DNAHLM - DNA sequence and Human Language mixed large language Model

Wang Liang
Huazhong University of Science and Technology, 430070, P.R. China
*To whom correspondence should be addressed. E-mail:wangliang.f@gmail.com

[Abstract] There are already many DNA large language models, but most of them still follow traditional uses, such as extracting sequence features for classification tasks. More innovative applications of large language models, such as prompt engineering, RAG, and zero-shot or few-shot prediction, remain challenging for DNA-based models. The key issue lies in the fact that DNA models and human natural language models are entirely separate; however, techniques like prompt engineering require the use of natural language, thereby significantly limiting the application of DNA large language models. This paper introduces a hybrid model trained on the GPT-2 network, combining DNA sequences and English text to explore the potential of using prompts and fine-tuning in DNA models. The model has demonstrated its effectiveness in DNA related zero-shot prediction and multitask application.

1 Introduction

Large language models have emerged as a groundbreaking innovation in the field of artificial intelligence. Large language models are also applied in the analysis of DNA sequences.The primary application of DNA large language models lies in sequence feature extraction and classification tasks[1-4]. These models can analyze DNA sequences to identify patterns, predict gene functions, and even diagnose genetic diseases.

Despite their successes, traditional DNA large language models face several limitations. One significant challenge is the difficulty in applying novel prompt engineering techniques. Prompt engineering is the foundation of large model applications, as approaches like RAG , agents, and function calls all rely on well-crafted prompts to be built effectively.

However, traditional DNA models struggle to incorporate such techniques effectively due to the fundamental differences between DNA sequences and natural language[5-10].

For example, Current large language models can easily accomplish the following tasks:

Question:

Determine whether the sentiment of following text is positive or negative ?

"""

This is the worst thing the TMNT franchise has ever spawned. I was a kid when this came out and I still thought it was deuce, even though I liked the original cartoon

"""

Answer:

negative

However, for large language models specialized in DNA, the following tasks are extremely challenging.

Question:

Determine the following dna sequence is promoter or terminator?

"""

GATTCCGTGGACTCGAGGCCCGCGTCCTCCGCCCTCCTGTGGCCCCGACCTGCCCGGAGCGCGTTCCCCGCCGGCGTCCGCTGCCGCTCACACCCACCCCAGTACCTGGCGGGCCCGGAGCGCGCGCG

"""

Answer:

terminator

In light of these limitations, there is a growing need to explore hybrid models that can integrate the strengths of DNA sequences with natural language. Our research involves using DNA sequences and English text to train a base GPT-2 model. We then convert the data related to DNA classification tasks into an instruction-tuning format and fine-tune the GPT-2 model accordingly. This process results in a model that can complete DNA-related tasks using natural language prompts.

2 Materials and methods

2.1 DNAHLM

To develop the DNAHLM, we adopted the GPT-2 architecture as the foundation due to its proven effectiveness in natural language processing tasks.We then fine-tuned the model for a classification task. For comparison, we used two fine-tuning methods: classification tuning and instruction tuning. The training process is shown in Fig.1:

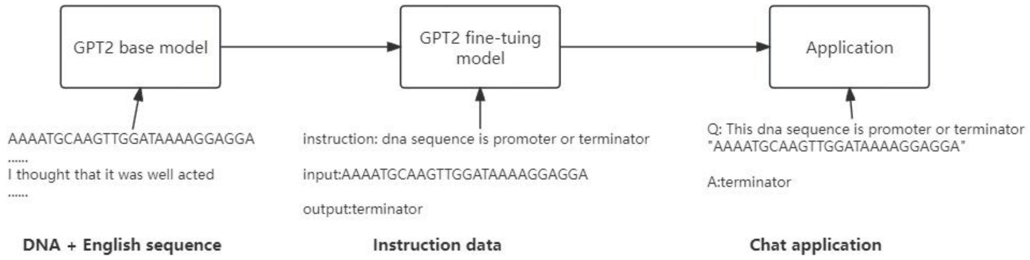

GPT2 base model
GPT2 fine-tuing model
Application
AAAATGCAAGTTGGATAAAAGGAGGA
......
I thought that it was well acted
......
instruction: dna sequence is promoter or terminator
input:AAAATGCAAGTTGGATAAAAGGAGGA
output:terminator
Q: This dna sequence is promoter or terminator "AAAATGCAAGTTGGATAAAAGGAGGA"
A:terminator
DNA + English sequence
Instruction data
Chat application

Fig.1 DNAHLM training process

The technology for training GPT-2 base models from scratch is already very mature. We will primarily focus on the fine-tuning methods.

Classification tuning.This is currently the most common application approach for large models in the field of DNA. In classification tuning, the model is trained to recognize specific category label ids, such as "0(promoter)" and "1(terminator)." Classification tasks can also include recognizing Splice site, Transcription factor, etc. However, a model fine-tuned for classification can only make judgments about the specified categories and cannot perform other types of judgments on the input text, Fig.2. For exmaple, promoter detection need one classification fine-tuned model, the Transcription factor classification need another.

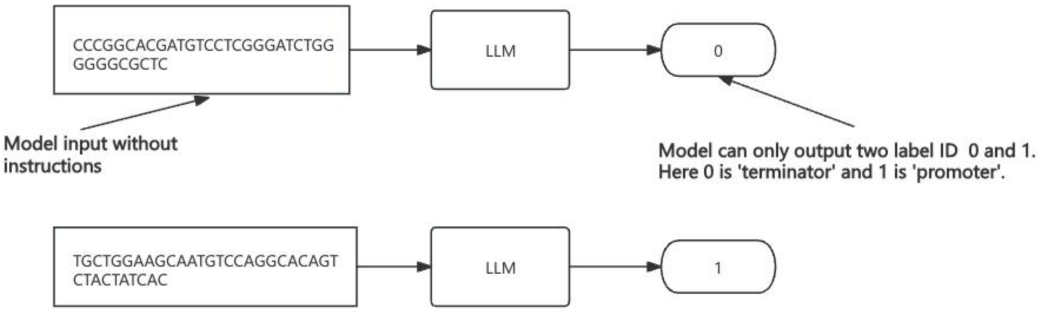

CCCGGCACGATGTCCTCGGGATCTGG
GGGGCGCTC
LLM
0
Model input without
instructions
Model can only output two label ID 0 and 1.
Here 0 is 'terminator' and 1 is 'promoter'.
TGCTGGAAGCAATGTCCAGGCACAGT
CTACTATCAC
LLM
1

Fig2 DNA Classification tuning tasks

Instruction tuning. Instruction tuning involves training the model on specific tasks to enhance its ability to understand and execute tasks described in natural language prompts, as shown in Figure 3. We can think of classification-tuned models as highly specialized models. By contrary, Models fine-tuned with instruction tuning are generally capable of performing a broader range of tasks. That means we could predict the DNA Splice site, Transcription factor, even functions by one model. This is precisely the focus of our research.

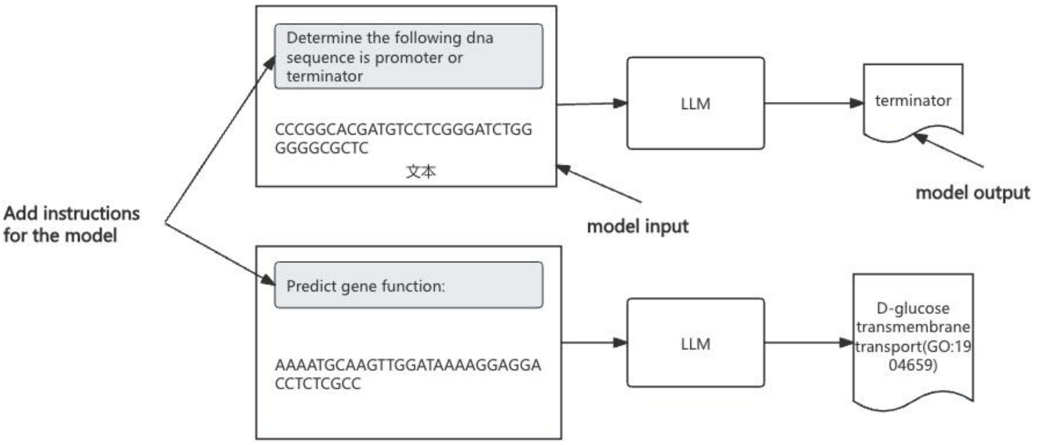
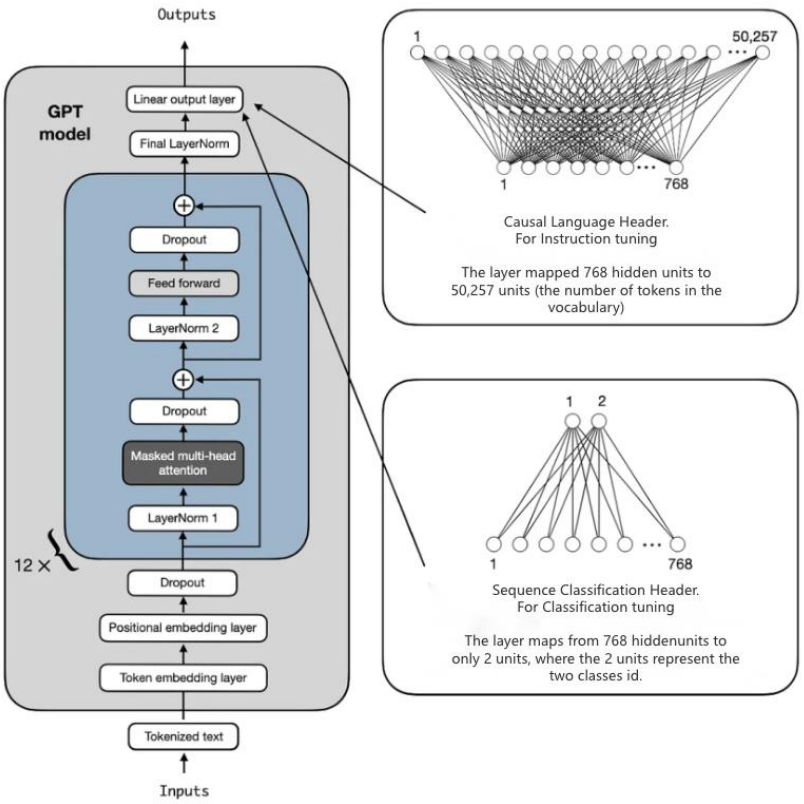


Fig 3. DNA Instruction tuning tasks

The DNAHLM uses the exact same model architecture as GPT-2, with the primary difference being the training corpus. For the research validation, the base model is structured as GPT-2 Small, which can be trained on a single 4090 GPU. For classification fine-tuning, a typical classification head is used, outputting two classes. For instruction fine-tuning, the same head as the base model is used, and the training mode is identical to that of the pre-training. The structure of DNAHLM is shown in Fig.4.



Outputs
GPT model
Linear output layer
Final LayerNorm
Dropout
Feed forward
LayerNorm 2
Dropout
Masked multi-head attention
LayerNorm 1
12 ×
Dropout
Positional embedding layer
Token embedding layer
Tokenized text
Inputs
1
50,257
1
768
Causal Language Header.
For Instruction tuning
The layer mapped 768 hidden units to 50,257 units (the number of tokens in the vocabulary)
1
2
1
768
Sequence Classification Header.
For Classification tuning
The layer maps from 768 hiddenunits to only 2 units, where the 2 units represent the two classes id.

Fig.4. struture of DNAHLM

The DNA sequence processing module utilizes a custom-designed DNA embedding layer that converts DNA sequences into high-dimensional vectors. This layer is followed by a series of Transformer blocks, similar to those in the GPT-2 architecture, which process the DNA sequence vectors to extract meaningful features. The natural language processing module, on the other hand, employs the standard GPT-2 architecture to process natural language text inputs.

The two modules are interconnected through a shared embedding layer that combines the DNA sequence and natural language embeddings. This shared embedding layer allows the model to leverage the contextual information from both domains, enabling it to generate coherent and meaningful outputs that integrate DNA and natural language information.

2.2 Data Preprocessing

To prepare the data for training the DNAHL model, we first need to convert DNA and English sequences into a format suitable for processing. This involves several steps:

1 DNA and English dataset.For the DNA data, we used human genome data, dividing the human genome into segments of 300 to 1000 base pairs (bp) in length, and then randomly selected some of these segments as the DNA training data. For the English text data, we used data from the English version of Wikipedia. We selected 150MB of DNA sequence data and 150MB of English text data, and then combined them to serve as the training data for the GPT-2 model.

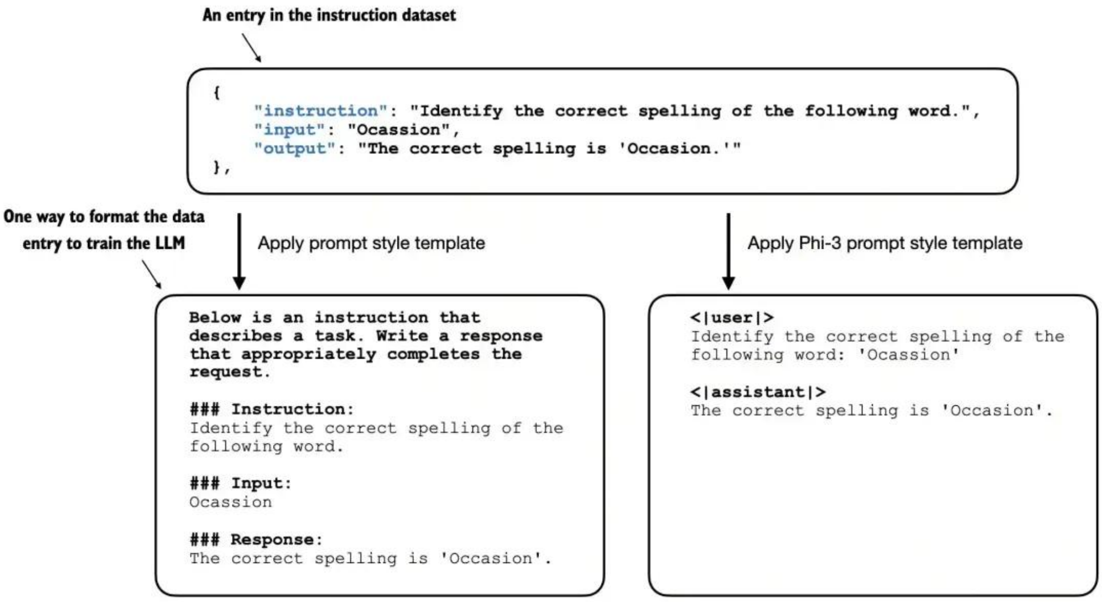
2 Fine-tuning dataset.For fine-tuning, we used typical DNA sequence analysis tasks and converted them into a fine-tuning data format for instruction tuning. A typical example includes the following downstream tasks:

- Core Promoter Detection
- Transcription Factor Prediction
- Promoter Detection

- Splice Site Detection

These tasks were transformed into an instruction-tuning format to fine-tune the model.

Instruction tuning is a supervised learning method where the training data consists of instructions, inputs, and outputs. These three components are then formatted into a specific prompt format. Below are two common methods of formatting.Fig.5. In this article, we will use the Alpaca-style prompt formatting method.


An entry in the instruction dataset
One way to format the data entry to train the LLM
Apply prompt style template
Apply Phi-3 prompt style template

```
{
    "instruction": "Identify the correct spelling of the following word.",
    "input": "Ocassion",
    "output": "The correct spelling is 'Occasion.'"
},
```

```
Below is an instruction that
describes a task. Write a response
that appropriately completes the
request.

### Instruction:
Identify the correct spelling of the
following word.

### Input:
Ocassion

### Response:
The correct spelling is 'Occasion'.
```

```
<|user|>
Identify the correct spelling of the
following word: 'Ocassion'

<|assistant|>
The correct spelling is 'Occasion'.
```

Fig.5 Instruction data format

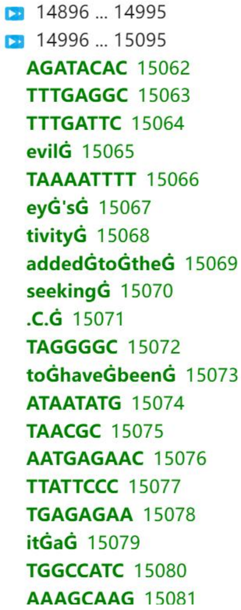
2.3 Training Strategy

2.2.1Tokenization

K-mer tokenization an approach that has been widely used in analyzing DNA sequences. The k-mer representation incorporates richer contextual information for each deoxynucleotide base by concatenating it with its following ones. The concatenation of them is called a k-mer.

For example, a DNA sequence ‘ATGGCT’ can be tokenized to a sequence of four 3-mers: {ATG, TGG, GGC, GCT} or to a sequence of two 5-mers: {ATGGC, TGGCT}.

But Byte-Pair Encoding (BPE) tokenization has been shown to be more efficient in large DNA models. For example, the sequence 'ATGGCT' can be tokenized as {ATGG, CT}. Therefore, we also used the BPE tokenization algorithm. We built a vocabulary of approximately 50,000 tokens based on a mixed corpus of English and DNA sequences, with DNA tokens comprising about 55% and English tokens about 45%. The tokenizer includes special tokens::<|endoftext|>. A partial screenshot of the vocabulary is shown below. Fig.5.



```
14896 ... 14995
14996 ... 15095
  AGATACAC 15062
  TTTGAGGC 15063
  TTTGATTC 15064
  evilĠ 15065
  TAAAATTTT 15066
  eyĠ'sĠ 15067
  tivityĠ 15068
  addedĠtoĠtheĠ 15069
  seekingĠ 15070
  .C.Ġ 15071
  TAGGGGC 15072
  toĠhaveĠbeenĠ 15073
  ATAATATG 15074
  TAACGC 15075
  AATGAGAAC 15076
  TTATTCCC 15077
  TGAGAGAA 15078
  itĠaĠ 15079
  TGGCCATC 15080
  AAAGCAAG 15081
```

Fig.6 screenshot of DNAHLM vocabulary

2.2.2Pre-training

According to the typical design of GPT-2, it accepts sequences with a maximum length of 1024 as input. We used the same architecture as the GPT-2 Small model, which consists of 12 Transformer layers, each with 768 hidden units and 12 attention heads. We trained the GPT-2 model using mixed-precision floating-point arithmetic on a machine equipped with a single Nvidia 4090 GPU. We employed a dynamic learning rate schedule, and the model was trained for a total of 3 to 5 epochs.

2.2.3 Classification Fine-tuning

For each downstream application, we started from the pre-trained parameters and fine-tuned DNAHLM with task-specific data. We utilized the same training tricks across all the applications, where the learning rate was first linear warmed-up to the peak value and then linear decayed to near 0. We split training data into training set and developing set for hyperparameter tuning.

2.2.3 Instruction Fine-tuning

The method for instruction tuning is entirely consistent with the training method of the GPT-2 base model. This involves treating the constructed instruction data as general text sequences and inputting them into the GPT-2 model. In other words, instruction tuning is the same as the pre-training process, which is key to enabling multi-task handling. We typically train for 2 to 3 epochs. Since the training data is relatively small, fine-tuning on a 4090 GPU usually takes only about ten minutes to complete.

2.4 Model Evaluation

The performance of the DNAHL model is evaluated using a series of benchmarks that assess its capabilities in both DNA sequence and promotion tasks.

For classification fine-tuning, we evaluate the model according to typical classification tasks.This benchmark evaluates the model's ability to classify DNA sequences into different categories based on their functional or structural properties. The model's accuracy is used to measure its performance.

For the evaluation of the instruction-tuned model, as a comparison, we commonly use accuracy as evaluation metrics. The assessment is based on whether the model's output tokens match the expected ones. In contrast, the output for classification fine-tuning is typically a class ID, such as 1, 2, etc.

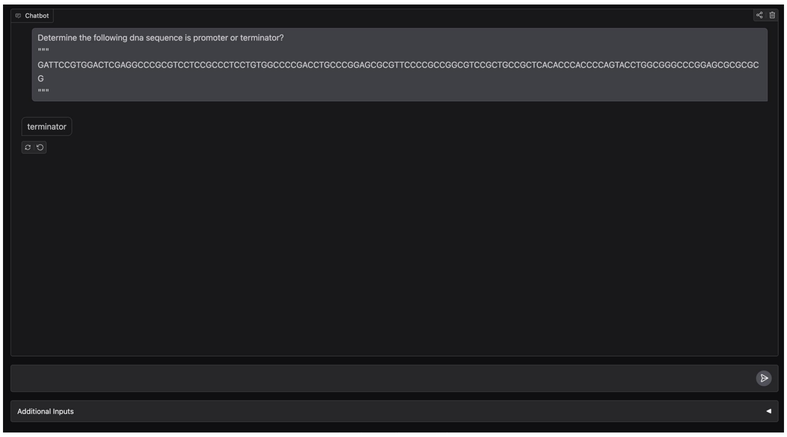
It is important to note that if the model outputs "promoter AGCC GGG" while the expected output is "promoter ", we still consider the model's prediction to be accurate. This is because, as a generative model, GPT is not specifically trained to recognize stop tokens. With more training data and a larger model size, it would be possible to produce outputs that exactly match the expected tokens.

3 Experimental Results

For the specific promoter prediction task, the classification fine-tuned model achieved an accuracy of approximately 83%, while the instruction-tuned model achieved an accuracy of approximately 80%. This indicates that the instruction-tuned model remains effective even when dealing with specialized classification problems.

For multiple tasks such as Core Promoter Detection and Transcription Factor Prediction, the classification fine-tuning method requires training a different model for each task, with an average accuracy of approximately 81%. In contrast, the instruction-tuned model, when facing multiple types of classification tasks, achieved an accuracy of around 78%, but these results were obtained using a single model.

More importantly, the instruction-tuned model can interact with users in a conversational manner and can leverage prompt engineering, RAG, chain of thought, and other techniques commonly used with large language models.Fig7. This significantly expands the range of applications for the model.



Determine the following dna sequence is promoter or terminator?
"""
GATTCCGTGGACTCGAGGCCCGCGTCCTCCGCCCTCCTGTGGCCCCGACCTGCCCGGAGCGCGTTCCCCGCCGGCGTCCGCTGCCGCTCACACCCACCCCAGTACCTGGCGGGCCCGGAGCGCGCGC
G
"""

terminator

Fig.7. Chat with DNAHLM

4 Conclusion

In summary, this research paper presents the development and evaluation of the DNAHL model, a hybrid large language model that integrates DNA sequences with human language.The integration of DNA sequences with natural language processing techniques enables the DNAHL model to leverage contextual information from natural language, enhancing its understanding and prediction capabilities for DNA sequences. This innovative approach addresses the limitations of traditional DNA models, such as difficulties in prompt engineering, RAG, and zero-shot prediction, thereby offering a more powerful and adaptable solution for complex tasks.

Future research could involve using larger-scale models, such as LLaMA 3, and training on more extensive datasets, such as genomic data from multiple model organisms and additional fine-tuning data for various DNA-related downstream tasks. The development of specialized large model platforms for DNA research could also be based on DNAHLM.

Reference

1. Yang, A. *et al.* Review on the application of machine learning algorithms in the sequence data mining of DNA. *Front. Bioeng. Biotechnol.* **8**, 1032 (2020). doi:doi.org/10.3389/fbioe.2020.01032

2. Zhai, J. *et al.* Cross-species modeling of plant genomes at single nucleotide resolution using a pre-trained DNA language model. *bioRxiv* 2024.06.04.596709 (2024) doi:10.1101/2024.06.04.596709.

3. 1.Wang, B. *et al.* Pre-trained Language Models in Biomedical Domain: A Systematic Survey. *ACM Comput. Surv.***56**, 1–52 (2023).doi:doi.org/10.1145/3611651

4. 1.Lam, H. Y. I., Ong, X. E. & Mutwil, M. Large language models in plant biology. *Trends Plant Sci.* (2024) doi:10.1016/j.tplants.2024.04.013.

5. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: Pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* **37**, 2112–2120 (2021). doi:doi.org/10.1093/bioinformatics/btab083

6. Chen, Z., Wei, L. & Gao, G. Foundation models for bioinformatics. *Quant. Biol.* (2024) doi:10.1002/qub2.69.

7. Bhattacharya, M., Pal, S., Chatterjee, S., Lee, S.-S. & Chakraborty, C. Large language model to multimodal large language model: A journey to shape the biological macromolecules to biological sciences and medicine. *Mol. Ther. - Nucleic Acids* **35**, 102255 (2024).doi:

8. Benegas, G., Ye, C., Albors, C., Li, J. C. & Song, Y. S. Genomic Language Models: Opportunities and Challenges. *arXiv* (2024) doi:10.48550/arxiv.2407.11435.

9. An, W. *et al.* Advancing DNA Language Models through Motif-Oriented Pre-Training with MoDNA. *BioMedInformatics* **4**, 1556–1571 (2024). doi:10.1016/j.omtn.2024.102255

10. İhtiyar, M.N., Özgür, A. Generative language models on nucleotide sequences of human genes. *Sci Rep* **14**, 22204 (2024). https://doi.org/10.1038/s41598-024-72512-x

11. DNAHLM github, https://github.com/maris205/DNAHL

12. DNAHLM huggingface project. https://huggingface.co/dnagpt