Important factors for CRISPR KO expression screening

Dennis Harding
Moa Ögren

June 5, 2025

Abstract

CRISPR knockout (KO) screening is a high-throughput gene screening method used to validate the knockout (inactivation) effect caused using the restriction enzyme Cas9. By analyzing different factors in a guide RNA sequence, like the GC content and nucleotide positions, one can predicted knockout effectivity. By Including these factors and filtering for active genes using RNA sequence data, we were able to build a machine learning model that predicted the efficiency of a sgRNA sequence. However, the results showed close to zero predictions rates, concluding the challenges raised with predicting CRISPR knockout expression efficiency. In the future of the projects, one may be able to increase the predictions rate by for example adding data describing the chromatin accessibility and exon positions.

1 Introduction

Since the discovery of the DNA molecule, the dream of gene editing and the endless possibilities that come with it has inspired scientists all around the world[4]. The discovery of the CRISPR-Cas9 system at Umeå University has opened the door to advancements in gene based cancer treatment[2], gene therapy[5], and other fields in bioengineering. The CRISPR/Cas9 system has outgrown its counterparts in popularity by a great distance[1]. However there are still challenges which question the reliability of the method such as off target effects, which is when nuclease cuts at sites different to the intended one, leading to adverse effects[3]. Providing researchers with more information will guide us in the design of sgRNAs, which is the component of CRISPR which guides the nuclease to its target. The purpose of this computational analysis was to gain more insight into KO efficiency by answering questions such as Which nucleotides at what positions have a greater impact? And can this be analyzed through a purely computational screen using publicly available data?

2 Theory used for data analysis

2.1 CRISPR

In general the CRISPR/cas system has two parts, the first component is the nuclease, which is responsible for cutting the DNA, The other part is the sgRNA that guides the nuclease to the complementary sequence on the DNA. At this point in time, the CRISPR/cas system has been modified in several different ways, so there are multiple different nucleases which can perform different operations on the DNA using its active site. The nuclease we are studying is the Cas9 variant which performs a double-strand break on the DNA. This gives the CRISPR virus infected organism no other choice but to perform Non-Homologous End Joining (NHEJ) which is an inactive gene or at the very least a dysfunctional gene product.

2.2 Datasets

The sgRNA data provided the values for the effects which each sgRNA on a gene. This is the backbone of the analysis and the choice of this dataset and the filtering of it will have a great affect on the outcume of the analysis. The other two datasets are reference datasets which provide complementary information such as nucleotide sequence and RNA expression levels from Gecko library and RNA -seq respectively. The pairwise common column in each dataset allow for efficient selection of data in the database using the join function. This allows for efficient filtering and outlier identification. For exaple we can select a RNA seq cutoff which will determine which data points are used in the feature impact analysis. The following datasets were found on the web and chosen with assistance of the course leader.

2.2.1 sgRNA data

The sgRNA dataset[sgRNAdata] used provided:

- Gene name - sgRNA target gene
- Gecko id - id which can be connected to a nucleotide sequence
- Log Fold Change (LFC) - change in amount of functional genes

2.2.2 Gecko library

Gecko library A[libraryA] and B[libraryB] was merged into one database table, which provided:

- Gecko id
- sequence

2.2.3 RNA-seq

human RNA-seq[RNAseq] dataset provided:

- Gene name - sgRNA target gene
- FKPM-counted - a balanced RNA expression count

2.3 Features affecting LFC

We chose LFC as the dependent variable, and the nucleotide sequence and GC-content as explaining variables. However, there are more factors than these two which will in the end determine the LFC. The LFC is calculated as follows:

$$\text{LFC} = \log_2\left(\frac{\text{Dropout\_count} + \epsilon}{\text{Control\_count} + \epsilon}\right)$$

The count is determined by the comparing the amount of sgRNA in infected cells compared to a control. The DNA in the sample of cells is synthesized and and the reads which can be mapped to the

sgRNA is counted and a count is calculated using the MAGeCK tool. If the count is greater in the dropout than in the control it would indicate that the the knockout results in an increased proliferation of the cells. This means that the result in the measurement of out dependent variable is completely dependent on the function of the targeted gene. If it is a gene which is needed for an essential process the knockout is expected to result in low LFC, if the target gene is a highly situational gene there might be no effect of the knockout at all. Attempting to model the efficiency of a sgRNA sequence based on such a highly inconsistent dependent variable appears like a near impossible challenge, however we will use RNA-seq data in an attempt to narrow down the dataset as a way to reduce noise. The data was thus filtered by excluding instances where the RNA expression is low, with the reasoning that genes that are expressed less, are in general more situational and specific.

2.4 Machine learning

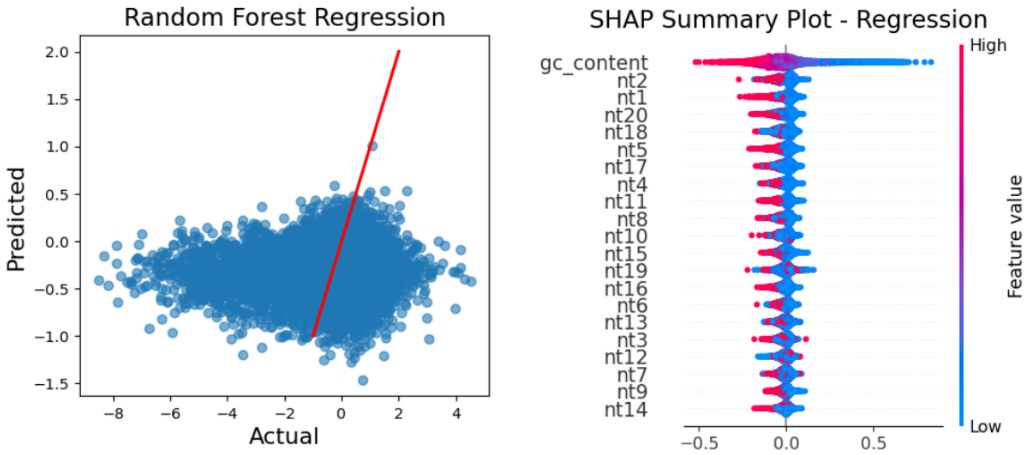
Random forest is a supervised machine learning algorithm which build a large number of decision trees; as in the tree data structure. Each tree is allocated its unique subset of the training data will split this subset in terms of different features such as having a specific nucleotide at a specific location, and then based on the input give a prediction as the output. In the regression application the final result will be the average value which all of the "trees" in the "forest" predicts and in classification output is determined by majority vote.

In the assignment we performed both a classification and a regression model. In the regression model the output was a scalar value for the LFC and in the classification model the output was a binary output where 0 indicated an absolute value of below 0.6 and 1 indicated above. The reason we did both was to gain different perspectives on the dataset, hoping to find more information. The main reason that the random forest algorithm was chosen was that it was recommended by the course leader, however the reasons for this is likely because it is simple to implement, it is resistant to noise and outliers and it can handle missing data, which can often occur in larger datasets. The negatives of this method is that it is slow in prediction since the prediction data needs to pass through all of the trees, this could be compared with linear regression which can directly map the input to an output through a mathematical function.[10]

We considered making a Recurrent Neural Network (RNN) as well, however, due to compatibility issues, time restraints and discouraging results from the random forest models we decided not to. A RNN would have given us the opportunity to experiment with a different hypothesis space and fine tuning the complexity of the model and testing out LSTM layers. Since RNNs are known for its strengths with sequential data it could potentially have been the best option for our application. If more time and also computational power would have been provided this would have been a solid alternative.[11]

2.5 Results

The results for the classification and regression random forest models are presented below. The main theme in the plots is that the model was not able to fit the data. The regression model has an R-squared at -0.015. Which means that it is a negative correlation and that 1.5 percent of the variation in the data is explained by the model. In the SHAP plot we can see that GC-content was the best predictor and that lower GC-content was correlated with a higher LFC value. The nt features in the SHAP plots are problematic since we do not know which of the nucleotides are set to high and what the other three are which are set to low.



(a) Predicted vs actual random forest model.

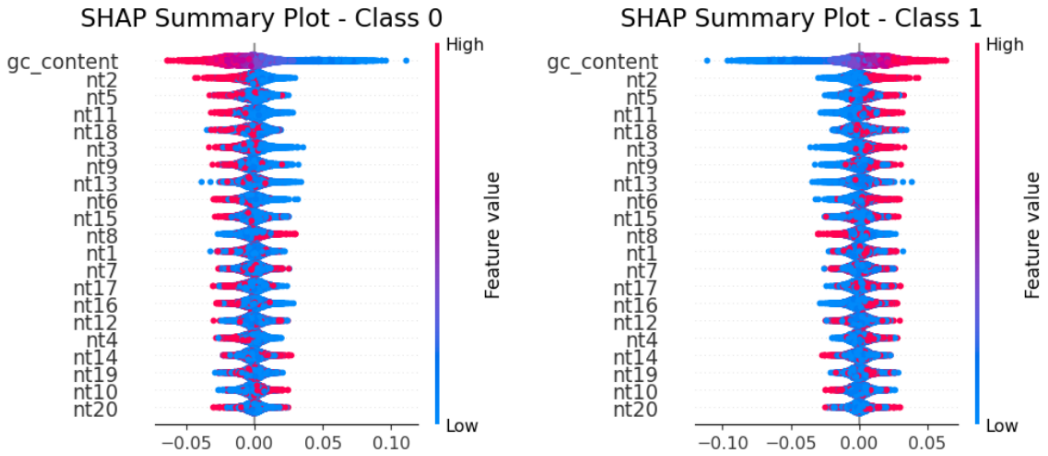
(b) Shap feature influence summary prot.

Figure 1: Regression plots.

The Classification results tell a similar story as the regression, The results for the classification is about the same. The following is the confusion matrix output:

		Predicted	
		Positive	Negative
Actual	Positive	5821	3682
	Negative	3953	5498

Table 1: Confusion matrix



(a) Class 0 summary

(b) Class 1 summary

Figure 2: SHAP feature influence summary plots

3 Implementation

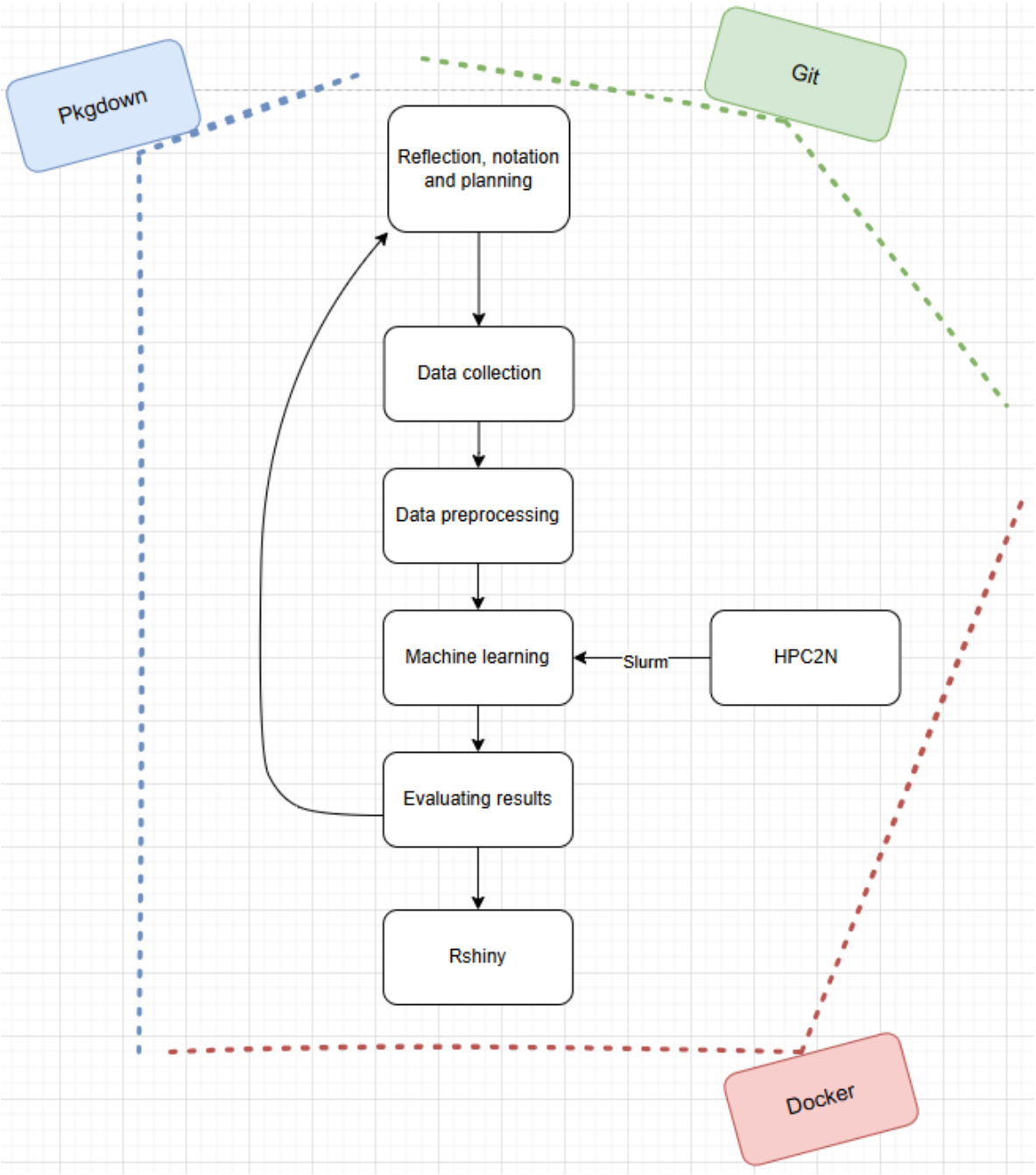


Figure 3: Enter Caption

Throughout the analysis process, various tools and technologies were integrated into the workflow, including Git, Docker, SLURM, RShiny, Pkgdown.

3.1 Git

Git is a widely used version control system widely that enables developers to easily modify and track changes in projects. Similarly to the version control features seen in Microsoft Word, Git allows user to modify, updated, and revert old code. Although in contrast to word, Git also allows multiple users to make changes at the same time at the same place without conflicts. The conflicts are later solved through a pull request and a merge. This gives developers the freedom of working simultaneously and thus more effective. Git goes under the Generarl Public Licesne (GPL) and is thus free of charge. The project code can easily be uploaded and viewed by others using the website Github which is also free of charge. Git requires command-line interfaces and is often used in Linux-based environment which can make the learning curve steeper compared to other version controls systems [8]. Although Git has solved this complication by launching Github desktop, and Git is also integrated into various integrated development environments like VS Code and RStudio.

3.2 Pkgdown

Pkgdown is an R package which automates the creation of a web application for standardized and structured package documentation. It makes sharing of packages easier and allows users to access the package information and functions through a simple URL link.

During the implementation we encountered some issues due to that we first created a git repository which r considered a package. Then when trying running `pkgdown::build site()` in the console there were som infinite loops. so we decided to make a new folder, build the site there, turn it into a repository and merge that repository with the older one. This gave us some experience with git which we otherwise would not have which was good, but it took more time than might have been intended.

Overall we consider pkgdown to be very useful in any application where work is intended to be shared with other people. With more experience i imagine that the creation and usage of it will work seemlessley in most cases.

3.3 Database

A database is an effective and organized way of storing data. A developer can easily construct an ER-diagram which gives an overview of the keys and constraints found in the database. A database also saves the data more efficiently, and requires less space which speeds up the working process. One can designed a database using different data models, such as XML-based models or relational databases. The first is a hierarchal, tree-like structure, and the second follows a more organized structure using tables with columns and rows which can have specific constraints like `PRIMARY KEY` and `UNIQUE` [6]. In this project a SQLite database was used which is designed by a relational algebra base. Saving the data into a database made working with the high-performance computer HPC2N, increasingly more efficient since the data could easily be uploaded to the computer without running excessive code.

3.4 SLURM

SLURM is a workload manager helping scheduling jobs in an effective queue system. By using SLURM multiple participants can work on the same high performance computer system efficiently without conflicts. The participant send in their job in the queue system and can there on continue working on there project while the code is running. In this project, the HPC2N computer was used for the heavy computational work of calculating the SHAP-values for the machine learning models.

3.5 Rshiny

Shiny is an open source R package available through CRAN that allows developers to easily construct an intercative web app without using languages such as JavaScript, HTML or CSS. Instead the only language is used is R. Shiny is since 2022 also available through Python [9]. A website was created using the Rshiny library where Figures, interactive plots are displayable and a for fun password protected administrative access tab. Sample screenshots with description can be seen below.

3.6 Docker

Docker is a tool for containerizing applications allowing users to run the program on their local machine. The software is delivered to the user through a container which is configured by a Docker image. The Docker image defines the application's environment and dependencies. By using Docker one ensures that everything needed to run the application is included and packed together, allowing participants which different computer environemnt and versions to run the application. Docker runs under the Apache License 2.0 which is free of charge [7]. In this project the Docker image can be build using the dockerfile included in the project setup.

4 Conclusion

This project concludes the difficulties with predicting the CRISPR KO expression effeiciency. By adding data like CHIP-seq data or exon positions, one could potentially improve the predictions. However, more research is needed to understand what makes a sgRNA sequence efficient before further work can be done.

References

[1] Nina Duensing, Thorben Sprink, Wayne A. Parrott, Jeffrey D. Wolt, and Detlef Bartsch. Novel features and considerations for era and regulation of crops produced by genome editing. *Frontiers in Bioengineering and Biotechnology*, 6:79, 2018. Accessed: 2025-06-05.

[2] National Cancer Institute. Crispr in cancer research and treatment. https://www.cancer.gov/news-events/cancer-currents-blog/2020/crispr-cancer-research-treatment, 2020. Accessed: 2025-06-05.

[3] Huan Li, Yuxuan Yang, Wei Hong, Min Huang, Ming Wu, and Xue Zhao. Applications and challenges of crispr-cas gene-editing to disease treatment in clinics. *Cell & Bioscience*, 11(1):1–11, 2021. Accessed: 2025-06-05.

[4] The Nobel Prize. The nobel prize in chemistry 2020 - popular information. https://www.nobelprize.org/prizes/chemistry/2020/popular-information/, 2020. Accessed: 2025-06-05.

[5] Fahad Uddin, Charles M. Rudin, and Triparna Sen. Crispr gene therapy: Applications, limitations, and implications for the future. *Frontiers in Oncology*, 10:1387, 2020. Accessed: 2025-06-05.

[6] Wikipedia contributors. Database. https://en.wikipedia.org/wiki/Database, 2025. Accessed: 2025-06-05.

[7] Wikipedia contributors. Docker (software). https://en.wikipedia.org/wiki/Docker_(software), 2025. Accessed: 2025-06-05.

[8] Wikipedia contributors. Git. https://en.wikipedia.org/wiki/Git, 2025. Accessed: 2025-06-05.

[9] Wikipedia contributors. Shiny (web framework). https://en.wikipedia.org/wiki/Shiny_(web_framework), 2025. Accessed: 2025-06-05.