

SNP2Vec: Scalable Self-Supervised Pre-Training for Genome-Wide Association Study

Samuel Cahyawijaya^{1*}, Tiezheng Yu^{1*}, Zihan Liu^{1*}, Tiffany T.W. MAK^{2,3},
Xiaopu Zhou^{2,3,4}, Nancy Y. Ip^{2,3,4}, Pascale Fung¹

¹Center for Artificial Intelligence Research (CAiRE), Department of Electronic and Computer Engineering,
The Hong Kong University of Science and Technology, Hong Kong, China
{scahyawijaya, tyuah, zliucr, pascale}@ust.hk

²Division of Life Science, State Key Laboratory of Molecular Neuroscience, Molecular Neuroscience Center,
The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China
{tiffanytze, xpzhou, boip}@ust.hk

³Hong Kong Center for Neurodegenerative Diseases, Hong Kong Science Park, Hong Kong, China
{tiffanytze, xpzhou, boip}@ust.hk

⁴Guangdong Provincial Key Laboratory of Brain Science, Disease and Drug Development;
Shenzhen–Hong Kong Institute of Brain Science, HKUST Shenzhen Research Institute, Shenzhen, China
{xpzhou, boip}@ust.hk

Abstract

Self-supervised pre-training methods have brought remarkable breakthroughs in the understanding of text, image, and speech. Recent developments in genomics has also adopted these pre-training methods for genome understanding. However, they focus only on understanding haploid sequences, which hinders their applicability towards understanding genetic variations, also known as single nucleotide polymorphisms (SNPs), which is crucial for genome-wide association study. In this paper, we introduce SNP2Vec, a scalable self-supervised pre-training approach for understanding SNP. We apply SNP2Vec to perform long-sequence genomics modeling, and we evaluate the effectiveness of our approach on predicting Alzheimer’s disease risk in a Chinese cohort. Our approach significantly outperforms existing polygenic risk score methods and all other baselines, including the model that is trained entirely with haploid sequences. We release our code and dataset on <https://github.com/HLTCHKUST/snp2vec>.

1 Introduction

Self-supervised pre-training has become an indispensable step for almost all natural language processing (NLP) tasks (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019). Pre-trained language models, thanks to the usage of massive text corpora, are effective in handling data scarcity and generalizing to unseen examples (Brown et al., 2020;

Cahyawijaya et al., 2021; Wilie et al., 2020; Yu et al., 2021; Liu et al., 2021; Winata et al., 2021). Inspired by the success of pre-trained language models, pre-trained genomic models have been proposed to cope with genomic sequence prediction tasks (Zaheer et al., 2020; Ji et al., 2021). However, these models only focus on modeling the four nucleobases (i.e., A, T, C, and G), while ignoring genomic variations in the pre-training stage. Although they are effective in haploid pattern analysis, such as promoter region and chromatin-profile prediction, they fail to tackle more complex and challenging tasks, such as genome-wide association study (GWAS) (The Wellcome Trust Case Control Consortium, 2007; Corvin et al., 2010; Bush and Moore, 2012), which require an in-depth understanding of long genomic sequences and the genomic variation between a homologous chromosome pair.

To address these shortcomings, we introduce a self-supervised pre-training approach called SNP2Vec, which leverages the single-nucleotide polymorphism (SNP, pronounced ‘snip’) information gathered from a large-scale SNP database to inject genomic variations in the pre-training stage. SNP2Vec enables the model to learn the semantics of a diploid sequence (genotype) pattern in a diploid cell. We apply SNP2Vec to a linear-attention model, Linformer (Wang et al., 2020), to allow the model to encode long genomic sequences for Alzheimer’s disease risk prediction in a Chinese cohort. We compare SNP2Vec with non-pretrained models, as well as an existing strong baseline poly-

* These authors contributed equally.

genic risk scoring (PRS) model, to demonstrate the effectiveness of our approach.

Our contributions are summarized as follows:

- We are the first to introduce a scalable self-supervised pre-training approach (SNP2Vec) to learn genomic variations, which is popular for genome-wide association study.
- We demonstrate a method for modeling long diploid sequences with a length of >20,000 base pairs (bps) using an attention-based model within a single forward pass.
- We demonstrate the effectiveness of SNP2Vec, which significantly outperforms all the baselines, including a widely-used polygenic risk scoring (PRS) method, by 5-7% accuracy and AUROC for the Alzheimer’s disease prediction task in a Chinese elderly cohort.
- We conduct comprehensive analyses to show the effectiveness of SNP encoding and Byte Pair Encoding (BPE) tokenization compared to the other commonly used methods for genomics modeling.

2 Related Works

2.1 Genome-Wide Association Study

To this day, predicting the risk of hereditary diseases from a given genotype is done through genome-wide Association Study (GWAS) by applying a polygenic risk score (PRS). PRS utilizes GWAS data to identify important single nucleotide variations (SNVs) over a certain range from the gene of interest. The SNVs are first filtered according to a statistical measure to reduce the bias towards a certain population and the filtered SNVs are then used to build a classifier, which can be applied to a new genotype to determine the likelihood of getting the disease. This method has been applied by many works and has provided valuable insights for researchers to diseases including heart attack, diabetes, and different types of cancer (Lello et al., 2019). Moreover, PRS model has also been used in research and clinical practice for Alzheimer’s disease (Zhou et al., 2021). Nevertheless, all these methods fail to incorporate the patterns of the genomics sequence that determines the actual function. This is likely to lead the model towards non-representative bias, especially when the experimental data is small.

2.2 Statistical Modeling for Genomics

Tokenization in Genomics k -mer (synonymous to n -gram) tokenization is the most commonly used tokenization method in existing genome modelling works (Min et al., 2017; Shen et al., 2018). Gapped k -mer tokenization (Ghandi et al., 2014; Shrikumar et al., 2019) is a more efficient variant of k -mer tokenization by introducing the gap parameter L , which constitutes the stride between each k -mer window. However, the gapped k -mer approach will lead to the loss of some information when L is larger than k . In recent years, subword tokenization approaches (Sennrich et al., 2015; Kudo and Richardson, 2018) have also been explored in genomics (Zaheer et al., 2020).

Machine Learning in Genomics The support vector machine (SVM) is a traditional machine learning approach used to quickly and accurately interpret the nonlinear gapped k -mer (Shrikumar et al., 2019). Hill et al. (2018) leverage a deep recurrent neural network (RNN) to discover complex biological rules to decipher RNA protein-coding potential. Zhuang et al. (2019) incorporate convolutional neural network (CNN) to predict enhancer–promoter interactions with DNA sequence data. Shen et al. (2018) introduce a RNN to predict transcription factor binding sites. They treat each k -mer as a word and pre-train a word representation model though word2vec algorithm (Mikolov et al., 2013). Zaheer et al. (2020) propose BigBird and pre-train it on the human reference genome and improves the performance on downstream tasks.

2.3 Self-Supervised Pre-training

Recently, using self-supervised pre-training models on large scale unlabeled data and then fine-tuning them using a small amount of labeled data has become the norm in machine learning. BERT (Devlin et al., 2019) is a deep bidirectional transformer pre-trained on BooksCorpus (Zhu et al., 2015) (800M words) and English Wikipedia (2500M words) for language understanding. Liu et al. (2019) introduces Roberta, which has a similar architecture as BERT but trained on a much larger corpus (160GB of text) and consequently achieves better performance. In recent years, pre-training generative models (Radford et al., 2019; Raffel et al., 2019; Lewis et al., 2019) has significantly improved the performance of various language generation tasks such as machine translation, question answering, conversational AI, etc.

Self-supervised learning approaches have also been adopted in genomics (Zaheer et al., 2020; Ji et al., 2021) and proteomics (Madani et al., 2020; Elnaggar et al., 2021). These methods pre-train models using large-scale unlabelled datasets such as the human reference genome from the Genome Reference Consortium (GRC) (Church et al., 2011; Schneider et al., 2017) and protein sequence databases such as SWISS-PROT and TrEMBL (Boeckmann et al., 2003). In this paper, we focus on genomics and conduct the human reference genome for pre-training. Genomics data does not have the same structure as human languages; it has no known syntax or grammatical rules and it consists of very long sequences with only a number of differences between each human subject.

3 SNP2Vec

Existing pre-training methods in genomics, such as BigBird (Zaheer et al., 2020) and DNABERT (Ji et al., 2021), are only optimized to understand the pattern of a haploid sequence (haplotype) based on the reference genome. This hinders the model from learning genomic variations, which is essential for understanding traits in humans. In contrast to prior works in genomics pre-training, we develop SNP2Vec to enable pre-training for encoding and understanding patterns of genomic variations in a diploid sequence. Figure 1 depicts the overall structure of the SNP2Vec pre-training method. We elaborate on our SNP2Vec method in 3 subsections: 1) SNP Encoding, i.e., how we encode a diploid sequence as a sequence of SNP tokens; 2) Self-Supervised SNP Dataset, i.e., how we construct a self-supervised dataset using the SNP token; 3) Self-supervised SNP Pre-training, i.e., how we perform self-supervised pre-training for learning the sequence pattern of SNP tokens.

3.1 Preliminaries

What are haploid and diploid sequences? A diploid is a cell or organism that has paired chromosomes, one from each parent¹. Human cells are mostly diploid, except for the sex cells. In this sense, a diploid sequence (genotype) refers to a pair of homologous sequences (allele) inside the diploid chromosome, while a haploid sequence (haplotype) refers to the DNA sequence from the specific allele of the diploid sequence. The haploid sequence is

¹<https://www.genome.gov/genetics-glossary/Diploid>

suitable for understanding the regulatory function of a DNA pattern (Zhou and Troyanskaya, 2015; Ouyang et al., 2008), such as determining a binding site for a certain type of protein, as it provides the representation of the actual nucleotides. A diploid sequence, on the other hand, is more suitable for understanding the phenotype (Levy et al., 2007; Wang et al., 2008) over population since it allows understanding of the genomic variations between two homologous DNA sequences, which tells the dosage information and the gene expression level of a variation. These genomic variations are gathered by comparing them to a genome reference sequence, and they can be categorized based on its dosage, i.e., wild-type (normal), heterozygous, or homozygous, and based on their differences, i.e., substitution, insertion, and deletion. The depiction of haploids and diploids along with their variations is shown in Figure 2.

How do we get the haploid and diploid sequence? As most human cells are predominantly diploid, performing genome sequencing on such homologous chromosome pair will produce a diploid sequence rather than a haploid sequence, because the primer binds to both of the homologous regions from each chromosome (Ye et al., 2012). Extracting haploid sequences from a diploid sequence requires an additional step through an estimation process called phasing (Stephens et al., 2001). Despite their effectiveness, the quality of phasing methods (Browning and Browning, 2007, 2009; Patterson et al., 2014) is not perfect and tends to decrease significantly especially when the gap between the SNPs is large (Choi et al., 2018).

3.2 SNP Encoding

We first extend the nucleotide tokens from 5 token types ‘A’, ‘T’, ‘C’, ‘G’, and ‘N’ into 11 tokens by adding 6 insertion-deletion (indel) tokens ‘AI’, ‘TI’, ‘CI’, ‘GI’, ‘NI’, and ‘DEL’, where ‘XI’ token represents any insertion after the nucleotide ‘X’, and ‘DEL’ represents the nucleotide deletion. There can be many different possibility for insertion, e.g., a nucleotide ‘T’ can be inserted into “TG”, “TGGG”, or “TAAA”; therefore we aggregate all the insertions into a single token to reduce the sparsity of the indel representation as indel occurs relatively rarely compared to substitution, with an around 1:5 ratio (Chen et al., 2009). To encode a diploid sequence, we construct all the combinations with replacement (${}_{11}C_r^R$) of the 11 haploid

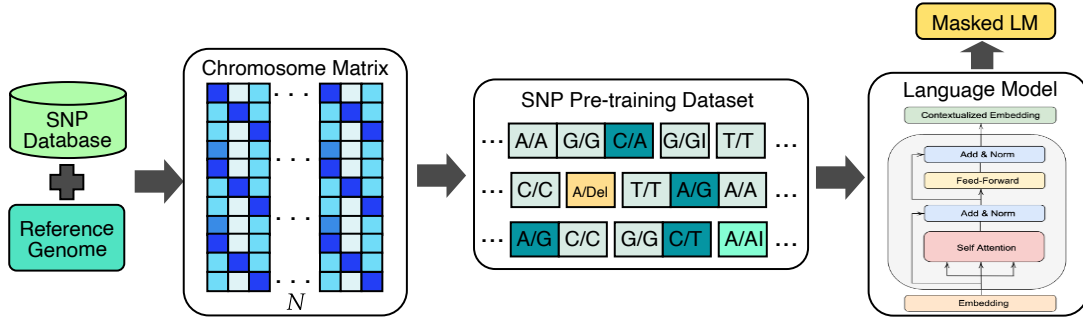


Figure 1: SNP2Vec Pre-training Pipeline. SNP2Vec merges information from reference genome and SNP database to form a chromosome matrix which is then utilized to construct SNP pre-training dataset following the SNP encoding’s token format. This pre-training dataset is employed to train a genome language model through the masked language modeling task.

Diploid Sequence	Wild-type			Substitution			Pattern		
	H1:	A	A	A	H1:	A		A	A
	H2:	A	A	A	H2:	A		C	A
	D:	A/A	A/A	A/A	D:	A/A		A/C	A/A
	Heterozygous			Insertion					
	H1:	A	A	A	H1:	A		AG	A
H2:	A	T	A	H2:	A	A	A		
D:	A/A	A/T	A/A	D:	A/A	A/AG	A/A		
Homozygous			Deletion						
H1:	A	G	A	H1:	A	-	A		
H2:	A	G	A	H2:	A	A	A		
D:	A/A	G/G	A-A	D:	A/A	A/Del	A/A		
Ref: A A A									

Figure 2: Diploid sequence variations. The box on the top-left shows the wild-type sequence, while others are its variations. **H1** and **H2** denote the haploid sequence for each parent allele. **D** represents the diploid sequence of the two alleles.

and indel tokens with $n = 11$ and $r = 2$, producing a total of 66 types of SNP tokens consisting of wild-type, heterozygous, and homozygous variation tokens. The resulting SNP tokens are represented as ‘ X_1/X_2 ’, where ‘ X_1 ’ and ‘ X_2 ’ denote aligned nucleotide or indel tokens from the two alleles ordered alphabetically. A depiction of the SNP tokens is shown in Figure 3. To reduce the size and facilitate more straightforward representation for downstream processes such as pre-processing, tokenization, and modeling, we map the SNP tokens into a single character representation. The mapping of the SNP token into a single character representation is shown in Appendix A.

By incorporating the SNP encoding, variant call-

ing information gathered from the DNA sequencing machine can be directly converted into a sequence of SNP tokens, that are then used for the model fine-tuning and inference. However, this is not directly applicable for self-supervised pre-training since DNA sequencing data is hard to obtain and it is unethical to share publicly as it contains very sensitive and personal information of the human subject. In the next section, we discuss in detail how we can construct an inexpensive and reliable pre-training dataset to perform self-supervised pre-training on the SNP tokens by utilizing publicly available genomics data sources.

3.3 Self-Supervised SNP Dataset

Prior self-supervised pre-training approaches in genomics (Zaheer et al., 2020; Ji et al., 2021) only utilize the human reference genome (Church et al., 2011; Schneider et al., 2017) as the unlabelled data for haploid genomics pre-training, the latter does not capture any genomic variations. We extend these haploid modeling techniques into a diploid modeling method, which allows the model to learn patterns of genomic variations by generating unlabelled pre-training data for learning SNP tokens. More specifically, we use the genome sequence from the human reference genome and genome variation from a large-scale SNP database, namely dbSNP (Smigielski et al., 2000), to generate the pre-training data.

Human reference genome The human reference genome is a genome sequence derived from the DNA collected from a number of people (Pollard et al., 2017), which was first released in 2000 and is periodically updated. There are two most commonly used versions of the human reference

Wild-Type & Homozygous Substitution	A/A	T/T	C/C	G/G	N/N
Heterozygous Substitution	A/T	A/C	...	C/T	C/G
Heterozygous Insertion	A/AI	T/TI	...	C/CI	G/GI
Heterozygous Deletion	A/Del	T/Del	...	C/Del	G/Del
Homozygous Indel	AI/AI	TI/TI	...	CI/CI	Del/Del

Figure 3: SNP tokens consist of a total of 66 types of token covering all possible variations in a diploid sequence including wild-types, heterozygous variations, and homozygous variations.

genome, namely GRCh37 (Church et al., 2011)² and GRCh38 (Schneider et al., 2017)³. A human reference genome consists of the genome sequence information for all human chromosomes with ~ 3 B sequence length in total. Most of the positions are mapped and represented as either ‘A’, ‘T’, ‘C’, or ‘G’, while the others are unmapped and flagged with the unknown (‘N’) token.

dbSNP dbSNP (Smigielski et al., 2000)⁴ is a central public repository of human SNPs. dbSNP covers a broad collection of simple genetic variations with a length of variation ≤ 50 bps long, which includes single-base nucleotide substitutions, small-scale multi-base deletions, and small-scale multi-base insertions. A single SNP in the dbSNP contains the following information: chromosome number, position in the chromosome, SNP identifier, reference sequence (REF), alternative sequence(s) (ALTS), probability of the REF and ALTS, and other metadata. The REF is a single-base or multi-base sequence that comes from the human reference genome used for detecting the SNPs. The ALTS can consist of one or more alternative variations and each can represent a substitution, a deletion, or an insertion.

Dataset Construction We construct a pre-training dataset consisting of sequences of SNP to-

²https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.13/

³https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.26/

⁴https://www.ncbi.nlm.nih.gov/projects/SNP/snp_summary.cgi

kens by combining the sequence information from the human reference genome and the genomic variations from the dbSNP. For each chromosome, we generate an $11 \times N$ matrix, where N is equal to the length of the corresponding chromosome and 11 represents the probability of each nucleotide and indel token. We name this matrix a **chromosome matrix**. We fill the chromosome matrix using all SNPs labelled as COMMON in the dbSNP by filling the corresponding matrix position with the REF and ALTS probability of the corresponding SNP record. Since the SNPs from the dbSNP do not cover all of the genome positions, we fill up all the other gap positions with a probability of 1 to the nucleotide token in the corresponding position on the human reference genome.

For constructing the self-supervised pre-training dataset, we closely follow the setup in the typical NLP pre-training dataset construction pipeline. Specifically, we convert the chromosome matrix into a set of segments S where each segment $s \in S$ comprises of a number of SNP tokens. To construct the sentences S , we sample multiple sequences from different positions of a chromosome. For each position in the sequence, we apply a sampling function F to collect ‘X1’ and ‘X2’ (the nucleotide or indel tokens on the corresponding position) and construct the SNP token “X1/X2”. The dataset construction method can be applied to all the chromosome pairs except for the sex chromosome, which is always haploid. The details of our dataset construction approach is shown in Algorithm 1.

3.4 Self-Supervised SNP Pre-training

Inspired by BERT (Devlin et al., 2019), SNP2Vec is trained using the masked language modeling (MLM) objective using a transformer-based model (Vaswani et al., 2017). The goal of MLM is to predict the representations of the masked tokens given their neighbouring sequence as the context. As complex genomic tasks, such as disease risk prediction, require the understanding long-genome sequence (>1000 bps), we apply two methods to process long input sequences. First, we apply a transformer variant with a linear-attention mechanism, which enables the model to reduce the computational complexity from $O(N^2)$ to $O(N)$. Second, we apply a BPE tokenization (Sennrich et al., 2015) to encode the sequence of SNP tokens to compress the sequence via aggregation of neighbouring tokens. Unlike k-mer (Min et al., 2017; Ji

Algorithm 1 Self-Supervised Pre-training dataset construction for diploid SNP Encoding

Require: C : chromosome matrix

Require: f : SNP sampling function

Require: T : number of iterations

Require: K : start position threshold

Require: L^{inf} : lower bound of segment length

Require: L^{sup} : upper bound of segment length

```
1: Initialize  $S = \emptyset$ 
2:  $P =$  sample  $T$  positions from range  $[0 \dots K]$ 
3: for all  $p \in P$  do
4:   while  $p < |C|$  do
5:      $l \sim \mathcal{U}(L^{inf}, L^{sup})$ 
6:      $z =$  segment from  $p$  to  $p + L$  in  $C$ 
7:      $s =$  Sample SNP tokens using  $f$  from
       each position in  $z$ 
8:      $S = S \cup s$ 
9:      $p = p + l$ 
10:  end while
11: end for
```

et al., 2021) and gapped k-mer (Ghandi et al., 2014; Shrikumar et al., 2019) tokenizations, BPE tokenization can merge dynamic-length tokens based on their co-occurrences efficiently without losing any information.

4 Experiment Settings

4.1 Dataset

For building the pre-training data, we utilize GRCh37 as the human reference genome and dbSNP version 153⁵ as the SNP database. We utilize a weighted random sampling based on the probability of SNPs on the corresponding position as the sampling function f . For the downstream-task, we construct a dataset of genome sequences for predicting late-onset Alzheimer’s disease (LOAD) (Rabinovici, 2019) on a Chinese Cohort from 624 Hong Kong elderly with a minimum age of 65. The subjects are diagnosed with Alzheimer’s by a medical professional through the Montreal Cognitive Assessment (MoCA) test (Nasreddine et al., 2005) adjusted for the demographic information. Out of 624 subjects, 384 are labelled as Alzheimer’s disease carriers (ADs) and 240 are labelled as non-carriers (NCs). For the genome sequence, we collect sequencing data from the APOE region located in chromosome 19 from each subject, which is known

⁵<https://ftp.ncbi.nih.gov/snp/archive/b153/00readme.txt>

to be highly correlated with Alzheimer’s disease in the Chinese cohort (Zhou et al., 2019, 2020). We use BWA-MEM (Li, 2013) assembler to align the sequencing data with the human reference genome.

4.2 Training and Evaluation Setting

For our experiment, we build a BPE tokenizer with a vocabulary size of 32,000 tokens. We pre-train a 6-layers linear-attention transformer-based model, Linformer (Wang et al., 2020), using a maximum sequence length of 4,096 tokens, a sequence projection length k of 128 tokens, and a model dimension size of 512. For simplicity, we refer to our pre-trained SNP2Vec model as **Dipformer**. The detail hyperparameters of the BPE tokenizer and the Dipformer model are described in Appendix B. We run MLM pre-training for 200,000 steps with a 15% token replacement rate, where we replace with [MASK] 80% of the time, replace with a random token 10% of the time, and keep the token as is 10% of the time. More detail about the pre-training hyperparameter setting is shown in Appendix C.

For the fine-tuning, we apply SNP encoding to the sequencing data, apply BPE tokenization, and add a [CLS] token as the prefix of the sequence to gather the sequence representation for predicting the risk of having Alzheimer’s disease. We apply fine-tuning for three input sequence length settings, i.e., only APOE gene with 3,611 bps (**APOE only**), APOE with additional 5,000 bps upstream and downstream (**APOE+10k**), and APOE with additional 10,000 bps upstream and downstream (**APOE + 20k**). For each experiment, we apply 10-fold cross validation to ensure the result is significance. We evaluate the model performance using three evaluation metrics: accuracy, area under the ROC curve (AUROC), and area under the precision-recall curve (AUPRC). More detail about the fine-tuning setup is described in Appendix D.

4.3 Baselines

To evaluate the effectiveness of the SNP encoding, we build two different deep learning models using haploid token representation. First, we incorporate DeepSEA (Zhou and Troyanskaya, 2015), a CNN-based model develop for short sequence chromatin profiling tasks (~ 200 -1000 bps), and then we build another Linformer model pre-trained with the human reference genome using haploid tokens, called **Hapformer**. For the haploid token fine-tuning, we generate the haploid sequence from the aligned sequencing data. We generate the variant calling

Models	Acc	AUROC	AUPRC
DeepSEA	0.591	0.579	0.703
PLINK PRS	0.592	0.607	0.705
Hapformer	0.572	0.615	0.715
Dipformer	0.643	0.673	0.734

Table 1: Results of our model and baselines. We refer the pre-trained SNP2Vec model as Dipformer.

data with GATK HaplotypeCaller (McKenna et al., 2010; DePristo et al., 2011) and apply phasing with Beagle (Browning and Browning, 2007, 2009). During fine-tuning, we feed each haploid sequence to the model and fuse the representation using a linear transformation. We also incorporate a logistic regression model from PLINK (Purcell et al., 2007), which is a widely used approach for PRS.

5 Results

The results of our model and baselines are shown in Table 1. We find that Dipformer is able to outperform existing strong baselines, such as DeepSEA and PLINK PRS, by a large margin. This confirms the effectiveness of our SNP2Vec pre-training, and the ability of our Dipformer to capture relevant features for AD prediction. Interestingly, Hapformer, which leverages large amounts of genomic sequences for pre-training, only performs comparably to DeepSEA and PLINK PRS. Moreover, by learning genomic variations in a diploid sequence during the pre-training, Dipformer significantly outperforms Hapformer with an around 5-7% improvement in terms of accuracy and AUROC metrics. This shows that simply using an enormous amount of pre-training data might not necessarily improve the AD prediction, and an effective genomics pre-training approach is essential to guarantee full use of the unlabelled genomics data. More detail on our results is shown in Appendix E.

6 Discussion

6.1 Effect of Different Tokenization Methods in Genomics

In this section, we study different tokenization methods for genome modeling, and explore their effectiveness in terms of capturing genomic patterns and features. We compare BPE tokenization with other common methods, such as k -mer and gapped k -mer (gkm) with various gap parameters. To achieve this, we conduct experiments on the chromatin profiling dataset from DeepSEA, which con-

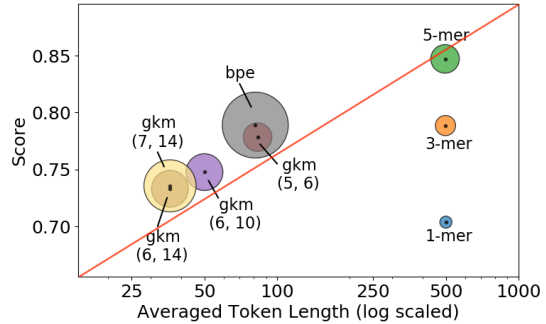


Figure 6: Performance efficiency trade-off of using different tokenization approaches. The score is averaged over the three models (Linear, CNN, and Transformer). The size of the dots represents the vocabulary size of the tokenization method.

sists of 4,863,024 chromatin profiles (4,400,000 training, 8000 validation, and 455,024 test) with 919 labels (690 transcription factor (TF) binding sites, 125 DNase marks, and 104 Histone marks). Three different models are incorporated in this experiment: a linear model with bag-of-words (BoW) representation, a CNN-based model following the DeepSEA architecture, and a transformer model. The models need to predict the TF, DNase, and Histone labels based on the input sequences using various tokenization methods. Hence, for the same model, a more effective tokenization method will lead to a higher prediction accuracy. Additionally, we use the average length of the tokenized sequences to measure the efficiency of different tokenization methods as it determines the input size for the model.

Table 2 provides the effectiveness and averaged token length of different tokenization methods in genome modeling. We find that, on the Linear BoW model, BPE significantly outperforms all other methods except 5-mer. On the CNN model, BPE remarkably surpasses all gapped k -mer methods except for the gkm (5,6). On the Transformer model, BPE performs similarly to 3-mer and gkm (5,6), and significantly outperforms 1-mer and other gkm methods. Moreover, in terms of the averaged score across all three models, BPE performs comparably well to 3-mer, and remarkably outperforms 1-mer and all gkm methods.

Figure 6 illustrates the trade-off between the performance and efficiency of different tokenization methods. We can see that compared to k -mer methods, BPE performs comparably to 3-mer and slightly worse than 5-mer, but it is much more ef-

Tokenization	1-mer	3-mer	5-mer	gkm (5,6)	gkm (6,10)	gkm (6,14)	gkm (7,14)	BPE
Avg. Token Length	500	498	496	83	50	36	36	81.19
BoW Linear	<u>0.499</u>	<u>0.698</u>	0.817	<u>0.771</u>	<u>0.759</u>	<u>0.749</u>	<u>0.753</u>	0.783*
CNN (DeepSEA)	0.890	0.903	0.898	0.808	<u>0.764</u>	<u>0.749</u>	<u>0.751</u>	0.811*
Transformer	<u>0.727</u>	0.788	0.825	0.785	<u>0.771</u>	<u>0.761</u>	<u>0.762</u>	0.789*
Average	<u>0.706</u>	0.796	0.847	<u>0.788</u>	<u>0.765</u>	<u>0.753</u>	<u>0.755</u>	0.795*

Table 2: Comparison of different tokenization methods in genome modeling (numbers denote the accuracy score), where gkm (k,l) denotes the gapped k -mer tokenization with the gap parameter l constituting the stride between each k -mer window. * denotes that BPE significantly outperforms the underlined baselines with a p-value < 0.01 .

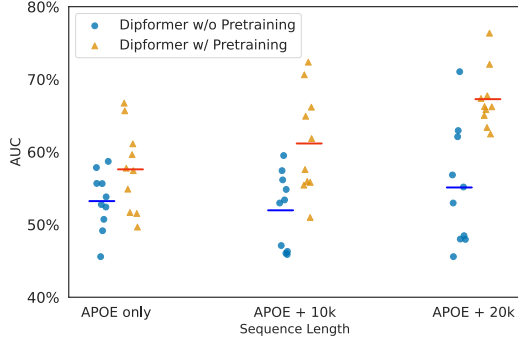


Figure 4: 10-folds AUROC Performance of Dipformer with and without pre-training on the Alzheimer's disease risk prediction over different sequence length input.

efficient due to a much shorter average length. In addition, BPE remarkably outperforms gkm methods with comparable or slightly worse efficiency. Furthermore, from the size of the dots, we can see that BPE has a much larger vocabulary size compared to other methods, which indicates that BPE can potentially capture richer genomics patterns.

6.2 Effect of Pre-training for Disease Risk Prediction

In this section, we focus on exploring the effectiveness of pre-training for disease risk prediction. Figure 4 illustrates the 10-fold AUROC results of our Dipformer model with and without pre-training on Alzheimer's disease risk prediction. The dashes in the figure represent the average AUROC for all 10-fold results. As shown in Figure 4, the average AUROC scores for pre-trained Dipformer significantly outperform the Dipformer without pre-training in all sequence length settings, APOE + 10k, and APOE + 20. Table 3 presents the quantitative results with additional metrics. The accuracy, AUROC, and AUPRC scores of pre-trained Dipformer consistently outperform the non-pre-trained Dipformer in all sequence length settings. By increasing the sequence length, the non-pre-

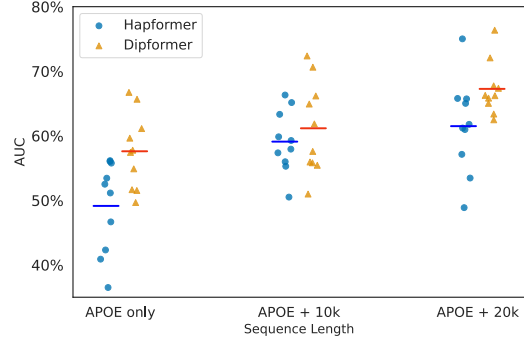


Figure 5: 10-folds AUROC performance of pre-trained Hapformer and Dipformer on the Alzheimer's disease risk prediction over different sequence length input.

trained Dipformer performs slightly better, while the pre-trained Dipformer improves by a large margin. This shows the importance of pre-training for understanding long-sequence features.

6.3 Effect of the SNP Encoding in Genomics

To study the effect of the SNP encoding, we pre-train and fine-tune a model with the same genomics data but using haploid tokens called Hapformer, as mentioned in the Section 4. Figure 5 shows the 10-fold AUROC results of pre-trained Hapformer and Dipformer on the AD risk prediction over different sequence length inputs. Among all three sequence length settings, Dipformer achieves better average AUROC scores than Hapformer with a p-value of 0.046 for the APOE + 20k setting, which indicates that the improvement of SNP encoding is significant. Meanwhile, the results in Table 3 shows that Dipformer also surpasses Hapformer in all other evaluation metrics. In addition, we also observe that both Hapformer and Dipformer achieve better results when the input sequence is longer. This shows that employing long sequence is essential for handling complex genomics tasks such as disease risk prediction.

Model	Accuracy	AUROC	AUPRC
<i>Without Pre-training</i>			
Dipformer (APOE only)	0.567	0.532	0.667
Dipformer (APOE + 10k)	0.571	0.520	0.608
Dipformer (APOE + 20k)	0.588	0.551	0.668
<i>With Pre-training</i>			
Hapformer (APOE only)	0.524	0.491	0.623
Hapformer (APOE + 10k)	0.565	0.591	0.705
Hapformer (APOE + 20k)	0.572	0.615	0.715
Dipformer (APOE only)	0.611	0.576	0.687
Dipformer (APOE + 10k)	0.574	0.612	0.710
Dipformer (APOE + 20k)	0.643	0.673	0.734

Table 3: Performance of Dipformer and Hapformer on the Alzheimer’s disease risk prediction over different lengths of the input sequences.

7 Conclusion

In this paper, we introduce SNP2Vec, a self-supervised pre-training method for understanding genomic variations in a diploid sequence. Unlike prior methods in genomics, SNP2Vec represents each genomics position with a SNP token which allows the model to capture genomic variations which is suitable for understanding complex genomics prediction tasks such as predicting phenotype. By utilizing SNP2Vec, we pre-train a Linformer model called Dipformer and evaluate it for predicting late-onset Alzheimer’s disease risk in a Chinese cohort. Experimental results suggest that Dipformer significantly improves the prediction quality by 5-7% Accuracy and AUROC over all other baselines including the widely used polygenic risk score model from PLINK, the haploid-variant of Dipformer, and a CNN-based genomics model called DeepSEA.

8 Future Work

For future works, we expect to focus on model explainability by using multiple analysis methods, such as analyzing the attention behaviour, analyzing the gradient saliency map, etc, to gather and verify insights from the model. Evaluation on larger scale dataset is also necessary to further demonstrate the effectiveness of SNP2Vec. Additionally, adoption of SNP2Vec to other hereditary disorders and other complex genomics tasks is also an essential direction for future works.

Acknowledgement

This work has been partially funded by School of Engineering PhD Fellowship Award, the Hong Kong University of Science and Technology and PF20-43679 Hong Kong PhD Fellowship Scheme, Research Grant Council, Hong Kong.

References

- Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J. Martin, Karine Michoud, Claire O’Donovan, Isabelle Phan, Sandrine Pilbout, and Michel Schneider. 2003. [The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003](#). *Nucleic Acids Research*, 31(1):365–370.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Brian L. Browning and Sharon R. Browning. 2009. [A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals](#). *The American Journal of Human Genetics*, 84(2):210–223.
- Sharon R. Browning and Brian L. Browning. 2007. [Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering](#). *The American Journal of Human Genetics*, 81(5):1084–1097.
- William S Bush and Jason H Moore. 2012. Chapter 11: Genome-wide association studies. *PLoS Comput. Biol.*, 8(12):e1002822.
- Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Khodra, Ayu Purwarianti, and Pascale Fung. 2021. [IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8875–8898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jian-Qun Chen, Ying Wu, Haiwang Yang, Joy Bergelson, Martin Kreitman, and Dacheng Tian. 2009. [Variation in the ratio of nucleotide substitution and in-](#)

- del rates across genomes in mammals and bacteria. *Molecular biology and evolution*, 26:1523–31.
- Yongwook Choi, Agnes P. Chan, Ewen Kirkness, Amalio Telenti, and Nicholas J. Schork. 2018. Comparison of phasing strategies for whole human genomes. *PLOS Genetics*, 14(4):1–26.
- Deanna M. Church, Valerie A. Schneider, Tina Graves, Katherine Auger, Fiona Cunningham, Nathan Bouk, Hsiu-Chuan Chen, Richa Agarwala, William M. McLaren, Graham R.S. Ritchie, Derek Albracht, Milinn Kremitzki, Susan Rock, Holland Kotkiewicz, Colin Kremitzki, Aye Wollam, Lee Trani, Lucinda Fulton, Robert Fulton, Lucy Matthews, Siobhan Whitehead, Will Chow, James Torrance, Matthew Dunn, Glenn Harden, Glen Threadgold, Jonathan Wood, Joanna Collins, Paul Heath, Guy Griffiths, Sarah Pelan, Darren Grafham, Evan E. Eichler, George Weinstock, Elaine R. Mardis, Richard K. Wilson, Kerstin Howe, Paul Flicek, and Tim Hubbard. 2011. Modernizing reference genome assemblies. *PLoS Biology*, 9(7):e1001091.
- A Corvin, N Craddock, and P F Sullivan. 2010. Genome-wide association studies: a primer. *Psychol. Med.*, 40(7):1063–1077.
- Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo del Angel, Manuel A Rivas, Matt Hanna, Aaron McKenna, Tim J Fennell, Andrew M Kernysky, Andrey Y Sivachenko, Kristian Cibulskis, Stacey B Gabriel, David Altshuler, and Mark J Daly. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, 43(5):491–498.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost. 2021. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1.
- Mahmoud Ghandi, Dongwon Lee, Morteza Mohammad-Noori, and Michael A Beer. 2014. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS computational biology*, 10(7):e1003711.
- Steven T Hill, Rachael Kuintzle, Amy Teegarden, Erich Merrill III, Padideh Danaee, and David A Hendrix. 2018. A deep recurrent neural network discovers complex biological rules to decipher rna protein-coding potential. *Nucleic acids research*, 46(16):8105–8113.
- Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. 2021. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120.
- Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.
- Louis Lello, Timothy G Raben, Soke Yuen Yong, Laurent CAM Tellier, and Stephen DH Hsu. 2019. Genomic prediction of 16 complex disease risks including heart attack, diabetes, breast and prostate cancer. *Scientific reports*, 9(1):1–16.
- Samuel Levy, Granger Sutton, Pauline C Ng, Lars Feuk, Aaron L Halpern, Brian P Walenz, Nelson Axelrod, Jiaqi Huang, Ewen F Kirkness, Gennady Denisov, Yuan Lin, Jeffrey R MacDonald, Andy Wing Chun Pang, Mary Shago, Timothy B Stockwell, Alexia Tsiamouri, Vineet Bafna, Vikas Bansal, Saul A Kravitz, Dana A Busam, Karen Y Beeson, Tina C McIntosh, Karin A Remington, Josep F Abril, John Gill, Jon Borman, Yu-Hui Rogers, Marvin E Frazier, Stephen W Scherer, Robert L Strausberg, and J. Craig Venter. 2007. The diploid genome sequence of an individual human. *PLoS Biology*, 5(10):1–32.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Heng Li. 2013. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Zhiwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13452–13460.
- Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. 2020. Progen: Language modeling for protein generation.
- A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernysky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo. 2010. The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9):1297–1303.

- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Xu Min, Wanwen Zeng, Ning Chen, Ting Chen, and Rui Jiang. 2017. Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding. *Bioinformatics*, 33(14):i92–i101.
- Ziad S. Nasreddine, Natalie A. Phillips, ValÃ©rie BÃ©dirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L. Cummings, and Howard Chertkow. 2005. The montreal cognitive assessment, MoCA: A brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699.
- Ching Ouyang, David D Smith, and Theodore G Krontiris. 2008. Evolutionary signatures of common human cis-regulatory haplotypes. *PLoS One*, 3(10):e3362.
- Murray Patterson, Tobias Marschall, Nadia Pisanti, Leo van Iersel, Leen Stougie, Gunnar W. Klau, and Alexander SchÃ¶nhuth. 2014. Whatshap: Haplotype assembly for future-generation sequencing reads. In *Research in Computational Molecular Biology*, pages 237–249, Cham. Springer International Publishing.
- Thomas D. Pollard, William C. Earnshaw, Jennifer Lippincott-Schwartz, and Graham T. Johnson. 2017. Chapter 7 - chromosome organization. In *Cell Biology (Third Edition)*, third edition edition, pages 107–122. Elsevier.
- Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A R Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul I W de Bakker, Mark J Daly, and Pak C Sham. 2007. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, 81(3):559–575.
- Gil D. Rabinovici. 2019. Late-onset alzheimer disease. *Continuum (Minneapolis, Minn.)*, 25(1):14–33.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Valerie A Schneider, Tina Graves-Lindsay, Kerstin Howe, Nathan Bouk, Hsiu-Chuan Chen, Paul A Kitts, Terence D Murphy, Kim D Pruitt, FranÃ§oise Thibaud-Nissen, Derek Albracht, Robert S Fulton, Milinn Kremitzki, Vincent Magrini, Chris Markovic, Sean McGrath, Karyn Meltz Steinberg, Kate Auger, William Chow, Joanna Collins, Glenn Harden, Timothy Hubbard, Sarah Pelan, Jared T Simpson, Glen Threadgold, James Torrance, Jonathan M Wood, Laura Clarke, Sergey Koren, Matthew Boitano, Paul Peluso, Heng Li, Chen-Shan Chin, Adam M Phillippy, Richard Durbin, Richard K Wilson, Paul Flicek, Evan E Eichler, and Deanna M Church. 2017. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Genome Res.*, 27(5):849–864.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Zhen Shen, Wenzheng Bao, and De-Shuang Huang. 2018. Recurrent neural network for predicting transcription factor binding sites. *Scientific reports*, 8(1):1–10.
- Avanti Shrikumar, Eva Prakash, and Anshul Kundaje. 2019. Gkmexplain: fast and accurate interpretation of nonlinear gapped k-mer svms. *Bioinformatics*, 35(14):i173–i182.
- E M Smigielski, K Sirotkin, M Ward, and S T Sherry. 2000. dbSNP: a database of single nucleotide polymorphisms. *Nucleic Acids Res.*, 28(1):352–355.
- Matthew Stephens, Nicholas J. Smith, and Peter Donnelly. 2001. A new statistical method for haplotype reconstruction from population data. *The American Journal of Human Genetics*, 68(4):978–989.
- The Wellcome Trust Case Control Consortium. 2007. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Jun Wang, Wei Wang, Ruiqiang Li, Yingrui Li, Geng Tian, Laurie Goodman, Wei Fan, Junqing Zhang, Jun Li, Juanbin Zhang, Yiran Guo, Binxiao Feng, Heng Li, Yao Lu, Xiaodong Fang, Huiqing Liang, Zhenglin Du, Dong Li, Yiqing Zhao, Yujie Hu, Zhenzhen Yang, Hancheng Zheng, Ines Hellmann, Michael Inouye, John Pool, Xin Yi, Jing Zhao, Jinjie Duan, Yan Zhou, Junjie Qin, Lijia Ma, Guoqing Li, Zhentao Yang, Guojie Zhang, Bin Yang, Chang Yu, Fang Liang, Wenjie Li, Shaochuan Li, Dawei Li, Peixiang Ni, Jue Ruan, Qibin Li, Hongmei Zhu, Dongyuan Liu, Zhike Lu, Ning Li, Guangwu Guo, Jianguo Zhang, Jia Ye, Lin Fang, Qin Hao, Quan Chen, Yu Liang, Yeyang Su, A San, Cuo Ping, Shuang Yang, Fang Chen, Li Li, Ke Zhou, Hongkun Zheng, Yuanyuan Ren, Ling Yang, Yang Gao, Guohua Yang, Zhuo Li, Xiaoli Feng, Karsten Kristiansen, Gane Ka-Shu Wong, Rasmus Nielsen, Richard Durbin, Lars Bolund, Xiuqing Zhang, Songgang Li, Huanming Yang, and Jian Wang. 2008. The diploid genome sequence of an asian individual. *Nature*, 456(7218):60–65.

- Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.
- Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, X. Li, Zhi Yuan Lim, S. Soleman, R. Mahendra, Pascale Fung, Syafri Bahar, and A. Purwarianti. 2020. Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.
- Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. [Are multilingual models effective in code-switching?](#) In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 142–153, Online. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.
- Jian Ye, George Coulouris, Irena Zaretskaya, Ioana Cutcutache, Steve Rozen, and Thomas L Madden. 2012. Primer-BLAST: a tool to design target-specific primers for polymerase chain reaction. *BMC Bioinformatics*, 13(1):134.
- Tiezheng Yu, Zihan Liu, and Pascale Fung. 2021. Adaptsum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.
- Jian Zhou and Olga G Troyanskaya. 2015. [Predicting effects of noncoding variants with deep learning-based sequence model.](#) *Nature Methods*, 12(10):931–934.
- Xiaopu Zhou, , Yu Chen, Kin Y. Mok, Timothy C. Y. Kwok, Vincent C. T. Mok, Qihao Guo, Fanny C. Ip, Yuewen Chen, Nandita Mullapudi, Paola Giusti-Rodríguez, Patrick F. Sullivan, John Hardy, Amy K. Y. Fu, Yun Li, and Nancy Y. Ip. 2019. [Non-coding variability at the APOE locus contributes to the alzheimer’s risk.](#) *Nature Communications*, 10(1).
- Xiaopu Zhou, Yu Chen, Fanny C F Ip, Nicole C H Lai, Yolanda Y T Li, Yuanbing Jiang, Huan Zhong, Yuewen Chen, Yulin Zhang, Shuangshuang Ma, Ronnie M N Lo, Kit Cheung, Estella P S Tong, Ho Ko, Maryam Shoai, Kin Y Mok, John Hardy, Vincent C T Mok, Timothy C Y Kwok, Amy K Y Fu, and Nancy Y Ip. 2020. Genetic and polygenic risk score analysis for alzheimer’s disease in the chinese population. *Alzheimers Dement. (Amst.)*, 12(1):e12074.
- Xiaopu Zhou, Yolanda YT Li, Amy KY Fu, and Nancy Y Ip. 2021. Polygenic score models for alzheimer’s disease: From research to clinical applications. *Frontiers in Neuroscience*, 15:346.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.
- Zhong Zhuang, Xiaotong Shen, and Wei Pan. 2019. A simple convolutional neural network for prediction of enhancer–promoter interactions with dna sequence data. *Bioinformatics*, 35(17):2899–2906.

A Mapping of SNP Tokens

As our resulting SNP tokens are represented as ‘ X_1/X_2 ’, to reduce the size and facilitate more straightforward representation for the downstream process in the NLP pipeline, such as pre-processing, tokenization, and modeling, we map all SNP tokens into a single character representation. The mapping of the SNP tokens into a single character representation is shown in Table 4. We use non-alphabetical characters as there are 66 SNP tokens in total, more than the available alphabetical characters, which consists of 52 characters (lower and upper case from ‘A’ to ‘Z’) in total. Also note that, all the SNP tokens related to the unknown token ‘N’ except ‘N/N’ (such as ‘A/N’, ‘G/NI’, ‘N/NI’, ‘NI/NI’, etc) are never been used since there is no actual SNP record corresponding to the unknown token ‘N’. The combinations of all ‘N’ and ‘NI’ tokens are listed on the table only for completion.

B Model Hyperparameters

We develop two Linformer (Wang et al., 2020) models, i.e., Dipformer and Hapformer, which is pre-trained using our proposed SNP tokens and the original nucleotide tokens, respectively. The two models have the same hyperparameter settings resulting in an equal number of parameters. We list all the hyperparameters of our Dipformer and Hapformer models in Table 5.

C Pre-Training Setup

During the pre-training phase, we build the BPE tokenizer with a vocab size of 32,000 for both the SNP tokens and nucleotide tokens datasets. We perform pre-training on both Dipformer and Hapformer models for 200,000 steps using masked language modeling with the cross entropy loss. During the pre-training, we apply a masking strategy similar to BERT (Devlin et al., 2019) with a 15% token replacement rate, where we replace with [MASK] 80% of the time, replace with a random token 10% of the time, and keep the token as is 10% of the time. We run the pre-training using 5 units of 2080Ti GPUs and an Intel(R) Xeon(R) Silver 4210 CPU. We use the same hyperparameter settings for pre-training both the Dipformer and Hapformer models. The hyperparameters of our pre-training are shown in Table 6.

D Fine-Tuning Setup

We fine-tune all models on Alzheimer’s disease risk prediction on a Chinese cohort consisting of 624 subjects in total, 384 of which are labelled as Alzheimer’s disease carriers (ADs) while 240 others are non-carriers (NCs). For predicting Alzheimer’s disease, we append a [CLS] token as the prefix of the sequence. During the fine-tuning, we take the output of the [CLS] token and perform a linear transformation on it to get the disease risk prediction. We evaluate the performance of all models using accuracy, area under the ROC curve (AUROC), and area under the precision-recall curve (AUPRC). We show all the hyperparameters of the fine-tuning phase in Table 7. We experiment with different learning rate for each model and find that the best setting is achieved when using a learning rate of $1e-4$ for models that are not pre-trained (non-pre-trained Dipformer and DeepSEA) and a learning rate of $1e-5$ for all pre-trained models (Dipformer and Hapformer).

E Detailed Results

In this section, we show the distribution of the 10-fold results from our experiment in the Alzheimer’s disease risk prediction task for all models (Dipformer, Hapformer, DeepSEA, and PLINK) on each evaluation metric. Figure 7 shows the distribution of the best 10-folds accuracy performance on the Alzheimer’s disease risk prediction task. Figure 8 shows the distribution of the best 10-folds AUROC performance on the Alzheimer’s disease risk prediction task. Figure 9 shows the distribution of the best 10-folds AUPRC performance on the Alzheimer’s disease risk prediction task.

A/A	A	DEL/A	腌	A/C	嘎	AI/C	爸	C/G	嚓	CI/G	懂
C/C	C	DEL/AI	拔	A/G	阿	AI/CI	比	C/N	拆	CI/GI	答
G/G	G	DEL/C	吃	A/N	呵	AI/G	八	C/T	礲	CI/N	达
N/N	N	DEL/CI	搭	A/T	钢	AI/GI	霸	C/CI	车	CI/NI	第
T/T	T	DEL/G	想	A/AI	吡	AI/N	巴	C/GI	床	CI/T	瘠
AI/AI	B	DEL/GI	香	A/CI	俺	AI/NI	逼	C/NI	穿	CI/TI	沓
CI/CI	D	DEL/N	学	A/GI	安	AI/T	把	C/TI	出	G/GI	高
GI/GI	H	DEL/NI	虚	A/NI	案	AI/TI	笔	GI/N	蝦	G/N	给
NI/NI	O	DEL/T	徐	A/TI	按	N/NI	讷	GI/NI	合	G/NI	股
TI/TI	U	DEL/TI	需	NI/T	喔	N/T	哪	GI/T	虾	G/T	个
DEL/DEL	X	T/TI	拓	NI/TI	依	N/TI	娜	GI/TI	盒	G/TI	该

Table 4: Mapping of SNP tokens into a single character representation.

Hyperparams	Value
#layers	6
dim	512
k	128
dropout	0.1
num heads	8
dim head	64
num embeddings	32000
single KV head	False
shared KV	False

Table 5: Model Hyperparameters

Hyperparams	Value
batch size	240
optimizer	AdamW
learning rate	1e-4
scheduler λ_1	1
scheduler λ_2	0.999991
#steps	200,000
warmup step	1000
loss fn	Cross Entropy
random seed	0

Table 6: Pre-Training Hyperparameters

Hyperparams	Value
batch size	16
optimizer	AdamW
learning rate	[1e-4..1e-6]
scheduler λ_1	1
scheduler λ_2	0.999991
#epoch	30
early stopping	3
loss fn	Cross Entropy
random seed	0

Table 7: Fine-Tuning Hyperparameters

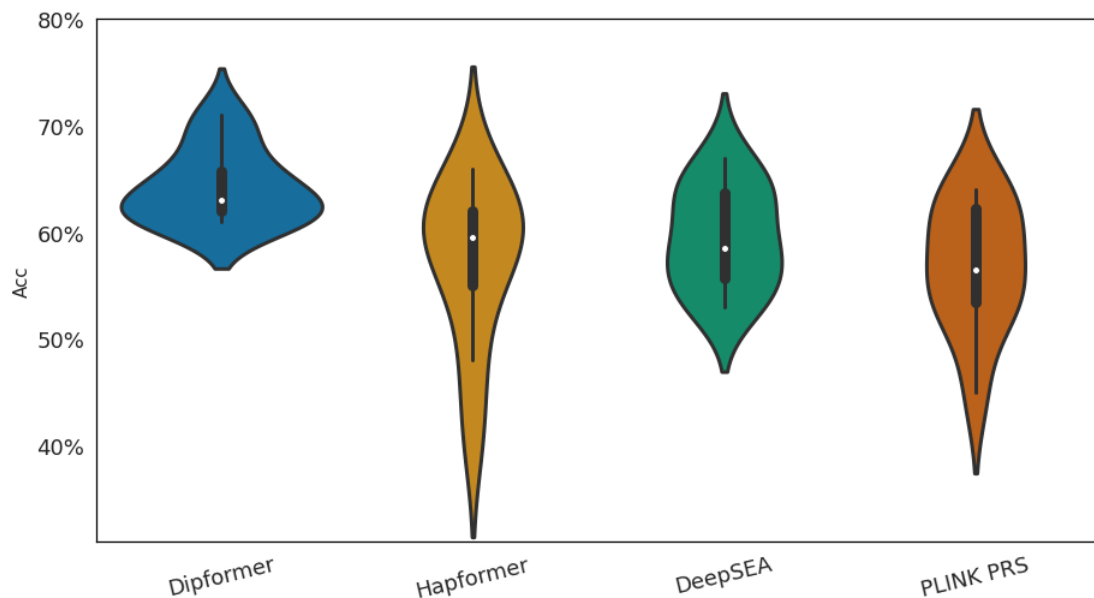


Figure 7: 10-folds accuracy performance of the best Dipformer, Hapformer, DeepSEA, and PLINK models on the Alzheimer's disease risk prediction.

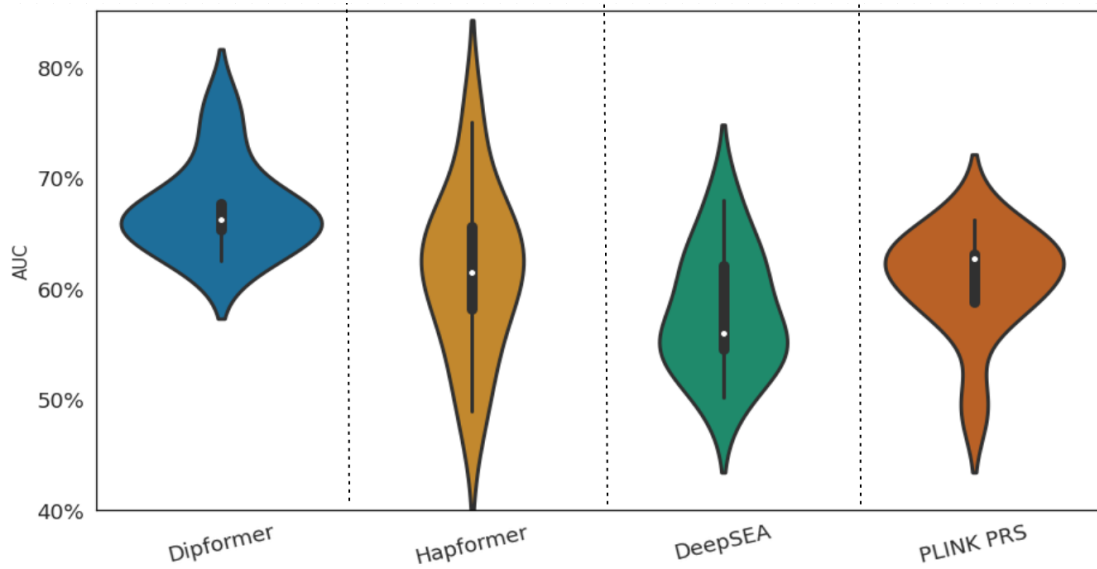


Figure 8: 10-folds AUROC performance of the best Dipformer, Hapformer, DeepSEA, and PLINK models on the Alzheimer's disease risk prediction.

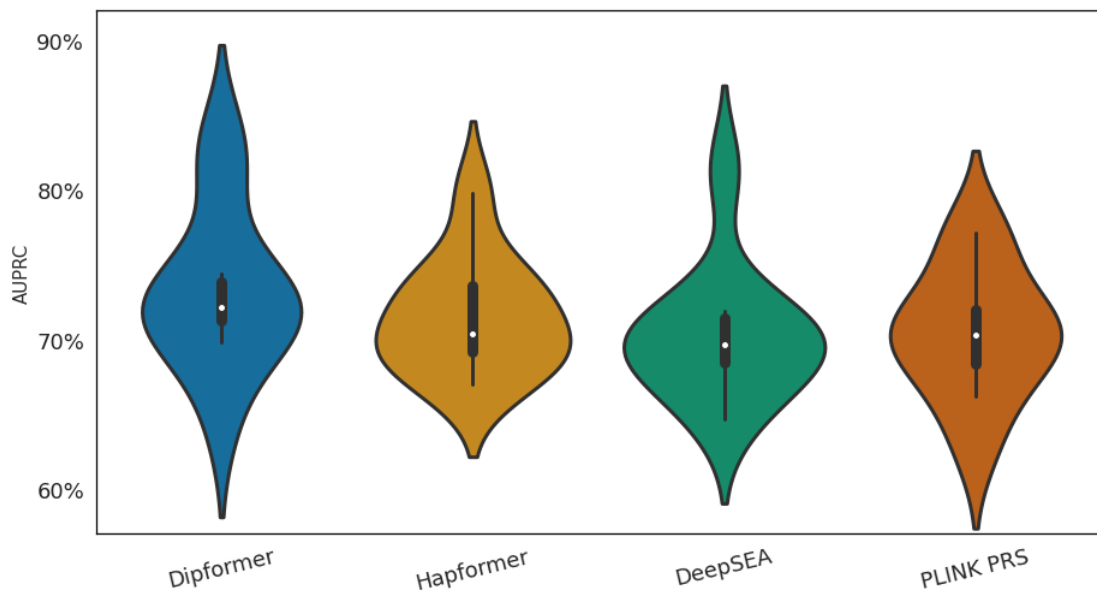


Figure 9: 10-folds AUPRC performance of the best Dipformer, Hapformer, DeepSEA, and PLINK models on the Alzheimer's disease risk prediction.