A deep learning framework combined with word embedding to identify Crispr Cas9/Cas12a gRNA efficiency

Ki-wook Lee[a,b]

[a] *Correspondence to : leek0502@gmail.com*
[b] *Department of Digital Health, SAIHST, Sungkyunkwan University, Seoul 06351, South Korea*

Abstract

<>

Keywords: Cripsr, Deep learning, Representation learning

Introduction

Clustered Regularly Interspaced Short Palindromic Repeat(CRISPR) and CRISPR Associated Protein(CAS) systems refer to a combination of enzyme and DNA sequence used to edit genomes. Cas9 or Cas12 are well known systems for their standard performance. Protein of such a system is guided by guide RNA to its complementary sequence flanking on PAM(Protospacer adjacent motif) sequence (ex. 5'-NGG-3') and breaking the DNA double strand results in editing the genome.

Previous studies show that genome editing efficiency depends on guide RNA sequence and composition of flanking sequences of target sequence, however, the determinative point of editing efficiency is not revealed or studied enough yet. Therefore, predicting more accurately genome editing efficiency relies on uncovering the mechanism and developing new algorithms or models to predict of the system such as Cas9 and Cas12a.

There are several tools based on deep learning methods to predict genome editing efficiency of Cas9 and Cas12 such as DeepSpCas9, DeepCpf1, DeepHF and CRISPRon which help to select appropriate guide RNA sequences. DeepHF is a tool for Cas9 system which takes 21 base pair DNA sequences as input and predicts efficiency of the sequence. It represents input guide RNA sequence as embedding vector and is learned by BiLSTM. Along with learned information from guide RNA sequence, it combines biological information such as position accessibilities and stem-loop of secondary structure, melting temperature and GC content resulting in accurate genome editing efficiency.

CRISPRon, using 3 different size kernels of convolution layers, encodes 30 base pair length DNA sequences by one-hot. Each output of convolution layers of different kernel sizes combines together as features, the input of a fully connected layer to predict guide RNA efficiency. DeepCpf1, developed for Cas12a efficiency prediction, takes 34 base pair sequences as input. And if there is accessible chromatin information, using it for more accurate prediction. It also uses one-hot encoded vectors and convolution layers for extracting features.

Although previous tools found their own methods of predicting efficiency by extracting informative features from DNA sequences, several limitations still remain. (1) using one-hot encoding to extract sequence features leads to overfitting for high dimensional models. (2) Embedding methods such as those used in DeepHF extract the distributed representation of the input guide RNA sequence only for the data used for training, so the model tends to depend on the training data. (3) one-hot encoding method itself; that is not an appropriate method when it calls to calculate similarity between sequence to sequence. For instance, ATCCG is similar to ATCC not GAGCA but in one-hot encoding method, the distances of each are the same.Although previous tools found their own methods of predicting efficiency by extracting informative features from DNA sequences, several limitations still remain. (1) using one-hot encoding to extract sequence features leads to overfitting for high dimensional models. (2) Embedding methods such as those used in DeepHF extract the distributed representation of the input guide RNA sequence only for the data used for training, so the model tends to depend on the training data. (3) one-hot encoding method itself; that is not an appropriate method when it calls to calculate similarity between sequence to sequence. For instance, ATCCG is similar to ATCC not GAGCA but in one-hot encoding method, the distances of each are the same.

To overcome these limitations, we propose DACO (Deep learning Architecture for Crispr cas9/cas12a On-target efficiency prediction tool). DACO is made of several tricks to find a way of shaking off limitations of previous ways. (1) More appropriate distributed representation of DNA sequence makes the model depend less on training data which is the by-product vector of word2vec model, a deep learning learner for whole human genome sequence. (2) sequences are grouping into k-mer for each base, and represented as a 100 dimensional vector. This makes the model learn deeper than a one-hot encoding method. (3) Multi scale feature extraction helps the model predict more robust features.

Various training and validation is made for evaluating performance of DACO that is not only using large scale dataset such as Wang, Kim,

Xiang databases, but also small scale dataset such as Wang hl60, hart hct116, and hela cell line data. As the result of evaluation, DACO shows higher performance than tools used widely. We strongly believe that this proposed method and architecture would contribute to predict guide RNA editing efficiency and even understanding mechanisms of CRISPR proteins.

Results

Robustness with different k-mer size and representation sequence methods are validated. Efficiency prediction results among separate datasets(DeepCas9 and CriprOn for Cas9, DeepCpf1 for Cas12a) are compared.

Parameter optimization of k-mers

Whether different sizes of segment (k) affect prediction performance of the model, 3 dataset from Wang(wild type, HF1, and esp), 1 dataset from Kim, and 1 dataset from Xiang were used to measure correlation between distributed vector representation and prediction model.

In all datasets, with increasing segment size from 3 to 5, model performance is increasing. and with increasing size from 6, performance drops. This tendency is the same with previous work which shows a correlation between segmentation size and prediction model performance. This is due to the increased trainable vocabulary with larger size of segments which makes dna2vec represent DNA sequence more accurately. and that is followed with higher performance of the model. On the other hand, with too large sizes of segments, distributed representation is getting much more complex and that makes it overfitting because of the limited number of data points, followed by lower performance of the model. Therefore, k-mer size 5 is decided to make an appropriate model. Otherwise, models trained with Xiang, Cas9 Kim and Cas12 Kim make no correlation between size of segments and model performance.

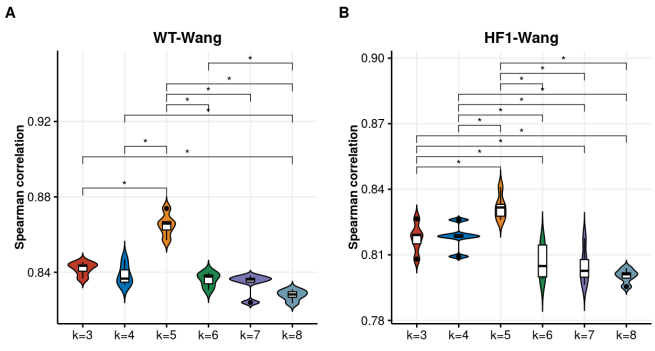

Figure 3. Comparison of Spearman correlation results according to k-mer size change

Distributed Representation Significantly Improves the Prediction Performance

To make sure there are any advantages in using distributed representation with word2vec rather than using one hot encoded vector, 5 datasets for Cas9 and 1 dataset for Cas12a are used to compare the two models. The segment size of word2vec for DACO was decided as k=5 by experiments before. There are statistical significance on models with some datasets (Wang wild type Cas9, Xiang, Kim 12). Specially, 3 datasets besides Xiang show large benefits at using word2vec representation.

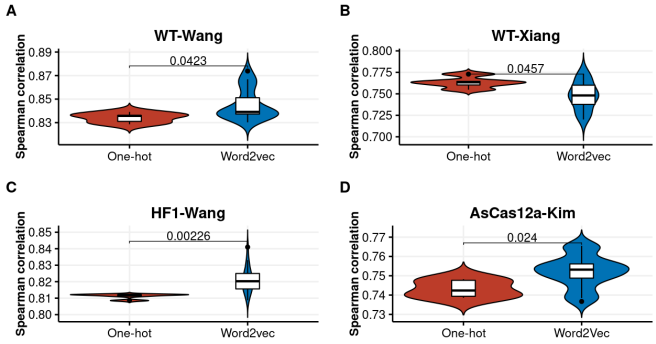

Figure 4. Comparison of Spearman correlation results according to DNA representation method

Comparison of DACO and other gRNA efficiency predictors

To validate performance of DACO, previous tools were used to compare on spearman correlation measurement. Using recent source code provided by arthurs, models are builded for each 6 datasets by DACO. DACO shows significantly better prediction performance than previous tools. and the differences are much clear on Cas12a than Cas9. These differences also show in comparison with the DeepHF model which uses embedding layers rather than one-hot encoding that is used by other tools like DeepSpCas9, CrisprOn, DeepCpf1. This is due to there being a small amount of datasets for learning embedding layers using only training data which makes representation bias. On the other hand, DACO uses huge data from whole human genome sequences to make pre-trained word embedding vectors which are less biased, and represent well for the meaning of guide RNA sequences. and that followed by better performance of the model for sure.

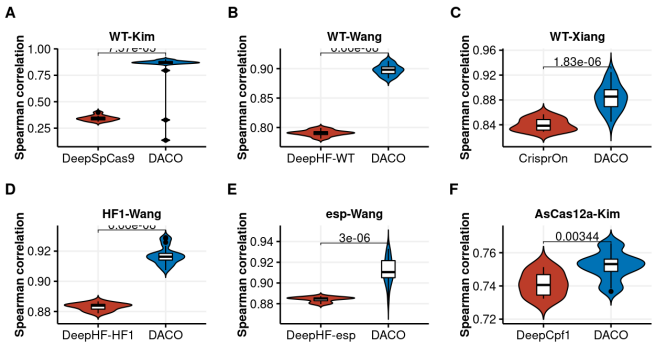

Figure 5. Comparison of DACO and the competing methods

Discussion

In this study, a new framework based on deep learning methods to predict Cas9/Cas12a genome editing efficiency is proposed. We hypothesize that there is a small number of data to train representation of guide RNA sequence via embedding layer. and also, one-hot encoding is not a good way to represent DNA sequences. To prove that, pre-trained word embedding vectors using human genome sequence are used to represent DNA sequences, and validating with one-hot method models and embedding layer models show remarkable superior performance of independent embedding techniques than others.

Methods

In this section, Benchmarks on independent test sets, Embedding methods, and details of model and hyperparameters will be described.

Data Source

We used 9 separated public datasets for training, tuning parameters, and discursive validation. For Cas9 efficiency, HEK293T cell line data which are SpCas9, eSpCas9, and SpCas9-HF1 Crispr/Cas system results from Wang et al. are used. And the same cell line data from Xiang et al which is SpCas9 results, 10 days after transduction is used. For Cas12 efficiency, Kim et al data is used which is AsCas12a results from HEK293T cell line.

Efficiency of guide RNA is defined as ratio of insertion and deletion reads counts per total reads from genome scale screening after treated with CRISPR proteins. In addition, public package data from Chuai et al is used for validation. In this package, Hela and HCT116 cell line data from hart et al and hl60 cell line data from Wang is composed. Each experiment is made of guide RNA sequences and measured knockout efficiency. For all 9 public datasets, duplicate guide RNA is removed. Only one-to-one matched guide RNA with ENCODE transcript version 103 was used for prevention of distortion in efficiency resulting from off target effect. Each dataset is divided by 75:15:15 ratio for training, validation and test sets.

Table 1. Data source

Crispr/Cas	Cell line	Num.	Note
SpCas9	HEK293T	46,405	Wang et al., [6]
eSpCas9	HEK293T	48,944	Wang et al., [6]
SpCas9-HF1	HEK293T	47,486	Wang et al., [6]
SpCas9	HEK293T	9,895	Kim et al., [4]
SpCas9	HEK293T	11,008	Xiang et al., [7]
AsCas12a	HEK293T	17,214	Kim et al., [5]

Sequence Representation

Target sequence added flanking 5bp sequence for each 3 and 5 prime sides of guide RNA sequence was used for input sequence. 33 and 34 base pair sequences including PAM sequence for Cas9 system and Cas12a system respectively were used as input sequence for model. input sequences were segmented by the k-mer method. the k-mer method is used to divide a sequence into a series of k-mers. For instance, if 33 base pair length DNA sequence (l=33) is segmented by 3-mer (k=3) methods with 1 length stride size (s=1), there would be 31 (l - k + 1) segments produced. and each segment has length 3 (k) and overlap with other segments would be 2 (k - stride).

To validate k-mer segmentation methods, segment size (k=3,4,5,6,7,8) and stride size (s=1) were tested by different values. Each segment is represented as l - k + 1 X 100 dimensional distributed representation vector which is the result of the dna2vec model with hg38 genome sequence.

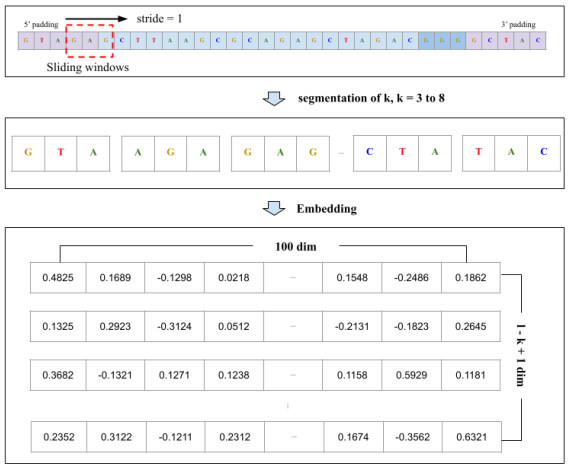

Figure 1. Representation of grna sequence

Model Architecture

DACO proposes a workflow like figure 2. Input sequences convert to 2 dimensional distributed representation matrix by word2vec model, and are passed to traditional convolution layer with kernel size 7, and to multi-scale feature extraction with kernel size of 3, 5 and 7 respectively. feature extracted results passed to pooling layer through fully connected layer to predict guide RNA efficiency. Details of each module follows.

Generally, the more layers composed of a convolutional neural network, the higher level abstracted features would be extracted. But, there are some bad side effects such as vanishing gradient and/or overfitting with huge size of parameters to connect each layer. To overcome these limitations, a proposed method is to use different scale kernels in parallel to make different features and combine that together. Multi-scale feature extraction is a method to extract features by different size kernel filters which make different features by kernel sizes enable various featured information. and Using these features known to prevent information loss which could be occured at abstracting input data and to make the model much robust to be show better performances with shallower network model. In DACO, different kernels sizes are setted by 3, 5 and 7 and features from convolution layers are pass to batch normalization, Relu, Dropout, and pooling layers.

Skip connection is a method to mediate vanishing gradient problems with a number of layers. To fix updating weights to the wrong direction with a bunch of layers, only residual values between output of previous trained model output and that of additional layers are trained. This makes calculation much simpler, and makes training easy in terms of size of error.

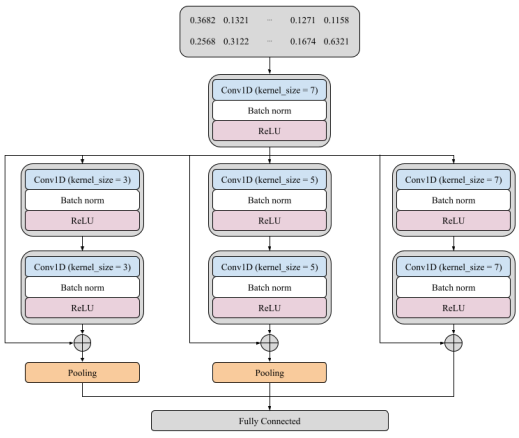

Figure 2. Architecture of DACO

Hyperparameter Optimization

DACO developed with PyTorch1.9 and source code with optimized model could be found in https://github.com/LEEKIWOOK/DACO. Training epoch used in DACO is 500 and early step is 10 for training. And the selected model is shown the highest spearman correlation in validation steps. and the highest model in 5 cross validation is finally chosen. DACO uses stochastic gradient descent optimizer and learning scheduler is stochastic gradient descent with warm restarts.

Author Contributions and Notes

Ki-wook designed research, performed research, wrote software, analyzed data; and wrote the paper.

Availability of Data and Code

All code is available at https://github.com/LEEKIWOOK/DACO.

References

1. Jinek M., Chylinski K., Fonfara I., Hauer M., Doudna J.A., Charpentier E. A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. Science. 2012;337:816–821.
2. Zetsche B., Gootenberg J.S., Abudayyeh O.O., Slaymaker I.M., Makarova K.S., Essletzbichler P. Cpf1 Is a single RNA-guided endonuclease of a class 2 CRISPR-Cas system. Cell. 2015;163:759–771.
3. Doench J.G., Fusi N., Sullender M., Hegde M., Vaimberg E.W., Donovan K.F. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat Biotechnol.
4. Kim H.K., Kim Y., Lee S., Min S., Bae J.Y., Choi J.W. SpCas9 activity prediction by DeepSpCas9, a deep learning-based model with high generalization performance. Sci Adv. 2019;5(11):eaax9249. doi: 10.1126/sciadv.aax9249.
5. Kim, H. K., Min, S., Song, M., Jung, S., Choi, J. W., Kim, Y., ... & Kim, H. H. (2018). Deep learning improves prediction of CRISPR–Cpf1 guide RNA activity. Nature biotechnology, 36(3), 239-241.
6. Wang, D., Zhang, C., Wang, B., Li, B., Wang, Q., Liu, D., ... & Wang, Y. (2019). Optimized CRISPR guide RNA design for two high-fidelity Cas9 variants by deep learning. Nature communications, 10(1), 1-14.
7. Xiang, X., Corsi, G. I., Anthon, C., Qu, K., Pan, X., Liang, X., ... & Luo, Y. (2021). Enhancing CRISPR-Cas9 gRNA efficiency prediction by data integration and deep learning. Nature communications, 12(1), 1-9.
8. Deng, L., Wu, H., Liu, X., & Liu, H. (2021). DeepD2V: A Novel Deep Learning-Based Framework for Predicting Transcription Factor Binding Sites from Combined DNA Sequence. International journal of molecular sciences, 22(11), 5521.
9. Ng, P. (2017). dna2vec: Consistent vector representations of variable-length k-mers. arXiv preprint arXiv:1701.06279.
10. Wang T, Wei JJ, Sabatini DM, Lander ES. Genetic screens in human cells using the CRISPR-Cas9 system. Science. 2014;343:80–4.

11. Hart, T., Chandrashekhar, M., Aregger, M., Steinhart, Z., Brown, K. R., MacLeod, G., ... & Moffat, J. (2015). High-resolution CRISPR screens reveal fitness genes and genotype-specific cancer liabilities. Cell, 163(6), 1515-1526
12. Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., ... & Liu, Q. (2018). DeepCRISPR: optimized CRISPR guide RNA design by deep learning. Genome biology, 19(1), 1-18.
13. ENCODE Project Consortium. (2004). The ENCODE (ENCyclopedia of DNA elements) project. Science, 306(5696), 636-640.
14. Bengio, Y. (2009). Learning deep architectures for AI. Now Publishers Inc.
15. Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.
17. Yuan, Q., Wei, Y., Meng, X., Shen, H., & Zhang, L. (2018). A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(3), 978-989.
18. Zhang, W., Tang, P., & Zhao, L. (2019). Remote sensing image scene classification using CNN-CapsNet. Remote Sensing, 11(5), 494.
19. Huan, E. Y., & Wen, G. H. (2019). Multilevel and multiscale feature aggregation in deep networks for facial constitution classification. Computational and mathematical methods in medicine, 2019.
20. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
21. Robbins, H., & Monro, S. (1951). A stochastic approximation method. The annals of mathematical statistics, 400-407.
22. Loshchilov, I., & Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.
23. Shen, Z., Bao, W., & Huang, D. S. (2018). Recurrent neural network for predicting transcription factor binding sites. Scientific reports, 8(1), 1-10.