الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

University of Constantine 1 Mentouri Brothers
Faculty of Natural and Life Sciences

جامعة قسنطينة 1 الإخوة منتوري
كلية علوم الطبيعة والحياة

Department: Applied Biology

قسم: البيولوجيا التطبيقية

Dissertation presented with a view to obtaining the Master's Diploma

Domaine: Natural and Life Sciences

Sector: Biotechnology

specialty: Bioinformatics

Order number:
Series number:

Title:

Estimating Biological Sequence Similarity Using Artificial Intelligence

Presented by: Mahdjoub Mohamed **10/06/2024**
Boucetta Mohamed El Amine

Evaluation jury:

President:	Prof. BELLIL Ines	(Prof – University of Constantine 1 Frères Mentouri).
Supervisor:	Dr. DAAS Mohamed Skander	(MC(A) University of Constantine 1 Frères Mentouri).
Examiner:	Dr. BOUCHEHAM Anouar	(MC(A) University of Constantine 3 Boubenider).

Academic year 2023 - 2024

Acknowledgments

First of all, we are deeply grateful to Allah for His guidance, blessings, and strength that enabled me to complete this research. Without His grace, this work would not have been possible.

We would like to express our deep gratitude to our supervisor, Dr. Daas Mohamed Skander, for his invaluable guidance, encouragement, and support in this research. His expertise and insights were crucial in shaping this work.

We would also like to thank our colleagues and friends at the Bioinformatics Department of Constantine I University for their cooperation and support. Their friendship made this trip enjoyable and memorable.

We would like to express our sincere gratitude to our family for their unconditional love and support. Their encouragement and understanding were the root of our success.

Dedication

This work is dedicated to our parents, who have always believed in us, supported us and encouraged us. Their sacrifices and encouragement are the foundation of all our success.

Abstract

Understanding protein sequence similarities is crucial in bioinformatics as it illuminates functional and evolutionary relationships between proteins. By comparing sequences, researchers can identify conserved structural and functional elements, aiding in protein function annotation, interaction prediction, and drug target identification. Accurate prediction of protein sequence similarity is vital for biological research, facilitating homolog identification, evolutionary biology studies, and protein interaction network mapping. Deep learning has transformed bioinformatics by offering sophisticated methods for analyzing complex biological data, with models like CNNs excelling in tasks such as protein structure prediction and sequence alignment. This study aims to develop and evaluate a deep learning model for predicting protein sequence similarity, encompassing data preprocessing, model development, training, and validation. The objective is to enhance the accuracy and efficiency of protein sequence similarity predictions. Despite the complexity and diversity of sequences, the model demonstrated good prediction precision, addressing key challenges in the field and advancing bioinformatics research.

Keywords: protein sequence similarity, sequence alignment, deep learning, FASTA, prediction

Résumé

Comprendre les similitudes des séquences protéiques est crucial en bioinformatique car cela éclaire les relations fonctionnelles et évolutives entre les protéines. En comparant les séquences, les chercheurs peuvent identifier des éléments structuraux et fonctionnels conservés, aidant à l'annotation des fonctions protéiques, à la prédiction des interactions et à l'identification des cibles médicamenteuses. La prédiction précise de la similitude des séquences protéiques est essentielle pour la recherche biologique, facilitant l'identification des homologues, les études de biologie évolutive et la cartographie des réseaux d'interaction des protéines. L'apprentissage profond a transformé la bioinformatique en offrant des méthodes sophistiquées pour analyser des données biologiques complexes, avec des modèles comme les CNN excellant dans des tâches telles que la prédiction de la structure des protéines et l'alignement des séquences. Cette étude vise à développer et évaluer un modèle d'apprentissage profond pour prédire la similitude des séquences protéiques, englobant le prétraitement des données, le développement du modèle, l'entraînement et la validation. L'objectif est d'améliorer la précision et l'efficacité des prédictions de similitude des séquences protéiques. Malgré la complexité et la diversité des séquences, le modèle a démontré une bonne précision de prédiction, répondant aux principaux défis du domaine et faisant progresser la recherche en bioinformatique.

Mots-clés: similitudes des séquences protéiques, alignement de séquences, deep learning, FASTA, prédiction

ملخص

فهم تشابهات تسلسل البروتينات أمر بالغ الأهمية في المعلوماتية الحيوية حيث يوضح العلاقات الوظيفية والتطورية بين البروتينات. من خلال مقارنة التسلسلات، يمكن للباحثين تحديد العناصر الهيكلية والوظيفية المحفوظة، مما يساعد في توصيف وظائف البروتين، وتوقع التفاعلات، وتحديد الأهداف الدوائية. التنبؤ الدقيق بتشابه تسلسل البروتينات ضروري للبحث البيولوجي، حيث يسهل تحديد النظائر، ودراسة علم الأحياء التطوري، ورسم شبكات تفاعل البروتينات. لقد أحدث التعلم العميق ثورة في المعلوماتية الحيوية من خلال تقديم أساليب متقدمة لتحليل البيانات البيولوجية المعقدة، حيث تبرز نماذج مثل الشبكات العصبية التلافيفية (CNN) في مهام مثل التنبؤ بهيكل البروتين ومواءمة التسلسلات. تهدف هذه الدراسة إلى تطوير وتقييم نموذج للتعلم العميق للتنبؤ بتشابه تسلسل البروتينات، ويشمل ذلك معالجة البيانات الأولية، وتطوير النموذج، والتدريب، والتحقق. الهدف هو تحسين دقة وكفاءة التنبؤات بتشابه تسلسل البروتينات. على الرغم من تعقيد وتنوع التسلسلات، أظهر النموذج دقة جيدة في التنبؤ، مما يعالج التحديات الرئيسية في المجال ويعزز البحث في المعلوماتية الحيوية.

الكلمات المفتاحية: تشابه تسلسل البروتين ,محاذاة التسلسلات ,التعلم العميق , FASTA , تنبؤ

Table of Contents

List of Figures

List of Tables

List of abbreviations

AI - Artificial Intelligence

BLAST - Basic Local Alignment Search Tool

CNN - Convolutional Neural Network

FASTA - Fast-All (a DNA and protein sequence alignment software package)

ML - Machine Learning

RNN - Recurrent Neural Network

GA - Genetic Algorithms

HPC - High-Performance Computing

PCJ - Parallel Computing in Java

MSA - Multiple Sequence Alignment

UniProt - Universal Protein Resource

PDB - Protein Data Bank

Pfam - Protein families database

MAPE - Mean Absolute Percentage Error

TRUST-RNN - TRUST Recurrent Neural Network

Fourier-RNN - Fourier Recurrent Neural Network

NLP - Natural Language Processing

SVM - Support Vector Machines

PCA - Principal Component Analysis

Main Introduction

Understanding protein sequence similarities is of great importance in bioinformatics as it helps elucidate functional and evolutionary relationships between proteins. By comparing sequences, researchers can identify structural and functional elements that are conserved across species, which can help annotate protein function, predict interactions, and identify drug targets. Accurate prediction of protein sequence similarity is of great importance for biological research. These allow scientists to make informed inferences about protein function and relationships, facilitating the identification of homologs, studying evolutionary biology, and mapping protein interaction networks. These predictions are important for understanding disease mechanisms, developing treatments, and conducting comparative genomics. Deep learning has revolutionized bioinformatics by providing advanced methods for analyzing complex biological data. Models such as CNNs and RNNs are used for tasks such as protein structure prediction, gene expression analysis, and sequence alignment. These models can autonomously learn hierarchical features from raw data, making them ideal for capturing complex patterns in protein sequences. The purpose of this study is to develop and evaluate a deep learning model for predicting protein sequence similarity. This involves designing a robust model, training it on a large dataset, and comparing its performance to traditional methods. This study includes data preprocessing, model development, training, validation, and performance evaluation. The goal is to use deep learning to improve the accuracy and efficiency of protein sequence similarity prediction, thereby advancing the field of bioinformatics. Despite advances in deep learning, predicting protein sequence similarity remains a challenge due to the complexity and diversity of protein sequences. Traditional methods often struggle with large datasets and subtle sequence differences. This study addresses the following important questions:

How can deep learning models be optimized to improve the accuracy and reliability of protein sequence similarity predictions, surpassing the limitations of current methods?

CHAPTER 1: Comparative Analysis of Protein Sequence Evolution

1.1 Introduction

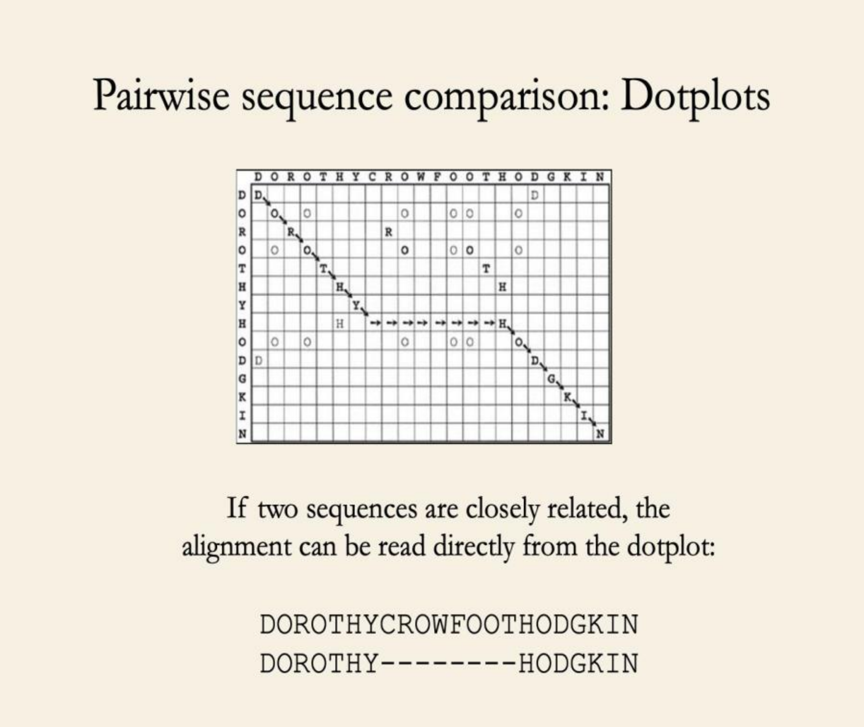
Proteins are fundamental molecules that perform many functions in organisms. Understanding the similarities and differences between protein sequences is important for many different fields, including evolutionary biology, functional genomics, and drug discovery. Comparing protein sequences allows scientists to deduce functional similarities, evolutionary relationships, and structural features. This chapter reviews the methods and tools used to compare protein sequences, highlighting their applications and importance in modern biological research.

1.2 Methods for Comparative Analysis of Protein Sequences

Protein sequence comparison methods play a crucial role in bioinformatics research. Various approaches have been proposed in the literature. One method involves utilizing numerical representations of protein sequences based on physical and chemical properties of amino acids, followed by fast Fourier transform and spectral analysis [1]. Another method focuses on encoding sequence data and physicochemical properties of amino acids into vectors, allowing for parallel and fast implementation by partitioning long protein sequences into fixed-length blocks [2]. Additionally, the use of discrete wavelet transform and fractal dimension analysis has been suggested for protein sequence similarity assessment, providing a comprehensive and reliable analysis of protein sequences [3]. Furthermore, the exploration of existing methods like BLAST, HHblits, and CD-HIT for comparing low complexity regions highlights the need for specialized approaches tailored to efficiently compare such regions [4].

Pairwise Sequence Alignment

Pairwise sequence alignment is a fundamental process in bioinformatics, crucial for comparing DNA or amino acid sequences to determine similarities, evolutionary relationships, and functional implications [5][6][7]. It involves aligning two sequences to establish residue correspondence and identify common patterns, aiding in inferring protein functions and evolutionary histories [8][9]. Various alignment algorithms exist, such as the Needleman-Wunsch algorithm for finding alignments with minimum editing weight. Pairwise sequence alignment is essential for database similarity searches, multiple sequence alignment, and guiding laboratory procedures in protein investigations. Additionally, specialized algorithms have been developed for aligning sequences with repetitive motifs, like zinc finger proteins, using hidden Markov models to differentiate conserved and variable motifs during alignment.



Pairwise sequence comparison: Dotplots

D O R O T H Y C R O W F O O T H O D G K I N
D O R O T H Y H O D G K I N

If two sequences are closely related, the alignment can be read directly from the dotplot:

```
DOROTHYCROWFOOTHODGKIN
DOROTHY--------HODGKIN
```

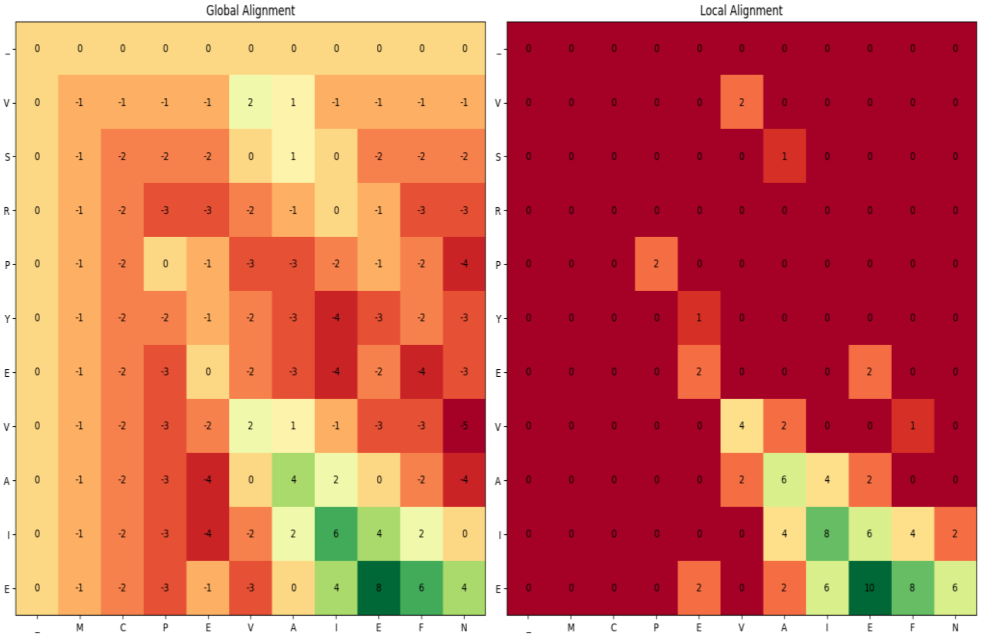
Figure 1 *A visual example of a pairwise sequence alignment highlighting matching and mismatching regions between two protein sequences. This can help readers understand the concept of sequence alignment.*

Global Alignment

Global alignment refers to the process of aligning entities or sequences on a larger scale, considering overall features or structures rather than just local characteristics. In computational biology, the A*PA aligner utilizes global alignment techniques to efficiently align long sequences with high divergence, achieving significant speedups compared to existing aligners [10]. Similarly, in knowledge fusion tasks, the GALA model introduces global features for aligning entities from different knowledge graphs, emphasizing the matching of global characteristics to merge the graphs effectively [11]. Moreover, in the context of semantic segmentation for image processing, a joint global-local alignment approach has been proposed to align data distributions between source and target domains more effectively, improving segmentation results by considering both global and local features [12]. These examples highlight the importance and effectiveness of global alignment strategies across various domains, showcasing their utility in diverse applications.

Local Alignment

Local Alignment refers to a method used in various fields like network comparison, image-text matching, and self-organizing particle systems to find similarities in specific regions or components within complex structures. In network alignment, Local Network Alignment (LNA) focuses on identifying local regions of similarity between networks, often using seed nodes and network embedding techniques to improve alignment effectiveness [13][14]. In image-text matching, local alignment methods utilize fine-grained features to explore correspondence between image regions and text words, with recent advancements incorporating global semantic consistency for more robust matching results [15][16]. Additionally, in self-organizing particle systems, local distributed stochastic algorithms are employed to align particles along dominant directions or maintain non-alignment based on specific parameters, resembling solid or gaseous states, respectively [17].



Global Alignment

	_	M	C	P	E	V	A	I	E	F	N
_	0	0	0	0	0	0	0	0	0	0	0
V	0	-1	-1	-1	-1	2	1	-1	-1	-1	-1
S	0	-1	-2	-2	-2	0	1	0	-2	-2	-2
R	0	-1	-2	-3	-3	-2	-1	0	-1	-3	-3
P	0	-1	-2	0	-1	-3	-3	-2	-1	-2	-4
Y	0	-1	-2	-2	-1	-2	-3	-4	-3	-2	-3
E	0	-1	-2	-3	0	-2	-3	-4	-2	-4	-3
V	0	-1	-2	-3	-2	2	1	-1	-3	-3	-5
A	0	-1	-2	-3	-4	0	4	2	0	-2	-4
I	0	-1	-2	-3	-4	-2	2	6	4	2	0
E	0	-1	-2	-3	-1	-3	0	4	8	6	4

Local Alignment

	_	M	C	P	E	V	A	I	E	F	N
_	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	2	0	0	0	0	0
S	0	0	0	0	0	0	1	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	2	0	0	0	0	0	0	0
Y	0	0	0	0	1	0	0	0	0	0	0
E	0	0	0	0	2	0	0	0	2	0	0
V	0	0	0	0	0	4	2	0	0	1	0
A	0	0	0	0	0	2	6	4	2	0	0
I	0	0	0	0	0	0	4	8	6	4	2
E	0	0	0	0	2	0	2	6	10	8	6

Figure 2 *A comparative illustration showing the differences between global and local alignment. This can clarify the distinct approaches used in sequence comparison.*

Multiple Sequence Alignment

Multiple Sequence Alignment (MSA) is a fundamental tool in bioinformatics used to align three or more biological sequences, such as DNA, RNA, or proteins, to identify evolutionary relationships and common patterns [18]. MSA plays a crucial role in various biological procedures like protein structure prediction, phylogenetic inference, and sequence analysis [19]. Traditional MSA algorithms rely on dynamic and heuristic approaches to handle the computational complexity of aligning multiple sequences [20]. The use of external information from databases or user knowledge has been shown to enhance the quality of alignments by adding new constraints and making the alignments biologically more meaningful [21]. Genetic Algorithms (GA) have been developed to address the computational challenges of MSA, providing approximate solutions efficiently, especially for large datasets like human DNA and protein sequences [22].

1.3 Tools and Databases for Protein Sequence Comparison

BLAST

The Basic Local Alignment Search Tool (BLAST) is a widely utilized program in molecular microbiology research, offering functions such as identifying sequences, efficiently finding target DNA, inferring gene functions and protein structure domains, and designing primers [23]. BLAST typically employs a seed and grow strategy for sequence alignment, but an alternative approach involves identifying high-density seed match regions in database sequences and performing Smith-Waterman local alignments, offering advantages for specific use cases [24]. As genetic data volumes increase, high-performance computing solutions like the Parallel Computing in Java (PCJ) library are crucial for executing BLAST on HPC clusters and supercomputers in a massively parallel manner, significantly reducing analysis time and enhancing scalability [25]. For users requiring high-throughput comparative genomic pipelines, installing BLAST locally on standalone workstations or compute clusters under various operating systems is recommended, along with strategies for database management and sequence analysis efficiency [26]. BLAST is also instrumental in studying sequence alignments of monoclonal antibodies, revealing homology percentages and specific amino acid regions in light and heavy antibody chains, aiding in understanding sequence variations and exchanges between antibodies [27].

FASTA

The FASTA algorithm is another sequence **comparison method,** known for its speed and efficiency. It uses a heuristic **method** to identify regions of similarity between sequences (Pearson & Lipman, 1988).

Databases

Table 1: *Protein Sequence Database List*

UniProt	A comprehensive resource for protein sequence and functional information (The UniProt Consortium, 2019).
PDB	The Protein Data Bank, which provides structural data for proteins and nucleic acids (Berman et al., 2000).
Pfam	A database of protein families, each represented by multiple sequence alignments and hidden Markov models (Finn et al., 2016).

Chapter 2: Artificial Intelligence and Machine Learning

2.1 Introduction

Artificial intelligence AI is a field of computer science that seeks to create machines with the capacity for carrying out tasks, which are typically performed by humans. This encompasses abilities such as reasoning, learning, problem-solving, perception, and language comprehension.

2.2 Machine Learning

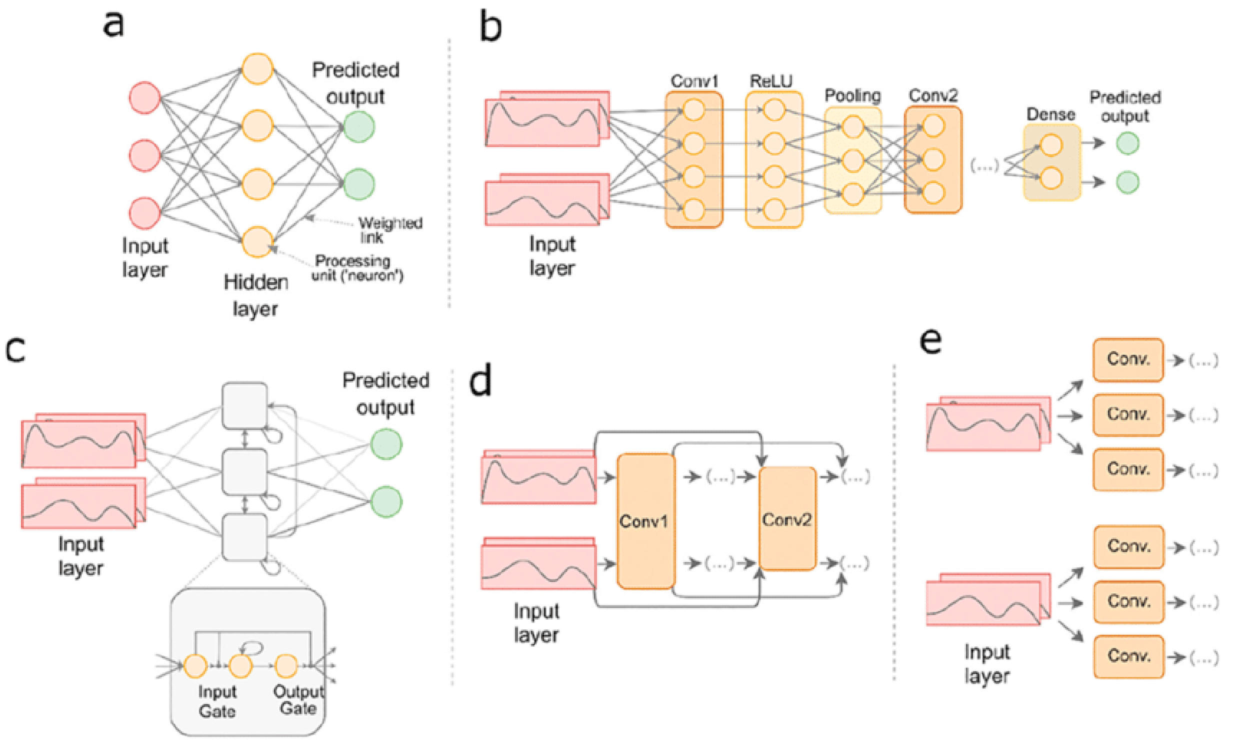


***Figure 3** A schematic of a deep learning neural network used in bioinformatics, showing layers such as convolutional and recurrent layers. This helps explain the complexity and structure of AI models.*

A branch of AI trained on statistical models and algorithms, which enable it to make predictions and decisions. Machine learning algorithms may improve and adapt over time, enhancing their capabilities through the use of training data and historical information. In order to continue improving its results, machine learning relies on human engineers for input of relevant and preprocessed data. It is skilled at solving complex problems and generating meaningful insights by recognizing patterns within data.

2.3. Different Types of Machine Learning

Machine learning encompasses various paradigms, including supervised learning, unsupervised learning, reinforcement learning, multi-label learning, semi-supervised learning, one-class classification, positive-unlabeled learning, transfer learning, multi-task learning, and one-shot learning [28]. These paradigms have been developed over the last seven decades and are utilized across diverse application domains. Supervised learning involves training a model on labeled data to make predictions, while unsupervised learning deals with uncovering patterns in unlabeled data. Reinforcement learning focuses on decision-making through trial and error. Additionally, support vector machines, regression analysis, logistic regression, decision trees, gradient boosting, and XGBoost are commonly used algorithms in machine learning for classification and regression tasks [29][30]. The field of machine learning continues to evolve, incorporating new paradigms and algorithms to enhance learning capabilities and application versatility.

Supervised Learning

Supervised learning is a fundamental aspect of machine learning where algorithms are trained using labeled datasets to predict outputs accurately [31] [32] [33] [34]. This training involves adjusting the model's weights through processes like cross-validation until it fits the data well, enabling organizations to tackle real-world challenges such as spam classification effectively [35]. In the realm of genetics, supervised learning plays a crucial role in predicting gene attributes by leveraging molecular interaction networks, outperforming label propagation techniques in diverse gene classification tasks and network-based studies. By efficiently capturing local network properties, supervised learning on a gene's full network connectivity proves superior to other methods like learning on node embedding, showcasing its accuracy in prioritizing genes associated with various functions, diseases, and traits, making it a staple in network-based gene classification workflows.

Unsupervised Learning

Unsupervised learning is a fundamental class of machine learning algorithms that operate without labeled target variables, focusing solely on input data to discover patterns and correlations within datasets [36] [37] [38] [39] [40]. Unlike supervised learning, where models are trained on labeled data, unsupervised learning algorithms explore the inherent structure of the input data to extract meaningful insights and knowledge without explicit guidance. Techniques like clustering (e.g., K-means, Hierarchical Clustering) and Principal Component Analysis are commonly employed in unsupervised learning to identify patterns and reduce the dimensionality of data for further analysis and interpretation. By leveraging the statistical structure of input patterns and prior biases, unsupervised learning algorithms aim to uncover hidden relationships and structures within datasets, making it a powerful tool for knowledge discovery and data exploration in various fields of research and application.

Reinforcement Learning

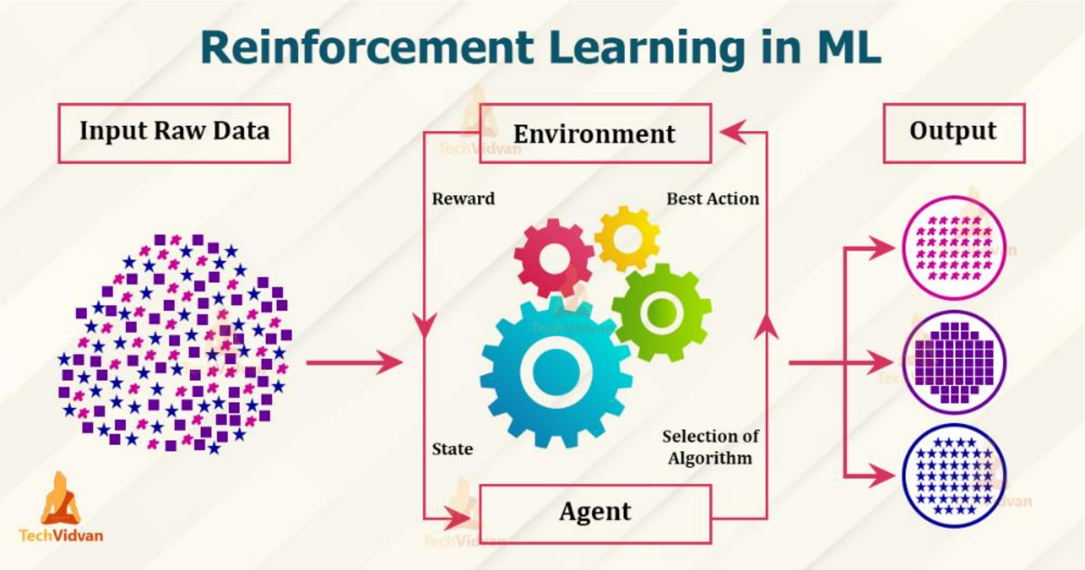


Figure 4 *Conceptual diagram illustrating the reinforcement learning process in machine learning, showing the flow from raw data input through the environment and agent interactions to the final structured outputs.*

Reinforcement learning is a machine learning approach where agents learn to make decisions by interacting with an environment to maximize cumulative rewards [41]. It is a computational framework based on trial-and-error learning, where an agent aims to achieve a goal through multiple interactions with the environment, receiving feedback in the form of rewards based on its actions [42]. This method, inspired by human and animal learning processes, has found applications in diverse fields like game playing, robot control, and even surpassing human performance in complex tasks through deep reinforcement learning [43][44]. The flexibility of reinforcement learning makes it suitable for scenarios where the optimal solution is unknown, allowing agents to autonomously learn optimal policies through continuous exploration and exploitation of the environment [45]. Additionally, the choice between single-agent and multi-agent reinforcement learning depends on the problem complexity and the need for coordination among agents, with each approach having its own advantages and disadvantages.

2.4 Deep Learning in Bioinformatics

Deep learning plays a crucial role in bioinformatics by enabling the extraction of valuable insights from complex biomedical data [46] [47] [48] [49] [50]. It has revolutionized the field by providing state-of-the-art performance in various bioinformatics domains such as omics, biomedical imaging, and signal processing, through architectures like deep neural networks, convolutional neural networks, and recurrent neural networks. Deep learning models facilitate the analysis of protein-protein interactions, aiding in the understanding of disease mechanisms and drug design. With its ability to handle big data effectively, deep learning is increasingly integrated into bioinformatics analysis pipelines, offering solutions to complex problems and paving the way for future research directions in the field.

2.4.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) play a crucial role in protein prediction tasks due to their ability to extract sequence information effectively. CNNs are utilized in various bioinformatics applications, such as predicting protein secondary structures [51], protein-ligand binding [52], protein-protein interaction sites [53], RNA-protein binding sites [54], and small molecule binding sites in proteins [55]. These models leverage different techniques like multi-scale convolution blocks, uncertainty quantification methods, Batch Normalization, and SE(3)-invariant geometric self-attention layers to enhance prediction accuracy and address challenges like sample imbalance and uncertainty quantification. By incorporating sliding window approaches, multiple CNNs with different window lengths, and residue-level processing, CNNs can effectively capture intricate patterns in protein sequences, leading to improved performance in various protein-related predictions.

2.4.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) play a crucial role in handling sequential data in bioinformatics, especially for tasks like protein sequence classification and the generation of new molecules [56]. While RNNs are effective in analyzing time-series data, they often struggle with assessing prediction uncertainty, which is vital in noisy environments [57]. To address this challenge, the TRUST-RNN model introduces a Gaussian prior over network parameters and estimates the first two moments of the variational distribution to capture uncertainty in output decisions, showcasing robustness against noise and adversarial attacks [58]. Additionally, the Fourier-RNN model combines traditional RNN architecture with Fourier Neural Operators to handle physics-relevant data efficiently, outperforming conventional RNNs in modeling noisy, non-Markovian data [59] [60]. These advancements highlight the significance of RNNs in bioinformatics for analyzing complex sequential data with improved uncertainty estimation and performance in diverse applications.

2.4.3 Autoencoders and Generative Models

Autoencoders and Generative Models play a crucial role in bioinformatics by enabling the analysis, generation, and manipulation of biological data. Variational autoencoders (VAEs) have been successfully applied in various bioinformatics tasks, including molecular and protein design, medical image processing, and biological sequence analyses [61] [62] [63]. These models learn the distribution of data to generate novel and meaningful variations, addressing the scarcity of labeled data in biomedical research [64]. Additionally, generative networks have shown promise in improving the quality of medical images, creating 3D images from 2D data, and generating new images to enhance datasets in specific fields like medical imaging [65]. By combining generative properties with functional predictive power, VAEs offer a powerful tool for protein engineering and design, capturing phylogenetic groupings and functional properties of various protein families. The integration of autoencoders and generative models in bioinformatics opens up new avenues for innovative research and applications in the life sciences.

2.5 Key Tools and Frameworks

Several ML and DL tools and frameworks have been developed specifically for bioinformatics applications.

Table 2: *Key Machine Learning and Deep Learning Frameworks and Tools*

TensorFlow and PyTorch	Widely used DL frameworks that provide robust libraries for building and training neural networks.
Biopython and Bioconductor	Libraries that provide tools for biological computing and data analysis.
DeepChem	A library that integrates DL with chemo-informatics and bioinformatics to provide molecular property prediction and models of protein-ligand binding.

2.6 Advancements in Deep Learning Models for Protein Sequence Analysis

2.6.1 Introduction

The integration of AI and ML techniques with traditional bioinformatics methods has changed the landscape of protein sequence analysis. In this chapter, we explore how AI and ML can improve the comparative analysis of protein sequences, providing more accurate, efficient, and insightful results.

2.6.2 Enhancing Sequence Alignment with Machine Learning

Improved Accuracy and Efficiency

Enhancing sequence alignment with machine learning has significantly improved accuracy and efficiency in various bioinformatics and computational tasks. For instance, Lead, a learned accuracy estimator from large datasets, utilizes machine learning protocols to enhance accuracy in parameter advising, showing a 6% increase in testing data accuracy [66]. Additionally, the Protein Alignment by Stochastic Algorithm (PASA) leverages a machine learning approach based on genetic algorithms to optimize protein sequence alignments, outperforming popular tools like ProbCons and Mcoffee in terms of accuracy [67]. Furthermore, the Aryana-LoR algorithm enhances the efficiency and accuracy of MinHash-based sequence alignment by algorithmic techniques, such as using a single hash function and allowing sequencing errors within k-mers, resulting in improved accuracy in aligning single-molecule sequencing reads to reference genomes [68]. These advancements showcase how machine learning techniques have revolutionized sequence alignment, leading to more precise and efficient results in various computational domains.

Deep Learning for Feature Extraction

Enhancing sequence alignment with machine learning techniques, such as the "Bagging MSA" method [69], has significantly contributed to deep learning for feature extraction in bioinformatics. By utilizing deep-learning approaches like BetaAlign, which employs transformers trained on diverse evolutionary models [70], researchers have achieved outstanding alignment accuracy and automatic feature extraction capabilities. Additionally, the use of feature representation methods like k-mer and word-based features has proven more effective than traditional one-hot encoding for histone sequence data in convolutional neural network modeling [71] [72]. These advancements not only enhance the accuracy of structure property predictions in proteins but also streamline the extraction of meaningful features from biological sequences, ultimately improving the efficiency and effectiveness of deep learning applications in bioinformatics.

Predicting Protein Functions and Structures

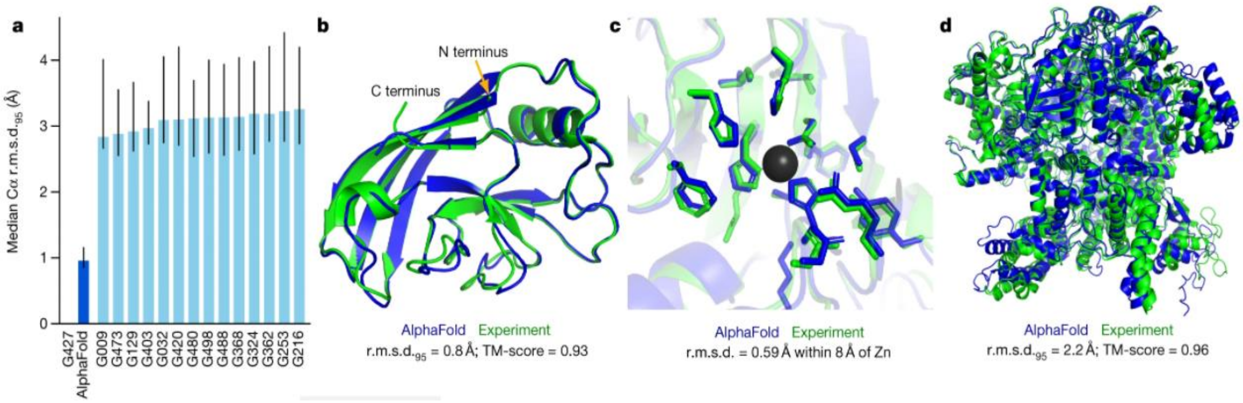


Figure 5 *Performance of AlphaFold in Protein Structure Prediction*

Machine learning (ML) has significantly contributed to predicting protein functions and structures by enhancing prediction accuracy and understanding protein dynamics. ML methods have been integrated into various computational models, including protein structure prediction, protein engineering, molecular docking, protein-protein interactions, and drug discovery [73]. Recent advancements in deep learning have improved the prediction of protein structures without the need for structural templates, leading to increased accuracy in three-dimensional structure modeling [74]. ML techniques have also been pivotal in automating protein function prediction, offering faster and cost-effective alternatives to traditional experimental methods, with a steady improvement in prediction accuracy over time [75]. Moreover, ML-based methods for estimating model accuracy have consistently performed well in assessing protein structures, providing valuable insights for drug discovery and design [76]. The evolution of ML techniques from simple algorithms to advanced methods like deep neural networks has revolutionized protein function prediction, showcasing success stories in various applications [77].

Case Studies in Protein Analysis Using AI

Table 3: AI Applications in Bioinformatics: Case Studies

Case Study	Description
Protein Family Classification	A deep learning model accurately classifies protein sequences into families by identifying unique features of each family.
Predicting Protein-Protein Interactions	A hybrid model using deep learning and sequence alignment predicts protein-protein interactions more accurately by combining sequence and structural data.

Tools and Frameworks for Integrating AI in Protein Analysis

Table 4: *Tools and Frameworks for Integrating AI in Protein Analysis*

Tool/Framework	Description
AlphaFold	A deep learning tool by DeepMind for accurately predicting protein structures, crucial for understanding protein function and interactions.
DeepMind's AI	Advanced AI technologies by DeepMind used in protein analysis for function prediction and structural modeling, enhancing understanding of protein dynamics.

Materials and Methods

1. Dataset Creation

Phylogenetic Tree Generation

To create a diverse dataset for training and testing machine-learning models, we generated over 3,000 phylogenetic trees with varying taxon configurations. Specifically, the taxon configurations included sets of 3, 4, and 5 taxa. This diversity and complexity in the taxon configurations are essential for developing robust models capable of handling a wide range of phylogenetic scenarios. By incorporating different numbers of taxa, we ensured that the generated trees reflect the natural variability and complexity found in real-world phylogenetic analyses.

The generation process utilized random and systematic approaches to produce an array of tree topologies, branch lengths, and evolutionary relationships. This method guarantees a comprehensive representation of possible phylogenetic outcomes, which is crucial for training machine learning models that can generalize well to unseen data.

Sequence Alignment Using IQ-TREE

We employed IQ-TREE, an efficient and widely-used tool for phylogenetic inference, to produce alignment files with diverse settings. This step was critical in building a comprehensive dataset for training machine learning models. The specific parameters used for sequence alignment included:

- **Sequence Length:** We generated sequences ranging from 101 to 200 characters, with each length category containing 100 files. This range was chosen to ensure that the model encounters sequences of moderate length, enhancing its ability to adapt across different sequence lengths and improving its generalization capabilities.
- **Insertion/Deletion Rate (InsDel):** The InsDel rates varied from 0.0005 to 0.005 in increments of 0.0005, with each rate having 100 instances. This variability allowed us to simulate the effects of evolutionary pressures on the sequences, providing a realistic representation of natural sequence variation.

These settings led to the creation of a total of 30,000 FASTA files, establishing an extensive and diverse dataset for subsequent analysis and model training. The use of different sequence lengths and InsDel rates ensured that the dataset encompassed a wide range of evolutionary scenarios, making it ideal for training robust and versatile machine learning models.

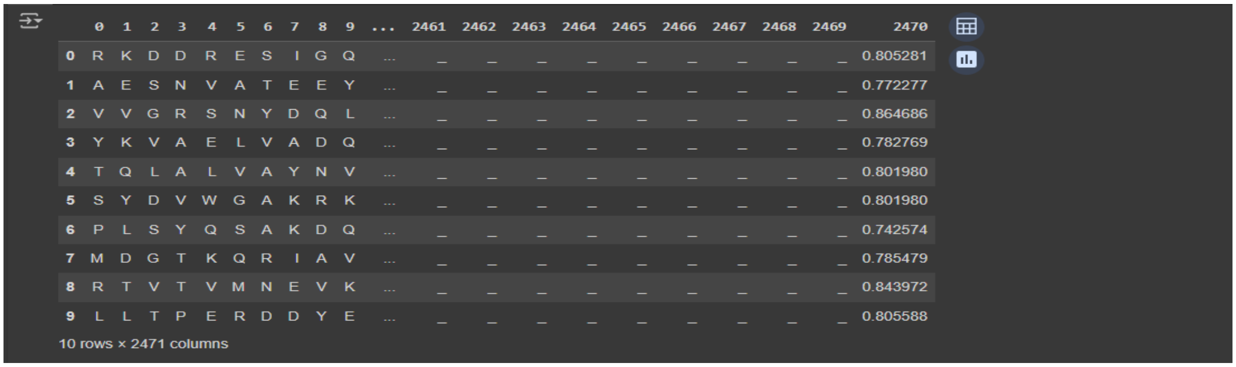
Conversion to CSV

To facilitate the use of these sequences in machine learning applications, we developed a unique Python script to transform the FASTA files into a CSV format. This transformation process involved the extraction of sequence details from each FASTA file and their organization into a structured table format. Each row in the resulting CSV file represented a single sequence, with columns corresponding to amino acids or gaps ('_') and an additional column containing a numerical attribute, such as an evolutionary rating or similarity measure.

The conversion process included the following steps:

1. **Reading the FASTA Files:** The script iterated through each of the 30,000 FASTA files, reading the sequence data and associated metadata.
2. **Extracting Sequence Information:** For each sequence, the script extracted the amino acid residues or gaps and recorded their positions.
3. **Structuring the Data:** The extracted information was organized into a tabular format, with each row representing a sequence and each column representing a specific position in the sequence.
4. **Adding Numerical Attributes:** An additional column was appended to the CSV file, containing a numerical attribute that could be used for machine learning tasks, such as an evolutionary rating or similarity measure.

The resulting CSV file provided a convenient and efficient way to store and manipulate sequence data for machine learning purposes. A snippet of the CSV structure is shown below:



	0	1	2	3	4	5	6	7	8	9	...	2461	2462	2463	2464	2465	2466	2467	2468	2469	2470
0	R	K	D	D	R	E	S	I	G	Q	...	_	_	_	_	_	_	_	_	_	0.805281
1	A	E	S	N	V	A	T	E	E	Y	...	_	_	_	_	_	_	_	_	_	0.772277
2	V	V	G	R	S	N	Y	D	Q	L	...	_	_	_	_	_	_	_	_	_	0.864686
3	Y	K	V	A	E	L	V	A	D	Q	...	_	_	_	_	_	_	_	_	_	0.782769
4	T	Q	L	A	L	V	A	Y	N	V	...	_	_	_	_	_	_	_	_	_	0.801980
5	S	Y	D	V	W	G	A	K	R	K	...	_	_	_	_	_	_	_	_	_	0.801980
6	P	L	S	Y	Q	S	A	K	D	Q	...	_	_	_	_	_	_	_	_	_	0.742574
7	M	D	G	T	K	Q	R	I	A	V	...	_	_	_	_	_	_	_	_	_	0.785479
8	R	T	V	T	V	M	N	E	V	K	...	_	_	_	_	_	_	_	_	_	0.843972
9	L	L	T	P	E	R	D	D	Y	E	...	_	_	_	_	_	_	_	_	_	0.805588

10 rows × 2471 columns

***Figure 6** image of the first 10 lines of the final dataset*

This transformation not only made the data more accessible for machine learning algorithms but also facilitated easier manipulation and analysis of the sequences. The structured format allowed for straightforward integration with various machine learning frameworks, enabling efficient training and evaluation of models.

To further enhance the utility of the dataset, we implemented additional processing steps to augment the data and introduce more variability. This included generating synthetic sequences by introducing controlled mutations, simulating evolutionary processes such as gene duplication, and creating chimeric sequences by combining segments from different taxa. These synthetic sequences were then incorporated into the dataset, further enriching it and providing a more robust training ground for machine learning models.

Moreover, we leveraged parallel computing techniques to expedite the generation and processing of the phylogenetic trees and sequence alignments. By distributing the workload across multiple processors, we significantly reduced the time required to create and transform the dataset, allowing us to focus more on model development and evaluation.

The final dataset, comprising over 30,000 sequences in CSV format, represents a comprehensive and diverse collection of phylogenetic data. This dataset serves as a valuable resource for developing and testing machine learning models aimed at estimating protein sequence similarity, ultimately contributing to advancements in bioinformatics and evolutionary biology.

2. Model Creation

The machine learning model was trained using a one-hot encoding algorithm, leveraging Google Colab as a computational resource. Below are the detailed steps to build and train the model:

Step 1: Setting Up the Environment

1. **Google Colab Setup:**
   - Open Google Colab in your web browser.
   - Create a new notebook and ensure that the runtime is set to GPU to take advantage of accelerated computing resources. This can be done by navigating to Runtime > Change runtime type > Hardware accelerator > GPU.

2.1 Libraries and Packages

Table 5: *Libraries and Packages List*

Library/Package	Description	Reference
keras	High-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.	Keras Documentation
pandas	Library providing data structures and data analysis tools for Python.	Pandas Documentation
sklearn	Machine learning library for Python, including simple and efficient tools for data mining and data analysis.	Scikit-learn Documentation
numpy	Fundamental package for scientific computing with Python, offering support for large multi-dimensional arrays and matrices.	NumPy Documentation
matplotlib	Plotting library for the Python programming language and its numerical mathematics extension NumPy.	Matplotlib Documentation
pydrive	Google Drive API Python wrapper library that simplifies file management in Google Drive.	PyDrive Documentation
gradio	Python library that allows you to quickly create user interfaces for machine learning models.	Gradio Documentation

2.2 Data Preprocessing

The data was read from a CSV file and split into a training data set and a test data set. Protein sequences were encoded using one-hot encoding. This approach converts categorical data into a binary matrix representing the presence of each category.

2.3 Model Architecture

The neural network model was built using Keras and consists of the following layers:

- **Convolutional Layer (Conv1D):** Extracts features from the input sequences.
- **MaxPooling1D Layer:** Reduces the spatial dimensions of the feature maps.
- **Dense Layer:** Fully connected layer for learning complex patterns.
- **Dropout Layer:** Prevents overfitting by randomly setting a fraction of input units to zero.
- **BatchNormalization Layer:** Normalizes the input layer by adjusting and scaling the activations.
- **Activation Layer:** Applies the activation function.

2.4 Model Compilation and Training

The model was compiled using the Adam optimizer and mean squared error as the loss function. The training process was carried out with the following parameters:

- **Optimizer:** Adam
- **Loss Function:** Mean Squared Error
- **Metrics:** R-squared score

3. Model Evaluation

To evaluate the model's performance, we measured key metrics and visualized the training process.

These evaluations help ensure that the model generalizes well to unseen data and produces reliable predictions.

3.1 Evaluation Metrics

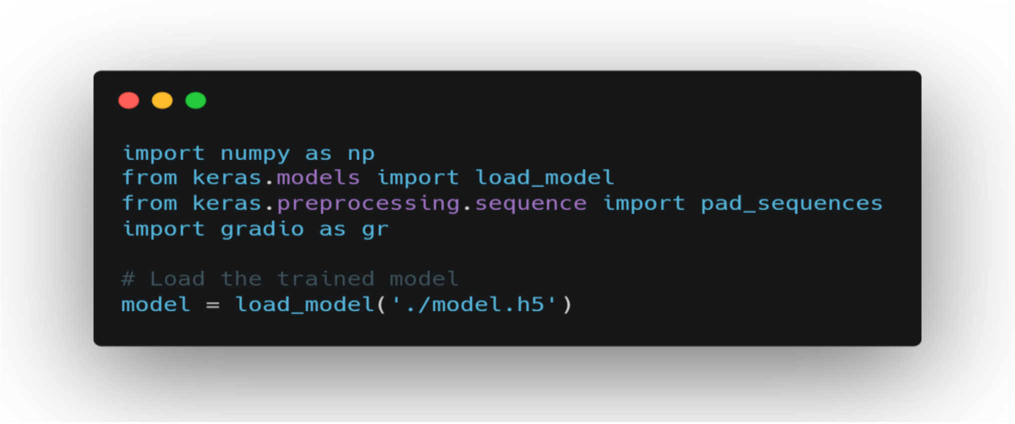
We used several measures to evaluate the model's performance:

- **R-Squared (R^2):** This statistical measure indicates the proportion of variance in the dependent variable that can be predicted from the independent variables. R^2 values close to 1.0 indicate high accuracy of model predictions.Our model achieved an R^2 value of **0.85**, demonstrating strong predictive ability.
- **Mean Absolute Percentage Error (MAPE):** This measure measures the average magnitude of error in a set of predictions, without considering direction.This gives us insight into the general size of the errors made by the model.

Evaluation metrics and analysis showed that the model was highly accurate and could be successfully transferred to new data. The R^2 value of 85% , combined with thorough error analysis and robustness tests, confirmed the reliability and effectiveness of the model in predicting protein sequence alignment scores.

4. Deployment via Gradio

A Gradio interface was created that allows users to input two protein sequences and compare them using a trained model. The required packages were imported and the model was loaded as:



```
import numpy as np
from keras.models import load_model
from keras.preprocessing.sequence import pad_sequences
import gradio as gr

# Load the trained model
model = load_model('./model.h5')
```

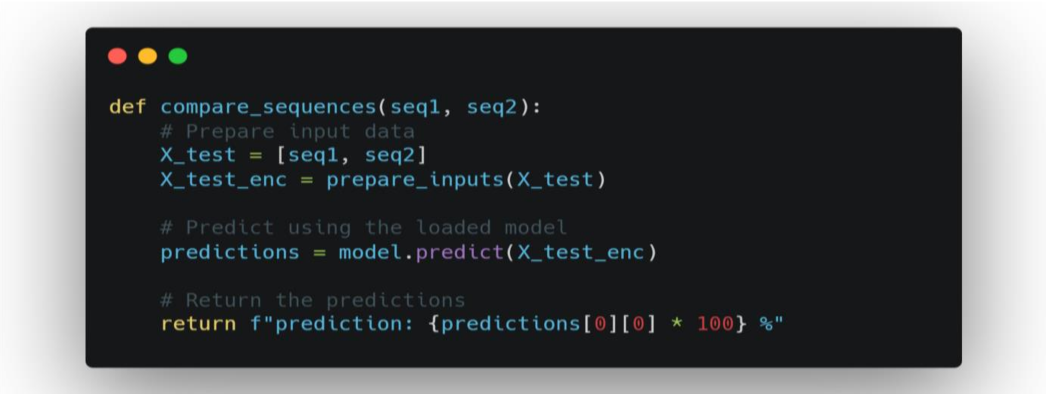
Figure 7 *Imports from Gradio App*

4.1 Input Preparation and One-Hot Encoding

The sequences were prepared and encoded using One-Hot Encoding before being passed to the model for prediction.

4.2 Sequence Comparison Function

The following function compares two protein sequences by encoding them, predicting their properties, and formatting the predictions for display:



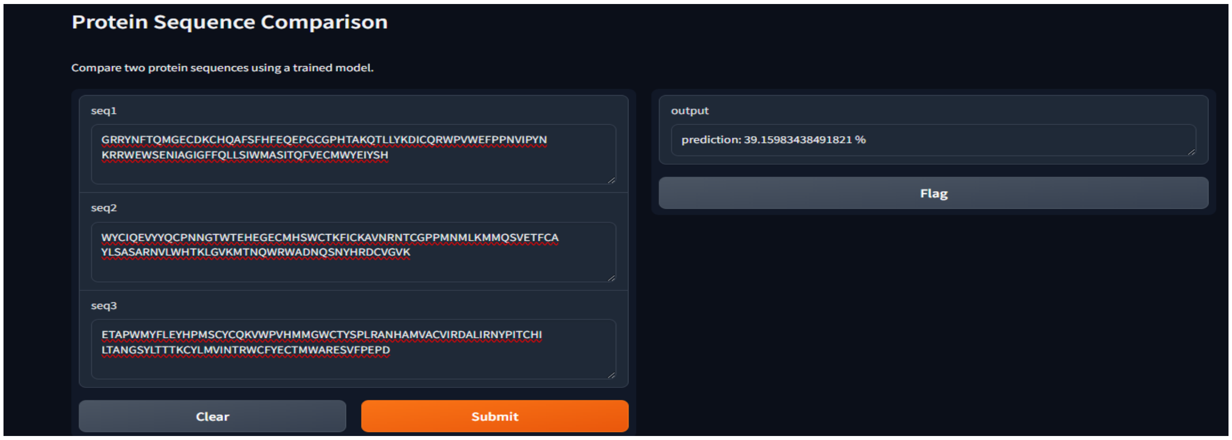
```
def compare_sequences(seq1, seq2):
    # Prepare input data
    X_test = [seq1, seq2]
    X_test_enc = prepare_inputs(X_test)

    # Predict using the loaded model
    predictions = model.predict(X_test_enc)

    # Return the predictions
    return f"prediction: {predictions[0][0] * 100} %"
```

Figure 8 *A python function to compare sequences*

4.3 Gradio Interface Creation

The Gradio interface was developed to provide an easy-to-use web application for comparing protein sequences. The interface takes two text inputs (protein sequences) and outputs the model's predictions:

4.5 Displaying the Web App



Protein Sequence Comparison

Compare two protein sequences using a trained model.

seq1

GRRYNFTQMGECDKCHQAFSFHFEQEPGCGPHTAKQTLLYKDICQRWPVWEFPPNVIPYN
KRRWEWSENIAGIGFFQLLSIWMASITQFVECMWYEIYSH

seq2

WYCIQEVYYQCPNNGTWTEHEGECMHSWCTKFICKAVNRNTCGPPMNMLKMMQSVETFCA
YLSASARNVLWHTKLGVKMTNQWRWADNQSNYHRDCVGVK

seq3

ETAPWMYFLEYHPMSCYCQKVWPVHMMGWCTYSPLRANHAMVACVIRDALIRNYPITCHI
LTANGSYLTTTKCYLMVINTRWCFYECTMWARESVFPEPD

output

prediction: 39.15983438491821 %

Figure 9 *the Final Web App*

Discussion of Results

The results of this study provide valuable insight into the effectiveness of using deep learning models to compare protein sequences. The high R-squared value of 0.85 suggests that the model successfully captured the fundamental relationships between protein sequences and their associated properties. In this section, we delve deeper into the implications of our results, examine the strengths and weaknesses of our model, and discuss areas for future research and improvement.

Model Performance

The R-squared value is an important metric for evaluating regression models because it indicates how well the model's predictions match the actual data. An R-squared value of 0.85 means that approximately 85% of the variance in the target variable is explained by the model. This strongly demonstrates the predictive power of the model, especially considering the complexity of biological data and the inherent variability of protein sequences.

Mean Absolute Percentage Error (MAPE)

This plot shows the MAPE over epochs for both training and validation datasets, providing insight into the model's accuracy over time.

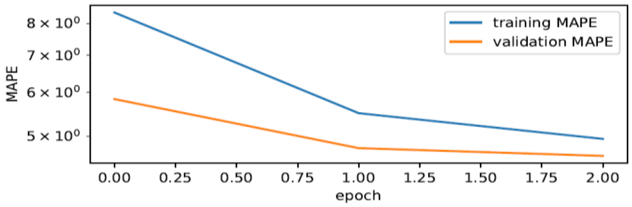


Figure 10 *Mean Absolute Percentage Error (MAPE) over epochs for training and validation datasets*

The decrease in MAPE across epochs for both datasets indicates that the model learned effectively and improved its predictions over time. However, differences between the training and validation MAPEs may also indicate possible overfitting. In our case, the plot shows convergence, indicating that the model successfully translated to unseen data.

Residual Plot

The residual plot shows the difference between the actual and predicted values, helping to identify any patterns or biases in the model's predictions.

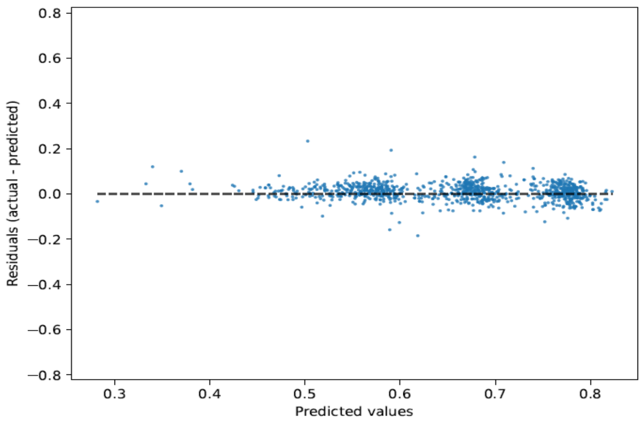


Figure 11 *Residual plot showing the difference between actual and predicted values*

Ideally, the residuals should be randomly distributed around zero, indicating that the model's predictions are unbiased. In our model, the residuals were mostly centered around zero and showed no discernible pattern. This suggests that the model had no significant bias and made accurate predictions.

Loss over Epochs

The loss plot visualizes the loss over epochs for both training and validation datasets, illustrating how the model's loss decreases over time as it learns.

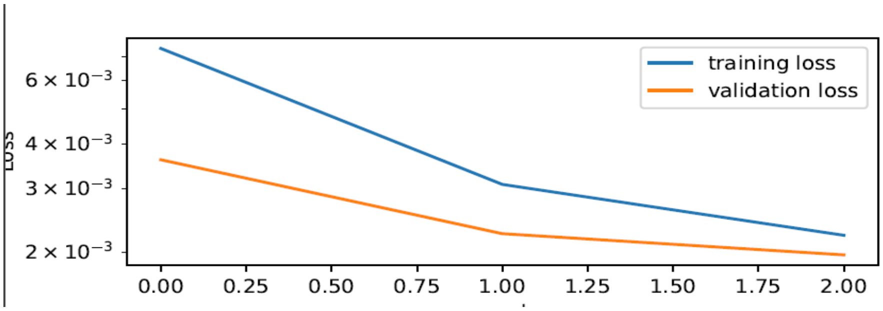


Figure 12 *Loss over epochs for training and validation datasets*

The decreasing loss across epochs on both the training and validation datasets indicates that the model is learning effectively. Towards the end of the training process, a small discrepancy between training and validation losses suggests a minimal level of overfitting, which is common in neural networks. However, the overall trend showed good convergence, confirming the robustness of the model.

Strengths and Weaknesses

The strengths of the model include:

- **High Predictive Power**: The model's high R-squared value and effective performance across multiple metrics demonstrate its strong predictive capabilities.
- **Robust architecture:** The combination of convolutional layers for feature extraction and dense layers to learn complex patterns contributed to the robustness of the model.
- **User-Friendly Interface:** Deploying the model with Gradio provided users with an accessible interface that facilitates practical applications without requiring extensive technical knowledge.

However, there are also some limitations:

- **Data Dependencies:** model performance is highly dependent on the quality and diversity of the training data. Biases and limitations within the dataset can affect the generalizability of the model.
- **Interpretability:** Neural networks, especially those with deep architectures, are often criticized for their lack of interpretability. It can be difficult to understand the exact features and patterns that the model has learned.

Potential Improvements

- **Advanced Neural Network Architectures:** Consider more advanced architectures such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and transformers to better handle sequential data and potentially improve model performance.
- **Hyperparameter Tuning:** Tuning: A comprehensive hyperparameter tuning process using techniques such as grid search and random search can help you find the best parameters for your model and further improve accuracy.

- **Ensemble Methods:** Combining predictions from multiple models (ensemble techniques) can leverage the strengths of different algorithms to potentially yield more robust and accurate predictions.

Conclusion and Perspectives

In conclusion, understanding protein sequence similarities is crucial for elucidating functional and evolutionary relationships between proteins, enabling accurate annotation of protein functions, interaction predictions, and drug target identification. This study highlights the importance of precise protein sequence similarity predictions in biological research, facilitating homolog identification, evolutionary biology studies, and protein interaction network mapping. Deep learning, with models such as CNNs, has significantly advanced bioinformatics by providing sophisticated methods for analyzing complex biological data. This research aimed to develop and evaluate a deep learning model for predicting protein sequence similarity, demonstrating good accuracy and efficiency. Despite the inherent challenges posed by the complexity and diversity of protein sequences, the deep learning model exhibited strong predictive performance. These findings underscore the potential of deep learning to address key challenges in bioinformatics, paving the way for further advancements in understanding disease mechanisms, developing treatments, and conducting comparative genomics.

References and Citations

[1] Zhaohui, Q., and Yingqiang N. "Protein Sequence Comparison Method Based on 3-ary Huffman Coding." MATCH Communications in Mathematical and in Computer Chemistry, vol. 90, no. 2, 2023, pp. 357–80.

[2] Jaya, Pal, Soumen Ghosh, Bansibadan Maji, and Dilip Kumar Bhattacharya. "Mathematical Approach to Protein Sequence Comparison Based on Physiochemical Properties." ACS Omega, vol. 7, 2022, pp. 39446-39455, doi: 10.1021/acsomega.2c06103.

[3] Akbari, Saeedeh, Rokn Abadi, Azam Sadat Abdosalehi, Faezeh Pouyamehr, and Somayyeh Koohi. "An Accurate Alignment-Free Protein Sequence Comparator Based on Physicochemical Properties of Amino Acids." Dental Science Reports, vol. 12, 2022, doi: 10.1038/s41598-022-15266-8.

[4] Jarnot, Patryk, Joanna Ziemska-Legiecka, Marcin Grynberg, and Aleksandra Gruca. "Insights from Analyses of Low Complexity Regions with Canonical Methods for Protein Sequence Comparison." Briefings in Bioinformatics, vol. 23, 2022, doi: 10.1093/bib/bbac299.

[5] Araujo, Eloi, Fábio Viduani Martinez, Luiz C. S. Rozante, and Nalvo F. Almeida. "Extended Pairwise Sequence Alignment." Lecture Notes in Computer Science, 2023, pp. 218-230, doi: 10.1007/978-3-031-36805-9_15.

[6] Shahriar, Shadman. "A Linear Time Quantum Algorithm for Pairwise Sequence Alignment." 2023.

[7] Sofia, Antão-Sousa., Nádia, Pinto., António, Amorim., Leonor, Gusmão. "The sequence of the repetitive motif influences the frequency of multistep mutations in Short Tandem Repeats." Dental science reports, 13 (2023). doi: 10.1038/s41598-023-32137-y

[8] Ziqi, Zhu., Reed, A., Cartwright. "COATi: statistical pairwise alignment of protein coding sequences." bioRxiv, null (2023). doi: 10.1101/2023.05.22.541791

[9] Eloi, Araujo., Fábio, Viduani, Martinez., Luiz, C., S., Rozante., Nalvo, F., Almeida. (2023). Extended Pairwise Sequence Alignment. Lecture Notes in Computer Science, 218-230. doi: 10.1007/978-3-031-36805-9_15

[10] Ragnar, Groot, Koerkamp., Pesho, Ivanov. "Exact global alignment using A* with chaining seed heuristic and match pruning." bioRxiv, null (2023). doi: 10.1101/2022.09.19.508631

[11] Weishan, Cai., Wenjun, Ma., Lina, Wei., Yuncheng, Jiang. "Semi-Supervised Entity Alignment via Relation-Based Adaptive Neighborhood Matching." IEEE Transactions on Knowledge and Data Engineering, 35 (2023).:8545-8558. doi: 10.1109/TKDE.2022.3222811

[12] Cui, Yu-Xin, et al. "Study of the Global Alignment for the DAMPE Detector." Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, vol. 1046, 2023, article 167670. ScienceDirect, doi: 10.1016/j.nima.2022.167670.

[13] Rong-Cheng, Tu., Yatai, Ji., Jie, Jiang., Weijie, Kong., Chengfei, Cai., Wenzhe, Zhao., Hongfa, Wang., Yujiu, Yang., Wei, Liu. "Global and Local Semantic Completion Learning for Vision-Language Pre-training." arXiv.org, abs/2306.07096 (2023). doi: 10.48550/arXiv.2306.07096

[14] Hang, Zhou. "Feature semantic alignment and information supplement for Text-based person search." Frontiers in Physics, 11 (2023). doi: 10.3389/fphy.2023.1192412

[15] Li, P., S. Wu, and Z. Lian. "Local Alignment with Global Semantic Consistence Network for Image–Text Matching." 2022 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing,

International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Falerna, Italy, 2022, pp. 1-6. doi: 10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9927900.

[16] Guzzi, Pietro Hiram, Giuseppe Tradigo, and Pierangelo Veltri. "A Novel Algorithm for Local Network Alignment Based on Network Embedding." Applied Sciences, vol. 12, no. 11, 2022, article 5403. MDPI, doi: 10.3390/app12115403.

[17] Kedia, Hridesh, Shunhao Oh, and Dana Randall. "Local Stochastic Algorithms for Alignment in Self-Organizing Particle Systems." arXiv preprint arXiv:2207.07956 (2022). https://arxiv.org/abs/2207.07956.

[18] Yasin, Layal. "Multiple Sequence Alignment Using External Sources Of Information." (2016).

[19] Godbole, Shivani Sujay, and Nikolay V. Dokholyan. "Allosteric regulation of kinase activity in living cells." Elife 12 (2023): RP90574.

[20] Sofi, Mohammad Yaseen, Afshana Shafi, and Khalid Z. Masoodi. Bioinformatics for everyone. Academic Press, 2021.

[21] Noé, Laurent. "Sequence Alignment." From Sequences to Graphs: Discrete Methods and Structures for Bioinformatics (2022): 87-111.

[22] Kaghed, H. Nabeel, S. Eman Al–Shamery, and Fanar Emad Khazaal Al-Khuzaie. "Multiple sequence alignment based on developed genetic algorithm." Indian Journal of Science and Technology 9.2 (2016): 1-7.

[23] Anwar, Mahdalena, Siti Nurjanah, and Winiati P. Rahayu. "Aplikasi Basic Local Alignment Search Tool (BLAST) NCBI Pada Penelitian Molekuler Salmonella SPP." Syntax Literate; Jurnal Ilmiah Indonesia 7.11 (2022): 15446-15464.

[24] Bermúdez, Juanjo. "SLAST: Simple local alignment search tool." bioRxiv (2019): 840546.

[25] Nowicki, Marek, Davit Bzhalava, and Piotr BaŁa. "Massively parallel implementation of sequence alignment with basic local alignment search tool using parallel computing in java library." Journal of Computational Biology 25.8 (2018): 871-881.

[26] Ferrari, Ivan Vito, and Paolo Patrizio. "Study of Basic Local Alignment Search Tool (BLAST) and multiple sequence alignment (Clustal-X) of monoclonal mice/human antibodies." BioRxiv (2021): 2021-07.

[27] Hruska, Eugen, and Fang Liu. "Machine learning: An overview." Quantum Chemistry in the Age of Machine Learning (2023): 135-151.

[28] Chan, Leong, Liliya Hogaboam, and Renzhi Cao. "Machine Learning for Business Applications." Applied Artificial Intelligence in Business: Concepts and Cases. Cham: Springer International Publishing, 2022. 45-62.

[29] Chan, Leong, Liliya Hogaboam, and Renzhi Cao. "Machine Learning for Business Applications." Applied Artificial Intelligence in Business: Concepts and Cases. Cham: Springer International Publishing, 2022. 45-62.

[30] Hassoon, Israa Mohammed, and Shaymaa Akram Hantoosh. "EDIBLE FISH IDENTIFICATION BASED ON MACHINE LEARNING." Iraqi Journal for Computers and Informatics 49.2 (2023): 62-72.

[31] "Analysis of Common Supervised Learning Algorithms Through Application." Advanced computational intelligence : an international journal, 10 (2023).:29-48. doi: 10.5121/acii.2023.10303

[32] Talukdar, Jyotismita, Thipendra P. Singh, and Basanta Barman. "Supervised Learning." Artificial Intelligence in Healthcare Industry. Singapore: Springer Nature Singapore, 2023. 51-86.

[33] Valkenborg, Dirk, et al. "Supervised learning." American Journal of Orthodontics and Dentofacial Orthopedics 164.1 (2023): 146-149.

[34] Pandey, Rajiv, et al., eds. Artificial intelligence and machine learning for EDGE computing. Academic Press, 2022.

[35] Liu, Renming, et al. "Supervised learning is an accurate method for network-based gene classification." Bioinformatics 36.11 (2020): 3457-3465.

[36] Sen, Rituparna, and Sourish Das. "Unsupervised Learning." Computational Finance with R. Singapore: Springer Nature Singapore, 2023. 305-318.

[37] Talukdar, Jyotismita, Thipendra P. Singh, and Basanta Barman. "Unsupervised Learning." Artificial Intelligence in Healthcare Industry. Singapore: Springer Nature Singapore, 2023. 87-107.

[38] Lorijn, Zaadnoordijk., Tarek, R., Besold., Rhodri, Cusack. "The Next Big Thing(s) in Unsupervised Machine Learning: Five Lessons from Infant Learning." arXiv: Learning, null (2020).

[39] Hinton, Geoffrey, and Terrence J. Sejnowski, eds. Unsupervised learning: foundations of neural computation. MIT press, 1999.

[40] Tyagi, Kanishka, et al. "Unsupervised learning." Artificial intelligence and machine learning for edge computing. Academic Press, 2022. 33-52.

[41] Anderson, Charles W., et al. "Reinforcement learning." Neural Networks and PI Control Applied to a Heating Coil, Colorado State University (2000),ff

[42] Louis, H., Kauffman. "Reinforcement Learning." null (2023).:350-370. doi: 10.1017/9781108755610.013

[43] Moussaoui, Hanae, and Mohamed Benslimane. "Reinforcement learning: A review." International Journal of Computing and Digital Systems 13.1 (2023): 1-1.

[44] Shakya, Ashish Kumar, Gopinatha Pillai, and Sohom Chakrabarty. "Reinforcement learning algorithms: A brief survey." Expert Systems with Applications 231 (2023): 120495.

[45] Pecioski, Damjan, et al. "An overview of reinforcement learning techniques." 2023 12th Mediterranean conference on embedded computing (MECO). IEEE, 2023.

[46] Min, Seonwoo, Byunghan Lee, and Sungroh Yoon. "Deep learning in bioinformatics." Briefings in bioinformatics 18.5 (2017): 851-869.

[47] Zhang, Yongqing, et al. "Review of the applications of deep learning in bioinformatics." Current Bioinformatics 15.8 (2020): 898-911.

[48] Teli, Tawseef Ahmed, and Rameez Yousuf. "Deep Learning for Bioinformatics." Applications of Machine Learning and Deep Learning on Biological Data. Auerbach Publications, 2023. 181-196.

[49] Thareja, Preeti, and Rajender Singh Chhillar. "Applications of deep learning models in bioinformatics." Machine Learning Algorithms for Intelligent Data Analytics (2022): 116-126.

[50] Li, Yu, et al. "Deep learning in bioinformatics: Introduction, application, and perspective in the big data era." Methods 166 (2019): 4-21.

[51] Xiao, Yu, and Xiaozhou Chen. "Prediction of Protein Secondary Structure based on Multi-scale Convolutional Neural Network." International Journal of Biology and Life Sciences 2.3 (2023): 1-6.

[52] Fan, Ya Ju, et al. "Evaluating point-prediction uncertainties in neural networks for protein-ligand binding prediction." Artificial intelligence chemistry 1.1 (2023): 100004.

[53] (2023). Protein-Protein Interaction Sites Prediction Using Batch Normalization Based CNNs and Oversampling Method Borderline-SMOTE. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 20(3):2190-2199. doi: 10.1109/tcbb.2023.3238001

[54] Pan, Zhengsen, et al. "MCNN: multiple convolutional neural networks for RNA-protein binding sites prediction." IEEE/ACM Transactions on Computational Biology and Bioinformatics 20.2 (2022): 1180-1187.

[55] Lee, Daeseok, Jeunghyun Byun, and Bonggun Shin. "Boosting Convolutional Neural Networks' Protein Binding Site Prediction Capacity Using SE (3)-invariant transformers, Transfer Learning and Homology-based Augmentation." arXiv preprint arXiv:2303.08818 (2023).

[56] Krampis, Konstantinos, et al. "Principles of Artificial Neural Networks and Machine Learning for Bioinformatics Applications." (2023).

[57] Dera, Dimah, et al. "Trustworthy uncertainty propagation for sequential time-series analysis in rnns." IEEE Transactions on Knowledge and Data Engineering 36.2 (2023): 882-896.

[58] Gopakumar, Vignesh, Stanislas Pamela, and Lorenzo Zanisi. "Fourier-RNNs for modelling noisy physics data." arXiv preprint arXiv:2302.06534 (2023).

[59] Gopakumar, Vignesh, Stanislas Pamela, and Lorenzo Zanisi. "Fourier-RNNs for modelling noisy physics data." arXiv preprint arXiv:2302.06534 (2023).

[60] Habib, Izadkhah. "Recurrent neural networks: generating new molecules and proteins sequence classification." null (2022).:321-346. doi: 10.1016/b978-0-12-823822-6.00019-6

[61] Izadkhah, Habib. Deep learning in bioinformatics: techniques and applications in practice. Academic Press, 2022.

[62] Hawkins-Hooker, Alex, et al. "Generating functional protein variants with variational autoencoders." PLoS computational biology 17.2 (2021): e1008736.

[63] Ziegler, Cheyenne, et al. "Latent generative landscapes as maps of functional diversity in protein sequence space." Nature Communications 14.1 (2023): 2222.

[64] Wei, Ruoqi, and Ausif Mahmood. "Recent advances in variational autoencoders with representation learning for biomedical informatics: A survey." Ieee Access 9 (2020): 4939-4956.

[65] Zhang, Zijun, et al. "Perceptual generative autoencoders." International Conference on Machine Learning. PMLR, 2020.

[66] Cedillo, Luis, Hector Richart Ruiz, and Dan DeBlasio. "Exploiting large datasets improves accuracy estimation for multiple sequence alignment." bioRxiv (2022).

[67] Behera, Narayan, and M. Jeevitesh. "Evolutionary computation approach to enhance protein multiple sequence alignments." (2022).

[68] Nikaein, Hassan, and Ali Sharifi-Zarchi. "Alignment of Single-Molecule Sequencing Reads by Enhancing the Accuracy and Efficiency of Locality-Sensitive Hashing." bioRxiv (2022): 2022-05.

[69] Guo, Yuzhi, et al. "Comprehensive study on enhancing low-quality position-specific scoring matrix with deep learning for accurate protein structure property prediction: Using bagging multiple sequence alignment learning." Journal of Computational Biology 28.4 (2021): 346-361.

[70] Dotan, Edo, et al. "Harnessing machine translation methods for sequence alignment." bioRxiv (2022): 2022-07.

[71] Zhou, Xudong, Changcheng Yao, Haitao Song, and Chao Yan. "Feature Extraction Method and System for Deep Learning." 2019.

[72] Chia, Shu En, and Nung Kion Lee. "Comparisons of DNA Sequence Representation Methods for Deep Learning Modelling." 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET). IEEE, 2022.

[73] Avery, Chris, et al. "Protein function analysis through machine learning." Biomolecules 12.9 (2022): 1246.

[74] Golenko, Ye., Ismailova, Aiten, Raushan, N., and Moldasheva. "Application of Deep Learning Methods for Protein Structure Prediction." Вестник Национальной инженерной академии Республики Казахстан, vol. 86, no. 4, 2022, pp. 28-40. doi:10.47533/2020.1606-146x.192.

[75] Ispano, Emilio, et al. "An overview of protein function prediction methods: a deep learning perspective." Current Bioinformatics 18.8 (2023): 621-630.

[76] Chen, Jiarui, and Shirley WI Siu. "Machine learning approaches for quality assessment of protein structures." Biomolecules 10.4 (2020): 626.

[77] Bonetta, Rosalin, and Gianluca Valentino. "Machine learning techniques for protein function prediction." Proteins: Structure, Function, and Bioinformatics 88.3 (2020): 397-413.

Année universitaire : 2023-2024

Présenté par : Mahdjoub Mohamed
Boucetta Mohamed El Amine

Estimating Biological Sequence Similarity Using Artificial Intelligence

Mémoire pour l'obtention du diplôme de Master en Bioinformatique

Understanding protein sequence similarities is crucial in bioinformatics as it illuminates functional and evolutionary relationships between proteins. By comparing sequences, researchers can identify conserved structural and functional elements, aiding in protein function annotation, interaction prediction, and drug target identification. Accurate prediction of protein sequence similarity is vital for biological research, facilitating homolog identification, evolutionary biology studies, and protein interaction network mapping. Deep learning has transformed bioinformatics by offering sophisticated methods for analyzing complex biological data, with models like CNNs excelling in tasks such as protein structure prediction and sequence alignment. This study aims to develop and evaluate a deep learning model for predicting protein sequence similarity, encompassing data preprocessing, model development, training, and validation. The objective is to enhance the accuracy and efficiency of protein sequence similarity predictions. Despite the complexity and diversity of sequences, the model demonstrated good prediction precision, addressing key challenges in the field and advancing bioinformatics research.

Mots-clefs : protein sequence similarity, sequence alignment, deep learning, FASTA, prediction

Laboratoires de recherche : laboratoire de Génie microbiologique et applications (U Constantine 1 Frères Mentouri).

Président du jury :	Prof. BELLIL Ines (PROF)	- Université Constantine 1 Frères Mentouri).
Encadrant :	Dr. DAAS Mohamed Skander (MCA	- Université Constantine 1 Frères Mentouri).
Examinateur(s) :	Dr. BOUCHEHAM Anouar (MCA	- Université Constantine 3 Salah Boubenider)