**CS 466 Project: Exploring Different Approaches in Predicting CRISPR Experiment Effectiveness**

Dave Istanto

**Abstract**

Clustered regularly interspaced short palindromic repeats (CRISPR) has grown in popularity due to its flexibility and versatility in genome editing [1]. Due to the nature of CRISPR systems, there is a demand for an accurate prediction model that predicts the effectiveness of the CRIPSR system given the variable component: guide RNA (gRNA). Multiple prior works have tackled this problem in both data-driven and physical-feature based models and resulted in a relatively effective prediction model. In this project, we attempt to improve these models by incorporating more complex machine-learning based prediction models and different approaches. We found that the conventional approach – classification approach - is more effective than the novel approach we investigated, regression approach. We also found that using a more complex prediction models yields greater accuracy in test data compared to the prediction models used by prior works. Furthermore, we also suggested that with improvement of dataset, addition of features and a more complex prediction model, higher test accuracy can be achieved.

**Prior works**

Multiple prior works have responded to the increase in popularity for clustered regularly interspaced short palindromic repeats (CRISPR) [1]. Most of these works have been predicting the effectiveness of the CRISPR DNA cleavage according to the property of the guide RNA.

Doench et. al have approached this problem using data-driven point of view. In their 2014 and 2016 papers, identifying which potential guide RNA features contributing to effectiveness of CRISPR systems have been explored via simpler machine learning classification algorithms, such as support vector machine (SVM) combined with logistic regression [2][3]. This approach has been shown to be popular and been exploited by other researchers as well, such as CHOPCHOP [4]. Furthermore, the prediction models were further strengthened using off-site factors to be included as well [3].

Zhang et. al however, took an additional approach by investigating the effectiveness of CRISPR systems directly from its physical features instead of only using data-driven approach [5]. uCRISPR further strengthens the predictive model by integrating physical features such as energetics R-loop [5].

Both Doench et. al and Zhang et. al were able to come up with a score for each sgRNA according to selected features. In practice, the software developed by both groups are able to give the best suggestions on what guide RNA to use on desired genes, locations, or CRISPR systems (cas9, cpf1, etc) [3][4][5].

**Methods**

*Data Preprocessing*

Gene-based sorting was done to rank the gRNA by its gene-ranking. The top 0.2 are labelled "top 20%"(positive) and the rest are labeled negative. This procedure is identical to what was conducted by Doench et. al, thus the label generation for the dataset is complete [2].

Feature extraction for each gRNA is derived from the Doench et. al papers [2][3]. The first and second set of features are taken from the 30-bp total region, with 10 bp prior and 10 bp after the 20-bp gRNA (excluding the PAM sequence). The first set consists of one-hot encoding vector, based on the type of base (A, T, G or C) on each location, totaling to 120 features (30 locations \* 4 possibilities) [2]. The second set of features consists of one-hot encoding vector based on the pair of bases for every location (AA, AT, AG, ..., CC) on each adjacent location, totaling to 464 features (29 locations \* 16 possibilities). The third set of features uses the G/C content of 20-bp gRNA excluding the PAM site. The first feature in the third set is the sum of the number of G and C base in the 20-bp gRNA. The second feature in the third set is whether the GC-count is more or less than 10 bases (above or below 50%). Thus, the third set of features yield 2 additional features in the list of features. The fourth set of the features include the melting temperature for various parts of the 30-bp site. This set of features uses calculates the melting temperature of all 30-bp, of base 7 - 12 (inclusive), of base 12 - 20 (inclusive), and of base 20 - 25 (inclusive), adding 4 additional features. These four sets of features total up to 590 features that are used in supervised machine learning prediction models. The codes for feature extraction can be found in supplementary material 2.

The dataset is then split to 80% training and 20% test, this is done using scikit-learn package in python.

Classification Approach (Supplementary Material 3, 5)

The objective of these methods is to rank gRNA based on their features, so the classification methods used are the ones that has hierarchal sense. Specifically, we focus on models that output the probability value of the gRNA being in the top quintile of genes we labeled earlier.

Some approaches are heuristic, and with different shuffling of samples, there will be differences in test accuracy, making them random variables. Therefore, the comparison method that will be used is the test accuracy for each approach using N = 10 samples and comparing the means using ANOVA and Tukey test if grouping is necessary (e.g. if of ANOVA is rejected).

*Classification Approach 1: SVM and Logistic Regression*

This procedure is replicated from the Doench et. al paper from 2014 [2] and will be the benchmark of the other methods. Since there is an exhaustive list of features, 590 to be precise, Doench et. al used linear kernel support vector machine with L-1 regularization [2]. Since L-1 regularization pushes feature weights to 0, it is possible to conduct feature selection using this method.

The features with weights 0 are then filtered out, leaving only a subset of the 590 features extracted. The filtered features are then used in a logistic regression classifier to predict the top 0.2 of each genes (labels). Scikit-learn was used for logistic regression without changing its default parameters (0.5 threhsold).

*Classification Approach 2: Logistic Regression*

Similar to the first machine learning method used, instead of filtering the 590 features, only logistic regression was used to predict top 0.2 gRNA of each genes.

*Classification Approach 3: Neural Network*

We use Pytorch package to build a 4-layer neural network that behaves similarly to logistic regression algorithms. Instead of having a 2-node in the output layer - as usually the number of nodes of output layer reflects the number of predicted classes, there is only one output layer that outputs a value from 0 to 1. We designed the neural network to output both the probability value of the guide RNA to be in the top quintile and the label (positive if above 0.5 threshold). The loss function used was cross entropy loss that we manually write due to the unconventional nature of the neural network design. In a sense, this neural network simulates a more complex logistic regression. We then trained the network for 850 epochs and compare the results with the other two previous methods.

Regression Approach (Supplementary Material 4)

In contrast to the classification approach, we also approached the problem by directly regressing the features with the gene rank of each gRNA. In contrast to the classification approach, this approach is novel, and was not used in previous works mentioned. Therefore, we will use the root mean squared error (RMSE) as the comparison method between methods.

*Regression Approach 1: Linear Regression*

We used linear regression package from scikit-learn to regress the features extracted directly to the gene rank. The predicted y value for linear regression can range from to as there is no transformation of the output value.

*Regression Approach 2: Single-layered Neural Network (1 layer)*

We designed a simple neural network that simulates logistic regression, that is a linear layer that takes in 590 features and outputs 1 value. Instead of having the output with the range to , this transformation will guarantee that the output will have the output values ranging from . We decided to try this approach because the gRNA uses % gene rank, with values between 0 and 1, so transforming the output to the same range is more appropriate. We use 1000 epochs to train this neural network.

*Regression Approach 3: Multi-layered Neural Network*

We designed a 3-layered neural network with transformed output similar to approach 2. However, we are expecting a lower RMSE value for this approach since it allows a more complex function instead of just a linear decision boundary.

**Experiments**

The dataset we used for this project is a real-world dataset from Doech et. al 2014 paper [2]. Specifically, we used supplementary table 7 that includes 1841 coding region (CDS) targeting sgRNAs. This dataset, albeit relatively small in size, is used due to its availability and readability. Supplementary table 7 of Doench et. al 2014 paper provides all information needed for both feature extraction and label generation, hence leading to the decision to use this dataset.

The other experiment we had is to determine the number of epochs to use on the neural networks, both for classification and regression. One common method to decide how many epochs to use is to look at the test loss. In the case of neural network for classification, we decided to use 850 epochs for train­­ing the neural network since the test loss plateaus around epoch 850 (Figure 1), and from the figure it looks that the model no longer learns and starts to overfit. In the case of neural networks, we used the root mean squared error (RMSE) of the test set instead. We decided that at around epoch 150, the test RMSE did not improve by much (Figure 2, 3).

**Results**

In the classification approach, we used ANOVA to compare test accuracy means (N = 10) between approaches and found there is enough evidence to conclude that is rejected and we adopted are equal. We continued the statistical analysis by conducting Tukey test, and found that approach 1 and 2 are in the same group, while approach 3, the neural network approach is in a different group(figure 4.). In the regression approach, we found that the test RMSE for linear regression approach is approximately 0.26 and the neural network approaches to be around 0.29.

**Discussion**

The regression approach seems to be not very effective in predicting gRNA ranks. 0.26 and 0.29 RMSE means the mean error gRNA prediction is 26% in the gene rank, which is not very useful when predicting the top 20% of the gRNA to be suggested. Therefore, we concluded that this approach was not very useful in predicting the effectiveness of gRNA. The potential analytical explanation of this observation could be addressed to the different numbers of gRNA for each gene. Thus, using gRNA directly in regression might not be wise.

According to the classification approach, we decided that there is a significant increase in quality of the classifier using approach 3, the neural network approach compared to the other approaches. Thus, we concluded that approach 3 that we developed will be more useful in predicting more effective gRNA. The resulting model will also be able to produce the scores that are similar to what the Doench et. al 2014 paper produced [2], since the neural network outputs a single value ranging from 0 – 1, similar to logistic regression.

**Future Works**

There could be improvements in several areas: dataset, neural network design, feature extraction. We believe that a dataset with more coverage (e.g. not just CDS regions, more genes, longer dataset) will be useful in strengthening the model. With a dataset with more coverage, the dataset becomes more similar to reality, thus the generalization power of the model could potentially increase. We also hypothesized that a more complex neural network design might also be able to contribute to higher accuracy of the prediction model, since there might be interactions between features that are not covered by a mere 4-layer neural network model. Another potential improvement to the model is to increase the number of features, such as the off-target and physical related features. These additional features could lead to more explanation towards the error or noise that contributes to inaccuracies of the model. We believe that all of these features could result in a better prediction model with a more generalized prediction, and ultimately higher accuracy in the real-world setting.

**References**

1. Jinek, Martin, et al. "A programmable dual-RNA–guided DNA endonuclease in adaptive bacterial immunity." Science 337.6096 (2012): 816-821.
2. Doench, John G., et al. "Rational design of highly active sgRNAs for CRISPR-Cas9–mediated gene inactivation." Nature biotechnology 32.12 (2014): 1262.
3. Doench, John G., et al. "Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9." Nature biotechnology 34.2 (2016): 184.
4. Labun, Kornel, et al. "CHOPCHOP v2: a web tool for the next generation of CRISPR genome engineering." Nucleic acids research 44.W1 (2016): W272-W276.
5. Zhang, Dong, et al. "Unified energetics analysis unravels SpCas9 cleavage activity for optimal gRNA design." Proceedings of the National Academy of Sciences 116.18 (2019): 8693-8698.