Transformer Model for Genome Sequence Analysis

Anonymous Author(s)
Affiliation
Address
email

Abstract

One major challenge of applying machine learning in genomics is the scarcity of labeled data, which often requires expensive and time-consuming physical experimentation under laboratory conditions to obtain. However, the advent of high throughput sequencing has made large quantities of unlabeled genome data available. This can be used to apply semi-supervised learning methods through representation learning. In this paper, we investigate the impact of a popular and well-established language model, namely *BERT*, for sequence genome analysis. Specifically, we adapt *DNABERT* [Ji et al., 2021] to *GenomeNet-BERT* in order to produce useful representations for downstream tasks such as classification and semi-supervised learning. We explore different pretraining setups and compare their performance on a virus genome classification task to strictly supervised training and baselines on different training set size setups. The conducted experiments show that this architecture provides an increase in performance compared to existing methods at the cost of more resource-intensive training.

1 Introduction

Just as human beings use languages to communicate, nature created its own language: Genomes. In order to understand this "language of life", Natural Language Processing (NLP) methods have been used with the aim of decoding instructions and information contained within [Asgari and Mofrad, 2015]. As to unravel the complex function and structures of cells hidden within their genomes, semi-supervised learning can be applied to improve the capabilities to identify new genome structures, impute missing nucleotides (NTs), and classify genomic data under sparse label conditions [BMBF, 2020]. Deep learning based methods have recently achieved breakthroughs in bioinformatics by exceeding the performance of previous state-of-the-art approaches [Zhang et al., 2021]. Self-supervised models have prevailed in NLP, as they take advantage of readily available amounts of unlabeled data in the form of texts to pretrain model weights and representations in a self-supervised manner, thereby leading to higher performance on downstream language tasks. [Devlin et al., 2018, Radford et al., 2018, Peters et al., 2018]. One architecture among those models that have proven particularly useful for representation learning of genomic data is the Bidirectional Transformer-Encoder [Devlin et al., 2018, Rives et al., 2020, Ji et al., 2021, Le et al., 2021, Mo et al., 2021, Avsec et al., 2021]. Using NLP methods for DNA data is an attractive idea, as both written human language and genome data coincide in their method of representing data as sequences of discrete information: Letters or words for language, and NTs in the case of DNA. However, many NLP methods, in particular *BERT*, rely on *tokenization* of text into discrete words or sub-word units [Wu et al., 2016]. While words as units of the information above the character-level are straightforward for humans to recognize and encode in the natural language domain, there is no readily apparent way

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

of tokenizing DNA sequences in general[1]. A simple and yet effective approach to tokenization of DNA data is to use k-mers: Encoding "words" of DNA as units of k sequential NTs. Ji et al. [2021] introduce this method to decode human genome data with *DNABERT* and achieved promising results.

In this work, we analyze the potential of the Bidirectional Transformer-Encoder applied to virus genome sequence data by implementing *GenomeNet-BERT*. In a small test framework, optimal hyperparameter and tokenization settings are explored. Furthermore, two additional strategies, differing in data preprocessing and pretraining setup to the original model implementation, are pursued. The evaluation of model performance is conducted on the task of identifying bacteriophages from short sequences of NTs over various label scarcity scenarios and sequence lengths, by comparing it to the performance of the same architecture trained in a fully supervised fashion, and the Self-GenomeNet [Gündüz et al., 2022] architecture. Compared to fully supervised methods, this one provides reusability for a variety of downstream tasks and is well suited for further improvements on data preprocessing and pretext task tuning, since its architecture is not dependent on these. Given that the *BERT* architecture is highly explicable with its attention layers, its application helps to understand the importance of nucleotide snippets in terms of classification. Further, the model uses raw nucleotide sequence data as input, which reduces the data preprocessing overhead [Buermans and den Dunnen, 2014].

2 Method

The *BERT* model is built of 12 stacked Transformer-Encoder blocks and relies solely on bidirectional attention and fully connected layers to learn representations for each input token. Two pretext tasks aid this purpose: Predicting randomly masked words and whether two input sequences are consecutive in the source they were extracted from. *DNABERT* [Ji et al., 2021] takes genome sequences as input. Unlike BERT, next sentence prediction is not used as a pretext task. To attain input sequences, genomes are split into non-overlapping sequences of sampled length and cut from randomly sampled locations. These are then tokenized using all permutations of the k-mer representation of size 768, which creates tokens with stride 1 (see Figure 1). Because of this overlap of NTs per token, it is possible to simply infer a masked token by its neighbors. Therefore, instead of randomly sampling a percentage of m tokens to mask, k consecutive tokens per sampled masking location are masked.

Procedure We devised a three-step-method, starting with hyperparameter optimization within a scaled-down framework to find values for learning rate, masking percentage, weight decay, and others, leading to the best performance for our tasks, as well as compare different strategies in regard to data preprocessing, tokenization and pretext task. Subsequently, three architecture designs based on *DNABERT* [Ji et al., 2021] (referred to as *GenomeNet-BERT* in the following), differing mainly in tokenization and pretext task, are pretrained full scale. *GenomeNet-BERT* models were all pretrained for 100k steps due to loss plateauing and associated comparability reasons between models. Finally, after subsequent supervised training on our bacteriophage classification task, the performance of our models is compared to the same architecture fully supervised trained and *self-genomenet*, a self-supervised model proposed by Gündüz et al. [2022]. Fine-tuning is performed over different training set sizes, representing various scenarios of label scarcity. Additionally, two distinct input sequence lengths (150 and 1000 nucleotide lengths, respectively) are examined. The data used is described in the Appendix (see Section A.2).

Architecture Designs After self-supervised pretraining, all setups are fine-tuned in a supervised manner on balanced, labeled subsets of the dataset, and macro-averaged recall $Recall_M$ (in %), as well as F_1-score, are then used to measure model performance on a separate prediction set. The first adaptation, *GenomeNet-BERT*, is a replica of the *DNABERT* setup adapted for our purposes. It tokenizes sequences of up to 510NTs to 6-mers and masks 6 consecutive tokens at 2.5% sampled token locations, leading to a token masking rate of 15%. A learning rate of 4×10^{-4}, linearly warmed

[1] One possible suggestion would be to use proteins corresponding to coding DNA, but this method would not cover non-coding DNA.

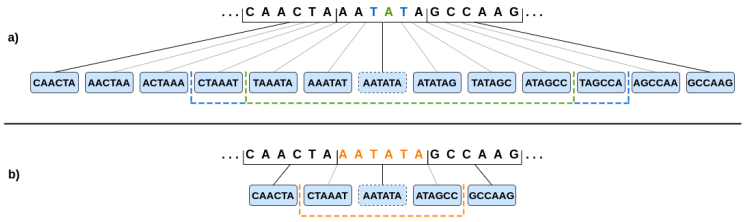

Figure 1: Creation of input tokens as k-mers from an excerpt of a nucleotide sequence.
a) 6-mer tokenization: *DNABERT6* setup, all permutations (stride 1), creating 13 tokens from 18 NTs. **b)** 6-mer tokenization: *GenomeNet-BERT-stride3* setup, creating 5 tokens from 18 NTs.
Dashed border: Sampled masking location during pretraining. Dashed green box: Tokens masked for defined masking location for base *GenomeNet-BERT* setup. Blue extensions: Mask range addition performed for *GenomeNet-BERT-mask8* setup. Orange box: Masked tokens, hidden distinct NTs are highlighted.

up over 5% of the total steps and AdamW-Optimizer [Loshchilov and Hutter, 2019] are applied. In contrast to the other setups, the hyperparameters are based on the original settings proposed by Ji et al. [2021]. The main reasoning behind altering pretraining strategies stems from how k-mer tokenization and masking interact, to only hide 2.5% of distinct NTs for this first setup, which was perceived as low. Therefore, the second adaptation, *GenomeNet-BERT-mask8*, is designed to mask more NTs. It masks 8 consecutive tokens at 2.875% sampled token locations, which leads to more than double the amount of NTs hidden from the model, while the ratio of masked tokens remains the same at 15% (see Figure 1). The third adaptation, *GenomeNet-BERT-stride3*, is intended to work with longer input sequences and train faster and up to 1000NTs long sequences are tokenized to 6-mers of stride 3 and 3 consecutive tokens at 5% sampled token locations are masked. This leads to a sixfold increase in the number of NTs hidden during pretraining compared to the first *GenomeNet-BERT* implementation. Tokenizing sequences with up to 1000NTs to 6-mers with stride 3 leads to 332 input tokens in total and the model is further hard-limited to input sequences up to 340 tokens compare to the standard 512 tokens for BERT.

3 Experiments

Label availability scenarios are artificially created by limiting access to a specific subset of FASTA files during training. As in the semi-supervised protocol of Henaff [2020], 1% and 10% labeled data is used. In addition, a very sparse label setting of 0.1% is trained. Since a k of 6 performs best in the experiments by Ji et al. [2021], 6-mers are used in all setups.

Hyperparameter Optimization The impact of hyperparameter settings is evaluated using the *bert-small* configuration [Wolf et al., 2020], since it is closely related to the original emphBERT architecture, but at the same time allows for performing experiments in a less time-consuming fashion. The detailed results of these experiments are listed in the Appendix in Figure A.3. *GenomeNet-BERT-mask8* and *GenomeNet-BERT-mstride3* are accordingly trained with an increased learning rate of 1×10^{-3} and a longer linear warmup of 20k steps. While no masking setup is consistently better across the range of learning rates trialed, setups that mask 8 consecutive tokens can perform equally well or better than the standard setup at the same learning rate, producing the best model of all stride 1 setups.

Fine-Tuning To fine-tune our models, network heads of the pretext task and all representations except the first token are removed. This starting token collects sequence-level information and is fed exclusively to a classifier via an additional projection layer to predict the class of an input sequence (see Figure A.2). Sequences of length 1000NTs tokenized as k-mers with a stride of 1 surpass the input token limitation of *BERT*-base. They are split into two parts, traversed individually, and then concatenated again by a fully connected layer for all *DNABERT*-based models on the 1000NTs task except for *GenomeNet-BERT-stride3*, which can handle inputs up to 1024nt.

	10%		1%		0.1%	
150NTs length	$Recall_M$	F_1	$Recall_M$	F_1	$Recall_M$	F_1
self-genomenet	78.2	0.785	75.3	0.751	67.2	0.700
supervised	71.8	0.710	67.6	0.673	62.4	0.608
GenomeNet-BERT	**85.7**	**0.851**	**82.2**	**0.821**	**77.8**	**0.780**
GenomeNet-BERT-mask8	85.0	0.845	81.7	0.812	76.7	0.762
GenomeNet-BERT-stride3	80.8	0.801	75.1	0.757	65.6	0.654
1000NTs length						
self-genomenet	94.0	-	85.9	-	73.1	0.846
supervised	81.5	0.871	77.2	0.867	70.6	0.773
GenomeNet-BERT	97.9	0.986	**94.4**	**0.968**	**87.8**	**0.930**
GenomeNet-BERT-mask8	97.2	0.983	91.7	0.953	81.7	0.901
GenomeNet-BERT-stride3	**98.1**	**0.988**	90.9	0.949	87.2	0.927

Table 1: Performance results through semi-supervised training on sequences of 150 nucleotide length (above) and 1000 nucleotide length (below). Percentages represent the three label availability scenarios during fine-tuning on the phage/non-phage virus task.

4 Results and Discussion

Table 1 compares model performance for sequences of 150 and 1000NTs, respectively. Throughout all 6 scenarios, *GenomeNet-BERT*-based models show superior performance compared to *self-genomenet*, the base *GenomeNet-BERT* appearing the best on average overall. The *mask8* variant performs very similar to the base *GenomeNet-BERT* model, while the *stride3* variant provides less successful class predictions for 150NTs input sequences. However, it can be seen that the *stride3* variant exhibits similar or better accuracy than the other variants for 1000NTs input sequences. Since the *stride3* model variant has a shorter input length, it trains notably faster than the other *GenomeNet-BERT* models. Generally, the pretrained *GenomeNet-BERT*-model manifests an about 20% increase in recall than the strictly supervised baseline in all scenarios and increasing in the label scarcer setups of 1% and 0.1%. The *GenomeNet-BERT* model also shows an impressive accuracy in the low label scenario of 0.1%, outperforming *self-genomenet* by about 16% and 20% in recall for 150 and 1000NTs, respectively.

We have shown that *DNABERT*, implemented for use with human genome sequences, is also capable of learning representations from virus genome sequences. Our virus pretrained version, referred to as *GenomeNet-BERT*, outperforms the given baseline at all input length and label availabilities on the task of identifying bacteriophages from read-level length genome sequence excerpts. The *GenomeNet-BERT* realization, which follows the original setup of *DNABERT6*, also outperforms both permutations (*mask8* and *stride3*) trialed in this task on average. However, since the *GenomeNet-BERT* realization was trained using the hyperparameters proposed by Ji et al. [2021], and HPO was performed using the *bert-small* [Wolf et al., 2020] configuration, it is possible that the method can be further improved by HPO based on the full *BERT* model. While *GenomeNet-BERT-stride3* is less accurate on the shorter input length task, it provides the same level of accuracy at the 1000NTs input length with much lower resource requirements for both pretraining and fine-tuning than the base *GenomeNet-BERT* model. In general, it must be acknowledged that the Transformer-Encoder model is very resource and training time intensive, even compared to other self-supervised models for genome sequence analysis.

An interesting observation in the experiments conducted is that all models trained in these experiments overpredicted the bacteriophage class in every setup. It is possible that the model learns to classify more noisy input sequences as phages, as these could be more diverse in short genome excerpts. For a more definitive evaluation of this model architecture, it is necessary to investigate its performance on a higher number of more diverse downstream tasks.

References

E. Asgari and M. R. K. Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *Plos One*, 10(11), 2015. doi: 10.1371/journal.pone.0141287.

Ž. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *bioRxiv*, 2021. doi: 10.1101/2021.04.07.438649. URL https://www.biorxiv.org/content/early/2021/04/08/2021.04.07.438649.

BMBF. Genomenet – entwicklung und evaluierung von genomenet für die de novo identifizierung von noch unbekannten genomischen strukturen und zur probabilistischen dna-sequenzimputation - dlr gesundheitsforschung, Apr 2020. URL https://www.gesundheitsforschung-bmbf.de/de/genomenet-entwicklung-und-evaluierung-von-genomenet-fur-die-de-novo-identifizierung-von-10890.php.

H. Buermans and J. den Dunnen. Next generation sequencing technology: Advances and applications. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1842(10):1932–1941, 2014. ISSN 0925-4439. doi: https://doi.org/10.1016/j.bbadis.2014.06.015. URL https://www.sciencedirect.com/science/article/pii/S092544391400180X.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

H. A. Gündüz, M. Binder, X.-Y. To, R. Mreches, P. C. Münch, A. C. McHardy, B. Bischl, and M. Rezaei. Self-genomenet: Self-supervised learning with reverse-complement context prediction for nucleotide-level genomics data, 2022. URL https://openreview.net/forum?id=92awwjGxIZI.

O. Henaff. Data-efficient image recognition with contrastive predictive coding. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4182–4192. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/henaff20a.html.

Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 02 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab083. URL https://doi.org/10.1093/bioinformatics/btab083.

N. Q. K. Le, Q.-T. Ho, T.-T.-D. Nguyen, and Y.-Y. Ou. A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. *Briefings in Bioinformatics*, 22(5), 02 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab005. URL https://doi.org/10.1093/bib/bbab005. bbab005.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

S. Mo, X. Fu, C. Hong, Y. Chen, Y. Zheng, X. Tang, Z. Shen, E. P. Xing, and Y. Lan. Multi-modal Self-supervised Pre-training for Regulatory Genome Across Cell Types. *arXiv e-prints*, art. arXiv:2110.05231, Oct. 2021.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018.

A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2020. doi: 10.1101/622803. URL https://www.biorxiv.org/content/early/2020/12/15/622803.

E. W. Sayers, M. Cavanaugh, K. Clark, J. Ostell, K. D. Pruitt, and I. Karsch-Mizrachi. GenBank. *Nucleic Acids Research*, 48(D1):D84–D86, 10 2019. ISSN 0305-1048. doi: 10.1093/nar/gkz956. URL https://doi.org/10.1093/nar/gkz956.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL `https://arxiv.org/abs/1609.08144`.

Y. Zhang, J. Yan, S. Chen, M. Gong, D. Gao, M. Zhu, and W. Gan. Review of the applications of deep learning in bioinformatics. *Current Bioinformatics*, 15(8):898–911, 2021. doi: 10.2174/1574893615999200711165743.

A Appendix

A.1 The DNABERT Architecture

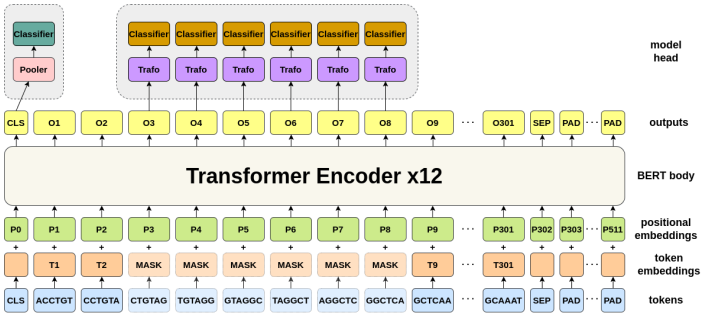

Figure A.2: Visualization of the *Genomenet-BERT* pipeline for an input example of 301 tokens. Right model head: masked language model training head attached during pretraining. Left model head: Sequence classification, present during fine-tuning.

A.2 Virus Data

All self-supervised learning models are trained and evaluated on a collection of viral genome sequences. On August 2nd, 2021, all available viral genome data was downloaded from *GenBank* [Sayers et al., 2019] and divided into two taxonomic classes: Bacteriophages, and Other Viruses. This collection of about 40k FASTA files includes about 1 billion NTs for the bacteriophage class and 0.5 billion for other viruses and poses the binary classification task of identifying whether a read-level length nucleotide sequence is an excerpt of a bacteriophage genome. All self-supervised models are pretrained using unlabeled nucleotide sequences generated from a training split of the data.

A.3 Trial Results

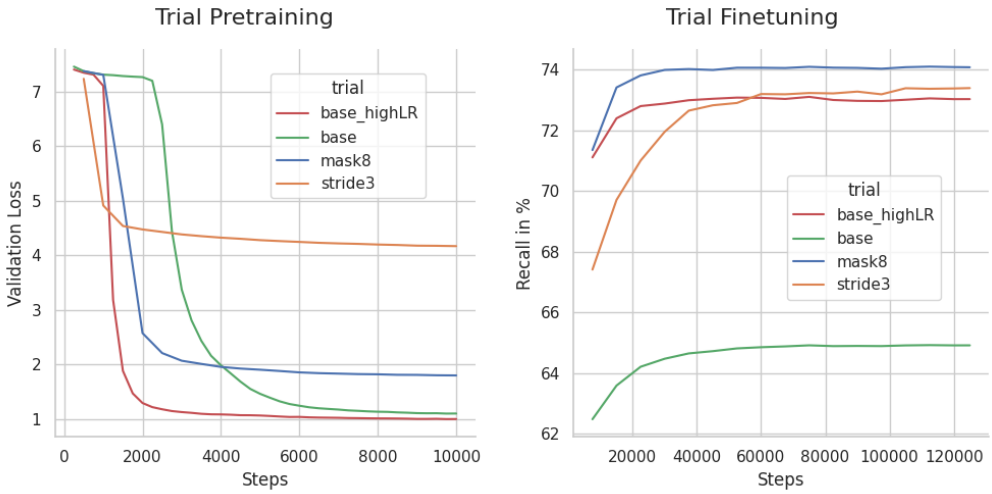

Figure A.3: Pretraining and Finetuning for some selected trials. *base* (green) here represents the scaled-down version of *GenomeNet-BERT* with all the same training parameters, while *mask8* (blue) and *stride3* (orange) do so for the other two variants pursued full scale. *base_highLR* (red) poses as a baseline to the higher learning rate setups of *stride3* and *mask8* with an equal learning rate of $1e^{-3}$ and is equal to *base* otherwise. **Left:** Cross-entropy loss of MLM on a validation set during self-supervised pretraining. **Right:** Class averaged recall during supervised finetuning with frozen representation model layers on the 150nt virus phage/non-phage classification task.

A.4 Computational Information

Pretraining was conducted for 190h on 8 nvidia-A100-40Gb GPUs for *GenomeNet-BERT* & *GenomeNet-BERT-mask8* and 141h on 5 of the same GPUs for *GenomeNet-BERT-stride3*.