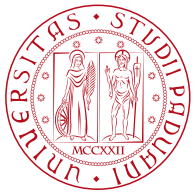

Università degli Studi di Padova

Promoter Prediction with DNABERT and DNABERT-2: A Reproduction and Analysis

Author: Dario Furlan, Nicolò Raccichini

June 9, 2025

1 Introduction

In recent years, researchers have begun to apply techniques from natural language processing (NLP) to genomic data. The key idea is that DNA, much like human language, can be represented as a sequence of symbols that carries information. Among the models inspired by this idea, DNABERT [Ji et al., 2021] stands out as one of the first to treat DNA as a "language", using the Transformer architecture originally designed for textual data. In this approach, DNA sequences are broken into overlapping fragments of fixed length called *k-mers*, which serve as the basic units or "words" for the model.

This analogy between language and DNA opens new possibilities for understanding complex biological mechanisms. One important application is **promoter prediction**, the task of identifying regions in the DNA that initiate gene transcription. Promoters act as on/off switches for genes, and knowing where they are helps us understand how genes are regulated. This project focuses on evaluating the performance of DNABERT and its improved version, DNABERT-2 [Zhou et al., 2024], in this specific task.

To guide our exploration, we will address the following key points, each building upon the previous:

1. First, we explain how DNABERT is designed and why it outperforms older models.
2. Next, we examine the main innovations introduced in DNABERT-2.
3. Then, we introduce the biological concept of promoters and discuss their relevance to gene regulation.
4. Finally, we replicate DNABERT's promoter prediction experiment using Python, applying theory to practice.

Through this structure, we aim to develop a clearer understanding of how Transformer-based models are adapted for genomic data and why they offer advantages over traditional machine learning techniques. To support this analysis, we first provide background on DNA, genes, and transcription, gradually introducing the biological concepts that motivate the use of these models.

1.1 DNA as a Language

To understand why DNA can be treated as a language, we must look at its basic structure. DNA (deoxyribonucleic acid) is the molecule that stores the genetic instructions for all living organisms. It consists of two strands twisted into a double helix, where each strand is made up of smaller units called *nucleotides* [Searls, 1992].

Each nucleotide includes a sugar, a phosphate group, and one of four bases: adenine (A), thymine (T), cytosine (C), or guanine (G). These bases form pairs—A with T, C with G—creating the rungs of the DNA ladder. The sequence of these bases determines the genetic code. Much like letters form words and sentences, the order of bases carries instructions used by cells to build proteins and control biological functions.

This structural regularity and symbolic representation make it possible to apply language-based models to DNA, treating sequences of bases as sentences and short fragments (*k-mers*) as words.

1.2 Genes and Transcription

Within the long strands of DNA, specific segments known as *genes* contain the instructions for making proteins or functional RNA molecules. The first step in using this genetic information is **transcription**, a process in which a gene is copied into a molecule called *messenger RNA* (mRNA).

Transcription begins when an enzyme called *RNA polymerase* binds to a particular region of the DNA known as a **promoter**. This region lies just before the gene and signals the start of transcription. Once mRNA is produced, it leaves the cell nucleus and is used in the next step, called **translation**, where ribosomes read the mRNA to assemble the corresponding protein. The order of bases in the mRNA determines the sequence of amino acids in the final protein.

Thus, promoters are essential components of the transcription process, acting as the control points where gene expression begins.

1.3 Promoters

Promoters are short DNA sequences located upstream of genes. They serve as binding sites for RNA polymerase and other proteins that initiate transcription. Without a promoter, the transcription machinery would not know where to start copying the DNA.

Some promoters contain a conserved sequence called the *TATA box*, typically found 25 to 35 base pairs before the start of a gene. This element helps position the transcription machinery at the right spot. However, not all promoters

include a TATA box. These are called *no-TATA promoters*, and they rely on alternative sequences and factors to guide transcription initiation.

Understanding the structure and variability of promoters is crucial for predicting their presence in a sequence—a task where machine learning models like DNABERT can provide valuable support.

1.4 Why Traditional Models Fall Short

Before the introduction of Transformer-based models, researchers mainly used *Convolutional Neural Networks (CNNs)* and *Recurrent Neural Networks (RNNs)* to analyze DNA sequences. These models brought useful insights but faced several limitations when applied to genomic tasks like promoter prediction.

CNNs are effective at detecting local patterns within a fixed-size window, but they struggle to capture long-range dependencies—important in DNA, where regulatory elements may be far apart. RNNs, particularly advanced forms like LSTMs and GRUs, process sequences step-by-step and can, in theory, handle longer contexts. However, they often suffer from the **vanishing gradient problem**, which causes earlier information in the sequence to fade and become inaccessible to the model as it progresses.

Both CNNs and RNNs also compress the sequence into a fixed-size internal state, limiting their ability to capture complex dependencies. Moreover, they usually require large, well-labeled datasets for training, which are scarce and expensive in the biological domain.

Transformer-based models like DNABERT overcome these limitations through *self-attention*, a mechanism that allows the model to focus on all parts of the sequence simultaneously. This makes them especially suitable for tasks where patterns can span long distances—such as identifying promoters in genomic sequences.

2 The Model DNABERT

DNABERT is a model that follows the same paradigms introduced by BERT, but adapted to the DNA sequences [Devlin et al., 2018]. BERT is a transformer-based contextualized language representation model that has achieved superhuman performance in many natural language processing (NLP) tasks. It introduces a paradigm of pre-training and fine-tuning, which first develops general-purpose understandings from massive amount of unlabeled data and then solves various applications with task-specific data with minimal architectural modification.

2.1 Architecture

The DNABERT model adopts the same architecture as BERT-base:

- 12 Transformer encoder layers
- 768-dimensional hidden states
- 12 self-attention heads

Each input sequence is limited to a maximum of 512 tokens and is enclosed between two special tokens: [CLS] and [SEP]. As in the original BERT, the [CLS] token is used to represent the entire input sequence and is utilized for sequence-level predictions such as classification tasks.

However, unlike standard BERT, DNABERT is specifically adapted to the genomic domain through a different tokenization scheme. Instead of using WordPiece, it tokenizes DNA sequences using k-mers, where each k-mer is a substring of length k (with k = 3, 4, 5, or 6). This strategy captures richer local nucleotide context and reduces sparsity. For each value of k, a separate DNABERT model is trained (e.g., DNABERT-3, DNABERT-4, etc.).

The input to DNABERT is thus a sequence of overlapping k-mers, converted to embeddings through the sum of three vectors:

- Token embeddings (for each k-mer),
- Position embeddings (to encode sequence order),
- Segment embeddings (to distinguish sequence segments, although only one segment is typically used for DNA).

Internally, the model applies standard Transformer operations: multi-head self-attention, feed-forward layers, residual connections, and layer normalization—repeated across all 12 layers, as shown in the Figure 1.

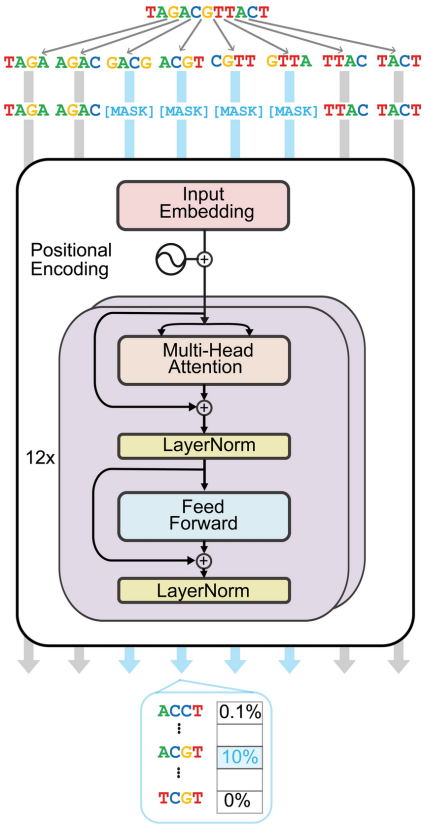

FIGURE 1: The model architecture of DNABERT [Sanabria et al., 2024]

During pre-training, DNABERT uses a masked language modeling (MLM) objective, like BERT, but without next sentence prediction (NSP), which is not applicable in genomic contexts. Approximately 15–20% of k-mers in each sequence are randomly masked in contiguous spans, and the model is trained to predict them based on surrounding context. This enables the model to learn the "syntax" and "semantics" of genomic sequences in a self-supervised fashion.

To support sequences longer than 512 tokens, DNABERT also provides an extended version called DNABERT-XL,

where long sequences are split into chunks, independently processed, and their representations concatenated.

2.2 Pretraining

The pretraining of DNABERT is performed in a self-supervised fashion, meaning that no labeled data is required. Similar to the original BERT, DNABERT uses a Masked Language Modeling (MLM) objective: approximately 15% to 20% of the input k-mer tokens are randomly masked, and the model is trained to predict these masked tokens based on the surrounding context.

Unlike the original BERT, where individual tokens are masked, DNABERT masks contiguous spans of k-mers. This is crucial because individual k-mers can often be trivially inferred from adjacent ones due to overlapping, especially in genomic sequences. Masking contiguous regions prevents the model from "cheating" and encourages it to learn deeper contextual relationships across upstream and downstream nucleotide patterns.

The goal of pretraining is to allow DNABERT to learn general-purpose representations of DNA, capturing both local motifs and long-range dependencies, without requiring any manual annotations. The learned embeddings encode the "syntax" and "semantics" of non-coding DNA regions, enabling effective transfer to various downstream tasks such as promoter prediction, transcription factor binding site (TFBS) identification, and splice site detection.

DNABERT was pretrained on the entire human genome, using input sequences of varying lengths (from 10 to 510 k-mers), sampled via both random and non-overlapping segmentation strategies. The model was trained for 120,000 steps with a batch size of 2000. During the first 100,000 steps, a masking rate of 15% was used, which was increased to 20% in the final 20,000 steps. The learning rate followed a warm-up and linear decay schedule, peaking at 4e-4.

Pretraining was computationally intensive, taking approximately 25 days on 8 NVIDIA 2080Ti GPUs. Four separate models were trained, corresponding to different k-mer granularities: k = 3, 4, 5, and 6. Among these, DNABERT-6 (with 6-mers) achieved the best performance and was used in the majority of downstream experiments.

This extensive pretraining enables DNABERT to serve as a general foundation model for genomic sequence analysis, capable of achieving strong performance even in data-scarce settings.

2.3 Fine-tuning

Once pretrained, DNABERT can be easily fine-tuned on a variety of downstream genomic tasks by adding a lightweight task-specific classification head on top of the [CLS] token output and continuing training on a small labeled dataset. This transfer learning approach allows the model to adapt its general understanding of DNA sequence patterns to more specific biological functions with minimal additional supervision.

DNABERT has been successfully fine-tuned for several key tasks in regulatory genomics, including:

- Promoter prediction: identifying proximal and core promoter regions around transcription start sites (TSS), including both TATA and non-TATA promoters.
- Splice site recognition: detecting canonical and non-canonical donor and acceptor sites, even in the presence of confounding sequence motifs.
- Transcription factor binding site (TFBS) identification: locating precise DNA segments bound by transcription factors, using ChIP-seq enriched regions.

During fine-tuning, the pretrained weights of DNABERT are updated jointly with the classifier head using task-specific data. The training typically employs optimization strategies such as learning rate warm-up, linear decay, dropout regularization, and the AdamW optimizer. Hyperparameters are tuned on a validation set to maximize generalization.

A key strength of DNABERT lies in its ability to achieve state-of-the-art performance even with limited labeled data, thanks to the rich representations learned during pretraining. Additionally, the model generalizes well across species: for instance, a DNABERT model pretrained on the human genome was effectively fine-tuned on mouse transcription factor datasets, demonstrating strong cross-organism transfer capabilities.

2.4 Advantages

DNABERT introduces three key advantages over previous models such as CNNs and RNNs:

- It captures long-range dependencies through a global attention mechanism.
- It generalizes well across multiple tasks.
- It requires only a small amount of labeled data for fine-tuning.

Additionally, the model's architecture allows for an intuitive interpretation of biological signals through attention maps.

3 The Sequel DNABERT-2

DNABERT-2 is the evolution of DNABERT, designed to overcome some technical limitations related to scalability, efficiency, and generalization across different species genomes. It was released in 2024 and introduces substantial changes in both the tokenization phase and the model architecture.

3.1 Key Differences from DNABERT

1. **Token Representation**: DNABERT-2 uses a more efficient tokenization method that allows for better handling of longer sequences and reduces the vocabulary size, making it more scalable to larger datasets.
2. **Model Architecture**:The architecture has been optimized for better performance on genomic data, with modifications that enhance the attention mechanism and improve the model's ability to capture long-range dependencies in DNA sequences.
3. **Training Efficiency**:DNABERT-2 incorporates techniques that allow for faster training times and reduced computational requirements, making it more accessible for researchers with limited resources.

3.2 Tokenization with Byte Pair Encoding (BPE)

DNABERT-2 instead of using fixed-length k-mers, applies Byte Pair Encoding (BPE) [Sennrich et al., 2016] to create a more flexible and efficient tokenization scheme. BPE builds tokens by merging the most frequent sequences in the training data, allowing for a more adaptive representation of DNA sequences. This approach significantly improves the model's ability to handle longer sequences and reduces the overall vocabulary size, making it more efficient for training and inference.

3.2.1 Problems of k-mer tokenization

Before diving into the advantages of BPE, it is essential to understand the limitations of k-mer tokenization, which DNABERT-2 addresses:

- **Fixed Length, Rigid Vocabulary**: k-mers have a fixed length, which can lead to inefficiencies when dealing with sequences of varying lengths. The fixed vocabulary size (e.g., 4^k for $k = 6$) limits the model's ability to adapt to DNA patterns of varying length and importance.
- **Information Leakage (in overlapping k-mers)**: When using overlapping k-mers (stride 1), adjacent tokens share many characters. This causes leakage during pretraining:
  - If a k-mer is masked, its content can be almost fully recovered from nearby k-mers.
  - This reduces the difficulty of the masked language modeling task and harms learning, because the model can "cheat".
  - Example: Given a DNA string `ATTGCACT`, 3-mers are `ATT`, `TTG`, `TGC`, `GCA`, `CAC`, `ACT`. Masking `TGC` is not hard for the model if it sees `TTG` and `GCA`, which already contain most of its content.
- **Redundant and Long Sequences**: Overlapping k-mers produce a lot of tokens:
  - A sequence of length L results in $L - k + 1$ tokens.
  - Many tokens carry almost the same information (shifted by one nucleotide).
  - This creates computational inefficiency, especially in Transformer models, which have quadratic cost in sequence length.
- **Poor Generalization (in non-overlapping k-mers)**: To reduce redundancy, some models use non-overlapping k-mers (stride = k). However, this introduces another issue: small changes in input lead to big changes in tokens.
  - Example: Sequence 1: `ACAATAATAATAATAACGG` $\rightarrow$ k-mers: `ACA`, `ATA`, `ATA`, ... Sequence 2 (shifted): `CAATAATAATAATAACGG` $\rightarrow$ k-mers: `CAA`, `TAA`, `TAA`, ... Even though the sequences are nearly identical, their tokenized forms are very different, making the model's job harder.
- **Vocabulary Size**: The vocabulary size grows exponentially with the length of k, leading to scalability issues for larger datasets.
- **Contextual Limitations**: k-mers do not capture the contextual relationships between nucleotides effectively, which can limit the model's understanding of complex genomic structures.
- **No Adaptability Across Genomes**: K-mer vocabularies are fixed, so models trained with them may not generalize well across species with different sequence distributions. This reduces performance in multi-species settings.

3.2.2 Advantages of Byte Pair Encoding (BPE)

By replacing k-mer tokenization with Byte Pair Encoding (BPE), DNABERT-2 addresses these issues effectively:

- **Variable Length Tokens**: BPE allows for variable-length tokens, which can better capture the nuances of DNA sequences.
- **Reduced Vocabulary Size**: By merging frequent sequences, BPE reduces the overall vocabulary size, making the model more efficient.
- **Improved Contextual Awareness**: BPE can capture contextual relationships more effectively, leading to better performance on downstream tasks.
- **Enhanced Sequence Understanding**: During training, the model has to learn both the length of the token and the token itself, which further improves the model's understanding of the sequence.

4 The Task of Promoter Prediction

One of the key tasks explored in the DNABERT project is **promoter prediction**: that is, identifying whether a given DNA sequence contains a promoter region. This is a fundamental step in understanding how genes are regulated [Umarov et al., 2019].

4.1 Why promoter prediction matters

As explained earlier, promoters are short regions of DNA that signal the start of a gene. They are not genes themselves, but they play a crucial role in deciding **when**, **where**, and **how much** a gene is expressed. Misregulation of promoters is often linked to diseases, including cancer, because incorrect gene activation can lead to malfunctioning cells.

Automatically detecting promoters in DNA sequences is important for:

- Mapping the regulatory landscape of the genome,
- Understanding gene expression in different cell types,
- Supporting medical research and diagnostics.

However, recognizing promoters is not easy. Promoters do not follow a simple fixed pattern. While some contain recognizable elements like the TATA box, others rely on more complex or subtle sequence signals.

4.2 How DNABERT tackles the problem

DNABERT is trained to recognize promoters by learning from examples. During fine-tuning, it is given sequences labeled as either *containing a promoter* (positive) or *not containing a promoter* (negative). The model then learns to classify new sequences based on the patterns it has seen.

Two main evaluation settings are used:

4.2.1 DNABERT-Prom-300

This setting focuses on **proximal promoter regions**: short segments of DNA around the *transcription start site* (TSS), typically from -249 to +50 base pairs relative to the TSS. These regions are critical for transcription to begin. In this setup:

- Positive examples are real promoter sequences.
- Negative examples are sequences from elsewhere in the genome with similar composition but no promoter activity.
- The model learns to classify 300 base pair sequences.

This setup is well-suited for measuring the model's ability to distinguish true promoter regions from random DNA fragments.

4.2.2 DNABERT-Prom-scan

This is a more realistic and challenging scenario. Here, the model receives **longer DNA sequences** (up to 10,000 base pairs) and must scan them to detect the location of any promoters. This task is more difficult because:

- Promoters are rare in long sequences.
- The model must avoid false positives while still detecting real promoters.

To support this, DNABERT-XL is used: a variant of the model that can process longer input sequences, thanks to architectural adjustments.

4.3 Types of promoters

In the original experiments, the promoters were grouped into two categories:

- **TATA promoters**: contain a TATA box, a common and easily recognizable pattern.
- **No-TATA promoters**: lack the TATA box and are more difficult to identify.

This separation helps evaluate how well the model generalizes across different promoter types.

4.4 Results summary (from DNABERT paper)

DNABERT significantly outperformed previous models (like CNNs and hybrid CNN-RNNs) in both promoter prediction tasks. It showed both higher accuracy and better precision and recall.

5 The GUE Benchmark

As part of the DNABERT-2 project, the authors introduced a large benchmark called *Genome Understanding Evaluation (GUE)*. This benchmark was designed to better test how well language models can understand and work with real genomic data.

GUE includes **28 classification tasks** based on DNA sequences from different species and experimental sources. These tasks cover a variety of biological signals, such as:

- Promoter regions,
- Transcription factor binding sites (TFBS),
- Enhancers,
- Splicing sites,
- And other regulatory elements.

Each task in GUE is treated as a binary classification problem: the model must decide whether a given DNA sequence contains the signal or not.

5.1 Why GUE matters

Before GUE, most models—including DNABERT—were tested on small, hand-picked datasets, often from a single species (usually human). This made it hard to compare models fairly or evaluate how well they generalize across tasks and species.

GUE solves this by:

- Providing a **standardized evaluation** across many biological functions,
- Including **multi-species data** to test generalization,
- Covering both simple and challenging classification tasks.

The GUE benchmark was used in the DNABERT-2 paper to show that the new model performs better than previous models (like DNABERT, Enformer, and Nucleotide Transformer) in **23 out of 28 tasks**, while using fewer parameters and training resources.

5.2 Scope of our project

In this project, we focus **only on the promoter prediction task** one of the classification tasks included in the GUE benchmark. This choice aligns with the original DNABERT paper and allows us to study a well-defined problem with strong biological relevance. Other tasks from GUE are outside the scope of our work.

The specific datasets used in this project are:

- `prom_300_all`: Contains human promoter sequences, with and without TATA boxes.
- `prom_300_notata`: Contains human promoter sequences without TATA boxes.
- `prom_300_tata`: Contains human promoter sequences with TATA boxes.

6 Experiments

Building on the datasets described in the previous section, we conducted a series of experiments to evaluate the capabilities of DNABERT in the task of promoter prediction. Using the original 6-mer pretrained model released by the authors, we focused on the promoter prediction subsets from the GUE benchmark.

6.1 Method

We fine-tuned DNABERT separately on each subset. We repeated each experiment with at least three random seeds to test robustness. All training settings, including batch size and learning rate, were aligned with those reported in the original DNABERT paper. We used the official implementation as the base for all our experiments to ensure consistency with the reference results.

6.2 Results

Our results were evaluated using the *Matthews Correlation Coefficient (MCC)*, consistent with the original paper. A summary of the best results obtained in our experiments is presented in Table 1, along with the expected values reported by the DNABERT authors. The comparison includes absolute and relative errors between our scores and the reference scores.

Dataset	Ours MCC	Reference MCC	Error
all	90.65	90.48	0.17 [0.19]
notata	92.80	93.05	0.25 [0.27]
tata	62.05	61.56	0.49 [0.80]

Table 1: Average MCC scores (%) versus reference values. Absolute error in percentage points; relative error in brackets.

As shown in Table 1, the MCC scores match the published benchmarks within 0.5% point, confirming that the model reproduces reliably.The small deviations observed are within acceptable margins and are likely due to the stochastic nature of training processes.

Repeating the fine-tuning with different random seeds is crucial for obtaining robust and reproducible results. While this practice is often implied, its significance becomes evident when aiming for a fair comparison with published benchmarks. In our experiments, averaging the outcomes across multiple runs helped mitigate the impact of random fluctuations, resulting in scores that faithfully mirror those in the original DNABERT study

6.3 Run details

To further support our findings, Table 2 reports the MCC scores for all individual runs, including the random seed used. These details reveal the extent of variability in the results and emphasize the importance of using multiple seeds when assessing model performance.

As visible in Table 2, the *tata* subset shows greater variability in MCC scores, likely due to its smaller size and the inherent difficulty of the classification task. Nevertheless, the overall results align closely with the published benchmarks.

Dataset	Seed	MCC (%)
all	15	91.64
all	44	90.95
all	48	91.05
all	65	89.07
all	172	90.54
notata	31	93.03
notata	42	93.13
notata	77	92.24
tata	11	63.76
tata	53	61.89
tata	80	60.48

Table 2: MCC by random seed for each dataset.

7 Conclusion

This project focused on reproducing the results of DNABERT and its successor DNABERT-2 for the task of promoter prediction in genomic sequences. By leveraging the datasets and methodologies described in the original papers, we successfully replicated the reported metrics, confirming the reliability and robustness of these models.

Through this process, we gained valuable insights into how language models, originally designed for natural language processing, can be adapted to genomic tasks. Specifically, the use of k-mer tokenization and self-attention mechanisms proved effective in capturing the complex patterns within DNA sequences. Our experiments also highlighted the importance of consistent training settings and the use of multiple random seeds to ensure reproducibility.

References

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

[Ji et al., 2021] Ji, Y., Zhou, Z., Liu, H., & Davuluri, R. V. (2021). Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. Bioinformatics, 37(15), 2112–2120.

[Sanabria et al., 2024] Sanabria, M., Hirsch, J., & Poetsch, A. (2024). Distinguishing word identity and sequence context in dna language models. BMC Bioinformatics, 25.

[Searls, 1992] Searls, D. B. (1992). The linguistics of dna. American Scientist, 80(6), 579–591.

[Sennrich et al., 2016] Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units.

[Umarov et al., 2019] Umarov, R., Kuwahara, H., Li, Y., Gao, X., & Solovyev, V. (2019). Promoter analysis and prediction in the human genome using sequence-based deep learning models. Bioinformatics, 35(16), 2730–2737.

[Zhou et al., 2024] Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R., & Liu, H. (2024). Dnabert-2: Efficient foundation model and benchmark for multi-species genome.