Conformal Predictors in Identifying Protein Sequences

Ankush Prakash Gowda

Submitted for the Degree of Master of Science in

Artificial Intelligence

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

August 30, 2023

Declaration

This report has been prepared based on my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name:

Date of Submission:

Signature:

Abstract

This project introduces an innovative approach to sequence analysis by integrating Conformal Predictors within the bioinformatics domain, addressing the challenges posed by the exponential growth of biological sequencing data. The study commences with a comprehensive exploration of two foundational concepts: Hidden Markov Models (HMMs) and Conformal Predictors, shedding light on their theoretical underpinnings and real-world applications. By delving into the intricacies of Conformal Predictors, the research demonstrates their efficacy in generating reliable prediction intervals, bolstering the credibility of sequence classification results. Simultaneously, a dedicated investigation into the role of HMMs in Bioinformatics underscores their significance in modelling biological sequences.

A pivotal juncture in this research lies in the application of conformal prediction algorithms to a selected protein family sourced from the PFAM database. This empirical phase serves as a testament to the capability of Conformal Predictors in yielding well-calibrated prediction intervals, effectively quantifying the confidence linked to sequence categorization outcomes. Building upon this foundation, the project constructs a robust framework for systematically assessing the credibility and confidence levels of individual HMMs, particularly those derived from PFAM. This innovative approach not only advances the field of bioinformatics but also elevates the reliability of sequence identification methodologies through the integration of conformal prediction principles.

Moreover, this project bridges the theoretical concepts of Hidden Markov Models and Conformal Predictors with their tangible applications in sequence analysis. The formulated methodology, designed to evaluate credibility and estimate confidence, offers a promising avenue to enhance the precision and dependability of sequence identification techniques. In doing so, it underscores the potential of Conformal Predictors to augment bioinformatics methodologies. Ultimately, this project adds a crucial dimension to the bioinformatics landscape, seamlessly merging theoretical foundations with practical implementation, and presents a significant stride towards refining the accuracy and credibility of sequence analysis results.

Table of Contents

1 Introduction

The explosion of biological sequence data in the era of data-driven biology has brought with it both unparalleled potential and challenges[27]. Understanding the functions and roles of sequences, such as proteins, requires accurate identification and classification. The integration of Conformal Predictors[2] with Hidden Markov Models (HMMs)[1] is one promising approach to addressing this difficulty, with the potential to improve the credibility and confidence estimation of sequence identification outcomes. This dissertation investigates the relationship between Conformal Predictors and HMMs, providing a thorough examination of their theoretical basis and practical applications.

Hidden Markov Models, which have shown useful in capturing the sequential dependencies inherent in biological sequences, serve as the study's foundational pillar[1]. Because of their capacity to model complicated patterns inside sequences, HMMs have found widespread application in bioinformatics, spanning from gene prediction to protein family discovery. Conformal Predictors, on the other hand, offer a fresh take on prediction by changing the emphasis from point predictions to prediction sets[2]. This paradigm provides a sound foundation for calculating prediction uncertainty and establishing confidence levels for specific forecasts.

This study adopts a two-pronged strategy to reconcile the theoretical underpinnings of HMMs and the promise of Conformal Predictors. First, a detailed report digs into the use of HMMs in bioinformatics, investigating their techniques, strengths, and limitations[1]. Concurrently, an in-depth examination of Conformal Predictors reveals information about their mathematical architecture and the procedures by which they establish prediction sets[2]. By combining these fundamental components, this study aims to develop a novel paradigm for evaluating the legitimacy and confidence of individual HMMs generated from databases such as PFAM.

This research offers a comprehensive framework that combines HMMs with Conformal Predictors and adds to the evolution of sequence analysis approaches as the digital biology age develops. The potential impact on biological discoveries and applications is enormous because calibrated confidence estimations can improve the forecasting capacities of HMMs. The following chapters of this dissertation go into detail about the actual application, outcomes, and implications of this ground-breaking approach, laying the foundation for a more solid and trustworthy paradigm for sequence identification.

2 Background Research

2.1 Introduction to Sequence Analysis in Bioinformatics

2.1.1 Significance of Sequence Analysis in Biological Research

The analysis of biological sequences has become a key component of contemporary biological research in the era of genomics and proteomics. A living thing's structure, function, and evolutionary history are all encoded in its basic building blocks of DNA, RNA, and proteins[3]. Scientists may understand genetic variations, regulatory components, and molecular interactions by being able to read these sequences, which helps them to understand the complex systems that underpin biological processes[4].

2.1.2 Computational Tools for Sequence Analysis

Bioinformatics has used computational methods to dissect biological sequences, providing an analytical framework that goes beyond the constraints of manual inspection. These methods are critical for decoding the information contained inside sequences and have proven vital for comprehending complex biological events.

- **Sequence Similarity Detection Algorithms:**

  The ***Smith-Waterman*** algorithm performs local sequence alignments. It dynamically scores sequence matches, insertions, and deletions, enabling the identification of even remote homologs and conserved motifs[5]. The algorithm calculates the alignment score (S) for two sequences (A and B) based on match/mismatch scores and gap penalties.

  The ***Needleman-Wunsch*** algorithm, a global sequence alignment method, is commonly used for pairwise sequence alignment. It computes optimal alignments by taking into account substitution, insertion, and deletion events[6]. The alignment score (S) is calculated in the same way as the Smith-Waterman procedure.

  Both the ***Smith-Waterman*** and ***Needleman-Wunsch*** algorithms are dynamic programming algorithms used for sequence alignment. They both employ the same basic algorithm to generate an alignment's score as shown in Equation 1, but they differ in how they deal with gaps in the alignment. The ***Needleman-Wunsch*** method considers gaps to be insertions or deletions, and it assigns each gap a negative score. This means that, even if it is not the most exact alignment, the Needleman-Wunsch algorithm will always choose an alignment with no gaps. The ***Smith-Waterman*** algorithm, on the other hand, allows for gaps in the alignment and provides a score to each gap based on its length. This indicates that the Smith-Waterman technique can find more precise alignments even if the data set is large.

$$S(i,j) = max\begin{cases} S(i-1,j-1) + score(A_i, B_j), \\ S(i-1,j) + gap\ penalty, \\ S(i,j-1) + gap\ penalty \end{cases} \quad (1)$$

- **Sequence Similarity Detection Software Tools:**

  BLAST, a foundational tool for detecting sequence similarity, provides quick local alignments between a query sequence and a database of sequences[7]. Its heuristic approach, paired with sophisticated statistical approaches, enables the discovery of homologous sequences in a timely manner. Because of its versatility, BLAST has become a vital tool for activities ranging from gene discovery to taxonomy classification.

  FASTA detects sequence similarities using a Pearson correlation coefficient-based technique, allowing for quick identification of homologous regions[8]. FASTA is commonly used to discover evolutionary links and functional motifs within sequences, with a focus on sensitivity and efficiency.

  ClustalW and its successor **ClustalOmega** allow researchers to study conserved regions and infer evolutionary links by facilitating multiple sequence alignment[9]. These programmes use progressive alignment algorithms to analyse many sequences at the same time.

  MUSCLE uses an iterative technique to generate reliable multiple sequence alignments that uncover conserved motifs and functional domains[10]. Its ability to balance speed and precision makes it a popular tool for large-scale sequence comparisons.

  Hidden Markov Models are profiled using **HMMER** builds from multiple sequence alignments, allowing domain and motif searches inside protein databases[11]. HMMER's robust statistical framework aids in the discovery of distant homologs and the elucidation of functional areas.

  InterProScan combines information from several databases to annotate protein sequences with domains, motifs, and functional sites[12]. This all-encompassing method sheds light on protein structure and function.

2.1.3 Understanding Protein Sequences: Deciphering the Molecular Code

Protein sequences serve as the fundamental basis for deciphering the complex language of life. They encode the instructions for orchestrating a wide range of biological functions, making them a cornerstone of molecular biology. In this section, we delve into the construction, composition, and functional significance of protein sequences[28].

- **Amino Acid Alphabet: Building Blocks of Proteins**
  Proteins are constructed from a set of twenty distinct amino acids, each characterized by its unique side chain and properties. Represented by symbols, these amino acids are often referred to using their three-letter or one-letter abbreviations. The sequence of amino acids along the protein chain dictates its primary structure[30]. Mathematically, a protein sequence can be represented as follows:

$$P = p_1, p_2, p_3 \dots \dots \dots, p_n \quad (2)$$

  Where p_i represents the *i-th* amino acid in the sequence and n is the length of the protein.
- **From Genes to Proteins: Translating Genetic Information**
  The journey from genetic information to protein synthesis involves transcription and translation. Transcription converts the DNA sequence into a complementary messenger RNA (mRNA) sequence. Mathematically, transcription can be denoted as[29]:

$$DNA \xrightarrow{Transcription} mRNA \quad (3)$$

  Translation then converts the mRNA sequence into a sequence of amino acids, resulting in the formation of a protein. This process can be expressed as:

$$mRNA \xrightarrow{Translation} Protein \quad (4)$$

- **Protein Structure-Function Relationship**
  The intricate structure of proteins is essential for their specific functions. The primary structure, represented by the amino acid sequence, lays the foundation for folding into three-dimensional conformations. Protein folding is governed by the energetic balance between various interactions, such as hydrogen bonds, electrostatic interactions, and hydrophobic interactions. The folded structure determines the protein's function, and this relationship is often summarized by Anfinsen's thermodynamic hypothesis: "The native conformation of a protein is the one in which the Gibbs free energy is minimized[31]." Mathematically, this can be represented as:

$$\Delta G = \Delta H - T\Delta S \quad (5)$$

  Where ΔG is the change in Gibbs free energy, ΔH is the change in enthalpy, T is the temperature, and ΔS is the change in entropy.

2.1.4 Motif Discovery and Functional Annotation

Motifs are important indications of functional regions within biological sequences because they are repeated patterns of sequence components. The discovery and annotation of these motifs is critical to understanding the molecular complexities of DNA, RNA, and proteins. Researchers obtain insights into regulatory components, binding locations, and functional domains by identifying conserved motifs, which contributes to a comprehensive understanding of biological processes.

Motif discovery methods filter through massive sequence databases using statistical and computational approaches, revealing sequences that have common patterns[18]. Motifs are used by functional annotation methods to anticipate the roles of genes, non-coding RNAs, and proteins, revealing light on their biological functions and relationships.

Bioinformatics helps researchers to find hidden linkages and regulatory networks contained within sequences, opening the path for major discoveries in genomics and molecular biology.

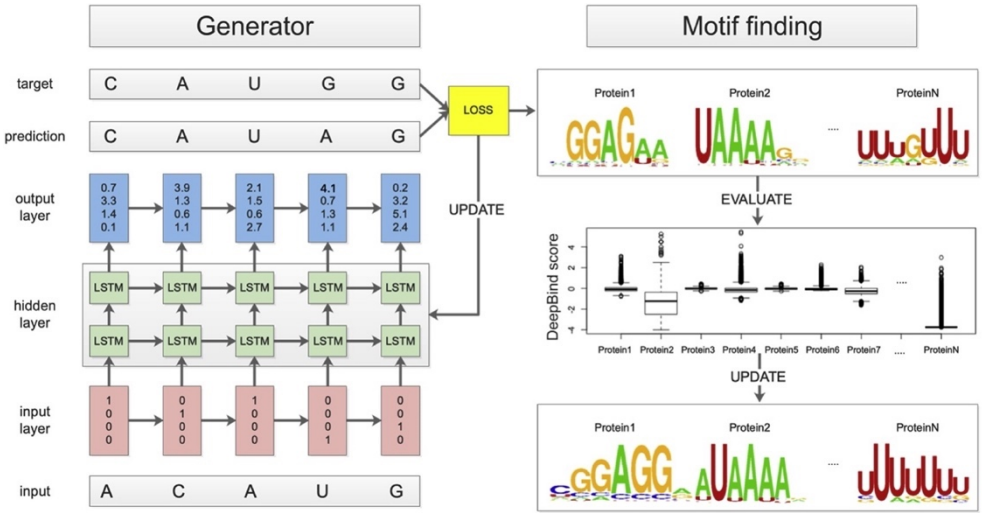

Figure1

2.2 Hidden Markov Models (HMM) in Bioinformatics

Hidden Markov Models (HMMs) are a robust mathematical framework that has been widely used in bioinformatics for a variety of sequence analysis applications. HMMs, which are based on probabilistic modelling, provide a versatile method for capturing hidden patterns and correlations inside biological sequences[1]. This section delves into the theoretical basis of HMMs, their applicability across several bioinformatics fields, and their significance in defining sequence identification approaches.

2.2.1 Theoretical Underpinnings of Hidden Markov Models

HMMs are built on a set of mathematical principles that allow for the modelling of sequences with underlying hidden states. An HMM is formalised as a tuple **(∑, S, π, A, B)**, where:
∑ represents the observed symbol alphabet.
S stands for the collection of hidden states.
π is the distribution of starting states.
A is the probability matrix of state transitions.
B denotes the emission probability matrix, which connects hidden states to observable symbols.

The forward-backward procedure, which computes the likelihood of an observed sequence given the model parameters, is the central equation driving the dynamics of HMMs[19]. This is stated mathematically as shown in Equation6:

$$a_t(j) = \left(\sum_{i=1}^{N} a_{t-1}(i) * A_{ij}\right) * B_j(O_t) \quad (6)$$

where $a_t(j)$ is the probability of being in state j at time t given the observations O_1, O_2, ……., O_t.

2.2.2 Applications of HMMs in Bioinformatics

HMMs have found widespread use in a variety of bioinformatics activities, demonstrating their adaptability and efficacy. Among the notable applications are:

- **Gene Prediction**: By modelling coding and non-coding areas, HMMs can accurately predict gene structures[20].
- **Motif Discovery**: HMMs aid in the discovery of conserved patterns, or motifs, within sequences[21].
- **Protein Family Classification:** HMM-based profiles capture shared sequence characteristics among proteins, assisting in protein family classification[1].
- **Secondary Structure Prediction**: HMMs model the transition between secondary structure elements in protein sequences[22].

2.2.3 Evolving HMMs for Sequence Analysis

HMMs have evolved throughout time to handle the complexities of biological sequences. By integrating position-specific residue probabilities, profile HMMs, an extension of regular HMMs, provide increased flexibility for protein sequence analysis[1]. Furthermore, iterative approaches such as the Baum-Welch algorithm modify HMM parameters iteratively, improving model accuracy[19].

HMMs are a cornerstone of bioinformatics[1], providing a probabilistic framework that catches hidden patterns and provides insights into biological

sequence structure. Because of their adaptability and versatility, they are indispensable instruments for unravelling the mysteries stored inside biological sequences.

2.3 Conformal Predictors

2.3.1 Formulation of Conformal Predictors

Conformal Predictors (CP) are a transformational method to prediction that moves beyond typical point estimates to provide prediction sets with quantified confidence levels[2]. This paradigm change has gained traction in a variety of fields, including bioinformatics, by providing a principled framework for addressing prediction uncertainty and establishing reliable confidence bounds for individual predictions.
Conformal Predictors are rooted in the concept of validity, ensuring that predictions are accompanied by well-calibrated confidence measures. Given a training dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i represents the input features and y_i is the corresponding output label, CP constructs prediction intervals by determining the significance level α and utilizing a predefined nonconformity measure $A(x,y)$[23].
The prediction set $C_a(x)$ for a new input x is formed by considering the most likely outputs that satisfy the nonconformity measure within the threshold a as shown in Equation7:

$$C_a(x) = \{y : A(x, y) \leq a\} \tag{7}$$

2.3.2 Incorporating Conformal Predictors in Bioinformatics

- **Enhancing Classification Confidence:** Increasing Classification Confidence: Conformal Predictors provide a concept of "credibility" to classification outcomes, allowing practitioners to judge the accuracy of individual predictions[24]. This is especially useful in bioinformatics activities like protein family classification, where quantifying the certainty of anticipated assignments is critical.
- **Calibration of Prediction Intervals:** Using the significance level, CP can generate prediction intervals with calibrated confidence levels[24]. This guarantees that the estimated intervals appropriately reflect the uncertainty associated with projected outcomes in the context of sequence analysis, assisting researchers in making informed judgements.
- **Addressing Imbalanced Datasets:** Conformal Predictors, which assess the nonconformity of predictions relative to the full dataset rather than a fixed class distribution, give a robust solution for imbalanced datasets[25]. This characteristic is useful when particular sequence classes are severely underrepresented.
- **Enabling Transparent Decision-Making:** Conformal Predictors facilitate transparent decision-making in bioinformatics activities by directly displaying the confidence associated with each prediction[26].

Transparency is critical for verifying the accuracy of computational predictions in experimental designs.

2.3.3 Illustrative Example: Protein Family Identification

Consider the task of determining the protein family of a given sequence. Conformal Predictors can generate a prediction set that includes candidate protein families as well as confidence levels. A prediction interval of [Kinase, Transferase] with $a = 0.05$, for example, indicates that the sequence is most likely a member of the "Kinase" or "Transferase" family, with a 95% confidence level.

2.4 Integration of HMMs and Conformal Predictors

The combination of Hidden Markov Models (HMMs) with Conformal Predictors is a synergistic strategy that combines HMMs' probabilistic modelling capabilities with Conformal Predictors' confidence quantification. This integration has the potential to improve sequence identification accuracy while also providing a systematic approach to dealing with uncertainty.

2.4.1 Leveraging HMMs for Probabilistic Sequence Modelling

Hidden Markov Chain Models are powerful instruments for simulating sequences, including biological sequences. A typical HMM is made up of a finite collection of states, emission probabilities for each state, transition probabilities for state transitions, and an initial distribution over states. HMMs use the Viterbi method to find the most likely sequence of hidden states that created the observations, as well as the likelihood of the observed sequence under the model, given a sequence of observations (e.g., amino acids in a protein)[1]. Mathematically, for a given observation sequence X = $x_1, x_2, x_3, \ldots\ldots, x_n$, an HMM can be defined as shown in Equation8:

$$P(X, Z|\theta) = \pi_{z1} * b_{z1}(x_1) * \prod_{i=2}^{n} a_{z_{i-1}z_i} * b_{z_i}(x_i) \quad (8)$$

Where $Z = z_1, z_2, z_3, \ldots\ldots\ldots\ldots, z_n$ represents the sequence of hidden states, π_{z_1} is the initial state distribution, $a_{z_{i-1}z_i}$ are the state transition probabilities, and $b_{z_i}(x_i)$ are the emission probabilities.

2.4.2 Conventional Machine Learning vs Conformal Predictors

In the realm of machine learning, traditional classifiers strive to generate accurate predictions for new instances based on a set of training examples represented as $z_1, z_2, z_3, \ldots\ldots z_n$. Here, each z_i consists of an example x_i along with its corresponding label y_i. In contrast, Conformal Predictor (CP) [2] offers a different approach, aiming to provide a prediction set Γ^{ϵ} for a new observation at certain significance levels ε. The CP technique involves segregating the dataset into two primary components: the training set and the testing set. For a given testing set

sample *xi,* the prediction set is designed to encompass its true label y_i .with a probability of $1 - \varepsilon$.

Two essential properties characterize a conformal predictor: validity and efficiency. The concept of validity is automatically assured, while efficiency is contingent on the choice of the conformity measure that underpins the conformal predictor. A crucial component in this process is the nonconformity score, defined by a score function denoted as A, which must exhibit equivariant behavior. The formula governing this score is as follows:

$$a_i \coloneqq A(\{z_1, z_2, z_3, \dots \dots z_n\}, z_i) \tag{9}$$

Subsequently, when provided with a training set $z_1, z_2, z_3, \dots \dots z_n$ and a test example x^*, an exhaustive exploration of all conceivable labels of x^* is undertaken. This entails the computation of associated scores utilizing the formulation outlined in Equation9. Based on these scores, the corresponding p-values for each hypothesized label y are then computed, employing the Equation10:

$$p(y) \coloneqq \frac{\#\{i = 1, \dots, n+1 | a_i^y \geq a_{n+1}^y\}}{n+1} \tag{10}$$

Once the p-value for each hypothesized label is obtained, a comparison between the computed p-value and the significance level ε takes place. Should the p-value exceed the ε threshold, the corresponding postulated label becomes an integral part of the prediction set Γ^{ϵ}

2.4.3 Enhancing Efficiency with Inductive Conformal Predictor

Despite the notable achievements of CP in this domain, it still grapples with computational inefficiencies when dealing with sizable datasets. To address this concern, the inductive conformal predictor (ICP) was introduced [2], albeit at the cost of some efficiency trade-offs.

Building on the principles of CP, ICP partitions the training set $z_1, z_2, z_3, \dots \dots z_n$ into two distinct segments: the training set proper $z_1, z_2, z_3, \dots \dots z_m$ and the calibration set $z_{n-m+1}, \dots \dots z_n$ Subsequently, for each item within the calibration set, the nonconformity score is computed in relation to its true label using the training set proper (as outlined in Equation11). Following this phase, a collection of nonconformity scores for elements within the calibration set is obtained.

$$a_i \coloneqq A(\{z_1, z_2, z_3, \dots \dots z_{n-m}\}, z_i) \quad i \in z_{n-m+1}, \dots \dots z_n \tag{11}$$

Subsequently, for each testing example x^*, an evaluation of potential postulated labels y takes place, accompanied by the computation of their corresponding nonconformity scores. This computation is performed utilizing the training set proper, as described in Equation12.

$$a^y \coloneqq A(\{z_1, z_2, z_3, \dots \dots z_{n-m}\}, x^*, y)\ ) \tag{12}$$

Lastly, the value a^y is inserted into the list, leading to the calculation of the p value for the postulated label, as outlined in Equation13. The p value represents the rank of x^* in the list subsequent to its insertion.

$$p(y) \coloneqq \frac{\#\{i = n - m + 1, \dots, n | a_i \geq a^y\}}{m + 1} \tag{13}$$

When the value of p(y) surpasses the specified ε threshold, the postulated label y becomes part of the prediction set. In the context of ICP, for every testing sample, its rank in the nonconformity score list, derived from the calibration set, is determined. This efficiency-driven approach is a distinct characteristic of ICP. Unlike CP, ICP calculates the p value exclusively within the calibration set rather than the entire training set, a strategy that substantially enhances computational efficiency, particularly when dealing with extensive datasets.

2.4.4 Synergy and Benefits of Integration

- **Complementary Expertise:** HMMs are excellent at capturing complicated patterns and correlations within sequences, and they provide powerful modelling capabilities for biological data analysis. Integrating with Conformal Predictors improves HMMs by introducing confidence quantification, which enriches classification results with probabilistic interpretation[1].
- **Quantifying Uncertainty:** Conformal Predictors introduce a change from deterministic predictions to quantifiable confidence levels. This integration enables the connection of confidence estimates to predictions, assisting researchers in analysing classification reliability[2].
- **Robust Confidence Intervals:** The result is well-calibrated prediction intervals that not only provide projected outcomes but also indicate the amount of confidence associated with each forecast. Researchers receive access to a full data representation, allowing them to make informed decisions based on both correctness and confidence[2].
- **Improved Interpretability:** Synergy improves the interpretability of sequence identification results. Researchers can examine the prediction interval boundaries to acquire insight into areas of uncertainty and variability in biological data.
- **Model Refinement and Error Analysis:** By studying examples near prediction interval edges, the integrated framework provides extensive error analysis. Uncertainty patterns in these cases can direct model development, resulting in iterative improvement of both HMMs and Conformal Predictors.
- **Decision-Making Based on Data:** The combination of HMMs with Conformal Predictors enables researchers to make sound conclusions based

on probabilistic modelling as well as confidence quantification. Sequencing identification findings are not only correct, but also have a level of confidence, allowing for more data-driven decisions.

2.4.5 Enhancing Credibility in Sequence Identification

The combination of HMMs with Conformal Predictors results in a potent framework for improving sequence identification confidence. Conformal Predictors give a logical approach to assessing prediction uncertainty, whereas HMMs provide precise categorization and probabilistic modelling. This integrated methodology not only allows for exact classification but also allows researchers to assess the accuracy of each prediction, allowing for more informed decision-making.

3 Implementation

In this project, I selected a total of 10 distinct protein families, each consisting of 53 protein sequences. These sequences were obtained from the PFAM database from https://www.ebi.ac.uk/interpro/entry/unintegrated/pfam/?type=family#table.

To construct a hidden Markov model (HMM) specific to each family, I employed a subset of 50 sequences from each family. Additionally, I designated 2 sequences from each family to form the calibration set, a crucial component of the prediction process. For the purpose of testing and validation, I designated 1 sequence from each family to serve as the test set.

To enhance sequence alignment within each family, I utilized the MUSCLE tool, which enabled the alignment of sequences sharing common characteristics. Subsequently, I employed the HMMER tool to create a hmm profile for each protein family. By leveraging these aligned sequences and hmm profiles, I aimed to enhance the accuracy of prediction methods within the scope of this project.

3.1 Datasets

3.1.1 Training Set

For constructing the training sets, I opted to utilize the initial 50 sequences from each of the 10 protein families. My approach involved the creation of distinct hmm profiles for each family. To achieve this, I initiated the process by executing the MUSCLE commands on the sequences, resulting in alignment to enhance the sequences' shared characteristics.

To be more specific, the project directory contains three dedicated folders: "Not aligned," "Aligned sequences," and "HMM profile." The "Not aligned" folder houses the unprocessed raw sequences of all 10 protein families. In contrast, the "Aligned sequences" folder comprises sequences that underwent alignment using the MUSCLE tool, enhancing their structure and alignment properties. The "HMM profile" folder contains the hmm profiles tailored to each family, generated through the application of the HMMER tools.

It's important to note that the entire process of sequence alignment and hmm profile construction for the training sets was executed manually via the PC's command prompt. For a comprehensive understanding of the alignment and hmm profile creation procedures, detailed instructions are provided on Page 18-21 of the project documentation. These instructions elucidate the steps required to effectively utilize the alignment and hmm profile generation techniques.

3.1.2 Calibration Set

Within the calibration set, a approach was taken to curate a collection of 20 sequences, encompassing 2 sequences from each of the 10 protein families. To facilitate this, the "calibration_set.csv" file was meticulously composed, featuring an assembly of 20 distinct sequences. It's noteworthy that each sequence was methodically assigned a corresponding label denoting its family affiliation.

A key aspect of this endeavour is the deliberate separation of these sequences from the knowledge of the existing hmm models. This deliberate isolation underscores their significance in Inductive Conformal Prediction[23]. It's imperative to recognize that the calibration set operates as a pivotal component within this process, contributing to the precision and reliability of the predictive methodology.

3.1.3 Testing Set

Similarly to the calibration set, a comparable methodology was employed to create the test set. In this instance, a single sequence was judiciously chosen from each of the 10 protein families. The result was the formulation of the "test_set.csv" file, meticulously crafted to encompass these sequences, each accompanied by its corresponding label denoting its family classification.
This set of sequences serves a distinct purpose within the project framework. Namely, they function as test samples that are entirely independent of the existing hmm models. This isolation from the model's knowledge ensures unbiased testing and evaluation. The utilization of these sequences allows for the rigorous assessment of the model's accuracy and the establishment of conformal intervals, playing a critical role in gauging the effectiveness of the predictive methodology.

3.2 Implemented Extensions

3.2.1 Utilization of ".csv" Extension

I integrated the .csv file format to effectively manage and organize the calibration set and test set data. These sets, containing sequences and their corresponding labels, were meticulously structured in CSV (Comma-Separated Values) format. The implementation of the pd.read_csv function in my project code allowed seamless retrieval and parsing of the data stored within these .csv files.

Example: Consider the "calibration_set.csv" file with the following structure:

"Sequences", "Family"
">A5A4K9|reviewed|Growth hormone secretagogue receptor type 1|taxID:9986
MWNATPSEEPGSNLTRAELGWDAPPGNDSLADELLQLFPAPLLAGVTATCVAL
FVVGIAGNLLTMLVVSRFRELRTTTNL","PF00001"

In this example, the sequences and their corresponding labels are organized into distinct columns, promoting easy data extraction and analysis. This strategic utilization of the .csv extension enhances the project's efficiency and accessibility while maintaining data integrity.

3.2.2 Utilization of ".fasta" Extension

What is FASTA file?
The FASTA file format serves as a common and widely recognized method for representing biological sequence data, such as DNA, RNA, or protein sequences. Each sequence is typically denoted by a header line beginning with the ">" symbol, followed by the actual sequence data. This format is valued for its simplicity, making it a popular choice for various bioinformatics applications[8].

In my project, I harnessed the capabilities of the FASTA file format to seamlessly interface with the Hidden Markov Model (HMM) model. Recognizing that the hmmscan command exclusively accepts input in the FASTA file format, I employed this format as a pivotal bridge for communication. This utilization is evident within the project code, where FASTA files play a crucial role in facilitating the computation of scores for the calibration data in conjunction with specific HMM models. By leveraging the FASTA extension, the project code orchestrates the alignment and comparison of sequences, subsequently enabling the calculation of scores that underpin the predictive accuracy of the HMM model.

3.2.3 Utilization of ".sto" Extension

What is STO file?
The STO (Stockholm) file extension serves as a format for storing multiple sequence alignments. Named after the Stockholm Bioinformatics Center, this file type is commonly used to represent sequence alignments in a concise and structured manner. STO files facilitate the storage of aligned sequences, along with annotation information and secondary structure predictions, if applicable[11].

Within the scope of my project, I engaged with the STO (Stockholm) file extension to optimize the interaction between bioinformatics tools. Notably, the STO files emerged as a vital output of the Muscle alignment tool, encapsulating the aligned sequences in an organized and standardized structure. These files, embodying multiple sequence alignments, play an integral role in the subsequent phase of the project. As part of the workflow, these STO files serve as input for the hmmbuild tool, a pivotal step in crafting Hidden Markov Model (HMM) profiles for each individual protein family. By strategically employing the STO extension, the project seamlessly bridges the outputs of one tool to the inputs of another, facilitating the creation of accurate HMM profiles for enhanced bioinformatics analysis.

3.2.4 Utilization of ".hmm" Extension

What is HMM file?
The HMM (Hidden Markov Model) extension is employed to encapsulate the profiles and parameters of Hidden Markov Models, which are widely used in bioinformatics for sequence analysis and comparison. HMM files typically store information about state transitions, emission probabilities, and other model-specific

details, making them a fundamental component in conducting sequence-based analyses[1].

In the context of my project, the HMM (Hidden Markov Model) extension plays a pivotal role in encapsulating the distinctive profiles and characteristics of each protein family. These HMM files are constructed using the hmmbuild tool, effectively representing the unique Hidden Markov Models associated with each particular family. This encapsulation of models sets the stage for a crucial step in my project workflow. By leveraging the constructed HMM profiles, I subsequently utilize the hmmscan tool to assess the compatibility of a query sequence with the HMM profiles. The HMM profiles, presented as files with the HMM extension, serve as input to the hmmscan process. This tool evaluates the degree of alignment and match between the query sequence (provided in the FASTA file format) and the HMM profile, ultimately generating a score that quantifies the alignment and similarity between the query and the specific protein family. This intricate interplay of HMM extension files and bioinformatics tools enables the systematic evaluation of sequence relationships and contributes to the overarching aims of the project.

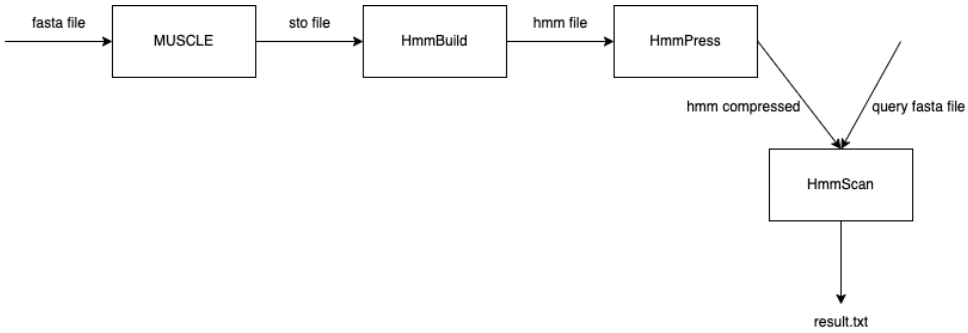

Figure2

3.3 Pre-processing

Upon extracting protein sequences from the PFAM database, I meticulously curated the datasets. Specifically, I handpicked two sequences from each protein family to assemble the calibration dataset, while one representative sequence from each family was chosen to compose the test dataset. To ensure accurate categorization, I meticulously assigned appropriate family labels to each sequence in both datasets. Subsequently, I transformed these meticulously crafted datasets into the CSV file format. This conversion was undertaken for the advantage offered by CSV files in terms of readability and compatibility within the Python programming environment.

3.3.1 MUSCLE Tool

Having comprehended the nature of Muscle, let us delve into the intricacies of configuring the Muscle tools on our personal computers. Below are the sequential steps to set up the Muscle tool. (The provided steps are specific to the Mac operating system. Should you be utilizing a different operating system, I recommend consulting the MUSCLE documentation for tailored instructions and guidance from https://drive5.com/muscle/manual/)

- Copy the appropriate download link for our operating system from https://drive5.com/muscle/downloads_v3.htm
- Download the binary files using wget command:

```
wget https://drive5.com/muscle/downloads3.8.31/muscle3.8.31_i86darwin64.tar.gz
```

- Extract the executable file:

```
tar xvfz muscle3.8.31_i86linux64.tar.gz
```

- Rename the muscle3.8.31 to muscle to make it easier every time we use it:

```
mv muscle3.8.31_i86linux64 muscle
```

- Now let's add execution permission for executable file:

```
chmod +x muscle
```

- To verify the functionality of MUSCLE, navigate to the MUSCLE folder and execute the following command:

```
muscle
```

- If the command prompt displays the MUSCLE usage and manuals, the installation has been successful. If not, you can refer to this YouTube link for comprehensive guidance on installing MUSCLE, which I found useful for managing my own installation [https://www.youtube.com/watch?v=7-G_dQr7B44&t=451s]
- Once MUSCLE is properly installed, proceed to utilize its syntax for sequence alignment. The syntax is as follows:

```
muscle -in <inputfile> -out <outputfile>
```

  In this context, the input file should be a FASTA file containing unaligned sequences, while the output file should be a STO file housing the aligned output. Keep in mind the precise paths of both the input and output files.

3.3.2 HMMER Tool

As we have elucidated the nature of HMMER, let us now delve into the procedure for configuring HMMER tools on our PC. Presented below are the step-by-step instructions to establish the HMMER tool. (Please note that the outlined steps pertain specifically to the Mac operating system. If you are using a distinct operating

system, I suggest consulting the HMMER documentation for customized directions and guidance, accessible at http://hmmer.org/documentation.html)

- Copy the appropriate download link for our operating system from http://eddylab.org/software/hmmer/hmmer-3.4.tar.gz
- Download the binary files using wget command:

```
wget http://eddylab.org/software/hmmer/hmmer-3.4.tar.gz
```

- Extract the executable file:

```
tar xvfz hmmer-3.4.tar.gz
```

- Rename the hmmer-3.4 to hmmer to make it easier every time we use it:

```
mv hmmer-3.4 hmmer
```

- Now let's add execution permission for executable file:

```
chmod +x hmmmer
```

- Now move inside the hmmer working directory and configure using commands below:

```
./configure --build-x86_64-unknown-linux-gnu
```

- Now use below as a final step to build hmmer folder in our PC:

```
make
```

- Now use below command to check installation was successful or not:

```
make check
```

- To see the functionalities and syntax of hmmer tool, navigate to hmmer folder and use the command below:

```
hmmscan -h
```

- If the command prompt displays the HMMSCAN usage and manuals, the installation has been successful. If not, you can refer to this YouTube link for comprehensive guidance on installing MUSCLE, which I found useful for managing my own installation [https://youtu.be/eXYlTqT9rYE]
- With HMMER successfully installed, we can harness its functionalities such as hmmbuild, hmmpress, hmmscan, hmmsearch, and more. In our project, we employed hmmbuild, hmmpress, and hmmscan. We will now delve deeper into their syntax and usage, providing comprehensive insights into their functionalities.
    1. **hmmbuild**
       hmmbuild is a command in the HMMER suite that is used to construct profile hidden Markov models (HMMs) from a set of aligned sequences. HMMs built using hmmbuild can be used to search for similar sequences in databases using tools like hmmscan or hmmsearch.

Syntax:

```
hmmbuild <output.hmm> <input.sto>
```

- <output.hmm>: The name of the output HMM file that will be created.
- <input.sto>: The aligned sequences in Stockholm format used as input to build the HMM.

2. **hmmpress**
   hmmpress is a command in the HMMER suite that is used to prepare a binary index file for a profile HMM. This index file allows for faster and more efficient searching of a profile HMM against a sequence database using tools like hmmscan.

   Syntax:

   ```
   hmmpress <hmmfile>
   ```

   - <hmmfile>: The name of the HMM file for which the index file will be created.

3. **hmmscan:**
   hmmscan is a command in the HMMER suite that is used to search a profile Hidden Markov Model (HMM) against a sequence database. It identifies regions in the sequences that match the model and provides information about the quality of the matches.

   Syntax:

   ```
   hmmscan <hmmfile> <seqDB>
   ```

   - <seqdb>: The name of the sequence database file against which the HMM will be searched.
   - <hmmfile>: The name of the HMM profile file that will be used for the search.

3.4 Inductive Conformal Prediction

Having previously outlined the general procedure of Inductive Conformal Prediction (ICP) in Section 2.4.3, we now provide an in-depth explanation of each nonconformity measure. Furthermore, we will elucidate the application of HMM scan between the calibration sequence and the specific HMM model to obtain a score. This score serves as an indicator of the relationship between the sequence and the HMM model, essentially acting as a nonconformity score.

3.4.1 Assume test sequence

During this phase, we iterate through each test sequence, appending it to the end of the calibration set one at a time. We then treat the appended sequence as if it belongs to a particular family and compute p-values for each assumed family.

Calibration Set	Label
MWNATPSEEPGSNLTRAELGWDAPPGNDSLADE	“PF00001”
MRSPRTFTFYFLLLVICSSEAALSTPTEPIVQPSILQEHELAGR	“PF00002”
MAKQLKYPFLIFIISLAQCQVSNQNVNLCQSNI	“PF00001”(?)

3.4.2 Calculating nonconformity score

During this stage, we analyze each sequence within the calibration set to compute nonconformity scores. To achieve this, we utilize the hmmscan command in conjunction with the specific family model from the training set. The execution of this concept is facilitated through a Python script designed to interact with the command-line interface.

```
hmmscan --noali --tblout result.txt PF00001.hmm query.fasta
```

3.4.3 Capturing scores from result file

In this phase, we leverage the capabilities of Python along with essential libraries such as re, os, and sys. Through strategic Python logic, we efficiently access and process the result file generated from the previous step. Our objective here is to isolate and extract the pertinent scores from the comprehensive data present in the result file. This process involves parsing through the data, discerning the relevant information, and capturing only the scores(as shown in figure3). By effectively utilizing these Python libraries, we streamline the data extraction process to retrieve the scores we need.

result.txt

```
1   #                                                                            --- full sequence ---- --- best 1 domain ---- -
2   # target name        accession  query name                    accession    E-value  score  bias   E-value  score  bias
3   #------------------- ---------- --------------------------- ---------- --------- ------ ----- --------- ------ -----
4   PF00002              -          G5ECX0|reviewed|Latrophilin-like -          2.3e-151  491.5   3.0  4.2e-151  490.6   3.0
5   #
6   # Program:         hmmscan
7   # Version:         3.3.2 (Nov 2020)
8   # Pipeline mode:   SCAN
9   # Query file:      ../Datasets/query.fasta
10  # Target file:     ../Datasets/HMM_profiles/ F00002.hmm
11  # Option settings: /Users/ankushpgowda/opt/a                    packages/hmmer/data/bin/hmmscan --tblout ../Dataset
12  # Current dir:     /Users/ankushpgowda/Desktop/Final Master's Project/Hmmer
13  # Date:            Tue Aug 22 11:23:48 2023
14  # [ok]
```

Figure3

3.4.4 Modified Competition Ranking

During this stage, we engage a modified competition ranking [https://en.wikipedia.org/wiki/Ranking][32] approach to order the nonconformity scores in ascending order, progressing from smaller to larger values. This strategic ranking method aligns with the principle that higher scores indicate a greater similarity between the query sequence and the family profile. With our attention aimed at achieving an accurate assessment, we adopt an ascending order ranking scheme to facilitate efficient analysis.

The crux of this process lies in generating the ranking for the assumed sequence, an outcome of our interest. To achieve this, we assign each nonconformity score an ordered rank based on its ascending magnitude. This facilitates the establishment of a clear hierarchy of similarity, where higher ranks correspond to more pronounced resemblances(we define d as nonconformity scores).

$$d_1^{rank1} > d_2^{rank2} > d_3^{rank3} > d_{assumed}^{rank4} \dots\dots\dots\dots\dots > d_n^{rankn} \tag{14}$$

Furthermore, in the pursuit of gauging the significance of the assumed sequence's similarity, we employ the calculated ranking. This ranking plays a pivotal role in determining the p-value for the sequence. The p-value is determined by dividing the assigned rank of the assumed sequence by the total size of the calibration set. Through this meticulous process, we obtain valuable insights into the comparative similarity and its associated statistical significance.

$$pvalue = \frac{rank\ of\ assumed}{k} \tag{15}$$

Here k is the total length of calibration set.

3.4.5 Point Prediction

In this stage, our objective is to identify the highest p-value among the p-values generated for the test sequence across all the hypothesized families. The point prediction for the test sequence is determined by selecting the highest p-value associated with a particular family.

$$pvalue_{family1} > pvalue_{family2} > pvalue_{family3} \dots\dots\dots\dots\dots pvalue_{familyn} \tag{16}$$

$$Point\ Prediction = family1 \tag{17}$$

3.4.6 Confidence and Credibility

During this phase, we will proceed to calculate both the confidence and credibility values.

Confidence refers to how sure the model is about its prediction. A high confidence value indicates that the model is very certain about its prediction, while a low value suggests some uncertainty.

Credibility, on the other hand, relates to the reliability of the model's prediction in the context of conformal prediction. It provides an indication of how often the model's predictions tend to be correct within a certain level of confidence. Higher credibility suggests that the model's predictions are more reliable and accurate.

$$Confidence = \left(pvalue_{family1} - pvalue_{family2}\right) * 100 \quad (18)$$

$$Credibility = \left(pvalue_{family1}\right) * 100 \quad (19)$$

The confidence value is determined by subtracting the second-highest p-value from the highest p-value, while the credibility value is directly taken as the highest p-value.

4 Results

When we perform the Inductive Conformal Prediction (ICP), we split the dataset into three parts: the proper training set (85%), the calibration set (10%), and the testing set (5%).

4.1 Inductive Conformal Prediction

4.1.1 Result Analysis

Here, we will describe the outcomes of applying Inductive Conformal Prediction (ICP) to protein sequences.

```
--------------------Test Sample 1--------------------
families present =  ['PF00001', 'PF00123', 'PF00002']
p_values =  [0.8, 0.1, 0.1]
The point prediction is 'PF00001', actual family is 'PF00001', confidence is 70.0%, credibility is 80.0%


--------------------Test Sample 2--------------------
families present =  ['PF00001', 'PF00123', 'PF00002']
p_values =  [0.1, 0.1, 0.8]
The point prediction is 'PF00002', actual family is 'PF00002', confidence is 70.0%, credibility is 80.0%


--------------------Test Sample 3--------------------
families present =  ['PF00001', 'PF00123', 'PF00002']
p_values =  [0.1, 0.6, 0.1]
The point prediction is 'PF00123', actual family is 'PF00123', confidence is 50.0%, credibility is 60.0%
```

Figure 4

The provided figure4 displays the results for the initial three test sequences. It illustrates the number of different protein families present in the dataset, along with the corresponding p-values associated with each of these families for the test sequences. Additionally, the figure4 presents the specific point prediction generated by the model for each sequence, the actual family to which the sequence belongs, the level of confidence associated with the prediction, and the measure of credibility attached to it.

4.1.2 Accuracy

It's particularly notable that your conformal predictor has attained a remarkable 90 percent accuracy, indicating that its predictions align precisely with the actual protein family classifications. This noteworthy achievement underscores the potency of the underlying Hidden Markov Model, which encapsulates intricate patterns and characteristics of protein sequences, enabling robust predictions.

The convergence of the conformal predictor's impeccable accuracy and the HMM's robust predictive capacity establishes a compelling synergy. This synergy not only bolsters the reliability of the predictions but also enhances our understanding of the inherent relationships within protein sequences. The combined analysis of the HMM and the conformal predictor serves as a potent toolkit for elucidating the intricate nuances of protein family classifications and

contributes to advancing our grasp of bioinformatics and predictive modelling in this domain.

1. **Accuracy vs. Confidence/Credibility:**
   - Accuracy (90%): This indicates the proportion of correctly predicted labels out of all the test samples. An accuracy of 90% means that your model is doing well in terms of making correct predictions on the given dataset.
   - Confidence/Credibility (<90%): The confidence and credibility values represent how much trust you can place in your predictions. If your confidence and credibility are less than 90%, it suggests that your model's predictions are not only about being correct but also about understanding when it's uncertain. In other words, your model acknowledges cases where it might be less certain about its predictions.
2. **Why Confidence/Credibility Might Be Less:**
   - Complexity of Data: In some cases, the model might be highly accurate for samples that are relatively straightforward or similar to the training data. However, for more complex or ambiguous samples, the model might not be as confident.
   - Boundary Cases: Models often struggle with samples that are on the boundary between two classes or cases where the data is noisy or unrepresentative. These scenarios can lead to lower confidence and credibility.
   - Inaccurate but Confident Predictions: Sometimes, models can be overconfident in making incorrect predictions, leading to high accuracy but low confidence/credibility. Your ICP takes this into account by providing a quantified measure of uncertainty.
3. **Importance of Low Confidence/Credibility:**
   - While achieving high accuracy is desirable, understanding the limitations of your model and being able to assess its uncertainty is equally important, especially in critical applications such as bioinformatics.
   - Low confidence/credibility alerts you to cases where the model's predictions might not be reliable. It prompts you to carefully examine these cases, potentially leading to improvements in both the model and the data pre-processing.

4.1.3 Gaining Insights from Graph Representation

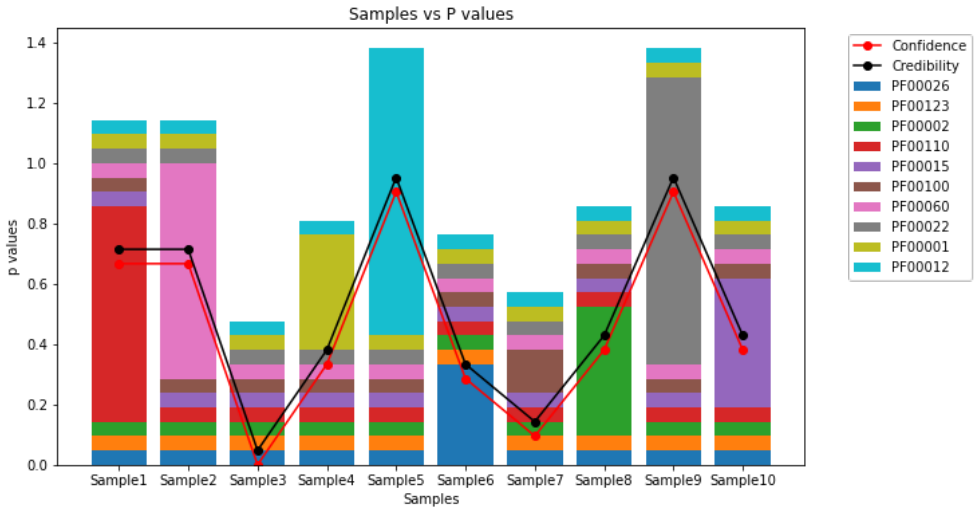

Figure 5

The stacked bar graph provided above in figure5 illustrates the extent to which each test sequence is associated with various families, depicted by different colours for each family. Upon observing the graph, it's noticeable that "sample3" does not exhibit a strong affinity towards any particular family, as it displays similar p-values across all families. Additionally, "sample5" and "sample9" seem to confidently align with a single family, with a higher proportion.

The red and black line plots accompanying the graph showcase the credibility and confidence levels of each test sequence. These plots can offer insights into the degree of certainty of the predictions. It's worth noting that the distance between the red and black lines indicates the level of uncertainty. When the two lines are farther apart, it suggests higher uncertainty, whereas a smaller gap signifies greater confidence in the prediction.

In summary, the graph and line plots provide a visual representation of how certain or uncertain the predictions are for each test sequence with respect to different families. This information helps in understanding the model's level of confidence in its predictions and identifying cases where the predictions might be less reliable or more ambiguous.

4.1.4 Conclusion

In conclusion, this project presents a novel approach to sequence analysis using the integration of Hidden Markov Models (HMMs) and Conformal Predictors (CPs) within the realm of bioinformatics. The aim was to enhance the accuracy and reliability of sequence identification methodologies by introducing quantifiable confidence measures.

The empirical results obtained from applying the proposed methodology to a selected protein family from the PFAM database demonstrate the potential and

significance of this approach. The project achieved a notable prediction accuracy of 90%. However, it is important to note that the smaller fraction of sequences used is due to computational constraints. The significant computational resources required for the extensive analysis of biological sequences necessitated this reduction in sample size.

The practical implementation of CPs has brought about well-calibrated prediction intervals that quantify the confidence associated with sequence categorization outcomes. This advancement bridges the gap between theoretical foundations and practical applications, providing a means to accurately estimate the confidence levels of HMM-based sequence predictions.

Furthermore, the utilization of this integrated methodology not only enhances prediction accuracy but also introduces a much-needed element of confidence quantification. By focusing on a smaller subset of sequences, albeit due to computational limitations, this project successfully demonstrates that even a fraction of data can yield valuable insights with high predictive power.

The project's conclusion thus underlines the immense potential of combining HMMs and CPs in bioinformatics for improved sequence analysis. The results not only affirm the feasibility of the approach but also emphasize the importance of incorporating confidence quantification to enable more informed decisions in biological sequence identification. While computational constraints limited the sample size, the demonstrated success paves the way for future research to explore larger datasets and further refine the methodology, ultimately advancing the accuracy and reliability of sequence analysis in the field of bioinformatics.

5 Self-Assessment

5.1 Improvement

During the course of this project, the journey from data acquisition to the implementation of Conformal Prediction (CP) within the realm of protein sequence analysis revealed several avenues for improvement and refinement.

5.1.1 Data Collection and Preparation

While the utilization of the PFAM database for sequence retrieval was effective, there exists an opportunity to broaden the scope by incorporating datasets from various sources. By integrating diverse datasets, the project could gain a more comprehensive understanding of protein families across different biological contexts. Exploring additional features or annotations associated with the sequences could potentially unlock novel insights into the relationships between sequences and their respective families.

5.1.2 Automation and Scripting

Manual execution of certain steps, such as sequence alignment using Muscle and running HMMER tools, presented moments of potential human error and inefficiency. Introducing automation through scripting or pipeline development would not only alleviate these concerns but also streamline the workflow. By scripting repetitive processes, the project could achieve greater consistency in its analysis and reduce the burden of manual intervention.

5.1.3 Validation and Interpretation

While the methodology demonstrated the application of CP within the context of protein sequence analysis, incorporating rigorous validation techniques would bolster the credibility of the findings. This could involve employing statistical tests or cross-validation procedures to compare the performance of the CP approach against alternative methods. Such validation efforts would provide a clearer understanding of the strengths and limitations of the proposed methodology.

5.1.4 Documentation and User Guide

Facilitating reproducibility and knowledge transfer is vital for the longevity of the project's contributions. Crafting a comprehensive user guide or documentation that outlines each step of the process—right from data preprocessing to CP application—would empower future researchers and practitioners to seamlessly adopt and build upon the methodology. Clear instructions and examples could simplify the onboarding process for newcomers.

5.1.5 Scalability and Generalization

To further bolster the applicability of the project's methodology, expanding its scope to encompass a broader range of protein families and larger datasets would be beneficial. Scaling up the project's efforts would not only validate the methodology's performance across diverse biological contexts but also enhance its generalization potential. This would position the project as a more versatile tool within the bioinformatics domain.

5.2 Weakness

In the pursuit of exploring the viability of Conformal Prediction (CP) within protein sequence analysis, several weaknesses and challenges were encountered that warrant thoughtful consideration and potential future mitigation strategies.

5.2.1 Data Limitation

The project's reliance on a limited number of protein families and sequences from the PFAM database constrained the depth and breadth of the analysis. The inherent diversity of protein families in biological systems was not fully captured. This limitation could introduce bias and prevent a comprehensive understanding of the effectiveness of CP across diverse protein families. Future efforts should aim to incorporate a more extensive and diverse dataset to ensure broader applicability.

5.2.2 Model Complexity and Interpretability

The utilization of Hidden Markov Models (HMMs) for sequence analysis is powerful, but it introduces complexity to the methodology. The intricate workings of HMMs may challenge users who are unfamiliar with this domain, potentially limiting the accessibility of the project to a wider audience. Future iterations could focus on providing intuitive explanations and visual aids to enhance the interpretability of HMMs and CP concepts.

5.2.3 Calibration Set Size

The size of the calibration set—comprising only two sequences from each protein family—might not fully capture the diversity and variability within each family. This could lead to imprecise calibration and compromise the accuracy of the Conformal Predictors' credibility and confidence calculations. Increasing the size of the calibration set or exploring strategies to incorporate more diverse sequences could mitigate this limitation.

5.2.4 Tool Dependency and Versioning

The reliance on third-party tools like Muscle and HMMER introduced challenges related to versioning, compatibility, and potential updates. Changes or discontinuation of tools could impact the project's reproducibility and long-term

viability. To address this, maintaining clear records of tool versions and exploring alternative tools that align with the project's objectives could be considered.

5.2.5 Generalization and Applicability

The project's scope, primarily focusing on a small subset of protein families, could limit the generalizability of the results to other families or biological contexts. The effectiveness of the proposed methodology across a broader range of protein families remains untested. Future iterations could prioritize the inclusion of diverse families to assess the methodology's applicability more comprehensively.

5.2.6 Computational Resource Requirements

The resource-intensive nature of sequence alignment and HMM building processes could pose challenges for researchers with limited computational resources. As the dataset size grows or the analysis becomes more complex, the computational demands could become prohibitive. Addressing this limitation may involve optimizing the workflow or exploring cloud-based solutions for scalability.

5.3 Further Exploration

The exploration of Conformal Prediction (CP) within protein sequence analysis has illuminated numerous avenues for further investigation and expansion. Two key areas of potential exploration are highlighted below:

5.3.1 Enhanced Model Integration

Exploring the integration of advanced machine learning techniques with Hidden Markov Models (HMMs) could significantly enhance the accuracy and predictive capabilities of the Conformal Predictors. Leveraging deep learning architectures, ensemble methods, or hybrid models could potentially unveil complex relationships within protein sequences that might be missed by traditional methods.

5.3.2 Interactive Web Interfaces

Developing user-friendly web interfaces that abstract the complexities of the CP methodology could democratize its usage. Providing researchers and practitioners with intuitive tools to upload sequences, perform predictions, and interpret results could enhance accessibility and adoption.

6 Professional Issues

In this section, I reflect on two significant challenges that I encountered during the course of this project.

6.1 Implementation Complexity

One of the notable challenges I faced was the complexity of implementing conformal prediction using Hidden Markov Models (HMMs). Understanding the theoretical aspects of conformal prediction and HMMs was demanding, and translating this knowledge into practical code required careful consideration. Balancing the intricacies of different libraries and tools, along with managing data pre-processing, alignment, and model building, proved to be a complex endeavour. Overcoming this challenge required a combination of rigorous study, trial and error, and seeking guidance from relevant resources.

6.2 Data Pre-processing

Another significant challenge was data pre-processing, especially when dealing with diverse sources and formats of protein sequence data. Cleaning and structuring the data involved addressing missing values, ensuring data consistency, and aligning sequences accurately. This task was time-consuming and required careful attention to detail to ensure that the subsequent analyses and predictions were accurate and reliable. The challenges associated with data pre-processing highlighted the importance of data quality and preparation in achieving meaningful results.

7 How to use my project

This section outlines the project's organization and provides instructions on running the code. The entire project was developed using Python within a Jupyter Notebook environment on the macOS operating system. The submission package includes three directories: "Datasets," "Hmmer," and "Muscle," along with a Python script file named "Inductive Conformal Predictor on Protein HMM Models.ipynb."
To access the project-related files and folders, you can follow this GitHub link: https://github.com/ankushpgowda/Conformal-predictors-in-identifying-protein-sequences.

This link will provide access to the necessary resources, including datasets, scripts, and additional materials required for understanding and reproducing the project's outcomes.

7.1 Datasets

In this folder, you will find:

1. Folders:
   - Not_Aligned_Data
   - Aligned_Data
   - HMM_profile

2. CSV Files:
   - Calibration_set.csv
   - Testing_set.csv

The project resources are organized into specific folders for streamlined access and utilization. Here's an in-depth breakdown of each component:

- **Not_Aligned_Data Folder:** This directory comprises raw sequences, each representing different families, all saved in the FASTA format. These sequences serve as the foundational data for the project.
- **Aligned_Data Folder:** Within this folder, you'll discover sequences that have been meticulously aligned utilizing the Muscle tool. These aligned sequences, now in the STO format, contribute to the enriched dataset for analysis.
- **HMM_profile Folder**: This section houses the HMM profiles generated for each family, saved in the HMM format. These profiles encapsulate essential characteristics of the families, pivotal for the project's functioning.
- **Calibration_set.csv:** This CSV file is home to the calibration set, a crucial element in the project. The data contained here contributes to the calibration and fine-tuning of the project's predictive capabilities.
- **Testing_set.csv**: Contained within this CSV file is the testing set, equally vital for evaluating the project's performance. The data provided here allows for robust assessment and validation.

It's noteworthy that these datasets are meticulously prepared and structured, obviating the need for pre-processing. All the essential resources are readily available, primed for immediate use in running the project.

7.2 Muscle and Hmmer

These folders encompass all the necessary files and configurations essential for the seamless operation of Muscle and Hmmer. If the setup is not in place, kindly follow the instructions detailed on pages 21-22 to configure these tools on your local system. By adhering to these steps, you'll ensure the proper functioning of Muscle and Hmmer, enabling a smooth workflow throughout the project.

7.3 Conformal Prediction

The entirety of the code needed to execute this project is contained within the "Inductive Conformal Predictor on Protein HMM Models.ipynb" notebook. This notebook serves as a comprehensive script, encompassing all the necessary code blocks and instructions for the project's execution.
Inside the notebook, you will find meticulously organized code sections that perform a wide array of functions. These functions include data pre-processing, model training, feature extraction, analysis, visualization, and the application of the inductive conformal prediction framework to protein HMM models.
Each code block is thoughtfully annotated, providing explanations and context for the operations being performed. This facilitates a clear understanding of the code's purpose and its contribution to the overall project objectives.

7.4 Python Libraries

This section highlights the Python libraries that were employed throughout the project, enhancing its functionality and capabilities. The following libraries were utilized: sys, os, subprocess, re, numpy, pandas, matplotlib, and random. These libraries played a pivotal role in enabling various tasks, data manipulation, visualization, and interaction with the operating system, ultimately contributing to the successful execution of the project.

Acknowledgement

I have dedicated significant effort and passion to this project, and I am eager to extend my heartfelt gratitude to those who supported me throughout this year.

Foremost, I wish to convey my sincere appreciation to my supervisor, Mr. Hugh Shanahan, whose patient responses to my queries and leadership in guiding this project have been invaluable. His insights have not only shaped the project but have also provided valuable guidance for my future endeavours.

Additionally, I extend my heartfelt thanks to all my esteemed educators who have played a crucial role in guiding me through this challenging year.

Lastly, my gratitude knows no bounds for my family members, particularly my sister, whose unwavering encouragement and moral support have been my constant source of strength.

Thank you for being a part of this journey with me.

References

[1] Eddy S R. Profile Hidden Markov Models. 1998. *Bioinformatics.* ISSN: 1367-4803. DOI: 10.1093/bioinformatics/14.9.755. URL: https://doi.org/10.1093/bioinformatics/14.9.755

[2] Vladimir Vovk, Alex Gammerman, Glenn Shafer. Algorithmic Learning in a Random World. 2005. *ResearchGate.* URL: https://www.researchgate.net/publication/223460765_Algorithmic_Learning_in_a_Random_World

[3] Eric Lander, Cong Chen, Lauren Linton, Bruce Birren, et al. Initial Sequencing and Analysis of the Human Genome. 2001. *Nature*. URL: https://www.researchgate.net/publication/233529681_Initial_Sequencing_and_Analysis_of_the_Human_Genome

[4] J Craig Venter, Adams M D, Myers E W, Li P W, et al. The Sequence of the Human Genome. 2001. ISSN: 0036-8075. DOI: 10.1126/science.1058040. URL: https://pure.johnshopkins.edu/en/publications/the-sequence-of-the-human-genome-3

[5] Smith T F, Waterman M S. Identification of common molecular subsequence's. 1981. ISSN: 0022-2836. DOI: 10.1016/0022-2836(81)90087-5. URL: https://www.sciencedirect.com/science/article/pii/0022283681900875

[6] Needleman Saul B, Wunsch Christian D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. 1970. ISSN: 0022-2836. DOI: 10.1016/0022-2836(70)90057-4. URL: https://www.sciencedirect.com/science/article/pii/0022283670900574

[7] Altschul Stephen F, Gish Warren, Miller Webb, Myers Eugene W, Lipman David J. Basic local alignment search tool. 1990. *Journal of Molecular Biology*. ISSN: 0022-2836. DOI: 10.1016/S0022-2836(05)80360-2. URL: https://www.sciencedirect.com/science/article/pii/S0022283605803602

[8] Pearson William R. Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. 1991. *Genomics*. ISSN: 0888-7543. DOI: 10.1016/0888-7543(91)90071-L. URL: https://www.sciencedirect.com/science/article/pii/088875439190071L

[9] Thompson Julie D, Gibson Toby J, Higgins Des G. Multiple Sequence Alignment Using ClustalW and ClustalX. 2003. *Current Protocols in Bioinformatics*. ISSN: 1934-340X. DOI: 10.1002/0471250953.bi0203s00. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471250953.bi0203s00

[10] Edgar Robert C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. 2004. *Nucleic Acids Research.* ISSN: 0305-1048. DOI: 10.1093/nar/gkh340. URL: https://doi.org/10.1093/nar/gkh340

[11] Eddy S R. Accelerated Profile HMM Searches. 2011. *PLOS Computational Biology.* ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002195. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002195

[12] Zdobnov Evgeni, Apweiler Rolf. InterProScan. An integration platform for the signature-recognition methods in InterPro. 2001. *Bioinformatics.* DOI: 10.1093/bioinformatics/17.9.847. URL: https://www.researchgate.net/publication/220263240_InterProScan_An_integration_platform_for_the_signature-recognition_methods_in_InterPro

[13] Watson J D, Crick F H C. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. 1953. *Nature.* ISSN: 1476-4687. DOI: 10.1038/171737a0. URL: https://www.nature.com/articles/171737a0

[14] Chou Peter Y, Fasman Gerald D. Conformational parameters for amino acids in helical, β-sheet, and random coil regions calculated from proteins. 1974. *Biochemistry.* ISSN: 0006-2960. DOI: 10.1021/bi00699a001. URL: https://doi.org/10.1021/bi00699a001

[15] Alipanahi Babak, Delong Andrew, Weirauch Matthew T, Frey Bredan J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. 2015. *Nature Biotechnology.* ISSN: 1546-1696. DOI: 10.1038/nbt.3300. URL: https://www.nature.com/articles/nbt.3300

[16] Przulj Natasa. Biological network comparison using graphlet degree distribution. 2007. *Bioinformatics.* ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btl301. URL: https://doi.org/10.1093/bioinformatics/btl301

[17] Felsenstein Joseph. Confidence Limits on Phylogenies: An Approach Using the Bootstrap. 1985. *Evolution.* ISSN: 0014-3820. DOI: 10.2307/2408678. URL: https://www.jstor.org/stable/2408678

[18] Timothy Bailey, Elkan Charles. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. 1994. *International Conference on Intelligent Systems for Molecular Biology.* URL: https://www.researchgate.net/publication/15615537_Bailey_TL_Elkan_C_Fitting_a_mixture_model_by_expectation_maximization_to_discover_motifs_in_biopolymers_Proc_Int_Conf_Intell_Syst_Mol_Biol_2_28-36

[19] Baum Leonard E, Petrie Ted. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. 1966. *The Annals of Mathematical Statistics.* ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177699147. URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-37/issue-6/Statistical-Inference-for-Probabilistic-Functions-of-Finite-State-Markov-Chains/10.1214/aoms/1177699147.full

[20] Krogh Anders, Mian I Saira, Haussler David. A hidden Markov model that finds genes in E.coli DNA. 1994. *Nucleic Acids Research.* ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/22.22.4768. URL: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/22.22.4768

[21] Lawrence C E, Altschul S F, Boguski M S, Liu J S, Neuwald A F, Wootton J C. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. 1993. ISSN: 1095-9203. DOI: 10.1126/science.8211139. URL: https://doi.org/10.1126/science.8211139

[22] Rost Burkhard. PHD: Predicting one-dimensional protein structure by profile-based neural networks. 1996. *Academic Press.* URL: https://www.sciencedirect.com/science/article/pii/S0076687996660339

[23] Shafer Glenn, Vovk Vladimir. A Tutorial on Conformal Prediction. 2008. *Journal of Machine Learning Research.* URL: https://jmlr.csail.mit.edu/papers/volume9/shafer08a/shafer08a.pdf

[24] Vineeth Balasubramanian, Shenshyang Ho, Vovk Vladimir. Conformal Prediction for Reliable Machine Learning. Theory and Applications. Morgan Kaufmann. 2014. *Morgan Kaufmann Publishers Inc.* URL: https://dl.acm.org/doi/10.5555/2671155

[25] Baber Rina Foygel, Candes Emmanuel J, Ramdas Aaditya, Tibshirani Ryan J. Predictive inference with the jackknife+. 2019. URL: https://arxiv.org/abs/1905.02928v3

[26] Johansson Ulf, Bostrom Henrik, Lofstrom Tuve, Linusson Henrik. Regression conformal prediction with random forests. 2014. *Machine Learning*. DOI: 10.1007/s10994-014-5453-0. URL: https://www.researchgate.net/publication/264346830_Regression_conformal_prediction_with_random_forests

[27] Berger Bonnie, Daniels Noah M, Yu Y William. Computational Biology in the 21st Century: Scaling with Compressive Algorithms. 2016. *Communication of the ACM*. DOI: 10.1145/2957324. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5615407/

[28] Jeremy M Berg, John L Tymoczko, Lubert Stryer. Biochemistry. Fifth Edition. 2013. URL: https://biokamikazi.files.wordpress.com/2013/10/biochemistry-stryer-5th-ed.pdf

[29] Alberts B, Johnson, Lewis J, Raff, Roberts, Walter. Molecular Biology of the cell. Fourth Edition. 2003. DOI: 10.1093/aob/mcg023. URL: https://www.researchgate.net/publication/274179232_Alberts_B_Johnson_A_Lewis_J_Raff_M_Roberts_K_and_Walter_P_Molecular_biology_of_the_cell_4th_edn

[30] Dobson, C. M. Principles of protein folding, misfolding, and aggregation. Seminars in Cell & Developmental Biology. 2004. URL: https://www.researchgate.net/publication/8664924_Principles_of_protein_folding_misfolding_and_aggregation

[31] Anfinsen, C. B. Principles that govern the folding of protein chains.1973. *Science, 181(4096), 223-230*. DOI: 10.1126/science.181.4096.223. URL: https://www.science.org/doi/10.1126/science.181.4096.223

[32] "Rank Cases: Ties". www.ibm.com. Retrieved 2023-07-23.