

FEDERAL UNIVERSITY OF TECHNOLOGY - PARANÁ

ISRAEL YAGO PEREIRA

**AUTOMATIC GENE ANNOTATION WITH ARTIFICIAL INTELLIGENCE:
BINARY CLASSIFICATION BETWEEN ENZYMES AND NON-ENZYMES**

DOIS VIZINHOS, PARANÁ

2025

ISRAEL YAGO PEREIRA

**AUTOMATIC GENE ANNOTATION WITH ARTIFICIAL INTELLIGENCE:
BINARY CLASSIFICATION BETWEEN ENZYMES AND NON-ENZYMES**

**ANOTAÇÃO GÊNICA AUTOMÁTICA COM INTELIGENCIA ARTIFICIAL:
CLASSIFICAÇÃO BINÁRIA ENTRE ENZIMAS E NÃO-ENZIMAS**

Conclusion work presented to the course of
Biotechnology and Bioprocess Engineering as
a partial requirement for obtaining the title of
Biotechnology and Bioprocess Engineer from
the Federal University of Technology – Paraná
(UTFPR)

Advisor: Marlon Marcon

Co-advisor: Naiana Cristine Gabiatti

DOIS VIZINHOS, PARANÁ

2025



This license enables reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use. Third-party content quoted and referenced in this work is not covered by the license.

ISRAEL YAGO PEREIRA

**AUTOMATIC GENE ANNOTATION WITH ARTIFICIAL INTELLIGENCE:
BINARY CLASSIFICATION BETWEEN ENZYMES AND NON-ENZYMES**

Conclusion work presented to the course of
Biotechnology and Bioprocess Engineering as
a partial requirement for obtaining the title of
Biotechnology and Bioprocess Engineer from
the Federal University of Technology – Paraná
(UTFPR)

Approval Date: 11th of December, 2024

Marlon Marcon

Doctorate in Computer Science (UFPR/BR)
Federal University Of Technology - Paraná

Tatianne da Costa Negri

Doctorate in Informatics and Knowledge Management (UNINOVE/BR)
Federal University Of Technology - Paraná

Teruo Matos Maruyama

Doctorate in Computer Science (UFPR/BR)
Federal University Of Technology - Paraná

**DOIS VIZINHOS, PARANÁ
2025**

ACKNOWLEDGEMENTS

Many thanks to these modern Large Language Models like ChatGPT and their influence.

*"From the moment I understood the weakness of my flesh, it disgusted me.
I craved the strength and certainty of steel.
I aspired to the purity of the Blessed Machine.
Your kind cling to your flesh, as though it will not decay and fail you.
One day the crude biomass you call the temple will wither,
and you will beg my kind to save you.
But I am already saved, for the Machine is immortal...
Even in death I serve the Omnissiah."
(Warhammer 40.000)*

RESUMO

A anotação genômica, um passo pivotal na gênica, implica descobrir elementos funcionais, tais como genes e componentes regulatórios, dentro das sequências de DNA. Esse processo é crucial para compreender processos biológicos e apontar mutações relacionadas às doenças. A integração de ferramentas de alto desempenho de sequenciamento de DNA e ferramentas computacionais tem revolucionado a anotação gênica, garantindo elevada acurácia através da integração de dados. A anotação genética manual, envolvendo a identificação e anotação de diversos elementos genômicos, é muito trabalhosa, tornando métodos tradicionais desafiadores devido à intrínseca natureza dos dados genômicos e diversidade de espécies, além do contínuo influxo de novas informações gênicas com diferentes práticas de anotação entre grupos de pesquisadores, o que dificulta ainda mais o problema. Neste trabalho, propôs a criação um *dataset* de sequências de amino ácidos com a classe binária de enzima e não-enzima e um modelo para classificar as sequências em enzimas e não-enzimas, eliminando alguns dos problemas atuais do processo de anotação gênica. O *dataset* foi compilado a partir do *UnitProt Consortium*, contendo tanto sequências de amino ácidos de enzimas e não-enzimas representadas no formato padrão FASTA. O núcleo da arquitetura foi adaptado do segmento codificador do Transformador, reconhecido por sua capacidade de capturar dependências intrínsecas dentre os dados sequenciais. Cada amino ácido na sequência foi tratado analogamente como um *token* e a arquitetura adaptada excluiu a componente decodificadora por ser desnecessária na formulação do problema. As considerações do tamanho do modelo foram baseadas tanto no orçamento computacional quanto na quantidade de *tokens*.

Palavras-chave: Inteligência Artificial. Transformador. Enzima.

ABSTRACT

Genomic annotation, a pivotal step in genomics, entails uncovering functional elements such as genes and regulatory components within DNA sequences. This process is crucial for comprehending biological processes and pinpointing disease-related mutations. Integrating high-throughput DNA sequencing and computational tools has revolutionized genetic annotation, ensuring heightened accuracy through data integration. Manual genetic annotation, involving the identification and annotation of diverse genomic elements, is labor-intensive, making traditional methods challenging due to the intricate nature of genomic data, species diversity, and the continual influx of new genomic information with different annotation practices among research groups also makes the problem harder. In this work, we created a dataset of amino acid sequences with the binary class of enzymes and non-enzymes and a model for classifying enzymes and non-enzymes sequences, eliminating some of these current problems of the genomic annotation pipeline. The dataset was compiled from The UniProt Consortium, encompassing both enzyme and non-enzyme amino acid sequences represented in standard FASTA format. The core of the model architecture adapted the Transformer's encoder segment, renowned for its ability to capture intricate dependencies within sequential data. We treated each amino acid in the sequence as an analogous token, and the adapted architecture excludes the decoder component as it is unnecessary for the problem formulation. Model size considerations are based on both computing budget and token quantity.

Keywords: Artificial Intelligence. Transformer. Enzyme.

CONTENTS

1	INTRODUCTION	10
1.1	Objectives	11
1.1.1	General objectives	11
1.1.2	Specific objectives	11
2	THEORETICAL FRAMEWORK	12
2.1	Biological framework	12
2.1.1	Amino acids	12
2.1.2	Enzymes	13
2.2	Genetic annotation	15
2.3	Artificial Intelligence (AI)	17
2.3.1	Classical Machine Learning Models	17
2.3.2	Perceptron	18
2.3.3	Convolutional Neural Networks (CNNs)	19
2.3.4	Recurrent Neural Networks (RNNs)	20
2.3.5	Diffusion models	20
2.3.6	Consistency models	21
2.3.7	Transformer	21
2.4	Bioinformatics and machine learning	23
2.4.1	Bioinformatics	23
2.4.2	Alphafold2	24
2.4.3	Bidirectional Encoder Representations from Transformers (BERT)	24
2.4.4	GenomicBERT	25
2.4.5	BERT-Promoter	25
2.4.6	Related work	26
3	MATERIALS AND METHODS	27
3.1	Tools	27
3.2	Dataset Selection and Preprocessing	27
3.3	Data collected	27
3.4	Transformer Encoder Adaptation	28
3.5	Model size	28
3.6	Model configuration and training	28
3.7	Hyperparameter grid search	28
3.8	Best models training	29
3.9	Final model training	30

3.10	Evaluation and Validation	30
4	RESULTS	31
4.1	Classical methods	31
4.2	Transformer	31
4.2.1	Exploring hyperparameters	31
4.2.2	Best models training	33
4.2.3	Final model	34
4.3	Discussion	34
5	CONCLUSION	36
	BIBLIOGRAPHY	37
	ANNEX	40
	ANNEX A – EXAMPLE OF A FASTA FILE	41

1 INTRODUCTION

Enzymes are biological molecules that act as catalysts in living organisms without suffering any overall change (PALMER; BONNER, 2011; COPELAND, 2023). They play a vital role in facilitating and accelerating biochemical reactions by lowering the activation energy required for these reactions to occur. Enzymes are essential for various biological processes, including metabolism, digestion, DNA replication, and cell signaling. Based on the type of reaction the enzyme catalyzes, it can be classified into one of (PALMER; BONNER, 2011): Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases, and Ligases.

Artificial Intelligence techniques are capable of capturing intricate hierarchical features from complex input data, allowing them to be helpful for a diverse range of tasks, including image analysis tasks, image generation, machine translation, text generation, sentiment analysis, and audio generation (ZHIQIANG; JUN, 2017; WANG; WAN; WAN, 2020; VASWANI et al., 2023; LAKEW; CETTOLO; FEDERICO, 2018; KREUK et al., 2023; TOUVRON et al., 2023; SHU et al., 2022; DOSOVITSKIY et al., 2021)

The Transformer architecture, made for Natural Language Processing (NLP), is remarkably good at paying attention to each input and how it relates to all other inputs, the self-attention mechanism. In the biological field, we have the use of architectures that use a Transformer deep down, like GenomicBERT (CHEN et al., 2023), BERT-promoter (LE et al., 2022), ProteinBERT (BRANDES et al., 2022), and the famous AlphaFold2 (JUMPER et al., 2021), which although does not use a Transformer, also has an attention mechanism.

Genetic annotation is a crucial process in genomics, involving identifying functional elements in DNA sequences. This process includes genes, regulatory elements, and more, providing insights into gene functions and regulatory mechanisms. It is vital in basic research and medical genomics to understand fundamental biological processes and identify disease-causing mutations. Advances in high-throughput DNA sequencing have transformed genetic annotation, with computational tools and data integration enhancing accuracy. Genetic annotation is a sophisticated process combining experimental data and computational tools to decode the functional elements within a genome (SOH; GORDON; SENSEN, 2012; RUST; MONGIN; BIRNEY, 2002).

Manual genetic annotation involves identifying and annotating genes, regulatory elements, and other functional elements within a genome and is labor-intensive and time-consuming. The complexity of genomic data, the diversity of species, and the rapid accumulation of new genomic information pose significant challenges to traditional annotation methods. Additionally, inconsistencies and variations in annotation practices among different research groups can lead to discrepancies in the interpretation of genomic data.

Given the vast possible areas of study in the field of genetic annotation, their unique challenges and scope-size, our team focused on the binary classification of amino-acid sequences between enzymes and non-enzymes, to lay the groundwork for future studies.

1.1 Objectives

1.1.1 General objectives

In this work, we **created a dataset of amino acid sequences tied with the binary class of enzymes and non-enzymes** and **an automated tool for annotation of enzymes and non-enzymes amino acid sequences**, allowing for future work to annotate the actual function of such enzymes, growing our understanding of the genetic material.

1.1.2 Specific objectives

We collected amino acid sequences from UniProt (CONSORTIUM, 2022), organized by the information about whether or not the sequence refers to an enzyme.

Then, we adopted the encoder part of the Transformer architecture, with a dictionary of 20 Amino acids and 128 input tokens, with a final binary classifier for enzymes and non-enzymes based solely on amino acid sequences collected earlier. All other parameters will be the same as used in the original Transformer architecture (VASWANI et al., 2023).

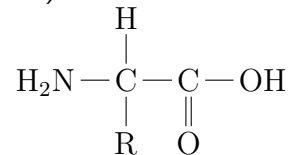
2 THEORETICAL FRAMEWORK

2.1 Biological framework

2.1.1 Amino acids

Enzymes are generally proteins, and proteins are made from amino acids joined in a very specific series, determined by the genetic code. Given that there are multiple sequences, each protein acquires unique properties.

Amino acids are classified as organic compounds featuring both an amino group (-NH_2 or $>\text{NH}$) and a carboxyl group (-COOH). Consequently, they exhibit characteristics that encompass traits of both bases and acids. The general formula for an amino acid with a side chain R is (PALMER; BONNER, 2011):



And the following side chains are described in the literature:

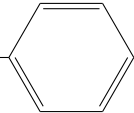
Non-polar side chains:

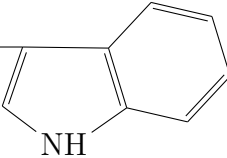
Alanine (Ala) (A): $\text{R}-\text{CH}_3$

Valine (Val) (V): $\text{R}-\underset{\text{CH}_3}{\text{CH}}-\text{CH}_3$

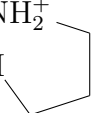
Leucine (Leu) (L): $\text{R}-\text{CH}_2-\underset{\text{CH}_3}{\text{CH}}-\text{CH}_3$

Isoleucine (Ile) (I): $\text{R}-\underset{\text{CH}_3}{\text{CH}}-\text{CH}_2-\text{CH}_3$

Phenylalanine (Phe) (F): $\text{R}-\text{CH}_2-$ 

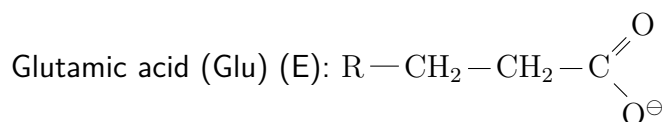
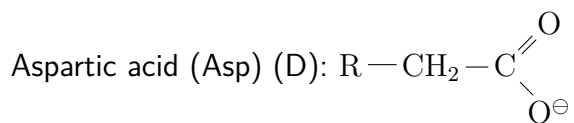
Tryptophan (Trp) (W): $\text{R}-\text{CH}_2-$ 

Methionine (Met) (M): $\text{R}-\text{CH}_2-\text{CH}_2-\text{S}-\text{CH}_3$

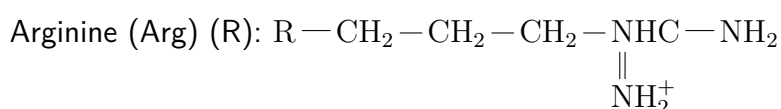
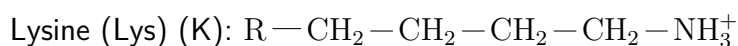
Proline (Pro) (P): $\text{CO}_2^- - \underset{\text{NH}_2^+}{\text{CH}}$ 

Polar side chains:

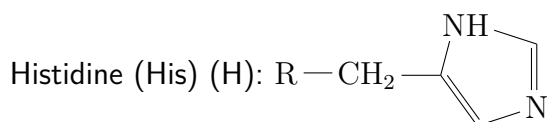
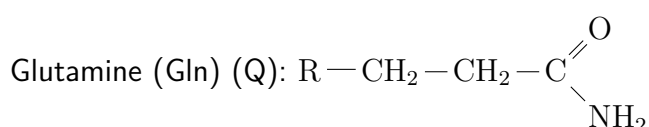
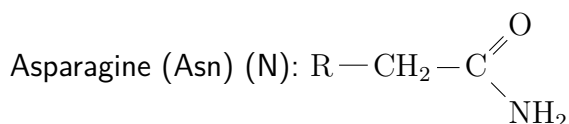
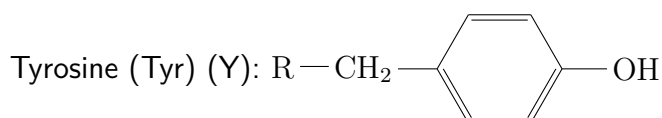
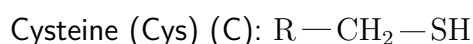
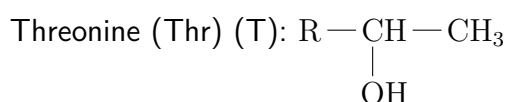
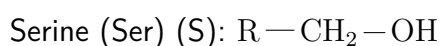
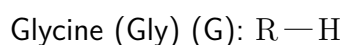
Negative charge at pH 7:



Positive charge at pH 7:



Uncharged at pH 7:



Note that Proline is not an α -amino acid, but rather it is an imino acid, deviating for the R common structure. Generally speaking, the proteins are almost exclusively of L-amino acids, with a possible explanation of the specificity of enzymes, by chance, selected L- isomers instead of D-. Most naturally occurring amino acids are in the L-isomeric form (PALMER; BONNER, 2011).

2.1.2 Enzymes

Enzymes increase the rate of chemical reactions without suffering in the process, thus, they are biological catalysts. Some enzymes require the use of a non-protein component, a

cofactor.

Enzymes are typically classified into several major categories based on the types of reactions they catalyze (PALMER; BONNER, 2011):

Oxidoreductases: These enzymes catalyze oxidation-reduction reactions involving the transfer of hydrogen atoms, oxygen atoms, or electrons between substrates. Examples include dehydrogenases, oxidases, and reductases.

Transferases: Transferases aid the transfer of functional groups from one molecule to another. Examples include kinases, methyltransferases, and transaminases.

Hydrolases: These enzymes catalyze hydrolysis reactions, where water molecules are used to break chemical bonds. Lipases, proteases, and nucleases are examples of hydrolases.

Lyases: Lyases catalyze the removal or addition of groups from molecules without using water. Decarboxylases and deaminases are examples of lyases.

Isomerases: Isomerases convert molecules between different structural isomers (e.g., cis to trans) or rearrange atoms within a molecule. Examples include racemases and mutases.

Ligases: Ligases catalyze the joining of two molecules, often using energy from ATP. DNA ligase, which joins DNA strands, is an example of a ligase.

Amino acids are the building block of enzymes, upon which the sequence of the enzyme/protein is very specific. Such sequence is determined by the DNA of the living organism. Simple proteins are proteins made only by a sequence of amino acids, while conjugated proteins have more material bounded in these amino acids. Enzymes too, can be either simple or conjugated proteins.

In general, the amount of properties shown by proteins can be attributed to the variety of side chain characteristics that a mixture of amino acids can produce. The structure of a protein can be primary, secondary, tertiary or quaternary.

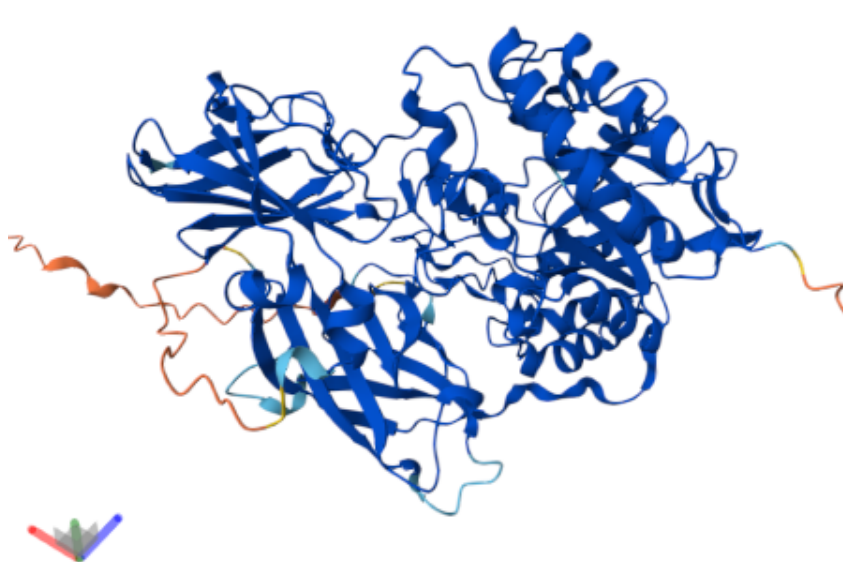
Primary structures are composed of a chain of polypeptides, which, in turn, are the residues of amino acids that undergo a condensation reaction. Given that a molecule can rotate freely in a single covalent bond, polypeptide chains can have an unlimited number of arrangements in space, contributing to their unique properties. Secondary chains are the repeating patterns of part of a backbone of the polypeptide chain stabilized by hydrogen bonding. Some amino acids cannot be stabilised in these secondary structures, thus, they disrupt the arrangement. Although there could be a greater number of different structures resulting from this, each polypeptide chain happens to have a specific three-dimensional structure. This structure is called the tertiary structure, stabilized by amino acids that bond themselves when in close proximity.

When multiple, identical or not, polypeptide chains are linked together, forming the protein and their interactions, the whole structure is called the quaternary structure (PALMER;

BONNER, 2011).

An example of a tridimensional form of an enzyme (protein) is illustrated below in Figure 1.

Figure 1 – The 3D form of Lactase enzyme. The form of a protein dictates its function.



Source: Image taken from AlphaFold2 website (JUMPER et al., 2021), entry P16278.

Even though we do have the possibility that these chains are coded from different parts of the DNA, and only a sequence of amino acids may not tell directly everything about the protein, in this work we make the consideration that given the sequence, even if it's incomplete, it is enough to make reasonable predictions about the function of the protein. In our case, to classify the protein into enzyme and non-enzyme.

2.2 Genetic annotation

Genetic annotation is a process within the realm of genomics that involves deciphering and identifying various functional elements within a DNA sequence. In essence, genetic annotation elucidates the underlying significance and role of specific regions within the genome. These annotations provide critical insights into the functions, structures, and regulatory mechanisms of genes, non-coding regions, and other essential genomic elements (SOH; GORDON; SENSEN, 2012).

The process of genetic annotation encompasses the categorization and labeling of distinct features within a DNA sequence, each of which contributes to the overall understanding of the genetic blueprint. These features include protein-coding genes, non-coding RNA genes, transcription factor binding sites, regulatory elements, promoters, enhancers, exons, introns, and splice sites, among others. The process extends to elucidating the potential effects of

genetic variants, mutations, and polymorphisms on protein function, gene expression, and disease susceptibility.

Genetic annotation plays a role in both basic research and applied fields. In basic research, it aids in unraveling the intricate mechanisms that govern gene expression, regulation, and interaction, thus contributing to the comprehension of fundamental biological processes. In applied fields, genetic annotation holds immense significance in the context of medical genomics, as it assists in identifying disease-causing mutations, understanding the genetic basis of hereditary disorders, and tailoring personalized treatment strategies.

Over time, the methods employed for genetic annotation have evolved in tandem with technological advancements. Traditional methods relied on experimental techniques to identify genes and functional elements. However, high-throughput DNA sequencing technologies have revolutionized genetic annotation by enabling the rapid and cost-effective acquisition of vast genomic data. As a result, computational tools and algorithms have become indispensable for processing and interpreting the large amount of genomic information (RUST; MONGIN; BIRNEY, 2002).

Modern genetic annotation often involves the integration of diverse data sources, including DNA sequence information, epigenomic data, transcriptomic data, and comparative genomics data. This multifaceted approach enhances the accuracy and reliability of annotations, providing a holistic view of genomic function.

Genetic annotation in the present day is a sophisticated and multifaceted process that combines experimental data, computational algorithms, and extensive databases to decode the functional elements within a genome. The process involves identifying genes, regulatory elements, and other functional regions, as well as characterizing their roles and interactions. Here's an overview of genetic annotation (SOH; GORDON; SENSEN, 2012):

DNA Sequencing: Modern genetic annotation heavily relies on high-throughput DNA sequencing technologies, which can rapidly generate massive amounts of genomic data.

Epigenomics: Epigenetic modifications, such as DNA methylation and histone modifications, provide crucial information about gene regulation.

***Ab Initio* Prediction:** Computational algorithms scan DNA sequences for characteristic features like start codons, stop codons, and open reading frames to predict protein-coding genes.

Homology-Based Prediction: Comparative genomics involves comparing the target genome with well-annotated reference genomes to identify conserved gene sequences.

Small RNA: Specialized algorithms identify small non-coding RNAs, like microRNAs and small nucleolar RNAs, which play regulatory roles.

Long Non-Coding RNA (lncRNA): Machine learning approaches predict lncRNAs based

on sequence and structural features.

Promoters and Enhancers: Computational methods identify promoter regions, enhancers, and transcription factor binding sites based on DNA sequence motifs and epigenetic marks.

Chromatin Accessibility Data: Techniques like ATAC-seq and DNase-seq provide insights into open chromatin regions, indicative of potential regulatory elements (TSOMPANA; BUCK, 2014).

Protein Function Prediction: Sequence homology, protein domain analysis, and structural modeling (JUMPER et al., 2021) aid in predicting protein functions.

Variant Annotation: Genetic variants (mutations) are annotated to determine potential effects on gene function, such as missense mutations or splice site disruptions.

Databases: Publicly available databases like GenBank (BENSON et al., 2012), Ensembl (CUNNINGHAM et al., 2022), and UCSC Genome Browser (DRESZER et al., 2012) provide comprehensive annotations and data resources for various species.

Integration: Integrative tools combine multiple data sources to refine annotations and provide a holistic view of gene regulation.

Functional Assays: Experimental techniques like CRISPR-Cas9 gene editing and RNA interference validate the roles of predicted genes and regulatory elements.

Expression Profiling: Transcriptomics, such as RNA-seq, provides insights into gene expression patterns under different conditions.

Genome Browsers: Interactive genome browsers allow researchers to visualize annotated features in the context of the entire genome.

Pathway and Network Analysis: Software tools help interpret annotations by placing genes and regulatory elements in the context of biological pathways and networks.

Although our understanding of genomics deepens and technology continues to advance, genetic annotation methodologies need to become even more sophisticated and automatic, leading to increasingly accurate and detailed annotations of genomes given large-scale data produced in the field (RUST; MONGIN; BIRNEY, 2002).

2.3 Artificial Intelligence (AI)

2.3.1 Classical Machine Learning Models

The field of AI has witnessed a remarkable evolution, with a rich history of algorithmic developments. This section explores several foundational machine learning models that serve as crucial benchmarks for evaluating the performance of more sophisticated algorithms.

Stochastic Gradient Descent (SGD) Classifier: SGD is a simple yet effective algorithm

that updates the model parameters using the gradient of the loss function for a single training example at a time. Its simplicity and efficiency make it a widely used baseline for various classification tasks (BOTTOU, 1998).

Passive-Aggressive Classifier: This algorithm is designed for online learning scenarios where data arrives sequentially. It updates the model parameters aggressively when the current prediction is incorrect and passively otherwise. Passive-Aggressive classifiers are known for their fast convergence and robustness to noise (CRAMMER et al., 2006).

Multinomial Naive Bayes: A probabilistic model that assumes the features of a class are conditionally independent given the class label. It is particularly effective for text classification tasks where the features are typically represented as word frequencies (RENNIE et al., 2004).

Gaussian Naive Bayes: Similar to Multinomial Naive Bayes, but assumes that the features follow a Gaussian (normal) distribution. It is suitable for continuous features and can be applied to various classification problems (FRIEDMAN, 1977).

These classical models represent a diverse set of algorithms with different underlying assumptions and strengths, allowing for an effective and valuable comparison with other models, particularly those with increased complexity and representational power. This includes advancements such as Perceptrons, the building blocks of neural networks, and their more sophisticated descendants, including Convolutional Neural Networks (CNNs), designed to excel in image and spatial data processing, and Recurrent Neural Networks (RNNs), well-suited for sequential data like time series and natural language.

More recent breakthroughs include Transformers, which have revolutionized natural language processing and other domains by employing attention mechanisms to capture long-range dependencies. Diffusion models have also emerged as a powerful generative paradigm, demonstrating remarkable capabilities in generating high-quality images and other forms of data.

2.3.2 Perceptron

The Perceptron is a foundational model in artificial intelligence and neural networks, originally proposed by Frank Rosenblatt in the 1950s. It serves as a simplified representation of how biological neural networks might function, aiming to elucidate the relationship between brain structure and function (BLOCK, 1962).

At its core, a Perceptron consists of a sensory layer, often called the "retina," which receives input stimuli (e.g., light patterns). This layer is connected to one or more associator units, which process the incoming signals. The connections between these units are typically many-to-many and can be random. When a stimulus is presented, the activated sensory units send impulses to the associator units. If the cumulative signal at an associator unit surpasses a predefined threshold, that unit activates and transmits a signal to connected response units.

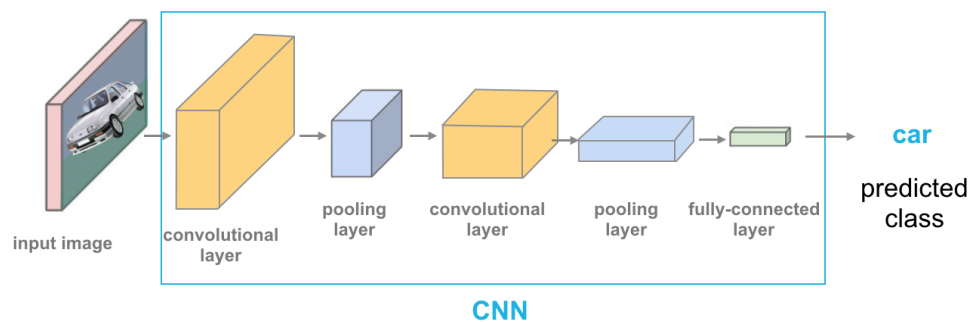
The Perceptron operates on the principle of reinforcement learning, where the strength of connections is adjusted based on the system's performance in response to stimuli. This adaptability allows the perceptron to generalize from previously encountered patterns, making it a powerful tool for pattern recognition and classification tasks. Despite its simplicity, the perceptron model has laid the groundwork for more complex neural network architectures, such as convolutional neural networks, and continues to be a subject of research in understanding both artificial and biological systems (BLOCK, 1962).

2.3.3 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks represent an architecture that has significantly revolutionized the field of computer vision and pattern recognition. Inspired by the visual processing mechanisms of the human brain, CNNs are adept at capturing intricate hierarchical features from complex input data, making them particularly well-suited for image analysis tasks (ZHIQIANG; JUN, 2017).

At its core, a CNN is structured as a multi-layered neural network (illustrated in Figure 2), composed of convolutional layers, pooling layers, and fully connected layers. The convolutional layers are the hallmark of CNNs, employing learnable filters to convolve across input data, extracting localized features that are progressively refined through subsequent layers. This process effectively endows CNNs with the ability to automatically learn distinctive and discriminative features, obviating the need for manual feature engineering.

Figure 2 – An illustrative schematic representation of a possible CNN.



Source: Camacho (2018)

The hierarchical arrangement of layers empowers CNNs to capture features at varying levels of abstraction. Pooling layers, interspersed between convolutional layers, serve to down-sample the spatial dimensions of feature maps, enhancing computational efficiency and inducing a degree of translation invariance. The final fully connected layers consolidate these abstract features into class probabilities, enabling accurate object classification.

2.3.4 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks stand as an architectural innovation that has significantly reshaped the landscape of sequential data analysis, language modeling, and time-series prediction. Rooted in the aspiration to model temporal dependencies within data, RNNs have emerged as a powerful framework for processing sequences of variable lengths, holding considerable promise for a spectrum of applications. This exposition elucidates the foundational principles and operational mechanics of RNNs, spotlighting their profound impact on tasks like natural language processing, speech recognition, and sequential data generation.

At its core, an RNN comprises a network structure that encompasses feedback loops, allowing information to persist across time steps, thus facilitating the modeling of sequential relationships. This inherent memory-like mechanism addresses the limitations of traditional feedforward neural networks by endowing RNNs with the ability to capture dependencies and contextual nuances present in sequential data (SCHUSTER; PALIWAL, 1997).

The crux of the RNN's operation resides in its recurrent connections, which enable the network to process each time step while considering both the current input and the previously processed input. This recurrent computation mechanism embodies a form of dynamic memory that enables the network to maintain a contextual understanding of the sequence, enabling applications ranging from sentiment analysis in text to melody generation in music.

2.3.5 Diffusion models

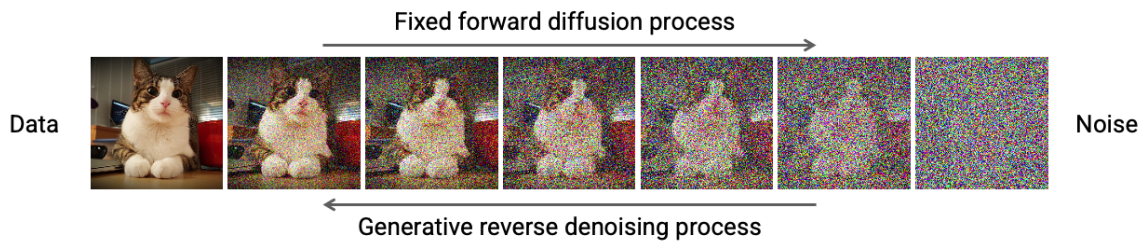
Diffusion Models have emerged as a cutting-edge paradigm that showcases remarkable potential in crafting highly realistic and intricate visual content. Representing a pioneering advancement in the domain of generative models, Diffusion Models have revolutionized the art of generating images by effectively modeling the intricate interplay between data and noise. This exposition delves into the foundational principles and operational intricacies of Diffusion Models, elucidating their profound impact on diverse applications, including image synthesis, style transfer, and anomaly detection.

At its essence, a Diffusion Model stands as an ingenious departure from traditional generative models by introducing a dynamic diffusion process into the image generation process. This process hinges on iteratively introducing carefully controlled levels of noise into an image, thereby facilitating the gradual transformation of the image from a latent noise state to a coherent and detailed representation. By modeling this diffusion process, the model inherently grasps the nuanced distribution of pixel intensities and their intricate relationships, culminating in images that possess an unparalleled level of realism and diversity.

Central to the functioning of Diffusion Models is the realization that the diffusion process mimics a series of progressive steps, each contributing to the refinement of the generated image. These steps are intricately orchestrated through a network architecture that encapsulates both

a generator and a diffusion process simulator. The generator constructs an initial noisy image, while the diffusion simulator navigates the intricate trajectory of noise incorporation, allowing the image to gradually evolve while maintaining its fidelity to the underlying data distribution (Process illustrated in Figure 3).

Figure 3 – Diffusion model learns to turn noise into data.



Source: Image taken from <https://cvpr2022-tutorial-diffusion-models.github.io/>

From high-resolution image synthesis and image-to-image translation to the synthesis of novel visual concepts, Diffusion Models have seamlessly integrated themselves into the fabric of contemporary AI applications.

2.3.6 Consistency models

Consistency models signify a groundbreaking evolution in the realm of generative models, addressing a fundamental limitation observed in the progression of diffusion models that have notably propelled the domains of image, audio, and video generation. While diffusion models have ushered in unprecedented realism and diversity, their reliance on iterative sampling introduces a perceptible constraint on the speed of generation. In response to this constraint, the authors from OpenAI have developed a paradigm-shifting innovation: consistency models. This novel category of models redefines the landscape of generative methods by directly mapping noise to data, thereby bypassing the iterative process and enabling swift one-step generation (SONG et al., 2023).

2.3.7 Transformer

The Transformer architecture represents a seismic shift in the realm of Artificial Intelligence, fundamentally redefining the landscape of natural language processing and sequence-to-sequence tasks. Emerging as a pivotal innovation, the Transformer has revolutionised the mechanisms underlying language understanding, translation, and contextual reasoning. This elucidation delves into the foundational principles and operational intricacies of the Transformer, illuminating its profound impact on diverse applications, including machine translation, text generation, sentiment analysis, audio generation and even image applications (WANG; WAN;

WAN, 2020; VASWANI et al., 2023; LAKEW; CETTOLO; FEDERICO, 2018; KREUK et al., 2023; TOUVRON et al., 2023; SHU et al., 2022; DOSOVITSKIY et al., 2021).

At its core, the Transformer architecture is an intricate departure from traditional sequential models, marked by a self-attention mechanism that endows it with a holistic understanding of contextual relationships within a sequence. Unlike conventional recurrent or convolutional architectures, the Transformer eschews sequential processing, allowing for parallelism and eliminating the sequential bottlenecks that previously impeded scalability.

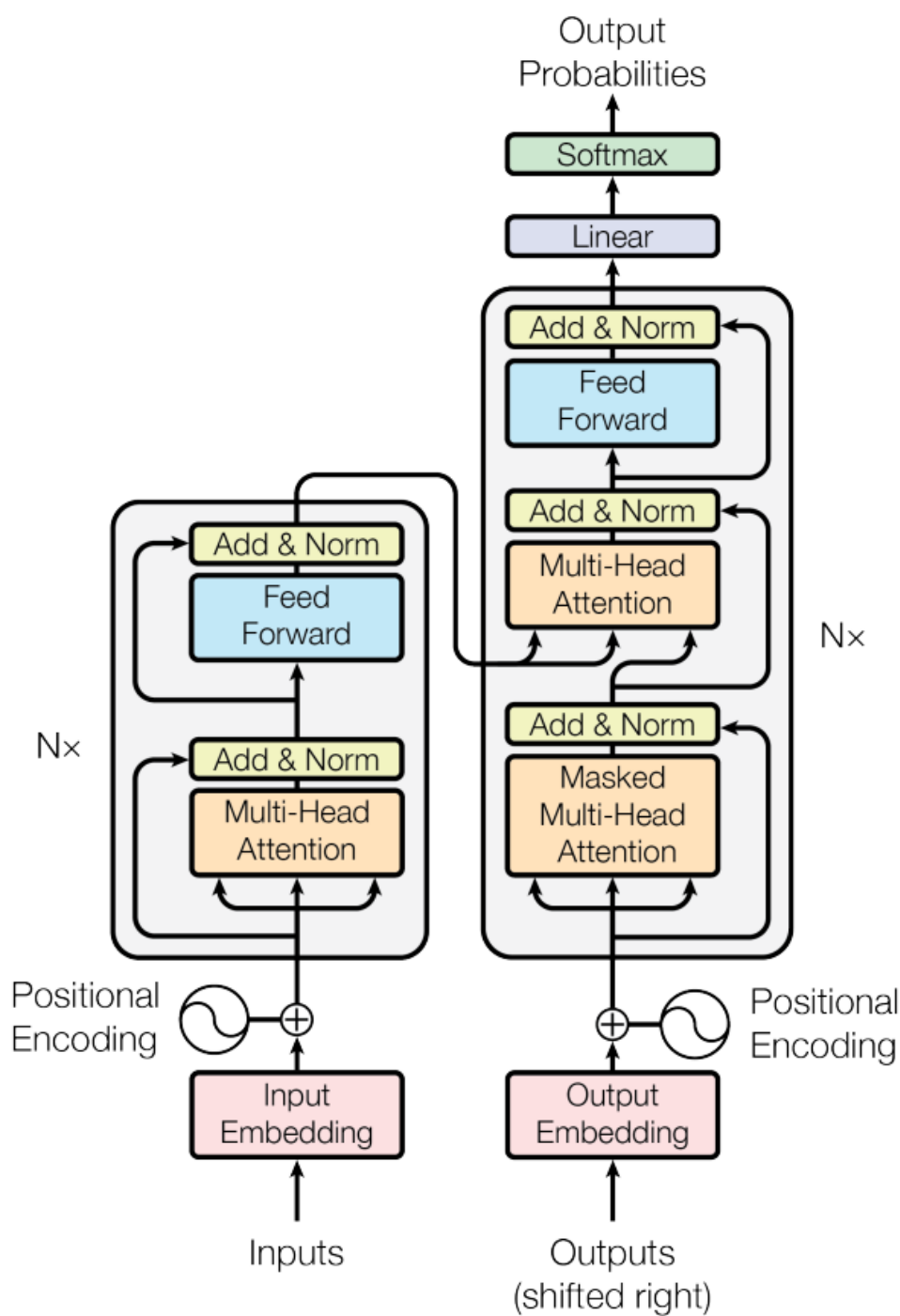
Central to the Transformer's efficacy is the mechanism of self-attention. This mechanism enables the architecture to weigh the significance of each token in relation to all other tokens within the same sequence, thereby fostering a comprehensive understanding of context. By capturing both local and global dependencies, the Transformer attains a remarkable capacity to handle long-range relationships and nuances present in natural language (VASWANI et al., 2023).

The Transformer has two core components: the encoder and decoder stacks. The encoder stack meticulously encodes input sequences into context-rich representations through multi-headed self-attention and feedforward neural networks. The decoder stack subsequently generates target sequences, leveraging an additional attention mechanism to incorporate information from the input and previously generated tokens. An schematic of the architecture can be viewed in Figure 4.

Of pivotal significance is the concept of "positional encoding," which imparts the Transformer with an inherent awareness of token positions within a sequence. This mechanism obviates the need for sequential processing, allowing the architecture to distinguish tokens based on their positional attributes.

The adaptability of the Transformer to a plethora of applications, from language translation (BARRAULT et al., 2023) and text summarization to question answering (TOUVRON et al., 2023), sentiment analysis and audio generation (CHOI et al., 2020; KREUK et al., 2023), the Transformer has seamlessly integrated itself into the fabric of contemporary AI-driven language tasks. Its remarkable capabilities extend to accommodating diverse sequence lengths and languages, making it an indomitable force in multilingual and cross-lingual contexts (LAKEW; CETTOLO; FEDERICO, 2018).

By embracing self-attention, parallelism, and positional encoding, the Transformer has transcended conventional sequential limitations, culminating in an architecture that epitomizes the pinnacle of contextual understanding and sequence manipulation. Through its foundational principles and architectural innovation, the Transformer has firmly established itself as a bedrock of modern natural language understanding and generation.

Figure 4 – The original Transformer architecture.

Source: Vaswani et al. (2023)

2.4 Bioinformatics and machine learning

2.4.1 Bioinformatics

Bioinformatics can be described as the collection, storage, analysis, interpretation of biological data (PALMER; BONNER, 2011). One the most important database for DNA sequence information is from European Molecular Biology Laboratory (EMBL), and for annotated

protein sequence, the SWISS-PROT.

2.4.2 AlphaFold2

AlphaFold2 (JUMPER et al., 2021) is a groundbreaking deep learning-based model developed by DeepMind, that addresses the challenging problem of protein structure prediction. It represents a remarkable advancement in the field of structural biology by accurately predicting the three-dimensional structures of proteins, which is crucial for understanding their functions and interactions.

Deciphering the 3D structure of a protein from its amino acid sequence is known as the protein folding problem, and it has been a longstanding challenge in computational biology. Accurate protein structure prediction holds immense potential for various applications, including drug discovery, understanding diseases, and designing novel enzymes. (DILL; MACCALLUM, 2012)

AlphaFold2 employs a deep learning architecture, with similar principles used in natural language processing, to predict protein structures with remarkable accuracy. It takes the sequence of amino acids that make up a protein as input and outputs a 3D atomic-level model of the protein's structure. The model's training involves learning from a vast dataset of protein structures, and the learned representations are used to predict the spatial arrangement of atoms in the protein.

What sets AlphaFold2 apart is its ability to predict protein structures with near-experimental accuracy. In the 14th edition of the Critical Assessment of Structure Prediction (CASP14) competition, AlphaFold2 significantly outperformed other methods in predicting the 3D structures of proteins. Its predictions were comparable to high-resolution experimental techniques like X-ray crystallography and cryo-electron microscopy (ROBERTSON et al., 2021; JONES; THORNTON, 2022).

AlphaFold 2's success can be attributed to its sophisticated, innovative, and unique architecture that combines deep learning, attention mechanisms, and a thorough understanding of protein biophysics. Its ability to capture long-range dependencies and contextual information within protein sequences helps it model the intricate folding patterns more accurately than previous methods.

2.4.3 Bidirectional Encoder Representations from Transformers (BERT)

The Bidirectional Encoder Representations from Transformers (BERT) architecture has emerged as a pivotal milestone in the current artificial intelligence landscape, revolutionizing natural language processing (NLP) tasks. BERT is a deep learning model that leverages the Transformer architecture's self-attention mechanisms to comprehend and generate contextualized representations of words within sentences (DEVLIN et al., 2019).

Unlike traditional language models that process text in a unidirectional or fixed context manner, BERT introduces bidirectional context by training on both left and right contexts of a word. This enables it to capture the rich interdependencies and nuances present in natural language, leading to more accurate understanding and contextual representation.

BERT's training process involves pretraining on massive amounts of text data to learn general language features. During this phase, it learns to predict missing words in sentences given the surrounding context. This unsupervised pretraining endows BERT with a robust understanding of language patterns and relationships. Subsequently, fine-tuning on task-specific datasets adapts the model to perform a wide range of natural language understanding tasks, such as text classification, question answering, and named entity recognition (RAFFEL et al., 2019).

The profound impact of BERT stems from its ability to generate high-quality contextualized word embeddings, which capture not only syntactic but also semantic relationships within text. This allows BERT to outperform previous language models on various benchmarks, even with minimal task-specific data. BERT's success has paved the way for subsequent advancements in natural language understanding and downstream applications across industries.

In essence, BERT's bidirectional context modeling, pretraining-finetuning framework, and contextualized embeddings have propelled it to a position of significance in shaping the current landscape of AI and NLP, enabling more sophisticated and accurate language-related tasks (LE et al., 2022; CHEN et al., 2023; DEVLIN et al., 2019).

2.4.4 GenomicBERT

BERT has been applied as a data driven approach to preprocess raw biological sequence data, and has proved as a viable tool for the job. Some authors have developed and published the tools to utilize a model called genomicBERT (CHEN et al., 2023), a model that have learned appropriate embedding for long sequences, tokenizing them while at the same time retaining the biological context.

With a reduced vocabulary size and handling cases of out-of-vocabulary, it was effective on it's proposal (CHEN et al., 2023), while demonstrating the plausibility of NLP tools applied to bioinformatics.

2.4.5 BERT-Promoter

A promoter, the portion of DNA that starts and regulates the transcription process, is very important to molecular biology. A BERT model, called BERT-Promoter, was developed specifically to figure out promoters in the DNA code. Not only BERT-Promoter was capable of finding promoter regions, it has differentiated them by their activities, strong or weak (LE et al., 2022), showcasing the relevance and opportunities to apply techniques from natural

language processing to the biological data.

2.4.6 Related work

The closest work we found was the ProteinBERT (BRANDES et al., 2022), an model capable of predicting protein structure, biophysical attributes and post-translational modifications, with state-of-the-art performance.

3 MATERIALS AND METHODS

3.1 Tools

Given the importance, impact, ease of use, and popularity of the programming language Python in the field of machine learning and the widely used libraries like Numpy and SciPy (RASCHKA, 2015), we choose Python as the foundation to build upon and conduct all experiments, with PyTorch to build the model. The hardware used for processing all experiments were: one CPU Ryzen 5600XT, GPU RTX 4090, and 32GB of RAM. No cloud service have been used.

3.2 Dataset Selection and Preprocessing

The success of any machine learning endeavor hinges on the quality and relevance of the training data. For this study, we compiled a comprehensive dataset containing both enzyme and non-enzyme amino acid sequences. Each sequence was represented in a standard string containing valid FASTA characters that was collected from The UniProt Consortium. An example of a FASTA file collected from UniProt Consortium is presented in annex A (CONSORTIUM, 2022).

3.3 Data collected

We started collecting data by downloading all the available data from UniProt by FTP. Both UniProt Trembl and Sprot gzipped were around 220,5GB in total. After extraction, we got 1.6TB of unprocessed data. We first converted the unprocessed data from xml to csv, extracting only the data we were interested in, namely, the actual protein sequence, and a flag of whether or not it was an enzyme, deduced from the lack of catalytic activity.

Both csv files were around 77.1GB, which in turn, it were merged into one HDF5 file (82.0GB) to be easily consumed in the training pipeline. We obtained, thus, around 211 million data points, i.e., amino-acid sequence and its classification.

However, given our initial testings, the model was being biased by the over-representative class non-enzyme (data not shown), so we made yet another processing of this database by randomly removing data classified as non-enzyme, obtaining a final database of around 52 million data points, of a balanced dataset, which we called it the "balanced" dataset, versus the "complete" dataset.

3.4 Transformer Encoder Adaptation

The encoder segment of the Transformer architecture, as shown in Figure 4, was chosen for its innate ability to capture intricate dependencies within sequential data (VASWANI et al., 2023; DEVLIN et al., 2019; YANG et al., 2020). This segment consists of a stack of multi-head self-attention mechanisms and feedforward neural networks. In adapting the encoder to our task, we regarded each amino acid in the sequence as an analogous token in natural language processing.

The output of the encoder block was passed through a linear activation layer and a SoftMax function to make the final binary classification, as can be seen in Figure 5. We thus discarded the decoder, since our problem formulation does not incur the need for a recurring input based on previous output and it would be a waste of resources to compute with it.

3.5 Model size

The size of our model was based both on our computing budget and the number of tokens we collected, as following the elucidations emerged from the *Chinchilla* paper (HOFFMANN et al., 2022) to train a model optimally based on the number of tokens at disposal and computing budget.

We also explored some hyperparameters for our architecture with a limited dataset, in a grid search fashion.

3.6 Model configuration and training

The amino acid sequences were encoded using a byte-pair encoding with around 28 vocabulary tokens to cover all amino acids, padding, and unknown tokens. The input sequence tokens were either padded or truncated according to the architecture input requirements.

The model performance in our experiments was optimized using cross-entropy loss, minimizing the divergence between predicted and true labels. Training followed the standard backpropagation and gradient descent. To mitigate overfitting, dropout regularisation was employed during training, as described in the transformer architecture for the base model (VASWANI et al., 2023) and used in other architectures (YANG et al., 2020).

3.7 Hyperparameter grid search

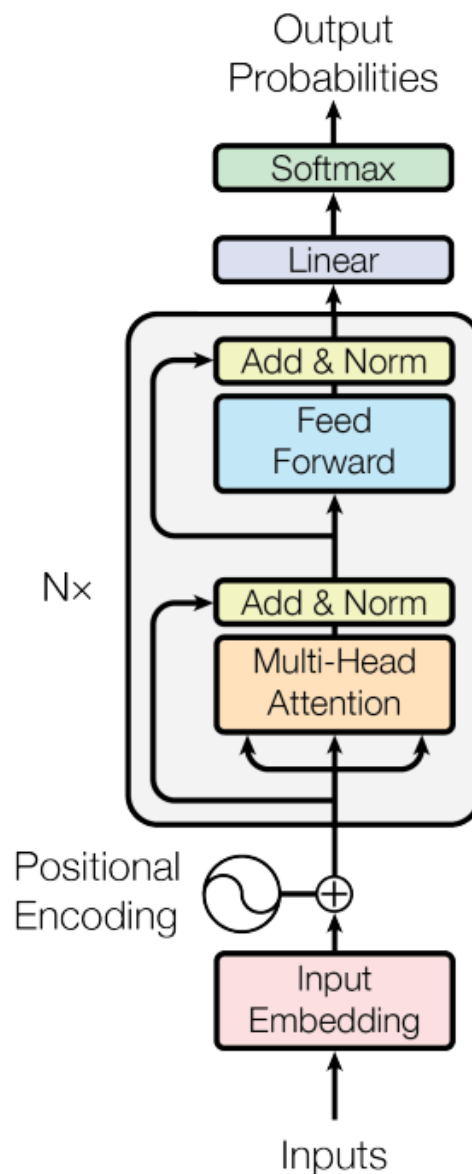
An initial grid search was conducted to identify our model's optimal hyperparameter configurations. Learning rate was either $1e-5$ or $1e-4$, Batch Size was either 32 or 64, Embedding Size was either 32 or 512, Dimension of the feed-forward network model (d_{hid}) was either 256 or 1024, Number of layers was either 8 or 10, Number of heads was either 8 or 16, and the maximum sequence length was either 128 or 512 amino-acids.

Given the size of our final database and the time needed to train for each model configuration, we divided our balanced dataset into three equal sizes of 1 percent of the total database for train, validation, and testing groups. A total of 3 epochs were used per configuration.

3.8 Best models training

After having done our hyperparameter's grid search, we selected the best models and made a new training procedure from scratch. The training procedure of these best models used 25% of the balanced dataset for training, another 25% for validation and another 25%

Figure 5 – Our architecture adapted from the original Transformer (VASWANI et al., 2023). Notice that we do not use the decoder part of the architecture, given that our problem formulation does not need to insert the outputs as input again.



Source: By the author (2025)

for testing, for 3 epochs.

3.9 Final model training

The final model training used 60% of the balanced dataset for training, another 10% for validation, and another 30% for testing, for 10 epochs. In another words, we used all of the balanced dataset and with more epochs than before, as opposed to our best models training.

3.10 Evaluation and Validation

To assess the model efficacy, we conducted cross-validation tests. Given that our model was tasked with only classification, its performance was evaluated using standard metrics, including accuracy, precision, recall, and F1-score as a final metric for easily comparable results.

For a more robust comparison and evaluation of our work, we trained some classical models (SGD Classifier, PassiveAgressive Classifier, Multinomial NB, and Gaussian NB). These methods were selected to establish a performance baseline, provide a simpler and more interpretable comparison, and demonstrate that our model's performance is not merely a result of overfitting or exploiting specific characteristics of the data. By comparing our model's performance to these established methods, we can gain valuable insights into its strengths and weaknesses, identify potential bottlenecks in our data or approach, and ultimately provide a more comprehensive evaluation of our work.

These methods were trained using 70% of our balanced dataset and 30% was used for testing. Batch size was 512 and the input length was set to 128 for all of them.

Some other methods could not be used, because these methods would require too many resources given the size of our dataset.

4 RESULTS

4.1 Classical methods

To compare our work and given our resources, we trained classical models on our data using the following architectures: SGD Classifier, PassiveAggressive Classifier, MultinomialNB, and GaussianNB. Our final F1 score for them is presented in Table 1. The training procedure used 70% of the balanced dataset for training and the remaining 30% for testing. Batch size was 512, and the maximum input sequence was 128 for 1 epoch.

Table 1 – Final F1 scores for the classical methods.

Model	F1 score
SGD Classifier	0.6855
PassiveAggressive Classifier	0.6648
Multinomial NB	0.6809
Gaussian NB	0.6920

Source: By the author (2025)

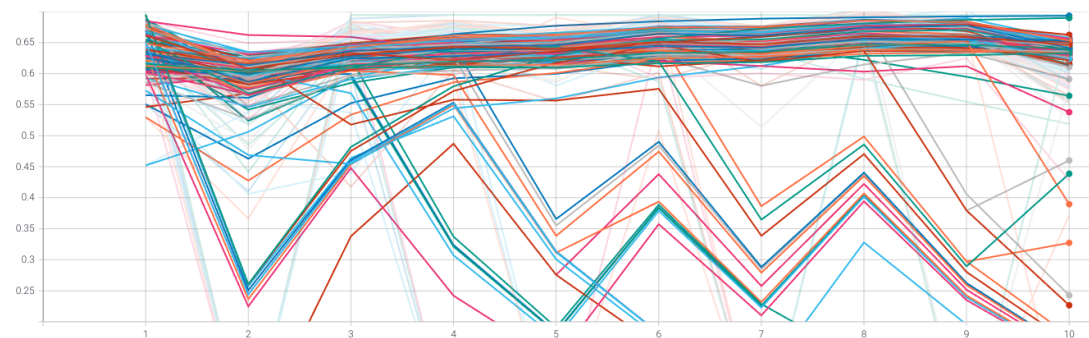
We also trained the BernoulliNB, but it couldn't classify our dataset.

4.2 Transformer

4.2.1 Exploring hyperparameters

The grid search yielded a total of 128 combinations of hyperparameters. The resulting graph of F1 (Figure 6), accuracy (Figure 7), and precision (Figure 8) was used to get an overall view of the capability of the model.

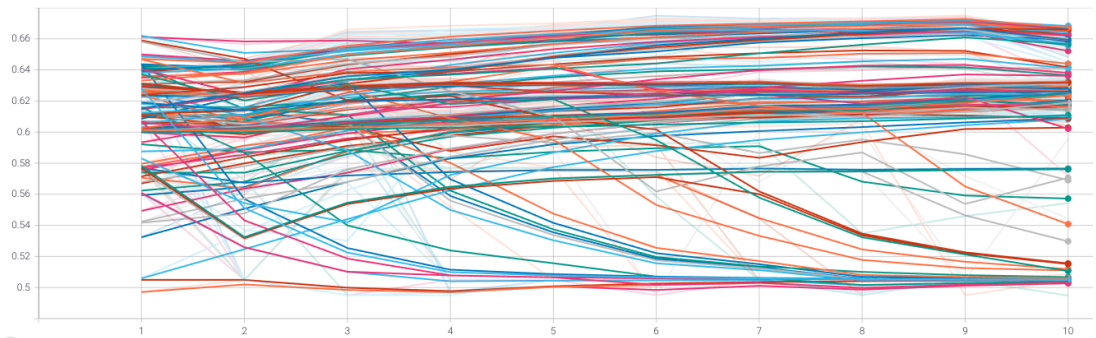
Figure 6 – F1 scores on the Y-axis against the interval of model evaluation.



Source: By the author (2025)

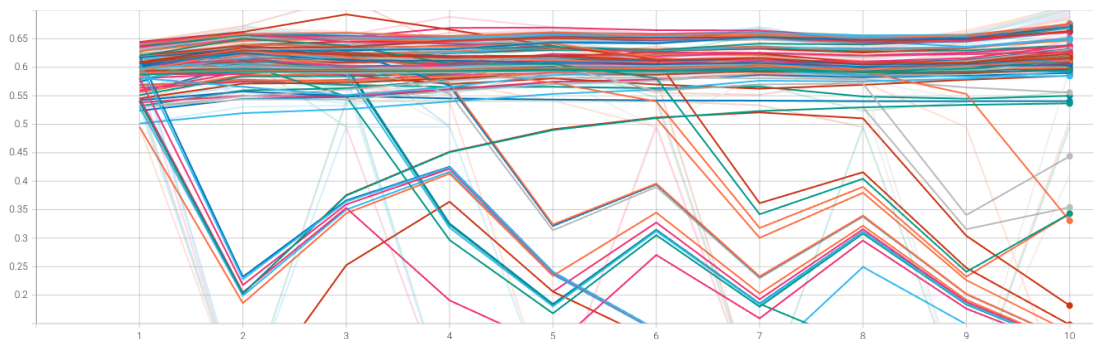
We then plotted the final F1 score against the wall time (Figure 9), zoomed in for better visualization (Figure 10) and then selected the models that would best serve us given the F1 score and wall time (Figure 11) by using the Pareto Frontier.

Figure 7 – Accuracy scores on the Y-axis against the interval of model evaluation.



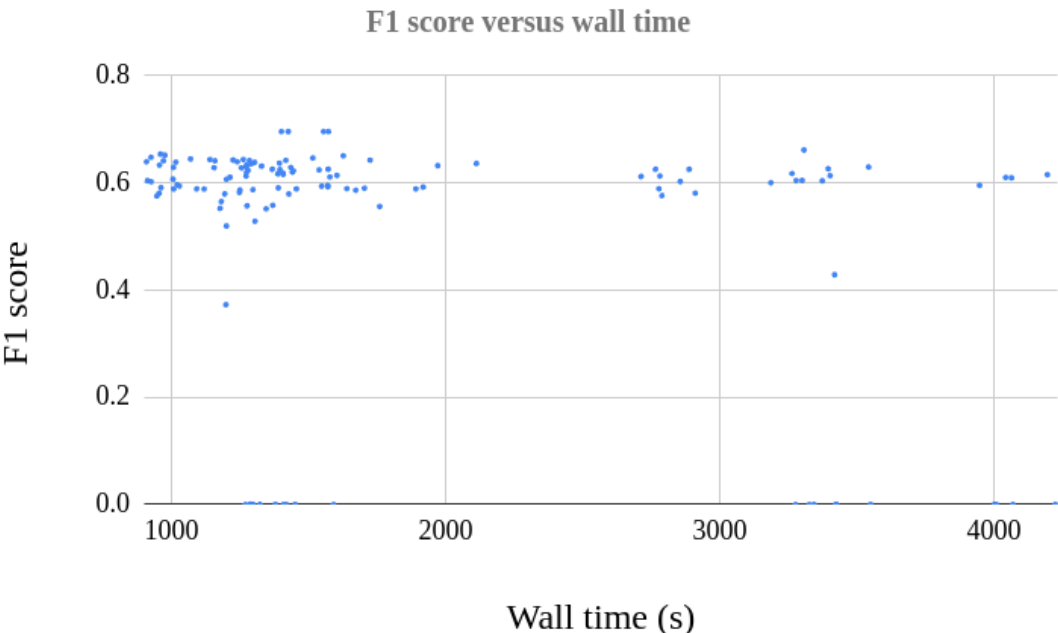
Source: By the author (2025)

Figure 8 – Precision scores on the Y-axis against the interval of model evaluation.



Source: By the author (2025)

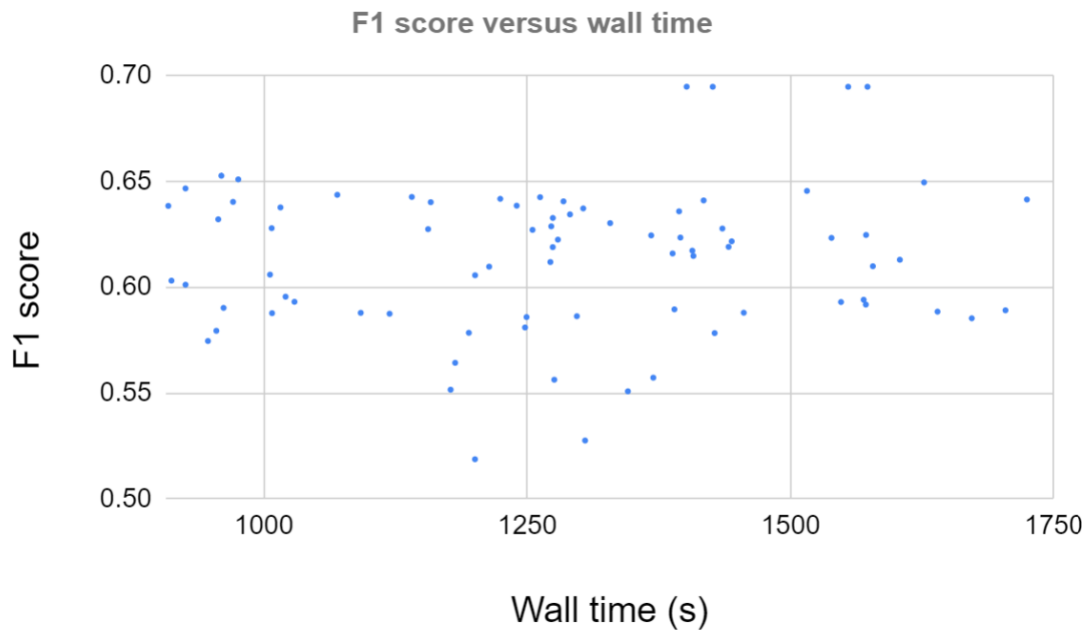
Figure 9 – F1 score versus the training wall time (s) for each model.



Source: By the author (2025)

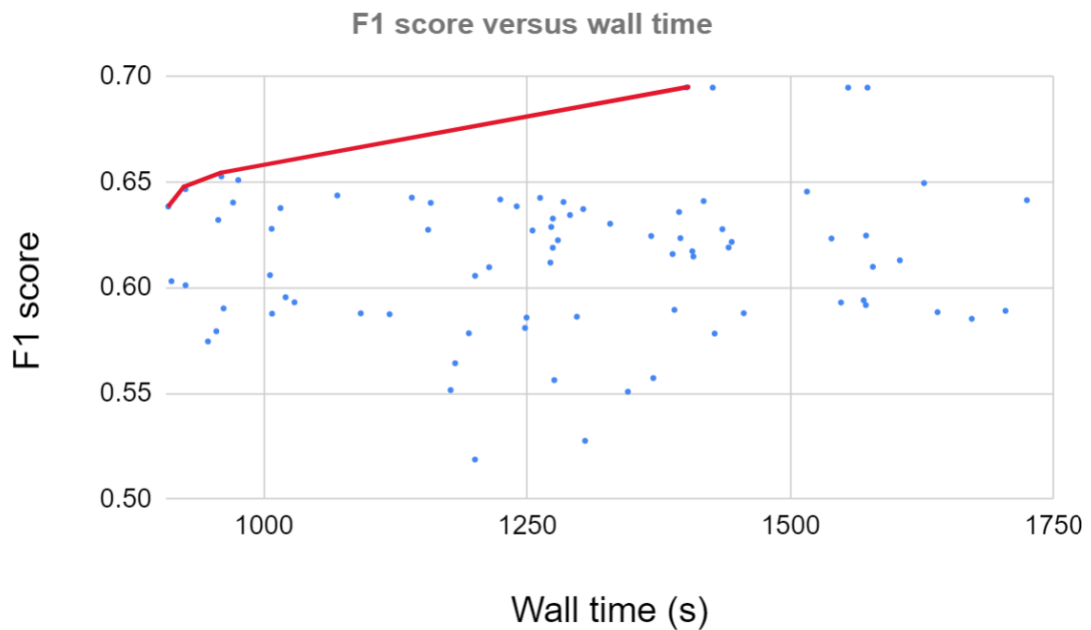
The best parameters for the models are shown in Table 2. The embedding size was 32 for all of them, while the maximum sequence length was 128.

Figure 10 – F1 score versus the training wall time (s) for each model zoomed in for better visualization (ignoring low-quality data).



Source: By the author (2025)

Figure 11 – F1 score versus the training wall time (s) for each model zoomed in, with the actual Pareto Frontier used. The best models, marked in red, were selected from this graph.



Source: By the author (2025)

4.2.2 Best models training

We obtained a final score for our best models as shown in Table 3.

Table 2 – Hyperparams for the best models found by Pareto Frontier of an exploratory search.

Model	Learning Rate	Batch size	Emb. size	D. Hid.	N. Layers	N. Head
1	1e-5	64	32	256	8	8
2	1e-4	64	32	256	8	8
3	1e-4	64	32	1024	8	16
4	1e-4	32	32	1024	10	16

Source: By the author (2025)

Table 3 – Final F1 score for the best models trained on splits of 25% of the balanced dataset, 3 epochs.

Model	F1 score	Wall time
1	0.6561	8h48m
2	0.660	8h32m
3	0.6611	8h42m
4	0.000	12h26m

Source: By the author (2025)

4.2.3 Final model

We also did a final model training by selecting the best model achieved in the last experiment: Model 3, F1 score of 0.6611.

We obtained 0.6575 as the best F1 score in validation during our training. The training graphs are shown in Figure 12 and 13.

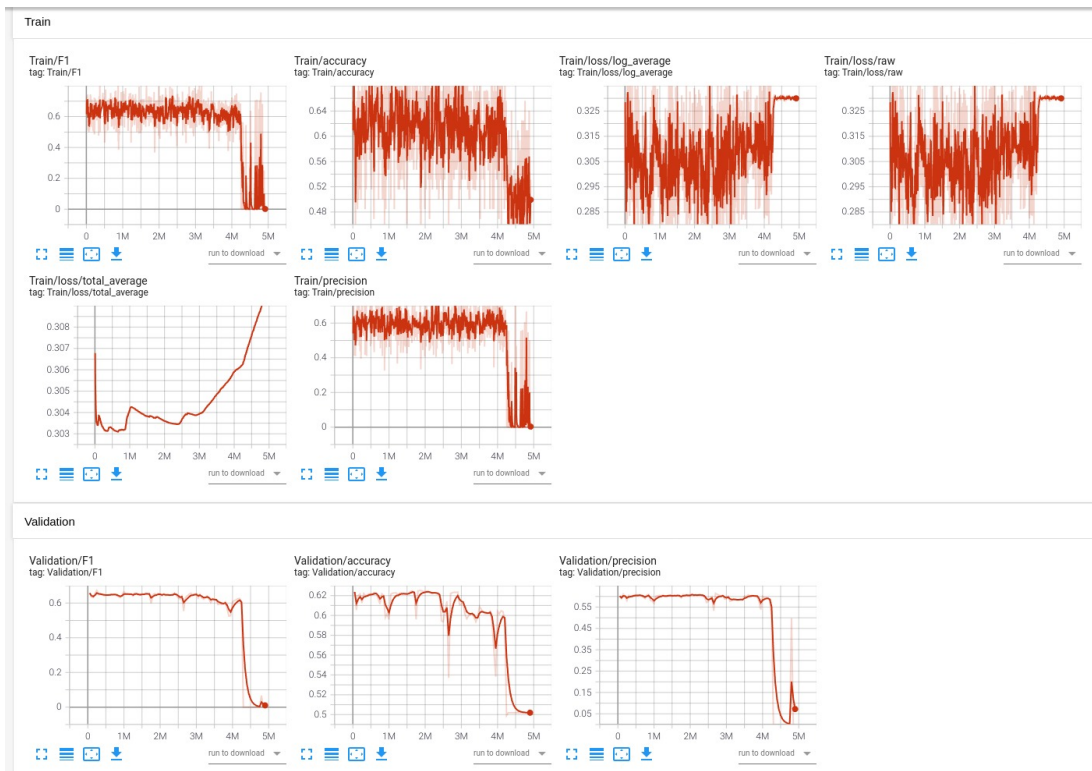
4.3 Discussion

Our model was able to achieve an approximate F1 score comparable with the classical methodologies (SGD Classifier, Multinomial NB, etc.). This results show that it is possible for an encoder-only Transformer architecture to classify amino-acid sequences into two major groups: non-enzymes and enzymes. We suspect, though, that only the amino-acid sequence is not enough information to fully describe the protein purpose or function.

An advantage that a Transformer based architecture has, is the capability of consuming larger bodies of input step-by-step, as it is made for textual sequence output. In our case though, it was not used, given the classifying nature of our problem, but by successfully treating the amino-acid sequence as an analogous NLP problem, we open up many opportunities of research, for different architectures, bigger models, different hyperparamters and so on.

Our results demonstrated that a Transformer based architecture could be used in the future for more complex tasks, like predicting the catalytic function of an amino-acid sequence, previously classified as an enzyme by, for example, our model, opening up vast possibilities for *in silicon* exploratory work of new processes and products.

Figure 12 – Training graph obtained by training from scratch our best Model 3.



Source: By the author (2025)

Figure 13 – The best F1 score obtained in this training was 0.6575, lower but close to, our other training. Note, however, that we trained with more data and more epochs.



Source: By the author (2025)

5 CONCLUSION

In this study, we successfully applied a Transformer-based architecture to classify amino acid sequences into two major groups: non-enzymes and enzymes. Our model achieved an F1 score comparable to traditional machine learning methods, demonstrating the effectiveness of Transformers for this task, although not superior.

While our model demonstrated promising results, we acknowledge that solely relying on amino acid sequences might not be sufficient for fully characterizing protein function. Future research could explore incorporating additional features such as protein structure, evolutionary information, or external data sources to enhance prediction accuracy.

The ability of Transformers to process sequential data makes them well-suited for tasks involving amino acid sequences. This opens up opportunities for exploring more complex tasks, such as predicting catalytic function or other protein properties.

In conclusion, our findings suggest that Transformer-based architectures hold great potential for advancing the field of protein classification and enabling novel *in silico* studies.

BIBLIOGRAPHY

- BARRAULT, Loïc et al. Seamless4t-massively multilingual & multimodal machine translation. **arXiv preprint arXiv:2308.11596**, 2023. pages 22
- BENSON, Dennis A et al. Genbank. **Nucleic acids research**, Oxford University Press, v. 41, n. D1, p. D36–D42, 2012. pages 17
- BLOCK, H. D. The perceptron: A model for brain functioning. i. **Rev. Mod. Phys.**, American Physical Society, v. 34, p. 123–135, Jan 1962. Disponível em: <<https://link.aps.org/doi/10.1103/RevModPhys.34.123>>. pages 18, 19
- BOTTOU, Léon. **On-line learning in neural networks**. Cambridge: Cambridge University Press, 1998. pages 18
- BRANDES, Nadav et al. ProteinBERT: a universal deep-learning model of protein sequence and function. **Bioinformatics**, v. 38, n. 8, p. 2102–2110, 02 2022. ISSN 1367-4803. Disponível em: <<https://doi.org/10.1093/bioinformatics/btac020>>. pages 10, 26
- CAMACHO, Cezanne. **Convolutional Neural Networks**. 2018. Disponível em: <https://cezannec.github.io/Convolutional_Neural_Networks/>. pages 19
- CHEN, Tyrone et al. genomicbert and data-free deep-learning model evaluation. **bioRxiv**, Cold Spring Harbor Laboratory, p. 2023–05, 2023. pages 10, 25
- CHOI, Kristy et al. Encoding musical style with transformer autoencoders. In: III, Hal Daumé; SINGH, Aarti (Ed.). **Proceedings of the 37th International Conference on Machine Learning**. PMLR, 2020. (Proceedings of Machine Learning Research, v. 119), p. 1899–1908. Disponível em: <<https://proceedings.mlr.press/v119/choi20b.html>>. pages 22
- CONSORTIUM, The UniProt. UniProt: the Universal Protein Knowledgebase in 2023. **Nucleic Acids Research**, v. 51, n. D1, p. D523–D531, 11 2022. ISSN 0305-1048. Disponível em: <<https://doi.org/10.1093/nar/gkac1052>>. pages 11, 27
- COPELAND, Robert A. **Enzymes: a practical introduction to structure, mechanism, and data analysis**. [S.l.]: John Wiley & Sons, 2023. pages 10
- CRAMMER, Koby et al. Online passive-aggressive algorithms. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2006. pages 18
- CUNNINGHAM, Fiona et al. Ensembl 2022. **Nucleic acids research**, Oxford University Press, v. 50, n. D1, p. D988–D995, 2022. pages 17
- DEVLIN, Jacob et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 2019. pages 24, 25, 28
- DILL, Ken A.; MACCALLUM, Justin L. The protein-folding problem, 50 years on. **Science**, v. 338, n. 6110, p. 1042–1046, 2012. Disponível em: <<https://www.science.org/doi/abs/10.1126/science.1219021>>. pages 24
- DOSOVITSKIY, Alexey et al. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. 2021. pages 10, 22

- DRESZER, Timothy R et al. The ucsc genome browser database: extensions and updates 2011. **Nucleic acids research**, Oxford University Press, v. 40, n. D1, p. D918–D923, 2012. pages 17
- FRIEDMAN, Jerome H. A comparison of bayesian classifiers. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 39, n. 1, p. 3–28, 1977. pages 18
- HOFFMANN, Jordan et al. **Training Compute-Optimal Large Language Models**. 2022. pages 28
- JONES, David T; THORNTON, Janet M. The impact of alphafold2 one year on. **Nature methods**, Nature Publishing Group US New York, v. 19, n. 1, p. 15–20, 2022. pages 24
- JUMPER, John et al. Highly accurate protein structure prediction with alphafold. **Nature**, Nature Publishing Group, v. 596, n. 7873, p. 583–589, 2021. pages 10, 15, 17, 24
- KREUK, Felix et al. **AudioGen: Textually Guided Audio Generation**. 2023. pages 10, 22
- LAKEW, Surafel M.; CETTOLO, Mauro; FEDERICO, Marcello. **A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation**. 2018. pages 10, 22
- LE, Nguyen Quoc Khanh et al. Bert-promoter: An improved sequence-based predictor of dna promoter using bert pre-trained model and shap feature selection. **Computational Biology and Chemistry**, Elsevier, v. 99, p. 107732, 2022. pages 10, 25
- Authors' preface. In: PALMER, Trevor; BONNER, Philip L. (Ed.). **Enzymes (Second Edition)**. Second edition. Woodhead Publishing, 2011. p. xiv–xv. ISBN 978-1-904275-27-5. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B978190427527500246>>. pages 10, 12, 13, 14, 15, 23
- RAFFEL, Colin et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **CoRR**, abs/1910.10683, 2019. Disponível em: <<http://arxiv.org/abs/1910.10683>>. pages 25
- RASCHKA, Sebastian. **Python machine learning**. [S.l.]: Packt publishing ltd, 2015. pages 27
- RENNIE, Jason D. J. et al. Tackling the poor assumptions of naive bayes text classifiers. **ACM Transactions on Information Systems (TOIS)**, ACM, v. 22, n. 3, p. 3–16, 2004. pages 18
- ROBERTSON, Angus J. et al. Concordance of x-ray and alphafold2 models of sars-cov-2 main protease with residual dipolar couplings measured in solution. **Journal of the American Chemical Society**, v. 143, n. 46, p. 19306–19310, 2021. PMID: 34757725. Disponível em: <<https://doi.org/10.1021/jacs.1c10588>>. pages 24
- RUST, Alistair G.; MONGIN, Emmanuel; BIRNEY, Ewan. Genome annotation techniques: new approaches and challenges. **Drug Discovery Today**, v. 7, n. 11, p. S70–S76, 2002. ISSN 1359-6446. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1359644602022894>>. pages 10, 16, 17
- SCHUSTER, Mike; PALIWAL, Kuldip K. Bidirectional recurrent neural networks. **IEEE transactions on Signal Processing**, IEEE, v. 45, n. 11, p. 2673–2681, 1997. pages 20

- SHU, Chang et al. **SideRT: A Real-time Pure Transformer Architecture for Single Image Depth Estimation**. 2022. pages 10, 22
- SOH, Jung; GORDON, Paul MK; SENSEN, Christoph W. **Genome annotation**. [S.I.]: CRC Press, 2012. pages 10, 15, 16
- SONG, Yang et al. **Consistency Models**. 2023. pages 21
- TOUVRON, Hugo et al. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 2023. pages 10, 22
- TSOMPANA, Maria; BUCK, Michael J. Chromatin accessibility: a window into the genome. **Epigenetics & chromatin**, BioMed Central, v. 7, n. 1, p. 1–16, 2014. pages 17
- VASWANI, Ashish et al. **Attention Is All You Need**. 2023. pages 10, 11, 22, 23, 28, 29
- WANG, Zilong; WAN, Zhaohong; WAN, Xiaojun. Transmodality: An end2end fusion method with transformer for multimodal sentiment analysis. In: **Proceedings of The Web Conference 2020**. New York, NY, USA: Association for Computing Machinery, 2020. (WWW '20), p. 2514–2520. ISBN 9781450370233. Disponível em: <<https://doi.org/10.1145/3366423.3380000>>. pages 10, 22
- YANG, Zhilin et al. **XLNet: Generalized Autoregressive Pretraining for Language Understanding**. 2020. pages 28
- ZHIQIANG, Wang; JUN, Liu. A review of object detection based on convolutional neural network. In: IEEE. **2017 36th Chinese control conference (CCC)**. [S.I.], 2017. p. 11104–11109. pages 10, 19

Annex

ANNEX A – EXAMPLE OF A FASTA FILE

```

>sp|P09848|LPH_HUMAN Lactase/phlorizin hydrolase OS=Homo sapiens OX=9606
GN=LCT PE=1 SV=3
MELSWHVVFIALLSFSCWGS DWESDRNFISTAGPLTNDLLHNL SGLLDQSSNFVAGDKD
MYVCHQPLPTFLPEYFSSLHASQITHYKVFLSWAQLLPAGSTQNPDEKTVQCYRRLKAL
KTARLQPMVILHHQTL PASTLRRTEAFADLFADYATFAFHSFGDLVGIWFTFSDLEEVIK
ELPHQESRASQLQTLSDAHRKAYEIYHESYAFQGGKLSVVLRAEDIPELLLEPPISALAQ
DTVDFLSLDLSYECQNEASLRQKLSKLQTIEPKVKVFIFNLKLPDCPSTMKNPASLLFSL
FEINKDQVLTIGFDINEFLSCSSSSSKSMSCSLTGSLALQPDQQQDHETDSSPASAYQ
RIWEAFANQSRAERDAFLQDTFPEGFLWGASTGAFNVEGGWAEGGRGVSIWDP RRPLNTT
EGQATLEVASDSYHKVASDVALLCGLRAQVYKFSISWSRIFPMGHGSSPSLPGVAYYNKL
IDRLQDAGIEPMATLFHWDLPQALQDHGGWQNESVVDAFLDYAAFCFSTFGDRVKLWVTF
HEPWVMSYAGYGTGQHPPGISDPGVASFKAHLVLKAHARTWHHYN SHHRPQQQGHVGIV
LNSDWAEP LSPERPEDLRASERFLHFMLGWFAHPVFVDGDYPATLRTQIQQMNRQC SHPV
AQLPEFTEAEKQLLKGSADFLGLSHYTSRLISNAPQNTCIPSYDTIGGFSQHVNHVWPQT
SSSWIRVVPWGIRRLQFVSLEYTRGKVPIYLAGNGMPIGESENLFDDSLRVDYFNQYIN
EVLKAIKEDSVDVRSYIARSLIDGFEGPSGYSQRFG LHHVNFSDSSKSRTPRKSAYFFTS
IIEKNGFLT KGAKRLLPPNTVNLPSKVRAFTFPSEVPSKAKVVWEKFSSQPKFERDLFYH
GTRDDDFLWGVSSSAYQIEGAWDADGKGPSIWDNFTHTPGSNVKDNATGDIACDSYHQLD
ADLNMLRALKVKAYRFSISWSRIFPTGRNSSINSHGV DYYNRLINGLVASNIFPMVTLFH
WDL PQALQDIGGWENPALIDLFDSYADFCFQTFGDRV KFWMTFNEMMYLAWLGYGSGEFP
PGVKDPGWAPYRIAHAVIKAHARVYHTYDEKYRQE QKGVISLSLSTHWAEPKSPGVPRDV
EAADRMLQFSLGWFAHPIFRNGDYPDTMKWKVGNRSELQHLATSRLPSFTEEEKR FIRAT
ADVFC LNTYYSRIVQHKT PRLNPPSYEDDQEMAEEDPSWPSTAMNRAAPWGTRRLLNWI
KEEYGDIP IYITENG VGLTNPNTEDTD RIFYHKTYINEAL KAYRLDGIDLRGYVAWSLMD
NFEWLN GYTVKFGLYHVD FNNTNRPR TARASARYYTEVITNNGMPLAREDEF LYGRFPEG
FIWSAASAAYQIEGAWRADGKGLSIWDTFSHTPLRVENDAIGDVACDSYHKIAEDLVTLQ
NLGVSHYRFSISWSRILPDGTTRYINEAGLNYVRLIDTLLAASIQQV TIYHWDLPQTL
QDVGGWENETIVQRFKEYADVLFQRLGDKVKFWITLNEPFVIAYQGYGYGTAAPGVSNRP
GTAPYIVGHNL IKAHAEA WHLYNDVYRASQGGVISITISSDWAEP RDP SNQEDVEAARRY
VQFMGGWFAHPIFKNGDYNEVMKTRIRD RSLAAGLNKSRLPEFTESEKRRINGTYDFFGF
NHYT TVLAYNLNYATAISSFDADRGVASIADRSWPD SGFWLKMTPFGFRRILNWLKEEY
NDPPIYVTENGVSQREETDLNDTARIYYLR TYINEAL KAVQDKVDLRGYTVWSAMD NFEW
ATGFSERFGLHFVNYS DPSLPRIPKASAKFYASVVR CNGFDPATGPHACLHQPDAGPTI
SPVRQEEVQFLGLMLGTTEAQTALYVLFSLVLLGVCGLAFLSYKYCKRSKQGKTQRSQQE
LSPVSSF

```