Predicting CRISPR Gene-Editing Efficiency with Deep Learning

Max O'Meara
mxomeara@bu.edu

Rehan Samaratunga
rdsam@bu.edu

1 Task

We want to build a CNN that can tell ahead of time whether a tiny "address tag" (RNA string) for genes will work well. In the lab, these tags are short strings that help turn off a chosen part of DNA. Scientists have already tried many tags in cells and measured how strong their effect was. Therefore the goal for the model is to learn these relationships automatically from experimental examples, so that it can make reliable predictions for new RNA sequences it has never seen before. This choice matters because it pushes the model to generalize to genes it hasn't seen before and it must rely on sequence based rules that transfer to other genes rather than relying on properties of specific genes.

In practice, the CNN will learn simple position-specific patterns near the PAM and along the guide (e.g., seed region signals and GC balance) that track with effect strength, producing a calibrated score that ranks candidates across different cell lines and readouts without using gene-level features.

2 Input and Output

User input to our CNN model will be an example CRISPR-Cas9 screen without gene level features. This forces the model to learn universal guide RNA design principles for new genes and RNA sequences it hasn't seen before. The user will input a 23 letter sequence consisting of A/C/G/T, the cell line model, the phenotype, chromosome and the strand.

Concretely, inputs are (seq23, celline, phenotype, chr, strand). seq23 is 23 letters as it includes the guide RNA of 20 characters with the 3 PAM characters which act as a binding and recognition signal for the Cas. celline encodes the experimental context and lets the model adjust baselines across cell types while still relying on sequence rules. phenotype tells the model what outcome is being measured. chr adds minimal genomic context since we exclude coordinates but allows it to contribute to the output without leaking gene specific information. strand indicates the target orientation and if strand is "$-$", we reverse-complement so the CNN always sees the PAM at positions 21–23.

An example input can look like: $GCAGCATCCCAACCAGGTGGAGG$, Jiyoye, viability, 10, $+$. The model will take the input and will output a single number showing the expected strength of the effect using the score: 0 = weak to 1 = strong. An example output can look like: 0.31590732393855947

We one-hot encode the sequence and include small learned vectors for celline, phenotype, chr, and strand; no gene-level inputs are used. Basic checks enforce length =23 and A/C/G/T only. The output is a single value in [0,1] (higher = stronger expected effect), obtained with a sigmoid and lightly calibrated so scores are comparable within the same experiment type.

3 Purpose

The purpose of this project is an attempt to save resources in labs. CRISPR-Cas9 a gene-editing tool that can treat genetic diseases, fight cancer, and combat infectious diseases. When biotech companies are designing new therapies, they're targeting genes they've never screened before. Instead of testing dozens of DNA tags to find one that works, researchers can start with the most promising options our model suggests. That means fewer trial and error experiments, faster progress on basic biology studies, and a smoother path for teams using this tool for disease research, drug studies, or classroom labs.

This reduces library size, sequencing, and hands-on work, freeing time for confirmatory tests and follow-up biology. It also gives a clear, repeatable ranking method that helps small labs and teach-

ing settings. Because the setup is configurable by cell line and phenotype, teams can run quick design–test–update cycles to steadily improve guide selection.