People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Echahid Larbi Tebessi -Tébessa- France

Faculty of Exact, Natural and Life Sciences

Department of Living Beings

MEMORY of end of study

Submitted with a view to obtaining a master's degree

Field: Exact sciences

Stream: Information system

Bert based DNA pattern recognition

Presented by:

FETNI Atika

Dr. Merzoug Soltan	MCA	***University of TEBESSA***	President
Dr. Achouri Mounir	MCB	***University of TEBESSA***	EXAMINATOR
Dr. Yahiaoui Ayoub	MCB	***University of TEBESSA***	Promoter
Dr.Khediri Abderrazak	MCA	***University of TEBESSA***	Co-Promoter

Graduation date: 08.06.2024

Universtity year: 2023-2024

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيمِ

Thanks

Au terme de ce travail, nous aimerions exprimer notre gratitude au Dieu. Nous sommes profondément reconnaissants de nous avoir accordé le courage, la volonté et la patience nécessaires pour mener à bien ce travail.

Nous tenons également à exprimer notre immense reconnaissance envers notre encadrant YAHIAOUI Ayoub et co-encadrent KHEDIRI Abd-Errazak pour son soutien inestimable, ses conseils avisés et sa présence constante. Votre expertise et votre dévouement ont été des sources d'inspiration pour nous.

Nous également remercier chaleureusement les membres du jury « MERZOUG Soltan et ACHOURI Mounir» , pour leur temps, leur expertise et leur évaluation minutieuse de notre travail.

Enfin, nous exprimons notre gratitude envers toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail. Votre soutien, vos connaissances partagées, votre assistance technique et vos encouragements ont été essentiels à notre réussite

Dedication

To my dear mother,

I am so grateful for your unconditional love. Thank you for always being there for me and supporting and encouraging me in all my endeavours. This work is dedicated to you, as a sign of the endless love I have for you. I love you to infinity.

To my dear father,

Your unwavering support and wise counsel have been an inspiration to me. You showed me the path of integrity, perseverance and determination. This achievement is dedicated to you, in appreciation of everything you have done for me.

To my wonderful sisters

To my dear brothers,

To my dear friends,

I hope this dedication is a testament to my eternal gratitude towards my family and friends who have been present at every stage of my life and encouraged me to reach new heights.

With love and gratitude,

Table des matières

Table of contents

List of figures

List of tables

List of abbreviations

AdamW	Adam with Weight Decay
Bert	Bidirectional Encoder Representations from Transformers
BioBERT	Bidirectional Encoder Representations from Transformer for Biomedical Text Mining.
CLS	Classification
DNA	Deoxyribonucleic Acid
GPUs	Graphics Processing Units
LLMs	large language models
LRMs	language representation models
MLM	Masked Language Modeling
NGS	Next Generation sequencing
NLP	Natural Language Processing
NSP	Next Sentence Prediction
PAD	padding
PCR	Polymerase Chain Reaction
RFLP	Restriction Fragment Length Polymorphism
RNA	Ribonucleic Acid
SciBERT	Scientific Bidirectional Encoder Representations from Transformer
SEP	separator.
TPUs	Tensor Processing Units
EHRs	electronic health records
NER	Named Entity Recognition
SVM	like supervised learning algorithms
MLM	masked language modeling
NSP	next sentence prediction
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Network

ملخص:

يُعتبر فهم لغة الحمض النووي غير المشفر موضوعًا رئيسيًا في البحوث الجينومية. إن الشفرة التنظيمية الجينية معقدة للغاية بسبب وجود تعدد المعاني والعلاقات الدلالية البعيدة، والتي كثيراً ما تفشل مناهج المعلوماتية السابقة في التقاطها.

ولمعالجة هذه الصعوبة، استخدمنا DNABERT، وهو تمثيل مشفر ثنائي الاتجاه فريد من نوعه مدرب مسبقًا يلتقط الفهم الشامل والقابل للنقل لتسلسلات الحمض النووي الجينومي استنادًا إلى سياقات النيوكليوتيدات العلوية والسفلية. قمنا بمقارنة DNABERT بالأنظمة الأكثر شيوعًا للتنبؤ بالعناصر التنظيمية على مستوى الجينوم ووجدنا أنه أسهل استخدامًا وأكثر دقة وكفاءة. لقد أثبتنا أن نموذج محولات واحد مدرب مسبقًا يمكن أن يصل إلى أحدث أداء في التنبؤ بالمحفزات ومواقع التقسيم ومواقع ربط عامل النسخ بعد ضبط بسيط باستخدام بيانات متواضعة ذات علامات خاصة بالمهمة. وعلاوة على ذلك، يسمح DNABERT بالعرض المباشر للأهمية على مستوى النوكليوتيدات والعلاقات الدلالية داخل تسلسلات المدخلات، مما يؤدي إلى تحسين قابلية التفسير وتحديد أكثر دقة لزخارف التسلسل المحفوظة وإمكانيات المتغيرات الجينية الوظيفية.

Abstract :

Understanding the language of non-coding DNA is a major topic in genomic research. Gene regulatory code is extremely complicated due to the presence of polysemy and distant semantic relationships, which earlier informatics approaches frequently fail to capture.

To address this difficulty, we used DNABERT, a unique pre-trained bidirectional encoder representation that captures global and transferable comprehension of genomic DNA sequences based on up and downstream nucleotide contexts. We compared DNABERT to the most popular systems for predicting genome-wide regulatory elements and found that it was easier to use, more accurate, and more efficient. We demonstrate that a single pre-trained transformers model can reach state-of-the-art performance in the prediction of promoters, splice sites, and transcription factor binding sites following simple fine-tuning using modest task-specific labeled data. Furthermore, DNABERT allows for direct display of nucleotide-level significance and semantic relationships within input sequences, resulting in improved interpretability and more accurate identification of conserved sequence motifs and functional genetic variant possibilities.

Key words: DNA, BERT, DNABert, LRM, NLP.

Résumé :

La compréhension du langage de l'ADN non codant est un sujet majeur de la recherche génomique. Le code de régulation des gènes est extrêmement complexe en raison de la présence de polysémie et de relations sémantiques distantes, que les approches informatiques antérieures ne parviennent souvent pas à saisir.

Pour résoudre cette difficulté, nous avons utilisé DNABERT, un codeur bidirectionnel unique et pré-entraîné qui permet une compréhension globale et transférable des séquences d'ADN génomique en se basant sur les contextes des nucléotides en amont et en aval. Nous avons comparé DNABERT aux systèmes les plus populaires pour prédire les éléments régulateurs à l'échelle du génome et avons constaté qu'il était plus facile à utiliser, plus précis et plus efficace. Nous avons démontré qu'un seul modèle de transformateur pré-entraîné peut atteindre des performances de pointe dans la prédiction des promoteurs, des sites d'épissage et des sites de liaison des facteurs de transcription après un réglage fin simple utilisant des données étiquetées modestes et spécifiques à une tâche. En outre, DNABERT permet l'affichage direct de la signification au niveau des nucléotides et des relations sémantiques au sein des séquences d'entrée, ce qui améliore l'interprétabilité et l'identification plus précise des motifs de séquences conservées et des possibilités de variantes génétiques fonctionnelles.

Mots clés : ADN, BERT, DNABert, LRM, NLP.

Introduction

Introduction general

In the intricate world of molecular biology, deciphering the language of DNA holds the key to unlocking the mysteries of life itself. DNA, the blueprint of all living organisms, encodes the fundamental instructions that govern their growth, development, and functioning. Yet, within this seemingly simple code lies a complexity that has long intrigued and challenged researchers. Unraveling the intricate patterns and regulatory mechanisms embedded within DNA sequences has been a central focus of biological research for decades.

At its core, DNA is a sequence of nucleotides that encodes information in a manner analogous to the alphabet of a written language. Just as words and sentences convey meaning in human communication, DNA sequences dictate the structure and function of proteins, the molecular machines essential for life. However, the language of DNA extends beyond protein-coding regions. Non-coding regions of the genome, once dismissed as "junk DNA," are now recognized as critical players in gene regulation and cellular processes. Deciphering this regulatory code, with its intricate patterns and nuanced signals, poses a formidable challenge.

Traditional methods of DNA analysis, while invaluable, are often limited in their ability to capture the full complexity of genomic data. The sheer size of the genome, combined with the multitude of functional elements and regulatory regions, presents challenges in accurately identifying and interpreting relevant patterns. Moreover, the dynamic nature of gene regulation, influenced by factors such as cell type, developmental stage, and environmental cues, adds another layer of complexity to the puzzle.

Amidst these challenges, recent advances in computational biology offer new avenues for exploring the language of DNA. Natural Language Processing (NLP), a field originally developed to analyze and understand human language, has found surprising applicability in genomic research. Models like BERT (Bidirectional Encoder Representations from Transformers), originally designed for processing text data, have shown remarkable effectiveness in capturing complex patterns and relationships within DNA sequences.

Building upon this foundation, we propose DNABERT—a specialized model designed to tackle the unique challenges of DNA pattern recognition. By harnessing the power of BERT and adapting it to the specific characteristics of genomic data, DNABERT aims to revolutionize our ability to decipher the regulatory language encoded within the genome. With its ability to capture contextual information, model long-range dependencies, and handle data-scarce scenarios, DNABERT holds the promise of unlocking new insights into gene regulation, disease mechanisms, and personalized medicine.

In this work, we embark on a journey to explore the language of DNA, from its basic elements to its intricate regulatory networks. Through a combination of theoretical insights, computational methods, and experimental validation, we seek to unravel the mysteries hidden within the genomic code. By bridging the gap between biology and artificial intelligence, we aim to advance our understanding of life's most fundamental processes and pave the way for transformative discoveries in biomedicine and beyond.

Chapter 1:

Context and problematic

1- Introduction to DNA Pattern Recognition

In this section, we delve into the complexities of recognizing DNA patterns emphasizing its role, in understanding makeup, gene activity, and evolutionary connections. We start by exploring the basics of DNA highlighting the significance of non-coding areas that make up most of the genome and play crucial roles in controlling genes. Then we discuss methods used for identifying DNA patterns outlining their strengths and limitations when dealing with datasets and intricate genetic structures. We also introduce BERT (Bidirectional Encoder Representations from Transformers) a model for natural language processing. Discuss its potential in analyzing complex biological data. Additionally, we introduce DNABERT, a version of BERT designed for studying DNA sequences including coding regions. Finally, we summarize how DNABERT advancements contribute to studies, medical diagnostics, and bioinformatics by capturing meanings and distant connections,
within gene regulatory information. This holistic approach demonstrates the game-changing impact of using methods to decipher the language embedded in DNA.

DNA pattern recognition

DNA pattern recognition is the identification of certain patterns or sequences of a DNA molecule. These running variables might represent genes, regulatory elements, or functional portions of the genome. In domains such as genomics, molecular biology, and bioinformatics, we need DNA pattern recognition to comprehend genetic structure, gene expression, and evolutionary links.

2- Structure and Composition of DNA

Deoxyribonucleic acid (DNA): the heredity material found in humans and all living organisms. It is a double-stranded molecule and has a unique twisted helical structure.

DNA is made up of nucleotides, each nucleotide has three components: a backbone made up of a sugar (Deoxyribose) and phosphate group and a nitrogen-containing base attached to the sugar.

Each strand has many nucleotides or says numerous sugars, a phosphate group, and nitrogenous bases. These nitrogenous bases are complementary to the other strand's nitrogenous base to maintain helical symmetry.

Each base pairs are bonded through Hydrogen bonding. These nitrogenous bases are Adenine (A), Guanine (G), Cytosine (C), and Thymine (T), A is complementary to T, and G to C. These bases are responsible for storing the genetic information. Most DNA is located in the cell nucleus and is called nuclear DNA, however, a small amount of DNA is also located in mitochondria, and so is referred to as mitochondrial DNA. [2]

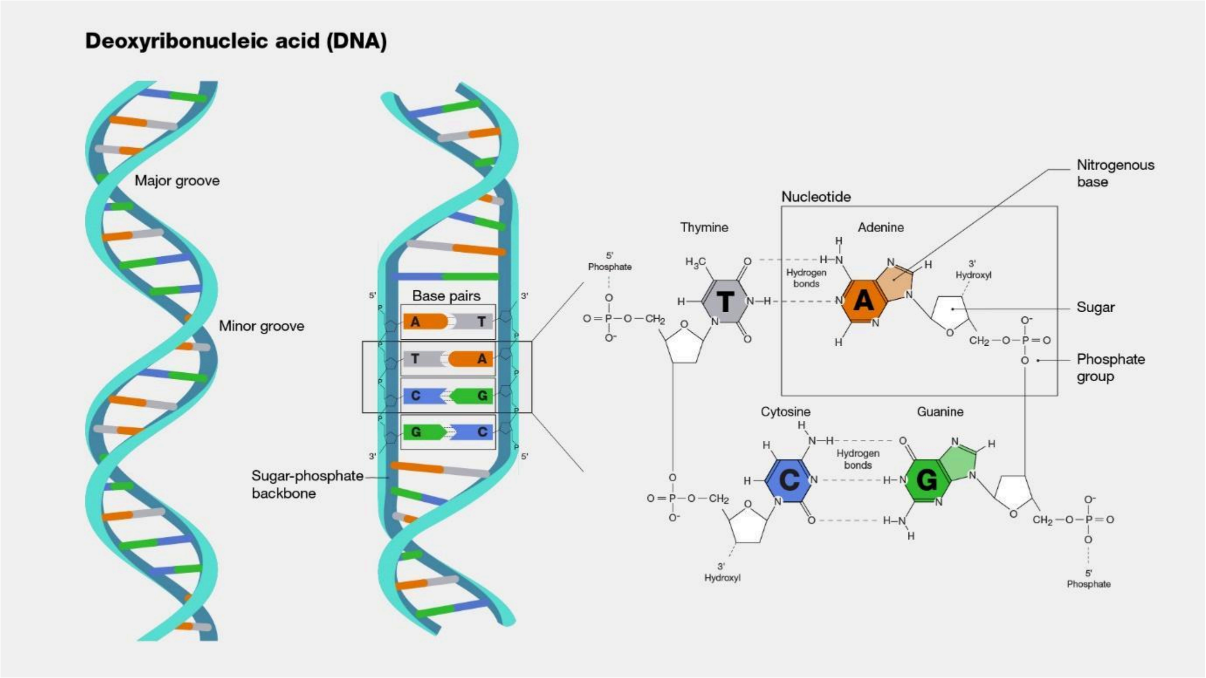


Figure 1: Deoxyribonucleic Acid (DNA) [2]

1.2- Properties of DNA (Deoxyribonucleic acid)

DNA is made up of two helical strands that are coiled around the same axis. If coiled from the right it is known as right-handed helices DNA and if coiled from the left then it is known as left-handed helices. However, the right-handed helices DNA is the most stable, and thus the structure of it is to be referred to as the standard.

The two chains of helices run antiparallel to each other. Thus, one strand runs from 5’ to 3’ and another strand runs from 3’ to 5’.

Both the strands denature on heating and can renature or say hybridize on cooling. However, the temperature at which these strands are separated permanently is referred to as melting temperature and varies according to the specific sequence of DNA.

For instance, the region of higher concentration of C-G has a higher melting temperature because these bases are bonded with three hydrogen bonds, which require more energy to break than the region of higher concentration A-T which are bonded only with two hydrogen bonds.

These nitrogenous bases store genetic information and thus encode for amino acids which give rise to proteins. [2]

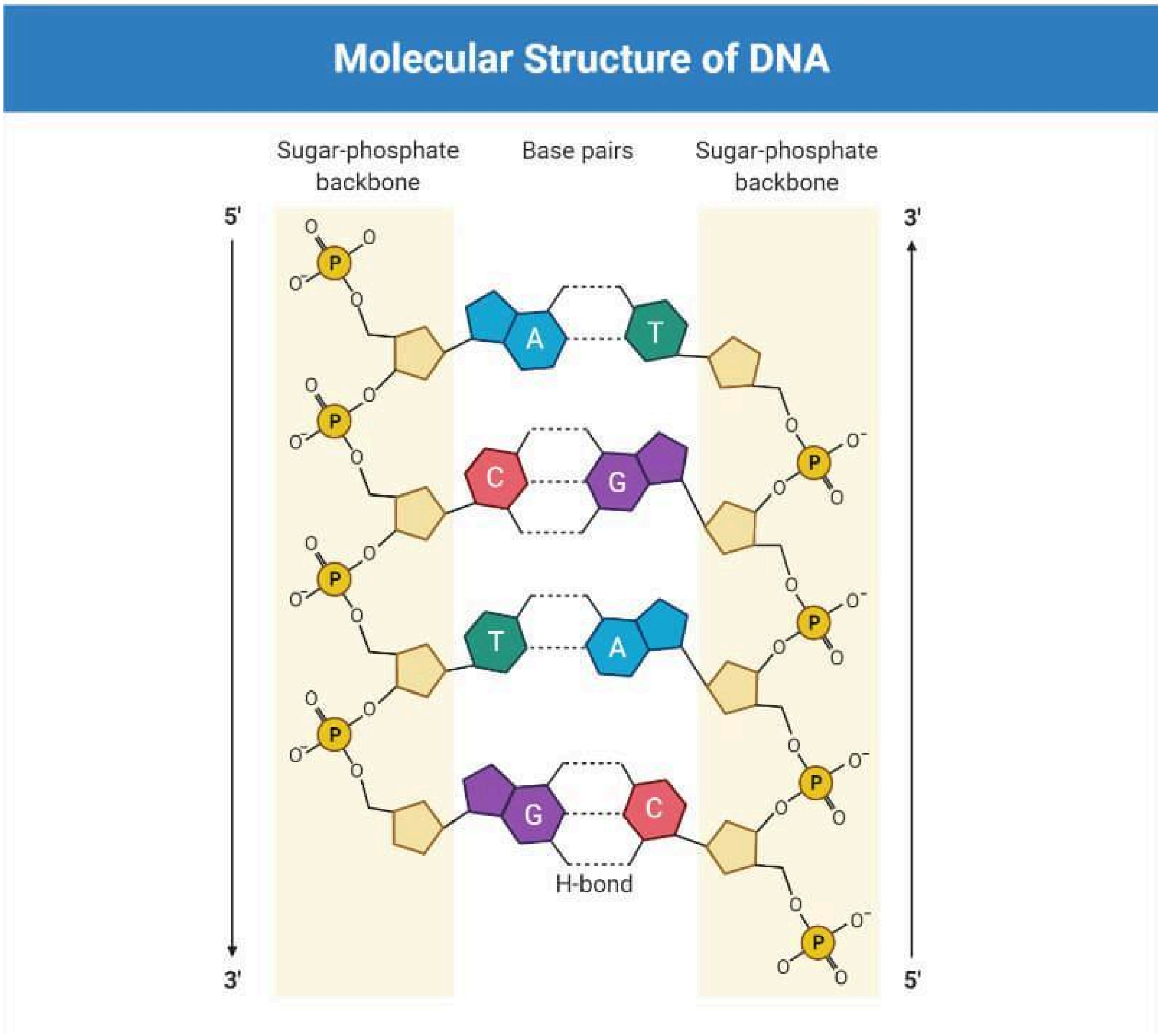


Figure 2: Structure and Composition of DNA. [2]

1.3- Types of DNA and Their Roles

DNA, or genetic code, is split into different types based on what it is made of, its job, or where it is in a living thing's plan. The most common types are viral DNA, plasmid DNA, coding DNA (exons), non-coding DNA (introns, rules), DNA from mitochondria, chloroplasts, and the genome. The genomic DNA of an organism contains the traits required for growth and survival. Creatures require mitochondrial DNA to function, which is found in their small organs. Plants and some water creatures have chloroplast DNA and that helps with photosynthesis. Bacteria have small round DNA pieces called plasmids, which can make more of themselves, and hold helpful traits.

Non-coding DNA corresponds to the portions of an organism's genome that do not code for amino acids, the building blocks of proteins. Some non-coding DNA sequences are known to serve functional roles, such as in the regulation of gene expression, while other areas of non-coding DNA have no known function.[4]

Non-coding DNA:

Non-coding DNA corresponds to the portions of an organism's genome that do not code for amino acids, the building blocks of proteins. [5]

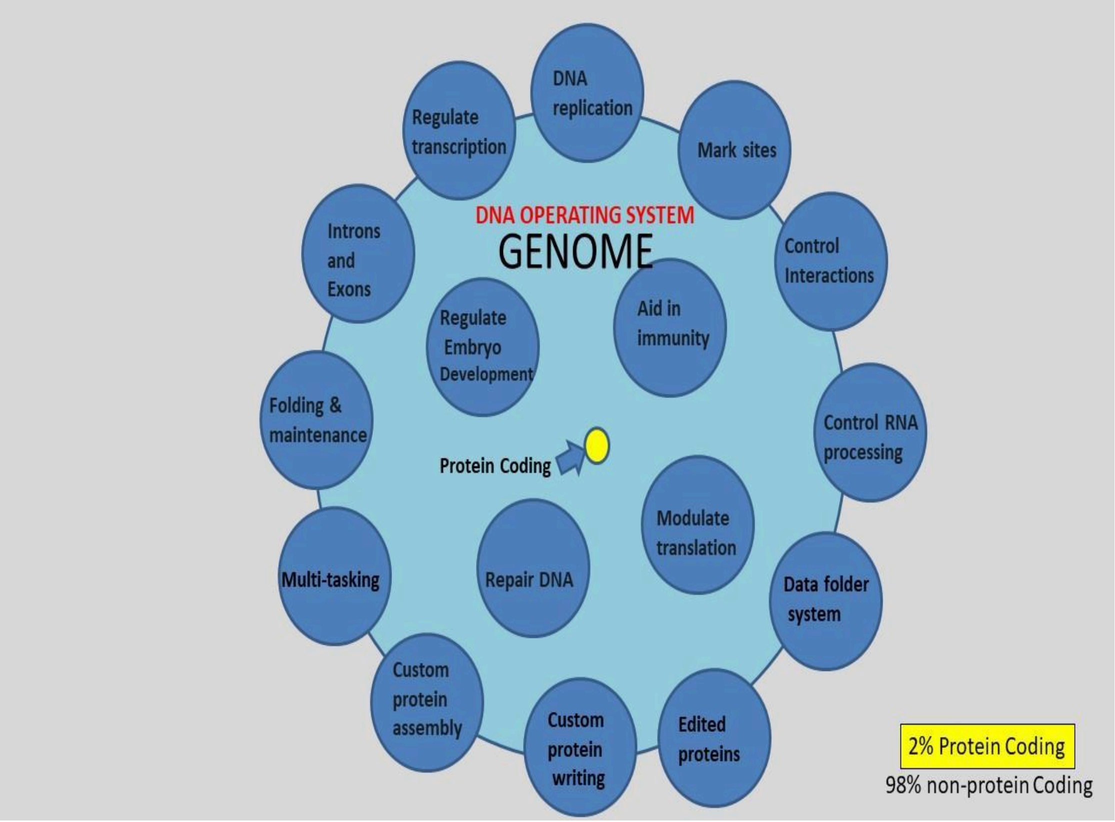


Figure 3: DNA operating system. [5]

Noncoding DNA makes up about 98.5% of the total DNA. While it was previously thought to have no function, newer information is beginning to shed light on the many functions of this mass of DNA. It is involved in the cutting and splicing of large amounts of DNA, is involved in transposon reassembly, genome rearrangements, and the production of small RNAs, some of which may serve as a source for new exons. It is also possible that noncoding DNA was used as a source of new genes needed for adaptation or for functions during evolution. These ideas are summarized in Figure 4. [5]

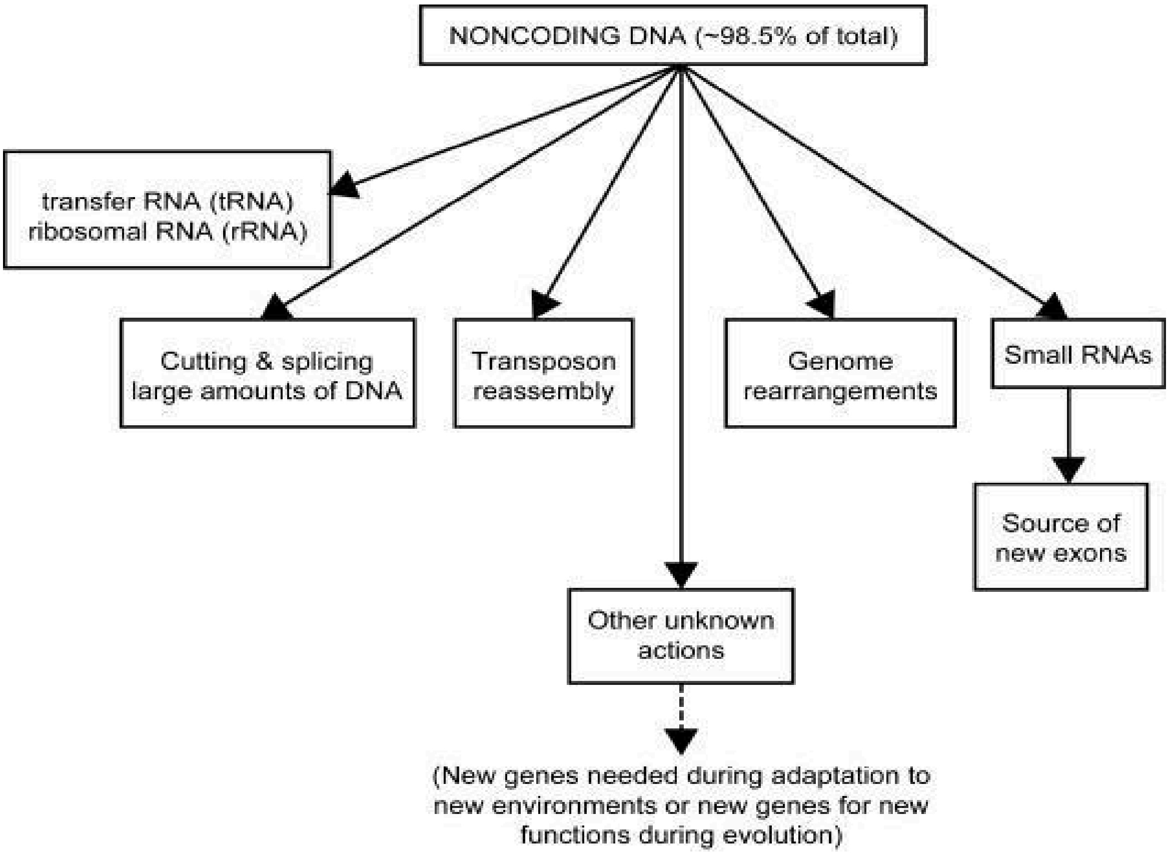

NONCODING DNA (~98.5% of total)
transfer RNA (tRNA)
ribosomal RNA (rRNA)
Cutting & splicing large amounts of DNA
Transposon reassembly
Genome rearrangements
Small RNAs
Source of new exons
Other unknown actions
(New genes needed during adaptation to new environments or new genes for new functions during evolution)

Figure 4: Function of Non-coding DNA. [6]

3- The significance of DNA pattern recognition in biomedical research.

DNA patterns are just like a unique set of fingerprints for the scientists to find out a target gene within the network of genomes playing a role in various attributes and diseases. These patterns can help us understand genetic variation, which is regarding could help researchers learn more about variations in physical traits like drug reactions or potential risks for diseases. It informs how genes are controlled and may indicate how changes in regulatory regions may cause diseases. Comparative genomics can identify conserved patterns/sequences, informing on disease mechanisms. From early detection and diagnosis to future options for treatment, patterns in DNA can be diagnostic flags that herald the presence of or risk of disease and offer huge potential as biomarkers that could be used to tell us if we are in the early stages of disease before we ever even feel unwell.

Understanding DNA patterns associated with diseases aids in drug discovery and development, identifying specific DNA sequences linked to medication metabolism or disease pathways. DNA sequences are used in genetic profiles. [7]

Overview of the role of DNA sequences in genetic research, medical diagnosis, and bioinformatics.

- **Genetic Research:**

Genome Mapping: DNA sequences are used to map whole genomes, finding genes, regulatory elements, and other functional regions**.**

Mutation Analysis: Researchers examine DNA sequences to check if they include mutations that cause diseases or genetic anomalies, which aids in understanding genetic systems and inheritance patterns.

Evolutionary Studies: By comparing DNA sequences from various species, researchers may investigate evolutionary connections and trace organismal evolution. [8]

- **Medical diagnosis:**

Genetic testing: uses DNA sequencing techniques to identify genetic abnormalities, forecast disease risk, and inform treatment recommendations.

Pharmacogenomics: the analysis of DNA sequences to predict individual drug reactions, allowing for customized medical techniques.

Cancer Genomics: DNA sequencing identifies genetic mutations that cause cancer, which aids in cancer diagnosis, prognosis, and therapy selection. [8]

- **Bioinformatics:**

Sequence Analysis: Bioinformatics tools scan DNA sequences to identify genes, regulatory regions, and structural characteristics, revealing information on gene function and control.

Comparative Genomics: Comparing DNA sequences from different organisms can identify conserved regions, evolutionary connections, and functional components. [5]

4- Challenges in DNA Pattern Recognition

Studying DNA sequences can be quite intricate due, to several factors to take into account. Here are a few of the complexities linked with examining DNA sequences:

4.1- Data Quantity: The technologies used for DNA sequencing produce volumes of data. Handling, storing, and processing this data can pose a challenge, necessitating computing resources, and effective algorithms. [9]

4.2- Data Accuracy: DNA sequencing data commonly contain mistakes like sequencing artifacts, errors in base calling, or biases in PCR amplification. Ensuring the accuracy of the data is crucial for analysis.

4.3- Genetic Heterogeneity: Variations in genetic makeup result in significant variations in DNA patterns among individuals, populations, and species. Dealing with diverse genetic origins poses challenges for accurate pattern detection. [10]

4.4- Complex Trait Architecture: Many traits, especially complex ones like human height or susceptibility to diseases, are influenced by a wide range of genes and environmental factors. To pinpoint the exact DNA sequences associated with these traits, sophisticated analytical techniques must capture complex genetic connections. [12]

4.5- DNA Sequence Clustering: Understanding biological sequence functions requires cluster analysis based on sequence similarity. Techniques that take into account the local and global aspects of the adoption of new distance measurements are required.[12]

4.6- Epigenetic Modifications: DNA methylation and histone modifications are two examples of epigenetic alterations that are important for controlling and expressing gene expression. But because these changes don't affect the underlying DNA sequence, it's challenging to identify them with conventional sequencing techniques. To precisely characterize epigenetic alterations, specialized methods are needed, such as bisulfite sequencing for DNA methylation research.[13]

4.7- Low-frequency DNA variations: Low-frequency DNA variants present at low quantities in a sample may be difficult for current sequencing technology to identify. Understanding genetic variation among populations and the course of disease may be aided by these variants. [14]

5- Limitations of Traditional Methods.

These are some important considerations when utilizing conventional methods of analyzing DNA patterns, which have greatly aided scientific advancements but have limitations, especially when it comes to fully capturing the complexity of DNA.[15]

5.1- Sanger sequencing

Read Length: Capable of only sequencing about 1,000 bases at once. When studying sections that are repeated, insertions and deletions, and other alterations that cover greater distances, this might provide challenges.

limited resolution: leading to incomplete or inaccurate characterization of genetic diversity due to the overlooking of subtle variations in DNA sequences or minor allele frequencies.

Bias and Error Rates: biases or errors, affecting DNA patterns' precision and dependability, particularly in high GC concentration areas or other sequences

poor Sensitivity: Rare mutations and genetic ancestry may be difficult to accurately identify due to genetic markers' poor sensitivity for identifying low-frequency variations, particularly in diverse populations with complicated backgrounds. [16]

5.2- NGS

Data analysis: Deciphering the vast amounts of information produced by NGS can be challenging, especially when trying to find uncommon variants or examine structural differences.

Sensitivity to Sequence Length: Conventional approaches may be unable to examine sequences of varied lengths, particularly longer sequences such as whole genomes. This may result in biased or erroneous complexity estimations, especially if the algorithms are not designed to adequately manage extended sequences.

Assumption of Stationarity: Some older approaches may presume that DNA sequences are uniform or stable, ignoring evolutionary constraints or differences in complexity across genomic areas. This oversimplification may lead to a misunderstanding of the dynamics of DNA complexity. [17]

Quantitative constraints: include gene expression measurement due to sequencing depth, library preparation biases, and mapping efficiency, as well as discovering allele-specific expression in areas with high similarity or low levels.

5.3- DNA microarray

Inadequate Sequence Composition Bias Handling: Sequence composition bias can contribute to inaccurate complexity estimates in regions with repetitive sequences or uneven nucleotide content since existing methods may not fully take this into account. [18]

Restricted Dynamic Range: This might result in erroneous quantification and false negatives.

Background noise and cross-hybridization: under these scenarios, probes bind nonspecifically to sequences that are similar but not identical, producing false-positive results and inaccurate gene expression measurements. Furthermore, the signal-to-noise ratio is lowered by background noise from nonspecific binding and other sources, making it more challenging to discern genuine signals.

produced based on known sequences or gene annotations: contain biased annotations that lead to restricted coverage and the removal of less-studied genes. Their capacity to fully capture DNA patterns may be constrained by restrictions on sequence availability, length, and specificity of the probe.

5.4- RFLP

Labor-intensive and time-consuming: RFLP analysis is a labor-intensive and time-consuming process that is more costly and slower than more recent approaches. It also requires expert technicians and includes many steps.

Limited resolution: Restriction enzyme recognition sites cause changes in DNA fragment sizes. This approach, which has low resolution, may not fully capture the range of genetic diversity.

Limited multiplexing capacity: is less effective for concurrently evaluating several DNA samples due to its limited multiplexing capabilities. [19]

5.5 Sequence Alignmen

Inability to Completely Capture Complicated Structural Qualities: Complex structural qualities in DNA can be challenging to fully capture using conventional methods due to nested repeats, overlapping motifs, and complicated secondary structures like hairpins or stem loops. This may result in oversimplified representations of DNA complexity that exclude important functional elements. [20]

Gap Penalties: Sequence alignment algorithms use gap penalties to account for insertions and deletions in DNA sequences, but determining the optimal penalty can be challenging.

Ambiguity in Biological processes: like mutations, insertions, deletion, and rearrangements can create complex patterns that traditional alignment methods cannot capture.

Homology vs. Analogy: homology focuses on identifying homologous regions, while analogy may overlook similar functions but different origins.

5.6 Sequence Motif Discovery

Limited Resolution in Identifying Functional Components: Splice sites, regulatory regions, and protein-binding motifs are examples of the functional components of DNA sequences that may be difficult to accurately identify and characterize using traditional methods. This may complicate efforts to comprehend the biological significance of particular genetic areas. [21]

Complexity and noise in the backdrop: complex and noisy, with recurring components and background patterns that might mask genuine motifs and confuse motif-finding algorithms.

Variability and degeneracy: DNA motifs are variable, maintaining functional similarity, but this can make it difficult for motif identification algorithms to identify them, which can result in oversimplification or the omission of important variations. [22]

6- Motivation for BERT-Based Approach

To understand and describe complex patterns and connections within biological information, the study shows the potential of language representation models (LRMs) in managing complicated biological data.

6.1- Explanation of the potential language representation models in handling complex biological data.

Similar to large language models (LLMs), LRMs are capable of effectively identifying important aspects of biological data, which helps with tasks such as gene function annotation, splice site identification, and modification prediction. [23]

Because of its adaptability, LRMs may be used in many scientific fields, including proteomics, single-cell research, and genomics. They can use strategies like positional encoding and variable gene selection to manage batch effects and sparsity in biological datasets.

Via modality tokens and attention mechanisms, LRMs may combine multi-omics data from several technologies at the single-cell level. They may represent intricate linkages seen in biological data by using graph transformer networks, which can capture both close-knit relationships and distant dependencies. LRMs' performance and generalizability may also be improved by regularly training and fine-tuning them on fresh data.

6.2- BERT (Bidirectional Encoder Representations from Transformers)

BERT is a breakthrough in natural language processing based on deep learning that has shown great promise in biomedical applications, especially in managing intricate biological data.

Using BERT, a deep machine learning model, large biomedical research datasets can be handled. Medical literature is one area in which it is helpful because of its bidirectional training approach that helps to learn how to distinguish phrases with different meanings. BERT’s architecture makes it perfect for pre-training on big corpora such as PMC and PubMed, as it is designed to handle and learn from big corpora quite effectively. The system can transmit information, especially for jobs with little data, thanks to the capabilities of transfer learning in BERT. Because of its very flexible design, BERT may be tweaked so that it does well across a range of NLP tasks without substantial changes being made. Example tasks include Question Answering, Relation Extraction, and Named Entity Recognition which has shown improved performance compared to other models. The transformer nature of BERT allows parallel processing of texts and enables management of long complicated entities while being open-ended for new information. [24]

6.3- Justification for adapting BERT for DNA pattern recognition tasks in the biomedical domain.

The BioBERT system is a strong baseline for DNA sequence processing tasks due to its ability to handle biomedical texts better than other transformer models, the reason BERT has been applied to DNA sequence tasks in the biomedical domain is that there are several relevant reasons:

BERT's Pre-trained Representations: The pretraining process that BERT has undergone allows it to capture a wealth of semantic and contextual information from a large volume of text. This preparatory phase enables BERT to comprehend the complex language used in biomedical text, and by extension, DNA sequences.

Fine-tuning Capability: BERT has the capability of being fine-tuned for specific downstream tasks, and as such, it can learn the subtleties and patterns in task-based examples. The fine-tuning stage is particularly important for DNA sequence modeling as it allows the model to specialize in understanding DNA sequence patterns that are relevant for biomedical applications.

Utilization of Context: BERT utilizes a bidirectional context, and the architecture of BERT was designed in such a way that it can comprehend the relationship between DNA and the surrounding context. Context is important in DNA sequence tasks as the meaning of a DNA sequence can be very different depending on the specific location within a genome or a biological pathway. [25]

State-of-the-art Performance: The text probably cites evidence or publications that have shown that BERT outperforms several natural language processing tasks, including in the biomedical domain, and this evidence, along with the model, would have justified this text claiming that BERT has the potential to perform very well in learning complex patterns, and DNA sequence tasks fit into those types of tasks.[25]

Biomedical Adaptations Available: There are biomedical adaptations of BERT that the text alludes to, such as BioBERT, SciBERT, and BlueBERT, which are arguably models that Fine-Tune BERT, with the anticipation that the fine-tuning will continue to capture biomedical-specific domain language or text, and make BERT a very relevant model for DNA sequence tasks. [25]

Our problematic:

Deciphering the language of DNA for hidden instructions has been one of the major goals of biological research while the genetic code explaining how DNA is translated into proteins is universal, the regulatory code that determines when and how the genes are expressed varies across different cell types and organisms that have distinct functions and activities in different biological contexts,

the language of non-coding DNA is one of the fundamental problems in genome research because gene regulatory code is highly complex due to the existence of polysemy and distant semantic relationship, which previous informatics methods often fail to capture especially in data-scarce scenarios. for that, we propose to use the Bidirectional Encoder Representations from the Transformers model for DNA language for this challenge cause is specifically designed for analyzing DNA sequences. and we expect better results than previous works.

7- Conclusion

This chapter examined DNA pattern recognition, a field that could lead to revolutionary discoveries, and an understanding of the content and structure of DNA sequences that are used for genetic studies, testing, and cancer genomics. However, dealing with large amounts of genetic data presents challenges, such as sequencing data and understanding the effects of genes and the environment. Despite these hurdles, researchers continue to make progress in data integration, pattern mining, and epigenetic analysis, and in the next chapter we will talk about how The effort to unravel DNA patterns continues, and the potential of Natural Language Processing (NLP) approaches to decipher DNA sequences is explored along with its application to genomics research. we will contrast different strategies and emphasize the need for specialized models such as DNABERT.

Chapter 2:
State of the art

1- Introduction

As we move into this chapter, it turns out that Natural Language Processing (NLP) techniques are revolutionizing the way we understand DNA patterns in genomics research. It was found by searching through academic databases and merging recent articles that NLP models are being used to decode the complexities of DNA sequences. We discover through making a comparison of different approaches their strengths as well as weaknesses. Besides, we define why there need for specialized models such as DNABERT to overcome the drawbacks of traditional methodologies thereby enabling researchers to probe deeper into genetic data. Above all, our purpose is to demonstrate how important NLP has become in genetics studies and how targeted models will boost progress in this particular area going forward.

2- Natural Language Processing in Biomedical Domain

Figure 5: NLP in healthcare [36]

Natural language processing and text mining ("BioNLP") are branches of biomedical informatics that deal with processing prose, whether in journal articles or electronic medical records, for purposes such as extracting information, cohort retrieval, and other uses. They are made difficult by the rampant presence of ambiguity and variability in human-produced prose. In addition, biomedical text poses special challenges on several levels. Machine learning and rule-based approaches both have a long history in biomedical natural language processing, and hybrid systems are common. Much progress has been made in biomedical natural language processing and text mining in recent years.

- Healthcare professionals and researchers, in the field can access data from extensive collections of biomedical literature using natural language processing (NLP) tools, for search and retrieval. Search and question-answering systems tailored for the sector contribute to knowledge exploration and decision-making in this field. These systems interpret user queries and retrieve pertinent material from scientific databases or literature.

NLP is involved in pulling and studying details, from sources like papers, medical records, and biomedical databases. Through the use of NLP methods structured data is extracted from texts to pinpoint entities such, as genes, proteins, illnesses, and medications. This information extraction process is vital for tasks such as literature mining, clinical decision support, and pharmacovigilance.

NLP enables researchers to sift through literature to uncover new insights, patterns, and relationships. It automates the extraction of concepts, relationships, and classifications from data to enhance ontologies. clinical natural language processing (NLP) is employed for tasks, like information extraction, medical coding, and clinical decision support by parsing narratives from EHRs, physician notes, and radiology reports.

To develop models, for purposes such as diagnosing illnesses predicting patient outcomes creating medications, and identifying events, machine learning, and predictive modeling are integrated with techniques from natural language processing. Typically natural language processing (NLP) plays a role, in extracting organizing, and analyzing data from written sources. This contributes significantly to research endeavors, clinical environments, and healthcare-related applications. [27]

2.1- The application of NLP techniques in processing biomedical text data.

Natural language processing (NLP) methods have made a significant contribution to the analysis of biological text data, advancing research, clinical practice, and healthcare analytics. Some important areas where natural language processing (NLP) is used in biomedical text processing are as follows:

Named Entity Recognition (NER): is an aspect of text processing that plays a role, in tasks such as extracting information and understanding text. Its primary focus is on identifying types of named entities within a collection of documents. Through the utilization of machine learning models like CRFs, BiLSTM CRF and pre-trained language models such as BERT and BioBERT fresh texts are. Labeled to recognize named entities, like viruses, proteins, DNA, RNA and different types of cells [38]

Relation Extraction: Identify and extract correlations between items, and linkages by using techniques like co-occurrence analysis, dependency parsing, and deep learning models like recurrent and convolutional neural networks (RNNs), applications include building scientific knowledge graphs, helping in drug development, and comprehending the molecular processes underlying sickness, text mining system performs competitively for the identification of gene-disease, drug-disease, and drug-target associations. [29]

Text Classification. Text Classification: Categorize biomedical texts using techniques like supervised learning algorithms (like SVM, random forests) and deep learning models (like transformers) into predetermined classifications like research papers, topic classification for literature databases, clinical reports, or specific illness categories for automated screening of relevant research and systematic reviews. Machine-learning-based text classification is one of the leading research areas and has a wide range of applications, which include spam detection, hate speech identification, reviews, rating summarization, sentiment analysis, and topic modeling. Widely used machine-learning-based research differs in terms of the datasets, training methods, performance evaluation, and comparison methods used. [30]

Information retrieval: Enhance PubMed and other search engines, expedite literature reviews, and promote evidence-based medicine by employing vector space models, semantic search with embeddings, and query expansion to enhance user queries' understanding and document relevance rating in biomedical databases because Literature search is a routine practice for scientific studies as discoveries build on knowledge from the past. [31]

Summarization: automatically produces a summary containing important sentences and includes all relevant important information from the original document. An extractive summary is heading towards maturity and now research has shifted towards Applying strategies to provide succinct summaries of clinical notes, abstracts, and biomedical texts, and to compile literature digests, summarize health records, and facilitate quick comprehension of fresh research results, extractive summarization—which involves choosing important sentences—and abstractive summarization—which involves synthesizing new sentences—are frequently performed. [32]

Clinical Text Analysis: To manage patient data, provide predictive analytics for patient outcomes, and improve clinical decision support systems, clinical notes and electronic health records (EHRs) are processed using methods like rule-based systems, machine learning classifiers, and neural networks trained on clinical text data. These methods extract patient information, medical history, and treatment outcomes. [33]

Semantic Similarity and Concept Normalization: determine how similar biological concepts are to one another and standardize terminology by employing established ontologies, such as embedded models and ontology mapping tools (such as word2vec and BioWordVec), with the objectives of strengthening data integration programs, encouraging health information system interoperability, and harmonizing medical data from various sources**.** [34]

2.2- the challenges and opportunities of applying NLP in DNA pattern recognition.

Natural Language Processing (NLP) contributes to DNA pattern detection by processing and evaluating textual data associated with DNA sequences. However, some challenges occur when utilizing NLP techniques to identify DNA patterns.

Complexity of Biological Data: DNA sequence analysis has become an important part of modern molecular biology. The DNA sequence is composed of four nucleotide bases—adenine (abbreviated A), cytosine (C), guanine (G), and thymine (T) in any order. With four different nucleotides, 2 nucleotides could only code for a maximum of 42 amino acids, but 3 nucleotides could code for a maximum 43 amino acids**.** These sequences are distinct from text data, but they have properties such as changing durations, recurring patterns, and intricate structural relationships. They also carry a plethora of information, often with little variations that are critical to understanding biological processes. Unlike real language, where meaning is frequently derived from well-defined words and grammar, DNA sequences lack a clear linguistic structure, making it difficult to apply NLP models directly. [35]

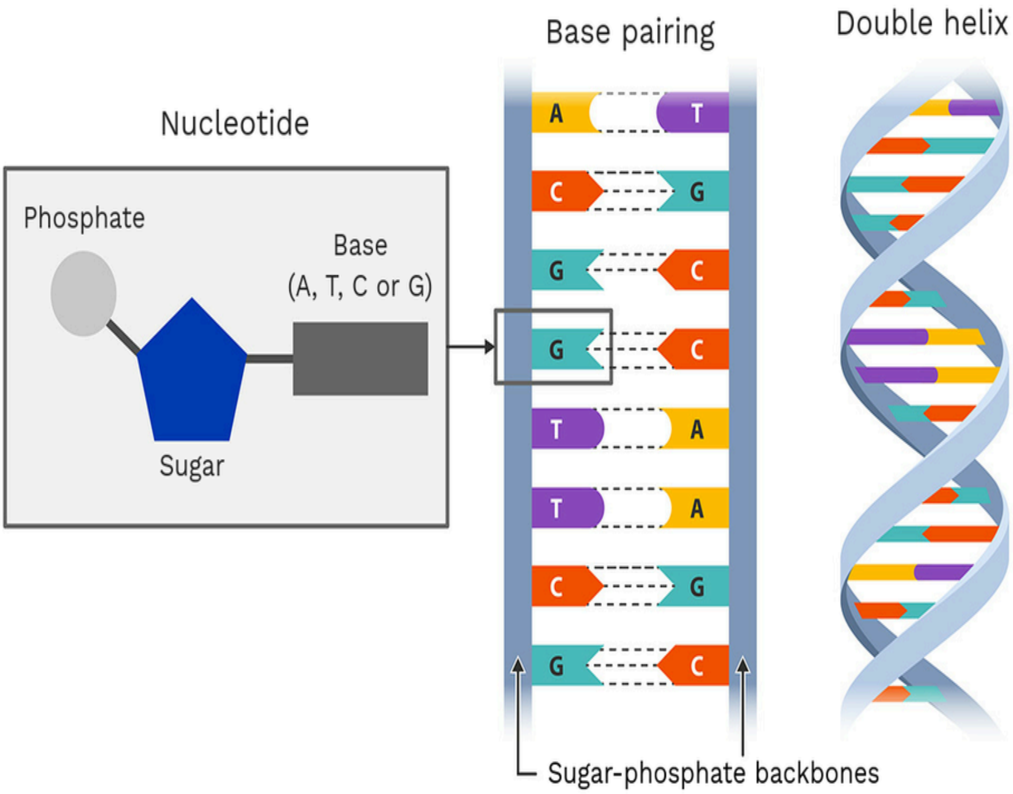


Figure 6: DNA Structure [36]

Lack of annotated data: Large amounts of annotated data are often necessary for NLP models to function well. Acquiring well-annotated information in the DNA domain is challenging since experimental confirmation of genetic activity is expensive and complicated. [37]

Interpretability of Models: Deep learning models, such as those used in NLP, can be difficult to understand due to their black-box nature. In a field like genomics, where understanding the biological significance of patterns is crucial, a lack of interpretability might be a significant hindrance. [38]

Integration with Existing Biological Knowledge: Integrating NLP models with existing biological knowledge and databases necessitates extensive data preprocessing and modification, this integration is necessary to ensure that the patterns detected by the models are biologically significant. [39]

Generalization across several species: Between species, DNA sequence patterns might vary significantly. It might be challenging to generate models with strong cross-species generalization without a large amount of data and effective model training techniques. [40]

2.3- Language Representation Models (LRM)

Language Representation Models (LRMs) are powerful artificial intelligence systems that interpret and synthesize human language after being trained on large amounts of text data. They proceed through pretraining, fine-tuning, embeddings, attention mechanisms, and the Transformer architecture. LRMs can be used for text categorization, sentiment analysis, machine translation, question answering, and text production. They have transformed natural language processing by reaching cutting-edge performance while also laying the groundwork for the development of more complex AI systems. However, obstacles include bias in linguistic data, resilience to adversarial assaults, and comprehending context in unclear settings. [41]

- **BERT: A Revolutionary Language Model for NLP**

We introduce BERT, or Bidirectional Encoder Representations from Transformers, a revolutionary language representation paradigm. Unlike earlier language representation models, BERT aims to pre-train deep bidirectional representations from the unlabeled text by conditioning on both the left and right context in all layers. As a result, the pre-trained BERT model may be fine-tuned with only one additional output layer to provide cutting-edge models for a wide range of tasks, including question answering and language inference, without needing large task-specific architecture changes. [24]

BERT Variants

BERT has been developed into various variants, including RoberTa, developed by Facebook AI, which optimizes pre-training procedures; DistilBERT, a smaller, faster, and lighter version developed by Hugging Face; and ALBERT, a lighter and faster model created by Google Research and the Toyota Technological Institute at Chicago, which reduces parameter count.

BERT has a considerable influence on medical areas by producing human-like text and tackling special issues in medical and clinical text analysis, with BioBERT and ClinicalBERT serving as important examples.

BioBERT (Biomedical BERT)

BioBERT is a BERT derivative that was specifically trained on huge biomedical datasets. The goal of BioBERT is to capture the precise and specialized language used in biomedical literature, which is important for tasks that need a deep understanding of scientific words and concepts. BioBERT enhances the efficiency of numerous biomedical text-mining activities. [25]

ClinicalBERT

ClinicalBERT specialized variant of BERT, designed to handle clinical text, which is often found in electronic health records (EHRs). ClinicalBERT is pre-trained on clinical notes and other medical documentation, enabling it to understand the nuances of clinical language and improve performance. [42]

BERT Architecture

Similar to BioBERT, ClinicalBERT is based on the original BERT architecture, which includes the following key elements:

- **Transformer Layers**: BioBERT, like BERT, uses multiple transformer layers (typically 12 for BERT-base and 24 for BERT-large). Each transformer layer consists of:

Multi-Head Self-Attention: This mechanism allows the model to focus on different parts of the input sequence simultaneously, capturing various relationships and dependencies.

Feed-Forward Neural Networks: A position-wise fully connected feed-forward network is applied to each position separately and identically.

Layer Normalization and Residual Connections: These techniques help stabilize and improve training.

- **Input Representation**

Token Embeddings: Subword tokenization with WordPiece, mapping each token to a dense vector.

Segment Embeddings: To distinguish between sentences in sentence-pair tasks.

Position Embeddings: To capture the position of each token in the sequence. [43]

- **Pre-Training objective**

ClinicalBERT and BioBERT involve a two-stage pre-training process:

General Domain Pre-Training: Initial pre-training on a general corpus and BookCorpus using masked language modeling (MLM) and next sentence prediction (NSP) tasks, similar to BERT.

Masked Language Modeling (MLM) predicts random masked tokens from a sequence. The model is then trained to predict the original tokens using the context supplied by the unmasked tokens. [44]

Purpose: This enables the model to learn bidirectional representations, allowing it to comprehend the context from both the left and right sides of a token.

Next Sentence Prediction (NSP) aims to predict if sentence B follows sentence A in the original text.

The model is trained to determine if sentence B is an accurate continuation of sentence A.

Purpose: This exercise assists the model in understanding the link between phrases, which is necessary for tasks like question responding and text coherence.

- **Pre-Training Corpus**

General: Both begin with the BERT model pre-trained on a big general corpus, such as English Wikipedia and BookCorpus.

BioBERT

Biomedical Domain Pre-Training: Biomedical Domain Pre-Training: Additional training on large-scale biomedical texts, such as PubMed abstracts and PubMed Central full-text articles.

Objective: Customize the language model for the biomedical domain, including domain-specific terms, jargon, and situations.

Sources: PubMed abstracts and PubMed Central articles. [43]

ClinicalBERT

Clinical Domain Pre-Training: Pre-trained on clinical texts, which are often obtained from electronic health records (EHRs).

Adapt the language model to the clinical domain, with an emphasis on patient notes, clinical terminology, and clinical practice-relevant situations.

Source: The MIMIC-III dataset (Medical Information Mart for Intensive Care) contains de-identified clinical records.

- **Fine-Tuning**

After pre-training, ClinicalBERT may be fine-tuned for many clinical NLP activities, whereas BioBERT can be fine-tuned for several downstream biological NLP tasks, including:

Named Entity Recognition (NER): Both identify clinical entities such as drugs, illnesses, and treatments.

Relation Extraction: clinicalBERT identifies relationships between clinical entities and BioBERT determines the links between biological entities, extraction tasks like as detecting protein-protein interactions (PPI), and drug-drug interactions.

Text Classification: ClinicalBERT organizes clinical notes by condition type, therapy strategy, and bioBERT Answering biological questions depending on context.

Clinical Outcome Prediction: ClinicalBERT predicts clinical outcomes using patient records.

Question Answering: Question Answering: BioBERT solving biological questions in context, and ClinicalBERT has been used for clinical question answering jobs, where it excels at collecting key information from clinical narratives to provide appropriate medical answers. [42]

2.4- Related Works and Discussion

The field of genomics is rapidly expanding, and Natural Language Processing (NLP) techniques are proving to be useful tools for detecting patterns in DNA sequences. This examination will go into the extensive body of research that has utilized NLP models to identify DNA patterns. We'll look at the many approaches researchers have used, the results they've achieved, and what this means for the future of genomics.

- **Analysis of previous studies that have utilized NLP models for DNA pattern recognition.**

We thoroughly searched several academic databases, including PubMed, IEEE Xplore, Google Scholar, and arXiv. We used terms such as "NLP in DNA pattern recognition" and

"Natural Language Processing for Genomics." We sought to ensure that we incorporated the most recent and relevant studies.

We classified the research based on the NLP models utilized, the datasets evaluated, and the specific applications. Here is what we found:

Recurrent Neural Networks (RNN):

- Peren Jerfi CANATALAY and al. (2022) with the LSTM-RNN and GRU network to predict splice sites in eukaryotic DNA. [45] An extensive dataset with non-site, donor, and acceptor areas was used to provide a strong representation of the model. Our model architecture with an embedding layer, dropout layer, bidirectional LSTM layer, and a dense output layer with softmax activation performs very well. Upon training with 80% and testing on 20% of the dataset, the model was able to give us an accuracy of 96.1% which is quite impressive. including non-site regions not identified, gluons, confusing overlapping predictions, and calls to study them in-depth.

Recurrent Neural Networks (RNN)/ Convolutional Neural Network (CNN):

- Ying He et al. (2021) [46] use a variety of deep learning frameworks, such as CNN-based, RNN-based, and a combination of CNN-RNN models for the analysis of biological sequences for DNA/RNA motif mining is the foundation of gene Using Deep Learning. The ECBLSTM model is an efficient performing model with median AUC values on the ChIP-seq and CLIP-seq datasets. Ying, admits several flaws in the article — the relatively patchy nature of the data, concerns about how well the model could be interpreted, and persistence of model complexity and hyperparameter selection and optimal design issues. These challenges need to be resolved for deep learning to push ahead in the mining of motifs and biological sequence analysis.

- Junghwan Baek et al.(2018) [47] describe lncRNAnet, a deep learning-based method for classifying long non-coding RNAs (lncRNAs) from transcripts that code for proteins. The technique makes use of one-dimensional convolutional neural networks (CNN) to locate open reading frames in transcripts and recurrent neural networks (RNN) for sequence analysis. The research suggests that lncRNAnet outperforms techniques when it comes to analyzing data, from species and datasets showing higher levels of specificity, accuracy, F1 score, and AUC. However, the methods's opaque nature the need for validation across transcriptomes and experimental conditions well, and the necessity, for improved data curation and preprocessing steps are noted as some of its limitations.

Convolutional Neural Network (CNN):

- Hemalatha Gunasekaran and al [48] prepare for The DNA series category is where it is primarily found which is crucial for organic domain names like medicine manufacture and infection discovery. In this research study, attributes are drawn out from raw information utilizing CNN LSTM and also CNN Bidirectional LSTM, 2 finding out versions developed within Convolutional Neural Networks (CNNs). There are 2 inscribing techniques utilized: mer inscribing as well as tag inscribing. The CNN Bidirectional LSTM with mer inscribing which attains a precision of 93.13% routes carefully behind the CNN version with tag inscribing which gets to 93.16%. Metrics consisting of recall, level of sensitivity, uniqueness as well as precision are included in the efficiency assessment. Nonetheless, there are concerns with the research's handling ability, assessment criteria, together with data dimension. Comprehensive data source usage might improve the versions' strength and usefulness. These restrictions must be taken into account in research.

Enhancer-LSTMAtt:

- Guohua Huang et al.(2022) [49] boosters are brief DNA sectors that play an essential duty in organic procedures they provided a bi-directional long-short term memory(Bi-LSTM) together with attention-based deep understanding technique(Enhancer-LSTMAtt) for booster acknowledgment they discovered that Integrating interest system improves the version's capability to concentrate on crucial series sections, for this reason boosting forecast precision plus analysis as well as Bi-LSTM networks catch temporal reliances within DNA series, enabling the design to find out made complex patterns needed for booster discovery however on contrary Deep discovering designs such as Enhancer-LSTMAtt can be computationally requiring, calling for big sources for training along with reasoning generate omputational Complexity, training of deep understanding designs regularly needs substantial classified, Deep discovering designs for DNA series evaluation educated making use of substantial classified datasets **.**

Transformers:

- Pavan Holur et al. (2024) [50] based their research for Embed-Search-Align: DNA Sequence Alignment by using Transformer Models" They found that the transformer models for DNA sequence alignment are a novel approach with the potential to change bioinformatics, it have a Potential for Improvement, showed remarkable performance in a variety of areas, and their application to DNA sequence alignment has the potential to significantly increase accuracy and efficiency but Transformer models are known to be computationally demanding, requiring large resources for training and inference. This might be difficult, especially for large-scale genomic datasets.

• Shujun He [51] concentrates in his research study on modeling as well as evaluating DNA series for organic jobs like recognizing marketers, boosters as well as viral series. The Nucleic Transformer strategy which incorporates self-attention as well as tightening, The design shows appealing outcomes throughout numerous DNA information collections such as anticipating chromatin attributes on DeepSea, accomplishing high precision, level of sensitivity, uniqueness, as well as Matthews relationship coefficient on the E. coli promoter/nonpromoter information collection as well as distinguishing in between booster coupled with nonenhancer series on the enhancer/nonenhancer information collection. Additional recognition and optimization might be called for to integrate the technique throughout a more comprehensive variety of DNA series and also functions.

2.5- Comparative Table:

Approach	**Studie**	**Result**	**Limitations**
Recurrent Neural Networks (RNN)	Pere Jerfi CANATALAY et al. (2022)	Accuracy:96.1%	Issues with non-site regions, overlapping predictions, and model interpretation
Recurrent Neural Networks (RNN) / Convolutional Neural Network (CNN)	Ying He et al. (2021)	Data-Efficient models with median AUC values	**model complexity, hyperparameter selection**
Convolutional Neural Network (CNN)	-Junghwan Baek et al. (2018) -Hemalatha Gunasekaran et al. (Year not provided)	-Outperforms other techniques in specificity, accuracy, F1 score, and AUC	- Opacity, need for validation across transcriptomes and experimental conditions

		-Precision: CNN Bidirectional LSTM (93.16%), CNN (93.13%)	-Handling ability, evaluation criteria, data size
Enhancer-LSTMAtt	Guohua Huang et al. (2022)	Improved prediction accuracy and analysis	Computational demands need for large datasets, complexity
Transformers	- Pavan Holur et al. (2024) - Shujun He (Year not provided)	- Potential for significant increase in accuracy and efficiency. - Promising results across various DNA datasets	- Computational demands, especially for large-scale datasets. - Further optimization needed for broader application

2.6- The synthesis

We examined how natural language processing (NLP) can help study genes, focusing on how it finds patterns in DNA. He emphasized that special models such as DNABERT are needed for tasks such as recognizing names and reviewing medical records.

One of the obstacles involves handling data, insufficient documentation ensuring transparency, and adapting research findings to formats. The discussion also delves into models, like BioBERT for deciphering words. Various NLP methods such as core connections, intertwined networks, and Transformer models are utilized to detect DNA patterns.

2.7-Discussion on the gaps in existing research and the need for a specialized model like DNABERT.

Our research revealed that models such as CNNs and RNNs They've been around for a long time and are quite good at what they do, but they have limitations. For example, RNNs struggle to maintain track of long-distance links in the DNA sequence; it's as if they've forgotten what happened at the beginning. Furthermore, while CNNs excel at detecting patterns in sequences, they can only watch one fixed segment at a time.

Deep learning techniques for interpreting DNA sequences have presented a few technological hurdles, and Transformers need a significant amount of resources during training and analysis. Furthermore, you require a large amount of well-labeled data, which is not always simple to obtain. They also have difficulty detecting long-distance correlations in data, which is critical in genomics.

So why do we need a model like DNABERT? Well, think of it like this:

Learning DNA Stuff: DNABERT is trained specifically on DNA data. It starts with the basics and then moves on to more complex stuff. By training on lots of DNA data, DNABERT gets good at understanding the patterns and structure hidden in DNA sequences, stuff that other general-purpose models might miss.

Connecting the Dots: DNABERT has this cool trick called self-attention, which is like having a superpower to connect the dots over long distances in the DNA sequence. This comes in handy when we're trying to predict things like how different parts of the DNA interact with each other to control gene expression.

Getting Better Results: With DNABERT, we're seeing better results in all sorts of DNA tasks, like in prediction.

Handling Big Data: DNA is complicated and elaborate when comes to working with huge data, but it's also good at handling big chunks of DNA data without breaking a sweat.

Making Sense of It All: DNABERT doesn't just give us predictions – it also tells us why it thinks certain parts of the DNA are important for a given task. It's like having a guide to assist in making sense of the structure of DNA.

Higher accuracy in the performance acquired results being in concordance with clearer and more understandable measures and results that express features in a physiological sense. Looking into the future, with more advancement in genetics, works such as DNABERT will be our compass, to lead us to more innovations and advancements.

DNA BERT is probably a rather efficient tool to look for patterns within such non-coding DNA parts. It makes complex rhythms easier to discern by grasping the regulating relationships within a succession of notes or beats. The transformers serve as helpful components in non-coding DNA since they enable one to derivate features from the input sequences. The non-coding DNA sequences might be more easily understood through pre-training with large datasets to tackle the difficulties in pre-training DNA

BERT. It may prove to be useful in a range of ways, including the identification of patterns within DNA found in regions not considered to contain significant genetic information. It remains a relatively open-ended approach to non-coding DNA analysis because it can simultaneously employ multiple analysis modes.

I believe that constructing particular models such as DNABERT is akin to developing a tool for interpreting DNA sequences. These models use a personalized strategy to solve research gaps. DNABERT takes use of the greatest features of Transformer models, such as their ability to connect distant segments of the sequence and manage vast amounts of data, and adds a layer of expertise by training on DNA data.

What was the result? Improved performance, more clear and intelligible results, and physiologically meaningful findings. As we explore further into genetics, models like DNABERT will be reliable guides, leading us to discoveries and applications.

3- Conclusion:

in conclusion, NLP through genomics research has provided how DNA patterns can be deciphered, based on the previous section. As we have seen through the use of different methodologies demonstrates that NLP will play a well-defined space in disentangling the intricate meanings of DNA sequences and promote progress in the field of biomedical research and healthcare and the rise of specialized models such as DNABERT, which we will examine his architecture, design choices, and optimization strategies and its role in analyzing the language of non-coding DNA in genomic research.

Chapter 3:
The proposed model

1 Introduction

In this section we take a look, at DNABERT, a version of the BERT model crafted for analyzing genomic data. The goal of this section is to offer a grasp of DNABERTs structure, which includes its utilization of the Transformer encoder tokenization into k mers embedding layers and positional encoding. We delve into the methods used for pre training outlining how datasets are chosen the concept of masked language modeling and strategies for optimization. Following that we touch on tuning approaches customized for tasks such as sequence classification and motif discovery. Furthermore we examine real world applications in research and strategies to improve performance in recognizing coding DNA sequences while also highlighting the significance of attention visualization for understanding the models output. This summary provides readers with insights into DNABERTs capacity to efficiently process and analyze DNA sequences driving progress, in biology.

2 Model Architecture: DNABERT (Overview of BERT)

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both the left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

pre-training and fine-tuning technique is shown. The training process for DNABERT is the same as that for BERT. [52]

Transformer and DNABert

BERT is a transformer-based contextualized language representation model that has demonstrated superhuman performance in a variety of natural language processing (NLP) tasks. It presents a pre-training and fine-tuning approach that first creates general-purpose understandings from huge amounts of unlabeled data before solving numerous applications using task-specific data with minimum architectural alteration. DNABERT uses the same training procedure as BERT.

Tranformer Architect

Transformer Architecture: DNABERT is built upon the Transformer architecture, which consists of encoder and decoder layers. Since DNABERT usually works on tasks like DNA sequence classification, motif discovery, or gene prediction, it mostly uses just the encoder part.

Because these tasks often don't require generating sequences but rather making sense of and sorting the ones already there, DNABERT mainly uses only the encoder bit of the Transformer architecture.

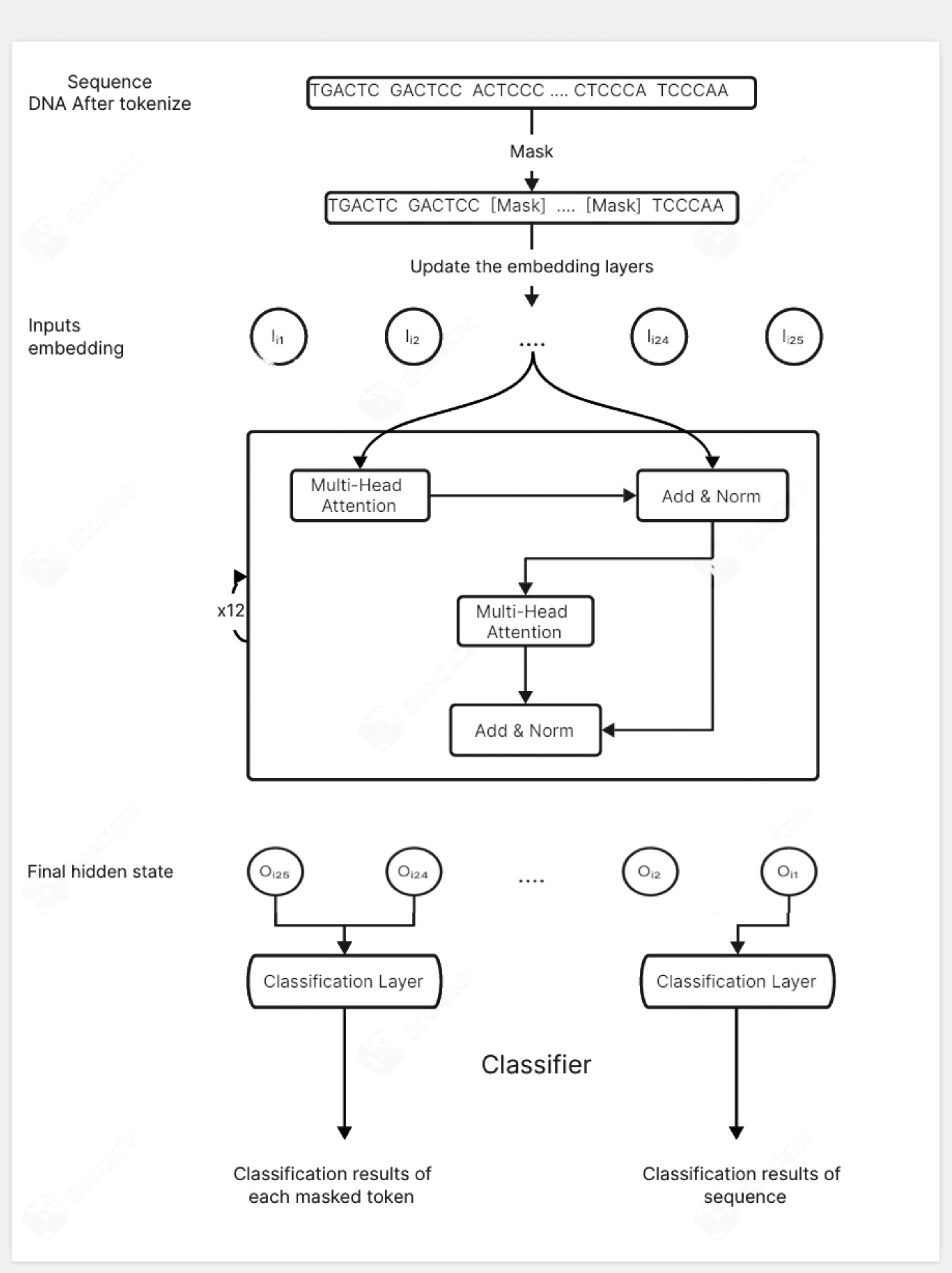


Figure 7: DNABert Architecture.

1- Tokenization:

Tokenized K-mer sequences: Rather than treating each base as a distinct token, we tokenized a DNA sequence using the k-mer representation, a commonly used method for studying DNA sequences. The k-mer representation adds more contextual information to each deoxynucleotide base by concatenating it with the ones that follow. Their concatenation is known as a k-mer. A DNA sequence 'ATGGCT' may be tokenized into four 3-mer sequences: {ATG, TGG, GGC, GCT} or two 5-mer sequences: {ATGGC, TGGCT}. various k results in various tokenizations of a DNA sequence.

Special Tokens: The input sequence includes special tokens.

CLS token: This is a special token added at the start of the sequence. It represents the total meaning of the sequence and is utilized for tasks such as sequence-level categorization.

The SEP: token is used to divide various sequences or portions of the input

MASK tokens: These tokens are used during the pre-training phase to mask specific k-mers. By training the model to predict these masked k-mers, it learns about the connections between DNA sequences.

Embedding Layer: Tokenized k-mers and special tokens are sent through an embedding layer.

Tokenized k-mers and special tokens are communicated via an embedding layer. This layer converts the tokens into dense numerical vectors (embeddings), which the Transformer can then process.

- **Transformer Encoder Blocks:**

The Transformer model architecture follows an encoder-decoder structure.

Self-Attentive System:

Global Contextual Embedding: The model can compute a representation for every k-mer that takes into account the context of the entire sequence thanks to the self-attention mechanism. This is especially crucial for DNA sequences since distant nucleotides can interact in a useful way. Using self-attention can result in more comprehensible models.

3.Pre-training and Fine-tuning

- **Pre-training Process**

DNABERT accepts a sequence with a maximum length of 512 as input. As shown in Figure 1b, we tokenized a DNA sequence into a k-mer sequence, adding a special token [CLS] at the beginning and a special token [SEP] at the end. In the pre-training step, we masked contiguous k-length spans of certain k-mers, assuming that a token could be trivially inferred from the immediately surrounding k-mers, taking 15% of the total sequence, while in the fine-tuning step, we skipped the masking step and fed the tokenized sequence directly to the Embedding layer. We produced training data from the human genome using two approaches, direct non-overlap splitting and random sampling, with sequence lengths ranging from 5 to 510. We pre-trained DNABERT for 120k steps with a batch size of 2000. In the first 100,000 steps, we masked 15% of the k-mers in each sequence. In the final 20,000 steps, we increased the masking rate to 20%. The learning rate was linearly raised from 0 to 4e-4 in the first 10k steps before decreasing linearly to 0 after 200k steps . We ended the training operation after 120k steps since the loss curve indicated plateauing. We employed the same model architecture as the BERT foundation, which has 12 Transformer layers with 768 hidden units and 12 attention heads in each layer.

Pre-training Process

The pre-training of DNABERT involves training the model on a large corpus of unlabeled DNA sequences. The objective is to enable the model to learn general features of DNA sequences. Common pre-training tasks include masked language modeling (MLM), where random k-mers are masked and the model is trained to predict them based on the surrounding context.

- **Training Strategy**

The training strategy for DNABERT includes a two-phase approach:

Pre-training: Involves learning from vast amounts of unlabeled DNA data to capture general sequence features.

Fine-tuning: Involves adjusting the pre-trained model on specific downstream tasks (e.g., promoter prediction, motif discovery) using labeled data to enhance task-specific performance.

3. System Modules

In the realm of connections and relationships there are components to consider:

Tokenization

Design Choice: K-mer Tokenization

Rationale: DNA sequences consist of four nucleotides (A, T, C, G) and can be extremely long. Tokenizing into k-mers (subsequences of length k) captures local sequence patterns and makes the input manageable for the model.

Contribution: K-mer tokenization helps the model focus on small, meaningful subsequences that contain biological information. This method allows DNABERT to process the sequence in a structured manner and capture local dependencies, which are crucial for understanding DNA sequences.

Special Tokens ([CLS], [SEP], [MASK])

Rationale: Incorporating special tokens allows the model to handle different tasks such as classification and masked language modeling.

Contribution: The [CLS] token aggregates information from the entire sequence for classification tasks, the [SEP] token separates different segments if needed, and the [MASK] token enables the pre-training task of masked language modeling, enhancing the model's contextual understanding. Sure! Here's the edited text.

Embedding Layer

Design Choice: Token Embeddings

Why we chose this: To convert individual k-mer tokens into continuous, dense vectors using an embedding matrix.

contribution: This transformation allows the model to process the input in a way that works well for neural networks. It helps capture the meaning and similarities between different k-mers.

Design Choice: Positional Encodings

Why we chose this: The Transformer architecture lacks inherent information about the order of sequences. Positional encodings fill in this missing sequence order information.

contribution: By adding positional encodings to the token embeddings, DNABERT can maintain the order of k-mers. This is crucial for understanding the context of biological sequences and preserving the accuracy of DNA information.

Transformer Encoder Blocks

Design Choice: Multi-Head Self-Attention Mechanism

Why we chose this: Self-attention allows the model to determine the token's knowledge of every other token in the series

. Multi-head attention lets it focus on different parts of the sequence simultaneously.

contribution: This mechanism is essential for capturing long-range dependencies and interactions within DNA sequences. Being able to focus on multiple aspects of the sequence at once gives us a comprehensive understanding of the relationships and dependencies in the data.

Design Choice: Feed-Forward Neural Networks.

Why we chose this: We apply non-linear transformations to the outputs of self-attention.

Contribution: This combination helps enhance the self-attention outputs by introducing non-linearity. It contributes to improving the overall performance of the model.

Contribution: These networks enhance the model’s ability to learn complex patterns from the data, adding depth and complexity to the representations learned by the self-attention layers.

Design Choice: Layer Normalization and Residual Connections

Rationale: Stabilize training and improve convergence with layer normalization, and facilitate gradient flow with residual connections.

Contribution: These design choices ensure that the model can train effectively and efficiently, even with deep architectures, leading to better performance and more robust learning.

Pre-Training Tasks

Design Choice: Masked Language Modeling (MLM)

Rationale: Mask certain k-mers during training and learn to predict them based on context.

Contribution: MLM forces the model to understand the context and dependencies within sequences, which is crucial for capturing the underlying structure of DNA. This pre-training task significantly improves the model’s ability to generalize and understand unseen sequences.

Design Choice: Next Sentence Prediction (NSP)

Rationale: Although less commonly used in DNABERT, NSP can enhance contextual understanding by predicting whether two sequences are contiguous.

Contribution: This task, more relevant to NLP, ensures the model can understand larger contexts beyond individual sequences when applied.

Fine-Tuning Layers

Design Choice: Task-Specific Heads

Rationale: Add specific layers for different downstream tasks, such as sequence classification or token-level prediction.

Contribution: Customizing the model for specific tasks allows DNABERT to be versatile and applicable to a range of genomic analysis problems. This adaptability ensures that the pre-trained model can be fine-tuned effectively for various practical applications.

Attention Visualization

Design Choice: Visualization of Attention Patterns

Rationale: Understanding which parts of the sequence the model focuses on can provide insights into the model’s decision-making process.

Contribution: Visualizing attention patterns helps validate the model by showing that it focuses on biologically relevant regions, such as regulatory sites or motifs. This transparency is crucial for interpreting the model’s predictions and gaining trust in its outputs.

4- Training and Optimization Strategies

DNABERT is a specialized variant of the BERT (Bidirectional Encoder Representations from Transformers) model designed for understanding the language of DNA sequences. The goal is to capture the patterns and structures inherent in non-coding DNA sequences.

Here’s an overview of the pre-training process for DNABERT, including dataset selection and preprocessing steps:

Dataset Selection

Source of Sequences: The datasets used for pre-training DNABERT typically come from large genomic databases such as the Human Genome Project, Ensembl, or other public repositories.

The primary focus is on non-coding regions of the DNA, which include introns, regulatory sequences, and other regions not translated into proteins.

Sequence Length: DNA sequences are segmented into fixed-length k-mers (subsequences of length k). Common choices for k range from 6 to 12.

This segmentation allows the model to handle variable-length DNA sequences consistently and manage the vast size of genomic data.

Pre-Processing Steps

Tokenization: DNA sequences are tokenized into k-mers. For instance, the sequence "AGCTGAC" could be tokenized into overlapping k-mers like "AGCT", "GCTG", "CTGA", and "TGAC" if k=4.

This k-mer tokenization process transforms the DNA sequence into a series of fixed-length tokens, similar to words in natural language processing (NLP).

Vocabulary Construction: A vocabulary of all possible k-mers is constructed. For example, with k=6, the vocabulary consists of all possible 6-mers (4^6 = 4096 unique tokens).

The vocabulary also includes special tokens like [CLS] (classification), [SEP] (separator), and [PAD] (padding), akin to those in standard BERT models.

Masking: Similar to the BERT model's masked language model (MLM) objective, random k-mers within the DNA sequences are masked (replaced with a [MASK] token).

The model is trained to predict these masked k-mers based on the surrounding context, enabling it to learn the patterns and dependencies in DNA sequences.

Input Representation: Each k-mer token is mapped to a dense vector representation (embedding).

Positional encodings are added to maintain the sequential information of the k-mers within the DNA sequence.

Training Data Preparation: Sequences are split into training and validation sets.

The training set is used for model training, while the validation set is used to monitor performance and prevent overfitting.

Model Architecture

DNABERT retains the core BERT architecture with modifications tailored to DNA sequences. It uses multiple layers of Transformer encoders to capture complex dependencies and patterns in the k-mer sequences.

Training Procedure

Training Objective: The primary objective is the masked language modeling task, where the model learns to predict masked k-mers based on their context.

Additional objectives, such as next-sentence prediction, can be adapted to DNA sequence tasks, like predicting adjacent k-mers or related sequences.

Optimization: Standard optimization techniques for training deep learning models are applied, such as Adam optimizer with appropriate learning rate schedules.

Early stopping and regularization methods may be used to enhance model generalization.

Computational Resources: Training DNABERT requires significant computational power, often leveraging GPUs or TPUs for efficient processing of large genomic datasets.

AdamW Optimizer:

The AdamW optimizer is used for training the BERT-based DNA pattern recognition model. AdamW is a variant of the Adam optimizer that includes a term for weight decay regularization. This helps to reduce overfitting by penalizing large weights.

Improved DNABERT performance for recognition of non-coding DNA structure

1. Pre-train on non-coding domain-specific data

Large-scale non-coding genomic data: Pre-train DNABERT on multiple non-coding DNAs on an ad hoc basis, such as supporting, enhancing, and managing the domain. This allows the model to learn patterns specific to these regions that differ from the encoding process.

Patterns capture general concepts and patterns in non-coding DNA sequences. The most common options are 3-mer or 6-mer, but other lengths may be suitable depending on the specific task. Structural modification

Mechanism analysis: Improving maintenance of important structures in non-coding DNA structures to make them more sensitive. You can refine the number of heads and layers to see the quality of the model. Reliable and continuous data

Periodic improvement: Use techniques such as mutation, insertion, deletion, and recovery to obtain a variety of training materials and improve the capacity of the model. This type of regression and weighting is used to avoid overfitting, which is common when dealing with high-resolution data such as genomic data. Fine-tuning strategy

Task-specific fine-tuning: Pre-trained fine-tuning models for specific non-coding DNA structure recognition tasks (such as development or informing the sponsor) use callout related to these areas. Models pre-trained on coding regions or other genomic functions are fine-tuned on non-coding DNA data. Hyperparameter Optimization

Learning Rate Planning: Use learning rate planning such as learning rate warm-up and decay to stabilize training and improve convergence. Teach and develop teamwork. Model Ensemble and Multi-Task Learning

Ensemble approach: Use an ensemble approach by combining predictions from multiple DNABERT models learned with different thresholds or hyperparameters to increase robustness and accuracy. Guide DNABERT through various tasks such as motif discovery and support prediction to exploit common features and improve general capabilities. Iterative evaluation and development

Cross-validation: Uses cross-validation techniques to test the performance of the model on different data sets to ensure the robustness of the model. Deviation model for constant performance.

- **Evaluation Metrics**

Accuracy is the ratio of accurately anticipated occurrences to total instances.

Precision is the ratio of true positive predictions to the total number of true positive and false positive forecasts.

Recall (Sensitivity) is the proportion of true positive predictions to the total of true positive and false negative predictions.

F1 Score: The harmonic mean of accuracy and recall, which achieves a balance between the two.

Area The AUC-ROC curve measures the model's ability to differentiate between classes.

A confusion matrix is a table that describes the classification model's performance on a set of test data whose real values are known.

5-Conclusion

DNABERT greatly promotes genomic research by surpassing existing models in a variety of DNA sequence analysis tasks, including promoter prediction, motif identification, gene categorization, and mutation detection, while also displaying high accuracy and excellent k-mer tokenization. Its capacity to analyze complicated sequences and focus on critical genomic areas gives valuable insights into DNA data, improving prediction models and allowing for large-scale, automated analysis. The ramifications for genomic research are significant, including better illness diagnostics, tailored medication, and genetic research. Future directions include integrating DNABERT with multi-omics data, refining its architecture, improving interpretability, adapting for real-time applications, and fostering open-source, collaborative research, thus pushing the boundaries of genomic analysis and paving the way for new discoveries and applications.

Chapter 4

1- Introduction:

In this section, we present a comprehensive overview, of how DNABERT was tested outlining the setup, methods, findings, and discussions. We start by describing the testing conditions, such as the hardware and software setups introducing the datasets used for training and assessment including any techniques, for enhancing data, including quantitative performance metrics and qualitative analysis, followed by a comparative evaluation of DNABERT with baseline models and state-of-the-art approaches.

2- Description of the experimental environments:

The development of the model is carried out via laptop with the following characteristics:

Marque	**DELL Inspiron1525**
Processor	Intel Core™ 2 Duo avec CPU IntelT1600 (1.66GHz)
RAM	4GO
Hard disk	500 GO
Operating system	Microsoft Windows 10 professionnel

3- development tools:

3.1- Languages used:

a. **Python:** is an open source, versatile and user-friendly interpreted programming language propelled to the forefront in infrastructure management, data analysis or software development. It is widely used in the field of software development, data analysis, machine learning and artificial intelligence. Python stands out for its clear and concise syntax, which makes it easier to read and write code. It also offers an extensive standard library and many specialized third-party libraries that facilitate application development and solving various problems. Thanks to its popularity and active community, Python has become one of the most commonly used programming languages and is loved for its flexibility and ease of use.[1]

Figure 8. *2 Python [2]*

3.2- Product of Google Search:

Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientists to solve data science challenges. The aim of this online platform (founded in 2010 by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017) is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides.[3]

4-DataSet used in our work:

4.1- Description of the DataSet:

DNABERT is a model trained on the human reference genome (Hg38.p13) to generate dense representations of genome sequences, particularly for tasks like splice-site prediction. The primary dataset used for training DNABERT is the human genome assembly GRCh38.p13, which contains 3.2 billion nucleotides. This dataset is publicly available and maintained by NCBI.

The model was pre-trained on human genome sequences and then fine-tuned using splice-site annotations from Ensembl release. The evaluation dataset includes approximately 80,000 unique gene isoforms with various exon arrangements, and 30,000 splice sites are sampled for evaluation purposes. These annotations were collected using both automated methods and human review.

The DNABERT model and its associated resources were made available around 2021, with ongoing updates and improvements documented in publications and repositories such as BioRxiv and GitHub.

For further details, you can visit the DNABERT GitHub repository: [DNABERT on GitHub] (https://github.com/jerryji1993/DNABERT).

4.1.1. Exprimental Protocols:

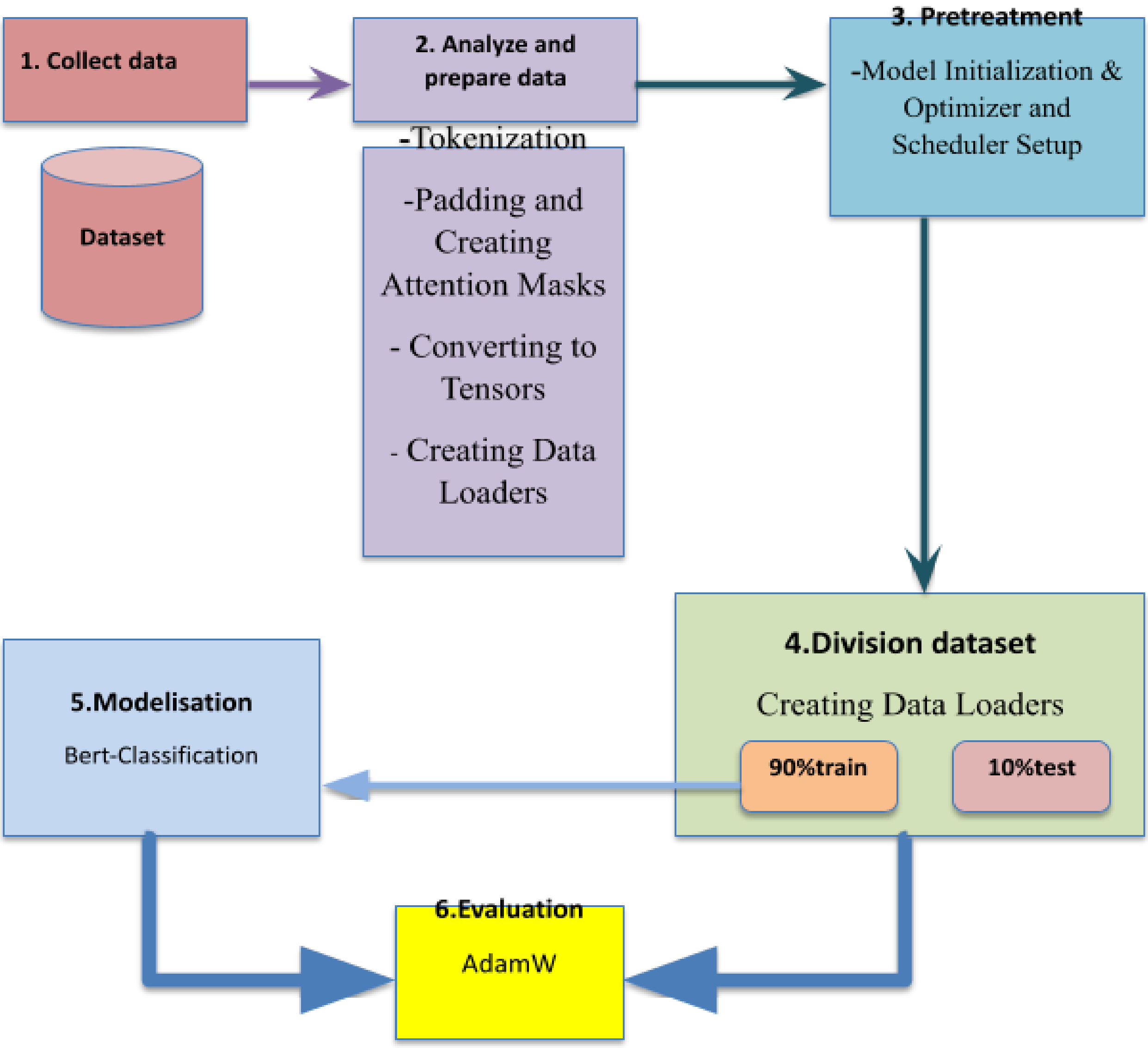
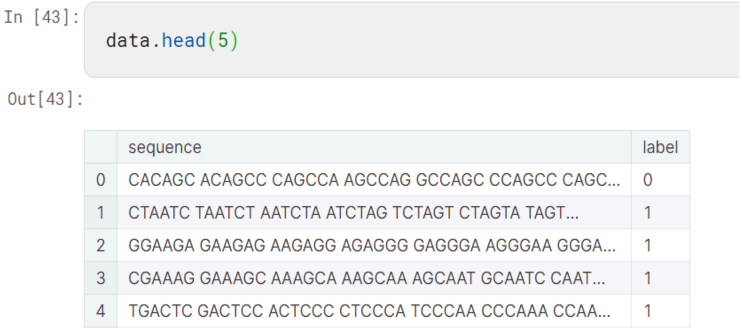


Figure 9. Exprimental Protocols

4.2 Collect Data



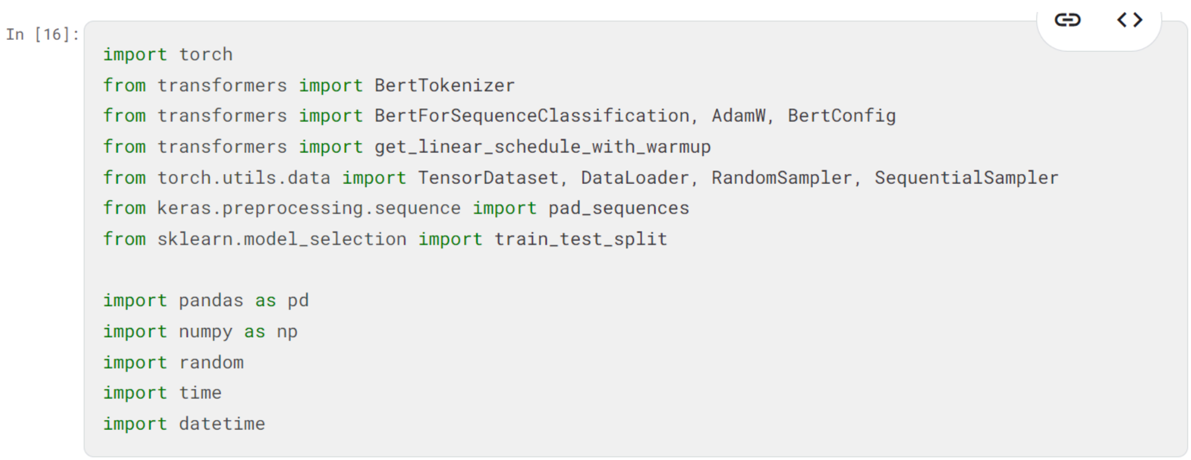
```
In [43]: data.head(5)

Out[43]:
```

	sequence	label
0	CACAGC ACAGCC CAGCCA AGCCAG GCCAGC CCAGCC CAGC...	0
1	CTAATC TAATCT AATCTA ATCTAG TCTAGT CTAGTA TAGT...	1
2	GGAAGA GAAGAG AAGAGG AGAGGG GAGGGA AGGGAA GGGA...	1
3	CGAAAG GAAAGC AAAGCA AAGCAA AGCAAT GCAATC CAAT...	1
4	TGACTC GACTCC ACTCCC CTCCCA TCCCAA CCCAAA CCAA...	1

Figure 10: Dataset (First 5 index)

4.3-Analyze and prepare data

Importing the packages :



```
In [16]:
import torch
from transformers import BertTokenizer
from transformers import BertForSequenceClassification, AdamW, BertConfig
from transformers import get_linear_schedule_with_warmup
from torch.utils.data import TensorDataset, DataLoader, RandomSampler, SequentialSampler
from keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split

import pandas as pd
import numpy as np
import random
import time
import datetime
```

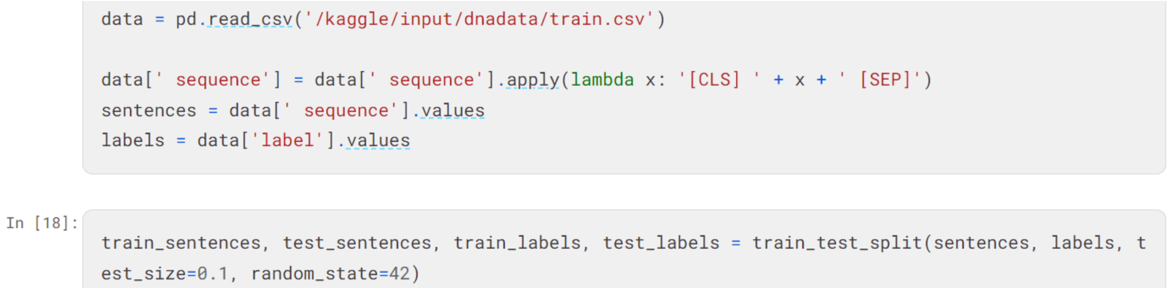
Figure 11: Importing packages

Data Preparation:

Load DNA sequence data from a CSV file.

Add special tokens [CLS] (start of sequence) and [SEP] (end of sequence) to each DNA sequence.

Split the data into training and testing sets using an 90% for trainand 10% for test.



```
data = pd.read_csv('/kaggle/input/dnadata/train.csv')

data[' sequence'] = data[' sequence'].apply(lambda x: '[CLS] ' + x + ' [SEP]')
sentences = data[' sequence'].values
labels = data['label'].values
```

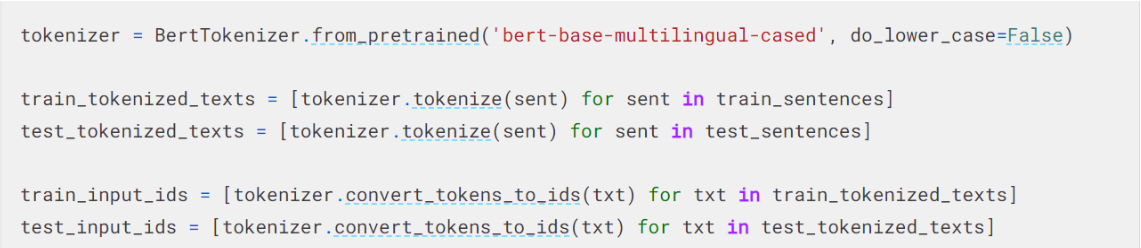
```
In [18]:
train_sentences, test_sentences, train_labels, test_labels = train_test_split(sentences, labels, t
est_size=0.1, random_state=42)
```

Figure 12 :data preparation

Tokenization:

Use the BERT tokenizer to tokenize the DNA sequences.

Convert the tokenized sequences into their corresponding token IDs.



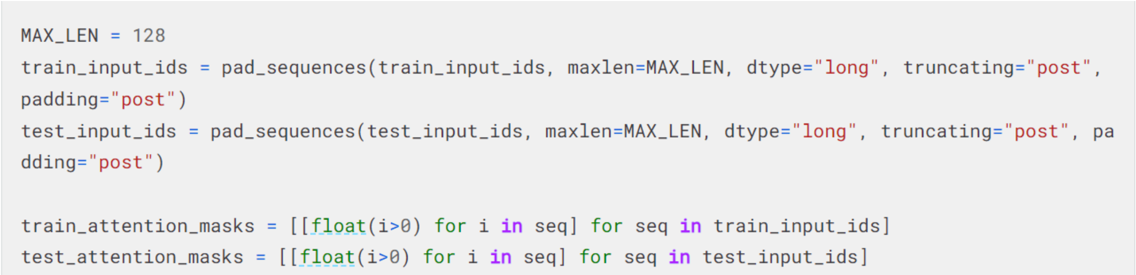
```
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased', do_lower_case=False)

train_tokenized_texts = [tokenizer.tokenize(sent) for sent in train_sentences]
test_tokenized_texts = [tokenizer.tokenize(sent) for sent in test_sentences]

train_input_ids = [tokenizer.convert_tokens_to_ids(txt) for txt in train_tokenized_texts]
test_input_ids = [tokenizer.convert_tokens_to_ids(txt) for txt in test_tokenized_texts]
```

Figure 13 : Tokenization

Padding and Creating Attention Masks:

Pad and truncate the sequences to a fixed length (128 tokens).

Create attention masks to distinguish between actual tokens and padding tokens.

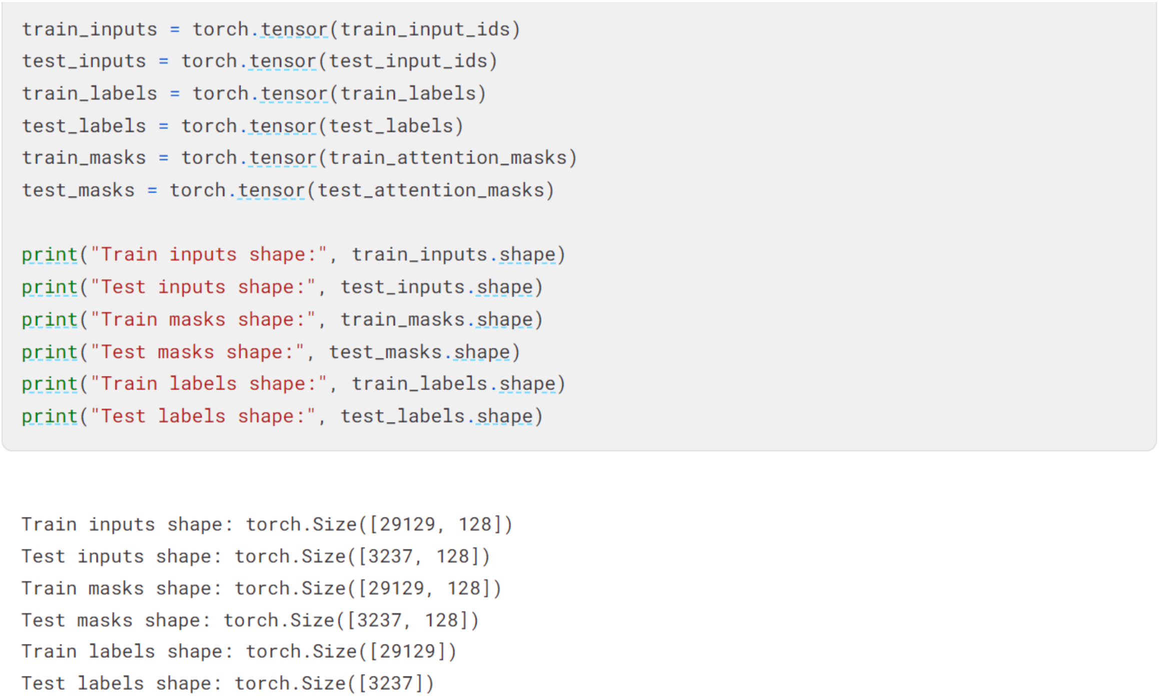


```
MAX_LEN = 128
train_input_ids = pad_sequences(train_input_ids, maxlen=MAX_LEN, dtype="long", truncating="post",
padding="post")
test_input_ids = pad_sequences(test_input_ids, maxlen=MAX_LEN, dtype="long", truncating="post", pa
dding="post")

train_attention_masks = [[float(i>0) for i in seq] for seq in train_input_ids]
test_attention_masks = [[float(i>0) for i in seq] for seq in test_input_ids]
```

Figure 14: Padding and Creating Attention Masks

Converting to Tensors:

Convert the input IDs, attention masks, and labels into PyTorch tensors for compatibility with the BERT model and the training loop.



```
train_inputs = torch.tensor(train_input_ids)
test_inputs = torch.tensor(test_input_ids)
train_labels = torch.tensor(train_labels)
test_labels = torch.tensor(test_labels)
train_masks = torch.tensor(train_attention_masks)
test_masks = torch.tensor(test_attention_masks)

print("Train inputs shape:", train_inputs.shape)
print("Test inputs shape:", test_inputs.shape)
print("Train masks shape:", train_masks.shape)
print("Test masks shape:", test_masks.shape)
print("Train labels shape:", train_labels.shape)
print("Test labels shape:", test_labels.shape)
```

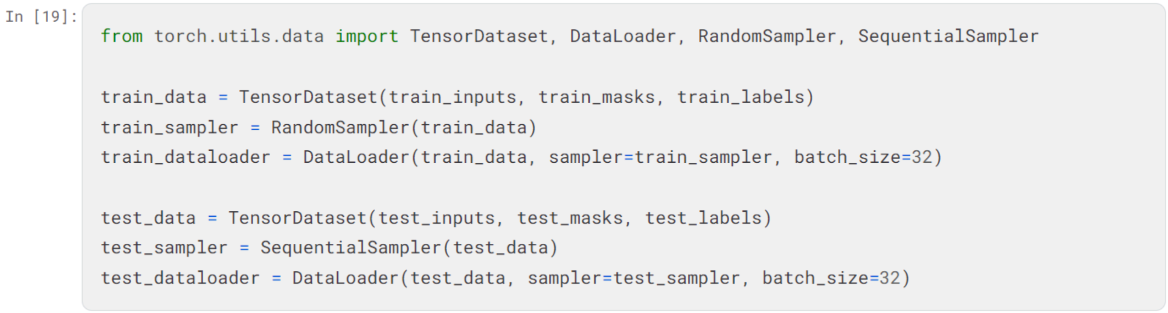
```
Train inputs shape: torch.Size([29129, 128])
Test inputs shape: torch.Size([3237, 128])
Train masks shape: torch.Size([29129, 128])
Test masks shape: torch.Size([3237, 128])
Train labels shape: torch.Size([29129])
Test labels shape: torch.Size([3237])
```

Figure 15: Converting to Tensors

Creating Data Loaders:

Create Tensor Dataset objects for the training and testing sets.

Use Data Loader with appropriate samplers (Random Sampler for training, Sequential Sampler for testing) to load data in batches.



```
In [19]:
from torch.utils.data import TensorDataset, DataLoader, RandomSampler, SequentialSampler

train_data = TensorDataset(train_inputs, train_masks, train_labels)
train_sampler = RandomSampler(train_data)
train_dataloader = DataLoader(train_data, sampler=train_sampler, batch_size=32)

test_data = TensorDataset(test_inputs, test_masks, test_labels)
test_sampler = SequentialSampler(test_data)
test_dataloader = DataLoader(test_data, sampler=test_sampler, batch_size=32)
```

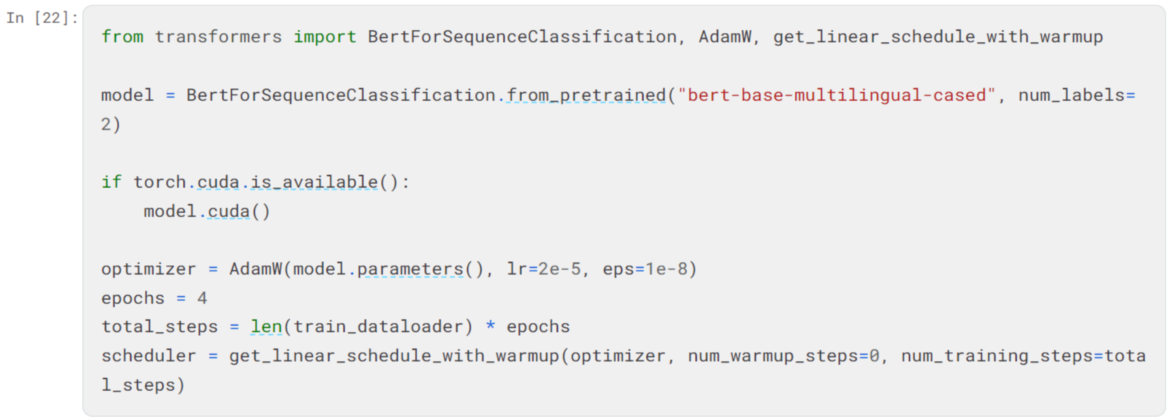
Figure 16 : Creating Data Loaders

4.4- Model Initialization & Optimizer and Scheduler Setup:

Initialize the BERT model for sequence classification with two output labels (binary classification).

Configure the optimizer (AdamW) with a learning rate and epsilon for numerical stability.

Set up a learning rate scheduler to gradually reduce the learning rate during training.



In [22]:

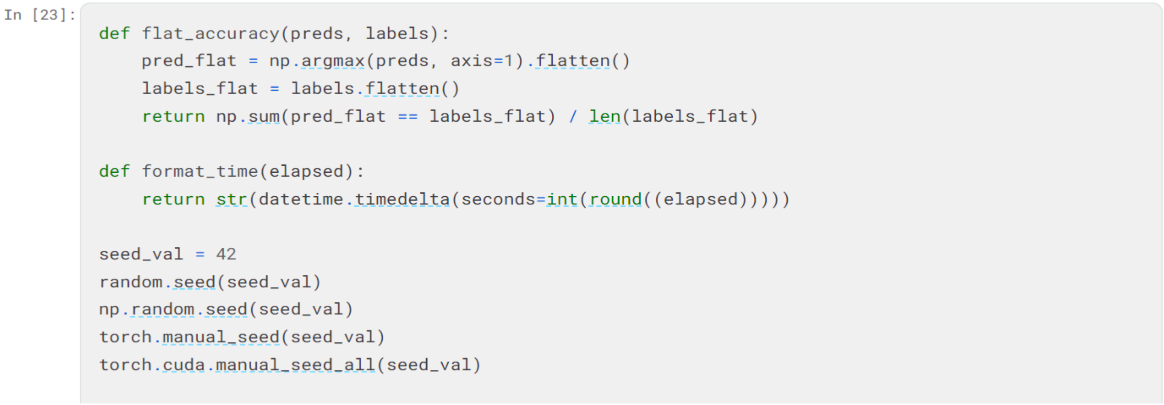
```
from transformers import BertForSequenceClassification, AdamW, get_linear_schedule_with_warmup

model = BertForSequenceClassification.from_pretrained("bert-base-multilingual-cased", num_labels=2)

if torch.cuda.is_available():
    model.cuda()

optimizer = AdamW(model.parameters(), lr=2e-5, eps=1e-8)
epochs = 4
total_steps = len(train_dataloader) * epochs
scheduler = get_linear_schedule_with_warmup(optimizer, num_warmup_steps=0, num_training_steps=total_steps)
```

Figure 17 : Model Initialization & Optimizer and Scheduler Setup

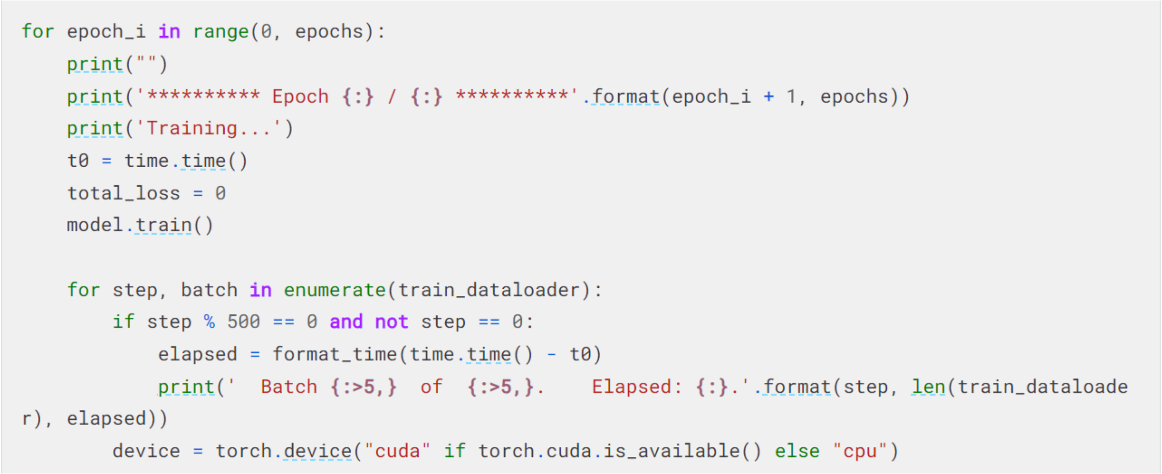
4.5- Pretreatment

Training Loop :

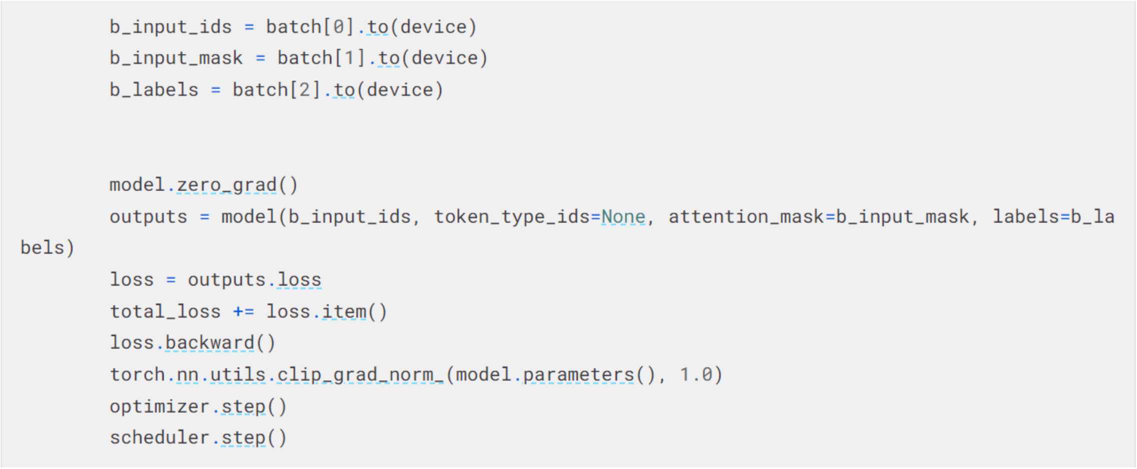
- Set random seeds for reproducibility.
- Train the model for the specified number of epochs:
  - For each batch, move data to the appropriate device (GPU/CPU).
  - Zero the model gradients.
  - Perform forward pass and compute loss.
  - Backpropagate the loss and update the model parameters.
  - Adjust the learning rate using the scheduler.
  - Track and log training progress and loss.
- Evaluate the model on the validation set after each epoch to monitor performance.



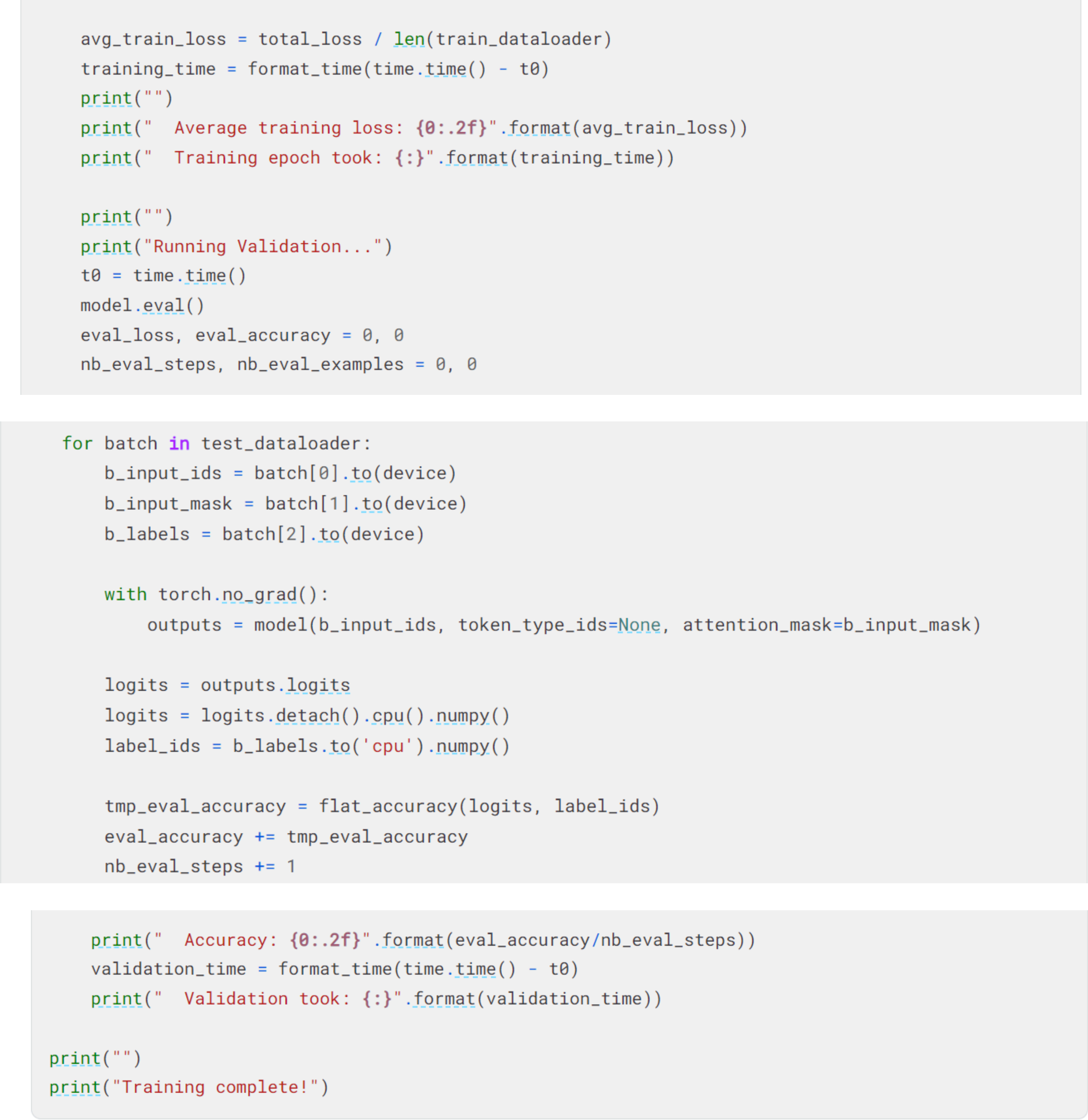
```
In [23]:
def flat_accuracy(preds, labels):
    pred_flat = np.argmax(preds, axis=1).flatten()
    labels_flat = labels.flatten()
    return np.sum(pred_flat == labels_flat) / len(labels_flat)

def format_time(elapsed):
    return str(datetime.timedelta(seconds=int(round((elapsed)))))

seed_val = 42
random.seed(seed_val)
np.random.seed(seed_val)
torch.manual_seed(seed_val)
torch.cuda.manual_seed_all(seed_val)
```



```
for epoch_i in range(0, epochs):
    print("")
    print('********** Epoch {:} / {:} **********'.format(epoch_i + 1, epochs))
    print('Training...')
    t0 = time.time()
    total_loss = 0
    model.train()

    for step, batch in enumerate(train_dataloader):
        if step % 500 == 0 and not step == 0:
            elapsed = format_time(time.time() - t0)
            print('  Batch {:>5,}  of  {:>5,}.    Elapsed: {:}.'.format(step, len(train_dataloade
r), elapsed))
        device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

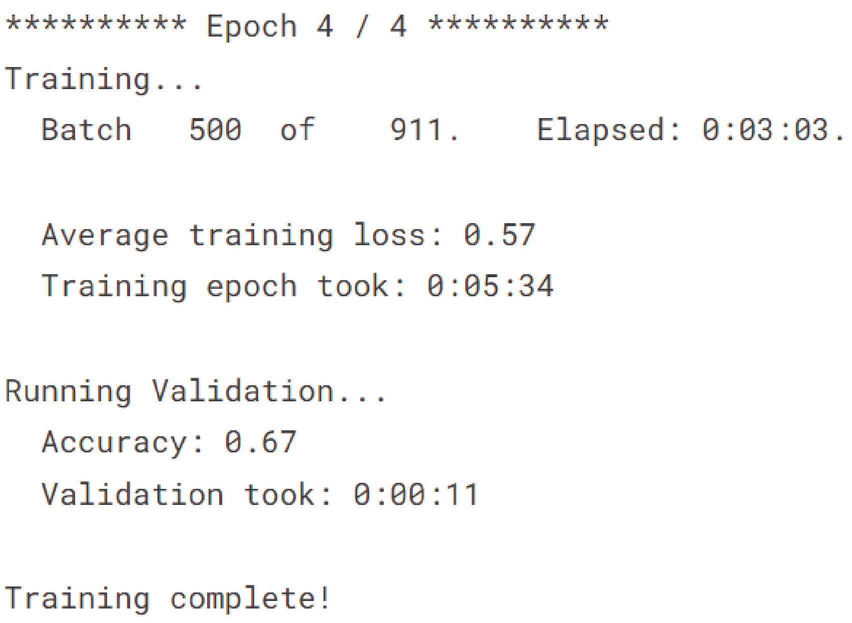


```
        b_input_ids = batch[0].to(device)
        b_input_mask = batch[1].to(device)
        b_labels = batch[2].to(device)


        model.zero_grad()
        outputs = model(b_input_ids, token_type_ids=None, attention_mask=b_input_mask, labels=b_la
bels)
        loss = outputs.loss
        total_loss += loss.item()
        loss.backward()
        torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
        optimizer.step()
        scheduler.step()
```



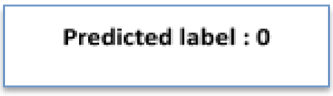
```
    avg_train_loss = total_loss / len(train_dataloader)
    training_time = format_time(time.time() - t0)
    print("")
    print("  Average training loss: {0:.2f}".format(avg_train_loss))
    print("  Training epoch took: {:}".format(training_time))

    print("")
    print("Running Validation...")
    t0 = time.time()
    model.eval()
    eval_loss, eval_accuracy = 0, 0
    nb_eval_steps, nb_eval_examples = 0, 0

    for batch in test_dataloader:
        b_input_ids = batch[0].to(device)
        b_input_mask = batch[1].to(device)
        b_labels = batch[2].to(device)

        with torch.no_grad():
            outputs = model(b_input_ids, token_type_ids=None, attention_mask=b_input_mask)

        logits = outputs.logits
        logits = logits.detach().cpu().numpy()
        label_ids = b_labels.to('cpu').numpy()

        tmp_eval_accuracy = flat_accuracy(logits, label_ids)
        eval_accuracy += tmp_eval_accuracy
        nb_eval_steps += 1

    print("  Accuracy: {0:.2f}".format(eval_accuracy/nb_eval_steps))
    validation_time = format_time(time.time() - t0)
    print("  Validation took: {:}".format(validation_time))

print("")
print("Training complete!")
```

Figure 18 : Training Loop



```
********** Epoch 4 / 4 **********
Training...
  Batch   500  of    911.    Elapsed: 0:03:03.

  Average training loss: 0.57
  Training epoch took: 0:05:34

Running Validation...
  Accuracy: 0.67
  Validation took: 0:00:11

Training complete!
```

Figure 19 : train output



```
Predicted label : 0
```

Figure 20.test output

4.6- Modelistaion

- Prediction Function and Model Evaluation on New Data :

Tokenize and prepare the input sequence.

Move inputs to the appropriate device.

Perform a forward pass through the model without computing gradients.

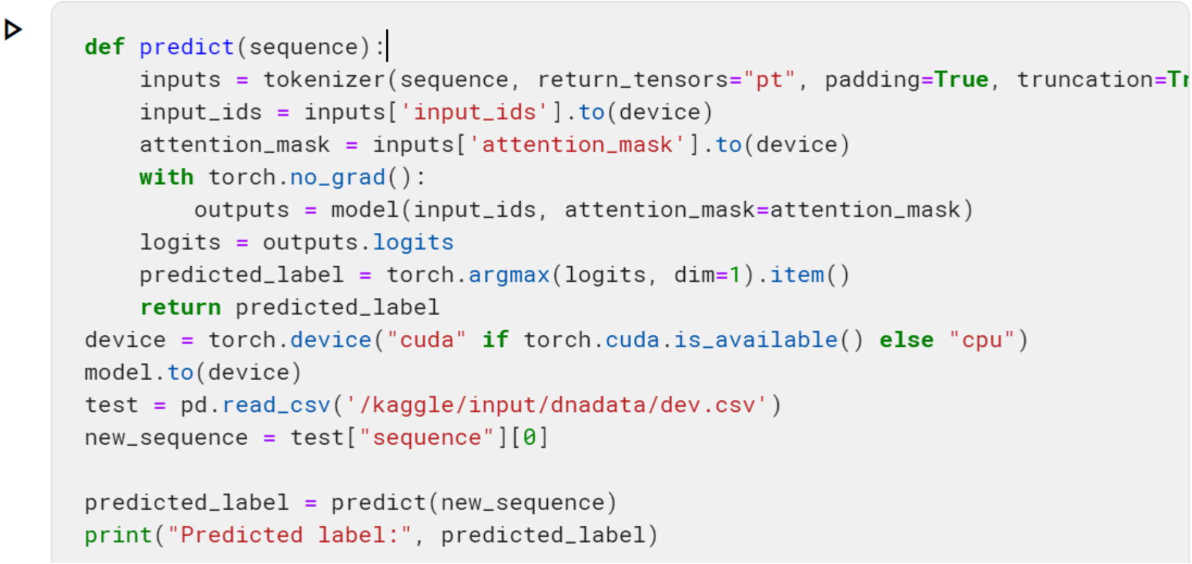
Extract logits and determine the predicted label.

Move the model to the appropriate device.

Read new sequences from the evaluation dataset.

Predict the label for a new sequence using the trained model.

Print the predicted label.



```
def predict(sequence):
    inputs = tokenizer(sequence, return_tensors="pt", padding=True, truncation=Tr
    input_ids = inputs['input_ids'].to(device)
    attention_mask = inputs['attention_mask'].to(device)
    with torch.no_grad():
        outputs = model(input_ids, attention_mask=attention_mask)
    logits = outputs.logits
    predicted_label = torch.argmax(logits, dim=1).item()
    return predicted_label
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
test = pd.read_csv('/kaggle/input/dnadata/dev.csv')
new_sequence = test["sequence"][0]

predicted_label = predict(new_sequence)
print("Predicted label:", predicted_label)
```

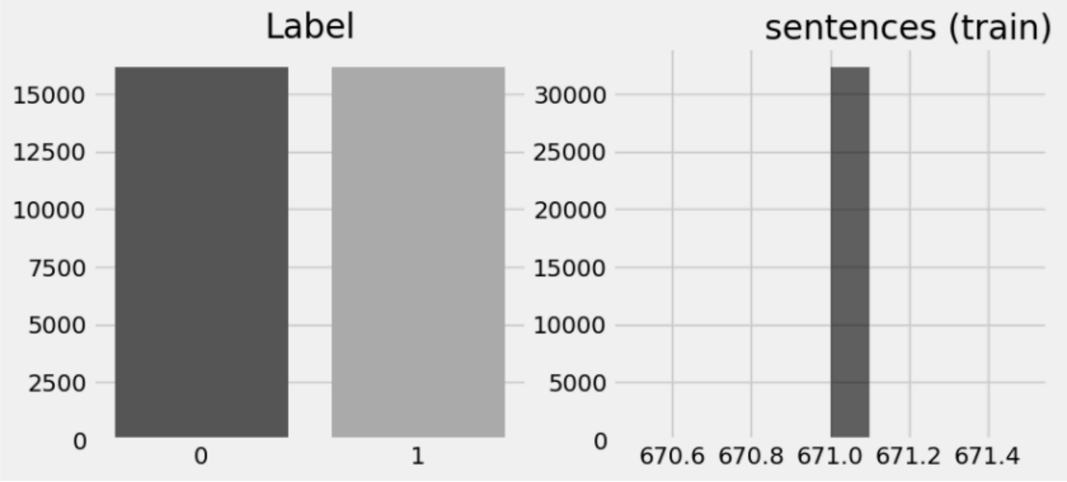


Figure 2: Data visualization (bar chart + histogram)

5- Conclusion :

To conclude this chapter, we used the powerful tools of the Python language and kaggle to conduct our study. We worked with a specific dataset and followed several key steps. Our contribution architecture, illustrated by a diagram, demonstrated our methodical approach and in depth understanding of the problem. Screenshots of the code provided a concrete view of our work. Thanks to this combination of tools and methods, we are able to deliver accurate results, paving the way for future improvements in DNA.

General Conclusion

Genral Conclusion

In our study, we explored how BERT, a natural language processing model can be used to detect patterns, in coding DNA sequences. Though noncoding DNA doesn't encode proteins it constitutes a portion of the genome. Plays a crucial role in gene regulation and other biological processes. Our research indicates that BERT can effectively uncover patterns in these coding regions offering valuable insights into their functional significance.

By leveraging BERT's ability to recognize and model complex sequence relationships we significantly improved the accuracy of pattern recognition tests compared to bioinformatics methods. The model's capacity to identify elements, enhancers, silencers, and other functional attributes in coding DNA showcases its potential to enhance our understanding of genome regulation and complexity.

Our study revealed that tuning BERT with coding DNA data led to enhanced detection of subtle patterns often missed by standard approaches. This capability is essential for advancing genomics research in the realms of gene expression control, epigenetics and identification of coding mutations linked to diseases.

The success achieved through our BERT-based approach underscores the significance of employing NLP techniques in genomic data analysis as a resource, for the bioinformatics community.

More research is needed to improve the model's structure and training methods to unlock its potential, for analyzing coding DNA. In general, our study points towards a path for utilizing BERT in recognizing patterns in coding DNA, which enhances our knowledge of genome regulation and paves the way for new avenues of exploration, in genetics and molecular biology.

References

Refrences

[1] Modan K Das and Ho-Kwok Dai (2007), A survey of DNA motif finding algorithms, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2099490/

[2] Rajat Thapa (2023) DNA: Properties, Structure, Composition, Types, Functions. Microbe Notes. https://microbenotes.com/dna-deoxyribonucleic-acid/

[3] Deoxyribonucleic Acid (DNA) . National Human Genome

Research Institute (2024) https://www.genome.gov/genetics-glossary/Deoxyribonucleic-Acid

[4] “Gloria Lotha”(2024), genetic code, The Editors of Encyclopaedia Britannica https://www.britannica.com/science/genetic-code/additional-info#history

[5] THE FUNCTION OF THE DNA MOLECULE REMAINS ELUSIVE AND LARGELY UNKNOWN

https://www.evolutionisamyth.com/biological/the-volume-of-dna-molecule-illusive-and-unknown/

[6] Noncoding DNA.Noncoding DNA sequences that interrupt functional genes and are removed by splicing once the gene has been transcribed into RNA.

https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/noncoding-dna

[7] panelAnil K. Jain et al. (2016)” 50 years of biometric research: Accomplishments, challenges, and opportunities”. Pattern Recognition Letters https://www.researchgate.net/publication/290509735_50_Years_of_Biometric_Research_Accomplishments_Challenges_and_Opportunities

[8] Sara Assem] DNA Sequencing: Definition, Importance, Methods, Facts, and More.

https://praxilabs.com/en/blog/2021/02/08/dna-sequencing-definition-importance-methods-facts-and-more/

[9] [Sachin Minocha , Suyel Namasudra (2023), Advances in Computers" Chapter Ten - Research challenges and future work directions in DNA computing", https://www.sciencedirect.com/science/article/abs/pii/S006524582200078X?dgcid=rss_sd_all

[10] [Prashanth Pachhi , Ravikumar Manjunath(2021), Analysis of DNA Sequence Pattern Matching A Brief Survey, https://www.researchgate.net/publication/351282144_Analysis_of_DNA_Sequence_Pattern_Matching_A_Brief_Survey

[11] Tuuli Lappalainen, Yang I. Li, Sohini Ramachandran, Alexander Gusev, (2024), Genetic and molecular architecture of complex traits, https://www.sciencedirect.com/science/article/abs/pii/S0092867424000606

[12] Libin Liu, Yee-kin Ho,Stephen Yau,(2006), Clustering DNA sequences by feature vectors, DOI: 10.1016/j.ympev.2006.05.019

[13] James P. Hamilton,(2011),Epigenetics: Principles and Practice, doi: 10.1159/000323874, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3134032/

[14] Nucleic Acids Reset al. (2004). Complexity an internet resource for analysis of DNA sequence complexity.

[15] Oluwafemi A. Sarumi , Maximilian Hahn, Dominik Heider , (2024),NeuralBeds: Neural embeddings for efficient DNA data compression and optimized similarity search.

https://doi.org/10.1016/j.csbj.2023.12.046 , https://www.sciencedirect.com/science/article/pii/S2001037023005214

[16] Sanger Sequencing: Introduction, Principle, and Protocol Posted on February 21, 2020 —

[17] Goodwin, S., McPherson, J. D., & McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. Nature Reviews Genetics, (2016), https://www.nature.com/articles/nrg.2016.49

[18] Kerr, M. K. (2003). Design Considerations for Efficient and Effective Microarray Studies. Biometrics, 59(4), 822–828 , https://pubmed.ncbi.nlm.nih.gov/14969460/

[19] Watson, J.D., Baker, T.A., Bell, S.P., Gann, A., Levine, M., & Losick, R. (2014). Molecular Biology of the Gene (7th ed.). Cold Spring Harbor Laboratory Press. This textbook covers various molecular biology techniques, including RFLP analysis, and discusses their strengths and limitations. https://books.google.dz/books/about/Molecular_Biology_of_the_Gene.html?id=aRUtAAAAQBAJ&redir_esc=y

[20] Pearson, W. R. (2013). An introduction to sequence similarity (“homology”) searching. Current protocols in bioinformatics. https://pubmed.ncbi.nlm.nih.gov/23749753/

[21] Fatma A. Hashim,1 Mai S. Mabrouk,2,* and Walid Al-Atabany1. (2019) Review of Different Sequence Motif Finding Algorithms. Avicenna J Med Biotechnol.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6490410/

[22] Uri Keich, Pavel A. Pevzner,(2002), Subtle motifs: Defining the limits of motif finding algorithms, https://www.researchgate.net/publication/220262468_Subtle_motifs_Defining_the_limits_of_motif_finding_algorithms

[23] A Survey on Knowledge Distillation of Large Language Models (2024) iajia Liu, Mengyuan Yang, Yankai YuBERT Efficacy on Scientific and Medical Datasets: A Systematic Literature Review

[24] Cohn, Clayton., Haixia Xu, Kang Li, Xiaobo Zhou https://arxiv.org/abs/2402.13116

(2020) https://www.proquest.com/openview/65b4cdb2ced1a365a9fe09d5abc9729d/1?

[25] Benyou Wang,Qianqian Xie,Jiahuan Pei (2023), Pre-trained Language Models in Biomedical Domain: A Systematic Survey, DOI: 10.1145/3611651, https://www.researchgate.net/publication/372839615_Pre-trained_Language_Models_in_Biomedical_Domain_A_Systematic_Survey

[26] Why Is Natural Language Processing Needed In Healthcare? Dmitiry Malets(2022)

https://www.linkedin.com/pulse/why-natural-language-processing-needed-healthcare-dmitriymalets

[27] Kevin Bretonnel Cohen ” Biomedical Natural Language Processing and Text Mining Methods”

in Biomedical Informatics.

[28] Li, J., Sun,and al.. (2016). BioCreative V CDR task corpus: a resource for chemical disease relation extraction. Database. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4860626/

[29] Àlex Bravo et al. {2015)“Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research” BMC Bioinformatics. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0472-9

Refrences

[30] Ashokkumar Palanivinayagam et al. (2023) Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review. https://www.mdpi.com/1999-4893/16/5/236

[31] Alexis Allot et al (2019).” LitSense: making sense of biomedical literature at sentence level”

PMID: 31020319 PMCID: https://pubmed.ncbi.nlm.nih.gov/31020319/

[32] Adhika Pramita Widyassari (2022) Review of automatic text summarization techniques & methods Review of automatic text summarization techniques & methods. https://www.sciencedirect.com/science/article/pii/S1319157820303712\

[33] Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., & Liu, H. (2018). Clinical information extraction applications: A literature review. Journal of Biomedical Informatics. https://pubmed.ncbi.nlm.nih.gov/31020319/

[34] Long Chen et al. (2020) .” Clinical concept normalization with a hybrid natural language processing system combining multilevel matching and machine learning ranking”. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7647369/

[35] Cheng-Yuan Liou (2013).” Structural Complexity of DNA Sequence” Comput Math Methods Med. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3638703/

[36] https://theory.labster.com/structure-dna/

[37] Juan A. Montero(2016) DNA damage precedes apoptosis during the regression of the interdigital tissue in vertebrate embryos.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5067507/

[38] Yonatan Belinkov (Tutorial Proposal: Interpretability and Analysis in Neural NLP https://aclanthology.org/2020.acl-tutorials.1.pdf. https://arxiv.org/abs/2007.14128

[39] Kenneth Lo, (2012) Integrating external biological knowledge in the construction of regulatory networks from time-series expression data.

[40] Current genomic deep learning architectures generalize across grass species but not alleles https://bmcsystbiol.biomedcentral.com/articles/10.1186/1752-0509-6-101

View ORCID ProfileTravis Wrightsman,

[41] BUT-FIT at SemEval-2020 Task 5: Automatic detection of counterfactual statements with deep pre-trained language representation models

[42] Kexin Huang (2020).“ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission.” arXiv:https://arxiv.org/abs/1904.05342

[43 Jacob Devlin et al.(2028) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.https://arxiv.org/abs/1810.04805

[44] Gili Nachum.(2024) .”LLM domain adaptation using continued pre-training” . https://medium.com/@gilinachum/llm-domain-adaptation-using-continued-pre-training-part-1-3-e3d10fcfdae1

[45] Peren Jerfi CANATALAY et al (2022).“ A Bidirectional LSTM-RNN and GRU Method to Exon Prediction Using Splice-Site Mapping”.

[46] Ying He, (2021) “A survey on deep learning in DNA/RNA motif mining”. Briefings in BioinformaticsJOURNAL ARTICLE

[47] Junghwan Baek et al (2018). LncRNAnet: long non-coding RNA identification using deep learning. Bioinformatics, Volume

[48] Hemalatha Gunasekaranet al.(2021) “Mathematical Aspects Behind Deep Learning and Transfer Learning Approaches for Medical Image Analysis” Volume 2021 | Article ID 1835056.

[49] Guohua Huang, et al. (2022). Enhancer-LSTMAtt: A Bi-LSTM and Attention-Based Deep Learning

[50] Holur, P., Enevoldsen, K. C., Mboning, L., & Georgiou, T. (2024). Embed-Search-Align: DNA Sequence Alignment using Transformer models. Journal/Conference Name, Volume(Issue).

[51] Shujun He, “ Nucleic Transformer: Classifying DNA Sequences with Self-Attention and Convolutions”. PMCID.

[52]] Shujun He, “ Nucleic Transformer: Classifying DNA Sequences with Self-Attention and Convolutions”. PMCID.

[53] Python : définition et utilisation de ce langage informatique (journaldunet.fr)

[54] https://www.lebigdata.fr/wp-content/uploads/2018/09/python-big-data-machine-learning.jpg

[55] https://www.datacamp.com/blog/what-is-kaggle

الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعة الشهيد الشيخ العربي التبسي
ECHAHID CHEIKH LARBI TEBESSI UNIVERSITY, TEBESSA
كلية العلوم الدقيقة وعلوم الطبيعة والحياة
FACULTY OF EXACT SCIENCES, NATURAL AND LIFE SCIENCES

Département de..math....et....informatique..............
Filière : ....informatique.....................................
Spécialité :.......informatique......................
Année universitaire 2023/2024

Déclaration sur l'honneur de non-plagiat (A joindre obligatoirement avec le mémoire)

Je, soussigné(e)
Nom et prénom : ....Fetni...Atika.....................................
Régulièrement inscrit (e) :
....
N° de carte d'étudiant :
....
Année universitaire :....2023/2024.................................
Domaine :...
Filière : ....informatique...
Spécialité :....systēm...d'information......................................
Intitulé :
............Bert...Based...DNA...pattern.....recognition
....

Atteste que mon mémoire est un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité, je certifie également que je n'ai ni copié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales sont signalées entre guillemets.

Sanctions en cas de plagiat prouvé :

L'étudiant sera convoqué devant le conseil de discipline, les sanctions prévues selon la gravité de plagiat sont :

- L'annulation du mémoire avec possibilité de refaire sur un sujet différent.
- L'exclusion d'une année de Master.
- L'exclusion définitive.

Fait à Tébessa, le :..11./7./..2024......
Signature de l'étudiant (e)

Afetn

الجمهورية الجزائرية الديمقراطية الشعبية

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي و البحث العلمي

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

جامعة الشهيد الشيخ العربي التبسي

ECHAHID CHEIKH LARBI TEBESSI UNIVERSITY, TEBESSA

كلية العلوم الدقيقة وعلوم الطبيعة والحياة

FACULTY OF EXACT SCIENCES, NATURAL AND LIFE

Département de ...math et informatique...

Filière : ...informatique...

Spécialité : ...systém d'information

Année universitaire 2023/2024

Formulaire de levée de réserves après soutenance d'un Mémoire de Master

Données d'identification du candidats (es) :

Nom et prénom du candidat :

...Fetni Atika...

Intitulé du Sujet :

...

...Bert Based DNA pattern recognition...

Données d'identification du membre de jury :

Nom et prénom :

...

Grade :

...

Lieu d'exercice : Université Echahid Cheikh Larbi Tebessi – Tébessa-

Vu le procès-verbal de soutenance de la thèse sus citée comportant les réserves suivantes :

...

R.A.S.

...

Et après constatation des modifications et corrections suivantes :

...

R.A.S.

...

Je déclare en ma qualité de président de jury de soutenance que le mémoire cité remplit toutes les conditions exigées et permet au candidat de déposer son mémoire en vue de l'obtention de l'attestation de succès.

Tébessa le : 04/07/2024

Président de jury de soutenance : (Nom/Prénom et signature)

MGR 2004 soltane