PMI-GUIDED MASKING STRATEGY TO ENABLE FEW-SHOT LEARNING FOR GENOMIC APPLICATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning effective gene representations is of great research interest. Lately, large-scale language models based on the *transformer* architecture, such as DNABert and LOGO, have been proposed to learn gene representations from the *Human Reference Genome*. Although these large language models outperform previous approaches, currently, no study empirically determined the best strategy for representing gene sequences as tokens. The uniform random masking strategy, which is the default during the pretraining of such masked language models, might lead to pretraining inefficiency, resulting in suboptimal downstream task performance in the few-shot setting. However, good few-shot performance is critical, with dataset sizes in (personalized) medicine often not exceeding a couple of hundred data points. In this paper, we develop a novel strategy to adapt *Pointwise Mutual Information (PMI) masking* used previously in the NLP setting to the domain of gene sequence modeling. PMI-masking masks spans of tokens that are more likely to co-occur, forming a statistically relevant span. First, we learn a vocabulary of tokens with a high PMI score from our pretraining corpus (the *Human Reference Genome*). Next, we utilize this side information (pre)train our model by masking tokens based on PMI scores. In extensive experiments, we evaluate the effectiveness of the PMI-masking strategy on two baseline models of DNABert and LOGO, over three benchmark datasets (two on promoters and one on enhancers), and on a variety of few-shot settings. We observe that our PMI-masking-guided baseline models substantially outperform the SOTA models. We further observe that almost all the top-ranked DNA tokens in terms of PMI score are closely associated with existing *conserved DNA sequence motifs*.

1 INTRODUCTION

Computational analysis of genomics has revolutionized the field of medical science (McGuire et al., 2020), particularly with the advent of the *Human Reference Genome* (Schneider et al., 2016). As seen in (Yue & Wang, 2018a), deep learning has been applied to various applications, such as protein structure analysis, gene expression data, and transcriptome analysis. Given the sequential nature of gene sequences, several deep learning models found to be effective in the *Natural Language Processing* (NLP) domain have been adopted for genomic applications (Yue & Wang, 2018b; Avsec et al., 2021). The input data for these tasks is often presented as a sequence of nucleotides. Each side of the double-helix DNA strand comprises the bases adenine (A), cytosine (C), guanine (G), and thymine (T)). However, unlike words or sentences in languages, there are no clear semantically demarcated tokens present within the gene sequence. Therefore, to come to a workable solution, researchers (Mo et al., 2021; Ji et al., 2021) use k-mer representation, which is a sliding window of k-length over the entire sequence. For example, if the gene sequence is $ATTCGATGC$, a 6-mer representation will be $ATTCGA$, $TTCGAT$, $TCGATG$ and $CGATGC$. Similar to the NLP domain, the standard approach in gene sequence modeling is to pretrain the transformer models by randomly masking (and predicting) tokens, the so-called masked language modeling objective (MLM). Given the lack of rigorous evaluations of whether the methods from the NLP domain transfer to modeling gene sequences, we question whether token selection, masking, or other components of the transformer architecture are restrictive and not grounded in biomedical domain knowledge.

We develop a principled statistical approach based on *Pointwise Mutual Information* (PMI) to automatically identify meaningful spans (tokens) from a DNA sequence. PMI involves taking both the frequencies of a token and its overall uniqueness in the dataset. However, the standard PMI formulation (Levine et al., 2021) sometimes favors tokens with a low frequency of occurrence. To adapt to the genomic setting, we modify the PMI metric to mitigate the above-mentioned issue. The state-of-the-art models (DNABert (Ji et al., 2021) and LOGO (Yang et al., 2021)) in gene regulatory sequence classification tasks are widely used in the literature (Yang et al., 2021; Mo et al., 2021; Badirli et al., 2021). Instead of randomly selecting tokens to mask, as done in DNABert (or LOGO), we use the PMI score to prioritize the *relevant* tokens to mask within a given gene sequence. This strategy is inspired by the work of Levine et al. (2021), which shows that PMI-masking improves over random span masking (Joshi et al., 2020) in NLP tasks. However, the main difference from our work is that they use it to develop a large masking vocabulary in the NLP domain, while we use the PMI score as a ranking function to choose the span of tokens to mask. Instead of using PMI score to formulate an absolute importance notion and consequently create a masking vocabulary, we use the PMI score to measure *relative importance* among DNA sequence k-mer tokens during MLM.

As current medical datasets often face data scarcity issues, the move towards personalized medicine requires models that perform well with limited training data at hand (Shaikhina & Khovanova, 2017; Hekler et al., 2019). We, therefore, evaluate our proposed PMI-masking strategy in the low-resource (few-shot) setting. Introducing biomedical domain knowledge in the form of side information is a promising research direction, specifically to deal with limited data (Kyono et al., 2019; Oei et al., 2021; Roy et al., 2021). Our PMI-guided approach builds on that idea and directly includes side information derived from the Human Reference Genome into the training procedure.

Our extensive experimentation shows that *PMI-masking-guided* DNABert and LOGO improve over the standard random masking-guided DNABert and LOGO (pretrained on the same number of steps) respectively, in few-shot settings ($10, 50, 100, 500$ and 1000 training data points per class) over three benchmark datasets of gene sequence classification (two on promoters such as *Prom-core, Prom-300* and one on enhancer such as *Cohn-enh*). We posit that PMI-masking helps incorporate non-trivial genetic knowledge because we observe that *PMI-masking-guided* DNABert pretrained for 10K steps even outperforms original DNABert pretrained for 120K steps for all few-shot settings in case of Prom-300 and Cohn-enh dataset.In addition, we perform motif[1] analysis and finetuning impact analysis to understand the domain knowledge learned by *PMI-guided DNABert.* To alleviate the issue of tremendous engineering effort needed to develop the experimental setup of gene sequence modeling, we plan to make the pretraining and finetuning datasets, model weights and checkpoints, and associated codebase publicly available, as the final contribution.

2 Background

Learning deep representations in the context of gene sequence modeling. Nguyen et al. (2016) encode base pair triples as one-hot vectors to feed into convolutional neural networks for DNA sequence classification tasks, whereas Badirli et al. (2021) convert the DNA barcodes represented by nucleotide sequences into a vector embedding useful for the task of fine-grained species classification. Ng (2017) utilize k-mers to represent gene sequences in their approach using a shallow neural network. Ji et al. (2021); Yang et al. (2021) also represent gene sequences as k-mers, by using a learned dense representation from an adapted BERT model. Mock et al. (2021) largely adapt the DNABert architecture for the task of taxonomy classification and model gene sequences as 3-mers, but also includes next sentence prediction along with MLM training loss. Instead of using k-mer representations, Zaheer et al. (2020) trained a SentencePiece tokenizer on the Human Reference Genome and applied to the tasks of chromatin-profile prediction and promoter region prediction. Mo et al. (2021) infuse domain knowledge into the model by proposing a multimodal pretraining setup comprising gene sequences and information on transcription factors and regions.

Random and PMI-masking for MLM training in NLP. The initial work applied *random token masking*, as performed by BERT (Devlin et al., 2018), where 15% of the input tokens are chosen to be masked uniformly. Previous work has also investigated jointly masking whole words (*whole word masking* (Sennrich et al., 2016; Devlin et al., 2019)) or entities (*entity masking* (Sun et al., 2019)) which both have been shown to be beneficial over masking tokens. Joshi et al. (2020) propose *random span masking* where random spans with lengths chosen from a geometric distribution

[1]Sequence motifs are short, recurring patterns in DNA that are presumed to have a biological function (D'haeseleer, 2006).

are masked at random positions. This simple method has been shown to outperform the more involved entity masking approach. When applied to gene sequences, especially entity and whole word masking have the problem that there is no well-defined concept of entities and words in gene sequences. The *PMI-masking* approach (Levine et al., 2021) builds on the ideas of entity and span masking, but instead of masking random spans, their approach involves measuring the relevance of colocated n-grams (based on their constructed masking vocabulary of 800K tokens) with pointwise mutual information (PMI) and masking these spans together. The authors show that masking PMI tokens (i) accelerates training while matching end-of-pretraining performance in roughly half the training steps and (ii) improves upon previous masking approaches at the end of pretraining.

3 BUILDING BLOCKS OF SOTA MODELS

We focus on the two recent transformer-based pretrained models of DNABert Ji et al. (2021) and LOGO Yang et al. (2021) that are adapted to the gene sequence modeling domain. Through MLM pre-training, these SOTA models learn powerful contextual representations for DNA fragments utilizing abundant unlabeled data from the *Human Reference Genome*, which contains around 3.2 billion base pairs in total over 24 chromosomes.

Implementation details of SOTA models and associated research challenges. The tokenization of gene sequences and MLM training is performed based on the author's codebase (DNA, 2021). However, they do not provide the dataset for pretraining or finetuning (downstream) tasks. Therefore, we follow the author's description to construct the corresponding datasets, which is nontrivial. We explain the pretraining data creation process in this section and later describe the finetuning dataset creation process in Section 5.1.

Tokenization of gene sequences. The gene sequence is first converted into a k-mer representation, which is commonly used in the literature (Ng, 2017; Ji et al., 2021). The k-mer representation is a sliding window of length k. For example, *AGCACGCAG* in 6-mer representation leads to 3 tokens - *AGCACG*, *GCACGA*, *CACGAG*. Thus, the vocabulary comprises all combinations (4^k length) and five special tokens - *CLS, PAD, UNK, SEP, MASK*. According to the set-up chosen by SOTA models, we consider $k = 6$ for all the experiments. Yang et al. (2021) note that 6-mers incorporate richer contextual information while keeping the memory and computational complexity manageable.

Pretraining data preparation. Since the pretraining data is not provided with the author's codebase, we follow the implementation details mentioned in the paper to construct the pretraining dataset, which is a non-trivial task. We obtain the Human Reference Genome from the Genome Reference Consortium Human Build 38 patch release 13 (GRCh38.p13) FASTA file Consortium (2019) from the NCBI website. It serves as a large-scale corpus of unlabeled gene sequence data, which we use for MLM training to obtain a contextual representation of the 6-mer tokens. We perform the following steps to convert the Human Reference Genome to a form that DNABert (or LOGO) can use to train with the MLM objective:

Step 1: For each chromosome c in the Human Reference Genome, we randomly choose the starting index between 1 and 1000 Zaheer et al. (2020).

Step 2: Given the chromosome number and its starting index (ST), we next determine the length of the DNA segment L as BERT has the limitation of accommodating a maximum of 512 tokens. We select L as 510 for 50% of the cases and a randomly selected length between 5 and 510 for the remaining 50% of cases Ji et al. (2021).

Step 3: We thus create the DNA segment comprising the base pairs between ST and $ST + L$ of chromosome c, corresponding to *data point in the pretraining dataset*. We filter out DNA segments that contain bases other than A, T, C, or G.

MLM training. The SOTA models are trained with masked language modeling loss similar to BERT Devlin et al. (2018). However, to mask a nucleotide, a contiguous sequence of tokens is masked to prevent information leakage, as each nucleotide is part of k consecutive k-mers. More formally, say a neucleotide is represented a DNA[i], while a 6-mer token is represented as $\mathrm{T}_6[i]$ = $\{$DNA[i-2], DNA[i - 1] $\cdots$ DNA[i + 3]$\}$, then the tokens T[j], $\forall (j)_{j=i-3}^{i+2}$ is masked. Given that $k = 6$ and 15% of tokens need to be masked (Devlin et al., 2018), the MLM probability is set at: $15\%/6$; that is, 2.5% of the nucleotides are chosen for masking.

4 PROPOSED ADAPTATION OF PMI-MASKING STRATEGY TO GENOMIC APPLICATIONS

In this paper, we use a PMI-based approach to identify spans of k-mers that co-occur much more than expected compared to their components (i.e., k-mers of shorter length, such as 4-mers or 5-mers). These spans then replace the (uniform) random masking strategy used by SOTA models such as DNABert and LOGO. This is to make the masked token prediction task more difficult by removing highly correlated local contexts, which subsequently may improve the pretraining efficiency (as shown by Levine et al. (2021)). In this section, we discuss the novel strategy to first adapt the PMI metric from NLP to the genomic setting (Section 4.1) and then explain how to use PMI-masking for improving the efficiency of MLM training (Section 4.2).

4.1 PMI SCORING FOR GENE SEQUENCES

Although NLP models are applied to the biomedical domain, no study empirically determined the best strategy for representing gene sequences. In this work, we consider a single nucleotide equivalent to a single token in NLP, and thus an *n-gram* from the NLP domain is equivalent to a *k-mer* from the gene sequence modeling literature. We propose a novel strategy to adapt PMI-based scoring to our genomic setting and can help us identify high PMI tokens to mask. *Pointwise Mutual Information* (PMI) quantifies how often two tokens occur compared to what is expected if they are independent. The PMI formula (proposed by Levine et al. (2021)) when extended to k-mers (where k >2) is:

$$\text{PMI}_k(w_1 \dots w_k) = \min_{\sigma \in \text{seg}(w_1 \dots w_k)} \log \frac{p(w_1 \dots w_k)}{\prod_{s \in \sigma} p(s)} \quad (1)$$

Here, seg$(w_1 \dots w_n)$ is the set of all contiguous segmentations of the n-gram "$w_1 \dots w_n$" (excluding the identity segmentation). In a valid segmentation (σ), the original sequence "$w_1 \dots w_n$" can be divided into any number of partitions of positive (>0) size. For example, say for $n=6$, some of the possible valid segmentations are: "($w_1 \dots w_3$), ($w_4 \dots w_6$)" or "(w_1, w_2) (w_3), ($w_4 \dots w_6$)". **The PMI_k formulation many times favors tokens with lower frequency**, that is, the number of times the n-gram gene sequence appears in the *Human Reference Genome*. We thus impose a **discounting factor** that penalizes rare tokens (Pantel & Lin, 2002). We refer to it as the Normalized PMI_k formula, which we finally use for scoring all the individual n-gram sequences.

$$\text{Normalized-PMI}_n(w_1 \dots w_k) = PMI_k * \frac{\log f(w_1 \dots w_k)}{\log(c) + \log f(w_1 \dots w_k)} \quad (2)$$

Here, $f(w_1 \dots w_k)$ refers to the frequency of occurrence of the k-mer sequence of $w_1 \dots w_k$. c refers to the minimum frequency of occurrence (**a constant value used as a threshold to remove rare tokens**). We determine the threshold c in Equation 2 based on the frequency distributions of the entire collection of k-mers (around 199 Million). In our case, we choose c equal to 101 which puts a cut-off beyond the first quartile (25 percentile) of k-mer frequencies.

In this paper, we focus only on computing Normalized-PMI$_\text{n}$ for all k-mer sequences where k=6, and use a top-down memorization approach to reduce algorithmic complexity, by storing already computed PMI scores in memory. As the final step, we develop **a ranked list (*RANK*) of all 6-mers** (4096 in total) based on the decreasing order of Normalized-PMI$_\text{k}$. Next, we discuss how we use the PMI scores as a measure to choose tokens to be masked during MLM training.

4.2 PMI-MASKING STRATEGY

The strategy aims to mask all the nucleotides simultaneously in the most correlated spans. It minimizes information leakage and helps the system to learn deeper patterns. Side by side, we would like to preserve the benefit drawn out of the traditional random masking strategy. Thus, to choose the tokens for MLM training, we perform the following steps.

1. Randomly select m nucleotides as mask center spread uniformly over the DNA string.
2. Corresponding to a selected nucleotide (say DNA[i]), obtain a PMI score $MPMI_i$ in a following manner.

   (a) A 6-mer token corresponding to DNA[i] : $\mathrm{T}_6[i]$ = {DNA[i-2] $\cdots$ DNA[i+3]}
   (b) Obtain the PMI rank of the token as PMI-Rank($\mathrm{T}_6[i]$)
   (c) $MPMI_i = max_{i=i-2}^{i+3}$ (PMI-Rank($\mathrm{T}_6[i]$)
3. Divide the m nucleotides into two sets based upon their MPMI scores, where the high setting is the $m/2$ nucleotides with the highest $MPMI$ scores.
4. For the high set, take all the nucleotides in T[i] (this ensures the masking of correlated spans together). In contrast, for the low set, take only the corresponding nucleotide DNA[i] (this mimics a random masking strategy) for masking.

Determining the value of *m*: To mask a six bp length sequence that is all the nucleotides in a particular token $\mathrm{T}_6[i]$, we need to mask a span of contiguous 11 tokens (6 mask centers, two tokens to the left and three tokens right) while as mentioned a single nucleotide induce masking of 6 tokens. Thus, the expected mask span length per mask center is computed as: $0.5 * 6 + 0.5 * 11 = 8.5$. Followingly, the *mlm probability* is updated from 2.5% to 1.765% (= $15\%/8.5$). Hence if the DNA length is 512, $m \approx 9$.

5 Experimental Setup

Here, we first provide the dataset details and evaluation setup, followed by the model training and baseline model details.

5.1 Datasets

We use three benchmark datasets of gene sequence classification for evaluation purposes. The two datasets of promoter region prediction are not directly made available and involve a significant amount of effort (including a paper implementation) for their construction. The enhancer cohn prediction dataset is directly made available from prior works (Martinek et al., 2022).

Promoter Region Prediction (Prom-core and Prom-300). A *promoter* is a DNA region typically located upstream of the gene, which is the site of transcription initiation (as defined in Zaheer et al. (2020)). The task is to classify a given DNA fragment as a promoter or non-promoter sequence, as followed by previous studies (Ji et al., 2021; Zaheer et al., 2020). However, we follow the instructions of Oubounyt et al. (2019) including the negative data creation) since the datasets are not provided by the authors. Thus, we obtained human TATA and non-TATA promoter data, i.e., including promoter sequences with and without a TATA box (a common promoter-related motif found between -30 to -25 bp (upstream) of a gene's transcription start site), from the Eukaryotic Promoter Database (Dreos et al., 2012), using the website API of the *EPD selection tool* (EPD, 2022).We extracted -249 to $+50$ bp sequences around TSS for the Prom-300 setting and -34 to $+35$ bp for Prom-core setting. We perform the standard train-test split of 70% and 30%, which leads to 53276 and 5920 data points respectively.

Enhancer Cohn Prediction (Cohn-enh). An *enhancer* is a DNA sequence that can bind specific proteins and increase the chance of transcription of a particular gene. This dataset has been adapted from Cohn et al. (2018) and is made available as a benchmark dataset by Martinek et al. (2022) in their Github repository (Gresova et al., 2022). Here, the input is a DNA sequence of 500 bp in length and a binary classification task. We use the train-test split used by Martinek et al. (2022) that leads to 20843 and 6948 data points as train and test datasets respectively.

5.2 Evaluation Setup

We report the standard metrics of accuracy (used for performance comparison) and AUC (stands for Area Under the Receiver Operating Characteristic Curve) used for classification tasks where the class labels are balanced. We follow the standard evaluation setup used in the few-shot text classification setting (Schick & Schütze, 2021a;b). Thus, we assume that we do not have access to a validation dataset (development set) to optimize the hyperparameters and investigate the performance for different training set sizes where $t = 10, 50, 100, 500$, and 1000. For each t, we obtain the training set τ by randomly choosing t number of examples from each class. We report the mean and standard deviation of accuracy and AUC by running the experiments **ten** times by randomly choosing t different training (fine-tuning) examples per class as well as a random seed in each run.

5.3 Implementation Details

Pretraining. The authors of DNABert train for 120K steps on 8 NVIDIA 2080Ti GPUs, which takes 25 days to complete. LOGO is trained for 25 epochs which took around 74 days to complete using four Nvidia Tesla V100 32G GPUs. In this paper, we train all the pretrained models and their variants for only 10000 steps, which takes about 2.5 days to complete for DNABert and around 20 hours for LOGO using four GTX 1080Ti 11GB. We select such a setup for two reasons — (i) to explore different pretrained model variants in a reasonable time because we observe that the perplexity of DNABert (SOTA model) has converged to a low score of 2.526 and is stable over the last 3000 pretraining steps (see Figure 1). (ii) As observed by Levine et al. (2021), PMI-masking learns fast and is thus quite efficient to reap the benefit even with a lower number of pretraining steps. We use the same hyperparameter setting as the original SOTA model (see Section A.2 to know more about training details and hyperparameters used).

Finetuning for gene sequence modeling downstream tasks. We use the same model finetuning hyperparameters and training setup as used by DNABert for the 500 and 1000-shot setting (Ji et al., 2021). The model is finetuned for 5 epochs at a learning rate of $5e^{-5}$, warmup steps percentage of 10%, hidden dropout probability of 0.1, and uses the *AdamW* optimizer. We only update the hyperparameters in order to adapt to the low few-shot setting (10, 50, and 100-shot), where we increase the learning from $5e^{-5}$ to $4e^{-4}$ and reduce the *per GPU train batch size* from 15 to 5. The idea behind these two hyperparameter changes is to indirectly increase the number of weight update operations during finetuning, without overfitting (given the limited training dataset size). We also increase the number of training epochs from 5 to 20 to mitigate finetuning performance stability issues that may occur due to random initialization in low-resource settings (Mosbach et al., 2021).

5.4 Baseline Models

We evaluate our work on two gene transformer-based models - DNABert (Ji et al., 2021) and LOGO (Yang et al., 2021). Both of these models follow **random masking** during the MLM training step instead of the proposed **PMI-masking** strategy. We will refer to the SOTA models that use random masking as the original SOTA model **(ORI)** without any PMI-guided masking. We use the original DNABert model pretrained on 120K steps based on the pretrained model weights provided by Ji et al. (2021), as a baseline model and denote it as **ORI 120K** model.

Fixed PMI-guided Masking Vocabulary (PMI-VOCAB): Similarly to the creation of a PMI-masking vocabulary by Levine et al. (2021), we create a masking vocabulary $\approx$ 10 times the DNABert vocabulary size of 4101 tokens. We first select all possible k-mer sequences ($2 \leq k \leq 10$) whose frequency of occurrence is ≥ 10000. We then rank them using our proposed PMI metric and select the top 40000 as the masking vocabulary.

We explore multiple hyperparameter settings for pretraining (HGA and WC) and finetuning (FS).

Half Gradient Accumulation (HGA): The *gradient accumulation steps* parameter is halved, reducing it from 25 (default DNABert configuration) to 12 steps and consequently reduces the effective batch size (it is the product of per GPU train batch size, GPU count and gradient accumulation steps) by 50%. This aims to reduce the *generalization gap* issue that arises when the training batch size is too large (Keskar et al., 2017; Hoffer et al., 2017).

Warmup Correction (WC): Instead of the fixed number of 10000 warmup steps as in the default configuration of DNABert, we compute it as a percentage of the maximum number of pretraining steps. In the original configuration, only the first 10K steps out of 200K (5% of the maximum number of steps). Since we reduce the maximum steps limit from 200K to 10K, we adjust the warmup number as 500 (5% of 10000 steps) accordingly. The importance of warmup in optimization is highlighted in multiple recent studies (Xiong et al., 2020; Liu et al., 2020; Mosbach et al., 2021).

Finetuning Stability (FS): Longer finetuning epochs of 20 instead of the standard 5 epochs, along with higher learning rate from $5e^{-5}$ to $4e^{-4}$. It improves the finetuning stability in model performance and solves random initialization issues in low-resource settings (Mosbach et al., 2021).

6 Experimental Results

Here, we investigate two research questions: (i) Does PMI-masking strategy lead to reasonable performance improvement over random masking in different few-shot settings? (ii) Does the PMI-based ranking show association with conserved DNA sequence motifs?

6.1 PMI-masking Results

We present the results of the model performance comparison in Table 1. PMI-best represents the best performance of the two model variants: PMI (WC + FS) and PMI (WC + HGA + FS); The performance results for the hyperparameter-based model variants are provided in Table 2. In the case of **DNABert**, PMI-best outperforms all the baseline models for the task of Prom-300 and Cohn-enh, except for 1000-shot Prom-300 setting, which also includes the original DNABert model trained for 120K steps (ORI 120K model). ORI 120K model outperforms PMI-best on shallow data settings (10, 50, and 100-shot) of the Prom-core task, whereas PMI-best outperforms again in 500 and 1000-shot settings. This may be because Prom-core is an easier task with a much shorter context (70 bp. in length as compared to 300 and 500 bp in length in the case of Prom-300 and Cohn-enh task). Thus, the ORI 120K model may simply memorize the gene sequence patterns instead of actually learning intrinsic (or extra) knowledge of gene sequences. Therefore, we also observe that such effect of memorization vanishes is higher data settings (above 500-shot). We observe the highest performance improvement of PMI-best over ORI of 7.47% and 4.68% in DNABert and LOGO, respectively, in the 10-shot (shallow data) setting. However, the performance gap slowly reduces in higher data settings in case of DNABert ($3.47\%, 0.95\%, 0.13\%$ and 0% for 50, 100, 500, and 1000-shot, respectively). In the case of **LOGO**, both for random masking and PMI, we consider the best-performing model setting of DNABert corresponding to a particular x-shot. We observe that the PMI-masking-guided model outperforms random masking-based SOTA models (ORI) in all settings across the three benchmark datasets except for 500-shot Prom core setting. It particularly improves by a large margin in shallow data settings of Prom-core (accuracy improvement of 6.72% and 8.14% in 10 and 50-shot respectively) and Prom-300 (8.09% in 50-shot).

We next report the performance improvement between PMI-masking and random masking (ORI model variant) in **DNABert** versus **LOGO** in terms of average accuracy over all few-shot settings for each task: Prom-core = 1.59% vs. 3.72%, Prom-300 = 2.61% vs. 7.67%, and Cohn-enh = 2.13% vs. 1.95%. We observe that the addition of more finetuning data shows the lowest performance improvement (in between 10-shot and 1000-shot) for Cohn-enh (13.50% and 19.35%) as compared to Prom-300 (32.84% and 64.74%) and Prom-core (27.24% and 33.89%) tasks in case of DNABert and LOGO respectively. However, we observe the *effect of PMI-masking on model performance to be higher in the case of LOGO for Prom-core and Prom-300*. It indicates that PMI-masking is more beneficial for lightweight models like LOGO; heavier models like DNABert might automatically learn a certain amount of span correlations (like PMI) information, thus reducing the independent impact of PMI-masking.

6.2 Ablation Analysis

We show the performance comparison among the different model hyperparameter-based variations (described in Section 5.4) in Table 2. We observe that **finetuning stability (FS)** works very well with limited training data (10, 50, and 100-shot settings), where substantial tremendous improvement is noticed. However, with more training data, such as 500 and 1000-shot settings, we observe a high-performance drop due to overfitting. We, therefore, do not use FS in 500 and 1000-shot settings. We observe that **half gradient accumulation (HGA)** helps to improve the model performance in 10 and 50-shot settings (10 and 50-shot for Prom-300 and 50-shot for Cohn-enh). It is empirically observed that if a large batch size is used to train deep neural networks, the trained models appear to generalize poorly (Keskar et al., 2017; Hoffer et al., 2017). HGA reduces the effective batch size by 50% and mitigates the *generalization gap* issue in such low data settings (10 and 50-shot). However, beyond the 100-shot setting, the impact is either negative or marginal. We observe that **warmup correction (WC)** provides a decent performance improvement across all the few-shot settings (4.46% in 50-shot, 1.56% in 500-shot, etc.) in comparison to finetuning stability (FS) and half gradient accumulation (HGA), where their utility is only observed to lower data settings of 10, 50, and 100-shot. The number of warmup steps in standard pretraining setups is usually

Data per class	Model type	Prom-core Accuracy	Prom-core AUC	Prom-300 Accuracy	Prom-300 AUC	Cohn-enh Accuracy	Cohn-enh AUC
	DNABert as Base model						
10	ORI 120K (FS)	**0.606 ± 0.045**	**0.661 ± 0.064**	0.638 ± 0.070	0.708 ± 0.088	0.582 ± 0.030	0.631 ± 0.044
	ORI 10K (WC+FS)	0.586 ± 0.051	0.641 ± 0.082	0.601 ± 0.065	0.657 ± 0.095	0.579 ± 0.047	0.655 ± 0.058
	PMI-VOCAB	0.6 ± 0.047	0.652 ± 0.067	0.653 ± 0.065	0.731 ± 0.066	0.596 ± 0.048	0.66 ± 0.062
	PMI-best (WC+FS)	0.602 ± 0.058	0.655 ± 0.078	**0.676 ± 0.054**	**0.779 ± 0.074**	**0.622 ± 0.050**	**0.701 ± 0.038**
50	ORI 120K (FS)	**0.687 ± 0.024**	**0.756 ± 0.03**	0.808 ± 0.019	0.89 ± 0.013	0.638 ± 0.020	0.679 ± 0.028
	ORI 10K (WC+FS)	0.653 ± 0.058	0.718 ± 0.064	0.789 ± 0.059	0.882 ± 0.04	0.634 ± 0.031	0.689 ± 0.044
	PMI-VOCAB	0.649 ± 0.058	0.738 ± 0.033	0.800 ± 0.027	0.893 ± 0.02	0.645 ± 0.014	0.706 ± 0.016
	PMI-best (WC+FS)	0.678 ± 0.026	0.744 ± 0.026	**0.815 ± 0.02**	**0.905 ± 0.013**	**0.654 ± 0.017**	**0.713 ± 0.011**
100	ORI 120K (FS)	**0.712 ± 0.009**	**0.781 ± 0.012**	0.842 ± 0.014	0.915 ± 0.012	0.669 ± 0.017*	0.736 ± 0.022*
	ORI 10K (WC+FS)	0.695 ± 0.014	0.765 ± 0.017	0.842 ± 0.018	0.923 ± 0.009	0.668 ± 0.015*	0.736 ± 0.011*
	PMI-VOCAB	0.697 ± 0.011	0.767 ± 0.013	0.835 ± 0.017	0.912 ± 0.014	0.65 ± 0.051	0.737 ± 0.011
	PMI-best (WC+FS)	0.708 ± 0.013	0.779 ± 0.015	**0.847 ± 0.029**	**0.920 ± 0.020**	**0.67 ± 0.017***	**0.737 ± 0.013***
500	ORI 120K	0.743 ± 0.008	0.819 ± 0.007	0.883 ± 0.006	0.951 ± 0.005	**0.698 ± 0.009**	**0.776 ± 0.011**
	ORI 10K (WC)	0.752 ± 0.007	**0.831 ± 0.003**	0.888 ± 0.007	**0.958 ± 0.002**	0.696 ± 0.009	0.767 ± 0.008
	PMI-VOCAB	0.738 ± 0.021	0.82 ± 0.017	0.884 ± 0.004	0.948 ± 0.008	0.692 ± 0.011	0.759 ± 0.011
	PMI-best (WC)	**0.753 ± 0.005**	**0.831 ± 0.003**	**0.89 ± 0.006**	0.957 ± 0.002	**0.698 ± 0.006**	0.771 ± 0.007
1000	ORI 120K	0.758 ± 0.006	0.835 ± 0.004	0.895 ± 0.005	0.957 ± 0.005	0.700 ± 0.009	0.769 ± 0.009
	ORI 10K (WC)	0.765 ± 0.004	0.839 ± 0.005	**0.901 ± 0.003**	**0.964 ± 0.002**	0.705 ± 0.005	0.776 ± 0.006
	PMI-VOCAB	0.759 ± 0.007	0.834 ± 0.006	0.895 ± 0.004	0.96 ± 0.004	0.698 ± 0.007	0.766 ± 0.009
	PMI-best (WC)	**0.766 ± 0.007**	**0.843 ± 0.006**	0.898 ± 0.005	0.962 ± 0.002	**0.706 ± 0.005**	**0.778 ± 0.006**
	LOGO as Base model						
10	ORI (WC+FS)	0.506 ± 0.017	0.557 ± 0.064	0.502 ± 0.005	0.557 ± 0.028	0.53 ± 0.043	0.599 ± 0.067
	PMI (WC+FS)	**0.54 ± 0.049**	**0.582 ± 0.088**	**0.519 ± 0.04**	**0.594 ± 0.086**	**0.553 ± 0.063**	**0.649 ± 0.067**
50	ORI (WC+FS)	0.565 ± 0.048	0.635 ± 0.037	0.618 ± 0.037	0.663 ± 0.05	0.627 ± 0.007	0.676 ± 0.006
	PMI (WC+FS)	**0.611 ± 0.04**	**0.676 ± 0.017**	**0.668 ± 0.014**	**0.733 ± 0.022**	**0.635 ± 0.006**	**0.692 ± 0.009**
100	ORI (WC+FS)	0.628 ± 0.014	0.677 ± 0.017	0.646 ± 0.033	0.694 ± 0.039	0.638 ± 0.008	0.691± 0.011
	PMI (WC+FS)	**0.644 ± 0.008**	**0.695 ± 0.012**	**0.705 ± 0.023**	**0.783 ± 0.023**	**0.639 ± 0.006**	**0.695 ± 0.008**
500	ORI (WC)	**0.693 ± 0.010**	**0.754 ± 0.007**	0.751 ± 0.034	0.817 ± 0.037	0.629 ± 0.006	0.679 ± 0.007
	PMI (WC)	0.69 ± 0.006	0.748 ± 0.009	**0.835 ± 0.008**	**0.907 ± 0.008**	**0.648 ± 0.004**	**0.709 ± 0.04**
1000	ORI (WC)	0.705 ± 0.014	0.769 ± 0.013	0.808 ± 0.015	0.884 ± 0.016	0.652 ± 0.008	0.709 ± 0.008
	PMI (WC)	**0.723 ± 0.005**	**0.785 ± 0.006**	**0.855 ± 0.004**	**0.925 ± 0.004**	**0.660 ± 0.004**	**0.727 ± 0.004**

Table 1: Performance comparison at 10K pretraining steps. All values are rounded off to 3 decimal places. **ORI** refers to the SOTA model with random masking, and **PMI-best** refers to the best performance of two PMI-masking model variants: *PMI (WC+FS)*, *PMI (WC+HGA+FS)*. FS is not performed where * is marked and for 500 and 1000-shot settings due to model overfitting.

Data per class	Model type	Prom-core Accuracy	Prom-core AUC	Prom-300 Accuracy	Prom-300 AUC	Cohn-enh Accuracy	Cohn-enh AUC
10	PMI	0.520 ± 0.020	0.545 ± 0.035	0.506 ± 0.015	0.514 ± 0.021	0.526 ± 0.031	0.562 ± 0.048
	PMI (FS)	0.565 ± 0.037	0.596 ± 0.056	0.526 ± 0.021	0.556 ± 0.047	0.575 ± 0.032	0.62 ± 0.047
	PMI (WC + HGA + FS)	0.585 ± 0.065	0.631 ± 0.088	**0.676 ± 0.054**	**0.779 ± 0.074**	0.604 ± 0.068	0.640 ± 0.098
	PMI (WC + FS)	**0.602 ± 0.058**	**0.655 ± 0.078**	0.625 ± 0.09	0.717 ± 0.111	**0.622 ± 0.050**	**0.701 ± 0.038**
50	PMI	0.602 ± 0.037	0.659 ± 0.048	0.629 ± 0.062	0.714 ± 0.085	0.602 ± 0.029	0.656 ± 0.048
	PMI (FS)	0.647 ± 0.032	0.71 ± 0.039	0.751 ± 0.043	0.847 ± 0.041	0.618 ± 0.036	0.669 ± 0.046
	PMI (WC + HGA + FS)	0.677 ± 0.033	**0.744 ± 0.032**	**0.815 ± 0.02**	**0.905 ± 0.013**	**0.654 ± 0.017**	0.713 ± 0.011
	PMI (WC +FS)	**0.678 ± 0.026**	**0.744 ± 0.026**	0.781 ± 0.097	0.893 ± 0.023	0.648 ± 0.016	**0.718 ± 0.015**
100	PMI	0.617 ± 0.022	0.677 ± 0.027	0.588 ± 0.042	0.652 ± 0.046	0.654 ± 0.016	0.718 ± 0.018
	PMI (FS)	0.676 ± 0.015	0.751 ± 0.015	0.795 ± 0.027	0.889 ± 0.022	0.642 ± 0.049	0.694 ± 0.067
	PMI (WC + HGA + FS)	0.694 ± 0.043	0.759 ± 0.058	**0.847 ± 0.029**	0.920 ± 0.020	**0.67 ± 0.017***	**0.737 ± 0.013***
	PMI (WC + FS)	**0.708 ± 0.013**	**0.779 ± 0.015**	0.843 ± 0.027	0.925 ± 0.010	0.655 ± 0.052*	0.722 ± 0.068*
500	PMI	0.744 ± 0.008	0.826 ± 0.004	0.873 ± 0.017	0.949 ± 0.004	0.687 ± 0.006	0.755 ± 0.008
	PMI (FS)	0.735 ± 0.010	0.811 ± 0.007	0.871 ± 0.011	0.945 ± 0.003	0.538 ± 0.075	0.576 ± 0.108
	PMI (WC + HGA)	0.75 ± 0.008	0.829 ± 0.005	0.887 ± 0.004	0.953 ± 0.004	0.674 ± 0.058	0.75 ± 0.038
	PMI (WC)	**0.753 ± 0.005**	**0.831 ± 0.003**	**0.89 ± 0.006**	**0.957 ± 0.002**	**0.698 ± 0.006**	**0.771 ± 0.007**
1000	PMI	0.761 ± 0.004	0.837 ± 0.003	0.894 ± 0.003	0.959 ± 0.003	0.674 ± 0.058	0.717 ± 0.128
	PMI (FS)	0.673 ± 0.109	0.719 ± 0.112	0.498 ± 0.005	0.523 ± 0.064	0.589 ± 0.089	0.629 ± 0.099
	PMI (WC + HGA)	**0.766 ± 0.007**	**0.843 ± 0.006**	**0.898 ± 0.004**	0.961 ± 0.002	0.699 ± 0.008	0.771 ± 0.011
	PMI (WC)	**0.766 ± 0.004**	0.841 ± 0.005	**0.898 ± 0.005**	**0.962 ± 0.002**	**0.706 ± 0.005**	**0.778 ± 0.006**

Table 2: Ablation analysis of PMI-masking-guided DNABert model variants at 10000 steps. FS is not performed where * is marked and for 500 and 1000-shot settings due to model overfitting

kept between $5 - 10\%$ (known as *warmup percentage*) of the maximum number of training steps; In models without WC, the warmup percentage is almost 100%, indicating that the learning rate is slowly increasing throughout the entire training process. However, since we use mini-batch Gradient Descent, the model noisily converges towards minima and may oscillate far away from the actual minima. As a result, the model convergence is significantly delayed, negatively impacting both model optimization and generalization.

Dataset	Motifs (Consensus Logo)	Normalized PMI rank (out of 4096)
Prom-core	n**GAGGAGG**v	AGGAGG (56), GAGGAG (278)
Cohn-enh	n**CCTGGCC**h	CCTGGC (25), CTGGCC (129)
Prom-core	n**TATAAA**r	242
Cohn-enh	**GTGGCT**sw	126
Prom-core, Cohn-enh	nCyy**CCTCC**n*	1, 11, 52, 175, 186
Prom-core	sCw**GCAGC**n	259, 516, 540, 570, 628
Prom-core	n**TATAAA**r	242
Cohn-enh	ks**CTGGG**m	5, 17, 20, 21, 71
Cohn-enh	**TTTTTT**TTTn	8

Table 3: PMI-based rankings based on Normalized-PMI_n score for the motifs present in finetuning datasets. The motifs are of lengths 5, 6, or 7. For length 7, we mention two rankings considering two 6-length sub-motifs. A motif of length 5 matches as a sub-string to multiple 6-mers, we only mention the top five ranks for all such matches.

6.3 Motif Analysis

Motifs are repetitive units having a certain biological significance. We here check whether the correlated tokens identified by PMI are in fact (part of) such units.

Association between the top-20 ranked 6-mers based on Normalized-PMI_n score and conserved DNA sequence motifs. We analyze whether highly ranked PMI tokens resemble meaningful concepts by checking their overlap with known motifs. We performed a Google search with the following query template: *"AATCTC" DNA sequence motif*. Double quotes are used as a Google wildcard to indicate that the term AATCTC must always be present in the returned results. We only considered the first page of Google results to consider hits to the query. Among the *top 20 ranked* 6-mers, we observed that all except 2 (CCAGGC - rank 9, GCCTGG - rank 10) are previously mentioned in the published biomedical literature.

PMI-based rankings capture motifs present in finetuning datasets. We use the R package *rGADEM* (Droit et al., 2022) to perform de novo motif discovery. As motif discovery is computationally expensive, we only provide a subset of the data: randomly sampled 1000 (prom300 - 300 bp and enhancers-cohn 500 bp) and 2000 (prom-core - 70 bp) data points. We obtain a total of 12 motifs from the Prom-core and Cohn-enh datasets, which are of varying lengths but are mostly concentrated around lengths 5, 6, and 7. We present our best matches and their corresponding ranks in the 6-mer PMI ranked list (*RANK*) in Table 3 (see Table 6 for the complete list, as well as their corresponding consensus logos in Figures 3 and 4). We observe that most of the 6-mers that match the discovered motifs are ranked very high. Within the top 25 percentile of *RANK*, that is, a rank of 1024. We further observe that our top-1 ranked 6-mer is present in motifs of both enhancers and promoters based on de-novo motif discovery and is also mentioned in previous biomedical literature (Chow et al., 1991). The TATA box, a well-known motif for promoters (rows 3 of Table 3) is ranked at 242; the best TATA box motif is *TATATA*, which has a PMI rank of 15. We thus conclude that top-ranked 6-mers (top 25 percentile) have a strong correlation with DNA sequence motifs.

7 Conclusion

Gene sequence classification is a challenging problem and requires a tremendous engineering effort even to develop the experimental setup. The publicly available infrastructure currently in place needs to be improved to facilitate such research. We plan to open-source the pretraining and finetuning datasets and associated codes, along with model weights and checkpoints, with the objective of broader adoption by the research community. We believe this would be an important contribution to the work besides the critical finding that the PMI-masking strategy improves over the random masking strategy in all the few-shot settings for two SOTA models - DNABert and LOGO over three gene sequence classification datasets. We also performed a detailed motif analysis and observed a strong correlation between top-ranked PMI tokens and conserved DNA sequence motifs, providing us with a biological reason behind the improvement. Finally, given the often limited dataset sizes in (personalized) medicine, the present framework of *entity-aware pretraining leading to improvement in a few-shot setting* needs to be explored in full, which will be our immediate future work.

Ethics Statement

The gene sequence data used for both pretraining and finetuning is obtained from publicly available sources and can be obtained directly without signing any explicit data use agreement. The three benchmark datasets of Prom-core, Prom-300, and Cohn-enh are also used in previous studies for the task of gene sequence classification (Zaheer et al., 2020; Ji et al., 2021; Martinek et al., 2022; Oubounyt et al., 2019). Our work does not involve patient-level data for the experiments. We do not foresee any negative social impacts of this work, but of course, the accumulation of improvements in ML could be misused as it may give more power to nefarious agents.

Reproducibility Statement

We develop the Normalized-PMI_n metric and also incorporate it into SOTA models as PMI-masking. In terms of motif analysis, we also perform the de novo motif discovery on the downstream task datasets using the *rGADEM* R package. We will provide the code which includes the pretraining and finetuning data construction, together with a detailed reproducibility report (file name: *reproducibilty-report-iclr-2023*) in the supplementary material. We will be releasing the codebase and the pretrained models after the review process.

References

Dnabert github repository, 2021. URL https://github.com/jerryji1993/DNABERT.

Epd selection tool of epd eukaryotic promoter database, 2022. URL https://epd.epfl.ch/human/human_database.php?db=human.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, Oct 2021. ISSN 1548-7105. URL https://doi.org/10.1038/s41592-021-01252-x.

Sarkhan Badirli, Zeynep Akata, George Mohler, Christine Picard, and Mehmet M Dundar. Fine-grained zero-shot learning with dna as side information. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19352–19362, 2021. URL https://proceedings.neurips.cc/paper/2021/file/a18630ab1c3b9f14454cf70dc7114834-Paper.pdf.

B.K. Chow, V. Ting, F. Tufaro, and R.T. MacGillivray. Characterization of a novel liver-specific enhancer in the human prothrombin gene. *Journal of Biological Chemistry*, 266(28):18927–18933, 1991. ISSN 0021-9258. doi: https://doi.org/10.1016/S0021-9258(18)55152-8. URL https://www.sciencedirect.com/science/article/pii/S0021925818551528.

Dikla Cohn, Or Zuk, and Tommy Kaplan. Enhancer identification using transfer and adversarial deep learning of dna sequences. *bioRxiv*, 2018. doi: 10.1101/264200. URL https://www.biorxiv.org/content/early/2018/02/14/264200.

ENCODE Project Consortium. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

Genome Reference Consortium. Genome reference consortium human build 38 patch release 13 (grch38.p13), 2019. URL https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Original bert github repository, 2019. URL https://github.com/google-research/bert.

Patrik D'haeseleer. What are dna sequence motifs? *Nature Biotechnology*, 24(4):423–425, Apr 2006. ISSN 1546-1696. doi: 10.1038/nbt0406-423. URL https://doi.org/10.1038/nbt0406-423.

René Dreos, Giovanna Ambrosini, Rouayda Cavin Périer, and Philipp Bucher. EPD and EPDnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic Acids Research*, 41(D1):D157–D164, 11 2012. ISSN 0305-1048. doi: 10.1093/nar/gks1233. URL https://doi.org/10.1093/nar/gks1233.

Arnaud Droit, Raphael Gottardo, Gordon Robertson, and Leiping Li. *rGADEM: de novo motif discovery*, 2022. R package version 2.44.1.

Katarina Gresova, Vlastimil Martinek, David Cechak, Petr Simecek, and Panagiotis Alexiou. Genomic benchmarks: A collection of datasets for genomic sequence classification. *bioRxiv*, 2022. URL https://www.biorxiv.org/content/10.1101/2022.06.08.495248.

Eric B. Hekler, Predrag V. Klasnja, Guillaume Chevance, Natalie M. Golaszewski, Dana Michelle Lewis, and Ida Sim. Why we need a small data paradigm. *BMC Medicine*, 17, 2019.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/a5e0ff62be0b08456fc7f1e88812af3d-Paper.pdf.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana 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.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 01 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00300. URL https://doi.org/10.1162/tacl_a_00300.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Anshul Kundaje, Wouter Meuleman, Jason Ernst, Misha Bilenky, Angela Yen, Alireza Heravi-Moussavi, Pouya Kheradpour, Zhizhuo Zhang, Jianrong Wang, Michael J Ziller, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.

Trent Kyono, Fiona J. Gilbert, and Mihaela van der Schaar. Multi-view multi-task learning for improving autonomous mammogram diagnosis. In *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2019, 9-10 August 2019, Ann Arbor, Michigan, USA*, volume 106 of *Proceedings of Machine Learning Research*, pp. 571–591. PMLR, 2019. URL http://proceedings.mlr.press/v106/kyono19a.html.

Melissa J. Landrum, Jennifer M. Lee, George R. Riley, Wonhee Jang, Wendy S. Rubinstein, Deanna M. Church, and Donna R. Maglott. ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Research*, 42(D1):D980–D985, 11 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1113.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. {PMI}-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3Aoft6NWFej.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgz2aEKDr.

Vlastimil Martinek, David Cechak, Katarina Gresova, Panagiotis Alexiou, and Petr Simecek. Fine-tuning transformers for genomic tasks. *bioRxiv*, 2022. doi: 10.1101/2022.02.07.479412. URL https://www.biorxiv.org/content/early/2022/02/10/2022.02.07.479412.

Amy L. McGuire, Stacey Gabriel, et al. The road ahead in genetics and genomics. *Nature Reviews Genetics*, 21(10):581–596, Oct 2020. ISSN 1471-0064. doi: 10.1038/s41576-020-0272-6.

Shentong Mo, Xi Fu, Chenyang Hong, Yizhen Chen, Yuxuan Zheng, Xiangru Tang, Zhiqiang Shen, Eric P Xing, and Yanyan Lan. Multi-modal self-supervised pre-training for regulatory genome across cell types, 2021. URL https://arxiv.org/abs/2110.05231.

Florian Mock, Fleming Kretschmer, Anton Kriese, Sebastian Böcker, and Manja Marz. Bertax: taxonomic classification of dna sequences with deep neural networks. *BioRxiv*, 2021.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=nzpLWnVAyah.

Patrick Ng. dna2vec: Consistent vector representations of variable-length k-mers. *CoRR*, abs/1701.06279, 2017. URL http://arxiv.org/abs/1701.06279.

Ngoc Giang Nguyen, Vu Anh Tran, Duc Luu Ngo, Dau Phan, Favorisen Rosyking Lumbanraja, Mohammad Reza Faisal, Bahriddin Abapihi, Mamoru Kubo, Kenji Satou, et al. Dna sequence classification by convolutional neural network. *Journal of Biomedical Science and Engineering*, 9(05):280, 2016.

Ronald Wihal Oei, Hao Sen Andrew Fang, Wei-Ying Tan, Wynne Hsu, Mong-Li Lee, and Ngiap-Chuan Tan. Using domain knowledge and data-driven insights for patient similarity analytics. *Journal of Personalized Medicine*, 11(8), 2021. ISSN 2075-4426. doi: 10.3390/jpm11080699. URL https://www.mdpi.com/2075-4426/11/8/699.

Mhaned Oubounyt, Zakaria Louadi, Hilal Tayara, and Kil To Chong. Deepromoter: Robust promoter predictor using deep learning. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00286.

Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pp. 613–619, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775138.

Soumyadeep Roy, Sudip Chakraborty, Aishik Mandal, Gunjan Balde, Prakhar Sharma, Anandhavelu Natarajan, Megha Khosla, Shamik Sural, and Niloy Ganguly. Knowledge-aware neural networks for medical forum question classification. In *Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management*, CIKM '21, pp. 3398–3402, 2021. doi: 10.1145/3459637.3482128.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269, Online, April 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.20. URL https://aclanthology.org/2021.eacl-main.20.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2339–2352, Online, June 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.185. URL https://aclanthology.org/2021.naacl-main.185.

Valerie A. Schneider, Tina Graves-Lindsay, et al. Evaluation of grch38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *bioRxiv*, 2016. doi: 10.1101/072116.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1162. URL https://doi.org/10.18653/v1/p16-1162.

Torgyn Shaikhina and Natasha A. Khovanova. Handling limited datasets with neural networks in medical applications: A small-data approach. *Artif. Intell. Medicine*, 75:51–63, 2017. doi: 10.1016/j.artmed.2016.12.003. URL https://doi.org/10.1016/j.artmed.2016.12.003.

S. T. Sherry, M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, 01 2001. ISSN 0305-1048. doi: 10.1093/nar/29.1.308.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

Peter M. Visscher, Naomi R. Wray, Qian Zhang, Pamela Sklar, Mark I. McCarthy, Matthew A. Brown, and Jian Yang. 10 years of gwas discovery: Biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017. ISSN 0002-9297. doi: https://doi.org/10.1016/j.ajhg.2017.06.005. URL https://www.sciencedirect.com/science/article/pii/S0002929717302409.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/xiong20b.html.

Atsuki Yamaguchi, George Chrysostomou, Katerina Margatina, and Nikolaos Aletras. Frustratingly simple pretraining alternatives to masked language modeling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3116–3125, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.249. URL https://aclanthology.org/2021.emnlp-main.249.

Meng Yang, Haiping Huang, Lichao Huang, Nan Zhang, Jihong Wu, Huanming Yang, and Feng Mu. Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. *bioRxiv*, 2021. doi: 10.1101/2021.09.06.459087. URL https://www.biorxiv.org/content/early/2021/09/06/2021.09.06.459087.

Tianwei Yue and Haohan Wang. Deep learning for genomics: A concise overview, 2018a. URL https://arxiv.org/abs/1802.00810.

Tianwei Yue and Haohan Wang. Deep learning for genomics: A concise overview, 2018b.

Manzil Zaheer, Guru Guruganesh, et al. Big bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020. URL https://proceedings.neurips.cc/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

A Supplementary Text Material

In this section, we provide the supplementary material associated with the paper.

A.1 BACKGROUND

Importance of understanding gene regulatory code. The long strands of DNA found in the human chromosomes can be classified into genes and the genes, in turn, comprise *coding* and *non-coding* parts. A *coding* part encapsulates the information required for converting the nucleotide to a *protein*. These proteins are the building blocks of all tissues. These genes interact with the non-coding regions which perform gene regulation. *Promoters/Enhancers* speed up the process of coding, *inhibitors* slow down the reaction. These non-coding genes are called *gene regulatory elements*(Yue & Wang, 2018a). The non-coding regions, accounting for over 98% of the whole genome, implement significant yet largely unknown regulatory functions. Recent large consortia projects, including the ENCyclopedia of DNA Elements (ENCODE) (Consortium, 2012), Roadmap Epigenomics (Kundaje et al., 2015), and the Genomics of Gene Regulation (GGR), have produced a large number of experimental mapping readouts to help annotate non-coding genome in specific tissues or cell-lines. On the other hand, Genome-wide association studies (GWAS) have discovered that the vast majority ($> 90\%$) of associated genome loci for complex disease and traits fall in non-coding regions (Visscher et al., 2017).

MLM training. Yamaguchi et al. (2021) explore alternative pretraining tasks compared to MLM such as shuffled word detection, random word detection, manipulated word detection (Shuffle + Random), masked token type classification, and masked first character prediction. Here, we choose the original DNABert configuration of MLM without Next Sentence Prediction and experiment with multiple masked token selection strategies.

A.2 PARAMETER COMPARISON OF THE SOTA MODELS

Parameter	DNABert	LOGO
Hidden Size	768	256
Hidden Layers	12	2
Attention Heads	12	8
Per GPU train batch size	10	5
Hidden Dropout Probability	0.1	0
Attention Dropout Probability	0.1	0
Intermediate Size	3072	3072
Embedding Size	512	512

Table 4: Difference between parameters of DNABert and LOGO

Finetuning parameter configuration. In the current setup, the models are fine-tuned on task-specific data for 5 epochs with warmup percentage as 0.1, hidden dropout probability as 0.1, and weight decay as 0.01.

A.3 EXPERIMENTAL RESULTS

Sequence Length	Total tokens in vocabulary
1	5
2	17
3	65
4	212
5	533
6	1465
7	3829
8	10271
9	17537
10	6071

Table 5: Masking vocabulary statistics of the baseline model, PMI-VOCAB

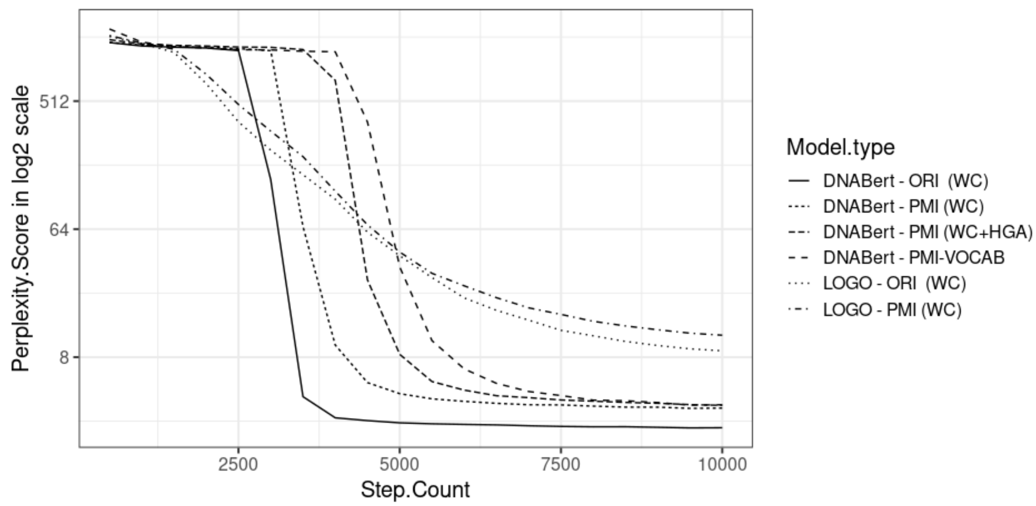

Figure 1: Perplexity score plot of pretrained models used in the experiments

A.3.1 Analyzing the effect of functional genetic variants

We aim to reproduce the variant analysis conducted by Ji et al. (2021), using dbSNP (Sherry et al., 2001) and ClinVar (Landrum et al., 2013), to compare the performance of PMI-masking-based model (PMI) with the original SOTA model (ORI) for the task of identifying functional genetic variants (using DNABert as the ORI model). $400,000$ variants are retrieved from dbSNP, and the corresponding genomic sequences (both original and mutated) are constructed by the original authors; this dataset is publicly available. When the original and mutated sequences offer significantly different prediction probabilities, the variant is queried in ClinVar to ascertain their importance. Since we do not have access to the specific code used by the authors to evaluate the importance of a given variant using Clinvar, we instead use the same finetuned DNABert model, pretrained for 120K steps and finetuned on the 10-shot Prom-core dataset (ORI 120K performs best in Prom-core low-resource setting) and obtain the original weights used by the authors (DNA, 2021)). We will look to evaluate settings other than 10-shot for the sake of completeness as a future research direction. As a result, DNABERT demonstrated its capability to capture and propose new and significant (disease-specific) variants in the future.

Experimental Details. The differences in prediction probabilities for the dataset mentioned above ($400,000$ data points) in the 10-shot setting and the data points are ranked in non-increasing order of the difference value. At this point, we obtain an individual ranked list of data points for the ORI and PMI models. We use the ranked (constructed similarly to ORI and PMI models) based on the finetuned DNABert model provided by the authors (described in the above paragraph) as ground truth, i.e., a proxy to identify all possible important functional variants. The degree of overlap with the ground-truth ranked list for multiple top-N settings for the ORI and PMI setting is provided in Figure 2. We observe that the PMI-guided DNABert model consistently reports higher overlap over the original model (ORI) in different top-N settings ($5000 \leq N \leq 50000$). Thus, we conclude that PMI-masking helps incorporate intrinsic (or relevant) genomics information into ORI models like DNABert and LOGO. The performance improvement is wider than just gene sequence classification tasks.

A.4 Motif Analysis

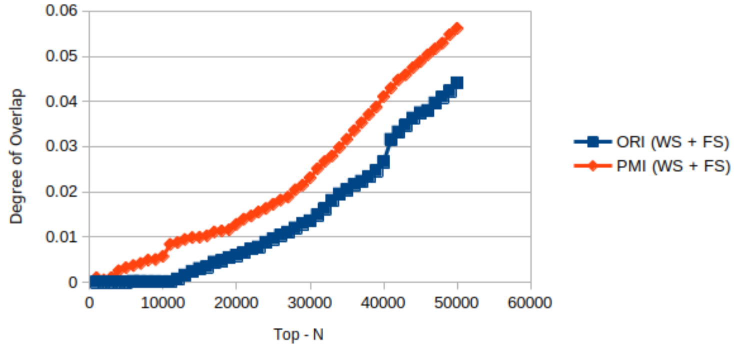

Figure 2: Performance comparison between "WS+FS" variant of ORI and PMI model in terms of overlap percentage for the task of analyzing the effect of functional genetic variants

Dataset	Motifs	Normalized PMI rank (Top 5 for 5-mers) (out of 4096)
Prom-core	nCyy**CCTCC**n*	1, 11, 52, 175, 186
Prom-core	sCs**CCGCC**sCCn	103, 1181, 1678, 2205, 2534
Prom-core	sCw**GCAGC**n	259, 516, 540, 570, 628
Prom-core	yy**TTTATA**n	286
Prom-core	n**TATAAA**r	242
Prom-core	n**GAGGAGG**v	AGGAGG (rank 56), GAGGAG (rank 278)
Prom-core	k**GCTGC**wGs	260, 510, 555, 590, 639
Cohn-enh	ks**CTGGG**m	5, 17, 20, 21, 71
Cohn-enh	n**CCTGGCC**h	CCTGGC (rank 25), CTGGCC (rank 129)
Cohn-enh	yy**CCAG**r**G**n	302, 593, 1247, 2778
Cohn-enh	**TTTTTT**TTTn	8
Cohn-enh	**GTGGCT**sw	126

Table 6: PMI-based rankings based on Normalized-PMI_n score for the motifs present in finetuning datasets. The motifs are of lengths 5, 6, or 7. For length 7, we mention two rankings considering two 6-length sub-motifs. A motif of length 5 matches as a sub-string to multiple 6-mers, we only mention the top five ranks for all such matches.

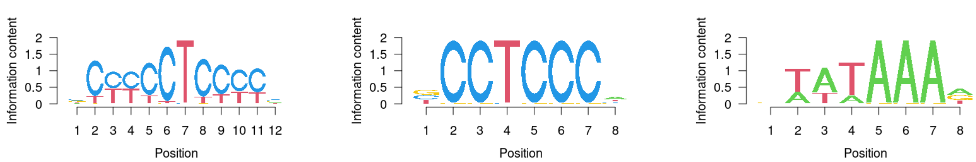

Figure 3: Consensus logo plot of motifs identified using de novo motif discovery tool. (left) nCyy-CCTCCyCn (middle) sCCTCCCw (right) nTATAAAr

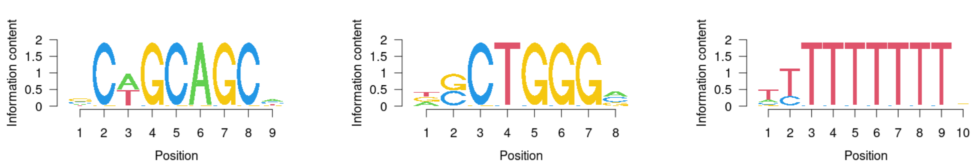

Figure 4: Consensus logo plot of motifs identified using de novo motif discovery tool. (left) sCwGCAGCm (middle) ksCTGGGm (right) TTTTTTTTTn