Columbia University 2018, 05–11

Application Note

|  |
| --- |
| Sequence Analysis  CrispRNN: Predicting CRISPR sgRNA effect  Natnicha Vanitchanant1,\* and Olga Lyudovyk1,\*  1Department of Biomedical Informatics, Columbia University, New York, 10032, United States.  \*To whom correspondence should be addressed.  Abstract  **Motivation:** CRISPR guide RNA sequences differ in their ability to allow the CRISPR-Cas9 complex to effectively edit a targeted gene. The on-target effect of CRISPR guide sequences has been experimentally measured, catalogued, and analyzed using regression-based techniques. Sequence specific features such as identities and positions of single nucleotides and dinucleotides were shown to be the best predictors of the effect. CrispRNN attempts to extract more complex sequence-based features automatically and learn to predict CRISPR guide sequence effect using a bidirectional recurrent neural network (RNN) with long short-term memory units (LSTM) built with TensorFlow.  **Results:** CrispRNN is able to predict the effect of sgRNA sequences based on the sequences themselves, without any additional features. The accuracy is poorer but comparable to linear regression-based models. Incorporation of additional information is expected to improve performance.  **Availability:** Source code is available on GitHub: https://github.com/numfah23/CrispRNN  **Contact:** ol2205@cumc.columbia.edu |

# Introduction

CRISPR-Cas9 is a molecular complex derived from a microbial adaptive immune defense system and designed to produce double-strand DNA breaks. It is a widely adopted genome editing tool in biomedical research for disease modeling, genetic screens, and therapeutic explorations. CRISPR-Cas9 complex is able to target a particular gene due to a single guide RNA (sgRNA) sequence that forms part of it and preferentially binds to the complementary sequence in the DNA1,2. The effectiveness of CRISPR-Cas9 to edit genes, or the on-target effect, depends on features of the guide sequence2,3,4. Precision with which only the targeted sites in the genome are affected, or the off-target effect, depends on the specificity of sgRNA. CrispRNN focuses on predicting the on-target effect of sgRNA sequences.

Applying modeling approaches such as linear regression, SVM, and Gradient-boosted regression tree, Doench et al. identified features that predict the on-target effect of sgRNAs.3 Single and dinucleotide identities in specific positions in the sequence contributed the most to the activity prediction (58% measured by Gini importance), followed by position-independent counts of single and dinucleotides (16%), and location within protein and melting temperature3. The accuracy of Doench et al.’s best model with the full set of manually crafted features is reported at .51 as measured by Spearman correlation of predicted versus actual effect. Kim et al. studied a related gene editing system, CRISPR-Cpf1, and demonstrated that a deep learning approach (CNN) using sequence information alone can predict the on-target effect with a .75 accuracy (Spearman correlation)4. The CNN-based results further improved when chromatin accessibility was incorporated as a feature.

We present CrisprRNN, a bidirectional Recurrent Neural Network (RNN) with long short-term memory units (LSTM), to predict on-target effect of sgRNAs in CRISPR-Cas9 using sequence information alone. The code to pre-process data, train the RNN, and predict the effect is available on GitHub (see Availability). In addition, a model trained on 80% of sequences (58,385) from Meyers et al. is available and can be used to generate predictions, avoiding the need for lengthy model training5. While the accuracy of the current model is low (Spearman correlation .42), it is comparable to the accuracy achieved by Doench et al. and spares the need for manual feature generation. Additionally, performance of CrisprRNN is expected to improve when trained on datasets with more sequences generated by various experimental designs and trained for more iterations.

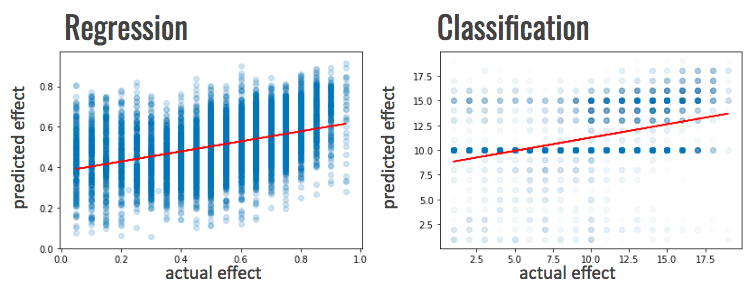
# Methods

## 2.1. Bidirectional RNN for target effect prediction

CrisprRNN implements in Python a bidirectional Recurrent Neural Network with long short-term memory units using TensorFlow. It is provided as a Jupyter Notebook and allows the user to specify the input files (training, validation, testing) and the model hyperparameters directly in the Jupyter Notebook.

## 2.2. Inputs and Outputs

CrisprRNN operates on 23-nucleotide-long RNA sequences, where each nucleotide is one-hot-encoded in a 4-element vector. The data pre-processing script available on GitHub reads as input a csv file (filename can be specified in Python notebook), which includes at minimum columns labeled “sequence” and “effect”. It then one-hot encodes a nucleotide at each position; processes the “effect” depending on the mode specified; and splits the dataset into training, validation, and testing sets according to the indicated proportions. For Regression, the effect is normalized to continuous values on a scale of 0 to 1; for Classification, it is recoded into 21 class labels. The output of the data pre-processing is a set of files in the pickle format that serve as input to CrispRNN.

The initial data used for training and testing the model are sequences contributed to the GenomeCRISPR database6 by Meyers et al5. The effect value was standardized as part of submission to GenomeCRISPR. CrisprRNN expects the effect in input files in the same format, as a value between -10 and 10.

The [full dataset](http://www.dkfz.de/signaling/crispr-downloads/GENOMECRISPR/) is available at [GenomeCRISPR](http://genomecrispr.dkfz.de/)6. The data from Meyers et al. can be retrieved by filtering the full database for PubMed ID 29083409. A second file was used for secondary testing (see Results). Two smaller sample input files available with CrisprRNN can be pre-processed for further analysis. Note that training the model on a small sample file may result in overfitting. For convenience, a model trained on the large dataset is made available.

## 2.2. CrispRNN design: Regression, Classification modes

Fig. 2. CrispRNN predictions evaluated on a test set.

CrispRNN is implemented as a bidirectional RNN with 23 timesteps, equal to the sequence length. Activation function, learning rate, number of iterations, number of hidden layers, and frequency of outputting intermediary results are set by default to optimum values for our dataset, but are configurable in the Jupyter notebook.

The model can be run in Regression or Classification mode. In Regression, it uses *mean squared error* as the loss function; the *sigmoid* activation function in the last layer outputs a value between 0 and 1. In Classification mode, its loss function is *softmax cross entropy with logits,* and the last layer’s *softmax* activation function outputs a label between 1 and 21. In both modes, the output is interpreted such that the higher number indicates the higher predicted effect.

## 2.3. Training the model and predicting effect

Running all sections of the Jupyter Notebook sequentially will result in training the model, predicting on a validation set and test set, and visualizing and saving results for the predictions and training loss. Depending on the number of iterations specified in the *training\_step* parameter and the infrastructure setup, training can be a lengthy process. If desired, training and validation can be skipped, and a saved pretrained model can be loaded and used for predictions directly. In order to do so, execute the section of Jupyter Notebook marked with “*predictions and accuracy for testing set*” with suggested specified model names.

## 2.4. Dependencies

CrispRNN is written in Python 2 and requires the following packages: *tensorflow*, *jupyter*, *numpy*, *pandas*, *pickle*, *matplotlib*, and *scipy*.

# Results

Predictions of CrispRNN were evaluated using Spearman Correlation by comparing predicted labels with actual labels for the test dataset. A second evaluation was performed on a subset of sequences contributed to GenomeCRISPR by Doench et al.3, preprocessed following the same process. After training for 1,000,000 iterations with parameters listed in Supplementary Table 1, Regression and Classification models had similar performance: Spearman correlation of 0.4269 (p-value rounding to 0) and 0.4154 (p-value = 3.671e-304). Results are plotted in Figure 2 below.

# Future work

We propose to improve the performance of CrispRNN by incorporating additional features that cannot be derived from the RNA sequence itself. Incorporation of these features is envisioned as an integration layer that combines output from the bidirectional RNN layers with the additional features (Supplementary Figure 1). The following features were found to be significant in prior studies and can be added to the current model:

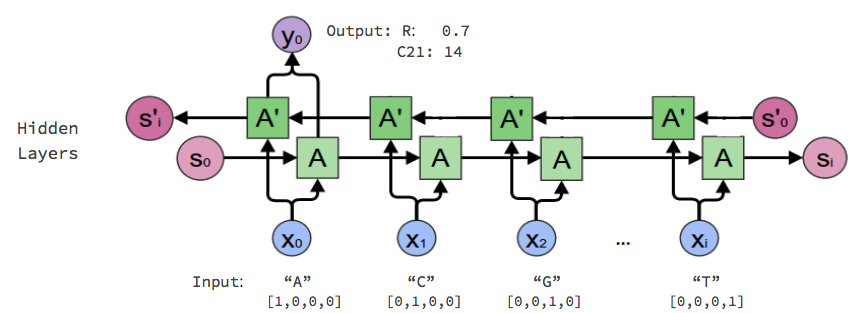
Melting temperature3

Fig. 1. CrispRNN schematic design (Adapted from Lee 20177)

Position within the gene: in exon, in a protein family domain3

Chromatin structure4

Additionally, the current model was trained only on effects measured under one experimental design. Training the model on the entire CRISPR database, with data contributed by 20 studies with different experimental setups, would enable it to learn generalizable sequence features and thus enable predictions to generalize better.

Finally, to elucidate what features the model learns during the training process to be predictive of the effect, we propose to add an attention layer in future implementations.

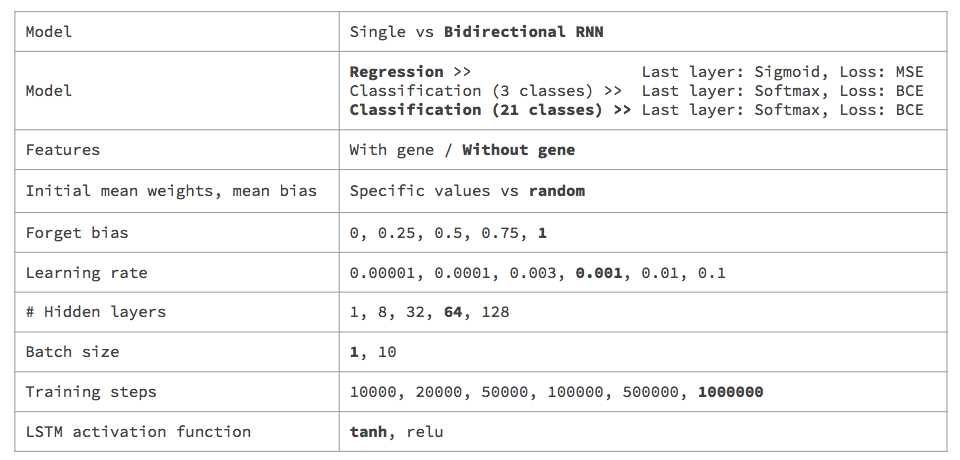
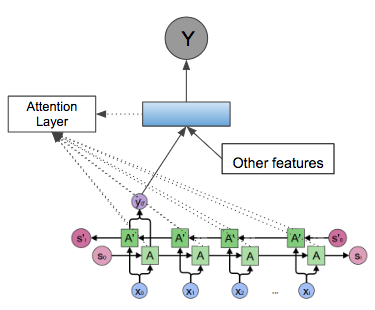
Acknowledgements

We thank Professor Pe’er and Tyler Joseph for their support and access to Google Cloud where development of CrispRNN was possible.

References

1. Lu, X.-J. et al. (2017) The applications and advances of CRISPR-Cas9 in medical research. *Briefings in Functional Genomics.* 16(1):1–3.
2. Doench, J.G. et al. (2014) Rational design of highly active sgRNAs for CRISPR-Cas9–mediated gene inactivation. *Nature Biotechnol*. 32:1262–1267
3. Doench,J.G. et al. (2016) Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nature Biotechnol.* 34:184–191
4. Kim, H.K. et al. (2018) Deep learning improves prediction of CRISPR–Cpf1 guide RNA activity. *Nature Biotechnol.* 36:239-241
5. Meyers, R.M. et al. (2017) Computational correction of copy number effect improves specificity of CRISPR-Cas9 essentiality screens in cancer cells. *Nature Genetics*. 12:1779-1784
6. Rauscher, B. et al. (2016) GenomeCRISPR - a database for high-throughput CRISPR/Cas9 screens. *Nucleic Acids Research.* 2017 Jan 4; 45
7. Lee, C. (2017) Understanding Bidirectional RNN in PyTorch. *Towards Data Science*. 2017 Nov 3. [Link](https://towardsdatascience.com/understanding-bidirectional-rnn-in-pytorch-5bd25a5dd66)

Supplementary Materials.



Supplementary Fig 1. Proposed design of CrispRNN 2.0.

Supplementary Table 1. Model parameters tested with CrispRNN. Values in Bold indicate the best performing parameters and are set as default.