications to genomes. Genome editing is an important research topic in today's day and age for several reasons. The applications of this research vary from disease prevention and cure to resurrection of species and creation of new, healthy foods. While genome editing is an attempt to bring positive change in different domains and is already being used in some domains, the widespread adoption of any genome editing technique is subject to technical barriers and ethical concerns[4].

CRISPR stands for Clustered Regularly Interspaced Short Palindromic Repeats. Cas9 is the CRISPR-associated enzyme that is instrumental in cutting the DNA at the targeted locus. The CRISPR-Cas9 is becoming increasing popular in the field of genome engineering due to advantages such as simplicity, cost-effectiveness, easy technology and capability to precisely target or manipulate the genomic loci[3]. While the CRISPR system is capable of precisely targeting the genomic loci, there is a possibility that the incorrect locus is targeted. The target sequence used in the CRISPR system may occur elsewhere in the genome leading to edits at the incorrect location. Modifications made by the CRISPR system at the incorrect site are referred to as off-target effects. There is still a pressing need to verify that the correct sites are being targeted with minimum to no off-target. Off-targets can potentially be lethal and it is imperative to improve the system so as to increase on-target specificity before the system becomes feasible for regular use.

1.1 Background

Traditional approaches for genome editing include using oligonucleotides, Zinc Finger Nucleases (ZFNs) and TAL Effector Nucleases (TALENs)[3]. In an early approach used for genome editing, disease-causing genes were inactivated using oligonucleotides. An oligonucleotide is a polynucleotide with few nucleotides that can be used to stop the transcription of a disease-causing gene by means of an antisense specific to the target gene. Another approach uses small molecules called siRNAs to disrupt the expression of a disease-causing gene, rendering it incapable of causing any harm. A newer approach uses ZFNs and TALENs attached to sequence-specific DNA to introduce breaks at specific positions. Following several experiments using the last approach, ZFNs and TALENs have been engineered for use in many organisms. These approaches, however, face problems in protein design and synthesis and are expensive[4].

The field of genome editing is undergoing a transformation due to the easy genome engineering technology called CRISPR-Cas9. The CRISPR-Cas9 system is developed from the type II CRISPR system in bacteria. The naturally occurring CRISPR system provides immunity in bacteria. Recent work has made it significantly easier to use CRISPR-Cas9 in mammalian genome engineering by reducing the number of components required to implement it[5, 6].

In the next chapter, we go over the natural CRISPR-Cas9 system and modifications made to the natural system to facilitate its adoption in the field of genome engineering.

CHAPTER 2

The CRISPR-Cas9 System

In order to understand the use of CRISPR-Cas9 in genome engineering and the revolution that it has brought, it is important to understand the naturally occurring CRISPR-Cas9 system and how its components have been modified to make it more applicable.

2.1 The Natural CRISPR-Cas9 System

In 1987, Yoshizumi Ishino and colleagues at Osaka University found short direct repeats separated by short sequences in Escherichia coli[3, 7]. Subsequently, similar repeats interspaced with short sequences were found in several bacteria and archaea. Numerous predictions and proposals were made about their role, including their potential significance in DNA repair [8, 9]. In 2006, it was proposed that the Cas system is an adaptive defense technique that uses RNAs as signatures of previous attacks by a virus or phage. This was followed by a spurt in experiments to understand the Cas system and its use in genome engineering, which has continued till date [10, 11].

The natural CRISPR-Cas arrays consist of identical direct repeats separated by spacers acquired from the DNA of an invading virus of phage[3].

The natural process of adaptive immunity by using CRISPR-Cas9 system occurs in three stages —Adaptation, Expression and Interference [1]. This process can be easily understood from Figure 1 which is part of the research by Donohoue et al. [1].

1. **Adaptation**: In adaptation, the host organism is attacked by an invading virus [1]. The protospacer is a sequence of the attacking virus that is acquired by the host organism and saved as a spacer within the CRISPR array. As the name suggests, a protospacer adjacent motif (PAM) is a very short sequence

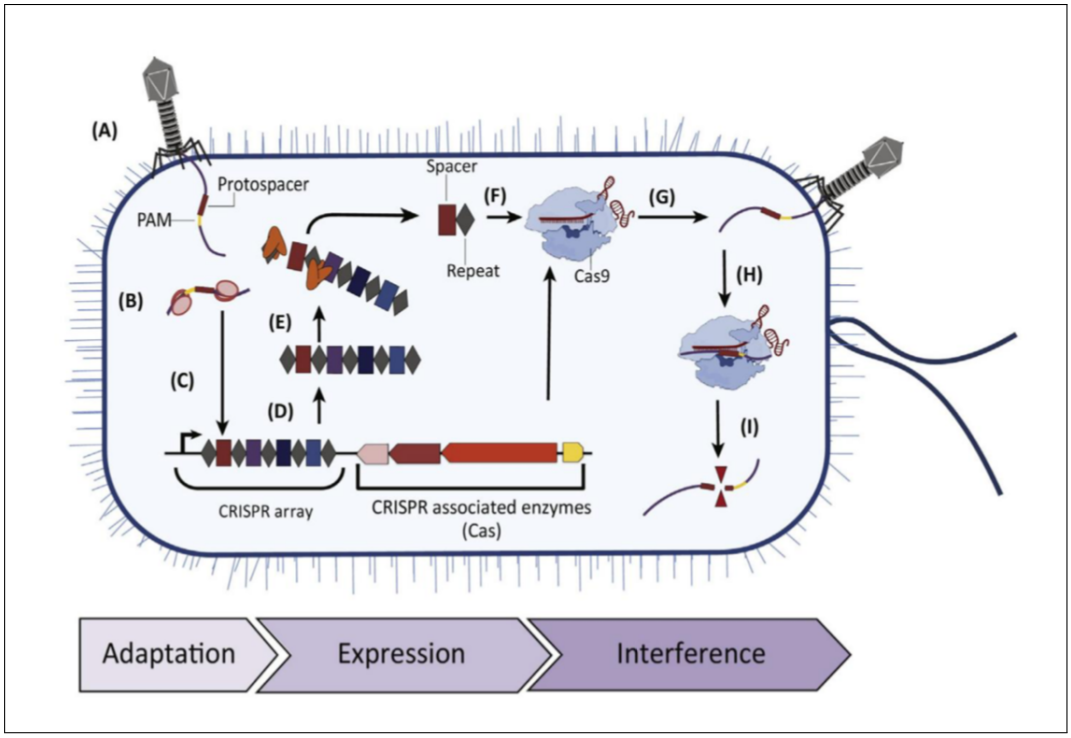

Figure 1: The Natural CRISPR-Cas system [1]

that lies next to the protospacer. The Cas proteins recognize the PAM and start targeting the nucleic acid based on its location. The Cas genes are expressed and bound to the protospacer sequence of the DNA. The DNA sequence is then incorporated into the CRISPR array as a spacer such that there is a repeat on both sides of the spacer.

2. **Expression**: When the same virus or phage attacks for the second time, the expression process starts. This process begins with the transcription of the complete CRISPR array. Repeat-spacer elements are processed to form the CRISPR RNA (crRNA). The crRNA binds to the Cas nuclease and leads to the formation of the Cas:crRNA complex. The Cas nuclease is sometimes referred to as the trans-activating crRNA (tracrRNA) and the complex is then called the tracrRNA:crRNA complex.

3. **Interference**: This complex then looks for the complementary protospacer in the invading DNA. The invading DNA is cleaved at the location where the complementary protospacer is found next to the PAM sequence.

2.2 The engineered single guide RNA chimera

In 2011, it was discovered that the tracrRNA was essential for crRNA maturation[12]. In 2012, it was shown that the Streptococcus Pyogenes CRISPR-Cas9 protein was a dual RNA- guided endonuclease[3, 5]. The tracrRNA:crRNA complex was found to be crucial for directed DNA cleavage. Thus, the natural CRISPR—Cas9 system consists of two functionally important components in the interference phase —tracrRNA and crRNA. These two components were then engineered into a single guide RNA (sgRNA)[2]. Figure 2 effectively differentiates between the natural system and the modified system. The sgRNA retained two properties —the 20 nucleotide sequence at its 5'end that is complementary to the target site and can be bound to it by Watson- Crick pairing, and the structure at the 3'end that binds to Cas 9. This development made the process of genome engineering easier and cost-effective. Zinc Finger Nucleases and TAL Effector nucleases required protein design and synthesis for each target DNA site. The CRISPR—Cas9 system on the other hand only needs a change in the single guide RNA. These factors have led to the widespread adoption of the CRISPR-Cas9 system in the genome engineering field.

The next chapter explains the importance of designing single guide RNAs in order to successfully use them in the field of genome engineering.

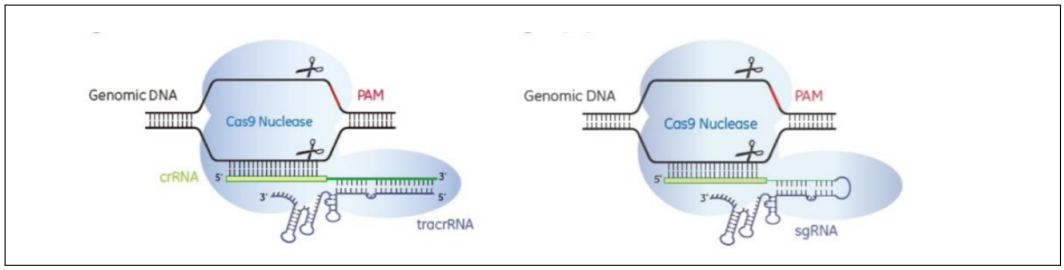

Figure 2: Difference between the natural CRISPR—Cas9 system and sgRNA chimera [2]

CHAPTER 3

Importance of designing single guide RNAs

Even though CRISPR-Cas9 is a cellular immune system, the genome engineering technique associated with the use of Cas9 protein and the single guide RNA (sgRNA) is often referred to as the CRISPR-Cas9 system in the literature and media [3]. The term CRISPR is actually related to the CRISPR arrays and not the technique that is used for genome engineering. Designing single guide RNAs (sgRNAs) is an important problem to solve in order to make CRISPR/Cas9 systems reliable.

Much research hypothesizes that the functional single guide RNAs are structurally and sequentially different from the non-functional single guide RNAs [6, 13, 14]. Potential guide RNA sequences are extracted from genomic sequences based on the location of a protospacer adjacent motif (PAM). In case of the Cas9 system, the PAM sequence is NGG (any nucleotide followed by two guanines). The length of a single guide RNA is 20 nucleotides [6]. Thus, the 20 nucleotides adjacent to a PAM can be considered as a potential sgRNA.

It is highly likely that the same sequence of 20 nucleotides is present elsewhere in the genome. When a sgRNA binds at the targeted genomic locus, it is said to be on-target. However, a sgRNA can potentially bind at one or more unintended sites resulting in non-specific genome editing at incorrect loci [15, 16]. Whether or not significant off-target effects are seen in CRISPR-Cas9 is still being debated [17]. But it is important to minimize the off-target effects that may appear as a result of genome engineering using CRISPR-Cas9 in order to enable efficient genome engineering [13]. Thus, predicting on-target efficacy and reducing off-target effects in an important computational challenge to make genome-editing using CRISPR/Cas9 systems feasible [18].

The process of designing single guide RNAs involves identifying the 20-nucleotide sequences adjacent to a protospacer adjacent motif that can potentially function as sgRNAs. The learning algorithm is then used with these potential sgRNAs as input to classify them into 2 groups - functional and non-functional sgRNAs based on the features that were used to train the learning algorithm.

In the next chapter, the process of data collection followed by previous researchers is explained in detail.

CHAPTER 4

Existing Datasets

Data collection, in single guide RNA design, involves two steps —collection of genomic sequences and collection of potential single guide RNAs from these genomic sequences. This step is not necessarily part of every sgRNA design research. For example, while the research for the Azimuth tool by the Genetic Perturbation Platform involved extensive data collection, the research for the tool WU-CRISPR made use of the Avana and Asiago libraries that were developed as a result of Azimuth [14]. This chapter goes over the process of data collection that was followed to create datasets by researchers.

The initial dataset for the Avana and Asiago libraries was created by targeting the transcripts listed by the Consensus Coding Sequence Database (CCDS) [13]. CCDS consists of 18,675 genes for the human genome and 20,077 genes for the mouse genome [19]. For a gene with over one CCDS ID, the one with the smaller transcript was chosen. The NGG protospacer adjacent motif (PAM) was annotated along both strands. The single guide RNAs were chosen for inclusion from these annotated sequences by dividing them into tiers based on three criteria. One of these criteria is based on the rule set 1 on-target efficacy score. This rule set was created as a result of an earlier research by the GPP group [14]. Through step-by-step relaxation of criteria, 6 sgRNAs were chosen per gene. The three criteria are as follows:

1. **Criterion A**:
    - Target site in 0-25% of the protein coding region
    - Target site in 25-50% of the protein coding region
    - Target site in 50-75% of the protein coding region
    - Target site in 70-100% of the protein coding region

2. **Criterion B**:

    - 13 nucleotides in the sgRNA sequence are unique
    - 17 nucleotides in the sgRNA sequence are unique
    - 20 nucleotides in the sgRNA sequence are unique
    - Sequence is not unique

3. **Criterion C**:

    - Rule set 1 on-target score in the range 0.9-1.0
    - Rule set 1 on-target score in the range 0.8-0.9
    - Rule set 1 on-target score in the range 0.7-0.8
    - Rule set 1 on-target score in the range 0.6-0.7
    - Rule set 1 on-target score in the range 0.5-0.6
    - Rule set 1 on-target score in the range 0.4-0.5
    - Rule set 1 on-target score in the range 0.3-0.4
    - Rule set 1 on-target score in the range 0.2-0.3
    - Rule set 1 on-target score in the range 0-0.2

The dataset created using CCDS and the above criteria can be used for designing single guide RNAs with high on-target specificity and low off-target effects.

The research for the tool DeepCRISPR also involved a dataset creation step. In this step, all potential sgRNAs (20 nucleotide sequences) adjacent to a NGG protospacer adjacent motif were collected from the coding and non-coding regions of the human genome [18]. This resulted in approximately 0.68 billion sgRNA sequences for the human genome. This acts as a large unlabeled dataset for the sgRNA design process.

While the dataset creation step is not followed in all models, each model uses different features to classify the single guide RNAs. The next chapter summarizes

the two main families of features used in different models and explains in detail the individual features in each.

CHAPTER 5

Features of single guide RNAs

Several examples of research and single guide RNA (sgRNA) design tools make use of the sequential and structural features to differentiate between a functional and non-functional sgRNA with the help of machine learning models.

5.1 Sequential Features

The first research group to include sequence features in evaluating potency of single guide RNAs was the Genetic Perturbation Platform [14]. Position-wise nucleotide composition and contiguous nucleotide presence are some sequence-based characteristics of single guide RNAs that have been considered in the design of sgRNAs.

A comprehensive list of sequence-based features of sgRNAs that have been considered in past research is as follows:

1. **Position-specific sequential features**:

   (a) **Order 1 position-wise nucleotide** : This feature takes into consideration which nucleotide is present at each position of the single guide RNA [6, 13, 20]. This is a categorical variable as each position can have one of four values —'A', 'C', 'T' or 'G'. In order to make this feature usable in machine learning models, this feature is converted into one-hot encoded vectors [13]. For each position, 4 binary vectors are created which can take the value 0 or 1. Each binary vector corresponds to a possible nucleotide that can be found at this position. Since each position can have a total of four possible nucleotides, 4 vectors are created for each position.

(b) **Order 2 position-wise nucleotides** : Similar to the order 1 position-wise nucleotide feature, this feature considers consecutive pairs of nucleotides as a feature [6, 13, 20]. Order 2 position-wise nucleotides are also categorical features and in order to use them to train the machine learning model, they are converted into 16 one-hot encoded vectors for each position [13]. For each position, 16 binary vectors are created which can take the value 0 or 1. Each binary vector corresponds to a possible nucleotide pair that can be found at that position. A nucleotide pair can have four different nucleotides in the first position and four different nucleotides in the second position, leading to a total of 16 possible pairs. Therefore, 16 binary vectors are created for each position.

(c) **Nucleotides relative to the PAM**: For Cas9 systems, the protospacer adjacent motif (PAM) is considered as 'NGG'. The two nucleotides adjacent to the two contiguous guanines ('NGGN') can also be considered as features in the classification model [8]. As this is a position specific feature with two nucleotides, it can be converted to 16 one-hot encoded vectors. There can be four different nucleotides in each of the positions in the nucleotide pair, resulting in 16 unique combinations. Therefore, 16 binary vectors which can take the value 0 or 1 are created for this feature.

2. **Position-independent sequential features**:

(a) **Nucleotide count**: The count of each individual nucleotide in the single guide RNA can also be considered as a feature [6, 13]. It has been found that the frequency of adenine can be a strong feature in differentiating

between functional and non-functional single guide RNAs (sgRNAs) [6, 13]. On the other hand, the nucleotide count of guanine, cytosine and thymine cannot be considered as a good indicator of whether a single guide RNA is functional or not. Nucleotide count will give four features for every single guide RNA —each of the four features will represent the count of each of the four nucleotides.

(b) **Dinucleotide count**: Dinucleotide count feature represents the count of every potential dinucleotide that can be found in a single guide RNA (sgRNA) [6, 13]. There are 16 possible dinucleotides that can be found in a sgRNA. Thus, dinucleotide count results in 16 features. Out of all the dinucleotides, it has been found that 'GG' is the most indicative dinucleotide in classification between functional and non-functional sgRNAs.

(c) **Trinucleotide count**: Similar to the nucleotide count and dinucleotide count features, trinucleotide count considers every sequence of three nucleotides that can be found in a sgRNA [6]. Out of all the trinucleotides, it has been found in past research that 'GGG' is the most important trinucleotide to be considered in the classification model for designing sgRNAs.

The count of contiguous nucleotides is stopped at trinucleotide count because any sgRNA with over three contiguous bases is eliminated. Repetitive bases are indicative of poor functionality in sgRNAs. Presence

of 4 contiguous adenines, 4 contiguous guanines, 5 contiguous cytosines or 5 contiguous thymines is considered as repetitive and the sgRNA is removed from further consideration.

5.2 Structural Features

The accessibility of individual nucleotides plays an important role in determining the potency of a single guide RNA. Only if each nucleotide is accessible can the sgRNA bind at the appropriate genomic loci. Self-folding free energy, structural stability and thermodynamic features play an important role in determining the structural accessibility of any nucleotide sequence.

A comprehensive list of structural features of sgRNAs that have been considered in past research is as follows:

1. **Self-folding free energy**: Secondary structures of the single guide RNA (sgRNA) can be calculated using RNA fold [6]. Thus, the self-folding free energy of the sgRNA can be found and used as a feature in the classification model.

2. **Accessibility of individual nucleotides**: The accessibility of nucleotides varies at specific positions for functional sgRNAs as compared to non-functional sgRNAs [6]. For example, the three nucleotides at positions 18-20 are more accessible in case of functional sgRNAs as compared to non-functional sgRNAs.

3. **Structural stability of sgRNA**: The structural stability of a single guide RNA (sgRNA) can be estimated based on its GC content [6, 13]. It has been observed that non-functional sgRNAs have higher GC content as compared to

functional sgRNAs.

4. **Melting points of sgRNA**: Melting point of a sgRNA can be used as a thermodynamic feature in the classification model [13]. The melting points can be estimated used the Biopython package.

5. **Melting points of parts of sgRNA** : This thermodynamic feature can be divided into three features —melting point of 5 nucleotides next to the protospacer adjacent motif (PAM), the next 8 nucleotides and the next 5 nucleotides [13]. Like the previous feature, these values can be calculated using the Biopython package.

Depending on the tool and the classification model used in the tool, different subsets of the features listed above are used to train the classification models. The next chapter describes the classification models used by some of these tools.

CHAPTER 6

Classification models for designing single guide RNAs

Humans are capable of classifying objects that they can see, hear or feel into categories. In order to do this, we identify characteristic features of the objects that we have to classify and develop a relationship between the features and the object. A machine learning based classification model works in a similar manner. To use a machine learning model for classification we need a set of features of each object and a set of class labels representing all the categories or classes into which the objects can be divided [21, 22].

Classification involves a set of data with pre-defined classes and features (attributes). Based on this data, which is commonly referred to as the training set, the model is built using a learning algorithm [21]. The model represents a relation between the set of features and the class label. This model can then be used to predict the classes of data for which features are known, but the class labels are unknown. Support vector machines, decision trees, nearest neighbors are commonly used approaches for classification models [23]. Besides these traditional machine learning approaches, deep learning can also be used to build a classification model.

6.1 Classification models in sgRNA design

Classification models for sgRNA design incorporate some of the features listed in the previous chapter to train the model to identify the differences between functional and non- functional sgRNAs.

Some of the classification models used in different research publications and tools are listed next:

1. **WU-CRISPR**: This Washington University tool used a support vector machine (SVM) with a radial basis function (RBF) kernel to differentiate between functional and non-functional single guide RNAs [6]. The RBF kernel is useful in classification of data that is not linearly separable.

2. **Genetic Perturbation Platform (GPP)**: The GPP group has developed two tools for sgRNA design. The first tool did not use a machine learning model for sgRNA design. The research for the second tool, developed in 2016, made use of several models in an attempt to identify the one that performed best [13]. The predictive classification models used in this tool are:

    - Linear Regression
    - L1 regularized Linear Regression
    - L2 regularized Linear Regression
    - Support Vector Machine plus Logistic Regression
    - Random Forest Classification
    - Gradient boosted Regression Tree
    - L1 Logistic Regression (Classifier)
    - Support Vector Machine Classification

3. **DeepCRISPR**: DeepCRISPR is one of the first sgRNA design tools that makes use of deep neural networks for the purpose. Specifically, it uses a deep convolutional de-noising neural network to implement a deep unsupervised representation learning algorithm [18].

4. **Study on key sequence features of sgRNAs**: This study, which only takes into consideration the sequential characteristics of single guide RNAs, makes use of a support vector machine to implement a classification model for functional

and non-functional sgRNAs [20].

5. **Library-on-library approach**: This study also implemented a support vector machine to capture relation between functional and non-functional sgRNAs [24]. Ten-fold cross validation and test data were used to evaluate the accuracy of the model.

The next chapter explains the different steps followed in this research and the experiments conducted.

CHAPTER 7

Methods

7.1 Dataset Collection

The Brunello and Brie datasets published by Doench and colleagues were used as a basis for the dataset used for classification [13]. The Brunello and Brie datasets contain the 4 best-ranked sgRNAs for each target transcript belonging to the human and mouse genome.

While this dataset provided a list of functional single guide RNAs, implementation of classifiers requires both functional and non-functional sgRNAs. In order to create a negative dataset, the accession numbers in the Brunello and Brie datasets were used. With these accession numbers and the BioPython package, the genomic sequence for each accession number was retrieved from NCBI [25, 26]. Single guide RNAs are sequences of 20 nucleotides found on the basis of the presence of a protospacer adjacent motif (PAM). Negative instances were found by scanning the genomic sequences for a PAM (NGG motif) in both directions. Each 20 nucleotide sequence adjacent to a NGG motif was used considered for inclusion in the dataset. If the sgRNA was found in the original dataset, it was assigned the class 1. Other sgRNAs were assigned the class 0. At the end of this step, a dataset was created with functional and non-functional sgRNAs.

The non-functional sgRNAs were very high in number in comparison to the functional sgRNAs. In order to build an effective classifier, the number of samples from each class should be more or less the same. The number of non-functional sgRNAs were restricted to the number of functional sgRNAs for each target transcript. This led to the creation of a balanced dataset. Each single guide RNA stored in this dataset includes the following details:

1. Target Transcript

2. Strand (sense/antisense)

3. sgRNA Target Sequence

4. Target Context Sequence

5. PAM Sequence

6. start

7. end

8. class (0 or 1)

7.2 Feature Extraction

To implement the feature-dependent machine learning classification models, sequential and structural features were used. The following features were extracted for each single guide RNA:

7.2.1 Sequential Features

Similar to earlier research, the sequential features used include position-specific and position-independent features. The following sequential features were used to identify single guide RNAs:

1. **Position-specific Sequential Features** :

   - **Order 1 position-wise nucleotide**: This feature holds the letter representing the nucleotide present at each position of the single guide RNA - A, C, G or T - as explained in 1a. Depending on the classifier used, these representations are converted to labeled features or one-hot encoded features.

     Labeling the one-hot encoded features involves converting each feature to a numerical representation. For example, A is represented as 0, C is

represented as 1, and so on. In order to label the order 1 nucleotides, the LabelEncoder function from the preprocessing package of the sklearn API was used [27].

To convert the categorical features into one-hot encoded features, each position-wise feature was converted into four one-hot encoded features - one for each possible nucleotide that could be found at each position. There are a total of 20 order 1 features for each sgRNA. When converted to one-hot encoded features, the total number of order 1 nucleotide features becomes 80. To convert one-hot encoded features to order 1 nucleotide features, the OneHotEncoder function from sklearn API's preprocessing package was used [27].

- **Order 2 position-wise nucleotides**: Order 2 position-wise nucleotides features consider consecutive pairs of nucleotides as features as explained in 1b. Similar to order 1 nucleotide features, order 2 features are also categorical in nature. Hence, they have to be converted into labeled or one-hot encoded features depending on the classifier being used.

  In case of order 2 nucleotides, labeling involves conversion of each possible pair of dinucleotides into numerical labels. For example, AA is represented as 0, AC is represented as 1, AG is represented as 2, and so on. In order to convert the string features into numerical label values, the LabelEncoder function from the preprocessing package of the sklearn API was used [27]. Similar to one-hot encoding in case of order 1 nucleotides, order 2 nucleotide features were converted into one-hot encoded features by using the OneHotEncoder function from the preprocessing package of the sklearn API [27]. When 19 order 2 nucleotide features for each sgRNA are converted into one-hot encoded features, a total of 304 features are obtained.

2. **Position-independent Sequential Features**:

   - **Nucleotide count**: This feature considers the count of each nucleotide in the single guide RNA as a feature. Thus, there are 4 features representing nucleotide count - one for adenine, cytosine, guanine and thymine each. Each sgRNA has 20 nucleotides.
   - **Dinucleotide count**: In this position-based feature, the count of every possible dinucleotide is considered as a feature. Each sgRNA has 19 dinucleotides. As there are 16 dinucleotides in all, 16 features represent dinucleotide count in the feature set.
   - **Trinucleotide count**: Trinucleotide count takes the count of each possible trinucleotide in the sgRNA. A sgRNA has 18 trinucleotides. In all, there are 64 possible trinucleotides. Therefore, 64 features represent the trinucleotide count in the feature set.

7.2.2 Structural Features

A single guide RNA's structure, self folding energy and thermodynamic features are symbolic of its stability. The following structural features have been extracted for the classification models:

1. **Melting Points of sgRNA**: The melting point of the entire single guide RNA is calculated using the Nearest Neighbor function in the BioPython package [25]. The nearest neighbor based melting point calculates melting temperatures based on nearest neighbor thermodynamics.

2. **Melting points of parts of sgRNA**: The feature is represented by 3 values. These 3 values are the melting points of the 3 parts of the sgRNA as explained in 5. Each melting point is calculated using the Nearest Neighbor function in the BioPython package [25].

3. **Self-folding free energy**: The self-folding free energy of an sgRNA is calculated using the ViennaRNA package.

7.3 Feature and Sequence Encoding

The features listed above include categorical and numerical features. For example, if we consider the position-wise nucleotide features, the feature values are categorical. There is a fixed set of possible values (categories) that each of these features can hold and the feature value is one of these categories. Similarly, if we consider the nucleotide, dinucleotide or trinucleotide counts, we can see that these values are numerical in nature. Based on the classification model, the features need to be encoded to ensure that the model can understand the features and derive meaningful insights.

7.3.1 One Hot Encoding

In order to incorporate categorical features in Support Vector Machine based classification, it is essential to convert them into one-hot encoded features. One hot encoding represents categorical variables as vectors. One hot encoding involves 2 steps:

1. **Fetching all possible categories**: In this process, all possible values that a feature can take are listed.

2. **Converting one feature into multiple one-hot encoded features**: Using the list of features obtained in the previous step, each feature is converted into a binary vector. This binary vector has the value 0 for all categories except the category to which it belongs.

Consider the order 1 position-wise nucleotide feature and the single guide RNA - 'GATCCACACTCCCAACAAGG'. At position 1, the single guide RNA has the nucleotide 'G'.

Table 1: One hot encoding for categorical variables

A	**C**	**G**	**T**
0	0	1	0

Table 2: Position 1 nucleotide features in five sgRNAs from the Brunello Dataset

sgRNA	**Position 1**
GATCCACACTCCCAACAAGG	G
CTGTTGTCACCATAACAACA	C
GAACCATACACAGACCACAA	G
TCATTGCCGAGTAATACAAA	T
GCACCGCCACCATTGCACCA	G

Step 1 of the one-hot encoding process involves listing all possible values that the feature can take. At any position, a single guide RNA can have one of the four nucleotides - 'A', 'C', 'G' or 'T'.

For this sgRNA, the one-hot encoded feature corresponding to 'G' will have have the value 1 and the other 3 features will have the value 0. Therefore, the single feature corresponding to order 1 position wise nucleotide will be divided into 4 one-hot encoded features as seen in Table 1.

Table 2 and Table 3 show the order 1 position-wise nucleotide feature and the corresponding one-hot encoded features for 5 single guide RNAs from the Brunello dataset.

7.3.2 Label Encoding

Label Encoding is an essential step for the Random Forest Classifier to understand text-based categories in categorical features. Label encoding involves the following three steps:

1. **Fetching all possible categories**: In this process, all possible values that a feature can take are listed.

Table 3: One hot encoded position 1 nucleotide features in five sgRNAs from the Brunello Dataset

sgRNA	**A**	**C**	**G**	**T**
GATCCACACTCCCAACAAGG	0	0	1	0
CTGTTGTCACCATAACAACA	0	1	0	0
GAACCATACACAGACCACAA	0	0	1	0
TCATTGCCGAGTAATACAAA	0	0	0	1
GCACCGCCACCATTGCACCA	0	0	1	0

Table 4: Labels for order 1 position-wise nucleotide feature

Position 1 feature value	**Label**
A	0
C	1
G	2
T	3

2. **Assigning labels to the categories**: For each of the categories listed in the previous step, a numerical value is assigned to the categories.

3. **Converting the feature values into labels**: The feature values are then converted into numerical labels based on the mapping generated in the previous step.

Consider the feature order 1 position-wise nucleotide feature. According to the first step, the possible values of this feature can be listed as - 'A', 'C', 'G' and 'T'.

The next step involves assigning numerical values to each of these values. This labeling can be observed in Table 4.

The last step involves assigning the labels to the respective features. Table 5 lists the original feature value for order 1 position-wise nucleotide value.

7.3.3 Sequence Encoding

Sequence encoding is used specifically for the Long Short Term Memory (LSTM) model. While sequence encoding is similar to label encoding, there are no features

Table 5: Position 1 nucleotide features in five sgRNAs from the Brunello Dataset

sgRNA	**Order 1 Position-wise nucleotide**	**Encoded Label**
GATCCACACTCCCAACAAGG	G	2
CTGTTGTCACCATAACAACA	C	1
GAACCATACACAGACCACAA	G	2
TCATTGCCGAGTAATACAAA	T	3
GCACCGCCACCATTGCACCA	G	2

involved in sequence encoding. Each input single guide RNA sequence is converted into a sequence of numerical labels to ensure that the LSTM model understands the input sequences.

Sequence encoding involves 3 steps:

1. **Fetching all possible subsequences for each position of the input sequence**: In case of the sgRNA design problem, our input sequences are single guide RNAs. For sgRNAs, each position can have one of the four nucleotides - 'A', 'C', 'G' and 'T'. Therefore, the possible subsequences for every position remains the same.

2. **Assigning labels for each unique input subsequence**: In the sgRNA design problem, as we have just 4 unique subsequences that can form the input sequence, we can create the subsequence-label mapping as seen in Table 8.

3. **Creating a numerical sequence from the input sequence**: For each sub-sequence (nucleotide) in the input sequence, the label for the sub-sequence is obtained from the mapping created in the earlier step. This step converts the input sequence into a list of numerical labels. For the first 5 sgRNAs in the Brunello dataset, the numerical sequences can be seen in Table 6.

Table 6: Labels for order 1 position-wise nucleotide feature

sgRNA	**Encoded Sequence**
A	0
C	1
G	2
T	3

Table 7: Encoded sequences for sgRNA sequences from the Brunello dataset

Sub-sequence	**Label**
GATCCACACTCCCAACAAGG	[2, 0, 3, 1, 1, 0, 1, 0, 1, 3, 1, 1, 1, 0, 0, 1, 0, 0, 2, 2]
CTGTTGTCACCATAACAACA	[1, 3, 2, 3, 3, 2, 3, 1, 0, 1, 1, 0, 3, 0, 0, 1, 0, 0, 1, 0]
GAACCATACACAGACCACAA	[2, 0, 0, 1, 1, 0, 3, 0, 1, 0, 1, 0, 2, 0, 1, 1, 0, 1, 0, 0]
TCATTGCCGAGTAATACAAA	[3, 1, 0, 3, 3, 2, 1, 1, 2, 0, 2, 3, 0, 0, 3, 0, 1, 0, 0, 0]
GCACCGCCACCATTGCACCA	[2, 1, 0, 1, 1, 2, 1, 1, 0, 1, 1, 0, 3, 3, 2, 1, 0, 1, 1, 0]

7.4 Machine Learning Classification Models

Traditional machine learning techniques can be used to build classification models with the help of training data and a set of predefined features. For single guide RNA design, sequential and structural features have been found to play an important role in determining the functionality of a sgRNA. Therefore, these features as listed above were used to for the classification model.

Two classification models were implemented using sequential and structural features - Support Vector Machine classifier and Random Forest classifier. Support vector machines try to identify a hyperplane in an N-dimensional space where N is the number of features used for classification such that the hyperplane effectively divides the set of inputs into distinct classes. Several hyperplanes may satisfy the purpose, but the aim is to identify a hyperplane that clearly distinguishes the points while

maintaining maximum distance between the points and the plane. Random Forest classifiers identify the class of the input sequence based on votes of a set of decision trees. These decision trees are created using subsets of the training data.

Using both sequential and structural features individually, the classifiers were trained to classify sgRNAs as functional or non-functional. Both implementations made use of the sklearn package [27].

7.5 Long Short Term Memory Classification Model

Recurrent Neural Networks (RNN) are networks with loops that allow the network to retain information. Due to the presence of these loops, RNNs can be used for applications that involve sequences. In traditional RNNs, the amount of context is very small in practice [28]. The effect of a particular input on the output may either blow up or diminish significantly as the input cycles through the loops in the RNN. This problem is referred to as the *vanishing gradient problem*.

After several solutions were proposed to solve the vanishing gradient problem, the Long Short Term Memory (LSTM) approach was proposed by Hochreiter and Schmidhuber in 1997 [28, 29]. LSTMs are a type of Recurrent Neral Networks (RNN) that learn long-term dependencies [30].

Sequence classification is a predictive problem that is solved by using LSTMs and word embeddings. As previous research has found relations between the sequential features of single guide RNAs and their functionality, we can treat sgRNA design as a sequence classification problem.

In case of text classification, a large vocabulary has to be created based on the words in the training sentences. This vocabulary is then used to label the input sequences. LSTMs require the input sequences to be equal in length. In order to make the sequences equal in length, padding is used. If a new word is encountered in the

test sentences that is not present in the training data, it is tagged as an unknown word. Thus, using LSTMs for text classification involves a lot of pre-processing. However, using LSTMs for sgRNA design is a simpler problem as it involves minimum pre-processing. The problem is reduced as the vocabulary can be reduced to 4 bases (words in traditional text classification) - A, C, G and T. The unknown tags are not required as the training and test data cannot have any bases besides the four listed earlier. The sgRNA sequences are always equal in length as each sgRNA is a 20 nucleotide sequence. Hence, the need to pad input sequences is eliminated.

An LSTM-based approach has not been used in the problem of single guide RNA design so far.

Several experiments were conducted to identify optimal configurations with best results. The next chapter goes over the results of these experiments.

CHAPTER 8

Results

Different subsets of the feature set were used with different feature engineering techniques to experimentally determine the best set of features. Table 8 summarizes the cross-validation accuracies and test set accuracy for each combination of dataset, feature set and feature engineering technique.

The Long Short Term Memory Network to classify functional and non-functional single guide RNAs was built using keras [31]. With an accuracy of 83.85%, this LSTM model classifies a single guide RNA provided as input into functional or non-functional sgRNA. Table 9 summarizes the results of the LSTM model with different configurations and experiments in detail.

Ten-fold cross-validation was performed for each of the models. Figure 3 compares the distribution of the cross-validation accuracies of each of the models.

Experiments were conducted to compare the results of different tools with individual sgRNAs. The results of the LSTM model were comparable to the results of existing tools like WU-CRISPR, CRISPR-ML (tool developed by the Genetic Perturbation Platform group in collaboration with Microsoft Research). It is difficult to prove the accuracy of any specific tool. The WU-CRISPR tool assigns a potency score or value to each sgRNA in the range 1-100, where sgRNAs with a score 1 are least functional and sgRNAs with a score 100 are highly functional. For the genomic sequences tested, the sgRNAs identified by the LSTM model as functional sgRNAs have a high potency value based on the WU-CRISPR tool. Similarly, the sgRNAs identified by the LSTM model as functional sgRNAs have a high on-target score according to the CRISPR-ML tool. Some sgRNAs identified by both WU-CRISPR and CRISPR-ML tool are not identified by the LSTM model as functional sgRNAs.

Table 8: Comparison of results of SVM and RF classifiers

Classifier	**Feature Set**	**Feature Engineering**	**Accuracy**
Linear SVM	Sequential Features	One Hot Encoding and Scaling	77.9%
Linear SVM	Structural Features	Scaling	56.4%
Random Forest	Sequential Features	Label Encoding	70.8%

Table 9: Comparison of results of LSTM classifiers with different configurations

Classifier	**Epochs**	**Sequence used**	**Dataset used**	**Accuracy**
Sequential API Keras	20	sgRNA sequence	Brie	71.2
Sequential API Keras	20	sgRNA sequence	Brunello	71.89
LSTM	20	sgRNA sequence	Brie	80.99
LSTM	20	sgRNA sequence	Brunello	81.88
LSTM	25	sgRNA sequence + PAM	Brie	82.78
LSTM	25	sgRNA sequence + PAM	Brunello	83.85

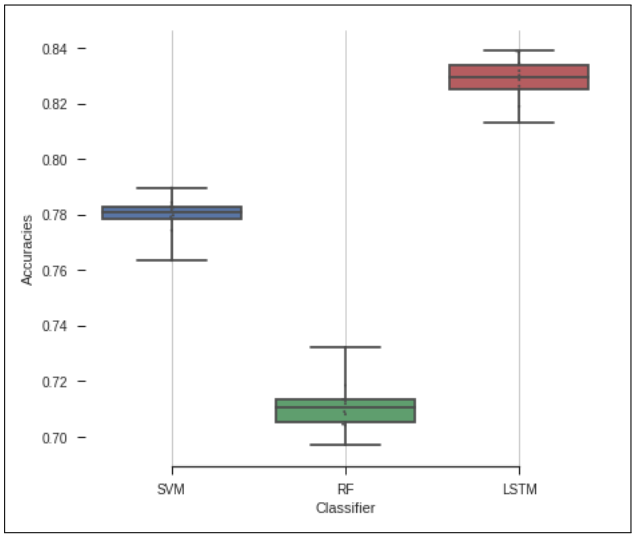

Figure 3: Distribution of cross-validation accuracies of different models

CHAPTER 9

Conclusion

In this project, we analysed several ways to design a single guide RNA (sgRNA) using different classifiers. Traditional machine learning algorithms and neural networks were both used to solve the problem. While traditional machine learning algorithms such as Support Vector Machines (SVM) and Random Forest (RF) Classifier used sequential and structural features of sgRNAs to differentiate between a functional and non-functional sgRNAs, the LSTM model was a type of featureless classification.

The LSTM model gave better results as compared to the SVM and RF models. Even though the LSTM model is featureless, the sequence encoded sgRNAs are used as input for the model. The LSTM model performs better when the protospacer adjacent motif (PAM) is used along with the 20-nucleotide sgRNA in classification. The 3-nucleotide PAM sequence is represented as NGG (any nucleotide followed by two guanines). Therefore, we can conclude that the N in the PAM sequence has a role to play in the classification. Use of structural features in sgRNA design is observed to not yield comparable results.

While the LSTM model and some existing tools give comparable results, it is recommended to use multiple tools and a consensus of their results when deciding whether or not a particular sgRNA is functional or not. Different tools use different features, datasets and classifiers to determine the functionality of a sgRNA, therefore using multiple tools allows one to consider all these features and algorithms before coming to a conclusion.

In the process of dataset creation, the number of non-functional sgRNAs extracted from each transcript was equal to the number of functional sgRNAs from the Brunello and Brie datasets found in the corresponding transcript. Therefore, if the Brunello or Brie dataset contained a certain number of functional sgRNAs, the same number of

potential non-functional sgRNAs were randomly selected from the list of all potential non-functional sgRNAs. It is possible that the random selection of the non-functional sgRNAs may have affected the results of the classifiers. As part of future extension of this work, the non-functional sgRNAs can be ordered using a metric and then the least functional sgRNAs from those can be included in the dataset. The results of the LSTM model can also be improved upon by trying different configurations. More neural network based classification algorithms can be tested to solve the sgRNA design problem.

LIST OF REFERENCES

[1] P. Donohoue, R. Barrangou, and A. May, ''Advances in industrial biotechnology using crispr-cas systems,'' *Trends in Biotechnology*, vol. 36, no. 2, pp. 134--146, February 2018.

[2] ''Crispr guide rna,'' https://dharmacon.horizondiscovery.com/gene-editing/crispr-cas9/crispr-guide-rna/, (Accessed on 12/16/2018).

[3] J. A. Doudna and E. Charpentier, ''The new frontier of genome engineering with crispr- cas9,'' *Science*, vol. 346, no. 6213, p. 1258096, November 2014.

[4] ''Genome editing,'' https://www.genome.gov/27569222/genome-editing/, (Accessed on 12/16/2018).

[5] M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, and E. Charpentier, ''A programmable dual rna guided dna endonuclease in adaptive bacterial immunity,'' *Science*, vol. 337, no. 6096, pp. 816--821, August 2012.

[6] 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, p. 218, November 2015.

[7] ''Breakthrough dna editor born of bacteria,'' https://www.quantamagazine.org/crispr-natural-history-in-bacteria-20150206/, (Accessed on 5/2/2019).

[8] K. S. Makarova, L. Aravind, N. V. Grishin, I. B. Rogozin, and E. V. Koonin, ''A dna repair system specific for thermophilic archaea and bacteria predicted by genomic context analysis,'' *Nucleic Acids Research*, vol. 30, no. 2, pp. 482--496, 2002.

[9] C. P. Guy, A. I. Majerník, J. P. J. Chong, and E. L. Bolt, ''A novel nuclease-atpase (nar71) from archaea is part of a proposed thermophilic dna repair system,'' *Nucleic Acids Research*, vol. 32, pp. 6176--6186, 2004.

[10] L. A. Marraffini and E. J. Sontheimer, ''Crispr interference limits horizontal gene transfer in staphylococci by targeting dna,'' *Science*, vol. 322, pp. 1843--1845, 2008.

[11] R. Barrangou, C. Fremaux, H. Deveau, M. Richards, P. Boyaval, S. Moineau, D. A. Romero, and P. Horvath, ''Crispr provides acquired resistance against viruses in prokaryotes,'' *Science*, vol. 315, no. 5819, pp. 1709--1712, 2007.

[12] E. Deltcheva, K. Chylinski, C. Sharma, K. Gonzales, Y. Chao, Z. A. Pirzada, M. R. Eckert, J. Vogel, and E. Charpentier, ''Crispr rna maturation by trans-encoded small rna and host factor rnase iii,'' *Nature*, vol. 471, no. 7340, pp. 602--607, March 2011.

[13] J. G. Doench, N. Fusi, M. Sullender, M. Hegde, E. Vaimberg, K. F. Donovan, I. Smith, Z. Tothova, C. Wilen, R. Orchard, H. W. Virgin, J. Listgarten, and D. Root, ''Optimized sgrna design to maximize activity and minimize off-target effects of crispr- cas9,'' *Nature Biotechnology*, vol. 34, pp. 184--191, January 2016.

[14] J. G. Doench, E. Hartenian, D. B. Graham, Z. Tothava, M. Hegde, I. Smith, M. Sullender, B. L. Ebert, R. J. Xavier, and D. E. Root, ''Rational design of highly active sgrnas for crispr-cas9âĂŞmediated gene inactivation,'' *Nature Biotechnology*, vol. 32, pp. 1262--1267, September 2014.

[15] S. Q. Tsai, Z. Zheng, N. T. Nguyen, M. Liebers, V. V. Topkar, V. Thapar, N. Wyvekens, C. Khayter, A. J. Iafarte, L. P. Le, M. J. Aryee, and J. K. Joung, ''Guide-seq enables genome-wide profiling of off-target cleavage by crispr-cas nucleases,'' *Nature Biotechnology*, vol. 33, pp. 187--197, December 2015.

[16] V. Pattanayak, S. Lin, J. P. Guilinger, E. Ma, J. Doudna, and D. R. Liu, ''High-throughput profiling of off-target dna cleavage reveals rna-programmed cas9 nuclease specificity,'' *Nature Biotechnology*, vol. 31, pp. 839--843, August 2013.

[17] ''Crispr off-targets: a reassessment,'' https://www.nature.com/articles/nmeth.4664, pp. 229--230, March 2018, (Accessed on 03/13/2019).

[18] G. Chuai, H. Ma, J. Yan, M. Chen, N. Hong, D. Xue, C. Zhou, C. Zhu, K. Chen, B. Duan, F. Gu, S. Qu, D. Huang, J. Wei, and Q. Liu, ''Deepcrispr: optimized crispr guide rna design by deep learning,'' *Genome Biology*, vol. 19, no. 1, June 2018.

[19] S. Pujar, N. A. O. Leary, C. M. Farrell, J. E. Loveland, J. M. Mudge, C. Wallin, C. Giron, and et al, ''Consensus coding sequence (ccds) database: a standardized set of human and mouse protein-coding regions supported by expert curation,'' *Nucleic Acids Research*, vol. 46, pp. D221--D228, January 2018.

[20] L. Chen, S. Wang, Y.-H. Zhang, J. Li, Z.-H. X. J. Yang, T. Huang, and Y.-D. Cai, ''Identify key sequence features to improve crispr sgrna efficacy,'' *IEEE Access*, vol. 5, pp. 26 582--26 590, 2017.

[21] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[22] S. B. Kotsiantis, ''Supervised machine learning: A review of classification techniques,'' *Informatica*, vol. 31, pp. 249--268, 2007.

[23] S. Sharma, J. Agarwal, S. Agarwal, and S. Sharma, ''Machine learning techniques for data mining: A survey,'' *IEEE Conference on Computational Intelligence and Computing Research*, pp. 1--6, 2013.

[24] R. Chari, P. Mali, M. Moosburner, and G. M. Church, ''Unraveling crispr-cas9 genome engineering parameters via a library-on-library approach,'' *Nature Methods*, vol. 12, pp. 823--826, 2015.

[25] P. Cock, T. Antao, J. Chang, B. Chapman, C. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. de Hoon, ''Biopython: freely available python tools for computational molecular biology and bioinformatics,'' *Bioinformatics*, vol. 25, pp. 1422--1423, 2009.

[26] S. F. Altschul, W. Gish, W. Miller, E. W.Myers, and D. J.Lipman, ''Basic local alignment search tool,'' *Journal of Molecular Biology*, vol. 215, pp. 403--410, 1990.

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

[28] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer-Verlag Berlin Heidelberg, 2012.

[29] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Computation*, pp. 1735--1780, 1997.

[30] C. Olah, ''Understanding lstm networks,'' August 2015, available at http://colah.github.io/posts/2015-08-Understanding-LSTMs/, Accessed: 2019-04-14.

[31] F. Chollet *et al.*, ''Keras,'' https://keras.io, 2015.