GENEFORMER: LEARNED GENE COMPRESSION USING TRANSFORMER-BASED CONTEXT MODELING

Zhanbei Cui[1,2] , Tongda Xu[1], Jia Wang[1,4], Yu Liao[1,3], Yan Wang[1]

[1]Institute for AI Industry Research (AIR), [2]Beijing University of Posts and Telecommunications, [3]University of Science and Technology of China, [4]Xidian University

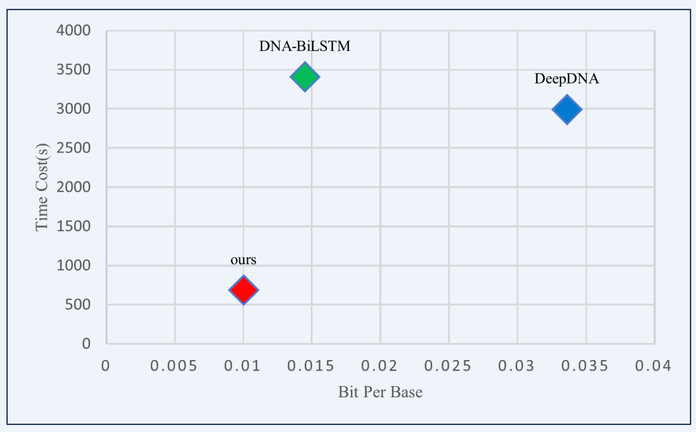
1 Introduction

The development of gene sequencing technology sparks an explosive growth of gene data. Thus, the storage of gene data has become an important issue.

As learning based entropy estimation combined with dynamic arithmetic coding has been applied in multi-media file compression, we propose a transformer-based gene compression method named GeneFormer.

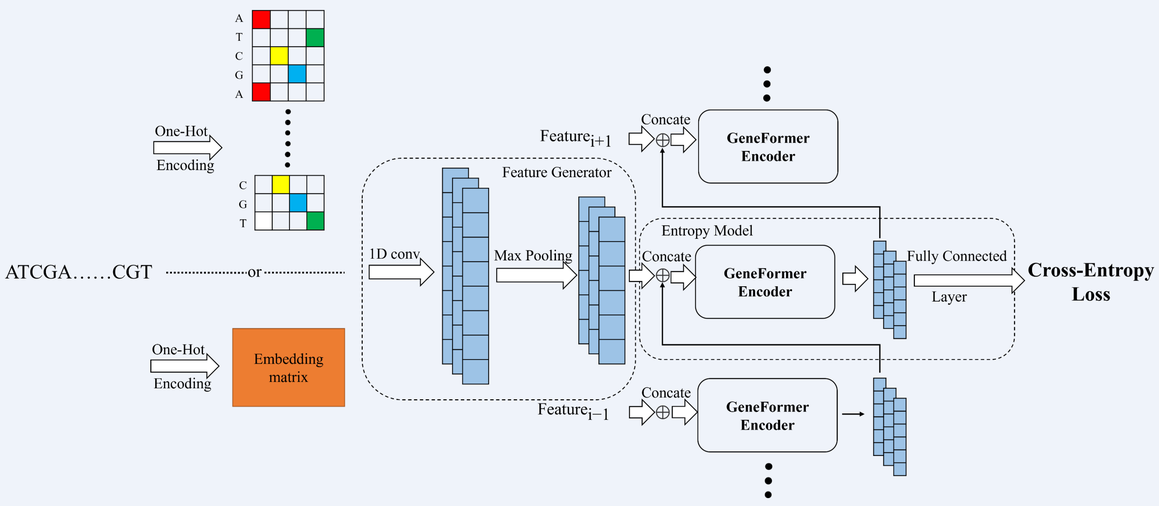
Specifically:

- We are **the first to introduce transformer structure for the problem of gene compression**. We introduce **latent array** into transformer-xl for the problem of gene compression.
- We design a **multi-level-grouping method** combining three grouping methods to improve compression ratio and reduce latency.
- Our model achieves **state-of-the-art** compression ratio, and is significantly faster than all the existing learning-based methods.



- The bpp-time cost of different methods.

2 Methodology



Model Structure:

- We propose GeneFormer to compress genome sequence data. GeneFormer contains two main components, a CNN-based feature generator to learn a latent representation, a transformer-based entropy model followed by a linear layer and softmax layer to predict the probability of the current base.

Transformer-based Model with Latent Array:

- We design the architecture of GeneFormer on the basis of transformer-xl. The original self-attention formula is as follows:

$$Attention(X_t) = Softmax(\frac{X_t W^Q (X_t W^K)^T}{\sqrt{d_k}}) X_t W^V$$

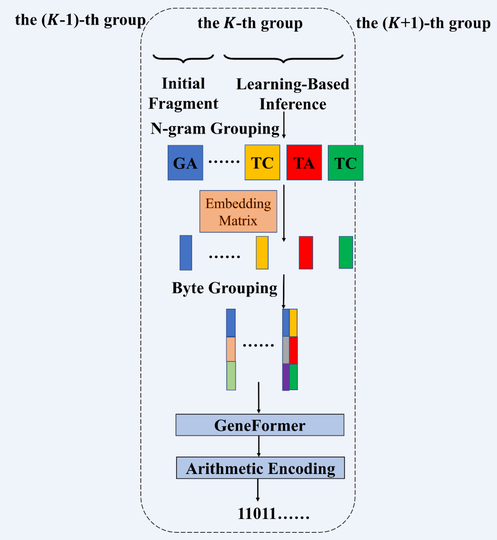
- We introduce a latent array in the GeneFormer encoder. As transformer-xl naturally has segment-Level recurrence mechanism, we adapt it as the ”latent array” of our encoder. After every forward propagation of input feature Xt, the GeneFormer stores the output Ht of the encoder as latent array, and concatenates it to the next input feature Xt+1. So that the encoder can restore all the information of the segments before current base without extra computation.

$$Attention(X_t) = Softmax(\frac{X_t W^Q (\hat{X}_t W^K)^T}{\sqrt{d_k}}) \hat{X}_t W^V$$

while $\hat{X}_t = [H_{t-1}, X_t]$, $[\cdot]$ means concatenating in the length dimension.

2 Multi-level-grouping method

- We combined 3 kinds of grouping methods including **FG,BG** and **NG** to improve our compression and decompression process and call the combination multi-level-grouping method.



- **Fixed-length grouping (FG).** We divide the gene sequence into groups and fix the length of bases for all the groups to trade off bpb and decoding latency.
- **Byte-grouping (BG).** Each byte is projected into h/g-dim vector space. Then they concatenate g adjacent bytes into a h-dim vector. So that compared to the original way, this byte-grouping method can extend the context length for g times.
- **N-gram grouping (NG).** We group the bases before feeding them into the embedding layer.

3 Experiments

We train and test our method on two datasets in the experiment to prove the superiority of our method.

Datasets:

- **Human dataset**(contains 1000 human complete mitochondrial sequence)
- **Fish dataset**(contains 2851 mitochondrial sequences of various fish specie)

Results:

Method Name	Human(bpb)	Human(Time Cost)	Fish(bpb)	Fish(Time Cost)
G-zip	1.4232	0m0.15s	2.446	0m0.62s
7zip	0.091	0m0.38s	1.2371	0m3.43s
bzip2	0.3558	0m0.22s	2.0303	0m0.44s
MFCompress	1.5572	0m1.56s	1.4119	0m5.44s
Genozip	0.136	0m2.36s	1.2869	0m1.87s
DeepDNA	0.0336	49m53s	1.3591	133m14s
DNA-BiLSTM	0.0145	56m47s	0.7036	146m34s
GeneFormer(ours)	0.0097	82m18s	0.6587	232m57s
GeneFormer+Byte-grouping(ours)	0.0075	84m34s	0.4085	229m07s
GeneFormer+Multi-level-grouping(ours)	0.01	11m24s	0.4794	30m09s
DeepDNA(hybrid dataset)	0.0701	-	1.1999	-
DNA-BiLSTM(hybrid dataset)	0.036	-	1.055	-
GeneFormer(hybrid dataset)	0.0215	-	0.8634	-

- Comparisons. Based on the datasets mentioned above, we compare GeneFormer with multiple methods including traditional methods and deeplearning-based methods.

- GeneFormer outperforms all other compression methods on the two datasets. The bpb of our algorithm without grouping (0.0097) is only 66.8% of the current state-of-the-art learning-based method DNA-BiLSTM [5], and only 0.7% of the popular G-zip method.

4 Key References

1. Rongjie Wang, Yang Bai, Yan-Shuo Chu, Zhenxing Wang, Yongtian Wang, Mingrui Sun, Junyi Li, Tianyi Zang, and Yadong Wang, “Deepdna: A hybrid convolutional and recurrent neural network for compressing human mitochondrial genomes,” in 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2018, pp. 270–274.
2. Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” arXiv preprint arXiv:1901.02860, 2019.
3. Wenwen Cui, Zhaoyang Yu, Zhuangzhuang Liu, Gang Wang, and Xiaoguang Liu, “Compressing genomic sequences by using deep learning,” in International Conference on Artificial Neural Networks. Springer, 2020, pp. 92–104.