SimCSP: A Simple Contrastive Model for Splice Site Prediction

Kevin Stull
University of Colorado Boulder
Email: kest3869@colorado.edu

Oluwatosin Oluwadare
University of Colorado Colorado Springs
Email: ooluwada@uccs.edu

Abstract

Splice site prediction plays a vital role in the gene expression pipeline and language models have leveraged the pre-training, fine-tuning paradigm to make such predictions with great success. A weakness of traditional BERT architectures is the robustness of their internal representations, which has been addressed in human language models through the introduction of a contrastive objective function during pre-training. Hence SimCSP, a Simple Contrastive model for Splice site Prediction, is proposed. However, since the effect of contrastive learning during pre-training on splice site prediction is not well understood, a new method has been developed to investigate the connection. Which leads to the conclusion that applying a contrastive learning objective function during pre-training can improve metrics correlated with accurate classification, but that does not necessarily lead to better downstream performance after fine-tuning. The paradigmatic phenomena commonly referred to as catastrophic forgetting may provide some insight into the surprising results elucidated by this study of the SimCSP algorithm and its effects on splice site prediction.

Introduction

Accurately modelling gene expression is one of the great unsolved problems in biology (Dev 2015). DNA Splice Site Prediction (SSP) is a critical step in that pipeline that needs more robust investigation. Given the large cost of experimentally determining those locations, computational models have received a lot of attention from the scientific community. The primary drawback of such an approach is their insufficient reliability for predicting locations correctly (Chen et al. 2023).

Recently, deep learning has provided a great deal of progress in the field through two approaches, Convolutional Neural Networks (CNN)s (Akpokiro et al. 2023) and Masked Language Models (MLM)s (Yelmen and Jay 2023). CNNs leverage large swaths of labelled data to suss out the features which inform the location of splice sites (Ji et al. 2021). MLMs further leverage the abundance of data through a process called pre-training. Pre-training is a self-supervised machine learning algorithm that allows a model to create internal representations of a language through automatic labelling of an unlabelled training corpus(Erhan et al. 2010). One such architecture applied to modelling DNA is the BERT (Devlin et al. 2019) architecture. **B**iderectional **E**ncoder **R**epresentations from **T**ranformers models pre-train on large unlabelled corpora by masking some portion of their inputs then predicting how the masks should be filled in.

It has been shown that the embeddings produced by BERT architectures can be improved through the use of contrastive learning (CL) (Gao, Yao, and Chen 2022). While this technique was used specifically to improve performance on semantic similarity tasks for human language, it is unclear how transferable this is to a DNA based tasks, particularly the classification of splice sites. The SimCSE algorithm did not see an improvement in all binary classification tasks, however there was no further investigation into the rationale for this phenomena since it was not the main focus of their study. This also turned out to be the case with the TaCL (Su et al. 2021) algorithm, which introduced CL at the token level instead of at the sentence level. The TaCL algorithm saw the least improvement in binary classification which provides SimCSP with the opportunity pick up where others in the field have left off, further exploring the connection between contrastive learning and binary classification for language models.

Related Work

Traditional Machine Learning

Before the mainstream adoption of deep learning, there were several different approaches to the problem of SSP. GeneSplicer (Pertea, Lin, and Salzberg 2001), for example, used an ensemble of feature detectors and Markovian techniques to detect splice sites. In 2003, support vector machines (Zhang et al. 2003) were applied to the problem. Later on, support vector machines were combined with other techniques, like principal component analysis (Pashaei et al. 2016) for improved results.

Convolutional Neural Networks and Long Short Term Networks

Deep learning's contributions to SSP began with the application of CNNs. Models such as Deep Splicer

(Fernandez-Castillo et al. 2022), Splice2Deep (Albaradei et al. 2020), and EnsembleSplice (Akpokiro, Martin, and Oluwadare 2022) surpass traditional machine learning approaches. Other notable CNN networks include, but are not limited to, SpliceRover (Zuallaert et al. 2018) and SpliceFinder (Wang et al. 2019). All of the models mentioned can suffer from the shortcomings commonly associated with CNNs, including limited receptive fields and a propensity to over-fit during training. Long-short term networks have also been applied (Singh, Nath, and Singh 2022) and while they do address the problem of a local receptive field, vanishing and exploding gradients limit the size of the input that can be processed by the model. Further, because CNNs are highly sensitive to the training set used (Scalzitti et al. 2021), they are commonly limited to fully supervised training.

Language Models

These facts motivate the introduction language modelling to the DNA SSP problem. There have been several successful generalizations of the BERT algorithm to DNA representation and SSP (Dalla-Torre et al. 2023) (Ji et al. 2021)(Mo et al. 2021) (Cahyawijaya et al. 2022). Further, it has been shown that evolutionary and genetic information is encoded in the layers of the transformer architecture (Chen et al. 2023). These encodings can be visualized and inform what features of a nucleotide sequence are most useful for the identification of splice sites (Chen et al. 2023). It is possible that the information gained from these visualizations could be used to inform the choices made during pre-training.

Problem Statement

Put succinctly, we investigate which changes made during pre-training, due to contrastive learning (CL), affect the downstream classification result after performing fine-tuning. Expressed formally, language models have hidden layers l and a classification head c. Therefore, a simplified language model m can be represented as $m = l + c$. We let c_2 be some general method to transform the output of the language model's hidden layers into some binary classification (BC).
However, the self-supervised task which the model is pre-trained on, usually masked language modelling (MLM) or in the case of SimCSP, MLM followed by CL, uses a different classifier whose many decision boundaries divide the space into subsets which each represent one member of the pre-training task's output space. We call this classification head c_p, where p is the number of outputs for the pre-training task where it is assumed that $p > 2$. Since only l is shared between the two models, we can summarize our two models as: $m_p = l + c_p$ and $m_f = l + c_2$, where m_p is the pre-trained model and m_f is the fine-tuned model. Since $p > 2$, there is no direct metric which can compare m_p and m_f.
Assuming $f()$ is some permutation of a model, MLM, CL, BC and $e()$ is some evaluation metric; F1, accuracy, AUC. The only qualitative means of measuring how $f(m_p)$ relates to $e(m_f)$ is to transfer l to a new model where a c_2 classification head can be fit to the downstream task of F1 score. Which is feasible when only fitting c, but becomes restrictive when l is also fine-tuned to the downstream task, as is commonly the case with language models.
Given that there exists some evaluation metrics $e^*()$ which can be applied directly to l. How does $f_{CL}(m_p)$ affect $f_{BC}(m_f)$ and is $e^*(l)$ sufficient to predict the relationship between them?

Approach

Dataset

For pre-training, all chromosomes of the primary assembly GRCh38/hg38 were used, the data set was loaded using The Nucleotide Transformer's (Dalla-Torre et al. 2023) HuggingFace train split. This is an unlabelled data set (with respect to SSP) containing DNA sequences from humans. For fine-tuning, the Spliceator data set (Scalzitti et al. 2021) is used, the code used to process and load the data set were obtained from the github page of the SpliceBERT paper (Chen et al. 2023). The data set contains DNA sequences of varying length, 400 or 600, that are labelled as a non-splice site, an acceptor site, or a donor sites. The acceptor donor distinction is not used as this study is interested in binary classification. That gives a final data set which contains 400 nucleotide sequences that are labelled as either 0 non-splicing or 1 splicing sites.
For the purpose of evaluating SimCSP in an self-supervised setting, the Spliceator data set was re-arranged into a new data set called Spliceator for Semantic Similarity (S3). In this new data set, the labelled sequences are randomly paired together without replacement. Then, if the elements share a label (both are splice sites or both are not splice sites), they are given a new label of 1, otherwise, the pair is given a label of 0. Each pair is considered a single training example of sequences that are (0) not semantically similar, or (1) semantically similar. Using the SCCS metric described in greater detail in the Evaluation Metrics section, this new data set, S3, can be used to evaluate the model's understanding of SSP during pre-training and during fine-tuning.
When bench-marking SimCSP for comparison with other methods, the zebra fish, fruit fly, worm, and arabidopsie were used.

Theoretical Foundations

A contrastive loss function is used for the pre-training of SimCSP. It is functionally identical to the one introduced for SimCSE (Gao, Yao, and Chen 2022), which was used for the unsupervised contrastive learning of sentence embeddings. If we let x_i be one of the inputs in a batch of N inputs to the model. Then $\hat{x}_i$ and $\bar{x}_i$ are two embeddings produced by that same input x_i to the encoder with two different dropout masks. A dropout mask is a shorthand term to describe the dropout applied throughout a standard BERT architecture. For readability's sake, subscripts are not applied to each dropout mask but it is assumed that no two are identical to one another. Then the contrastive loss is given by the expression:

$$loss(\hat{x}_i) = -\log \frac{\exp(sim(\hat{x}_i, \bar{x}_i))}{\sum_{j=1}^{N} \exp(sim(\hat{x}_i, \bar{x}_j))} \qquad (1)$$

Where $\text{sim}(x_1, x_2)$ is defined as the cosine similarity between two vectors. That is:

$$sim(x_1, x_2) = \frac{x_1^T x_2}{x_1 \cdot x_2} \quad (2)$$

In application, the Multiple Negatives Ranking Loss function is used from the sentence transformers library (Reimers and Gurevych 2019), where positive pairs are generated by taking the same sequence twice with different dropout masks and other sequences are assumed to be negative samples.

Evaluation Metrics

The foremost metric used to evaluate SimCSP is Splice Site Prediction (SSP) F1 score (SSP). ROC AUC is also used to validate the models during fine-tuning. These metrics can only be applied to labelled data, thus they cannot be used to directly quantify the changes caused by CL during pre-training, for that reason, other metrics are introduced.

Two metrics will be introduced, Normalized Mutual Information (NMI) and Spearman Correlation with Cosine Similarity (SCCS). The possibility that either of these serve as a proxy for F1 is investigated. These metrics have been selected because they can be directly applied to this hidden layers of the network without the need to pass through a classification head.

The Spearman Correlation with Cosine Similarity (SCCS) of the last layer's [CLS] token is used as one measure of how similar the model "believes" two sequences are. Cosine similarity can be used to evaluate both the fine-tuned model, and the pre-trained model that it is based on.

The authors of SpliceBERT (Chen et al. 2023) utilized the UMAP technique to visualize their nucleotide embeddings, then used the Leiden algorithm to cluster them. The SpliceBERT embeddings displayed a better degree of separability across coding and non-coding DNA inputs, quantified by the plot's higher Normalized Mutual Information (NMI). Better separability indicates that the its internal representation of a splice site is more robust. CL is applied to the model during pre-training, and its affect on F1 score is studied.

Architectural Characteristics

The SimCSP framework uses Contrastive Learning (CL) to investigate the pre-training of DNA Language Models (LM)s and its effects on SSP. It is used as an additional layer of pre-training after convergence is reached with Masked Language Modelling (MLM). This gives:

SimCSP Architecture

1. Pre-train MLM
2. Pre-train CL
3. Fine-tune SSP

In practice, the SpliceBERT-510.nt-human pre-trained model (Chen et al. 2023) is loaded and it's parameters are modified using Contrastive Learning (CL). A learning rate of $1 * 10^{-4}$ is used with a batch size of 512 and a weight decay of $1 * 10^{-6}$. There are 6 transformer blocks with a hidden size of 512 and 16 self-attention modules per block. The [CLS] token is used as the input to the classifier.

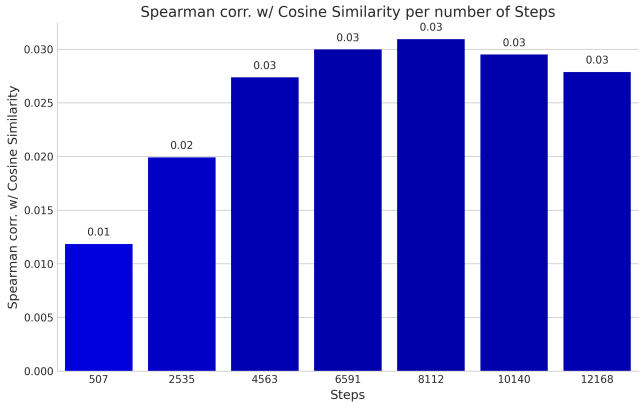

Figure 1: The Spearman correlation with cosine similarity of the [CLS] token with semantically similar and dissimilar validation examples from the Spliceator data set given varying amounts of pre-training.

Model Evaluation

The metrics used to evaluate SimCSP are SCCS, NMI, ROC AUC, and F1 score. The SCCS and NMI metrics can be applied to the hidden layers of the model during pre-training and fine-tuning. The best pre-trained model is selected using the Human Reference Genome (Dalla-Torre et al. 2023) for NMI and the Spliceator data set (Scalzitti et al. 2021) training split for SCCS. The benchmarks used to compare model performance across different architectures is only used for inference and inference was only performed once by the best model. The ROC AUC and F1 scores can only be used during fine-tuning, therefore the ROC AUC was used to score the models using the validation split of the Spliceator data set. The best model was chosen by taking the highest F1 score on the testing split of the Spliceator data set.

Results

Effect of Contrastive Learning on the [CLS] Token

When Contrastive Learning (CL) is applied to the [CLS] token of the DNA Language Model (DNA LM), the effects can be seen in Figure 1. As pre-training progresses, the Spearman Correlation with Cosine Similarity (SCCS) increases until reaching a peak at around 8,112 steps before slowing falling back off. The scale of the change is also worth noting, while model performance does more than double, the numerical distance in performance between the least and most performant model is around 2%.

Effect of Contrastive Learning on the NMI of SimCSP's Layers

Figure 2 is the highest NMI score for SimCSP and occurs after 2,535 batches of CL during pre-training. The plot is from the 4^{th} transformer layer supporting the result of SpliceBERT (Chen et al. 2023), which is that the $2^{nd} - 5^{th}$ layers of the network are the most informative for the prediction of splice sites. This is further supported by Figure 3, which shows the average NMI across a differing number of

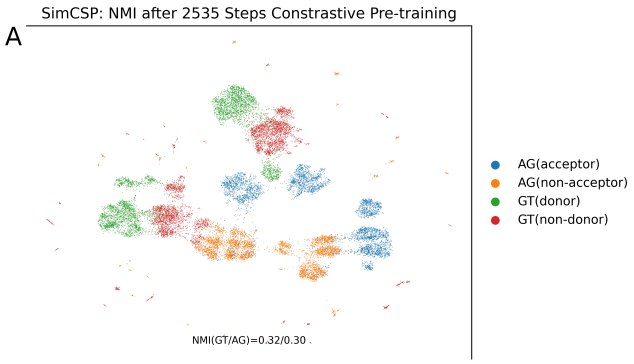

Figure 2: The NMI of the 4^{th} layer of SimCSP after 2,535 steps of Contrastive Learning during pre-training.

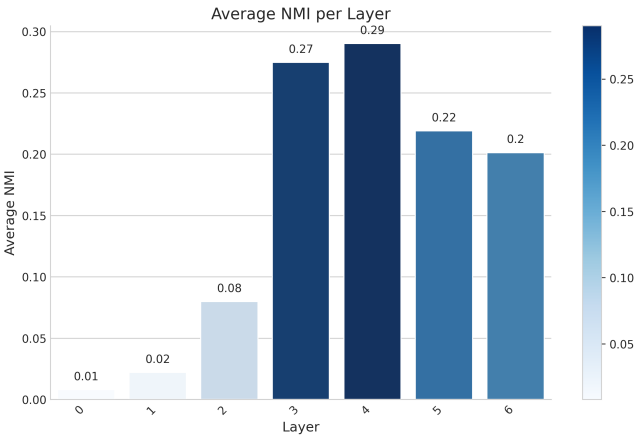

Figure 3: The average NMI score of each layer averaged across varying amounts of pre-training given a contrastive loss function.

training steps by layer. It suggests that most of the semantic information relating to SS are located in 3^{rd} and 4^{th} layers of the model. It is clear that in the long run, the NMI score decreases as more contrastive learning is introduced. However, there is a local peak early on in the epoch that is higher than the starting point, which is used as the best NMI pre-trained model.

Given the previous results, the 4^{th} layer of the model is analyzed in greater detail. Upon inspection, Figure 4 shows a general trend downward as more CL is introduced during pre-training. However, there is a small increase in NMI at 2535 pre-training steps.

Effect of Fine-tuning on NMI and SCCS

When the pre-trained model is fine-tuned, we observe a slight increase in NMI of the plots of the embeddings and a substantial increase in the model's performance on the SCCS metric. With NMI increasing from 0.13 to 0.16 and SCCS increasing from 0.03 to 0.80 on average.

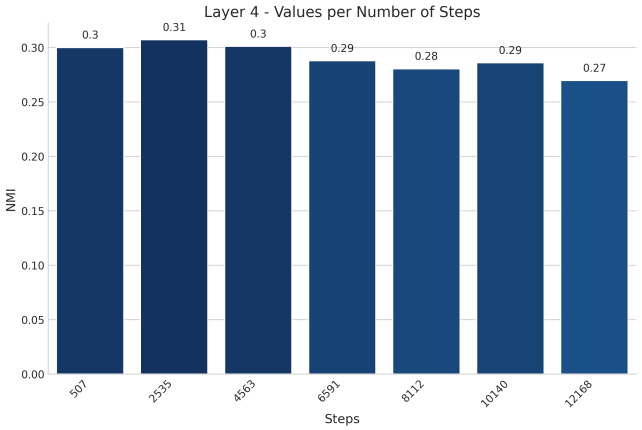

Figure 4: The NMI score of the 4^{th} transformer layer of the SimCSP architecture given varying amounts of pre-training with a CL loss function.

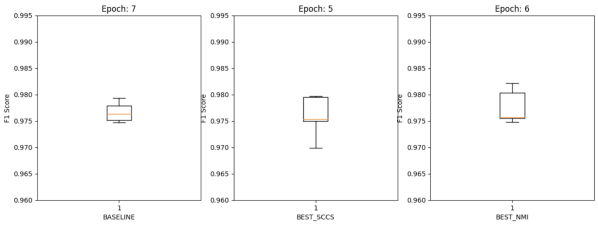

Figure 5: The F1 scores of the Baseline Pre-trained model, the best SCCS from pre-training and the best NMI from pre-training.

F1 Scores of Pre-trained models with best performance on SCCS and NMI

From figure 5, notice that box plots F1 scores of the baseline SpliceBERT-human model are all relatively close, with best NMI skewed towards being slightly better than baseline while best SCCS is skewed towards being slightly worse than baseline. The mean of the baseline model is 0.9766, the mean of the best SCCS model is 0.9759, and the mean of the best NMI model is 0.9777, which means that all models fall within 0.3% of one another in terms of performance.

Discussion

NMI and SCCS are impacted by Fine-tuning

While it is clear that SimCSP has marginal effect on the NMI and SCCS scores during pre-training. It is clear that fine-tuning plays a role in both. This implies that they can serve as proxies for F1 score in a setting where fine-tuning is not practical. While the pattern of slightly increased SCCS and NMI scores due to SimCSP is consistent, it is quite minor. One possible explanation can be provided when considering the differences between MLM and CL.

Contrastive Learning occurs at the Feature Level but MLM happens at the Token Level

CL seeks to improve the grouping of hidden features within the representation space of a model. MLM however, seeks to create generalized relationships between tokens. It is possible that these two objectives are not completely amicable to one another. CL seems to be improving the hidden classes of the representation space at the expense of token-level information. This phenomenon where the model learns new information but forgets old information is referred to as Catastrophic Forgetting (CF). It is possible that CF may be taking place during the training of SimCSP, which would explain why it is possible to optimize parameters associated with better F1 scores, while simultaneously, producing a model that is the same as or worse at its downstream task. CF helps inform why the best models, according to NMI and SCCS, occur after so few steps of CL. It could be the case that 1,000 steps of CL is a local minima where the most utility can be gained from CL before too much is lost due to CF.

Conclusion

Language models are a promising new technique for studying many facets of gene expression, including the prediction of splice sites. Even if a metric can be strongly correlated to better SSP, simply improving that metric by a method such as CL, as was the case for SimCSP, is not sufficient to endow a guaranteed improvement in the downstream performance of the language model. Introducing a new form of learning to a network can also lead to forgetting of information that is necessary for the downstream task of Splice Site Prediction. SimCSP reveals that there are no free lunches when training a deep learning model, each lesson comes at a price. This deeper understanding of contrastive learning's relationship to splice site prediction is crucial if the scientific field is going to produce robust DNA language models.

Acknowledgements

The work in this paper is supported by the National Science Foundation under grant No. 2050919. Any opinions, findings, conclusions, or recommendations expressed in this work do not necessarily reflect the views of the National Science Foundation.

References

Akpokiro, V.; Chowdhury, H. M.; Olowofila, S.; Nusrat, R.; and Oluwadare, O. 2023. Cnnsplice: Robust models for splice site prediction using convolutional neural networks. *Computational and Structural Biotechnology Journal*.

Akpokiro, V.; Martin, T.; and Oluwadare, O. 2022. EnsembleSplice: ensemble deep learning model for splice site prediction. *BMC Bioinformatics* 23(1):413.

Albaradei, S.; Magana-Mora, A.; Thafar, M.; Uludag, M.; Bajic, V. B.; Gojobori, T.; Essack, M.; and Jankovic, B. R. 2020. Splice2Deep: An ensemble of deep convolutional neural networks for improved splice site prediction in genomic DNA. *Gene* 763:100035.

Cahyawijaya, S.; Yu, T.; Liu, Z.; Mak, T. T.; Zhou, X.; Ip, N. Y.; and Fung, P. 2022. Snp2vec: Scalable self-supervised pre-training for genome-wide association study. *arXiv preprint arXiv:2204.06699*.

Chen, K.; Zhou, Y.; Ding, M.; Wang, Y.; Ren, Z.; and Yang, Y. 2023. Self-supervised learning on millions of pre-mRNA sequences improves sequence-based RNA splicing predictio. Technical report. Type: article.

Dalla-Torre, H.; Gonzalez, L.; Mendoza Revilla, J.; Lopez Carranza, N.; Henryk Grywaczewski, A.; Oteri, F.; Dallago, C.; Trop, E.; Sirelkhatim, H.; Richard, G.; et al. 2023. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv* 2023–01.

Dev, S. B. 2015. Unsolved problems in biology—the state of current thinking. *Progress in Biophysics and Molecular Biology* 117(2-3):232–239.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs].

Erhan, D.; Courville, A.; Bengio, Y.; and Vincent, P. 2010. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 201–208. JMLR Workshop and Conference Proceedings.

Fernandez-Castillo, E.; Barbosa-Santillán, L. I.; Falcon-Morales, L.; and Sánchez-Escobar, J. J. 2022. Deep Splicer: A CNN Model for Splice Site Prediction in Genetic Sequences. *Genes* 13(5):907. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Gao, T.; Yao, X.; and Chen, D. 2022. SimCSE: Simple Contrastive Learning of Sentence Embeddings. arXiv:2104.08821 [cs].

Ji, Y.; Zhou, Z.; Liu, H.; and Davuluri, R. V. 2021. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* 37(15):2112–2120. _eprint: https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/39622303/btab083.pdf.

Mo, S.; Fu, X.; Hong, C.; Chen, Y.; Zheng, Y.; Tang, X.; Shen, Z.; Xing, E. P.; and Lan, Y. 2021. Multi-modal Self-supervised Pre-training for Regulatory Genome Across Cell Types. arXiv:2110.05231 [cs, q-bio].

Pashaei, E.; Yilmaz, A.; Ozen, M.; and Aydin, N. 2016. A novel method for splice sites prediction using sequence component and hidden markov model. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3076–3079. IEEE.

Pertea, M.; Lin, X.; and Salzberg, S. L. 2001. GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Research* 29(5):1185–1190.

Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Scalzitti, N.; Kress, A.; Orhand, R.; Weber, T.; Moulinier, L.; Jeannin-Girardon, A.; Collet, P.; Poch, O.; and Thompson, J. D. 2021. Spliceator: multi-species splice site prediction

using convolutional neural networks. *BMC Bioinformatics* 22(1):561.

Singh, N.; Nath, R.; and Singh, D. B. 2022. Splice-site identification for exon prediction using bidirectional LSTM-RNN approach. *Biochemistry and Biophysics Reports* 30:101285.

Su, Y.; Liu, F.; Meng, Z.; Lan, T.; Shu, L.; Shareghi, E.; and Collier, N. 2021. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*.

Wang, R.; Wang, Z.; Wang, J.; and Li, S. 2019. SpliceFinder: ab initio prediction of splice sites using convolutional neural network. *BMC Bioinformatics* 20(23):652.

Yelmen, B., and Jay, F. 2023. An Overview of Deep Generative Models in Functional and Evolutionary Genomics. *Annual Review of Biomedical Data Science* 6(1):null. _eprint: https://doi.org/10.1146/annurev-biodatasci-020722-115651.

Zhang, X. H.; Heller, K. A.; Hefter, I.; Leslie, C. S.; and Chasin, L. A. 2003. Sequence information for the splicing of human pre-mrna identified by support vector machine classification. *Genome Research* 13(12):2637–2650.

Zuallaert, J.; Godin, F.; Kim, M.; Soete, A.; Saeys, Y.; and De Neve, W. 2018. SpliceRover: interpretable convolutional neural networks for improved splice site prediction. *Bioinformatics* 34(24):4180–4188.